INFORMATION TECHNOLOGY ESSENTIALS NOTES SEMESTER II

UNIT 1- WEB ESSENTIALS

Creating a Website – Working principle of a Website – Browser fundamentals – Authoring tools – Types of servers: Application Server – Web Server – Database Server

1.1 CREATING WEB SITE

Web site is a collection of web pages. Hence for a website design we need to design the webpages. Each webpage may contain texts, photos, videos, and social media buttons and so on. Technically, a webpage is a special type of document written in scripting languages such as HTML,CSS, JavaScript, PHP and so on. Web pages are written for web browsers. The web browsers are the programs like Internet Explorer, Google Chrome and Safari. These browsers have a simple but crucially important job: they read the web page document and display the perfectly formatted result.

Definition of website:

Website is a collection of webpages that are grouped together to achieve certain task under single domain name.

Why do people visit website?

The most important reason is to find the **required information.** This could be anything from a student looking for images for a school project, to finding the latest stock quotes, for getting the address of the nearest restaurant and so on. **To complete a task:** Visitors may want to buy the latest best-seller, download a software program, or participate in an online discussion about a favourite hobby.

Steps for creating the Website:

Step 1: Website creation:

Create a webpage using suitable scripting language. IF any image is associated with this web page then convert this image into appropriate format (JPEG or GIF is preferable). Embed this image appropriately in this webpage.

Step 2: Choosing the web hosting services

- Web hosting company hosts your webpages on web server. Thus your website will be available to anyone who knows your URL. Most web hosting companies offer hosting services for both personal and business use. The web host provides you with Internet access, email accounts and space for a personal or business website.
- If you are building a website for business use, your webhost can register a personalized domain name for your website. If you are building a website for business use, your web host can register a personalized domain name for your website.
- Small web sites (around 15-20 pages of contents) do not need much more than 1 or 2 MB of server space that hold all the HTML pages and graphics. Your web hosting package should provide atleast MB of space so your web page has room to grow.

Step 3: Registering Domain Name

Domain name is an alias that points to actual location of your web site on web server. Domain names are managed by the Internet Corporation for Assigned Names and Numbers (ICANN). ICANN has agreements with a number of vendors to provide domain name registration services.

Step 4: Planning your website

- **Type**: The type of site you need. Is this a news or informational site, a site for a company or service, a non-profit or cause driven site, an E-commerce shop etc. Each of these kinds of site has a slightly different focus that will influence its design.

- **Navigation**: Navigation means indication that how users will move around your site affects its information architecture as well as the overall usability of that site. Plan out the pages a site, create a sitemap, and develop a navigational structure from there.
- **Content**: The quality of your site's content will play an important role in it's success. Content is everything that your pages will contain, such as text, images, video and more. Before you start designing or building pages, you should have a clear strategy for the content that those pages will contain.

Step 5: Uploading Files

- To publish a website on the web, you must send the web pages created by you on the webserver using File Transfer Protocol (FTP). Using some software such as Microsoft Visual Studio or Adobe Dreamweaver one can upload the files on the webserver.

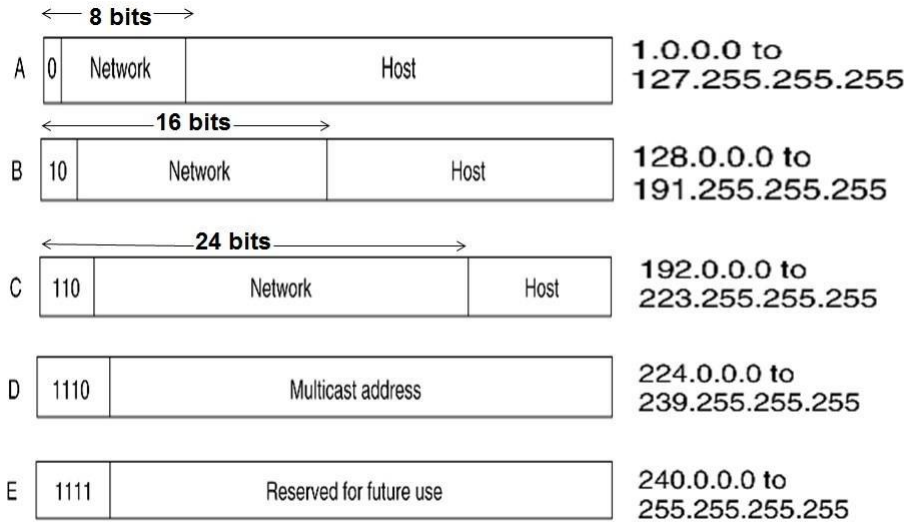
Testing the website:

Testing must be performed throughout the development of website

- **Multiple Browsers**: It is necessary to display the website on as many web browsers as possible to ensure that the contents of the website are consistently displayed and the work done is portable.
- **Multiple Operating Systems:** It is necessary to display the website on different operating systems.
- **Connection Speed:** Do not rely on the same connection speed when testing your website, especially if you work in a corporate environment where the connection to the Internet usually is faster than the average user's. Also test the download time for different connection speed.
- **Device Types**: Test the websites on the computer's having different screen size. It is necessary to ensure that pages are displayed consistently on all screen size.
- **Links:** Use a link validation tool to ensure that all of your links connect to a live page.
- **Security Testing:** This step is necessary to test the security vulnerabilities in application running on the website. Security is an important part of any web development plan.

1.2 IP ADDRESSING

- Each host on a TCP/IP network is assigned a unique 32-bit logical address that is divided into two main parts: the network number and the host number. This is called IP address. The IP address is grouped four into 8-bits separated by dots. Each bit in the octet has binary weight. There are five classes based on two categories, A, B, C, D and E.



- IP address is assigned to the devices participating in computer network. The IP protocol makes use of this address for communication between two computers. Using IP address particular node can be identified in the network.

1.3 DNS

- It is very difficult to remember numerical information but it is simple to remember the textual information. Consider that we want to access Priyanka's PC, then accessing it using the IP address www. 192.168.0.101 is definitely not comfortable, rather if we have the address www.priyanka@technical.com then accessing and remembering Priyanka's PC address is very simple. The names which are used to identify computer within a network are called domain names. Thus domain name is the name given to a network for human reference. Hence in DNS, instead of using the IP address name of the computer is used to access it. But two names can be the same. Hence to uniquely identify your computer the name must be referred using DNS hierarchy. Before understanding the hierarchy the commonly used domain names are:

Domain Names	Purpose
Com	Commercial organization
Gov	Government organization
Edu	Educational institutions
Int	International organization
Net	Network group
Org	Non profit organization
Mil	Military organization
In	Sub domain name used to refer India
Uk	Sub domain name used to refer uk
Jp	Sub domain name used to refer japan

- The domain name space is used to locate the computer uniquely. The internet logically arranges the domain names in an hierarchical form. There are some top level DNS such as com, org, edu. Etc. Then each domain is divided into sub-domains and then sub sub-domains and so on. For example the complete path for http://www.cse.tec.ac.in can be uniquely traced out the help of domain name space.

Working of DNS

There are two tasks that can be carried out by DNS servers:

- Accepting and then requesting the programs to convert domain names to IP address.
- Accepting and then requesting the other DNS servers to convert domain names to IP address.

Suppose PC A is interested in knowing the IP address of technicalpublications.org then it contacts nearest DNS server. This DNS server maintains huge database of domain names. The entry domain name technicalpublication.org is searched within this database and if the IP address for corresponding name is found then the IP address is returned to PC A. If it is not found, then the name of the another DNS is suggested. If the request is made for some invalid domain name then the error message is returned.

1.4 URL

The Uniform Resource Locator (URL) is unique address for the file that has to be accessed over the internet. When we want to access some website we enter it's URL in the address bar of the web browser. For example if we want to access www.google.com then we must specify its URL in the address bar. However any other file such as some text file or image file or some HTML file can also be specified. The URL contains names of the protocol such as http://. The URL may contain the names of the protocol such as ftp. For example : ftp://ftp.funet.fi/pub/standards/RFC/rfc2166.txt The protocol identifier and the resource name are separated by a colon and two forward slashes. The syntax of writing URL is given below: protocol://username@hostname/path/filename. Sometimes instead of domain name servers IP addresses can also be used, for example http://192.168.0.1. But use of IP address as URL is not preferred because human cannot remember numbers very easily but they can remember names easily.

Absolute and Relative URL

- The absolute URL is a URL which directly point to a file. It exactly specifies exact location of a file or directory on the internet . Each absolute URL is unique.
  For example: http://www.vtubooks.com/home.aspx
- The relative URL points to the file or a directory in relation to the present directory.
  For example: http://www.webie.com/myphotos/mother.jpg

1.5 WORKING PRINCIPLE OF A WEBSITE

Features of Website Design

- **Quality of Web content** – people desire information in fast and reliable fashion. For business websites, content should include important information. These type of websites need to display high quality pictures of their products, and the highlight for clients testimonials.
- **Clear, User- friendly navigation** – A user friendly navigation scheme allows visitors to quickly find the information needed. Important links must be easy to find and given logical, simple and includes easy to understand labels.
- **Simple and professional design** - The website design must be simple and professional. Google is an excellent example of such a site. To keep websites simple a balances distribution of contents and graphics is required. The use of slightly contrasting colours, and clear fonts is necessary. Also, one should break up sizeable blocks of texts with either spacing or images as appropriate.
- **Webpage speed** – People inherently lose patience quickly, when visiting a website. The website with heavy graphics, audio and video takes more time to load. A web design company must take care of all the controlling factors that will maintain the desirable speed of the website.
- **Search Engine optimization** – A well-designed website generally will receive many visitors, and one method to attract visitors is search engine optimization. This allows the insertion of search keywords in website content, an appropriate link profile, social media signals.
- **Web compatibility** – A website should easily render on various resolutions, screen sizes and browsers and with the increasing popularity of mobile devices, websites should function properly on these types of devices.

Web site Design Issues

- **Simplicity** – It is a general tendency of web designers to provide lot of animations, huge amount of information, extreme visuals and so on. This makes the web design enormous and it should be avoided. The web application must be moderate and simple.

- **Identity** – Web design must be based on the nature of the web application. It is driven by the objective of the web application, category of user using it. A web engineer must work to establish an identity for the web application through the design.
- **Consistency** – The contents of the web application should be constructed consistently. For example: text formatting, font style should be the same overall the text document of the web application. Similarly, the graphics design, color scheme and style must be identical over all the web pages of the web application. Navigation mechanism must be used consistently across web application elements.
- **Robustness** – The users always expects robust contents and functions of the web application. That means any required functionality should not be missing at all. If any function or content is missing or insufficient then that web application will fail.
- **Navigability** – The navigation should be simple and consistent. The design of navigations should intuitive and predictable in nature. That means any user should be in a position to make use of navigation links without any help.
- **Visual Appeal** – The web application are most visual and most dynamic and aesthetic in nature. There are various factors that contribute to visual appeal. The factors are – look and feel of the content, interface layout, color co-ordination, the balance of text, graphics and other media, navigation mechanism and so on.
- **Compatibility –** The web application can be used in variety of environment and configurations such as different browsers, internet connection types, operating systems and various browsers.

1.6 PHASES OF WEBSITE DEVELOPMENT

Web project can be designed in the four phases as given below -

Phase I: Strategy

In this phase, a strategic planner or project manager along with the client determines the objectives of the site. As an output of this phase **creative briefs** are prepared. The creative brief is a kind of document in which project objectives, requirements and key insights are clearly mentioned. Every team member makes use of creative brief as a guidelines fo the development.

Phase II: Design

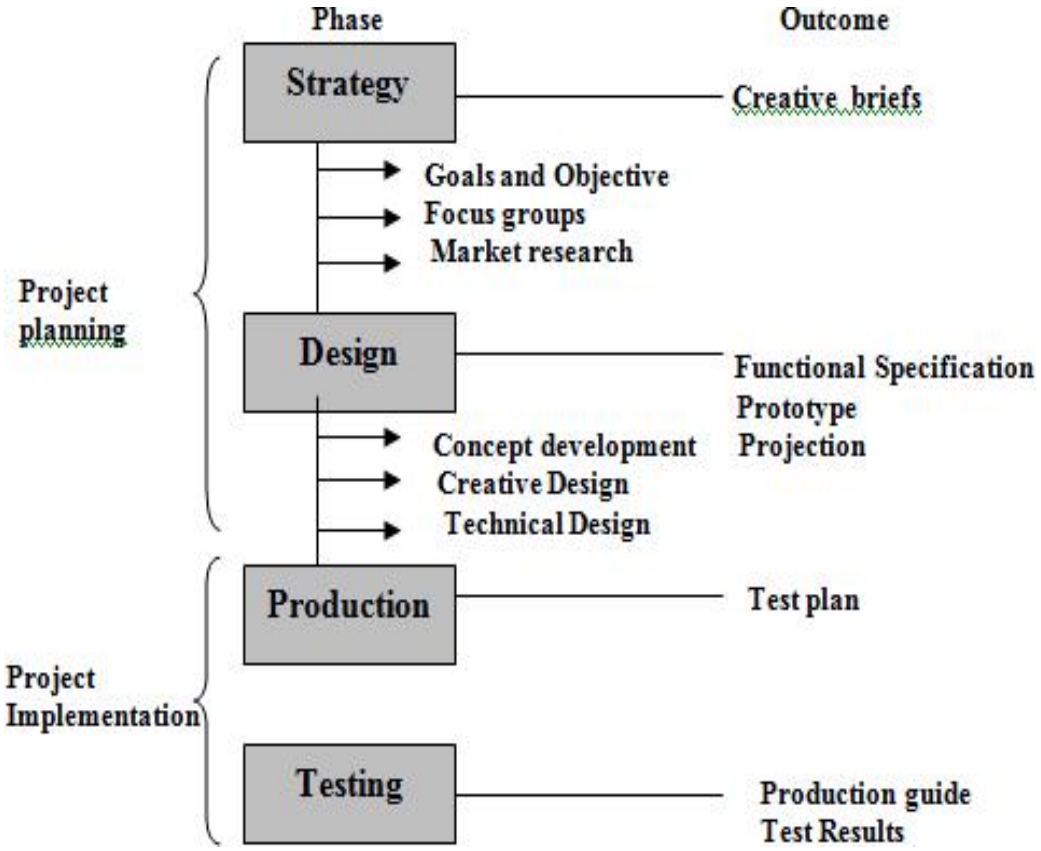
In this phase actual design of the website is done with the help of creative and technical team members. The front end is designed by the creative team in which user interface and interactions are designed. The back end is designed by the technical team which is responsible for designing the database architecture. As an outcome of this phase functional and technical specifications, site architecture are prepared.

Phase III: Production

During this phase actual site is built using the source code. Functionalities and features of the website are closely examined. If the client demands for a change in any functionality then a change order is issued. At the end of this phase a production guide is prepared.

Phase IV: Testing

At this phase all the functionalities and features of the website are tested, bugs are identified and resolved before launching the website. The QA manager develops the test plan. The test suit mentioned in it used to test the product thoroughly.



1.7 ENHANCING WEBSITE

There are varieties of ways by which one can enhance his website. The website can be enhanced using some key elements such as -

- **Contents -** This is the most important element of website. It helps to spread the business message in an appropriate manner. The content should be easy to understand. Those should be to the point and relevant. The information available on the website must be useful to the user.
- **Graphics -** Adding too much graphics in the webpage slows down its speed of loading. Hence Graphics is undesirable by any user. However the relevant and appealing graphics can be added to the website.
- **Color and Text -** The colours and text that is appearing on the website must be pleasant to the eyes. As a rule of thumb, the entire site must use at the most five to six colors. The text should not be too small or too large. The selection of the family of font for displaying the text must be appropriate so that the text can be readable.
- **Flash –** Use of flash animation makes the site attractive but at the same time there are many drawbacks that are associated with this key element. The first drawback is the flash files take a large amount time to load the data on the web page. Secondly if the flash animation is placed on the web page then the link for downloading the flash player must also be provided so that the animation can be viewed by the user.
- **Frames –** Frames must be avoided while designing the website. Instead of using frames the web designer must prefer the tables. The reason why the use of frames must be avoided in the web page is that – the search engine find it difficult to search the contents from the site containing the frames.
- **Organizing Files -** The files required by the website must be categorized and must be stored in sorted manner. This makes it easier to manage the information.

1.8 BROWSER FUNDAMENTALS

Definition:

Web browser is a kind of software which is basically used to use resources on the web.

- Over the networks, two computers communicate with each other. In this communication, when request is made by one computer then that computer is called **a client** and when the request gets served by another computer then that computer is called **server**. Thus exchange of information takes place via client-Server communication.
- When user wants some web document then he makes the request for it using the web browser. The browsers are the programs that are running on the client's machines. The request then gets served by the server and the requested page is then returned to the client. It is getting displayed to the client on the web browser. The web browser can browse the information on the server and hence is the name.
- Various web browsers that are commonly used are

Browser	**Vendor**
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp
Opera	Opera Software
Safari	Apple

- Web browser supports variety of protocols but the most commonly used protocol on the web browser is **Hyper Text Transfer Protocol (HTTP).** This protocol is typically used when browser communicates with the server.

Functions of a Web Browser:

1.Takes the website URL entered by the user and converts it into a proper HTTP request.

2.Converts the domain name (like www.example.com) into its corresponding IP address using DNS.

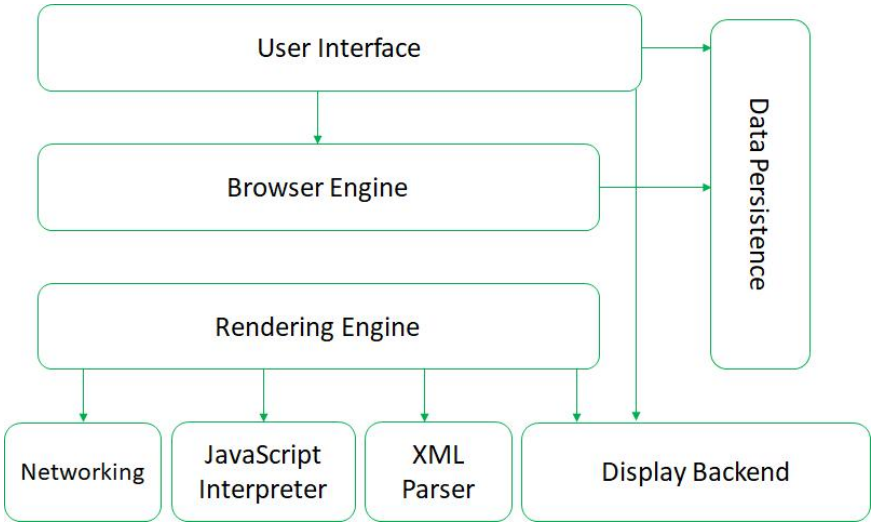
3.Establishes a TCP connection with the web server to process the request.

4.Sends the HTTP request to the web server.

5.Receives the response (web page) from the web server and displays it on the user's screen in a readable format.

Web Browser Architecture

The web browser architecture is represented by following figure –



The main components of web browser architecture are as follows -

User Interface:

Using the user interface user interacts with the browser engine. The user interface contains, Address bar, back/forward button, book mark menu and so on. The page requested by the user is displayed in this user interface.

Browser Engine:

It contains the mechanism by which the input of user interface is communicated to the Rendering Engine. The browser engine is responsible for querying the rendering engine according to various user interfaces.

Rendering Engine:

It is responsible for displaying the requested contents on the screen. The rendering engine interprets the HTML, XML and JavaScript that comprises the given URL and generates the layout that is displayed in the user interface. The main components of rendering engine are HTML parser. The job of the HTML parser is to parse the HTML mark-up into a parse tree. It is important to note that Chrome, unlike most browsers, holds multiple instances of the rendering engine – one for each tab, each tab is a separate process. Different browsers use different rendering engines – Internet Explorer uses Trident, Firefox uses Gecko, Safari uses Webkit, Chrome and Opera uses WebKit.

Networking:

The functionality of networking is to retrieve the URL using common internet protocols such as HTTP and FTP. The networking is responsible to handle the internet communication and security issues. The network component may use the cache for retrieved documents. This feature is useful for increasing the response time.

JavaScript Interpreter:

The interpreter executes the JavaScript code which is embedded in a web page.

User Interface Backend:

It is basically used to draw the widgets like combo boxes and windows.

Data Persistence:

This is a small database created on local drives of the computer where the browser is installed. The data storage manages user data such as book marks, cookies and preferences.

Working of Web Browser

We often browse the internet for several reasons. It is more interesting to know about how a web page demanded by us gets displayed on our browser. Following is the stepwise explanation of this process -

Step 1:

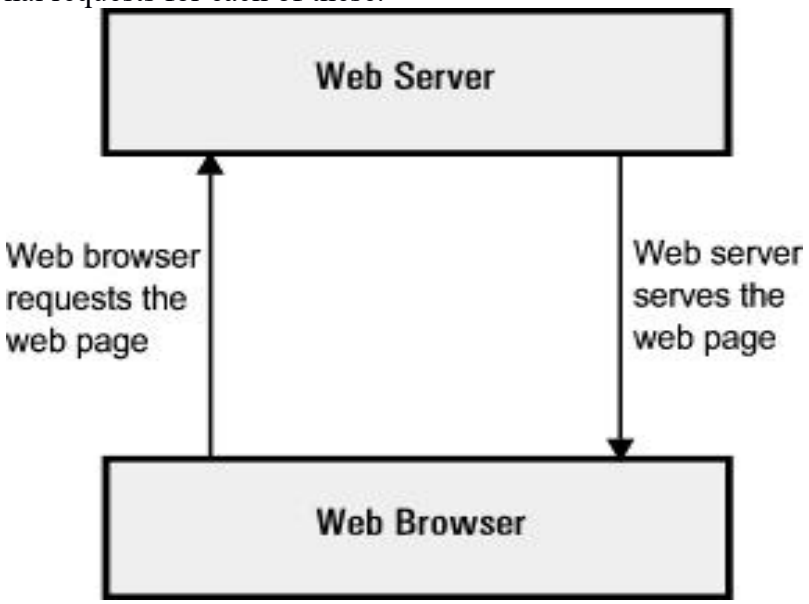
- First user types the website address for demanding the desired web page for example - **http://www.vtubooks.com/home.aspx** and then the home page of this website appears on the screen.
- The web address is divided into three parts:
  (i) The first part is the protocol. The **http** is a hypertext transfer protocol which tells the web browser that user wishes to communicate with web server on **port 80**. Port 80 is reserved for the communication between web server and web browser.
- The second part is the server address. This tells the web browser which server it needs to contact in order to retrieve the information you are looking for. The web browser communicates with a **Domain Name Server (DNS)** to find out the IP Address for the website. All communications on the internet use IP Addresses for communications. Use of the numeric address for accessing the web server is avoided because it is easier to remember textual information than that of numeric one. Hence normally the web server's addresses are textual.
- The third part of this address donates the resource user wants to see.

Step 2:

The web browser, on locating the IP Address which it requires (by communicating with the name server), send a request directly to the web server, using port 80, asking for the file **home.aspx**.

Step 3:

The web server sends the html for this page back to user's web browser. If there are additional files needed in order to show the web page (like some images for example) the web browser makes additional requests for each of these.



Basic features of Web Browsers

1. Web browsers should be able to look at the web pages throughout internet or connect to various sites to access information.
2. The Web browser must enable you to follow the hyperlinks on a Web and type in a URL for it to follow. One of the main features of a browser is to search the information on the current page as well as search the WWW itself.
3. Browser give you the facility to save a web page in a file on your computer, print a Web page and send the contents of a Web page e-Mail to others on the internet.
4. Web browser should be able to handle text, images of the World Wide Web, as well as the hyperlinks to digital video, or other types of information.
5. Web browsers interact not just with the Web, but also with your computer's operating system and with other programs, called plug-ins that gives the browser enhanced features.

6. Another important features to insist on in your browser is **caching**. A browser that caches keeps of the pages you visit so that it does not have to download them again if you want to return to them. Reloading a page from the cache is much quicker that downloading it again from the original source.
7. The most important feature of any browser is ease of use. While all Web browsers are fundamentally simple to use, it makes user comfortable.

Comparisons among popular Web Browsers:

S.No	Features	Firefox	Chrome	Internet Explorer
1.	Fast JavaScript engine for better performance	Yes	Yes	Yes
2.	Notifications when add-ons slow browser performance	No	No	Yes
3.	Simple browsing controls for better browsing expierence	Yes	Yes	Yes
4.	Combined search and address bar	No	Yes	Yes
5.	Protection from malicious cross-site scripting attacks	Yes	Yes	Yes
6.	Automatic recovery of crashed tabs	Yes	Yes	Yes
7.	Compatibility mode to view websites designed for web browsers	No	No	Yes
8.	Developer tools built-in to the browser	No	Yes	Yes
9.	Reopen accidently crashed tab	No	Yes	Yes
10.	Best protection against the phishing attacks	Yes	No	Yes
11.	Different operating system	Windows, Linux, MAC	Windows, Linux, MAC	Windows, MAC

1.9 HTTP PROTOCOL

- **Hyper Text Transfer Protocol (HTTP)** takes part in web browser and web server communication. Hence it is called a **Communication protocol.** The basic features of HTTP protocol are that it follows the **request response model.** The client makes a request for desired web page by giving the URL in the address bar. This request is submitted to the web server and then web server gives response to the web browser by returning the required web page.

HTTP Request Message Structure

The basic structure of request message is given by following general form -

<start line>
<Header fields>
<Blank Line>
<Message Body>

Let us discuss this structure in detail:

Start line

The **start line** consist of three parts which are separated by a single space. These parts are -

1) Request method 2) Request-URI 3) HTTP version

Request method:

The method defines the CONNECT method which is used during the web browser and server communication. It is always written in Upper Case letters. The primary method in HTTP is **GET.** The GET method is used when -

1. You type a URL in address bar.
2. When you click on some hyperlink which is present in the document.
3. When browser downloads images for display within a HTML document.

There is another commonly used method and i.e POST. The POST method is typically used to send information collected from a user form. Various methods used by HTTP are as given below-

HTTP Methods

Name	Description
GET	The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
HEAD	Same as GET, but it transfers the status line and the header section only.
POST	A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
PUT	Replaces all the current representations of the target resource with the uploaded content.
DELETE	Removes all the current representations of the target resource given by URI.
CONNECT	Establishes a tunnel to the server identified by a given URI.
OPTIONS	Describe the communication options for the target resource.
TRACE	Performs a message loop back test along with the path to the target resource.

Request URI

The **U**niform **R**esource **I**dentifier (URI) is a string used to identify the names or resources on the Internet. The URI is a combination of URL and URN. The URL stands for **U**niform **R**esource **L**ocator and URN stands for **U**niform **R**esource **N**ame. The web address denotes the URL and specific name of the place or a person or item denotes the URN. For examples

Urn :ISBN 978-81-8431-123-2 specifies the address of some book.

Every URI consist of two parts, the part before the colon: denotes the scheme and the part after colon depend upon the **scheme**. The URIs is case insensitive but generally written in lower case. If the URI is written in the form of http: then it is both an URI and URL but there are some other URI which can also be used as URL. For example

URL	Intended Server
ftp://ftp.mywebsite.com/index.txt	File can be located on FTP server
telnet://mywebsite.org	Telnet Server
mailtomyself@ mywebsite.org	Mail Box
http://www.mywebsite.org	Web Server

HTTP Version:
The first HTTP version was HTTP/0.9 but the official version of HTTP was HTTP/1.1.

Header Fields and Message body
The host header filed is associated with the HTTP request. The header files are in the form of field name and field value. Thus typical structure of http request is given in the diagram.

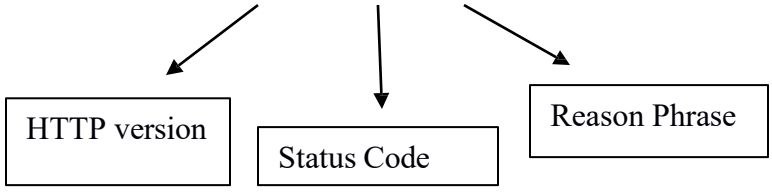
HTTP Request Message Structure:

HTTP Response Message Structure:
The structure of response message is similar to the request message structure. It is as follows
<status line>
<Header fields>
<Blank Line>
<Message Body>

Status Line:
Status line is similar to the start line in the request message. It consists of three fields.

HTTP Version	Status code	Reason phrase

The HTTP version denotes the HTTP version such as HTTP/1.1. The status code is a numeric code indicating the type of response. The reason phrase is in the text string form and presents the information about the status code.
For example –
HTTP / 1.1 200 OK



Following table explains some commonly used status codes:

Status Code	Reason Phrase	Description
200	OK	This is a Standard response for request
201	Created	It shows that the request is fulfilled and a new resource is being created
202	Accepted	When the request is accepted for processing but is not processed yet is denoted by this status code.
301	Moved Permanently	The URI for requested resource is moved at some another location.
401	Unauthorized	The requested resource is protected by some passwords and the usedr has not provided any password.
403	Forbidden	The requested resource is present on the server but the server is not able to respond it.

The header field in the response message is similar to that of the request message. The message body consists of response message.

For example
HTTP/1.1 200 OK
Date: Fri, 1 Jan 2010 07:59:01 GMT
Server:Apache/2.0.50 (Unix) mod_perl/1.99_10 perl/v5.8.4
Mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8
Last-Modified: Mon, 23 Feb 2009 08:32:41 GMT
Accept-Ranges: bytes
Conten-Length:2010
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>…</html>

The response header fields are enlisted in the following table:

Header field	Description
Date	It represents the date and thme at which the response is generated
Server	The name of the server which is responding.
Last-Modified	The date and time at which the response is last modified.
Accept-ranges	It specifies the unit which is used by the client to accept the range request.For example if there is a large document and only a single web page is currently needed then this specifies the Accept-range

Cache Control:
Many times the response header files are used in conjunction with cache control. Cache is used as a repository. Use of cache improves the system performance. Many browsers stores web pages viewed by the client in the cache memory. This brings efficiency in browsing web pages. For

instance, client reads a daily Newspaper on his PC then caching the corresponding web address or web pages will quickly display the web page.

HTTP Tunnelling:

HTTP Tunnelling is a mechanism by which the communication performed by various network protocol is encapsulated by the HTTP Protocol. HTTP tunnelling can be used in the chat like applications for communications from network locations with restricted connectivity.

The application that wishes to communicate with a remote host opens an HTTP connection to a mediator server. Using HTTP request the host communicates with the mediator server by encapsulating the actual communications within those requests. The mediator server then unwraps the data and sends to the remote destined hosts. The remote host when sends the response to the requesting hosts, wraps the response in the HTTP protocol and then the response is given. In this case the application becomes the tunnelling point.

Features of HTTP Protocol:

1. It is a communication protocol used between the web browser and web server.
2. This protocol is based on request-response messaging. That means client makes the request of desired web pages and then the server responds it by sending the requested resource.
3. It is a stateless protocol. That means HTTP protocol cannot remember the previous user's information not it remembers the number of times the user has visited particular website.
4. The request-response message consists of plain text fairly in readable form.
5. The HTTP protocol has a cache control. This is an advanced feature of HTTP. Most of the web browsers automatically store the recently visited pages. This is very useful feature because if the user requests the same web pages that have been visited already then it can be displayed from the cache memory instead of requesting the web server and bringing it from there.

1.10 AUTHORING TOOLS

Definition:

A web authoring tool is a software package which developers use to create and package e-learning content deliverable to end users. The multimedia authoring tools provide the capability for creating a complete multimedia presentation, including interactive user control.

Some examples of the authoring tools are:

1. Macromedia Flash
2. Macromedia Director
3. Author ware
4. Quest

Authoring software provides an integrated environment for combining the content and functions of a project. It enables the developer to create, edit, and import data. In multimedia authoring tools, multimedia elements and events are considered as object. Each object is assigned properties and modifiers. On receiving messages, the objects perform tasks depending upon properties and modifiers.

Features of Authoring tools:

1. Programming Features:

Authoring tools offer the programming using high level languages or support for scripting environment. The tools that offer the programming features are Macromedia Flash, HyperCard, Metacard and Tool Book. Some authoring tools offer direct importing of preformatted text, including facilities, complex text search mechanisms and hyper linkage tools. Visual authoring tools such as Author ware and Icon Author are suitable for slideshows and presentations.

2. Interactivity features:
The interactivity feature allows the user to have control flow of information. Using the interactivity features the contents are well organized by the user. The conditional branching support the complex programming logic, subroutines, event tracking and message passing among objects and elements.

3. Editing and organizing features:
The elements of multimedia – image, animation, text, digital audio and MIDI music and video clips need to be created, edited and converted to standard file formats and the specialized applications provide these capabilities. Editing tools for these elements, particularly text and still images are often included in as authoring tools. Some authoring tools provide a visual flowcharts system or overview facility for illustrating your project's structure at a macro level. Storyboards navigation diagrams too can help organize a project.

4. Delivery Features:
Delivering your project may require building run time version of the project using the multimedia authoring software. A run time version allows your project to play back without requiring the full authoring Multimedia Systems software and all its tools and editors. Many times the run time version does not allow user to access or change the content, structure and programming of the project.

5. Cross Platform feature:
By this feature, it is possible to transfer the content across the platform easily. The run time players are available for providing the compatibility to the authoring tool sto work in other platforms.

Examples of Authoring Tools:

1. Macromedia Flash:
Adobe Flash Player is a multimedia platform which has become the standard for implementing animation and interactivity into web pages to create ads, integrate video into websites.

2. HyperCard:
It is a hypermedia program created for Macintosh Computer. It combines database abilities with a graphical, flexible, user-modifiable interface. HyperCard also features Hyper talk, a programming language for manipulating data and the user interface.

3. Front Page:
It is a website administration tool from Microsoft for the Microsoft windows. FrontPage consists of a Split View option to allow the user to code in code view and preview in design view without the hassle of switching from the design view and code view tabs for each review. Interactive Buttons gives users a new way to create web graphics for navigation and links eliminating the need for a complicated package.

4. Dreamweaver:
Dreamweaver is a web authoring tool rather than a multimedia tool. It does however support a wide range of multimedia file types. These include graphics formats such a s JPEG, GIF, PNS as well as Short wave files. Support exists for embedding other media such as audio and video within HTML or a script. A range of interactive elements are pre-scripted as behaviours, including some that can be used for multimedia and interactivity. An extensive range of languages including HTML, XML, ASP, PHP, JSP, JavaScript and VBScript are supported by Dreamweaver.

Address bar, back/forward button, book mark menu and so on. The page requested by the user is displayed in this user interface.

Browser Engine:

It contains the mechanism by which the input of user interface is communicated to the Rendering Engine. The browser engine is responsible for querying the rendering engine according to various user interfaces.

Rendering Engine:

It is responsible for displaying the requested contents on the screen. The rendering engine interprets the HTML, XML and JavaScript that comprises the given URL and generates the layout that is displayed in the user interface. The main components of rendering engine are HTML parser. The job of the HTML parser is to parse the HTML mark-up into a parse tree. It is important to note that Chrome, unlike most browsers, holds multiple instances of the rendering engine – one for each tab, each tab is a separate process. Different browsers use different rendering engines – Internet Explorer uses Trident, Firefox uses Gecko, Safari uses Webkit, Chrome and Opera uses WebKit.

Networking:

The functionality of networking is to retrieve the URL using common internet protocols such as HTTP and FTP. The networking is responsible to handle the internet communication and security issues. The network component may use the cache for retrieved documents. This feature is useful for increasing the response time.

JavaScript Interpreter:

The interpreter executes the JavaScript code which is embedded in a web page.

User Interface Backend:

It is basically used to draw the widgets like combo boxes and windows.

Data Persistence:

This is a small database created on local drives of the computer where the browser is installed. The data storage manages user data such as book marks, cookies and preferences.

Working of Web Browser

We often browse the internet for several reasons. It is more interesting to know about how a web page demanded by us gets displayed on our browser. Following is the stepwise explanation of this process -

Step 1:

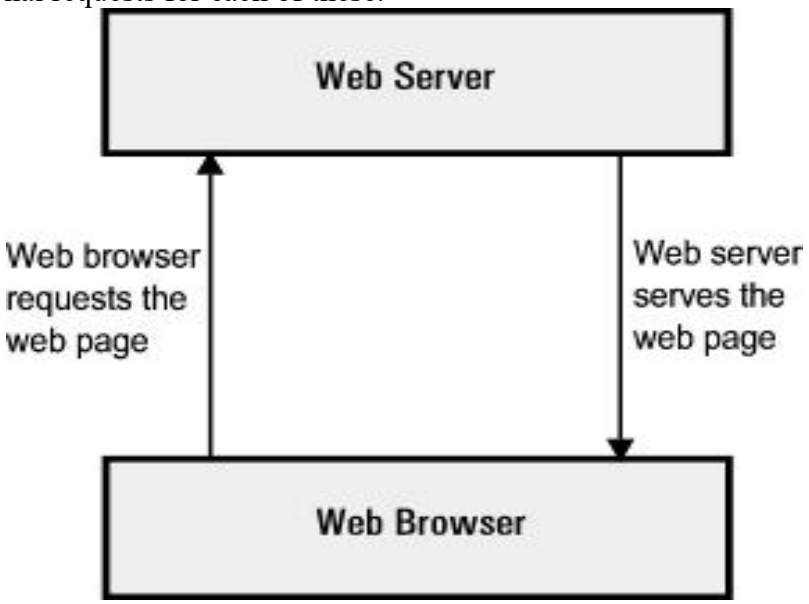
- First user types the website address for demanding the desired web page for example - **http://www.vtubooks.com/home.aspx** and then the home page of this website appears on the screen.
- The web address is divided into three parts:
  (i) The first part is the protocol. The **http** is a hypertext transfer protocol which tells the web browser that user wishes to communicate with web server on **port 80**. Port 80 is reserved for the communication between web server and web browser.
- The second part is the server address. This tells the web browser which server it needs to contact in order to retrieve the information you are looking for. The web browser communicates with a **Domain Name Server (DNS)** to find out the IP Address for the website. All communications on the internet use IP Addresses for communications. Use of the numeric address for accessing the web server is avoided because it is easier to remember textual information than that of numeric one. Hence normally the web server's addresses are textual.
- The third part of this address donates the resource user wants to see.

Step 2:

The web browser, on locating the IP Address which it requires (by communicating with the name server), send a request directly to the web server, using port 80, asking for the file **home.aspx**.

Step 3:

The web server sends the html for this page back to user's web browser. If there are additional files needed in order to show the web page (like some images for example) the web browser makes additional requests for each of these.



Basic features of Web Browsers

8. Web browsers should be able to look at the web pages throughout internet or connect to various sites to access information.
9. The Web browser must enable you to follow the hyperlinks on a Web and type in a URL for it to follow. One of the main features of a browser is to search the information on the current page as well as search the WWW itself.
10. Browser give you the facility to save a web page in a file on your computer, print a Web page and send the contents of a Web page e-Mail to others on the internet.
11. Web browser should be able to handle text, images of the World Wide Web, as well as the hyperlinks to digital video, or other types of information.
12. Web browsers interact not just with the Web, but also with your computer's operating system and with other programs, called plug-ins that gives the browser enhanced features.

13. Another important features to insist on in your browser is **caching**. A browser that caches keeps of the pages you visit so that it does not have to download them again if you want to return to them. Reloading a page from the cache is much quicker that downloading it again from the original source.

14. The most important feature of any browser is ease of use. While all Web browsers are fundamentally simple to use, it makes user comfortable.

Comparisons among popular Web Browsers:

S.No	Features	Firefox	Chrome	Internet Explorer
1.	Fast JavaScript engine for better performance	Yes	Yes	Yes
2.	Notifications when add-ons slow browser performance	No	No	Yes
3.	Simple browsing controls for better browsing expierence	Yes	Yes	Yes
4.	Combined search and address bar	No	Yes	Yes
5.	Protection from malicious cross-site scripting attacks	Yes	Yes	Yes
6.	Automatic recovery of crashed tabs	Yes	Yes	Yes
7.	Compatibility mode to view websites designed for web browsers	No	No	Yes
8.	Developer tools built-in to the browser	No	Yes	Yes
9.	Reopen accidently crashed tab	No	Yes	Yes
10.	Best protection against the phishing attacks	Yes	No	Yes
11.	Different operating system	Windows, Linux, MAC	Windows, Linux, MAC	Windows, MAC

1.11 HTTP PROTOCOL

- **Hyper Text Transfer Protocol (HTTP)** takes part in web browser and web server communication. Hence it is called a **Communication protocol.** The basic features of HTTP protocol are that it follows the **request response model.** The client makes a request for desired web page by giving the URL in the address bar. This request is submitted to the web server and then web server gives response to the web browser by returning the required web page.

HTTP Request Message Structure

The basic structure of request message is given by following general form -

<start line>
<Header fields>
<Blank Line>
<Message Body>

Let us discuss this structure in detail:

Start line

The **start line** consist of three parts which are separated by a single space. These parts are -

1) Request method 2) Request-URI 3) HTTP version

Request method:

The method defines the CONNECT method which is used during the web browser and server communication. It is always written in Upper Case letters. The primary method in HTTP is **GET.** The GET method is used when -

4. You type a URL in address bar.
5. When you click on some hyperlink which is present in the document.
6. When browser downloads images for display within a HTML document.

There is another commonly used method and i.e POST. The POST method is typically used to send information collected from a user form. Various methods used by HTTP are as given below-

HTTP Methods

Name	Description
GET	The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
HEAD	Same as GET, but it transfers the status line and the header section only.
POST	A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
PUT	Replaces all the current representations of the target resource with the uploaded content.
DELETE	Removes all the current representations of the target resource given by URI.
CONNECT	Establishes a tunnel to the server identified by a given URI.
OPTIONS	Describe the communication options for the target resource.
TRACE	Performs a message loop back test along with the path to the target resource.

Request URI

The **U**niform **R**esource **I**dentifier (URI) is a string used to identify the names or resources on the Internet. The URI is a combination of URL and URN. The URL stands for **U**niform **R**esource **L**ocator and URN stands for **U**niform **R**esource **N**ame. The web address denotes the URL and specific name of the place or a person or item denotes the URN. For examples

Urn :ISBN 978-81-8431-123-2 specifies the address of some book.

Every URI consist of two parts, the part before the colon: denotes the scheme and the part after colon depend upon the **scheme**. The URIs is case insensitive but generally written in lower case. If the URI is written in the form of http: then it is both an URI and URL but there are some other URI which can also be used as URL. For example

URL	Intended Server
ftp://ftp.mywebsite.com/index.txt	File can be located on FTP server
telnet://mywebsite.org	Telnet Server
mailtomyself@ mywebsite.org	Mail Box
http://www.mywebsite.org	Web Server

HTTP Version:
The first HTTP version was HTTP/0.9 but the official version of HTTP was HTTP/1.1.

Header Fields and Message body
The host header filed is associated with the HTTP request. The header files are in the form of field name and field value. Thus typical structure of http request is given in the diagram.

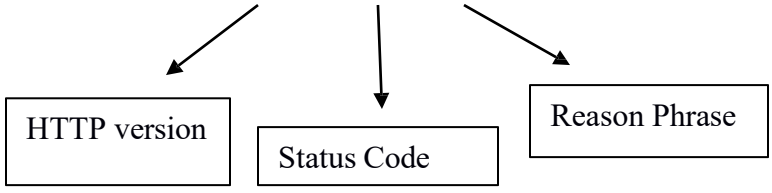
HTTP Request Message Structure:

HTTP Response Message Structure:
The structure of response message is similar to the request message structure. It is as follows
<status line>
<Header fields>
<Blank Line>
<Message Body>

Status Line:
Status line is similar to the start line in the request message. It consists of three fields.

HTTP Version	Status code	Reason phrase

The HTTP version denotes the HTTP version such as HTTP/1.1. The status code is a numeric code indicating the type of response. The reason phrase is in the text string form and presents the information about the status code.
For example –
HTTP / 1.1 200 OK



Following table explains some commonly used status codes:

Status Code	Reason Phrase	Description
200	OK	This is a Standard response for request
201	Created	It shows that the request is fulfilled and a new resource is being created
202	Accepted	When the request is accepted for processing but is not processed yet is denoted by this status code.
301	Moved Permanently	The URI for requested resource is moved at some another location.
401	Unauthorized	The requested resource is protected by some passwords and the usedr has not provided any password.
403	Forbidden	The requested resource is present on the server but the server is not able to respond it.

The header field in the response message is similar to that of the request message. The message body consists of response message.

For example
HTTP/1.1 200 OK
Date: Fri, 1 Jan 2010 07:59:01 GMT
Server:Apache/2.0.50 (Unix) mod_perl/1.99_10 perl/v5.8.4
Mod_ssl/2.0.50 OpenSSL/0.9.7d DAV/2 PHP/4.3.8
Last-Modified: Mon, 23 Feb 2009 08:32:41 GMT
Accept-Ranges: bytes
Conten-Length:2010
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>…</html>

The response header fields are enlisted in the following table:

Header field	Description
Date	It represents the date and thme at which the response is generated
Server	The name of the server which is responding.
Last-Modified	The date and time at which the response is last modified.
Accept-ranges	It specifies the unit which is used by the client to accept the range request.For example if there is a large document and only a single web page is currently needed then this specifies the Accept-range

Cache Control:
Many times the response header files are used in conjunction with cache control. Cache is used as a repository. Use of cache improves the system performance. Many browsers stores web pages viewed by the client in the cache memory. This brings efficiency in browsing web pages. For

instance, client reads a daily Newspaper on his PC then caching the corresponding web address or web pages will quickly display the web page.

HTTP Tunnelling:

HTTP Tunnelling is a mechanism by which the communication performed by various network protocol is encapsulated by the HTTP Protocol. HTTP tunnelling can be used in the chat like applications for communications from network locations with restricted connectivity.

The application that wishes to communicate with a remote host opens an HTTP connection to a mediator server. Using HTTP request the host communicates with the mediator server by encapsulating the actual communications within those requests. The mediator server then unwraps the data and sends to the remote destined hosts. The remote host when sends the response to the requesting hosts, wraps the response in the HTTP protocol and then the response is given. In this case the application becomes the tunnelling point.

Features of HTTP Protocol:

6. It is a communication protocol used between the web browser and web server.
7. This protocol is based on request-response messaging. That means client makes the request of desired web pages and then the server responds it by sending the requested resource.
8. It is a stateless protocol. That means HTTP protocol cannot remember the previous user's information not it remembers the number of times the user has visited particular website.
9. The request-response message consists of plain text fairly in readable form.
10. The HTTP protocol has a cache control. This is an advanced feature of HTTP. Most of the web browsers automatically store the recently visited pages. This is very useful feature because if the user requests the same web pages that have been visited already then it can be displayed from the cache memory instead of requesting the web server and bringing it from there.

1.12 AUTHORING TOOLS

Definition:

A web authoring tool is a software package which developers use to create and package e-learning content deliverable to end users. The multimedia authoring tools provide the capability for creating a complete multimedia presentation, including interactive user control.

Some examples of the authoring tools are:

5. Macromedia Flash
6. Macromedia Director
7. Author ware
8. Quest

Authoring software provides an integrated environment for combining the content and functions of a project. It enables the developer to create, edit, and import data. In multimedia authoring tools, multimedia elements and events are considered as object. Each object is assigned properties and modifiers. On receiving messages, the objects perform tasks depending upon properties and modifiers.

Features of Authoring tools:

6. Programming Features:

Authoring tools offer the programming using high level languages or support for scripting environment. The tools that offer the programming features are Macromedia Flash, HyperCard, Metacard and Tool Book. Some authoring tools offer direct importing of preformatted text, including facilities, complex text search mechanisms and hyper linkage tools. Visual authoring tools such as Author ware and Icon Author are suitable for slideshows and presentations.

7. Interactivity features:
The interactivity feature allows the user to have control flow of information. Using the interactivity features the contents are well organized by the user. The conditional branching support the complex programming logic, subroutines, event tracking and message passing among objects and elements.

8. Editing and organizing features:
The elements of multimedia – image, animation, text, digital audio and MIDI music and video clips need to be created, edited and converted to standard file formats and the specialized applications provide these capabilities. Editing tools for these elements, particularly text and still images are often included in as authoring tools. Some authoring tools provide a visual flowcharts system or overview facility for illustrating your project's structure at a macro level. Storyboards navigation diagrams too can help organize a project.

9. Delivery Features:
Delivering your project may require building run time version of the project using the multimedia authoring software. A run time version allows your project to play back without requiring the full authoring Multimedia Systems software and all its tools and editors. Many times the run time version does not allow user to access or change the content, structure and programming of the project.

10. Cross Platform feature:
By this feature, it is possible to transfer the content across the platform easily. The run time players are available for providing the compatibility to the authoring tool sto work in other platforms.

Examples of Authoring Tools:

5. Macromedia Flash:
Adobe Flash Player is a multimedia platform which has become the standard for implementing animation and interactivity into web pages to create ads, integrate video into websites.

6. HyperCard:
It is a hypermedia program created for Macintosh Computer. It combines database abilities with a graphical, flexible, user-modifiable interface. HyperCard also features Hyper talk, a programming language for manipulating data and the user interface.

7. Front Page:
It is a website administration tool from Microsoft for the Microsoft windows. FrontPage consists of a Split View option to allow the user to code in code view and preview in design view without the hassle of switching from the design view and code view tabs for each review. Interactive Buttons gives users a new way to create web graphics for navigation and links eliminating the need for a complicated package.

8. Dreamweaver:
Dreamweaver is a web authoring tool rather than a multimedia tool. It does however support a wide range of multimedia file types. These include graphics formats such a s JPEG, GIF, PNS as well as Short wave files. Support exists for embedding other media such as audio and video within HTML or a script. A range of interactive elements are pre-scripted as behaviours, including some that can be used for multimedia and interactivity. An extensive range of languages including HTML, XML, ASP, PHP, JSP, JavaScript and VBScript are supported by Dreamweaver.

9. Netobjects Fusion:
It is a tool that is a solution for small business websites, from planning, building and managing your site to promoting and growing online business quickly and effectively. One can drag images, text and other objects anywhere on the page and simply drop them in. The Netobjects fusion is the first program to remove the tedious hand coding from creating pixel-precise page layouts in HTML.

1.13 TYPES OF SERVER

Application Server - Web Server - Database Server

Web Server:

Web servers are computers that deliver (*serves up*) Web pages. Every Web server has an IP address and possibly a domain name. For example, if you enter the URL *http://www.webopedia.com/index.html* in your browser, this sends a request to the Web server whose domain name is *webopedia.com*. The server then fetches the page named *index.html* and sends it to your browser.

Any computer can be turned into a Web server by installing server software and connecting the machine to the Internet. There are many Web server software applications, including public domain software and commercial packages.

Functions of web server:

- The web server accepts the requests from the web browser.
- The user request is processed by the web server
- The web server responds to the users by providing the services which they demand for over the web browsers.
- The web servers serve the web based applications
- The DNS translate the domain names into the IP addresses
- The server verifies for the given address, finds the necessary files, runs appropriate scripts, exchange cookies if necessary and returns back to the browser
- Some servers actively participate in session handling techniques.

Examples of web servers: Apache web server, IIS web server

Apache web server	IIS web server
Apache web server is useful on both Unix based systems and on Windows platform	IIS web server is used on Windows Platform
It is an open source product that provides reliability and efficiency	It is vendor specific product and can be used on windows product only
The Apache web server can be controlled by editing the configuration file httpd.conf	For IIS web server, the behaviour is controlled by modifying the window based management programs called IIS snap-in. We can access IIS snap-in through the Control -Panel ->Administrative Tools
It is also called a free web server named as LAMP : (Linux/Apache/MySQL/PHP)	It is currently owned by Microsoft, and was designed with .NET frameworks.

Database Server:

Database is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organised into rows, columns and tables and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added.

Database Management is a piece of software that manages databases and lets you create, edit and delete databases.

DBMS examples include MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBase, Clipper, and FoxPro.

What is a database server?

It is similar to data warehouse where the website store or maintain their data and information. A Database Server is a computer in a LAN that is dedicated to database storage and retrieval. The database server holds the Database Management System (DBMS) and the databases. Upon requests from the client machines, it searches the database for selected records and passes them back over the network**.**

A database server can be defined as a server dedicated to providing database services. Such a server runs the database software. A database server can typically be seen in a client-server environment where it provides information sought by the client systems.

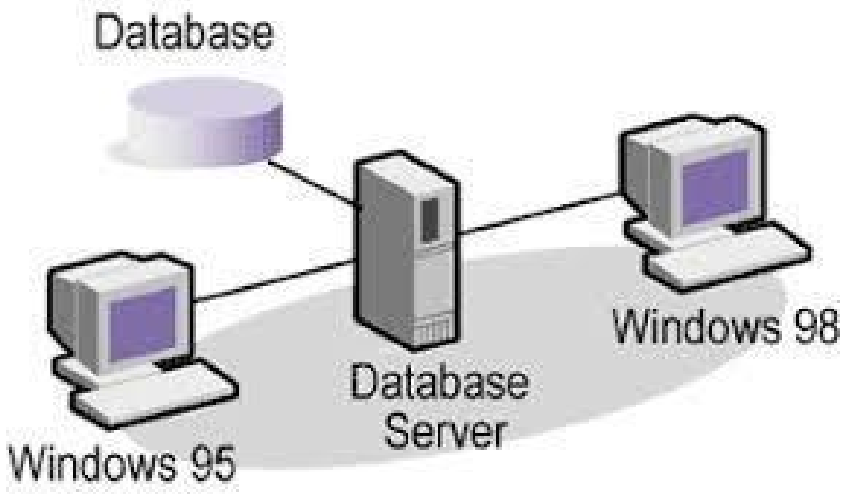
A database server is useful for organizations that have a lot of data to deal with on a regular basis. If you have client-server architecture where the clients need process data too frequently, it is better to work with a database server. Some organizations use the file server to store and process data. A database server is much more efficient than a file server.

In Database Network the client execute SQL requests to the database server. The Network Database Server Process the client database request and the executed answers of SQL command are come back over the network computer. In the whole concept Database server serves its own power to process the request or search the requested result. The Database server some time also known as SQL engine.

All database functions are controlled by the database server. Any type of computer can be used as database server. It may be microcomputer, minicomputer or mainframe computer. In large organization networks, the mainframe computers are used as server. Some people refer to the central DBMS functions as the back-end functions, whereas the application programs on the client computer as front-end programs. You can say that client is the application, which is used to interface with the DBMS, while database server is a DBMS.

The Database server manages the recovery security services of the DBMS. It enforces the constraints that are specified inside the DBMS. It controls and manages all the clients that are connected to it. It handles all database access and control functions. It provides concurrent access control. It provides better security and server hides the DBMS from clients. It provides the multi-

user environment. Several users can access the database simultaneously. All the data is stored on the data server therefore, the DBA can easily create the backup of the database.



Application Server:

An application server is a server program in a computer in a distributed network that provides the business logic for an application program. The application server is frequently viewed as part of a three-tier application, consisting of a graphical user interface (GUI) server, an application (business logic) server, and a database and transaction server. More descriptively, it can be viewed as dividing an application into:

- A first-tier, front-end, Web browser-based graphical user interface, usually at a personal computer or workstation
- A middle-tier business logic application or set of applications, possibly on a local area network or intranet server
- A third-tier, back-end, database and transaction server, sometimes on a mainframe or large server

The examples of application servers:

Jboss : open-source server from Jboss community
Glassfish: provided by Sun Microsystem, now acquired by Oracle
Weblogic : provided by Oracle
Websphere : provided by IBM

Comparison among Various Types Of Servers

Application Server	Web Server
A server that exposes business logic to client applications through various protocols including HTTP	A server that handles HTTP protocol
Application server is used to serve web based applications and enterprise based application (i.e servlets and JSP). Application server may contain a web server internally	Web server is used to serve web based applications (i.e servlets and JSP)
Resource utilization is high	Resource utilization is low

UNIT II- SCRIPTING ESSENTIALS

Need for Scripting languages - Types of scripting languages - Client side scripting - Server side scripting - PHP - Working principle of PHP - PHP Variables - Constants - Operators – Flow Control and Looping - Arrays - Strings - Functions - File Handling - PHP and MySQL - PHP and HTML - Cookies - Simple PHP scripts.

2.1 NEED FOR SCRIPTING LANGUAGES

Definition of Scripting Language: A scripting language is a programming language designed for integrating and communicating with other programming languages. Some of the most widely used scripting languages are HTML, JavaScript, VBScript, PHP, Perl, Python, Ruby, and ASP and so on.

- In general, scripting languages are easier to learn and faster to code in more structured and compiled languages such as C and C++.
- Scripting languages are useful tools for developing interactive web pages with minimum efforts.
- Scripting Languages are often interpreted (rather than compiled).
- The scripting languages are useful for producing dynamic web contents. That means web page can be changed using user input.

Advantages of Scripting Languages:

- Scripting languages are easy to learn.
- It requires minimum programming knowledge or experience to develop the web pages using scripting languages.
- The scripting languages allow simple creation and editing in variety of text editors.
- Using scripting languages we can develop dynamic and interactive web pages.
- There are some scripting languages that validate the information entered by the user.

2.2 TYPES OF SCRIPTING LANGUAGES

There are two types of scripting languages – Client side scripting language and server side scripting language.

Client Side Scripting Language:

The client side scripting is used to create the web pages as a request or response to server. These pages are displayed to the user on web browser. For example: HTML, CSS, JavaScript, PHP.

Server Side Scripting Language:

Server side scripting is used to create web pages that provide some services. These scripts generally run on web servers. For example: ASP, JSP, Servlet, PHP.

Difference between Client side and Server side scripting languages

Server Side Scripting	Client Side Scripting
The server side scripting is used to create the web pages that provide some services.	The client side scripting is used to create the web pages as a request or response to server. These pages are displayed to the user on web browser.
A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser.	The processing of these scripts takes place on the end user's computer. The source code is transferred from the user's computer over the internet and run directly in the browser.
Uses: processing of user request, accessing to databases.	**Uses**: making interactive web pages, for interacting with temporary storages such as cookies or local storage, sending request to server and getting the response and displaying that response in web browser.

These scripts generally run on web servers.	These scripts generally run on web browser.
Example: PHP, ASP.NET, C++, java and C#.	**Example**: HTML, CSS, JavaScript.

2.3 PHP

PHP was developed in 1994 by Apache group. PHP stands for PHP: Hypertext Pre-processor. PHP is a server-side scripting language. It is mainly used for form handling and database access. It is free to download and use.

2.4 WORKING PRINCIPLE OF PHP

PHP is a server side scripting language embedded in XHTML. It is an alternative to CGI, ASP, ASP.NET and JSP. The extension to PHP files are .php, .php3 or .phtml. The PHP processor works in two modes. If the PHP processor finds XHTML tags in the PHP script then the code is simply copied to the output file. But when the PHP processor finds the PHP code in the script then that code is simply interpreted and the output is copied to the output file.

If you click the view source on the web browser you can never see the PHP script because the output of the PHP script is send directly to the browser but you can see the XHTML tags. PHP makes use of dynamic typing that means there is no need to declare variables in PHP.

The type of variable gets set only when it is assigned with some value. PHP has large number of library functions which makes it flexible to develop the code in PHP.

Installation of PHP

For installing PHP either PHP installer is preferred or all in package like XAMPP/WAMPP is preferred. Before installing PHP, install Apache web server on your PC. The PHP installer can be downloadable from www.php.net/download.

***Exercise:* Explain how can you create a web based application using XAMPP. Give all the steps required in detail.**

Solution:

XAMPP is a free distribution package that makes it easy to install Apache web server, Mysql, PHP, PERL. Here in XAMPP(The X stands for any OS) or WAMPP(the W stands for Windows OS).

Step 1:Go to the site : https://www.apachefriends.og/index.html
Step 2: Click on download XAMPP for windows or Linux depending upon your operating system.
Step 3: When prompted for the download, click "Save" and wait for your download to finish.
Step 4: Install the program, and click on "RUN". Accept default settings by clicking next button. Finally you will get installation completion message.
Step 5: On your drive, the XAMPP folder will be created. Click on xampp_start file, this will enable to start Apache, Mysql and Tomact.
Step 6: Write a PHP script and save it in C:\XAMPP\htdocs\php-examples folder by giving the filename and extension as .php.
Step 7: Open the web browser and type http://localhost/php-examples/yourfilename.php
Step 8: The web application will be executed within your web browser.

For example:

```
<? php
$s="I like PHP";
echo $s;
?>
```

2.5 GENERAL SYNTACTIC CHARACTERISTICS OF PHP

- PHP code can be embedded in the XHMTL document. The code must be enclosed within <? php and ?>
- If the PHP script is stored in some another file and if it needs to be referred then include construct is used. The variable names in PHP begin, with the $ sign. Following are some reserved keywords that are used in PHP.

And	Default	False	If	Or	This
Break	Do	For	Include	Require	True
Case	Else	Foreach	List	Return	Var
Class	Elseif	Function	New	Static	Virtual
Continue	Extends	Global	Not	Switch	While
					Xor

- The comments in PHP can be #, //, /, /*.... */ .
- The PHP statements are terminated by semicolon.

How to write and execute PHP document?

Open some suitable text editor like Notepad and type the following code. Save the code by the extension **.php.** It is expected that the PHP code must be stored in **htdocs** folder of Apache.

As I have installed **xampp**, I have got the directory c:\xampp\htdocs. I have created a folder named php-examples inside the htdocs and stored all my PHP documents in that folder.
Hence, when i want to get the output of the PHP code I always give the URL.

Example:

```
<html>
<head>
<title>PHP DEMO</title>
</head>
<body>
<?php
$i=10;
echo “<h3>WELCOME </h3>”;
echo “ The value of the variable is =$i”;
?>
</body>
</html>
```

2.6 PHP Variable

- Variables are the entities that are used for storing the values. PHP is a dynamically typed language. That is PHP has no type declaration. The value can be assigned to the variable in the following manner:
  $variable_name=value;
- If the value is not assigned to the variable then by default the value is NULL. The unsigned variables are called unbound variables. If unbound variable is used in the expression then its NULL value is converted to the value 0.

Following are some rules that must be followed while using the variables:

- The variable must start with letter or underscore but it should not begin with a number.
- It consists of alphanumeric characters or underscore.
- There should not be space in the name of the variable

- While assigning the values to the variables the variable must start with the $. For example, $marks=100;
- Using the function IsSet the value of the variable can be tested. That means if isSet($marks) function returns TRUE then that means some value is assigned to the variable marks.
- If the unbound variable gets referenced then the error reporting can be done with the help of function error_reporting(7). The default error reporting level is 7.

Variable Scope

Local variables

Local variables are variables that are created within, and can only be accessed by, a function. They are generally temporary variables that are used to store partially processed results prior to the function's return.

One set of local variables is the list of arguments to a function.

Example:

```
<?php
$i=10;
Function add()
{
$i=20;
$j=$i+10;
Echo "Value of j is:".$j;
}
Add();
?>
```

Output:

//it takes the value $i=20 (Locally Scoped)

Value of j is :30

Global variables

There are cases when you need a variable to have *global* scope, because you want all your code to be able to access it. To declare a variable as having global scope, use the keyword global.

Syntax:

```
Global var_name;    //This will access the clobal values of the variable.
```

Example:

```
<?php
$i=10;
Function add()
{
$i=20;
global $i;
$j=$i+10;
Echo "Value of j is:".$j;
}
Add();
?>
```

Output:

//$i is declared as a global variable so it takes the value $i=10

Value of j is :20

Static variables

Static variables can be initialized only once. The static variable will be initialized for the first time. Static variable will not be initialized whenever it is declared.

```
<?php
function test()
{
static $count = 0;
echo $count;
$count++;
}
for($i=0;$i<5;$i++)
test();//function is called 5 times
?>
```

2.7 Data Types

There are four scalar types that are used in PHP and those are Integer, Boolean, Double and String.

Integer Type

- For displaying the integer value the Integer type is used.
- It is similar to the long data type in C.
- The size is 32 bit.

Double Type

- For displaying the real values the double data is used
- It includes the numbers with decimal point, exponentiation or both. The exponent can be represented by E or e followed by integer literal.
- It is not compulsory to have digits before and after the decimal point. For instance .123 or 123. is allowed in PHP.

String Type

- There is no character data type in PHP. If the character has to be represented then it is represented using the string type itself; but in this case the string is considered to be of length 1.
- The string literal can be defined using either single or double quotes.
- In single quotes the escape sequence or the values of the literals can not be recognized by PHP but in double quotes the escape sequence can be recognized. For example : “The total marks are=$marks” will be typed as it is but “The total marks are=$marks” will display the value of $mark variable.

Boolean Type

- There are only two types of values that can be defined by the Boolean type and those are TRUE and FALSE.
- If Boolean values are used in context of integer type variable then TRUE will be interpreted as 1 and FALSE will be interpreted as 0.
- If Boolean values are used in context of double type then the FALSE will be interpreted as 0.0.

Constants

- Constant is an identifier that contains some value. Once the constant value is assigned to this identifier it does not get changed. Constant is case sensitive by default.
- Generally the constant identifier is specified in upper case. The valid constant name must start with letters or underscore. It may then be followed by digits.
- Using define function we can assign value to the constant. The first parameter is define function is the name of the constant and the second parameter is the value which is to be assigned.

Example:

```
<?
//valid constants names
define("MYVALUE","10");
echo "MYVALUE";
//invalid constant names
define("1MYVALUE","SOMETHING")
echo "1MYVALUE";
?>
```

2.8 OPERATORS

Arithmetic Operators and Operations

- PHP supports the collection of arithmetic operators such as +, _, *, /,%,++ and – with their usual meaning.
- While using the arithmetic operators if both the operands if both the operands are integer then the result will be integer itself.
- If either of the two operands is double then the result will be double.

Operator	Description	Example	Result
+	Addition	$a = 2 + 5;	$a=7
-	Subtraction	$a = 10 - 2;	$a=8
*	Multiplication	$a = 2 * 5;	$a=10
/	Division	$a = 15 / 5;	$a=3
%	Modulus	$a = 23 % 7;	$a=3.28
++	Increment	$a =5; $a ++;	$a=6
--	Decrement	$a =5; $a --;	$a=4

- PHP has large number of predefined functions. Some of these functions are enlisted in the following table:

Function	Purpose
Floor	The largest integer less than or equal to the parameter is returned.
Ceil	The smallest integer less than or equal to the parameter is returned.
Round	Nearest integer is returned
Abs	Returns the absolute value of the parameter
Min	It returns the smallest element.
Max	It returns the largest element.

Arithmetic Operators Example

```
<html>
<body>
<?php
$i=(20 + 10);
$j=($i - 5);
$k=($j * 4);
$l=($k / 2);
$m=($l % 5);
echo "i = ".$i."<br/>"; echo "j = ".$j."<br/>"; echo "k = ".$k."<br/>"; echo "l = ".$l."<br/>"; echo
"m = ".$m."<br/>";
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

i = 30
j = 25
k = 100
l = 50
m = 0

Increment and Decrement Operators:
It is also called as the unary operator. It usually increments or decrements the value by one.

Operator	Name	Description
++$a	Pre-increment	Increments $a by one, then returns $a
$a++	Post-increment	Returns $a, then increments $a by one
--$a	Pre-decrement	Decrements $a by one, then returns $a
$a--	Post-decrement	Returns $a, then decrements $a by one

Increment and Decrement Operators Example

```
<html>
<body>
<?php
$i=10;
$j=20;
$i++;
$j++;
echo $i."<br/>";
echo $j."<br/>";
$k=$i++;
```

```
$l=++$j;
echo $k."<br/>"; echo $l;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

11
21
11
22

Assignment Operators in PHP

Assignment operator is used to assign a value to a variable

Operator	Example	Is the same as
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Logical operators

Logical operators produce true-or-false results, and therefore are also known as Boolean operators. There are four of them.

Operator	Description	Example
&&	And	$j == 3 && $k == 2
and	Low-precedence and	$j == 3 and $k == 2
||	Or	$j < 5 || $j > 10
or	Low-precedence or	$j < 5 or $j > 10
!	Not	! ($j == $k)
xor	Exclusive or	$j xor $k

```
<?php
$a = 1; $b = 0;
echo ($a AND $b) . "<br>";
echo ($a or $b) . "<br>";
echo ($a XOR $b) . "<br>";
echo !$a . "<br>";
?>
```

Output:
0
1
1
0

Relational Operators or Comparison Operator
Relational operators test two operands and return a Boolean result of either TRUE or FALSE. There are three types of relational operators: *equality*, *comparison*, and *logical*. It is also called as comparison operator.
Comparison operators are generally used inside a construct such as an if statement in which you need to compare two items. For example, you may wish to know whether a variable you have been incrementing has reached a specific value, or whether another variable is less than a set value, and so on
Note the difference between = and ==. The first is an assignment operator, and the second is a comparison operator.

Operator	Description
==	Equality
===	Identity(Checks both value and type)
!=	Not Equal
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Example:
```
<?php
$a = "1000";
$b = "+1000";
if ($a == $b) echo "1";//both will converted to same type and the it will compare.
if ($a === $b) echo "2";// this will not be executed. Because $a and $b are of different type
?>
```
Output:
1
Example2:
```
<?php
$a=22;
$b=25;
$c=30;
if(($a>$b)&($a>$c))
echo "A is greater";
elseif($b>$c)
echo "B is greater";
else
echo "C is greater";
?>
```
Output:
C is greater

String Operators in PHP

Operator	Name	Example	Result
.	Concatenation	$a = "Hello" $b = $a . " world!"	$b = "Hello world!"
.=	Concatenation Assignment	$a = "Hello" $a .= " world!"	$a = "Hello world!"

String Operators Example

```
<html>
<body>
<?php
$a = "Hello";
$b = $a . " Friend!";
echo $b;
echo "<br/>";
$c="Good";
$c .= " Day!";
echo $c;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Hello Friend!
Good Day!

Boolean Operators
Boolean operators AND, OR, and NOT are used to manipulate logical statements. Boolean operators are the core operators used in digital control systems as well as computer systems. AND and OR are binary operators, while NOT is a unary operator.

Operator	Meaning
And &&	The binary AND operation is performed
Or ||	The binary OR operation is performed
Xor	The XOR operation will be performed

2.9 Displaying Messages on the Web page

- The PHP is a server side scripting language and is used to submit the web page to the client browser. Hence it is a standard method of embedding the PHP code within the XHTML document. That means we can use the XHTML tags in PHP while displaying the output.
- The print function is used to create simple unformatted output. For example : The string can be displayed as follows: **print "I am proud of my country**
- The numeric value can also be displayed using the print. For example : **print(100);**

- PHP also makes use of the printf function used in C. For example: **printf("The student %d has %f marks", $roll_no, $marks);**

Example:

```
<html>
<head>
<title>Output Demo</title>
</head>
<body>
<?php
print "<h2>Welcome to my website </h2>";
print "<hr/>";
$roll_no=1;
$name=AAA;
printf( "<b> The roll number: %d </b>", $roll_no);
print "<br/><b>";
printf(" The name :%s",$name);
print "</b>";
?>
</body>
</html>
```

2.10 CONSTANTS

Constants are similar to variables, holding information to be accessed later, except that they are what they sound like—constant. In other words, once you have defined one, its value is set for the remainder of the program and cannot be altered.

Constant in PHP - define() function is used to set a constant

It takes three parameters they are:

- Name of the constant
- Value of the constant
- Third parameter is optional. It specifies whether the constant name should be case-insensitive. Default is false

Constant string Example

```
<html>
<body>
<?php
/* Here constant name is 'Hai' and 'Hello Friend' is its constant value and true indicates the constant value is case-insensitive */
define("Hai","Hello Friend",true);
echo hai;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Hello Friend

PHP Example to calculate the area of the circle

```
<html>
<body>
<?php
// defining constant value PI = 3.14
define("PI","3.14");
$radius=15;
```

```
$area=PI*$radius*$radius;
echo "Area=".$area;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Area=706.5

Predefined Constants

PHP comes ready-made with dozens of predefined constants that you generally will be unlikely to use as a beginner to PHP. However, there are a few—known as the *magic constants*—that you will find useful. The names of the magic constants always have two underscores at the beginning and two at the end.

Magic constant	Description
__LINE__	The current line number of the file.
__FILE__	The full path and filename of the file. If used inside an `include`, the name of the included file is returned. In PHP 4.0.2, __FILE__ always contains an absolute path with symbolic links resolved, whereas in older versions it might contain a relative path under some circumstances.
__DIR__	The directory of the file. If used inside an `include`, the directory of the included file is returned. This is equivalent to *dirname*(__FILE__). This directory name does not have a trailing slash unless it is the root directory. (Added in PHP 5.3.0.)
__FUNCTION__	The function name. (Added in PHP 4.3.0.) As of PHP 5, returns the function name as it was declared (case-sensitive). In PHP 4, its value is always lowercase.
__CLASS__	The class name. (Added in PHP 4.3.0.) As of PHP 5, returns the class name as it was declared (case-sensitive). In PHP 4, its value is always lowercase.
__METHOD__	The class method name. (Added in PHP 5.0.0.) The method name is returned as it was declared (case-sensitive).
__NAMESPACE__	The name of the current namespace (case-sensitive). This constant is defined at compile time. (Added in PHP 5.3.0.)

Example:

```
<?php
echo "This is line " . __LINE___. " of file " . __FILE__;
?>
```

Output:

This is line 2 of file sample.php

This causes the current program line in the current file (including the path) being executedto be output to the web browser.

2.11 FLOW CONTROL AND LOOP

2.11.1 The if Statement in PHP

- The **if statement**, the **if...else** statement or **if....elseif** statements are used as selection statements. The selection is based on some condition.
- If statement executes some code only if a specified condition is true

Syntax:

```
if (condition) {
code to be executed if condition is true;
}
```

The if Statement Example

```
<html>
<body>
<?php
$i=0;
/* If condition is true, statement is executed*/
if($i==0)
echo "i is 0";
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:

i is 0

The if…else Statement in PHP
If…else statement executes some code if a condition is true and some another code if the condition is false

Syntax:

```
if (condition)
{
code to be executed if condition is true;
}
else
{
code to be executed if condition is false;
}
```

The if…else Statement Example

```
<html>
<body>
<?php
$i=1;
/* If condition is true, statement1 is executed, else statement2 is executed*/
if($i==0)
echo "i is 0";   //statement1
else
echo "i is not 0";         //statement2
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:
i is not 0

The if…elseif…else Statement in PHP
If…elseif…else statement selects one of several blocks of code to be executed

Syntax:

```
if (condition)
 {
code to be executed if condition is true;
}
elseif (condition)
 {
code to be executed if condition is true;
}
else
{
code to be executed if condition is false;
}
```

The if…elseif…else Statement Example (Comparing two numbers)

```
<html>
<body>
<?php
$i=22;
$j=22;
/* If condition1 is true, statement1 is executed, if condition1 is false and condition2 is true, statement2 is executed, if both the conditions are false statement3 is executed */
if($i>$j)
echo "i is greater"; //statement1 elseif($i<$j)
echo "j is greater";      //statement2
else
echo "numbers are equal";      //Statement3
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:

numbers are equal

2.11.2 Switch Statements

Similar to if statement the switch statement can also be used for selection. Following is a simple PHP script for demonstrating switch statements

Syntax:

```
switch (n)
{ case label1:
code to be executed if n=label1;
break;
case label2:
code to be executed if n=label2;
break;
...
default:
code to be executed if n is different from all labels;
}
```

PHP Program

```
<?php
$today=getdate();
switch($today[‘weekday’])
{
case “Monday”: print “Today is Monday”;
                break;
case “Tuesday”: print “Today is Tuesday”;
                break;
case “Wednesday”: print “Today is Wednesday”;
                break;
case “Thursday”: print “Today is Thursday”;
break;
case “Friday” : print “Today is Friday”;
break;
case “Saturday” : print “Today is Saturday”;
break;
case “Sunday”: print “Today is Sunday”;
break;
default: print “Invalid Input”;
}
?>
```

2.11.3 Loop Statements

- The while, for and do-while statements of PHP are similar to Javascript.
- Following is a simple PHP script which displays the first 10 number.

For loop in PHP

PHP for loop executes a block of code, a specified number of times

Syntax:

```
for (initialization; test condition; increment/decrement)
 {
code to be executed;
}
```

For loop Example

```
<html>
<body>
<?php
echo "Numbers from 1 to 20 are: <br>";
for ($x=1; $x<=20; $x++) {
echo "$x ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Numbers from 1 to 20 are:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Example 2:

```
<?php
$sum=0;
$n=10;
for($i=1;$i<=$n;$i++)
{
$sum=$sum+$i;
}
echo "The sum of  ".$n." Natural numbers is: ". $sum;
?>
```

Output:

The sum of 10 Natural numbers is: 55

While Loop in PHP

While loop, loops through a block of code as long as the specified condition is true

Syntax:

```
while (condition)
{
code to be executed;
}
```

While Loop Example

```
<html>
<body>
<?php
$i=1;
/* here <condition> is a Boolean expression. Loop body is executed as long as condition is true*/
while($i<5){
echo "i is = $i <br>";
$i++;
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

Do While loop in PHP

Do while loop will always execute the block of code once, it will then check the condition, and if the condition is true then it repeats the loop

Syntax:

```
do {
code to be executed;
```

} while (condition);

Do While loop Example

```
<html>
<body>
<?php
$i=1;
/* here <condition> is a Boolean expression. Please note that the condition is evaluated after executing the loop body. So loop will be executed at least once even if the condition is false*/
do
{
echo "i is = $i <br>";
$i++;
}while($i<5);
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

Break statement

Break statement is used to terminate the loop. After the break statement is executed the control goes to the statement immediately after the loop containing break statement

Break statement example

```
<html>
<body>
<?php
/* when $i value becomes 3, the loop is Terminated*/
for($i=0;$i<5;$i++)
{
if($i==3)
break;
else
echo "$i ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
0 1 2
```

Continue statement

There are cases in which, rather than terminating a loop, you simply want to skip over the remainder of iteration and immediately skip over to the next. Continue statement is used to skip a particular iteration of the loop.

Continue statement example

```
<html>
<body>
<?php
/* when $i value becomes 3, it will skip the particular of the loop*/
for($i=0;$i<=5;$i++)
{
if($i==3)
continue;
else
echo "$i ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

0 1 2 4 5

2.11.4 Examples based on Control Statement

Write a PHP script to display the squares and cubes of 1 to 10 numbers.

Sol :

```
<html>
<head>
<title> Square and cube Table </title>
</head>
<body>
<center>
<?php
print “<table border=1”>”;
print “<tr>”;
print “<th> Numbers</th>”;
print “<th> Squares</th>”;
print “<th> Cube</th>”;
print “</tr>”;
for($i=1;$i<=10;$i++)
{
print “<tr>”;
print “<td>$i”;
print “</td>”;
print “<td>”;
print $i*$i;
print “</td>”;
print “<td>”;
print pow($i,3);
print “</td>”;
print “</tr>”;
}
print “</table>”;
?>
</center></body></html>
```

Example Programs:
Write a PHP Script to compute the sum and average of N numbers.

PHP Program

```
<html>
<head>
<title> Sum and Average </title>
</head>
<body>
<center>
<?php
$sum=0;
for($i=1;$i<=10;$i++)
{
        $sum += $i;
}
$avg=$sum/10;
print “The sum is : $sum”;
print “<br/>”;
print “The average is : $avg”;
?>
</center>
</body>
</html>
```

Write PHP programs to print whether current year is leap year or not.
Sol :

```
<html>
<head>
<title> Leap year demo</title>
</head>
<body>
<?php
$year=2016;
print “<br/>”;
if($year%4==1)
{
printf(“Year %d is not a leap year”,$year);
}
else
{
printf(“Year %d is a leap year”,$year);
}
?>
</body>
</html>
```

Write PHP programs to print whether given number is odd or even.

Sol :

```
<html>
<head>
<title> Even Odd Demo</title>
```

```
</head>
<body>
<?php
for($i=1;$i<=10;$i++)
{
$num=$i;
print “<br/>”;
if($num%2==1)
{
print(“Number %d is Odd”,$num);
}
else
{
print(“Number %d is Even”,$num);
}
}
?>
</body>
</html>
```

Write PHP Script to compute the sum of positive integer upto 30 using do-while statement.
Sol:

```
<html>
<head>
<title> Sum of Integer</title>
</head>
<body>
<?php
$sum=0;
$i=1;
do
{
$sum=$sum+$i;
$i++;
}while($i<=30);
printf(“The sum of first 30 positive integers is %d”,$sum);
?>
</body>
</html>
```

Write PHP Script to compute factorial of ‘n’ using while or for loop construct.
Sol:

```
<html>
<head>
<title> Factorial Program </title>
</head>
<body>
<?php
$n=5;
$factorial=1;
for($i=$n;$i>=1;$i--)
{
$factorial=$factorial*$i;
```

```
}
echo “Factorial of $n is $factorial”;
?>
</body>
</html>
```

Write PHP Script to display Fibonacci of length 10.
Sol:

```
<html>
<head>
<title>Fibonacci Series</title>
</head>
<body>
<?php
$i=1;
$j=1;
print “<b> Fibonacci Series <br/> </b> “;
printf(“%d,%d”,$i,$j);
for($count=1;$count<9;$count++)
{
        $k=$i+$j;
        $i=$j;
        $j=$k;
        printf(“%d”,$k);
}
?>
</body>
</html>
```

Construct a PHP Script to compute the squareroot, square, cube and Quad of 10 numbers.
Sol :

```
<html>
<head>
<title> SQUARE CUBE QUAD DEMO </title>
</head>
<body>
<?php
print”<b>Table</b><br/>”;
print”<table border=’1’>”;
print “<tr><th>num</th><th>Sqr</th><th>Cube</th><th>Quad</th></tr>”;
for($count=1;$count<=10;$count++)
{
$sq=$count * $count;
$cube=$count*$count*$count;
$quad=$count*$count*$count*$count;
print “<tr><th>$count</th><th>$sq</th><th>$cube</th><th>$quad</th></tr>”;
}
print “</table>”;
?>
</body>
</html>
```

With the use of PHP, switch case and if structure perform the following and print appropriate message. (i) Get today's date (ii) If date is 3, it is dentist appointment (iii)If date is 10, go to conference (iv) If date is other than 3 and 10, no events are scheduled.

Sol:

```
<html>
<body>
<?php
echo "Today date is". date("d/m/y")."<br/>";
if(date("d")<3) || (date("d")>10))
        echo "No Event!!";
else if((date("d")>3 && date("d")<10))
        echo "No Event!!";
else
{
switch(date("d"))
{
        case 3 :echo "Dentist Appointment";
        break;
        case 10: echo "Go to Conference";
        break;
}
}
?>
</body>
</html>
```

Write a PHP code to display the following pattern

```
1
01
101
0101
10101
```

Sol:

```
<html>
<head>
</head>
<body>
<?php
for($i=0;$i<7;$i++)
{
        for($j=1;$j<$i;$j++)
        {
                if($i+$j)%2==0)
                {
                        printf("0");
                }
                else
                {
                        printf("1");
                }}
print "<br/>";
```

```
}
?>
</body>
</html>
```

2.12 Arrays

- Arrays is a collection of similar type of elements but in PHP you can have the elements of mixed type together in single array.
- In each PHP, each element has two parts **Key** and **Value**.
- The key represents the index at which the value of the element can be stored.
- The **keys** are positive integers that are in ascending order.

2.12.1 Array Creation

In PHP there are two types of arrays -

1. Indexed Array: Indexed array are the arrays with numeric index. The array values can be stored from index 0. For example -

```
<html>
<head>
<title> PHP Indexed arrays</title>
</head>
<body>
<?php
$names=array("AAA","BBB","CCC");
print_r($names);//print array structure
?>
</body>
</html>
```

Here values gets stored at corresponding index as follows -

```
$mylist[0]=10;
$mylist[1]=20;
$mylist[2]=30;
$mylist[3]=40;
$mylist[4]=50;
```

We can directly assign some value at specific index.

```
$mylist[5]=100;
```

2. Associated Array : Associated arrays are the arrays with named keys. It is a kind of array with **name** and **value** pair. For example -

```
<html>
<head>
<title> PHP Associated Array</title>
</head>
<body>
<?php
$city["AAA"]="Pune";
$city["BBB"]="Mumbai";
$city["CCC"]="Chennai";

//printing array structures
print_r($city);
```

```
?>
</body>
</html>
```

2.12.2 Accessing Array Elements

- Using an array subscript we can access the array element. The value of subscript is enclosed within the square brackets. For example -
  - $citycode[‘Pune’]=005;
  - $Name[0]=”Chitra”;
- Multiple values can be set to a single scalar variable using array. For example -

  $people = array(“Meena”,”Teena”,”Heena”);
  list($operator,$accountant,$manager)=$people;
- By this assignment Meena become operator, Teena becomes accountant and Heena becomes manager.

Consider an associative array called person_age with name and age of 10 person. Write a PHP program to calculate the average age of this associative array.

Sol:

```
<html>
<head>
<title> PHP Associative array</title>
</head>
<body>
<?php
$person_age=array(
                ‘AAA’ => 10, ‘BBB’ => 22, ‘CCC’ => 45, ‘DDD’ => 31, ‘EEE’ => 33, ‘FFF’ =>
25,             ‘GGG’ => 56 , ‘HHH’ => 73, ‘III’ => 35, ‘JJJ’ => 47);
$sum=array_sum($person_age);
print(“The sum of all the ages is $sum”);
$avg=$sum/10;
print(“<br/> The average is $avg”);
?>
</body>
</html>
```

Multidimensional array in PHP

Multidimensional array is an array containing one or more arrays

Multidimensional array Example

```
<html>
<body>
<?php
/* Here $flower_shop is an array, where rose, daisy and orchid are the ID key which indicates rows
and points to array which have column values. */

$flower_shop = array(

"rose" => array( "5.00", "7 items", "red" ), "daisy" => array( "4.00", "3 items", "blue" ), "orchid" =>
array( "2.00", "1 item", "white" ), );
/* in the array $flower_shop['rose'][0], ‘rose’ indicates row and ‘0’ indicates column */
echo "rose costs ".$flower_shop['rose'][0]."items: ".$flower_shop['rose'][1]."<br>";
echo "rose costs ".$flower_shop['daisy'][0]."items: ".$flower_shop[' daisy '][1]."<br>";
```

```
echo "rose costs ".$flower_shop['orchid'][0]."items: ".$flower_shop[‘orchid '][1]."<br>";
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

rose costs 5.00, and you get 7 items.
daisy costs 4.00, and you get 3 items.
orchid costs 2.00, and you get 1 item.

Functions Dealing with Arrays

1. is_array
check whether a variable is an array

```
if(is_array($array)
{
Return true;
}
Else
Return false;
```

2. Count
count all the elements in the top level of an array.

```
echo count($fred);
```

3. sort
Sorting is so common that PHP provides a built-in function.

```
sort($fred);
```

4. shuffle
elements of an array to be put in random order

```
shuffle($cards);
```

5. explode
Sveral items separated by a single character (or string of characters) and then place each of these items into an array.

```
<?php
$temp = explode('***', "A***sentence***with***asterisks");
print_r($temp);
?>
```

Output:

```
Array
(
[0] => A
[1] => sentence
[2] => with
[3] => asterisks
)
```

6. Reset
It reset the array pointer to the first elemet of the array.

```
reset($fred); // Throw away return value
```

$item = reset($fred); // Keep first element of the array in $item

7. End

It moves the pointer to the end of the array.

```
end($fred);
$item = end($fred);
```

8. Unset

The unset function is used to remove particular element from the array. For example consider following PHP document

Example:

```
<?php
$mylist=array(10,20,30,40,50);
unset($mylist[1]);
for($i=0;$i<=4;$i++)
{
        print $mylist[$i];
        print  "";
}
?>
```

2.12.4 Sequential Access to Array Elements

- The array element references start at the first element and every array maintains an internal pointer using which the next element can be easily accessible. This helps to access the array elements in sequential manner.
- The pointer **current** is used to point to the current element in the array. Using the **next** function the next subsequent element can be accessed. Following PHP code illustrates this idea .

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderful!");
$myval=current($mylist);
print("The current value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
?>
```

1. Each

Using **each** function we can iterate through the array elements.

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderfull!");
while($myval=each($mylist)
{
$val=$myval["value"];
print("The current value of the array is <b>$val</b>");
print "<br/>";
}
?>
```

2. foreach

The **foreach** function is used to iterate through all the elements of the loop. The syntax of foreach statement is as follows -

```
foreach($array as $value)
{
    statements to be executed
}
```

The above code can be modified and written as follows -

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderfull!");
foreach($mylist as $value)
{
    print ("The current value of the array is <b> $value </b>");
    print "<br/>";
}
?>
```

2.12.5 Sorting Arrays

- Sorting is the process in which the element of arrays in some specific order. There are two types of ordering which are followed in sorting – ascending order and descending order. Basically PHP uses **sort** function for sorting the array elements. There are some other functions that are also available for sorting the arrays in desired manner.
- The **sort** function sorts the array based on the values. After applying the sort function this function assigns new keys to the values of the array.Following PHP document illustrates these functions -

PHP Program

```
<?php
$A=array("A" => "Supriya", "B" => "Monika", "C" => "Kumar", "D" => "Archana");
print "<b> Original Array:</b><br/>";
foreach($A as $key => $value)
print ("[$key] => $value <br/>");
sort($A);
print "<br/>";
print "<b> Sorted Array:<b><br/>":
foreach($A as $key => $value)
print ("[$key] => $value <br/>");
?>
```

- The **asort()** function sorts an array by the values. But the original keys are kept.
- The **ksort()** function sorts the array by keys but each value's original key is retained.

Example Programs:

1. Use an array to store student information such as enrolment no, name, semester and percentage. Output the data to a web page using PHP.

Sol:

```
<html>
<head></head>
<body>
<?php
$a=array(array(10,"AAA","II",60),array(20,"BBB","III",80),array(30,"CCC","IV",40));
echo "<table border='1'>";
echo "<tr>";
echo "<td>ENo</td><td>Name</td><td>Sem</td><td>Marks</td>";
echo "<tr/>";
for($i=0;$i<3;$i++)
{
echo "<tr>";
for($j=0;$j<4;$j++)
{
        echo"<td>";
        echo $a[$i]{$j];
        echo"</td>";
}
echo "</tr>";
}
echo "</table>";
?>
</body>
</html>
```

2.13 Strings

PHP String Manipulation

PHP provides a rich set of functions to manipulate strings. In this topic, we will discuss some common functions used by PHP developers to remove spaces from a string, count the number of characters of a string, convert a string to contain upper case or lower case letters, split a string or join strings, get substrings from a string, compare strings, search for a substring in a string, and replace and old substring with a new substring of a string, etc.

trim, ltrim, and rtrim functions

trim, ltrim, and rtrim functions are used to remove space from a string.
-trim(String) removes leading and trailing space from the string.
-ltrim(String) removes leading spaces.
-rtrim(String) removes trailing spaces.

strlen() function

The strlen(String) function is used to count the number of characters of a string.

strtolower, strtoupper, ucfirst, ucwords function

The strtolower, strtoupper, ucfirst, and ucwords functions are used to change change cases of a string:

- strtolower(String) changes a string to lowercase.
- strtoupper(String) changes a string to uppercase.
- ucfirst(String) capitalizes the first character of a string.
- ucwords(String) capitalizes the first character of each word in a string.

strcmp() and strcasecmp() functions
The strcmp(String1,String2) compares String1 with String2. It returns less than zero if String1 is less than String2. If String1 is greater than String2 it return greater than zero. If both strings are equal, it returns 0.This function compares two strings in case-sensitive manner. If you want to compare two strings without case-sensitivity, you can use strcasecmp() instead.

split() and join() functions
The split(Separator_char, String) function is used to split a string in to an array of strings by a separating character.

substr() function
The substr() method has two main forms:
- substr(String, Start) returns a substring from the Start position to the end of the string.
- substr(String,Start,Length) returns a substring from the Start position in which the length of the substring is equal to Length.

strpos() and str_replace() functions
The strpos(String, String_to_find) returns the position of the String_to_find in the String. The str_replace(Old_string,New_string,String) is used to replace the Old_string with the New_string.

Example:
```
<?php
echo strlen("Hello world!")."<br>";
echo str_word_count("Hello world!")."<br>";
echo strrev("Hello world!")."<br>";
echo strpos("Hello world!", "world")."<br>";
echo str_replace("world", "Dolly", "Hello world!")."<br>";
echo substr("Hello World",2)."<br>";
$arr=split("l","Hello")
Echo $arr[0]."<br>";
echo strtoupper("Hello")."<br>";
echo strtolower("HELLO")."<br>";
echo trim(" Hello")."<br>";
?>
```

Output:
```
12
2
!dlrow olleH
6
Hello Dolly!
llo World
He
HELLO
hello
Hello
```
Note that the variable **s** is assigned with the string. The string is given in double quotes. Then using the echo whatever string is stored in the variable **s** is displayed on the console.

2.14 Functions

The functions in PHP are very much similar to the functions in C. Let us discuss it in details -

2.14.1 General Characteristics of Functions

- The syntax of the function definition is as follows -

```
function name_of_function(parameter list)
{
  statements to be executed in function-name
  .....
  .....
}
```

- The function gets executed only after the call to that function. The call to the function can be from anywhere in the PHP code. For example -

PHP Program

```
<?php
function myfun()
{
print "<i> This statement is in myfun()</i>";
}
print "<b> The Function Demo Program</b>";
print "<br/>";
myfun();
?>
```

- The **return** statement is used for returning some value from the function body. Following PHP script shows this idea.

PHP Program

```
<?php
function Addition()
{
$a=10;
$b=20;
$c=$a+$b;
return $c;
}
print "<b> The Function Demo Program with return statement</b>";
print "<br/>";
print "10+20 =".Addition();
?>
```

2.14.2 Parameters

- The parameter that we pass to the function during the call is called the **actual parameter**. These parameter are generally the expressions.
- The parameters that we pass to the function while defining it is called the **formal parameter**. These are generally the variables. It is not necessary that the number of actual parameters should match with the number of formal parameters.
- If there are few actual parameter and more formal parameters then the value of formal parameter is will be some unbounded one. If there are many actual parameters and few formal parameters then the excess of actual parameters will be ignored.

- The default parameter passing technique in PHP is **pass by value.** The parameter passing by value means the values of actual parameters will be copied in the formal parameters. But the values of formal parameters will not be copied to the actual parameters.
- Following PHP script illustrates the functions with parameters

PHP Program

```
<?php
function Addition($a,$b)
{
 $c=$a+$b;
 return $c;
}
print "<b> The Function Demo Program with parameter passing and return statement</b>";
print "<br/>";
$x=10;
$y=20;
print "10+20 =".Addition($x,$y);
?>
```

There are two ways to pass parameters by reference.

1. Add & at the beginning of the name of the actual parameter. For example -

```
<?php
function add_some_extra($string)
{
$string='This is a string";
$str1=&$str;//adding & at the beginning of the name of actual paramater.
print "Before function call: $str<br/>";
add_some_extra($str1);
print "After function call: $str<br/>";
?>
```

2. Add & to actual parameter in the function call. For example -

```
<?php
function add_some_extra(&$string)
{
        $string = "This string is replaced";
}
$str="This is a string";
print "Before function call :$str<br/>";
add_some_extra($str);
print "After function call:$str<br/>";
?>
```

2.15 SCOPE OF THE VARIABLES:

In PHP the scope of the variable is local. That means we can define a variable within a function and by the same name another variable can be defined outside the function. These two variables are considered to be different entities.

Example:

```
<?php
function myfun()
{
        $a=10;
```

```
        print  "The value of a = $a";
}
myfun();
print "<br>";
$a=20;
print "the value of a=$a";
?>
```

Output:

The value of a =10

The value of a =20

We can define the global variable also. Following is the PHP script

```
<?php
$a=10;
function myfun()
{
        global $a;
        $a+=10;
print "The value of a=$a";
}
myfun();
print "<br>";
$a=$a-5;
print  "the value of a =$a";
?>
```

Output

The value of a = 20

The value of a =15

LIFETIME OF THE VARIABLES:

The lifetime of a variable can be defined as the time from which it is used first to the end of function execution. In PHP the static variable is used to remember the previous values.

The lifetime of static variable is the time when the variable if first used and it ends when the script terminates its execution.

For the declaration of the static variable the keyword static variable, the keyword static is used.

```
function myfun()
{
static $count = 0;
count ++;
print "The number of times you have visited this page is $count <br/>";
}
```

2.16 FUNCTION

Functions are group of statements that can perform a task

Defining a Function

The general syntax for a function is:

function *function_name*([*parameter* [, ...]])

```
{
// Statements
}
```

• A definition starts with the word function.

• A name follows, which must start with a letter or underscore, followed by any number of letters, numbers, or underscores.
• The parentheses are required.
• One or more parameters, separated by commas, are optional.

PHP Functions - Return values
Functions can also return the valuese to the point where they have called.
Return statement is used to return the value.
Syntax:

```
function func_name()
{
….
return $variable;
}
echo func_name();
```

Example:

```
 <?php
function add($x,$y)
{
 $total=$x+$y; return $total;
}
echo "1 + 16 = " . add(1,16);
?>
```

Call by Value:
The changes made in the formal arguments will not be reflected back to the actual arguments.
Example:
Swap Numbers PHP Example (Call by value)

```
<?php
$num1=10;
$num2=20;
 echo "Numbers before swapping:<br/>"; echo "Num1=".$num1;
 echo "<br/>Num2=".$num2;
swap($num1,$num2); //call by value
 function swap($n1,$n2)
{
 $temp=$n1;
$n1=$n2;
$n2=$temp;
 echo "<br/><br/>Numbers after swapping:<br/>";
echo "Num1=".$n1;
echo "<br/>Num2=".$n2;
}
?>
```

Call by Reference:
The changes made in the formal arguments will be reflected back to the actual arguments.

Swap Numbers PHP Example (Call by Reference)

```
<?php
$num1=10;
$num2=20;
```

```
 echo "Numbers before swapping:<br/>"; echo "Num1=".$num1;
 echo "<br/>Num2=".$num2;
swap($num1,$num2);
 function swap(&$n1,&$n2) //Call by reference
{
 $temp=$n1;
$n1=$n2;
$n2=$temp;
 echo "<br/><br/>Numbers after swapping:<br/>";
echo "Num1=".$n1;
echo "<br/>Num2=".$n2;
}
?>
```

Recursive Function:

A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result and the end of each iteration.

Example:

```
 <?php
function factorial($number)
 {
if ($number == 0)
 return 1;
return $number * factorial($number - 1);
}
echo factorial(5);
 ?>
```

Output:

120

2.17 FILE HANDLING:

PHP is known as a server side scripting language. Hence file handling functions suc as create, read, write, append are some file related operations that are supported by PHP.

Opening and closing files:

The first step in file handling is opening of the file.

It takes two parameters- The first parameter of this function contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened.

Modes	Description
r	Read only. Starts reading from the beginning of the file.
r+	Read/Write. Starts reading from the beginning of the file
w	Write only. Opens and clears the contents of the file; or creates a new file if it is not created.
w+	Read/ Write. Opens and clears the contents of the file; or creates a new file if it is not created.
a	Append. Opens and writes to the end of the file or creates a new file of it is not created.
a+	Read/Append. Preserves the file content by writing to the end of the file.

For example:
$my_file=’file.txt’;
$file_handle=fopen($my_file,’a’) or die(‘Cannot open file: ’.$my_file);
The fopen function returns TRUE if the required file is opened.

Reading from file:
The fread is the function which is used to read the file. It takes two parameters. The first parameter is the handle to the file and the second parameter is the number of bytes to be read. The filesize is the function which takes the filename as the parameter.

For example: $mystring = fread($file_handle, filesize(“file.txt”));

There is another function named file_get_contents using which the contents of the file can be obtained.
The fgets() function is used to read a single line from the file. For example, following code displays the contents of the file line by line.

```
while(!feof($file_handle))
{
        echo fgets($file_handle).”<br/>”;
}
```

Example Reading a file with fgets

```
<?php
$fh = fopen("testfile.txt", 'r') or
die("File does not exist or you lack permission to open it");
$line = fgets($fh);
fclose($fh);
echo $line;
?>
```

Output:
Line

Example - Reading a file with fread

```
<?php
$fh = fopen("testfile.txt", 'r') or
die("File does not exist or you lack permission to open it");
$text = fread($fh, 3);
fclose($fh);
echo $text;
?>
```

Output:
lin

Exercise: Write a PHP program to read a text file line by line and to display it on screen.

Solution:
Step-1: Create a input file Myfile.txt as follows.
Hello
everybody
how are you?

Step-2: Create a PHP script for reading the input file line by line as follows.

```
<?php
$file = fopen("Myfile.txt","r")
while(!feof($file))
{
echo fgets($file)."<br/>";
}
fclose($file);
?>
```

Writing to a file:

The fwrite is the function which is used to write the contents to the file. It takes two parameters – The first parameter is the handle to the file and the second parameter is the number of bytes to be written. For example -

```
$written_string = fread($file_handle, $my_data);
```

Locking Files:

Multiple PHP scripts can access the same file at a time. But this causes conflict problems. That means there can be the situation in which one script is reading the file and at the same time the other script is writing to that file. Sometimes there can be a situation in which two scripts are trying to write different data to the same file. These are totally undesirable in the file handling technique. The solution to this problem is to lock the file when one script is accessing it. Due to locking of the file, simultaneous access can be avoided. The PHP uses flock function for locking the files.

Syntax of flock is
flock(file, lock, block)
where
file – name of the file which needs to be accessed.
Lock- kind of lock being used. Possible values are:
LOCK-SH – Shared Lock(reader). Allow other processes to access the file
LOCK_EX – Exclusive Lock(writer). Prevent other processes from accessing the file.
LOCK_UN – Release a shared or exclusive lock
LOCK_NB- Avoids blocking other processes while locking block is optional parameter.

Example

```
<?php
$file = fopen("myfile.txt","w+");
flock($file, LOCK_EX)// exclusive lock
fwrite($file, "I am writing this line to file");
flock($file,LOCK_UN);//release lock
fclose($file);
?>
```

Copying Files
We can copy one file into another file using copy() function.
Syntax:
Copy('source file','destination file');

Copying a file

```
<?php // copyfile.php
copy('testfile.txt', 'testfile2.txt') or die("Could not copy file");
echo "File successfully copied to 'testfile2.txt'";
?>
```

If you check your folder again, you'll see that you now have the new file *testfile2.txt* in it. By the way, if you don't want your programs to exit on a failed copy attempt, you could try the alternate syntax.

Moving a File
To move a file, rename it with the rename function.

Example
```
<?php // movefile.php
rename('testfile2.txt', 'testfile2.new');
else echo "File successfully renamed to 'testfile2.new'";
?>
```
You can use the rename function on directories, too. To avoid any warning messages, if the original file doesn't exist, you can call the file_exists function first to check.

Deleting a File
Deleting a file is just a matter of using the unlink function to remove it from the filesystem, as in Example 7-10.
Example 7-10. Deleting a file
```
<?php // deletefile.php
if (!unlink('testfile2.new')) echo "Could not delete file";
else echo "File 'testfile2.new' successfully deleted";
?>
```

2.18 Form Handling

PHP is used for form handling. For that purpose the simple form can be desgined in XHTML and the values of the fields defined on the form can be transmitted to the PHP script using GET and POST methods. For forms that are submitted via "GET " method, we obtain the form via the $_GET array variable. For forms that are submitted via "POST" method we obtain the form via the $_POST array variable.

Exercise:Create HTML form with one text box to get user's name. Also write PHP code to show length of entered name when, the HTML form is submitted.

Solution:

The $_GET Function

- The built-in $_GET function is used to collect values from a form sent with method="get"
- Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's URL) and has limits on the amount of information to send (max. 100 characters)
- This method should not be used when sending passwords or other sensitive information. However, because the variables are displayed in the URL, it is possible to bookmark the page
- The get method is not suitable for large variable values; the value cannot exceed 100 characters

The $_POST Function
- The built-in $_POST function is used to collect values from a form sent with method="post"

- Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send
- However, there is an 8 Mb max size for the POST method, by default (can be changed by setting the post_max_size in the php.ini file)

The $_GET Function Example

Form1.html

```
<html>
<body>
/* form submitted using 'get' method, action specifies next page which is to be loaded when button is clicked*/
<form action="welcome.php" method="get">
textbox is to take user input Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
Submit button is to submit the value <input type="submit" />
</form>
</body>
</html>
```

welcome.php

```
<html>
<body>
$_GET to receive the data sent from Form1.html Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
</body>
</html>
```

The $_ POST Function Example

form1.html

```
<html>
<body>
/* form submitted using 'post' method, action specifies next page which is to be loaded when button is clicked */ <form action="welcome1.php" method="post">
textbox is to take user input Name: <input type="text" name="fname" /> Age: <input type="text" name="age" />
Submit button is to submit the value to next page <input type="submit" />
</form>
</body>
</html>
```

welcome1.php

```
<html>
<body>
$_GET to receive the data sent from form1.html Welcome <?php echo $_POST["fname"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old!
</body>
</html>
```

Step 1:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML-PHP Demo</title>
</head>
<body>
<form method= “post” action= “http://localhost/getdata.php”>
Name:<input type= “text” name= “myname” size=”20” />
<br/>
<input type = “submit” name=”submit” value= “Submit” />
</form>
</body>
</html>
```

Step 2:

The PHP script to display the length of the submitted name is as written below:

```
<?php
print “The name is”;
print $_POST[‘myname’];
$len=strlen($_POST[‘myname’]);
print $len;
?>
```

Exercise: Create HTML form to enter one number. Write PHP code to display the message about number is odd or even.

Solution

Step1: The HTML form for accepting number is created as below-

```
<!DOCTYPE html>
<head><title>HTML-PHP DEMO</title>
</head>
<body>
        <form method=”post” action =”http://localhost/getdata.php”>
         Enter Number:<input type=”text” name=”mynum” size=”5”/>
        <br/>
        <input type=”submit” name=”submit” value=”submit”/>
        </form>
</body>
</html>
```

Step 2: The PHP script deciding whether the number is odd or even is given below-

```
<?php
print “The number is:”;
print $_POST[“mynum”];
$a=$_POST[“mynum”];
if($a%2==1)
        print”<br/> The number is odd”;
else
        print”<br/> The number is even”;
?>
```

EXERCISE: Create a form containing information sl.no,title of the book,publishers,quantity,price read the data from the form and display it using PHP script.

SOLUTION:

Step 1: Create an HTML page for inputting the data.

HTML Document[input.html]

```
<!doctype html>
<html>
<head>
<title>Book Order Form</title>
</head>
<body>
<h3>Enter the Book Data</h3>
<form method=”post” action=”http://localhost/php-examples/formdemo.php”>
<table>
<tr>
<td> Sr.No</td>
<td><input type=”text” name=”SLNo”></td>
</tr>
<tr>
<td>Book name</td>
<td><input type=”text” name=”BName”></td>
</tr>
<tr>
<td>Publisher</td>
<td><input type=”text” name=”PUBname”></td>
</tr>
<tr>
<td>Price</td>
<td><input type=”text” name=”Price”></td>
</tr>
<tr>
<td>Quantity</td>
<td><input type= “text” name=”Qty”></td>
</tr>
<tr>
<td><input type= “submit” name=”submit”></td>
<td><input type= “submit” name=”Clear”></td>
</tr>
</table>
</form>
</body>
</html>
```

Step 2: Create a PHP script which will read out the data entered by the user using HTML form . The code is as follows :

formdemo.php:

```
<html>
<head>
<title>Book Information</title>
</head>
<body>
<?php
$Bname=$_POST[‘BName’];
```

```
$PUBName=$_POST[‘PUBName’];
$Price=$_POST[‘Price’];
$Qty=$_POST[‘Qty’];
?>
<center><h3>Book Data</h3>
<table border=1>
<tr>
<th>Book Name</th>
<th>Publisher</th>
<th>Price</th>
<th>Quantity</th>
</tr>
<tr>
<td><?php print (“$Bname”);?></td>
<td><?php print (“PUBName”);?></td>
<td><?php print (“$Price”);?></td>
<td><?php print (“$Qty”);?></td>
</tr>
</table>
</center>
</body>
</html>
```

Step 3: Open some suitable web browser and enter the address for the HTML file which you have created in step 1. Now click on the submit button and the PHP file will be invoked. The output will then be as follows.

Exercise : Write a PHP program to accept a positive integer ‘N’ through a HTML form and to display the sum of all the numbers from 1 to N

```
<!DOCTYPE html>
<html>
<head>
<title>PHP Demo</title>
</head>
<body>
<form method= “post” action= “getdata.php”>
<input type= “text” name= “num”  />
<input type = “submit”  value= “Submit” />
</form>
</body>
</html>
```

```
<?php

//getting values from HTML form

$n=intval($_POST[‘num’]);

$sum=0;

for($i=1;$i<=$n;$i++)

        $sum=$sum+i;

echo “The sum of all numbers from 1 to “.$n.”is”.$sum;?>
```

2.18 PHP and MySQL:

Benefits of using PHP and MySQL:

PHP is a server side scripting language and it has an ability to create dynamic pages with customized features. Using PHP-MySQL user friendly and interactive web sites can be created. Both PHP and MySQL are open-source technologies that work hand-in-hand to create rich internet applications. The purchased code provides you the encrypted source code to prevent replication or modification, whereas open-source programs encourage users to utilize, scrutinize and customize the code.

Due to the availability of these technologies as free of cost, the cost effective web solutions can be created. PHP-MySQL are stable technologies and have cross platform compatibility. Hence the web application developed using these technologies becomes portable. Since HTML can be embedded within the PHP, there is no need to write seperate code for web scripting. Open-source coding has been checked and doubled checked by thousands or even millions of people around the world. Hence one can built the reliable web application using these technologies.

The PHP has got the support from several content management programs such sa wrodpress, Joomla, Drupal and so on. It has got a strong support for developing e-commerce applications using the technologies such as Ecommerce, Drupal and so on. The most popular web sites being developed using PHP and MySQL technologies are:

1. Facebook
2. WordPress
3. Wikipedia

Structured Query Language(SQL)

MySQL is a open source database product and can be downloaded from the web site Http://dev.mysql.com/downloads/mysql . MySQL is a kind of database in which the records are stored in an entity called tables. In the tables the data is arranged in the rows and columns. We can query a database to retrieve particular information. Query is a request or a question for the database. There is a common practice of making use of Structured Query Language(SQL).

Connecting PHP to MySQL:

The PHP function mysql_connect connects to the MySQL Server. There are three parameters that can be passed to this function. For example-

mysql_connect("localhost", "root", "mypassword") or die(mysql_error());

where

localhost- Local host on which the MySQL is running

root- Root

mypassword- Password

The database can be selected by using the command mysql_selecet_db.

For example:

mysql_select_db("test") will select the database named test.

Requesting MySQL Operations:

1. Creating Database:

We can create a database using the function mysql_query. The mysql_error() function is used to get the error messages if any command gets failed.

mysql_query function in php is used to pass a sql query to myql database.

Syntax:

mysql_query(string query[, resource link_identifier])

This function returns the query handle for select queries, TRUE/FALSE for other queries, or FALSE on failure.

Example

mysql_query("CREATE DATABASE mydb", $con)
The mysql_connect() function open a connection to a MySQL Server.
Syntax:
mysql_connect(string server, string username, string password)
Returns a MySQL link identifier on success, or FALSE on failure.
Example:
$conn=mysql_connect("localhost", "root", "password");
The mysql_close() function is used to close the database connection.
Syntax:
mysql_close(Connection)
PHP program for Creating the database:

```
<?php
$conn=mysql_connect("localhost", "root", "password"); //Make a sql connection
if(!$conn)
{
die('error in connection'.mysql_error());
}
if(mysql_query("CREATE DATABASE mydb", $conn)) //Create a database
{
print "Database Created";
}
else
{
print "Error Creating database:".mysql_error();
}
mysql_close($conn); //closing the database
?>
```

2. Selecting the database:
The database can be selected using the function **mysql_select_db().**
Syntax:
mysql_select_db(string database_name [, resource link_identifier])
where mysql_select_db() attempts to select existing database on the server associated with the specified link identifier. It returns TRUE on success, or FALSE on failure.
For example-

```
<?php
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
mysql_close($conn); //closing the database
?>
```

3. Counting Number of Rows:
The numbe rof rows present in the table can be obtained using mysql_num_rows functions.
Syntax:
int mysql_num_rows(resource $result)
This returns number of rows in result on success, or NULL on error.
Example:

```
<?php
```

```
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$num_rows=mysql_num_rows($result);
//Print number of rows
echo ""Total number of rows are $num_rows";
mysql_close($conn); //closing the database
?>
```

4. Counting number of fields:

The mysql_num_fields() is used to get number of fields of the table.

Syntax:

mysql_num_fields(resource_name)

It returns the number of fields present in the resource and false on failure.

Example:

```
<?php
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$result=mysql_query("Select id, name from my_table where id='1' ");
echo mysql_num_fields($result);
mysql_close($conn); //closing the database
?>
```

5. Creating Table:

Before creating the table a database must be created and within which the table can be created. Note that before creating a table, desired database must be selected.

Example:

```
<?php
$conn=mysql_connect("localhost", "root", "password"); //Make a sql connection
if(!$conn)
{die('error in connection'.mysql_error());
}
if(mysql_query("CREATE DATABASE mydb", $conn)) //Create a database
{print "Database Created";
}
else
{print "Error Creating database:".mysql_error();
}
mysql_select_db("mydb",$conn); // Before creating a table, database must be selected.
$query="CREATE TABLE my_table (id INT(4), name VARCHAR(20))";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

6. Inserting Data in table:

For inserting a data into the table we use the INSERT query. For Example

```
$query= "INSERT INTO my_table(id, name) VALUES (1, 'SHILPA')";
mysql_query($query, $conn); // Execution of Query
```

Here is a PHP script in which insert query is used to insert two records in the table

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$query= "INSERT INTO my_table(id, name) VALUES (1, 'SHILPA')";
mysql_query($query, $conn);
$query= "INSERT INTO my_table(id, name) VALUES (2, 'MONIKA')";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

Sometimes values that can be inserted in the table can be obtained from someother script and these values might be present in the variables. Insertion of such data can be done using $_POST variables. It is as shown below-

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$query= "INSERT INTO my_table(id, name) VALUES ('$_POST[MyId]', '$_POST[MyName]')";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

7. Displaying or Retrieving Records:

For displaying the records present in the database table, we use SELECT query. For Example

```
// Execution of Query for displaying the data
$result=mysql_query("SELECT * FROM my_table");
```

The above execution returns a result handle. Then the mysql_fetch_array() is used to retrieve a row of data as an array from a MySQL result handle.

Syntax:

mysql_fetch_array(result, result_type)

PHP Script for Displaying records:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
```

```
mysql_select_db(“mydb”,$conn);
//Execution of Query for dsiplaying the data
$result=mysql_query(“SELECT * FROM my_table”);
while($row=mysql_fetch_array($result))
{
echo $row[‘id’]. “” .$row[‘name’]; // Each record will be displayed
echo “<br/>”;
}
mysql_close($conn); //closing the database
?>
```

8. Finding the number of affected rows:

The mysql_affected_rows query is used to get number of affected rows in previous MySQL operation such as INSERT, DELETE, UPDATE queries.

Syntax:

mysql_affected_rows(connection)

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “password”);
if(!$conn)
{
die(‘error in connection’.mysql_error());
}
mysql_select_db(“mydb”,$conn);
$query= “INSERT INTO my_table(id, name) VALUES (1, ‘SHILPA’)”;
mysql_query($query, $conn);
$query= “INSERT INTO my_table(id, name) VALUES (2, ‘MONIKA’)”;
mysql_query($query, $conn);
echo “Number of rows affected are:”.mysql_affected_rows();
mysql_close($conn); //closing the database
?>
```

MySQL Functions:

mysql_connect():

This function opens a link to a MySQL server on the specified host (in this case it's localhost) along with a username (root) and password (q1w2e3r4/). The result of the connection is stored in the variable $db.

mysql_select_db():

This tells PHP that any queries we make are against the mydb database.

mysql_query():

Using the database connection identifier, it sends a line of SQL to the MySQL server to be processed. The results that are returned are stored in the variable $result.

mysql_result():

This is used to display the values of fields from our query. Using $result, we go to the first row, which is numbered 0, and display the value of the specified fields.

mysql_result($result,0,"position")):

This should be treated as a string and printed.

2.19 COOKIES

Introduction to cookies:
Cookie is a small file that server embeds in the user's machine. Cookies are used to identify the users. A cookie consists of a name and a textual value. A cookie is created by some software system on the server. In every HTTP communication between browser and server, a header is included. The header stores the information about the message. The header part of http contains the cookies. There can be one or more cookies in browsers and server communications.

PHP Support for Cookies:
PHP can be used to create and retrieve the cookies. The cookie can be set in PHP using the functions called setcookie().
The syntax for setcookie is -
setcookie(name, value,expire_period, path, domain)

Example PHP program to set cookie:

```
<?php
$cookie_period=time()+60*60*24*30;
setcookie("Myname", "Monika", $cookie_period);
?>
```

Note that you have got the blank screen it indicates that the cookie is set . In above PHP document we have set the PHP script for one month.

Example: [CookieRead.php]

```
<html>
<head><title>Reading cookies</title>
<body>
<?php
if(isset($_COOKIE["Myname"]))
        echo "<h3>Welcome".$_COOKIE["Myname"]. "!!!</h3>";
else
        echo "Welcome guest!";
?>
</body>
</html>
```

The isset function is used for checking whether or not the cookie is set. Then using the $_COOKIE the value of the cookie can be retrieved.

Exercise: Write a PHP Script to create a new database table with 4 fields of your choice and perform following database operations. i)insert ii)update iii)Delete
Solution:
We will create a table in the database test. The name of the table is mytable. Then we will insert the record into the table using the INSERT command, update particular field of the record using the command UPDATE and delete the record using the command DELETE.

PHP Document[DBDemo.php]

```
<?php
// Make a MySQL Connection
mysql_connect("localhost","root","mypassword") or die(mysql_error());
mysql_select_db("test") or die(mysql_error());
echo "Connected to database";
mysql_query("CREATE TABLE mytable(id INT NOT NULL AUTO _INCREMENT,PRIMARY
KEY(id), name VARCHAR(30),phone INT,emailId VARCHAR(30))") or die(mysql_error());
```

```
print"<br/>";
echo "Table Created";
//Insert a row of information into tabele "example"
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('abcd','1111','abc@gmail.com')") or die (mysql_error());
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('xyz','2222','xyz@gmail.com')") or die (mysql_error());
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('Kumar,'3333','pqr@gmail.com')") or die (mysql_error());
print"<br/>";
echo "Data Inserted";
$result=mysql_query("SELECT * from mytable")
or die(mysql_error());
print"<br/>";
print"<b> User Databse</b>";
echo"<table border='1'>";
echo"<tr><th>ID</th><th>Name</th> <th> Phone</th><th> Email_ID</th></tr>";
while($row=mysql_fetch_array($result))
{
//Print out the contents of each row into a table
echo"<tr><td>";
echo $row['id'];
echo"</td><td>;
echo $row['name'];
echo"</td><td>";
echo $row['phone'];
echo"</td><td>";
echo $row['emailId'];
echo"</td><tr>";
}
echo "<table>";
$result=mysql_query("UPDATE mytable SET phone="5555" where phone="2222")
or die(mysql_error());
print"<br/>";
echo "Data Updated";
$result=mysql_query("SELECT * from mytable")
or die(mysql_error());
print"<br/>";
print"<b> User Datbase</b>";
echo"<table border='1'>";
echo"<tr><th>ID</th><th>Name</th><th>Phone</th><th>Email-ID</th></tr>";
while($row=mysql_fetch_array($result))
{
// Print out the contentsof each row into a table
echo"<tr><td>";
echo $row['id'];
echo"</td><td>;
echo $row['name'];
echo"</td><td>";
echo $row['phone'];
echo"</td><td>";
echo $row['emailId'];
echo"</td><tr>";
```

```
}
echo “<table>”;
result=mysql_query(“DELETE from mytable  where phone=”3333”)
or die(mysql_error());
print”<br/>”;
echo “Data Deleted”;
$result=mysql_query(“SELECT * from mytable”)
or die(mysql_error());
print”<br/>”;
print”<b> User Datbase</b>”;
echo”<table border=’1’>”;
echo”<tr><th>ID</th><th>Name</th><th>Phone</th><th>Email-ID</th></tr>”;
while($row=mysql_fetch_array($result))
{
// Print out the contentsof each row into a table
echo”<tr><td>”;
echo $row[‘id’];
echo”</td><td>;
echo $row[‘name’];
echo”</td><td>”;
echo $row[‘phone’];
echo”</td><td>”;
echo $row[‘emailId’];
echo”</td><tr>”;
}
echo “<table>”;
```

Exercise: CREATE a HTML form “result.html” with a text box and a submit button to accept registration number of the student. Write a “result.php” code to check the status of the result from the table to display whether the student has “PASS” or “FAIL” status. Assume that the Mysql database “my_db” has the table “result_table” with two columns REG_NO and STATUS

Step 1:
Create a database named my_db. Create a table result_table for this database and insert the values in this table. The table is created as follows:

Step 2:
Create an HTML form to accept the registration number, the HTML document is as follows-
result.html

```
<!DOCTYPE html>
<html>
<head>
<title>STUDENT RESULT</title>
</head>
<body>
<form name = “myform” method= “post” action= “http://localhost/php-examples/result.php”>
<input type= “text” name= “reg_no” />
<input type = “submit” value= “Submit” />
</form>
</body>
</html>
```

Step 3: Create a PHP Script to accept the registration number. This PHP script will connect to MYSQL database and the status (PASS or FAIL) of the corresponding registration number will be displayed.

Result.php

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$reg_no = intval($_POST['reg_no']);
$result=mysql_query("SELECT REG_NO, STATUS FROM result_table where REG_NO=$reg_no");
while($row=mysql_fetch_array($result))
{
echo $row['REG_NO']. "is". $row['STATUS'];
echo "<br/>";
}
mysql_close($conn);
?>
```

Step 4: Load the HTML form created in Step 2 and click the submit button by entering some registration number

Exercise : Write a PHP program to delete a record from the result_table:

Solution:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$reg_no = intval($_POST['reg_no']);
$result=mysql_query("DELETE FROM result_table where REG_NO=$reg_no");
while($row=mysql_fetch_array($result))
{
echo $row['REG_NO']. "is". $row['STATUS'];
echo "<br/>";
}
mysql_close($conn);
?>
```

Exercise: Write a user defined function 'CalculateInterest' using PHP to find the simple interest to be paid for a loan amount. Read the loan amount, the number of years and the rate of interest from a database table called LOANDETAILS having three fields AMT, YEARS, and RATE and Calculate the interest using the user defined function.

Solution:

Step 1:

Create a database table named LOANDETAILS having three fields AMT, YEARS and RATE. Insert the values in it. The sample table will be as follows -

AMT	YEARS	RATE
10000	5	6.9

Step 2:
The PHP code for calculating the simple interest the above values in a function will be as follows:
Interest.php

```
<?php
function CalculateInterest()
{
<?php
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “”);
if(!$conn)
{
die(‘error in connection’.mysql_error());
}
mysql_select_db(“mydb”,$conn);
$result=mysql_query(“SELECT * FROM LOANDETAILS”);
$row=mysql_fetch_array($result);
$amount=$row[‘AMT’];
$rate=$row[‘RATE’];
$years=$row[‘YEARS’];
$interest=($amount*$rate*$years)/100;
mysql_close($conn);
return $interest;
}
print “Simple Interest =”.CalculateInterest();
}
```

Exercise: Consider a table PRODUCT_DETAILS with PID, PNAME and UNIT_PRICE. Write HTML form to accept PID and QUANTITY_REQUIRED from the user and dsiplay the total amount to be paid in another textbox. Get the unit price for given PID from a database table.
Solution:
Bill.php

```
<!DOCTYPE html>
<html>
<head>
<title>PHP Demo</title>
</head>
<body>
<form method= “post”>
<input type= “text” name= “PRODUCT_ID” />
<input type= “text” name= “QUANTITY” />
<input type = “submit” value= “Submit” />
<?php
//getting values from HTML Form
$id=intval($POST[‘PRODUCT_ID’]);
$qty=intval($POST[‘QUANTITY’]);
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “”);
```

```
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$result=mysql_query("SELECT UNIT_PRICE FROM product_table where PID=$id");
$row=mysql_fetch_array($result);
$price=$row['UNIT_PRICE'];
$total=$price*$qty;
mysql_close($conn);
?>
<input type= "text" name= "amount" value="<?php echo @$total;?>"/>
</form>
</body>
</html>
```

Exercise:Write a Script to insert and delete record from database.

Index.html:

```
<!DOCTYPE html>
<body bgcolor="#bbffa0">
<center>
<br><h1>Database Application<h1>
<h3>You can Insert, delete and search your data</h3>
<a href="insert.php">Insert</a>
<br><br>
<a href="delete.php">Delete</a>
<br><br>
</center>
<body>
```

Insert.php:

```
<!doctype html>
<body bgcolor="#aacb00">
<form method="POST" action="insert_process.php">
<center>
<h1>Insert your Details here</h1>
<h2>
Name:
<br><input type="text" name="name">
<br>
<br>
Register Number:
<br><input type="text" name="reg">
<br><br>
Age:
<br> <input type="text" name="age">
<br><br><br>
<input type="submit"  value="Insert" name="insert">
<br><br>
</h2>
</center>
</form>
```

```
</body>
```
Insertprocess.php
```
</html>
<?php
// Create connection
$conn = new mysqli("localhost", "root", "", "itessentials");
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$name=$_POST['name'];
$regno=$_POST['reg'];
$age=$_POST['age'];
$sql = "INSERT INTO student (name, reg_no, age)VALUES ('$name','$regno','$age')";
if ($conn->query($sql) === TRUE) {
   echo "Inserted successfully";
}
?>
<html>
<br>
<a href="database.html">Goto Home</a>
</html>
```
Delete.php:
```
<!DOCTYPE html>
<body bgcolor="#bbccf0">
<br>
<center>
<h1>Delete Your Data</h1><h2>
<form method="POST" action="del_process.php">
Enter the Register no :<input type="text" name="reg">
<br>
<br>
<input type="submit" value="DeleteData"></h2>
</center>
</form>
</body>
</html>
```
Deleteprocess.php:
```
<?php
$conn = new mysqli("localhost", "root", "", "itessentials");
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$regno=$_POST["reg"];
$sql = "DELETE FROM student WHERE reg_no=$regno";
if ($conn->query($sql) === TRUE) {
   echo "Record Deleted successfully";
}
?>
<html><br>
<a href="database.html">Goto Home</a>
</html>
```

Exercise:Write a database application to insert and search the data.

```
<!doctype html>
<html lang="en">
<head>
		<title>Simple Information Reterival System</title>
</head>
<body background="lighthouse.jpg">
	<h1><center>Simple Information Reterival System</center></h1>
	<center><ul>
		<h2><li><a href="create.php"><strong>Create</strong></a> - add a user</h2></li>
		<h2><li><a href="search.php"><strong>Read</strong></a> - find a user</h2></li>
	</ul></center>

</body>
</html>
```

Create.php:

```
<html>
<body bgcolor="cyan">
<h2>Add a user</h2>

<form method="post" action="insert.php">
	<label for="firstname">First Name</label><br>
	<input type="text" name="firstname" id="firstname" required><br>
	<label for="lastname">Last Name</label><br>
	<input type="text" name="lastname" id="lastname" required><br>
	<label for="dob">Date of Birth</label><br>
	<input type="text" name="dob" id="dob" required><br>
      <label for="email">Email Address</label><br>
	<input type="email" name="email" id="email" required><br>
	<label for="age">Age</label><br>
	<input type="number" name="age" id="age" required><br>
	<label for="location">Location</label><br>
	<input type="text" name="location" id="location" required><br><br>
	<input type="submit" name="submit" value="Submit">
</form>
*All feilds are mandate.<br>
<a href="index.php">Back to home</a>
</body>
</html>
```

Insert.php:

```
<!dhtml>
<body bgcolor="pink">
<?php
$conn = new mysqli("localhost","root","","itessentials");

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}$firstName=$_POST['firstname'];
$lastName=$_POST['lastname'];
$dob=$_POST['dob'];
$email=$_POST['email'];
```

```
$age=$_POST['age'];
$location=$_POST['location'];
$query="INSERT INTO user_details (first_name, last_name,dob, email, age,location) VALUES (?,
?,?,?,?,?)";
$stmt = mysqli_prepare($conn,$query);
mysqli_stmt_bind_param($stmt,"sssiss",$firstName,$lastName,$dob, $email,$age,$location);
mysqli_stmt_execute($stmt);

echo "<h2>Thank You ".$firstName ."You are Registered Successfully</h2>";
echo "<a href='index.php'>Back to home</a>";
mysqli_close($conn); ?>
</body>
```

Search.php:

```
</html>
<h1>Search a User</h1>
<form method="post" action="retrieve.php">
        <table><tr style="width=100%">
        <td>Enter search query  <input type="text" name="searchquery" id="se"><br></td></tr>
        </tr></table>
        <table><tr>
        <td> <label for="search">Search by</label><br></td>
        <td>  <input type="radio" name="searchby" value="first_name" checked> First
Name<br></td>
        <td> <input type="radio" name="searchby" value="last_name" > Last Name<br></td>
        <td> <input type="radio" name="searchby" value="email"> Email<br></td>
        <td> <input type="radio" name="searchby" value="location"> Location<br></td>
        </tr>  </table>
        <input type="submit" name="submit" value="Submit">
        </form>
```

Retrieve.php:

```
<?php
$host="localhost";
$username   = "root";
$password   = "";
$dbname     = "itessentials";
$conn = new mysqli($host, $username, $password,$dbname);
$searchby=$_POST['searchby'];
$searchquery=$_POST['searchquery'];
$query="SELECT * FROM user_details WHERE ".$searchby ."= '". $searchquery."'";
$result=mysqli_query($conn,$query);
while($row=mysqli_fetch_array($result,MYSQLI_ASSOC)){
        echo "<h1>Welcome ".$row['first_name']." ".$row['last_name']."</h1>";
        echo "<h3>Age ".$row['age']."</h1>";
        echo "<h3>Registered email ".$row['email']."</h1>";
        echo "<h3> Location".$row['location']."</h1>";
        }// Check connection
if ($conn->connect_error) {    die("Connection failed: " . $conn->connect_error);
}
mysqli_close($conn); ?>
```

Web Server	Database Server
Web server makes use of the languages like PHP , ASP, JSP. It makes use of the protocols such as FTP and HTTP	The database server has its own specific program language or query language.
Web server is used to save the static and dynamic contents and pages of website	Database server deals with the storing and managing the data of a computer or computer programs
Web server only performs web based services	Database server can manage the web based, enterprise based services at the same time
Apache HTTP server, Microsoft Internet Information Services (IIS), Google Web Server(GWS) and Sun Java Systems web server are examples of web server	Oracle , SAP, MySQL and DB2 are some common examples of database server.

Arithmetic Operators Example

```
<html>
<body>
<?php
$i=(20 + 10);
$j=($i - 5);
$k=($j * 4);
$l=($k / 2);
$m=($l % 5);
echo "i = ".$i."<br/>"; echo "j = ".$j."<br/>"; echo "k = ".$k."<br/>"; echo "l = ".$l."<br/>"; echo
"m = ".$m."<br/>";
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

i = 30
j = 25
k = 100
l = 50
m = 0

Increment and Decrement Operators:
It is also called as the unary operator. It usually increments or decrements the value by one.

Operator	Name	Description
++$a	Pre-increment	Increments $a by one, then returns $a
$a++	Post-increment	Returns $a, then increments $a by one
--$a	Pre-decrement	Decrements $a by one, then returns $a
$a--	Post-decrement	Returns $a, then decrements $a by one

Increment and Decrement Operators Example

```
<html>
<body>
<?php
$i=10;
$j=20;
$i++;
$j++;
echo $i."<br/>";
echo $j."<br/>";
$k=$i++;
```

```
$l=++$j;
echo $k."<br/>"; echo $l;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

11
21
11
22

Assignment Operators in PHP

Assignment operator is used to assign a value to a variable

Operator	Example	Is the same as
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Logical operators

Logical operators produce true-or-false results, and therefore are also known as Boolean operators. There are four of them.

Operator	Description	Example
&&	And	$j == 3 && $k == 2
and	Low-precedence and	$j == 3 and $k == 2
||	Or	$j < 5 || $j > 10
or	Low-precedence or	$j < 5 or $j > 10
!	Not	! ($j == $k)
xor	Exclusive or	$j xor $k

```
<?php
$a = 1; $b = 0;
echo ($a AND $b) . "<br>";
echo ($a or $b) . "<br>";
echo ($a XOR $b) . "<br>";
echo !$a . "<br>";
?>
```

Output:
0
1
1
0

Relational Operators or Comparison Operator
Relational operators test two operands and return a Boolean result of either TRUE or FALSE. There are three types of relational operators: *equality*, *comparison*, and *logical*. It is also called as comparison operator.
Comparison operators are generally used inside a construct such as an if statement in which you need to compare two items. For example, you may wish to know whether a variable you have been incrementing has reached a specific value, or whether another variable is less than a set value, and so on
Note the difference between = and ==. The first is an assignment operator, and the second is a comparison operator.

Operator	Description
==	Equality
===	Identity(Checks both value and type)
!=	Not Equal
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Example:
```
<?php
$a = "1000";
$b = "+1000";
if ($a == $b) echo "1";//both will converted to same type and the it will compare.
if ($a === $b) echo "2";// this will not be executed. Because $a and $b are of different type
?>
```
Output:
1
Example2:
```
<?php
$a=22;
$b=25;
$c=30;
if(($a>$b)&($a>$c))
echo "A is greater";
elseif($b>$c)
echo "B is greater";
else
echo "C is greater";
?>
```
Output:
C is greater

String Operators in PHP

Operator	Name	Example	Result
.	Concatenation	$a = "Hello" $b = $a . " world!"	$b = "Hello world!"
.=	Concatenation Assignment	$a = "Hello" $a .= " world!"	$a = "Hello world!"

String Operators Example

```
<html>
<body>
<?php
$a = "Hello";
$b = $a . " Friend!";
echo $b;
echo "<br/>";
$c="Good";
$c .= " Day!";
echo $c;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Hello Friend!
Good Day!

Boolean Operators
Boolean operators AND, OR, and NOT are used to manipulate logical statements. Boolean operators are the core operators used in digital control systems as well as computer systems. AND and OR are binary operators, while NOT is a unary operator.

Operator	Meaning
And &&	The binary AND operation is performed
Or ||	The binary OR operation is performed
Xor	The XOR operation will be performed

2.19 Displaying Messages on the Web page

- The PHP is a server side scripting language and is used to submit the web page to the client browser. Hence it is a standard method of embedding the PHP code within the XHTML document. That means we can use the XHTML tags in PHP while displaying the output.
- The print function is used to create simple unformatted output. For example : The string can be displayed as follows: **print "I am proud of my country**
- The numeric value can also be displayed using the print. For example : **print(100);**

- PHP also makes use of the printf function used in C. For example: **printf("The student %d has %f marks", $roll_no, $marks);**

Example:

```
<html>
<head>
<title>Output Demo</title>
</head>
<body>
<?php
print "<h2>Welcome to my website </h2>";
print "<hr/>";
$roll_no=1;
$name=AAA;
printf( "<b> The roll number: %d </b>", $roll_no);
print "<br/><b>";
printf(" The name :%s",$name);
print "</b>";
?>
</body>
</html>
```

2.20 CONSTANTS

Constants are similar to variables, holding information to be accessed later, except that they are what they sound like—constant. In other words, once you have defined one, its value is set for the remainder of the program and cannot be altered.

Constant in PHP - define() function is used to set a constant

It takes three parameters they are:

- Name of the constant
- Value of the constant
- Third parameter is optional. It specifies whether the constant name should be case-insensitive. Default is false

Constant string Example

```
<html>
<body>
<?php
/* Here constant name is 'Hai' and 'Hello Friend' is its constant value and true indicates the constant value is case-insensitive */
define("Hai","Hello Friend",true);
echo hai;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Hello Friend

PHP Example to calculate the area of the circle

```
<html>
<body>
<?php
// defining constant value PI = 3.14
define("PI","3.14");
$radius=15;
```

```
$area=PI*$radius*$radius;
echo "Area=".$area;
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Area=706.5

Predefined Constants

PHP comes ready-made with dozens of predefined constants that you generally will be unlikely to use as a beginner to PHP. However, there are a few—known as the *magic constants*—that you will find useful. The names of the magic constants always have two underscores at the beginning and two at the end.

Magic constant	Description
__LINE__	The current line number of the file.
__FILE__	The full path and filename of the file. If used inside an `include`, the name of the included file is returned. In PHP 4.0.2, __FILE__ always contains an absolute path with symbolic links resolved, whereas in older versions it might contain a relative path under some circumstances.
__DIR__	The directory of the file. If used inside an `include`, the directory of the included file is returned. This is equivalent to *dirname*(__FILE__). This directory name does not have a trailing slash unless it is the root directory. (Added in PHP 5.3.0.)
__FUNCTION__	The function name. (Added in PHP 4.3.0.) As of PHP 5, returns the function name as it was declared (case-sensitive). In PHP 4, its value is always lowercase.
__CLASS__	The class name. (Added in PHP 4.3.0.) As of PHP 5, returns the class name as it was declared (case-sensitive). In PHP 4, its value is always lowercase.
__METHOD__	The class method name. (Added in PHP 5.0.0.) The method name is returned as it was declared (case-sensitive).
__NAMESPACE__	The name of the current namespace (case-sensitive). This constant is defined at compile time. (Added in PHP 5.3.0.)

Example:

```
<?php
echo "This is line " . __LINE___. " of file " . __FILE__;
?>
```

Output:

This is line 2 of file sample.php

This causes the current program line in the current file (including the path) being executedto be output to the web browser.

2.21 FLOW CONTROL AND LOOP

2.21.1 The if Statement in PHP

- The **if statement**, the **if...else** statement or **if....elseif** statements are used as selection statements. The selection is based on some condition.
- If statement executes some code only if a specified condition is true

Syntax:

```
if (condition) {
code to be executed if condition is true;
}
```

The if Statement Example

```
<html>
<body>
<?php
$i=0;
/* If condition is true, statement is executed*/
if($i==0)
echo "i is 0";
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:

i is 0

The if…else Statement in PHP
If…else statement executes some code if a condition is true and some another code if the condition is false

Syntax:

```
if (condition)
{
code to be executed if condition is true;
}
else
{
code to be executed if condition is false;
}
```

The if…else Statement Example

```
<html>
<body>
<?php
$i=1;
/* If condition is true, statement1 is executed, else statement2 is executed*/
if($i==0)
echo "i is 0";   //statement1
else
echo "i is not 0";         //statement2
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:
i is not 0

The if…elseif…else Statement in PHP
If…elseif…else statement selects one of several blocks of code to be executed

Syntax:

```
if (condition)
 {
code to be executed if condition is true;
}
elseif (condition)
 {
code to be executed if condition is true;
}
else
{
code to be executed if condition is false;
}
```

The if…elseif…else Statement Example (Comparing two numbers)

```
<html>
<body>
<?php
$i=22;
$j=22;
/* If condition1 is true, statement1 is executed, if condition1 is false and condition2 is true, statement2 is executed, if both the conditions are false statement3 is executed */
if($i>$j)
echo "i is greater"; //statement1 elseif($i<$j)
echo "j is greater";      //statement2
else
echo "numbers are equal";      //Statement3
?>
<body>
</html>
```

OUTPUT of the above given Example is as follows:

numbers are equal

2.21.2 Switch Statements

Similar to if statement the switch statement can also be used for selection. Following is a simple PHP script for demonstrating switch statements

Syntax:

```
switch (n)
{ case label1:
code to be executed if n=label1;
break;
case label2:
code to be executed if n=label2;
break;
...
default:
code to be executed if n is different from all labels;
}
```

PHP Program

```
<?php
$today=getdate();
switch($today[‘weekday’])
{
case “Monday”: print “Today is Monday”;
                break;
case “Tuesday”: print “Today is Tuesday”;
                break;
case “Wednesday”: print “Today is Wednesday”;
                break;
case “Thursday”: print “Today is Thursday”;
break;
case “Friday” : print “Today is Friday”;
break;
case “Saturday” : print “Today is Saturday”;
break;
case “Sunday”: print “Today is Sunday”;
break;
default: print “Invalid Input”;
}
?>
```

2.21.3 Loop Statements

- The while, for and do-while statements of PHP are similar to Javascript.
- Following is a simple PHP script which displays the first 10 number.

For loop in PHP

PHP for loop executes a block of code, a specified number of times

Syntax:

```
for (initialization; test condition; increment/decrement)
 {
code to be executed;
}
```

For loop Example

```
<html>
<body>
<?php
echo "Numbers from 1 to 20 are: <br>";
for ($x=1; $x<=20; $x++) {
echo "$x ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

Numbers from 1 to 20 are:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Example 2:

```
<?php
$sum=0;
$n=10;
for($i=1;$i<=$n;$i++)
{
$sum=$sum+$i;
}
echo "The sum of  ".$n." Natural numbers is: ". $sum;
?>
```

Output:

The sum of 10 Natural numbers is: 55

While Loop in PHP

While loop, loops through a block of code as long as the specified condition is true

Syntax:

```
while (condition)
{
code to be executed;
}
```

While Loop Example

```
<html>
<body>
<?php
$i=1;
/* here <condition> is a Boolean expression. Loop body is executed as long as condition is true*/
while($i<5){
echo "i is = $i <br>";
$i++;
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

Do While loop in PHP

Do while loop will always execute the block of code once, it will then check the condition, and if the condition is true then it repeats the loop

Syntax:

```
do {
code to be executed;
```

} while (condition);

Do While loop Example

```
<html>
<body>
<?php
$i=1;
/* here <condition> is a Boolean expression. Please note that the condition is evaluated after executing the loop body. So loop will be executed at least once even if the condition is false*/
do
{
echo "i is = $i <br>";
$i++;
}while($i<5);
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

Break statement

Break statement is used to terminate the loop. After the break statement is executed the control goes to the statement immediately after the loop containing break statement

Break statement example

```
<html>
<body>
<?php
/* when $i value becomes 3, the loop is Terminated*/
for($i=0;$i<5;$i++)
{
if($i==3)
break;
else
echo "$i ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

```
0 1 2
```

Continue statement

There are cases in which, rather than terminating a loop, you simply want to skip over the remainder of iteration and immediately skip over to the next. Continue statement is used to skip a particular iteration of the loop.

Continue statement example

```
<html>
<body>
<?php
/* when $i value becomes 3, it will skip the particular of the loop*/
for($i=0;$i<=5;$i++)
{
if($i==3)
continue;
else
echo "$i ";
}
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

0 1 2 4 5

2.21.4 Examples based on Control Statement

Write a PHP script to display the squares and cubes of 1 to 10 numbers.

Sol :

```
<html>
<head>
<title> Square and cube Table </title>
</head>
<body>
<center>
<?php
print “<table border=1”>”;
print “<tr>”;
print “<th> Numbers</th>”;
print “<th> Squares</th>”;
print “<th> Cube</th>”;
print “</tr>”;
for($i=1;$i<=10;$i++)
{
print “<tr>”;
print “<td>$i”;
print “</td>”;
print “<td>”;
print $i*$i;
print “</td>”;
print “<td>”;
print pow($i,3);
print “</td>”;
print “</tr>”;
}
print “</table>”;
?>
</center></body></html>
```

Example Programs:
Write a PHP Script to compute the sum and average of N numbers.

PHP Program

```
<html>
<head>
<title> Sum and Average </title>
</head>
<body>
<center>
<?php
$sum=0;
for($i=1;$i<=10;$i++)
{
        $sum += $i;
}
$avg=$sum/10;
print “The sum is : $sum”;
print “<br/>”;
print “The average is : $avg”;
?>
</center>
</body>
</html>
```

Write PHP programs to print whether current year is leap year or not.
Sol :

```
<html>
<head>
<title> Leap year demo</title>
</head>
<body>
<?php
$year=2016;
print “<br/>”;
if($year%4==1)
{
printf(“Year %d is not a leap year”,$year);
}
else
{
printf(“Year %d is a leap year”,$year);
}
?>
</body>
</html>
```

Write PHP programs to print whether given number is odd or even.

Sol :

```
<html>
<head>
<title> Even Odd Demo</title>
```

```
</head>
<body>
<?php
for($i=1;$i<=10;$i++)
{
$num=$i;
print “<br/>”;
if($num%2==1)
{
print(“Number %d is Odd”,$num);
}
else
{
print(“Number %d is Even”,$num);
}
}
?>
</body>
</html>
```

Write PHP Script to compute the sum of positive integer upto 30 using do-while statement.
Sol:

```
<html>
<head>
<title> Sum of Integer</title>
</head>
<body>
<?php
$sum=0;
$i=1;
do
{
$sum=$sum+$i;
$i++;
}while($i<=30);
printf(“The sum of first 30 positive integers is %d”,$sum);
?>
</body>
</html>
```

Write PHP Script to compute factorial of ‘n’ using while or for loop construct.
Sol:

```
<html>
<head>
<title> Factorial Program </title>
</head>
<body>
<?php
$n=5;
$factorial=1;
for($i=$n;$i>=1;$i--)
{
$factorial=$factorial*$i;
```

```
}
echo “Factorial of $n is $factorial”;
?>
</body>
</html>
```

Write PHP Script to display Fibonacci of length 10.
Sol:

```
<html>
<head>
<title>Fibonacci Series</title>
</head>
<body>
<?php
$i=1;
$j=1;
print “<b> Fibonacci Series <br/> </b> “;
printf(“%d,%d”,$i,$j);
for($count=1;$count<9;$count++)
{
        $k=$i+$j;
        $i=$j;
        $j=$k;
        printf(“%d”,$k);
}
?>
</body>
</html>
```

Construct a PHP Script to compute the squareroot, square, cube and Quad of 10 numbers.
Sol :

```
<html>
<head>
<title> SQUARE CUBE QUAD DEMO </title>
</head>
<body>
<?php
print”<b>Table</b><br/>”;
print”<table border=’1’>”;
print “<tr><th>num</th><th>Sqr</th><th>Cube</th><th>Quad</th></tr>”;
for($count=1;$count<=10;$count++)
{
$sq=$count * $count;
$cube=$count*$count*$count;
$quad=$count*$count*$count*$count;
print “<tr><th>$count</th><th>$sq</th><th>$cube</th><th>$quad</th></tr>”;
}
print “</table>”;
?>
</body>
</html>
```

With the use of PHP, switch case and if structure perform the following and print appropriate message. (i) Get today's date (ii) If date is 3, it is dentist appointment (iii)If date is 10, go to conference (iv) If date is other than 3 and 10, no events are scheduled.

Sol:

```
<html>
<body>
<?php
echo "Today date is". date("d/m/y")."<br/>";
if(date("d")<3) || (date("d")>10))
        echo "No Event!!";
else if((date("d")>3 && date("d")<10))
        echo "No Event!!";
else
{
switch(date("d"))
{
        case 3 :echo "Dentist Appointment";
        break;
        case 10: echo "Go to Conference";
        break;
}
}
?>
</body>
</html>
```

Write a PHP code to display the following pattern

```
1
01
101
0101
10101
```

Sol:

```
<html>
<head>
</head>
<body>
<?php
for($i=0;$i<7;$i++)
{
        for($j=1;$j<$i;$j++)
        {
                if($i+$j)%2==0)
                {
                        printf("0");
                }
                else
                {
                        printf("1");
                }}
print "<br/>";
```

```
}
?>
</body>
</html>
```

2.22 Arrays

- Arrays is a collection of similar type of elements but in PHP you can have the elements of mixed type together in single array.
- In each PHP, each element has two parts **Key** and **Value**.
- The key represents the index at which the value of the element can be stored.
- The **keys** are positive integers that are in ascending order.

2.22.1 Array Creation

In PHP there are two types of arrays -

3. Indexed Array: Indexed array are the arrays with numeric index. The array values can be stored from index 0. For example -

```
<html>
<head>
<title> PHP Indexed arrays</title>
</head>
<body>
<?php
$names=array("AAA","BBB","CCC");
print_r($names);//print array structure
?>
</body>
</html>
```

Here values gets stored at corresponding index as follows -

```
$mylist[0]=10;
$mylist[1]=20;
$mylist[2]=30;
$mylist[3]=40;
$mylist[4]=50;
```

We can directly assign some value at specific index.

```
$mylist[5]=100;
```

4. Associated Array : Associated arrays are the arrays with named keys. It is a kind of array with **name** and **value** pair. For example -

```
<html>
<head>
<title> PHP Associated Array</title>
</head>
<body>
<?php
$city["AAA"]="Pune";
$city["BBB"]="Mumbai";
$city["CCC"]="Chennai";

//printing array structures
print_r($city);
```

```
?>
</body>
</html>
```

2.22.2 Accessing Array Elements

- Using an array subscript we can access the array element. The value of subscript is enclosed within the square brackets. For example -
  - $citycode[‘Pune’]=005;
  - $Name[0]=”Chitra”;
- Multiple values can be set to a single scalar variable using array. For example -

  $people = array(“Meena”,”Teena”,”Heena”);
  list($operator,$accountant,$manager)=$people;
- By this assignment Meena become operator, Teena becomes accountant and Heena becomes manager.

Consider an associative array called person_age with name and age of 10 person. Write a PHP program to calculate the average age of this associative array.

Sol:

```
<html>
<head>
<title> PHP Associative array</title>
</head>
<body>
<?php
$person_age=array(
                ‘AAA’ => 10, ‘BBB’ => 22, ‘CCC’ => 45, ‘DDD’ => 31, ‘EEE’ => 33, ‘FFF’ =>
25,             ‘GGG’ => 56 , ‘HHH’ => 73, ‘III’ => 35, ‘JJJ’ => 47);
$sum=array_sum($person_age);
print(“The sum of all the ages is $sum”);
$avg=$sum/10;
print(“<br/> The average is $avg”);
?>
</body>
</html>
```

Multidimensional array in PHP

Multidimensional array is an array containing one or more arrays

Multidimensional array Example

```
<html>
<body>
<?php
/* Here $flower_shop is an array, where rose, daisy and orchid are the ID key which indicates rows
and points to array which have column values. */

$flower_shop = array(

"rose" => array( "5.00", "7 items", "red" ), "daisy" => array( "4.00", "3 items", "blue" ), "orchid" =>
array( "2.00", "1 item", "white" ), );
/* in the array $flower_shop['rose'][0], ‘rose’ indicates row and ‘0’ indicates column */
echo "rose costs ".$flower_shop['rose'][0]."items: ".$flower_shop['rose'][1]."<br>";
echo "rose costs ".$flower_shop['daisy'][0]."items: ".$flower_shop[' daisy '][1]."<br>";
```

```
echo "rose costs ".$flower_shop['orchid'][0]."items: ".$flower_shop[‘orchid '][1]."<br>";
?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:

rose costs 5.00, and you get 7 items.
daisy costs 4.00, and you get 3 items.
orchid costs 2.00, and you get 1 item.

Functions Dealing with Arrays

9. is_array
check whether a variable is an array

```
if(is_array($array)
{
Return true;
}
Else
Return false;
```

10. Count
count all the elements in the top level of an array.

```
echo count($fred);
```

11. sort
Sorting is so common that PHP provides a built-in function.

```
sort($fred);
```

12. shuffle
elements of an array to be put in random order

```
shuffle($cards);
```

13. explode
Sveral items separated by a single character (or string of characters) and then place each of these items into an array.

```
<?php
$temp = explode('***', "A***sentence***with***asterisks");
print_r($temp);
?>
```

Output:

```
Array
(
[0] => A
[1] => sentence
[2] => with
[3] => asterisks
)
```

14. Reset
It reset the array pointer to the first elemet of the array.

```
reset($fred); // Throw away return value
```

$item = reset($fred); // Keep first element of the array in $item

15. End

It moves the pointer to the end of the array.

```
end($fred);
$item = end($fred);
```

16. Unset

The unset function is used to remove particular element from the array. For example consider following PHP document

Example:

```
<?php
$mylist=array(10,20,30,40,50);
unset($mylist[1]);
for($i=0;$i<=4;$i++)
{
        print $mylist[$i];
        print  "";
}
?>
```

2.12.4 Sequential Access to Array Elements

- The array element references start at the first element and every array maintains an internal pointer using which the next element can be easily accessible. This helps to access the array elements in sequential manner.
- The pointer **current** is used to point to the current element in the array. Using the **next** function the next subsequent element can be accessed. Following PHP code illustrates this idea .

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderful!");
$myval=current($mylist);
print("The current value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
print "<br/>";
$myval=next($mylist);
print("The next value of the array is <b>$myval</b>");
?>
```

3. Each

Using **each** function we can iterate through the array elements.

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderfull!");
while($myval=each($mylist)
{
$val=$myval["value"];
print("The current value of the array is <b>$val</b>");
print "<br/>";
}
?>
```

4. foreach

The **foreach** function is used to iterate through all the elements of the loop. The syntax of foreach statement is as follows -

```
foreach($array as $value)
{
    statements to be executed
}
```

The above code can be modified and written as follows -

PHP Program

```
<?php
$mylist=array("Hello","PHP","You","Are","Wonderfull!");
foreach($mylist as $value)
{
    print ("The current value of the array is <b> $value </b>");
    print "<br/>";
}
?>
```

2.12.5 Sorting Arrays

- Sorting is the process in which the element of arrays in some specific order. There are two types of ordering which are followed in sorting – ascending order and descending order. Basically PHP uses **sort** function for sorting the array elements. There are some other functions that are also available for sorting the arrays in desired manner.
- The **sort** function sorts the array based on the values. After applying the sort function this function assigns new keys to the values of the array.Following PHP document illustrates these functions -

PHP Program

```
<?php
$A=array("A" => "Supriya", "B" => "Monika", "C" => "Kumar", "D" => "Archana");
print "<b> Original Array:</b><br/>";
foreach($A as $key => $value)
print ("[$key] => $value <br/>");
sort($A);
print "<br/>";
print "<b> Sorted Array:<b><br/>":
foreach($A as $key => $value)
print ("[$key] => $value <br/>");
?>
```

- The **asort()** function sorts an array by the values. But the original keys are kept.
- The **ksort()** function sorts the array by keys but each value's original key is retained.

Example Programs:

1. Use an array to store student information such as enrolment no, name, semester and percentage. Output the data to a web page using PHP.

Sol:

```
<html>
<head></head>
<body>
<?php
$a=array(array(10,"AAA","II",60),array(20,"BBB","III",80),array(30,"CCC","IV",40));
echo "<table border='1'>";
echo "<tr>";
echo "<td>ENo</td><td>Name</td><td>Sem</td><td>Marks</td>";
echo "<tr/>";
for($i=0;$i<3;$i++)
{
echo "<tr>";
for($j=0;$j<4;$j++)
{
        echo"<td>";
        echo $a[$i]{$j];
        echo"</td>";
}
echo "</tr>";
}
echo "</table>";
?>
</body>
</html>
```

2.23 Strings

PHP String Manipulation

PHP provides a rich set of functions to manipulate strings. In this topic, we will discuss some common functions used by PHP developers to remove spaces from a string, count the number of characters of a string, convert a string to contain upper case or lower case letters, split a string or join strings, get substrings from a string, compare strings, search for a substring in a string, and replace and old substring with a new substring of a string, etc.

trim, ltrim, and rtrim functions

trim, ltrim, and rtrim functions are used to remove space from a string.
-trim(String) removes leading and trailing space from the string.
-ltrim(String) removes leading spaces.
-rtrim(String) removes trailing spaces.

strlen() function

The strlen(String) function is used to count the number of characters of a string.

strtolower, strtoupper, ucfirst, ucwords function

The strtolower, strtoupper, ucfirst, and ucwords functions are used to change change cases of a string:

- strtolower(String) changes a string to lowercase.
- strtoupper(String) changes a string to uppercase.
- ucfirst(String) capitalizes the first character of a string.
- ucwords(String) capitalizes the first character of each word in a string.

strcmp() and strcasecmp() functions
The strcmp(String1,String2) compares String1 with String2. It returns less than zero if String1 is less than String2. If String1 is greater than String2 it return greater than zero. If both strings are equal, it returns 0.This function compares two strings in case-sensitive manner. If you want to compare two strings without case-sensitivity, you can use strcasecmp() instead.

split() and join() functions
The split(Separator_char, String) function is used to split a string in to an array of strings by a separating character.

substr() function
The substr() method has two main forms:
- substr(String, Start) returns a substring from the Start position to the end of the string.
- substr(String,Start,Length) returns a substring from the Start position in which the length of the substring is equal to Length.

strpos() and str_replace() functions
The strpos(String, String_to_find) returns the position of the String_to_find in the String. The str_replace(Old_string,New_string,String) is used to replace the Old_string with the New_string.

Example:
```
<?php
echo strlen("Hello world!")."<br>";
echo str_word_count("Hello world!")."<br>";
echo strrev("Hello world!")."<br>";
echo strpos("Hello world!", "world")."<br>";
echo str_replace("world", "Dolly", "Hello world!")."<br>";
echo substr("Hello World",2)."<br>";
$arr=split("l","Hello")
Echo $arr[0]."<br>";
echo strtoupper("Hello")."<br>";
echo strtolower("HELLO")."<br>";
echo trim(" Hello")."<br>";
?>
```

Output:
```
12
2
!dlrow olleH
6
Hello Dolly!
llo World
He
HELLO
hello
Hello
```
Note that the variable **s** is assigned with the string. The string is given in double quotes. Then using the echo whatever string is stored in the variable **s** is displayed on the console.

2.24 Functions

The functions in PHP are very much similar to the functions in C. Let us discuss it in details -

2.24.1 General Characteristics of Functions

- The syntax of the function definition is as follows -

```
function name_of_function(parameter list)
{
  statements to be executed in function-name
  .....
  .....
}
```

- The function gets executed only after the call to that function. The call to the function can be from anywhere in the PHP code. For example -

PHP Program

```
<?php
function myfun()
{
print "<i> This statement is in myfun()</i>";
}
print "<b> The Function Demo Program</b>";
print "<br/>";
myfun();
?>
```

- The **return** statement is used for returning some value from the function body. Following PHP script shows this idea.

PHP Program

```
<?php
function Addition()
{
$a=10;
$b=20;
$c=$a+$b;
return $c;
}
print "<b> The Function Demo Program with return statement</b>";
print "<br/>";
print "10+20 =".Addition();
?>
```

2.24.2 Parameters

- The parameter that we pass to the function during the call is called the **actual parameter**. These parameter are generally the expressions.
- The parameters that we pass to the function while defining it is called the **formal parameter**. These are generally the variables. It is not necessary that the number of actual parameters should match with the number of formal parameters.
- If there are few actual parameter and more formal parameters then the value of formal parameter is will be some unbounded one. If there are many actual parameters and few formal parameters then the excess of actual parameters will be ignored.

- The default parameter passing technique in PHP is **pass by value.** The parameter passing by value means the values of actual parameters will be copied in the formal parameters. But the values of formal parameters will not be copied to the actual parameters.
- Following PHP script illustrates the functions with parameters

PHP Program

```
<?php
function Addition($a,$b)
{
 $c=$a+$b;
 return $c;
}
print "<b> The Function Demo Program with parameter passing and return statement</b>";
print "<br/>";
$x=10;
$y=20;
print "10+20 =".Addition($x,$y);
?>
```

There are two ways to pass parameters by reference.

3. Add & at the beginning of the name of the actual parameter. For example -

```
<?php
function add_some_extra($string)
{
$string='This is a string";
$str1=&$str;//adding & at the beginning of the name of actual paramater.
print "Before function call: $str<br/>";
add_some_extra($str1);
print "After function call: $str<br/>";
?>
```

4. Add & to actual parameter in the function call. For example -

```
<?php
function add_some_extra(&$string)
{
        $string = "This string is replaced";
}
$str="This is a string";
print "Before function call :$str<br/>";
add_some_extra($str);
print "After function call:$str<br/>";
?>
```

2.25 SCOPE OF THE VARIABLES:

In PHP the scope of the variable is local. That means we can define a variable within a function and by the same name another variable can be defined outside the function. These two variables are considered to be different entities.

Example:

```
<?php
function myfun()
{
        $a=10;
```

```
        print  "The value of a = $a";
}
myfun();
print "<br>";
$a=20;
print "the value of a=$a";
?>
```

Output:

The value of a =10

The value of a =20

We can define the global variable also. Following is the PHP script

```
<?php
$a=10;
function myfun()
{
        global $a;
        $a+=10;
print "The value of a=$a";
}
myfun();
print "<br>";
$a=$a-5;
print  "the value of a =$a";
?>
```

Output

The value of a = 20

The value of a =15

LIFETIME OF THE VARIABLES:

The lifetime of a variable can be defined as the time from which it is used first to the end of function execution. In PHP the static variable is used to remember the previous values.

The lifetime of static variable is the time when the variable if first used and it ends when the script terminates its execution.

For the declaration of the static variable the keyword static variable, the keyword static is used.

```
function myfun()
{
static $count = 0;
count ++;
print "The number of times you have visited this page is $count <br/>";
}
```

2.26 FUNCTION

Functions are group of statements that can perform a task

Defining a Function

The general syntax for a function is:

function *function_name*([*parameter* [, ...]])

```
{
// Statements
}
```

• A definition starts with the word function.

• A name follows, which must start with a letter or underscore, followed by any number of letters, numbers, or underscores.
• The parentheses are required.
• One or more parameters, separated by commas, are optional.

PHP Functions - Return values
Functions can also return the valuese to the point where they have called.
Return statement is used to return the value.
Syntax:

```
function func_name()
{
….
return $variable;
}
echo func_name();
```

Example:

```
 <?php
function add($x,$y)
{
 $total=$x+$y; return $total;
}
echo "1 + 16 = " . add(1,16);
?>
```

Call by Value:
The changes made in the formal arguments will not be reflected back to the actual arguments.
Example:
Swap Numbers PHP Example (Call by value)

```
<?php
$num1=10;
$num2=20;
 echo "Numbers before swapping:<br/>"; echo "Num1=".$num1;
 echo "<br/>Num2=".$num2;
swap($num1,$num2); //call by value
 function swap($n1,$n2)
{
 $temp=$n1;
$n1=$n2;
$n2=$temp;
 echo "<br/><br/>Numbers after swapping:<br/>";
echo "Num1=".$n1;
echo "<br/>Num2=".$n2;
}
?>
```

Call by Reference:
The changes made in the formal arguments will be reflected back to the actual arguments.

Swap Numbers PHP Example (Call by Reference)

```
<?php
$num1=10;
$num2=20;
```

```
 echo "Numbers before swapping:<br/>"; echo "Num1=".$num1;
 echo "<br/>Num2=".$num2;
swap($num1,$num2);
 function swap(&$n1,&$n2) //Call by reference
{
 $temp=$n1;
$n1=$n2;
$n2=$temp;
 echo "<br/><br/>Numbers after swapping:<br/>";
echo "Num1=".$n1;
echo "<br/>Num2=".$n2;
}
?>
```

Recursive Function:

A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result and the end of each iteration.

Example:

```
 <?php
function factorial($number)
 {
if ($number == 0)
 return 1;
return $number * factorial($number - 1);
}
echo factorial(5);
 ?>
```

Output:

120

2.27 FILE HANDLING:

PHP is known as a server side scripting language. Hence file handling functions suc as create, read, write, append are some file related operations that are supported by PHP.

Opening and closing files:

The first step in file handling is opening of the file.

It takes two parameters- The first parameter of this function contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened.

Modes	Description
r	Read only. Starts reading from the beginning of the file.
r+	Read/Write. Starts reading from the beginning of the file
w	Write only. Opens and clears the contents of the file; or creates a new file if it is not created.
w+	Read/ Write. Opens and clears the contents of the file; or creates a new file if it is not created.
a	Append. Opens and writes to the end of the file or creates a new file of it is not created.
a+	Read/Append. Preserves the file content by writing to the end of the file.

For example:
$my_file=’file.txt’;
$file_handle=fopen($my_file,’a’) or die(‘Cannot open file: ’.$my_file);
The fopen function returns TRUE if the required file is opened.

Reading from file:
The fread is the function which is used to read the file. It takes two parameters. The first parameter is the handle to the file and the second parameter is the number of bytes to be read. The filesize is the function which takes the filename as the parameter.

For example: $mystring = fread($file_handle, filesize(“file.txt”));

There is another function named file_get_contents using which the contents of the file can be obtained.
The fgets() function is used to read a single line from the file. For example, following code displays the contents of the file line by line.

```
while(!feof($file_handle))
{
        echo fgets($file_handle).”<br/>”;
}
```

Example Reading a file with fgets

```
<?php
$fh = fopen("testfile.txt", 'r') or
die("File does not exist or you lack permission to open it");
$line = fgets($fh);
fclose($fh);
echo $line;
?>
```

Output:
Line

Example - Reading a file with fread

```
<?php
$fh = fopen("testfile.txt", 'r') or
die("File does not exist or you lack permission to open it");
$text = fread($fh, 3);
fclose($fh);
echo $text;
?>
```

Output:
lin

Exercise: Write a PHP program to read a text file line by line and to display it on screen.

Solution:
Step-1: Create a input file Myfile.txt as follows.
Hello
everybody
how are you?

Step-2: Create a PHP script for reading the input file line by line as follows.

```
<?php
$file = fopen("Myfile.txt","r")
while(!feof($file))
{
echo fgets($file)."<br/>";
}
fclose($file);
?>
```

Writing to a file:

The fwrite is the function which is used to write the contents to the file. It takes two parameters – The first parameter is the handle to the file and the second parameter is the number of bytes to be written. For example -

```
$written_string = fread($file_handle, $my_data);
```

Locking Files:

Multiple PHP scripts can access the same file at a time. But this causes conflict problems. That means there can be the situation in which one script is reading the file and at the same time the other script is writing to that file. Sometimes there can be a situation in which two scripts are trying to write different data to the same file. These are totally undesirable in the file handling technique. The solution to this problem is to lock the file when one script is accessing it. Due to locking of the file, simultaneous access can be avoided. The PHP uses flock function for locking the files.

Syntax of flock is
flock(file, lock, block)
where
file – name of the file which needs to be accessed.
Lock- kind of lock being used. Possible values are:
LOCK-SH – Shared Lock(reader). Allow other processes to access the file
LOCK_EX – Exclusive Lock(writer). Prevent other processes from accessing the file.
LOCK_UN – Release a shared or exclusive lock
LOCK_NB- Avoids blocking other processes while locking block is optional parameter.

Example

```
<?php
$file = fopen("myfile.txt","w+");
flock($file, LOCK_EX)// exclusive lock
fwrite($file, "I am writing this line to file");
flock($file,LOCK_UN);//release lock
fclose($file);
?>
```

Copying Files
We can copy one file into another file using copy() function.
Syntax:
Copy('source file','destination file');

Copying a file

```
<?php // copyfile.php
copy('testfile.txt', 'testfile2.txt') or die("Could not copy file");
echo "File successfully copied to 'testfile2.txt'";
?>
```

If you check your folder again, you'll see that you now have the new file *testfile2.txt* in it. By the way, if you don't want your programs to exit on a failed copy attempt, you could try the alternate syntax.

Moving a File
To move a file, rename it with the rename function.

Example
```
<?php // movefile.php
rename('testfile2.txt', 'testfile2.new');
else echo "File successfully renamed to 'testfile2.new'";
?>
```
You can use the rename function on directories, too. To avoid any warning messages, if the original file doesn't exist, you can call the file_exists function first to check.

Deleting a File
Deleting a file is just a matter of using the unlink function to remove it from the filesystem, as in Example 7-10.
Example 7-10. Deleting a file
```
<?php // deletefile.php
if (!unlink('testfile2.new')) echo "Could not delete file";
else echo "File 'testfile2.new' successfully deleted";
?>
```

2.28 Form Handling

PHP is used for form handling. For that purpose the simple form can be desgined in XHTML and the values of the fields defined on the form can be transmitted to the PHP script using GET and POST methods. For forms that are submitted via "GET " method, we obtain the form via the $_GET array variable. For forms that are submitted via "POST" method we obtain the form via the $_POST array variable.

Exercise:Create HTML form with one text box to get user's name. Also write PHP code to show length of entered name when, the HTML form is submitted.

Solution:

The $_GET Function

- The built-in $_GET function is used to collect values from a form sent with method="get"
- Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's URL) and has limits on the amount of information to send (max. 100 characters)
- This method should not be used when sending passwords or other sensitive information. However, because the variables are displayed in the URL, it is possible to bookmark the page
- The get method is not suitable for large variable values; the value cannot exceed 100 characters

The $_POST Function
- The built-in $_POST function is used to collect values from a form sent with method="post"

- Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send
- However, there is an 8 Mb max size for the POST method, by default (can be changed by setting the post_max_size in the php.ini file)

The $_GET Function Example

Form1.html

```
<html>
<body>
/* form submitted using 'get' method, action specifies next page which is to be loaded when button is clicked*/
<form action="welcome.php" method="get">
textbox is to take user input Name: <input type="text" name="fname" />
Age: <input type="text" name="age" />
Submit button is to submit the value <input type="submit" />
</form>
</body>
</html>
```

welcome.php

```
<html>
<body>
$_GET to receive the data sent from Form1.html Welcome <?php echo $_GET["fname"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
</body>
</html>
```

The $_ POST Function Example

form1.html

```
<html>
<body>
/* form submitted using 'post' method, action specifies next page which is to be loaded when button is clicked */ <form action="welcome1.php" method="post">
textbox is to take user input Name: <input type="text" name="fname" /> Age: <input type="text" name="age" />
Submit button is to submit the value to next page <input type="submit" />
</form>
</body>
</html>
```

welcome1.php

```
<html>
<body>
$_GET to receive the data sent from form1.html Welcome <?php echo $_POST["fname"]; ?>.<br />
You are <?php echo $_POST["age"]; ?> years old!
</body>
</html>
```

Step 1:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML-PHP Demo</title>
</head>
<body>
<form method= “post” action= “http://localhost/getdata.php”>
Name:<input type= “text” name= “myname” size=”20” />
<br/>
<input type = “submit” name=”submit” value= “Submit” />
</form>
</body>
</html>
```

Step 2:

The PHP script to display the length of the submitted name is as written below:

```
<?php
print “The name is”;
print $_POST[‘myname’];
$len=strlen($_POST[‘myname’]);
print $len;
?>
```

Exercise: Create HTML form to enter one number. Write PHP code to display the message about number is odd or even.

Solution

Step1: The HTML form for accepting number is created as below-

```
<!DOCTYPE html>
<head><title>HTML-PHP DEMO</title>
</head>
<body>
        <form method=”post” action =”http://localhost/getdata.php”>
         Enter Number:<input type=”text” name=”mynum” size=”5”/>
        <br/>
        <input type=”submit” name=”submit” value=”submit”/>
        </form>
</body>
</html>
```

Step 2: The PHP script deciding whether the number is odd or even is given below-

```
<?php
print “The number is:”;
print $_POST[“mynum”];
$a=$_POST[“mynum”];
if($a%2==1)
        print”<br/> The number is odd”;
else
        print”<br/> The number is even”;
?>
```

EXERCISE: Create a form containing information sl.no,title of the book,publishers,quantity,price read the data from the form and display it using PHP script.

SOLUTION:

Step 1: Create an HTML page for inputting the data.

HTML Document[input.html]

```
<!doctype html>
<html>
<head>
<title>Book Order Form</title>
</head>
<body>
<h3>Enter the Book Data</h3>
<form method=”post” action=”http://localhost/php-examples/formdemo.php”>
<table>
<tr>
<td> Sr.No</td>
<td><input type=”text” name=”SLNo”></td>
</tr>
<tr>
<td>Book name</td>
<td><input type=”text” name=”BName”></td>
</tr>
<tr>
<td>Publisher</td>
<td><input type=”text” name=”PUBname”></td>
</tr>
<tr>
<td>Price</td>
<td><input type=”text” name=”Price”></td>
</tr>
<tr>
<td>Quantity</td>
<td><input type= “text” name=”Qty”></td>
</tr>
<tr>
<td><input type= “submit” name=”submit”></td>
<td><input type= “submit” name=”Clear”></td>
</tr>
</table>
</form>
</body>
</html>
```

Step 2: Create a PHP script which will read out the data entered by the user using HTML form . The code is as follows :

formdemo.php:

```
<html>
<head>
<title>Book Information</title>
</head>
<body>
<?php
$Bname=$_POST[‘BName’];
```

```
$PUBName=$_POST[‘PUBName’];
$Price=$_POST[‘Price’];
$Qty=$_POST[‘Qty’];
?>
<center><h3>Book Data</h3>
<table border=1>
<tr>
<th>Book Name</th>
<th>Publisher</th>
<th>Price</th>
<th>Quantity</th>
</tr>
<tr>
<td><?php print (“$Bname”);?></td>
<td><?php print (“PUBName”);?></td>
<td><?php print (“$Price”);?></td>
<td><?php print (“$Qty”);?></td>
</tr>
</table>
</center>
</body>
</html>
```

Step 3: Open some suitable web browser and enter the address for the HTML file which you have created in step 1. Now click on the submit button and the PHP file will be invoked. The output will then be as follows.

Exercise : Write a PHP program to accept a positive integer ‘N’ through a HTML form and to display the sum of all the numbers from 1 to N

```
<!DOCTYPE html>
<html>
<head>
<title>PHP Demo</title>
</head>
<body>
<form method= “post” action= “getdata.php”>
<input type= “text” name= “num”  />
<input type = “submit”  value= “Submit” />
</form>
</body>
</html>
```

```
<?php

//getting values from HTML form

$n=intval($_POST[‘num’]);

$sum=0;

for($i=1;$i<=$n;$i++)

        $sum=$sum+i;

echo “The sum of all numbers from 1 to “.$n.”is”.$sum;?>
```

2.18 PHP and MySQL:

Benefits of using PHP and MySQL:

PHP is a server side scripting language and it has an ability to create dynamic pages with customized features. Using PHP-MySQL user friendly and interactive web sites can be created. Both PHP and MySQL are open-source technologies that work hand-in-hand to create rich internet applications. The purchased code provides you the encrypted source code to prevent replication or modification, whereas open-source programs encourage users to utilize, scrutinize and customize the code.

Due to the availability of these technologies as free of cost, the cost effective web solutions can be created. PHP-MySQL are stable technologies and have cross platform compatibility. Hence the web application developed using these technologies becomes portable. Since HTML can be embedded within the PHP, there is no need to write seperate code for web scripting. Open-source coding has been checked and doubled checked by thousands or even millions of people around the world. Hence one can built the reliable web application using these technologies.

The PHP has got the support from several content management programs such sa wrodpress, Joomla, Drupal and so on. It has got a strong support for developing e-commerce applications using the technologies such as Ecommerce, Drupal and so on. The most popular web sites being developed using PHP and MySQL technologies are:

4. Facebook
5. WordPress
6. Wikipedia

Structured Query Language(SQL)

MySQL is a open source database product and can be downloaded from the web site Http://dev.mysql.com/downloads/mysql . MySQL is a kind of database in which the records are stored in an entity called tables. In the tables the data is arranged in the rows and columns. We can query a database to retrieve particular information. Query is a request or a question for the database. There is a common practice of making use of Structured Query Language(SQL).

Connecting PHP to MySQL:

The PHP function mysql_connect connects to the MySQL Server. There are three parameters that can be passed to this function. For example-

mysql_connect("localhost", "root", "mypassword") or die(mysql_error());

where

localhost- Local host on which the MySQL is running

root- Root

mypassword- Password

The database can be selected by using the command mysql_selecet_db.

For example:

mysql_select_db("test") will select the database named test.

Requesting MySQL Operations:

9. Creating Database:

We can create a database using the function mysql_query. The mysql_error() function is used to get the error messages if any command gets failed.

mysql_query function in php is used to pass a sql query to myql database.

Syntax:

mysql_query(string query[, resource link_identifier])

This function returns the query handle for select queries, TRUE/FALSE for other queries, or FALSE on failure.

Example

mysql_query("CREATE DATABASE mydb", $con)
The mysql_connect() function open a connection to a MySQL Server.
Syntax:
mysql_connect(string server, string username, string password)
Returns a MySQL link identifier on success, or FALSE on failure.
Example:
$conn=mysql_connect("localhost", "root", "password");
The mysql_close() function is used to close the database connection.
Syntax:
mysql_close(Connection)
PHP program for Creating the database:

```
<?php
$conn=mysql_connect("localhost", "root", "password"); //Make a sql connection
if(!$conn)
{
die('error in connection'.mysql_error());
}
if(mysql_query("CREATE DATABASE mydb", $conn)) //Create a database
{
print "Database Created";
}
else
{
print "Error Creating database:".mysql_error();
}
mysql_close($conn); //closing the database
?>
```

10. Selecting the database:
The database can be selected using the function **mysql_select_db().**
Syntax:
mysql_select_db(string database_name [, resource link_identifier])
where mysql_select_db() attempts to select existing database on the server associated with the specified link identifier. It returns TRUE on success, or FALSE on failure.
For example-

```
<?php
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
mysql_close($conn); //closing the database
?>
```

11. Counting Number of Rows:
The numbe rof rows present in the table can be obtained using mysql_num_rows functions.
Syntax:
int mysql_num_rows(resource $result)
This returns number of rows in result on success, or NULL on error.
Example:

```
<?php
```

```
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$num_rows=mysql_num_rows($result);
//Print number of rows
echo ""Total number of rows are $num_rows";
mysql_close($conn); //closing the database
?>
```

12. Counting number of fields:

The mysql_num_fields() is used to get number of fields of the table.

Syntax:

mysql_num_fields(resource_name)

It returns the number of fields present in the resource and false on failure.

Example:

```
<?php
//Make a MySQL Connection
$conn=mysql_connect("localhost:3306/mydb", "root", "mypassword");
if(!$conn)
{die('error in connection'.mysql_error());
}
//Select a database
mysql_select_db("mydb", $conn);
$result=mysql_query("Select id, name from my_table where id='1' ");
echo mysql_num_fields($result);
mysql_close($conn); //closing the database
?>
```

13. Creating Table:

Before creating the table a database must be created and within which the table can be created. Note that before creating a table, desired database must be selected.

Example:

```
<?php
$conn=mysql_connect("localhost", "root", "password"); //Make a sql connection
if(!$conn)
{die('error in connection'.mysql_error());
}
if(mysql_query("CREATE DATABASE mydb", $conn)) //Create a database
{print "Database Created";
}
else
{print "Error Creating database:".mysql_error();
}
mysql_select_db("mydb",$conn); // Before creating a table, database must be selected.
$query="CREATE TABLE my_table (id INT(4), name VARCHAR(20))";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

14. Inserting Data in table:

For inserting a data into the table we use the INSERT query. For Example

```
$query= "INSERT INTO my_table(id, name) VALUES (1, 'SHILPA')";
mysql_query($query, $conn); // Execution of Query
```

Here is a PHP script in which insert query is used to insert two records in the table

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$query= "INSERT INTO my_table(id, name) VALUES (1, 'SHILPA')";
mysql_query($query, $conn);
$query= "INSERT INTO my_table(id, name) VALUES (2, 'MONIKA')";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

Sometimes values that can be inserted in the table can be obtained from someother script and these values might be present in the variables. Insertion of such data can be done using $_POST variables. It is as shown below-

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$query= "INSERT INTO my_table(id, name) VALUES ('$_POST[MyId]', '$_POST[MyName]')";
mysql_query($query, $conn);
mysql_close($conn); //closing the database
?>
```

15. Displaying or Retrieving Records:

For displaying the records present in the database table, we use SELECT query. For Example

```
// Execution of Query for displaying the data
$result=mysql_query("SELECT * FROM my_table");
```

The above execution returns a result handle. Then the mysql_fetch_array() is used to retrieve a row of data as an array from a MySQL result handle.

Syntax:

mysql_fetch_array(result, result_type)

PHP Script for Displaying records:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "password");
if(!$conn)
{
die('error in connection'.mysql_error());
}
```

```
mysql_select_db(“mydb”,$conn);
//Execution of Query for dsiplaying the data
$result=mysql_query(“SELECT * FROM my_table”);
while($row=mysql_fetch_array($result))
{
echo $row[‘id’]. “” .$row[‘name’]; // Each record will be displayed
echo “<br/>”;
}
mysql_close($conn); //closing the database
?>
```

16. Finding the number of affected rows:

The mysql_affected_rows query is used to get number of affected rows in previous MySQL operation such as INSERT, DELETE, UPDATE queries.

Syntax:

mysql_affected_rows(connection)

Example:

```
<?php
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “password”);
if(!$conn)
{
die(‘error in connection’.mysql_error());
}
mysql_select_db(“mydb”,$conn);
$query= “INSERT INTO my_table(id, name) VALUES (1, ‘SHILPA’)”;
mysql_query($query, $conn);
$query= “INSERT INTO my_table(id, name) VALUES (2, ‘MONIKA’)”;
mysql_query($query, $conn);
echo “Number of rows affected are:”.mysql_affected_rows();
mysql_close($conn); //closing the database
?>
```

MySQL Functions:

mysql_connect():

This function opens a link to a MySQL server on the specified host (in this case it's localhost) along with a username (root) and password (q1w2e3r4/). The result of the connection is stored in the variable $db.

mysql_select_db():

This tells PHP that any queries we make are against the mydb database.

mysql_query():

Using the database connection identifier, it sends a line of SQL to the MySQL server to be processed. The results that are returned are stored in the variable $result.

mysql_result():

This is used to display the values of fields from our query. Using $result, we go to the first row, which is numbered 0, and display the value of the specified fields.

mysql_result($result,0,"position")):

This should be treated as a string and printed.

2.19 COOKIES

Introduction to cookies:
Cookie is a small file that server embeds in the user's machine. Cookies are used to identify the users. A cookie consists of a name and a textual value. A cookie is created by some software system on the server. In every HTTP communication between browser and server, a header is included. The header stores the information about the message. The header part of http contains the cookies. There can be one or more cookies in browsers and server communications.

PHP Support for Cookies:
PHP can be used to create and retrieve the cookies. The cookie can be set in PHP using the functions called setcookie().
The syntax for setcookie is -
setcookie(name, value,expire_period, path, domain)

Example PHP program to set cookie:

```
<?php
$cookie_period=time()+60*60*24*30;
setcookie("Myname", "Monika", $cookie_period);
?>
```

Note that you have got the blank screen it indicates that the cookie is set . In above PHP document we have set the PHP script for one month.

Example: [CookieRead.php]

```
<html>
<head><title>Reading cookies</title>
<body>
<?php
if(isset($_COOKIE["Myname"]))
        echo "<h3>Welcome".$_COOKIE["Myname"]. "!!!</h3>";
else
        echo "Welcome guest!";
?>
</body>
</html>
```

The isset function is used for checking whether or not the cookie is set. Then using the $_COOKIE the value of the cookie can be retrieved.

Exercise: Write a PHP Script to create a new database table with 4 fields of your choice and perform following database operations. i)insert ii)update iii)Delete
Solution:
We will create a table in the database test. The name of the table is mytable. Then we will insert the record into the table using the INSERT command, update particular field of the record using the command UPDATE and delete the record using the command DELETE.

PHP Document[DBDemo.php]

```
<?php
// Make a MySQL Connection
mysql_connect("localhost","root","mypassword") or die(mysql_error());
mysql_select_db("test") or die(mysql_error());
echo "Connected to database";
mysql_query("CREATE TABLE mytable(id INT NOT NULL AUTO _INCREMENT,PRIMARY
KEY(id), name VARCHAR(30),phone INT,emailId VARCHAR(30))") or die(mysql_error());
```

```
print"<br/>";
echo "Table Created";
//Insert a row of information into tabele "example"
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('abcd','1111','abc@gmail.com')") or die (mysql_error());
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('xyz','2222','xyz@gmail.com')") or die (mysql_error());
mysql_query("INSERT INTO mytable(name,phone,emailId)
VALUES('Kumar,'3333','pqr@gmail.com')") or die (mysql_error());
print"<br/>";
echo "Data Inserted";
$result=mysql_query("SELECT * from mytable")
or die(mysql_error());
print"<br/>";
print"<b> User Databse</b>";
echo"<table border='1'>";
echo"<tr><th>ID</th><th>Name</th> <th> Phone</th><th> Email_ID</th></tr>";
while($row=mysql_fetch_array($result))
{
//Print out the contents of each row into a table
echo"<tr><td>";
echo $row['id'];
echo"</td><td>;
echo $row['name'];
echo"</td><td>";
echo $row['phone'];
echo"</td><td>";
echo $row['emailId'];
echo"</td><tr>";
}
echo "<table>";
$result=mysql_query("UPDATE mytable SET phone="5555" where phone="2222")
or die(mysql_error());
print"<br/>";
echo "Data Updated";
$result=mysql_query("SELECT * from mytable")
or die(mysql_error());
print"<br/>";
print"<b> User Datbase</b>";
echo"<table border='1'>";
echo"<tr><th>ID</th><th>Name</th><th>Phone</th><th>Email-ID</th></tr>";
while($row=mysql_fetch_array($result))
{
// Print out the contentsof each row into a table
echo"<tr><td>";
echo $row['id'];
echo"</td><td>;
echo $row['name'];
echo"</td><td>";
echo $row['phone'];
echo"</td><td>";
echo $row['emailId'];
echo"</td><tr>";
```

```
}
echo “<table>”;
result=mysql_query(“DELETE from mytable  where phone=”3333”)
or die(mysql_error());
print”<br/>”;
echo “Data Deleted”;
$result=mysql_query(“SELECT * from mytable”)
or die(mysql_error());
print”<br/>”;
print”<b> User Datbase</b>”;
echo”<table border=’1’>”;
echo”<tr><th>ID</th><th>Name</th><th>Phone</th><th>Email-ID</th></tr>”;
while($row=mysql_fetch_array($result))
{
// Print out the contentsof each row into a table
echo”<tr><td>”;
echo $row[‘id’];
echo”</td><td>;
echo $row[‘name’];
echo”</td><td>”;
echo $row[‘phone’];
echo”</td><td>”;
echo $row[‘emailId’];
echo”</td><tr>”;
}
echo “<table>”;
```

Exercise: CREATE a HTML form “result.html” with a text box and a submit button to accept registration number of the student. Write a “result.php” code to check the status of the result from the table to display whether the student has “PASS” or “FAIL” status. Assume that the Mysql database “my_db” has the table “result_table” with two columns REG_NO and STATUS

Step 1:
Create a database named my_db. Create a table result_table for this database and insert the values in this table. The table is created as follows:

Step 2:
Create an HTML form to accept the registration number, the HTML document is as follows-
result.html

```
<!DOCTYPE html>
<html>
<head>
<title>STUDENT RESULT</title>
</head>
<body>
<form name = “myform” method= “post” action= “http://localhost/php-examples/result.php”>
<input type= “text” name= “reg_no” />
<input type = “submit” value= “Submit” />
</form>
</body>
</html>
```

Step 3: Create a PHP Script to accept the registration number. This PHP script will connect to MYSQL database and the status (PASS or FAIL) of the corresponding registration number will be displayed.

Result.php

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$reg_no = intval($_POST['reg_no']);
$result=mysql_query("SELECT REG_NO, STATUS FROM result_table where REG_NO=$reg_no");
while($row=mysql_fetch_array($result))
{
echo $row['REG_NO']. "is". $row['STATUS'];
echo "<br/>";
}
mysql_close($conn);
?>
```

Step 4: Load the HTML form created in Step 2 and click the submit button by entering some registration number

Exercise : Write a PHP program to delete a record from the result_table:

Solution:

```
<?php
//Make a SQL Connection
$conn=mysql_connect("localhost", "root", "");
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$reg_no = intval($_POST['reg_no']);
$result=mysql_query("DELETE FROM result_table where REG_NO=$reg_no");
while($row=mysql_fetch_array($result))
{
echo $row['REG_NO']. "is". $row['STATUS'];
echo "<br/>";
}
mysql_close($conn);
?>
```

Exercise: Write a user defined function 'CalculateInterest' using PHP to find the simple interest to be paid for a loan amount. Read the loan amount, the number of years and the rate of interest from a database table called LOANDETAILS having three fields AMT, YEARS, and RATE and Calculate the interest using the user defined function.

Solution:

Step 1:

Create a database table named LOANDETAILS having three fields AMT, YEARS and RATE. Insert the values in it. The sample table will be as follows -

AMT	YEARS	RATE
10000	5	6.9

Step 2:
The PHP code for calculating the simple interest the above values in a function will be as follows:
Interest.php

```
<?php
function CalculateInterest()
{
<?php
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “”);
if(!$conn)
{
die(‘error in connection’.mysql_error());
}
mysql_select_db(“mydb”,$conn);
$result=mysql_query(“SELECT * FROM LOANDETAILS”);
$row=mysql_fetch_array($result);
$amount=$row[‘AMT’];
$rate=$row[‘RATE’];
$years=$row[‘YEARS’];
$interest=($amount*$rate*$years)/100;
mysql_close($conn);
return $interest;
}
print “Simple Interest =”.CalculateInterest();
}
```

Exercise: Consider a table PRODUCT_DETAILS with PID, PNAME and UNIT_PRICE. Write HTML form to accept PID and QUANTITY_REQUIRED from the user and dsiplay the total amount to be paid in another textbox. Get the unit price for given PID from a database table.
Solution:
Bill.php

```
<!DOCTYPE html>
<html>
<head>
<title>PHP Demo</title>
</head>
<body>
<form method= “post”>
<input type= “text” name= “PRODUCT_ID” />
<input type= “text” name= “QUANTITY” />
<input type = “submit” value= “Submit” />
<?php
//getting values from HTML Form
$id=intval($POST[‘PRODUCT_ID’]);
$qty=intval($POST[‘QUANTITY’]);
//Make a SQL Connection
$conn=mysql_connect(“localhost”, “root”, “”);
```

```
if(!$conn)
{
die('error in connection'.mysql_error());
}
mysql_select_db("mydb",$conn);
$result=mysql_query("SELECT UNIT_PRICE FROM product_table where PID=$id");
$row=mysql_fetch_array($result);
$price=$row['UNIT_PRICE'];
$total=$price*$qty;
mysql_close($conn);
?>
<input type= "text" name= "amount" value="<?php echo @$total;?>"/>
</form>
</body>
</html>
```

Exercise:Write a Script to insert and delete record from database.

Index.html:

```
<!DOCTYPE html>
<body bgcolor="#bbffa0">
<center>
<br><h1>Database Application<h1>
<h3>You can Insert, delete and search your data</h3>
<a href="insert.php">Insert</a>
<br><br>
<a href="delete.php">Delete</a>
<br><br>
</center>
<body>
```

Insert.php:

```
<!doctype html>
<body bgcolor="#aacb00">
<form method="POST" action="insert_process.php">
<center>
<h1>Insert your Details here</h1>
<h2>
Name:
<br><input type="text" name="name">
<br>
<br>
Register Number:
<br><input type="text" name="reg">
<br><br>
Age:
<br> <input type="text" name="age">
<br><br><br>
<input type="submit"  value="Insert" name="insert">
<br><br>
</h2>
</center>
</form>
```

```
</body>
```
Insertprocess.php
```
</html>
<?php
// Create connection
$conn = new mysqli("localhost", "root", "", "itessentials");
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$name=$_POST['name'];
$regno=$_POST['reg'];
$age=$_POST['age'];
$sql = "INSERT INTO student (name, reg_no, age)VALUES ('$name','$regno','$age')";
if ($conn->query($sql) === TRUE) {
   echo "Inserted successfully";
}
?>
<html>
<br>
<a href="database.html">Goto Home</a>
</html>
```
Delete.php:
```
<!DOCTYPE html>
<body bgcolor="#bbccf0">
<br>
<center>
<h1>Delete Your Data</h1><h2>
<form method="POST" action="del_process.php">
Enter the Register no :<input type="text" name="reg">
<br>
<br>
<input type="submit" value="DeleteData"></h2>
</center>
</form>
</body>
</html>
```
Deleteprocess.php:
```
<?php
$conn = new mysqli("localhost", "root", "", "itessentials");
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$regno=$_POST["reg"];
$sql = "DELETE FROM student WHERE reg_no=$regno";
if ($conn->query($sql) === TRUE) {
   echo "Record Deleted successfully";
}
?>
<html><br>
<a href="database.html">Goto Home</a>
</html>
```

Exercise:Write a database application to insert and search the data.

```
<!doctype html>
<html lang="en">
<head>
		<title>Simple Information Reterival System</title>
</head>
<body background="lighthouse.jpg">
	<h1><center>Simple Information Reterival System</center></h1>
	<center><ul>
		<h2><li><a href="create.php"><strong>Create</strong></a> - add a user</h2></li>
		<h2><li><a href="search.php"><strong>Read</strong></a> - find a user</h2></li>
	</ul></center>

</body>
</html>
```

Create.php:

```
<html>
<body bgcolor="cyan">
<h2>Add a user</h2>

<form method="post" action="insert.php">
	<label for="firstname">First Name</label><br>
	<input type="text" name="firstname" id="firstname" required><br>
	<label for="lastname">Last Name</label><br>
	<input type="text" name="lastname" id="lastname" required><br>
	<label for="dob">Date of Birth</label><br>
	<input type="text" name="dob" id="dob" required><br>
      <label for="email">Email Address</label><br>
	<input type="email" name="email" id="email" required><br>
	<label for="age">Age</label><br>
	<input type="number" name="age" id="age" required><br>
	<label for="location">Location</label><br>
	<input type="text" name="location" id="location" required><br><br>
	<input type="submit" name="submit" value="Submit">
</form>
*All feilds are mandate.<br>
<a href="index.php">Back to home</a>
</body>
</html>
```

Insert.php:

```
<!dhtml>
<body bgcolor="pink">
<?php
$conn = new mysqli("localhost","root","","itessentials");

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}$firstName=$_POST['firstname'];
$lastName=$_POST['lastname'];
$dob=$_POST['dob'];
$email=$_POST['email'];
```

```
$age=$_POST['age'];
$location=$_POST['location'];
$query="INSERT INTO user_details (first_name, last_name,dob, email, age,location) VALUES
(?,
?,?,?,?,?)";
$stmt = mysqli_prepare($conn,$query);
mysqli_stmt_bind_param($stmt,"sssiss",$firstName,$lastName,$dob,
$email,$age,$location); mysqli_stmt_execute($stmt);

echo "<h2>Thank You ".$firstName ."You are Registered Successfully</h2>";
echo "<a href='index.php'>Back to home</a>";
mysqli_close($conn); ?>
</body>
```

Search.php:

```
</html>
<h1>Search a User</h1>
<form method="post" action="retrieve.php">
        <table><tr style="width=100%">
        <td>Enter search query  <input type="text" name="searchquery" id="se"><br></td></tr>
        </tr></table>
        <table><tr>
        <td> <label for="search">Search by</label><br></td>
        <td>    <input    type="radio"    name="searchby"    value="first_name"    checked>
                First Name<br></td>
        <td> <input type="radio" name="searchby" value="last_name" > Last Name<br></td>
        <td> <input type="radio" name="searchby" value="email"> Email<br></td>
        <td> <input type="radio" name="searchby" value="location"> Location<br></td>
        </tr>    </table>
        <input type="submit" name="submit" value="Submit">
        </form>
```

Retrieve.php:

```
<?php
$host="localhost";
$username   = "root";
$password   = "";
$dbname     = "itessentials";
$conn = new mysqli($host, $username, $password,$dbname);
$searchby=$_POST['searchby'];
$searchquery=$_POST['searchquery'];
$query="SELECT * FROM user_details WHERE ".$searchby ."= '". $searchquery."'";
$result=mysqli_query($conn,$query);
while($row=mysqli_fetch_array($result,MYSQLI_ASSOC)){
        echo "<h1>Welcome ".$row['first_name']."
        ".$row['last_name']."</h1>"; echo "<h3>Age ".$row['age']."</h1>";
        echo "<h3>Registered email
        ".$row['email']."</h1>"; echo "<h3>
        Location".$row['location']."</h1>";
        }// Check connection
if ($conn->connect_error) {    die("Connection failed: " . $conn->connect_error);
}
mysqli_close($conn);
 ?>
```

UNIT III: Database Essentials

Database management - Database terms - MySQL - commands – Data types – Indexes – Functions – Accessing MySQL using PHP.

Database Management System (DBMS)

1. Introduction to DBMS

- A **Database Management System (DBMS)** is software that allows users to create, manage, and manipulate databases efficiently.
- Examples: MySQL, Oracle, PostgreSQL, MongoDB, SQL Server.

2. Types of DBMS

- **Hierarchical DBMS**: Data is organized in a tree-like structure (e.g., IBM IMS).
- **Network DBMS**: Uses graph structure with multiple relationships (e.g., IDMS).
- **Relational DBMS (RDBMS)**: Stores data in tables with relationships (e.g., MySQL, Oracle).
- **Object-Oriented DBMS (OODBMS)**: Stores data as objects (e.g., db4o, ObjectDB).
- **NoSQL DBMS**: Stores unstructured or semi-structured data (e.g., MongoDB, Cassandra).

3. Functions of DBMS

- **Data Storage & Retrieval**: Efficiently stores and retrieves data.
- **Data Security**: Ensures unauthorized users cannot access data.
- **Data Integrity**: Maintains accuracy and consistency of data.
- **Transaction Management**: Ensures ACID (Atomicity, Consistency, Isolation, Durability) properties.
- **Concurrency Control**: Manages multiple users accessing data simultaneously.
- **Backup & Recovery**: Prevents data loss.

4. Advantages of DBMS

- **Data Redundancy Reduction**: Eliminates duplicate data.
- **Data Consistency**: Ensures uniform data across applications.
- **Improved Data Security**: Provides user authentication and access control.
- **Data Independence**: Allows changes in database structure without affecting applications.
- **Efficient Query Processing**: Enables quick data retrieval using SQL.

5. Components of DBMS

- **Hardware**: Physical devices (servers, storage).
- **Software**: The DBMS itself.
- **Data**: Raw facts and figures stored in the database.
- **Users**:
  - **Database Administrator (DBA)** – Manages the database.
  - **Application Programmers** – Develop database applications.
  - **End Users** – Interact with the database through applications.
- **Procedures**: Rules and guidelines for database operations.

6. DBMS vs. File System

Feature	DBMS	File System

Feature	DBMS	File System
Data Redundancy	Reduced	High
Security	High	Low
Data Integrity	Maintained	Not ensured
Concurrency	Supports multiple users	Limited support
Query Language	SQL	No standard query language

7. Database Models

- **Hierarchical Model**: Parent-child relationship.
- **Network Model**: Many-to-many relationships.
- **Relational Model**: Uses tables (most widely used).
- **Object-Oriented Model**: Stores objects.

8. ACID Properties

Ensures reliable transactions:

- **Atomicity**: Complete all operations or none.
- **Consistency**: Maintains valid database state.
- **Isolation**: Transactions do not interfere.
- **Durability**: Ensures data persistence after a transaction.

9. SQL (Structured Query Language)

- **DDL (Data Definition Language)**: CREATE, ALTER, DROP
- **DML (Data Manipulation Language)**: SELECT, INSERT, UPDATE, DELETE
- **DCL (Data Control Language)**: GRANT, REVOKE
- **TCL (Transaction Control Language)**: COMMIT, ROLLBACK, SAVEPOINT

10. Types of Database Users

- **End Users**: Interact via applications.
- **Database Administrator (DBA)**: Manages the database.
- **Developers**: Build applications using the database.

Database Terms

1. Database

- A structured collection of data that can be accessed, managed, and updated efficiently.
- Example: A student database storing names, roll numbers, and marks.

2. DBMS (Database Management System)

- Software that helps in creating, managing, and querying databases.
- Examples: MySQL, PostgreSQL, Oracle, MongoDB.

3. RDBMS (Relational Database Management System)

- A type of DBMS that organizes data in tables with relationships.
- Examples: MySQL, SQL Server, PostgreSQL.

4. SQL (Structured Query Language)

- A language used to interact with relational databases.
- Commands: SELECT, INSERT, UPDATE, DELETE.

5. Table

- A collection of rows and columns representing data in an organized way.
- Example: A **Customers** table with Name, Email, and Phone Number columns.

6. Row (Record, Tuple)

- A single entry in a table, representing a complete set of related data.
- Example: A single student's details in a Student table.

7. Column (Field, Attribute)

- A specific category of data stored in a table.
- Example: "Email" in a Users table.

8. Primary Key (PK)

- A unique identifier for each row in a table.
- Example: Student ID in a Student table.

9. Foreign Key (FK)

- A field in one table that refers to the primary key in another table, creating a relationship.
- Example: "CustomerID" in an Orders table referring to the Customers table.

10. Index

- A database structure that speeds up search queries.
- Works like an index in a book.

11. Normalization

- Process of organizing data to reduce redundancy and improve efficiency.
- Normal Forms (1NF, 2NF, 3NF, BCNF).

12. ACID Properties

- Ensures reliable transactions in a database:
    - **Atomicity**: All or nothing.
    - **Consistency**: Valid database state.
    - **Isolation**: Transactions do not interfere.
    - **Durability**: Data is permanent.

13. Transaction

- A sequence of operations treated as a single unit.
- Commands: COMMIT, ROLLBACK.

14. Joins

- Used to combine data from multiple tables.

- **INNER JOIN**: Only matching records.
- **LEFT JOIN**: All records from the left table + matching from the right.
- **RIGHT JOIN**: All records from the right table + matching from the left.
- **FULL JOIN**: All records from both tables.

15. Views

- A virtual table based on a query.
- Used for security and simplified data access.

16. Stored Procedure

- A set of SQL statements stored for reuse.
- Improves performance and security.

17. NoSQL Database

- Non-relational databases designed for large-scale, unstructured data.
- Types: Document-based (MongoDB), Key-Value Store (Redis), Graph-based (Neo4j).

18. Data Warehouse

- A large storage system used for analytics and reporting.
- Example: Google BigQuery, Amazon Redshift.

19. Big Data

- Large datasets that require special tools to process, like Hadoop and Spark.

20. Cloud Database

- A database hosted on cloud services like AWS, Google Cloud, and Azure.

MySQL Commands with Examples

1. Data Definition Language (DDL) – Defines Database Structure

- **Create a Database:**

```
CREATE DATABASE school;
```

- **Use a Database:**

```
USE school;
```

- **Create a Table:**

```
CREATE TABLE students (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50) NOT NULL,
   age INT,
   grade VARCHAR(10));
```

- **Alter Table – Add a Column:**

```
ALTER TABLE students ADD email VARCHAR(100);
```

- **Drop a Table (Deletes Structure & Data):**

```
DROP TABLE students;
```

- **Truncate a Table (Deletes Data but Keeps Structure):**

```
TRUNCATE TABLE students;
```

2. Data Manipulation Language (DML) – Manages Data

- **Insert Data into Table:**

```
INSERT INTO students (name, age, grade) VALUES ('John Doe', 15, '10th');
```

- **Select Data from Table:**

```
SELECT * FROM students;
```

- **Update Data in a Table:**

```
UPDATE students SET age = 16 WHERE name = 'John Doe';
```

- **Delete Data from a Table:**

```
DELETE FROM students WHERE name = 'John Doe';
```

3. Data Query Language (DQL) – Retrieves Data

- **Retrieve Specific Columns:**

```
SELECT name, age FROM students;
```

- **Filter Data with WHERE Clause:**

```
SELECT * FROM students WHERE age > 14;
```

- **Sort Data using ORDER BY:**

```
SELECT * FROM students ORDER BY age DESC;
```

- **Find Unique Values using DISTINCT:**

```
SELECT DISTINCT grade FROM students;
```

- **Use LIKE for Pattern Matching:**

```
SELECT * FROM students WHERE name LIKE 'J%';  -- Names starting with J
```

- **Use BETWEEN for a Range of Values:**

```
SELECT * FROM students WHERE age BETWEEN 14 AND 18;
```

4. Data Control Language (DCL) – Controls Access

- **Grant User Privileges:**

```
GRANT SELECT, INSERT ON school.* TO 'user1'@'localhost';
```

- **Revoke User Privileges:**

```
REVOKE INSERT ON school.* FROM 'user1'@'localhost';
```

5. Transaction Control Language (TCL) – Manages Transactions

- **Start a Transaction:**

```
START TRANSACTION;
```

- **Insert Data Within a Transaction**

```
INSERT INTO students (name, age, grade) VALUES ('Alice', 17, '12th');
```

- **Rollback (Undo Changes if Needed):**

```
ROLLBACK;
```

- **Commit (Save Changes Permanently):**

```
COMMIT;
```

6. Joins – Combining Data from Multiple Tables

- **Create Two Tables for Joins:**

```
CREATE TABLE teachers (
   id INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50),
   subject VARCHAR(50)
);

CREATE TABLE courses (
   course_id INT PRIMARY KEY AUTO_INCREMENT,
   course_name VARCHAR(50),
   teacher_id INT,
   FOREIGN KEY (teacher_id) REFERENCES teachers(id)
);
```

- **INNER JOIN (Match Records in Both Tables:**

```
SELECT teachers.name, courses.course_name
FROM teachers
INNER JOIN courses ON teachers.id = courses.teacher_id;
```

- **LEFT JOIN (All Records from Left Table + Matches from Right Table):**

```
SELECT teachers.name, courses.course_name
FROM teachers
LEFT JOIN courses ON teachers.id = courses.teacher_id;
```

- **RIGHT JOIN (All Records from Right Table + Matches from Left Table):**

```
SELECT teachers.name, courses.course_name
FROM teachers
RIGHT JOIN courses ON teachers.id = courses.teacher_id;
```

7. Constraints – Data Integrity Rules

- **Create Table with Constraints:**

```
CREATE TABLE employees (
   emp_id INT PRIMARY KEY AUTO_INCREMENT,
   emp_name VARCHAR(100) NOT NULL,
   emp_salary DECIMAL(10,2) CHECK (emp_salary > 0),
   department VARCHAR(50) UNIQUE
);
```

8. Indexing – Improves Query Performance

- **Create an Index on a Column:**

```
CREATE INDEX idx_students_age ON students(age);
```

- **Drop an Index:**

```
DROP INDEX idx_students_age ON students;
```

9. Views – Virtual Tables

- **Create a View:**

```
CREATE VIEW student_view AS
SELECT name, age FROM students WHERE grade = '10th';
```

- **Select Data from View:**

```
SELECT * FROM student_view;
```

- **Drop a View:**

```
DROP VIEW student_view;
```

10. Stored Procedures – Reusable SQL Code

- **Create a Stored Procedure:**

```
DELIMITER $$
CREATE PROCEDURE GetAllStudents()
BEGIN
   SELECT * FROM students;
END $$
DELIMITER ;
```

- **Call a Stored Procedure:**

```
CALL GetAllStudents();
```

- **Drop a Stored Procedure:**

```
DROP PROCEDURE GetAllStudents;
```

Databases, Data Types, Indexes, and Functions

A **database** is an organized collection of data that allows users to store, manage, and retrieve data efficiently. Databases are managed using **Database Management Systems (DBMS)** like MySQL, PostgreSQL, SQL Server, and MongoDB.

1. Data Types in Databases

Data types define the kind of data that can be stored in a database column. Common categories of data types include:

a) Numeric Data Types

Used to store numbers.

- **INTEGER (INT)** – Whole numbers (e.g., 1, 100, -50)
- **FLOAT / DOUBLE** – Decimal numbers with precision (e.g., 12.34, 0.0005)
- **DECIMAL (NUMERIC)** – Fixed-precision decimal numbers, often used for currency (e.g., 99.99)

b) String (Character) Data Types

Used to store text.

- **CHAR(n)** – Fixed-length string (e.g., CHAR(10) always stores 10 characters)
- **VARCHAR(n)** – Variable-length string (e.g., VARCHAR(50) can store up to 50 characters)
- **TEXT** – Large text field, often for descriptions or logs

c) Date and Time Data Types

Used to store date and time values.

- **DATE** – Stores only date (YYYY-MM-DD)
- **TIME** – Stores only time (HH:MM:SS)
- **DATETIME / TIMESTAMP** – Stores both date and time

d) Boolean Data Type

Stores TRUE or FALSE values.

- **BOOLEAN / BOOL** – Typically stored as 1 (true) or 0 (false)

e) Other Data Types

- **BLOB (Binary Large Object)** – Stores binary data like images or files
- **UUID (Universally Unique Identifier)** – Stores unique identifiers

2. Indexes in Databases

An **index** is a database object that improves the speed of data retrieval operations.

Types of Indexes

1. **Primary Index** – Automatically created on the primary key.
2. **Unique Index** – Ensures that all values in a column are unique.
3. **Composite Index** – An index on multiple columns.
4. **Full-Text Index** – Used for searching text data efficiently.
5. **Clustered Index** – Stores data physically in the order of the index.
6. **Non-Clustered Index** – Maintains a separate structure for the index and data.

Example: Creating an Index in SQL

```
CREATE INDEX idx_employee_name ON employees(name);
```

Advantages of Indexes

- Faster query execution
- Efficient searching and sorting

Disadvantages

- Uses extra storage space
- Slows down INSERT, UPDATE, DELETE operations

3. Functions in Databases

Functions in databases are predefined or user-defined operations used to manipulate data.

a) Aggregate Functions

Used to perform calculations on multiple rows.

- SUM() – Returns the sum of values
- AVG() – Returns the average value
- COUNT() – Returns the number of rows
- MAX() – Returns the maximum value
- MIN() – Returns the minimum value

Example:

```
SELECT AVG(salary) FROM employees;
```

b) String Functions

Used to manipulate text.

- UPPER() – Converts text to uppercase
- LOWER() – Converts text to lowercase
- CONCAT() – Combines strings

Example:

```
SELECT CONCAT(first_name, ' ', last_name) FROM employees;
```

c) Date Functions

Used for date and time calculations.

- NOW() – Returns the current date and time
- DATE_ADD() – Adds days to a date
- DATEDIFF() – Finds the difference between two dates

Example:

SELECT DATEDIFF('2025-01-01', '2024-12-01');

d) User-Defined Functions (UDF)

Users can create their own functions for custom operations.

Example: Creating a function in SQL

```
CREATE FUNCTION get_discount(price DECIMAL) RETURNS DECIMAL
DETERMINISTIC
BEGIN
   RETURN price * 0.90; -- 10% discount
END;
```

Accessing MySQL Using PHP

PHP provides multiple ways to connect to a MySQL database and interact with it. The two most commonly used extensions are:

1. **MySQLi (MySQL Improved)**
2. **PDO (PHP Data Objects)**

1. Connecting to MySQL Using MySQLi

MySQLi can be used in two ways:

i. **Procedural** **ii. Object-Oriented**

Procedural MySQLi Connection

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "my_database";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);

// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Object-Oriented MySQLi Connection

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "my_database";

// Create connection
$conn = new mysqli($servername, $username, $password, $database);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

---

2. Performing Database Operations with MySQLi

Inserting Data

```php
<?php
$sql = "INSERT INTO users (name, email) VALUES ('John Doe', 'john@example.com')";
if (mysqli_query($conn, $sql)) {
    echo "Record inserted successfully";
} else {
    echo "Error: " . mysqli_error($conn);
}
?>
```

Retrieving Data

```php
<?php
$sql = "SELECT id, name, email FROM users";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " . $row["email"] . "<br>";
    }
} else {
    echo "No records found";
}
?>
```

Updating Data

```php
<?php
$sql = "UPDATE users SET email='newemail@example.com' WHERE name='John Doe'";
if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}
?>
```

Deleting Data

```php
<?php
$sql = "DELETE FROM users WHERE name='John Doe'";
if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
```

```
}
?>
```

3. Connecting to MySQL Using PDO

PDO is a more flexible way to connect to MySQL and supports multiple databases.

Connecting to MySQL with PDO

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "my_database";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$database", $username, $password);
    // Set error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

Inserting Data with PDO

```
<?php
$sql = "INSERT INTO users (name, email) VALUES (:name, :email)";
$stmt = $conn->prepare($sql);
$stmt->execute(['name' => 'Jane Doe', 'email' => 'jane@example.com']);
echo "Record inserted successfully";
?>
```

Retrieving Data with PDO

```
<?php
$sql = "SELECT id, name, email FROM users";
$stmt = $conn->query($sql);
while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
    echo "ID: " . $row["id"] . " - Name: " . $row["name"] . " - Email: " . $row["email"] . "<br>";
}
?>
```

Feature	MySQLi	PDO
Database Support	Only MySQL	Multiple databases (MySQL, PostgreSQL, etc.)
Security	Supports prepared statements	Supports prepared statements (more flexible)
Performance	Slightly faster for MySQL	Slightly slower but more secure
Object-Oriented	Yes	Yes
Recommended for	Small MySQL projects	Large applications with multiple databases

UNIT IV NETWORKING ESSENTIALS

Fundamentals of computer network – Types of computer networks – Network layer – TCP / IP model – Wireless Local Area Network – Ethernet – Wifi – Network Routing – Switching – Network Components

1. FUNDAMENTALS COMPUTER NETWORKS

Communication means to convey a message, a picture, speech or an idea that is received and understood clearly and correctly by the person for whom it is conveyed. Network is a set of devices connected by media links. The link connecting the devices is often called communication channels. Computer networking consists of two or more computers that are linked in order to share resources, exchange files, or allow electronic communications. The computers on a network may be linked through cables, telephone lines, radio waves, satellites, or infrared light beams. Data communication consists of five elements. They are sender, receiver, message, transmission medium and protocol.

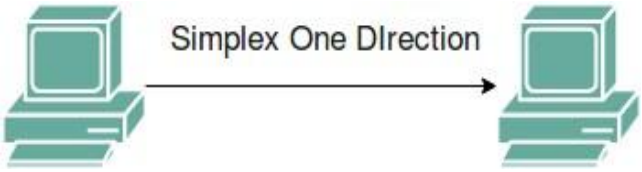
- **Sender** : Sender machine creates data and send it to the receiver machine.
- **Receiver:** Receive data and information from sender
- **Message :** The message is the information or data, that is to be communicated. It may consist of text, numbers, pictures, sounds, videos or any combination devices.
- **Protocol:** A set of rules that defines how data is formatted and processed on a network.
- **Transmission media**: It is a path between sender and receiver .Message is transmitted through this medium.

Point – to – Point link : In data communication, the point to point is commonly used to establish a direct connection between two networking devices. Point – to – point networks provide a dedicated link between any two stations. The data packets are sent from source station to the destination.

Multi-point link : Multi-point communication means one to many i.e. one source machine communicate with multiple receiver machine.

Transmission Modes in Computer Networks (Simplex, Half-Duplex and Full-Duplex) a. Simplex Mode

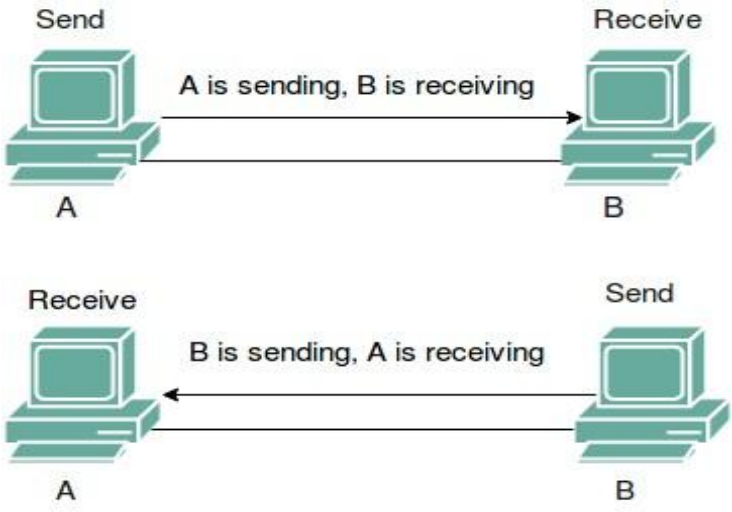
In Simplex mode, the communication is unidirectional, as on a one-way street. Only one of the two devices on a link can transmit, the other can only receive. The simplex mode can use the entire capacity of the channel to send data in one direction. Example: Keyboard and traditional monitors. The keyboard can only introduce input, the monitor can only give the output.



b.Half-Duplex Mode

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa. The half-duplex mode is used in cases where there is no need for communication in both direction at the same time. The entire capacity of the channel can be utilized for each direction.

Example: Walkie- Talkie in which message is sent one at a time and messages are sent in both the directions.



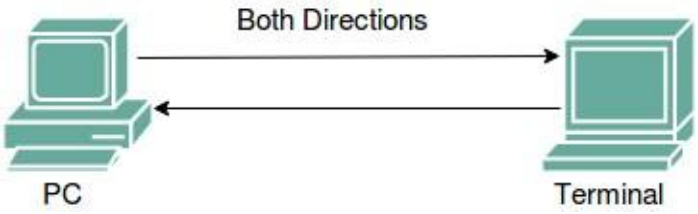
c.

Full-Duplex

In full-duplex mode, both stations can transmit and receive simultaneously. In full_duplex mode, signals going in one direction share the capacity of the link with signals going in other direction, this sharing can occur in two ways:

- Either the link must contain two physically separate transmission paths, one for sending and other for receiving.
- Or the capacity is divided between signals travelling in both directions.

Full-duplex mode is used when communication in both direction is required all the time. The capacity of the channel, however must be divided between the two directions. Example: Telephone Network in which there is communication between two persons by a telephone line, through which both can talk and listen at the same time.



Topologies

The physical topology of a network refers to the configuration of cables, computers, and other peripherals:

- Mesh
- Star
- Ring
- Bus
- Hybrid

a) Mesh Topology

Key Characteristics:

- Fully connected
- Robust – Highly reliable
- Not flexible
- Poor expandability

Media used for the connection (links) can be twisted pair, co-axial cable or optical fiber.. Mesh Topology is not flexible and has a poor expandability as to add a new node **‘ n’**links have to be laid because that new node has to be connected to each of the existing nodes via dedicated link, for the same reason the cost of cabling will be very high for a larger area.

b) Star Topology

- Each machine is connected to a central hub or switch.

- It allows each machine on the network to have a point to point connection to the central hub.
- All of the traffic which transverses the network passes through the central hub.
- The hub acts as a signal booster or repeater which in turn allows the signal to travel greater distances.
- Most widely implemented, Hub is the single point of failure.

Advantages	**Disadvantages**
Easily expanded without disruption to the network	Requires more cable
Cable failure affects only a single user	A central connecting device allows for a single point of failure
Easy to troubleshoot and isolate problems.	More expensive than bus topologies because of the cost of the hubs

c) Ring Topology

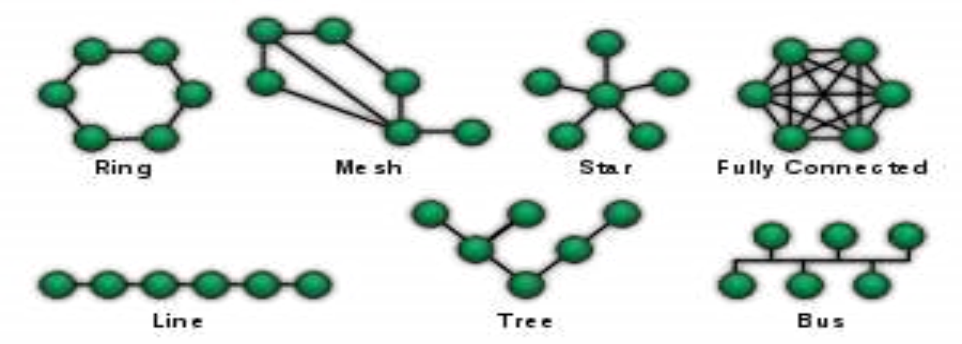
- Each computer is connected to the network in a closed loop or ring.
- Each machine or computer has a unique address that is used for identification purposes.
- The signal passes through each machine or computer connected to the ring in one direction.
- Ring topologies typically utilize a token passing scheme, used to control access to the network. Ring technique is based on the use of a small frame called a token that circulates when all the stations are idle. Whenever a station wishes to send a frame it waits until it receives a token. Since ring topologies use token passing to control access to the network, the token is returned to sender with the acknowledgement.
- Ring speeds are 4Mbps, 16 Mbps and 100 Mbps and uses twisted pair and fibre optic cable.
- By utilizing this scheme, only one machine can transmit on the network at a time.

Advantages	**Disadvantages**
Cable faults are easily located, making troubleshooting easier	Expansion to the network can cause network disruption
Ring networks are moderately easy to install	A single break in the cable can disrupt the entire network.

d) Bus Topology

- Each machine is connected to a single cable.
- Each computer or server is connected to the single bus cable through some kind of connector.
- A signal from the source travels in both directions to all machines connected on the bus cable until it finds the address on the network that is the intended recipient.
- If the machine address does not match the intended address for the data, the machine ignores the data.
- Alternatively, if the data does match the machine address, the data is accepted.

Advantages	**Disadvantages**
Cheap and easy to implement	Network disruption when computers are added or removed
Require less cable	High cost of managing the network Single point of failure.
Does not use any specialized network equipment	A break in the cable will prevent all systems from accessing the network. Difficult to troubleshoot.



.

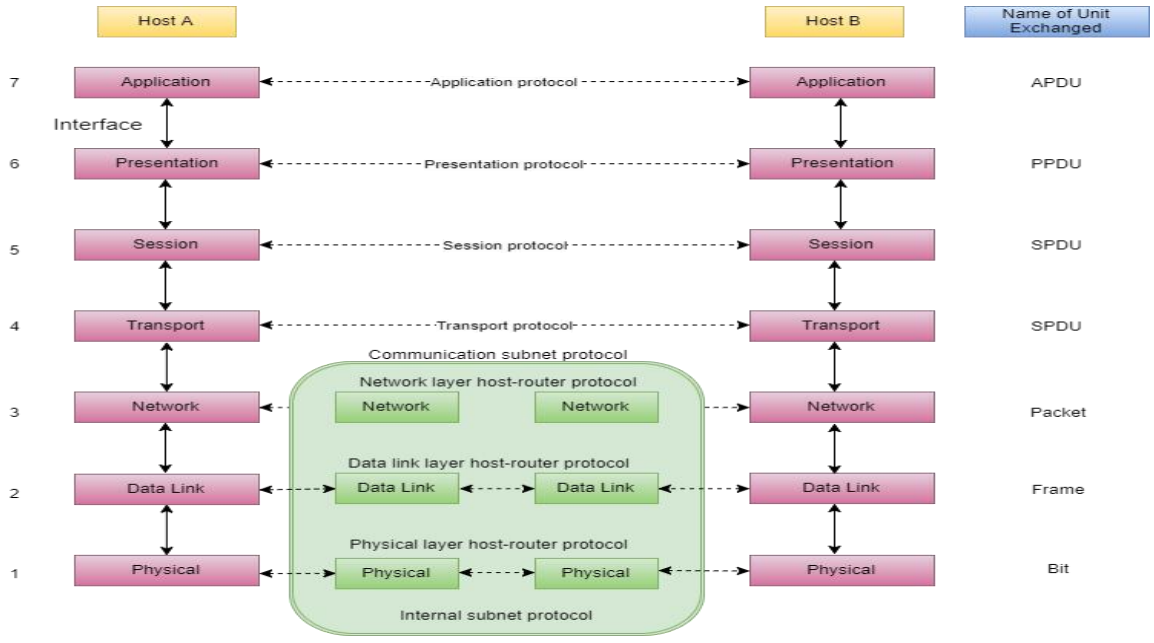
1.1 THE OSI MODEL

There are n numbers of users who use computer network and are located over the world. So to ensure, national and worldwide data communication, systems must be developed which are compatible to communicate with each other ISO has developed a standard. ISO stands for **International organization of Standardization**. This is called a model for **Open System Interconnection** (OSI) and is commonly known as OSI model.

The ISO-OSI model is a seven layer architecture. It defines seven layers or levels in a complete communication system. They are:

1. Application Layer
2. Presentation Layer
3. Session Layer
4. Transport Layer
5. Network Layer
6. Datalink Layer
7. Physical Layer

Below we have the complete representation of the OSI model, showcasing all the layers and how they communicate with each other.



In the table below, we have specified the **protocols** used and the **data unit** exchanged by each layer of the OSI Model.

Layer	Name of Protocol	Name of Unit exchanged
Application	Application Protocol	APDU - Application Protocol Data Unit
Presentation	Presentation Protocol	PPDU - Presentation Protocol Data Unit
Session	Session Protocol	SPDU - Session Protocol Data Unit
Transport	Transport Protocol	TPDU - Transport Protocol Data Unit
Network	Network layer host-router Protocol	Packet
Data LInk	Data link layer host-router Protocol	Frame
Physical	Physical layer host-router Protocol	Bit

Feature of OSI Model

1. Big picture of communication over network is understandable through this OSI model.
2. We see how hardware and software work together.
3. We can understand new technologies as they are developed.
4. Troubleshooting is easier by separate networks.
5. Can be used to compare basic functional relationships on different networks.

Principles of OSI Reference Model

The OSI reference model has 7 layers. The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that architecture does not become unwieldly. **Functions of Different Layers**

Following are the functions performed by each layer of the OSI model. This is just an introduction, we will cover each layer in details in the coming tutorials.

Layer 1: The Physical Layer

1. Physical Layer is the lowest layer of the OSI Model.
2. It activates, maintains and deactivates the physical connection.
3. It is responsible for transmission and reception of the unstructured raw data over network.
4. Voltages and data rates needed for transmission is defined in the physical layer.
5. It converts the digital/analog bits into electrical signal or optical signals.
6. Data encoding is also done in this layer.

Layer 2: Data Link Layer

1. Data link layer synchronizes the information which is to be transmitted over the physical layer.
2. The main function of this layer is to make sure data transfer is error free from one node to another, over the physical layer.
3. Transmitting and receiving data frames sequentially is managed by this layer.
4. This layer sends and expects acknowledgements for frames received and sent respectively. Resending of non-acknowledgement received frames is also handled by this layer.
5. This layer establishes a logical layer between two nodes and also manages the Frame traffic control over the network. It signals the transmitting node to stop, when the frame buffers are full.

Layer 3: The Network Layer

1. Network Layer routes the signal through different channels from one node to other.
2. It acts as a network controller. It manages the Subnet traffic.
3. It decides by which route data should take.

4. It divides the outgoing messages into packets and assembles the incoming packets into messages for higher levels.

Layer 4: Transport Layer

1. Transport Layer decides if data transmission should be on parallel path or single path.
2. Functions such as Multiplexing, Segmenting or Splitting on the data are done by this layer
3. It receives messages from the Session layer above it, convert the message into smaller units and passes it on to the Network layer.
4. Transport layer can be very complex, depending upon the network requirements.

Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.

Layer 5: The Session Layer

1. Session Layer manages and synchronize the conversation between two different applications.
2. Transfer of data from source to destination session layer streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.

Layer 6: The Presentation Layer

1. Presentation Layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data.
2. While receiving the data, presentation layer transforms the data to be ready for the application layer.
3. Languages(syntax) can be different of the two communicating systems. Under this condition presentation layer plays a role of translator.
4. It perfroms Data compression, Data encryption, Data conversion etc.

Layer 7: Application Layer

1. Application Layer is the topmost layer.
2. Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc are services provided by application layer.
3. This layer mainly holds application programs to act upon the received and to be sent data.

Merits of OSI reference model

1. OSI model distinguishes well between the services, interfaces and protocols.
2. Protocols of OSI model are very well hidden.
3. Protocols can be replaced by new protocols as technology changes.
4. Supports connection oriented services as well as connectionless service.

Demerits of OSI reference model

1. Model was devised before the invention of protocols.
2. Fitting of protocols is tedious task.
3. It is just used as a reference model.

2.TYPES OF COMPUTER NETWORKS

Computer Networks can be categorized depending on their size, distance and the structure namely: LAN (Local Area Network), MAN (Metropolitan Area Network), WAN (Wide Area Network).

LAN (Local Area Network)

A **Local Area Network** is a privately owned computer **network** covering a **small Networks geographical area,** like a home, office, or groups of buildings e.g. a school Network. It is used to connect the computers and other network devices so that the devices can communicate with each other to share the resources. The resources to be shared can be a hardware device like printer, software like an application program or data. The size of LAN is usually small.

Characteristics of LAN:

- Easy resource sharing
- Data transfer rate are high
- Small area covered by LAN
- Cost of setting up the network is usually low
- Flexibility, low error rate, reliability of operation and simple maintenance

Advantages of LAN:

- Availability to share hardware and software resources
- Support for heterogeneous forms of hardware and software
- Access to other LANs and WANs
- Private ownership
- Secure transfers at high speed with low error rates

MAN (Metropolitan Area Networks)

MAN stands for Metropolitan Area Networks is one of a number of types of networks. A MAN is a relatively new class of network. MAN is larger than a local area network and as its name implies, covers the area of a single city. MANs rarely extend beyond 100 KM and frequently comprise a combination of different hardware and transmission media.A MAN can be created as a single network such as Cable TV Network, covering the entire city or a group of several Local Area Networks (LANs). It this way resource can be shared from LAN to LAN and from computer to computer also. MANs are usually owned by large organizations to interconnect its various branches across a city.

The two most important components of MANs are security and standardization. Security is important because information is being shared between dissimilar systems. Standardization is necessary to ensure reliable data communication.

A MAN often acts as a high speed network to allow sharing of regional resources (similar to a large LAN). It is also frequently used to provide a shared connection to other networks using a link to a WAN. MAN provides the transfer rates from 34 to 150 Mbps.

WAN (Wide Area Networks)

A wide area network (WAN) is a telecommunication network. A wide area network is simply a LAN of LANs or Network of Networks. WANs connect LANs that may be on opposite sides of a building, across the country or around the world. WANS are characterized by the slowest data communication rates and the largest distances. WANs can be of two types: an enterprise WAN and Global WAN.

WANs (wide area networks) generally utilize different and much more expensive networking equipment than do LANs (Local Area Networks). Key technologies often found in WANs (wide area networks) include SONET, Frame Relay, and ATM. Each node in a WAN is a router that accepts an input packet, examines the destination address, and forwards the packet on to a particular telecommunication line. A router must select the one transmission line that will provide a path to the destination and in an optimal manner.
In a WAN, when the packet is sent from one router to another via one or more intermediate routers, the packets is received at each intermediate routers in its entirety. This packet is stored in that router until the required output line is free. WAN uses hierarchical addressing because they facilitate routing. Addressing is required to identify which network input is to be connected to which network output.

3. TCP/IP MODEL:

TCP/IP stands for Transmission Control Protocol/ Internet Protocol. This model is based on a five layer model for networking: Physical layer, Datalink layer, network layer, transport layer and application layer. The TCP/IP protocol stack is open. The TCP/IP protocol stack models a series of protocol layers for networks and systems that allows communication between any types of devices.
TCP breaks messages into packets, hands them off to IP software for delivery, and then orders and reassembles the packets at their destinations. IP stands for the Internet Protocol. It deals with the routing of packets through the maze of interconnected networks to their final destination. At the physical and Datalink layers, the TCP/IP protocols do not define any standards.

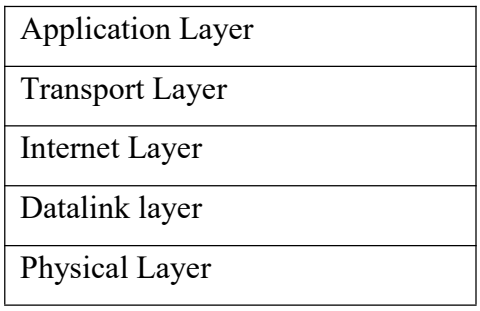


Fig: TCP/IP Protocol suite

Functions of TCP/IP Layers:

1. Application Layers: Application layer includes all process and services that uses the transport layer to deliver the data. The original TCP/IP Specification described a number of different applications that fit into the top layer of the protocol stack. These applications include Telnet, FTP, SMTP and DNS. TELNET is the Network Terminal Protocol, which provides remote login over the network. FTP is used for interactive file transfer. SMTP delivers electronic mail.

2. Transport Layer: This layer provides communication session management between host computers. It also defines the level of services and status of the connection used when transporting data. It also manages connection oriented steams, flow control, reliable transport and multiple transmission. Application programs send the data to the transport layer protocols TCP and UDP. An application is designed to choose either TCP/UDP based on the services it needs. The transport layer provides peer entities on the source and destination hosts to carry on a conversation. Data may be user data or control data. Two modes are available-full duplex and half duplex. In full-duplex operation ,both sides can transmit and receive data simultaneously, whereas in half-duplex, a side can only send or receive at one time.

3. Network or Internet Layers : Packages data into IP datagrams, which contain source and destination address information that is used to forward the datagrams between hosts and across networks. It performs routing of IP datagrams.
The internet/network level protocol (IP,ARP, ICMP) handles machine to machine communication. The primary protocol used to move data is the IP which provides fragmentation and addressing services. IP provides a connectionless method of delivering data from one host to another. It does not guarantee delivery and does not provide sequencing of datagrams. It attaches a header to datagram that includes source address and destination address, both of which are unique Internet Addresses.

4.Network Interface Layer: It contains two sub layers : DataLink Layer and Physical Layer. This layer is also called as host to network layer. This layer cannot define any protocol. It is responsible for accepting and transmitting IP datagrams. This layer may consists of a device driver in the operating systems and the corresponding network interface card in the machine. It specifies details of how data is sent physically sent through the network including how the bits are electrically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber or twisted pair copper wire.

Addressing:
An Internet employing TCP/IP protocols uses four level of addresses :

Physical Address: It is the lowest address and is also referred to as link address. The physical address of the node is defined by its LAN or WAN. The physical address is included in the frame by the data link layer.

Logical Address: Logical addresses are necessary for universal communications. It is a 32-bit address which uniquely defines host connected to Internet.

Port Address: In TCP/IP architecture, the label assigned to a process is called Port address. In TCP/IP the port address is of 16 bit.

Specific Address: They get changed to corresponding port and logical addresses by the station or the host who sends it.

Difference between TCP and IP

TCP	IP
TCP is used to transfer packet data	IP is responsible for logical addressing
TCP guarantees transfer of packet on a particular address	IP obtains that particular address
TCP breaks messages to packets and hands them off to the IP for software delivery, orders and then reassemble the packets at their destinations	IP deals with the routing of packets through the maze of interconnected networks to their final destination.
TCP is connection oriented protocol	Connectionless protocol
Reliable	Not Reliable

4. NETWORK LAYER

The main objective of the network layer is to allow end systems, connected to different networks, to exchange information through intermediate systems called router. The unit of information in the network layer is called a packet. It is responsible for addressing messages and data so that they are sent to the correct destination, and for translating logical addresses and names into physical addresses. This layer is also responsible for finding the path through the network to the destination computer. Lowest layer that deals with host-to-host communication, call this end-to-end communication.

Functions of Network Layer:

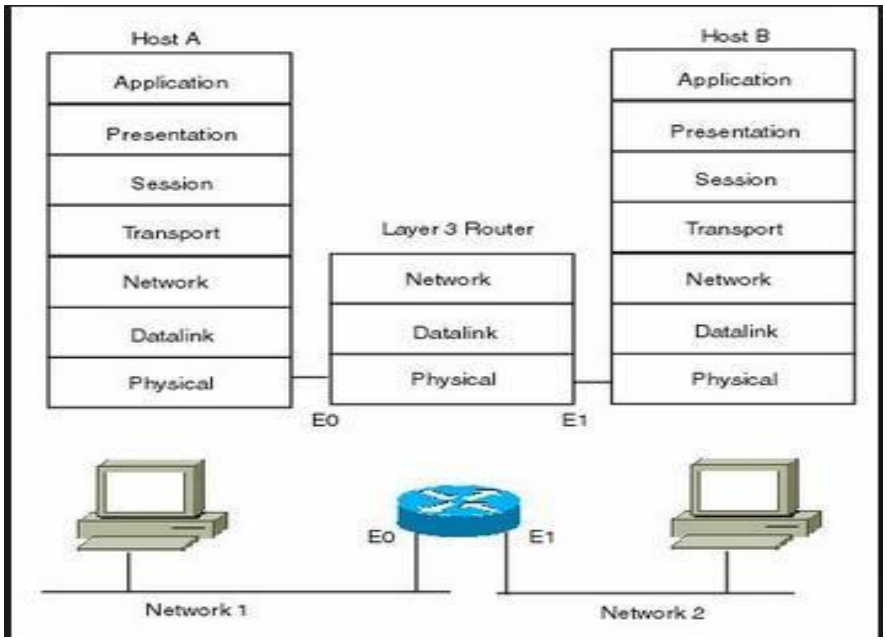
a) Logical Addressing- Data link layer implements physical addressing. When a packet passes network boundary, an addressing system is needed to distinguish source and destination, network layer performs these functions.

b)Routing- Network layer route or switch the packets to its final destinations in an internetwork.

c)Frame Fragmentation- If it determines that a downstream router's Maximum Transmission Unit(MTU) size is less than the frame size, a router can fragment a frame for transmission and re-assemble at the destination station.

Principle of network Layer:

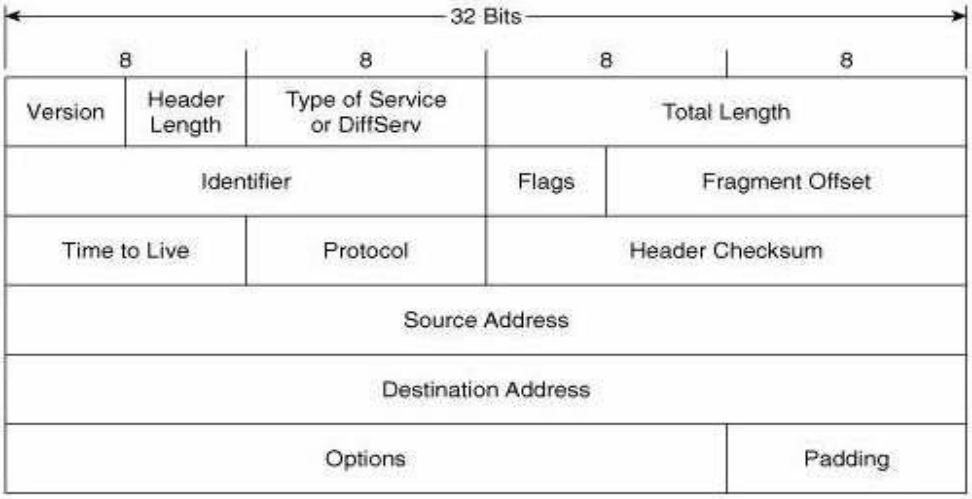
Each network layer entity is identified by a network layer address. This address is independent of the data link layer addresses that it may use. The network layer is conceptually divided into :Data plane and the control Plane. The data plane consists of protocols and mechanisms that allow hosts and routers to exchange packets carrying user data. The control plane contains the protocols and mechanisms that enable routers to efficiently learn how to forward packets towards their final destination.Datagram is used to provide a connectionless service while a virtual circuit is used in networks that provide a connection oriented service. Datagram is a type of packet that happens to be sent in a connectionless manner over the network. Every datagram carries enough information to let the network forward the packet to its correct destination. The network layer limits the maximum packet size.



5. IP Address:

Each computer in a TCP/IP network must be given a unique identifier, or IP address. This address, which operates at Layer 3, allows one computer to locate another computer on a network. All computers also have a unique physical address, which is known as a MAC address. These are assigned by the manufacturer of the NIC. MAC addresses operate at Layer 2 of the OSImodel.An IP address (IPv4) is a 32-bit sequence of ones and zeros.Tomake the IP address easier to work with, it is usually written as four decimal numbers separated by periods. For example, an IP address of one computer is 192.168.1.2.. This dotted decimal notation also prevents a large number of transposition errors that would result if only the binary numbers were used.

IPV4 Format:



Version:(4 bits): Indicates the version number, to allow evolution of the protocol.
Internet Header Length(IHL 4 bits): Length of header in 32 bit words. The minimum value is five for a minimum header length of 20octets.
Type-of-Service: The Type-of-Service field contains an 8-bit binary value that is used to determine the priority of each packet. This value enables a Quality-of-Service (QoS) mechanism to be applied to high priority packets, such as those carrying telephony voice data. The router processing the packets can be configured to decide which packet it is to forward first based on the Type-of-Service value.

Total length: total datagram length,in octets.
Identifier (16 bits): A sequence number that, together with the source address, destination address, and user protocol, is intended to uniquely identify a datagram.
Fragment Offset: A router may have to fragment a packet when forwarding it from one medium to another medium that has a smaller MTU. When fragmentation occurs, the IPv4 packet uses the Fragment Offset field and the MF flag in the IP header to reconstruct the packetwhen it arrives at the destination host. The fragment offset field identifies the order in which to place the packet fragment in the reconstruction.

Flags(3 bits): Only two of the bits are currently defined: MF(More Fragments) and DF(Don't Fragment):
More Fragments flag (MF): The More Fragments (MF) flag is a single bit in the Flag field used with the Fragment Offset for the fragmentation and reconstruction of packetsWhen a receiving host sees a packet arrive with the MF = 1, it examines the Fragment Offset to see where this fragment is to be placed in the reconstructed packet. When a receiving host receives a frame with the MF = 0 and a non-zero value in the Fragment offset, it places that fragment as the last part of the reconstructed packet. An unfragmented packet has all zero fragmentation information (MF = 0, fragment offset =0).
Don't Fragment flag (DF): The Don't Fragment (DF) flag is a single bit in the Flag field that indicates that fragmentation of the packet is not allowed. If the Don't Fragment flag bit is set, then fragmentation of this packet is NOT permitted. If a router needs to fragment a packet to allow it to be passed downward to the Data Link layer but the DF bit is set to 1, then the router will discard this packet.
IP Destination Address: The IP Destination Address field contains a 32-bit binary value that represents the packet destination Network layer host address.
IP Source Address: The IP Source Address field contains a 32-bit binary value that represents the packet source Network layer host address.
Time to Live: The Time-to-Live (TTL) is an 8-bit binary value that indicates the remaining "life" of the packet. The TTL value is decreased by at least one each time the packet is processed by a router (that is, each hop). When the value becomes zero, the router discards or drops the packet and it is removed from the network data flow. Decrementing the TTL value at each hop ensures that it eventually becomes zero and that the packet with the expired TTL field will be dropped.
Protocol: This 8-bit binary value indicates the data payload type that the packet is carrying. The Protocol field enables the Network layer to pass the data to the appropriate upper-layer protocol.Example values are:01 ICMP, 06 TCP, 17 UDP.
Header checksum (16 bits): An error-detecting code applied to the header only. Because some header fields may change during transit (e.g., time to live, segmentation-related fields), this is re-verified and recomputed at each router. For purposes of computation, the checksum field is itself initialized to a value of zero.
Options (variable): Encodes the options requested by the sending user.
Padding (variable): Used to ensure that the datagram header is a multiple of 32 bits.

5. WLANS - WIRELESS LOCAL AREA NETWORKS

A WLAN, or wireless LAN, is a network that allows devices to connect and communicate wirelessly. Unlike a traditional wired LAN, in which devices communicate over Ethernet cables, devices on a WLAN communicate via Wi-Fi.

The primary difference is how the data is transmitted. In a LAN, data is transmitted over physical cables in a series of Ethernet packets containing. In a WLAN, data is transmitted over the air using one of Wi-Fi 802.11 protocols.

Advantages of WLAN

Devices are connected wirelessly, eliminating the need for cables.

It also provides a way for small devices, such as smartphones and tablets, to connect to the network.

WLANs are not limited by the number of physical ports on the router and therefore can support dozens or even hundreds of devices.

Disadvantages of WLAN

Wireless networks are naturally less secure than wired networks.

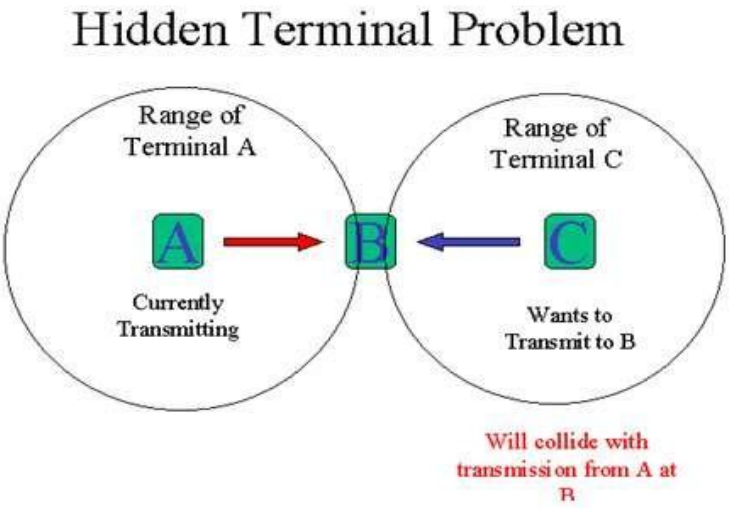
Any wireless device can attempt to connect to a WLAN, so it is important to limit access to the network if security is a concern. wireless networks are more susceptible to interference from other signals or physical barriers.

- The range of a WLAN can easily be extended by adding one or more repeaters.

Hidden Terminal:

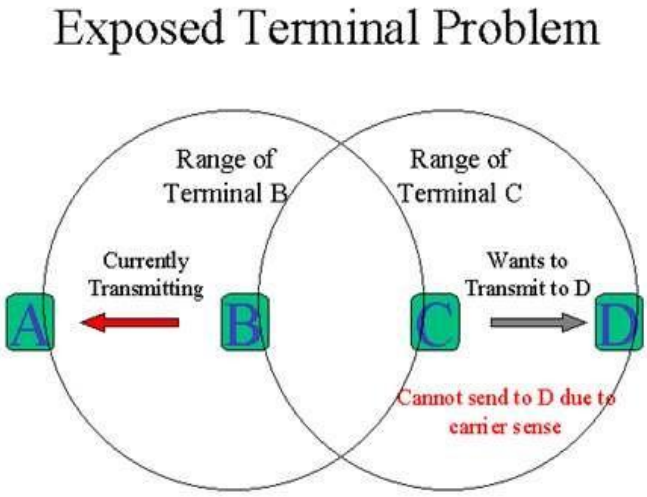
- As seen in the above problem, the transmission range of A reaches B but not C.
  Similarly, the range of C reaches B but not A. Also the range of B reaches both A and C.
- Now, the node A starts to send something to B and C doesn’t receive this transmission.
- Now C also wants to send data to B and senses the carrier. As it senses it to be free, it also starts sending to B.

- Hidden terminal problem occurs when two nodes that are outside each other's range performs simultaneous transmission to a node that is within the range of each of them resulting in a collision.
- That means the data from both parties A and C will be lost during the collision.
- Hidden nodes mean increased probability of collision at receiver end.
- One solution to avoid this is to have the channel sensing range much greater than the receiving range. Another solution is to use the Multiple Access with Collision Avoidance (MACA).



Exposed Terminal:

- Consider the same above diagram. Here imagine a situation wherein the B node is currently sending some data to node A.
- Now the other node C which is right now free want to send data to some node D(not in diagram) which is outside the range of A and B.
- Now before starting transmission it senses the carrier and realizes that the carrier is busy (due to interference of B's signal).
- Hence, the C node postpones the transmission to D until it detects the medium to be idle.
- However such a wait was un-necessary as A was outside the interference range of C.
- Also a collision at B will be a weak enough to be unable to penetrate into C
- Exposed terminal problem occurs when the node is within the range of a node that is transmitting and it cannot be transmitted to any node.Exposed node means denied channel access unnecessarily which ultimately results in under-utilization of bandwidth resources. It also results in wastage of time-resource



WLAN Protocols :

Multiple Access with Collision Avoidance (MACA) is a slotted media access control protocol used in wireless LAN data transmission to avoid collisions caused by the hidden station problem and to simplify exposed station problem.The basic idea of MACA is a wireless network node makes an announcement before it sends the data frame to inform other nodes to keep silent. When a node wants to transmit, it sends a signal

called *Request-To-Send* (RTS) with the length of the data frame to send. If the receiver allows the transmission, it replies the sender a signal called *Clear-To-Send* (CTS) with the length of the frame that is about to receive. Meanwhile, a node that hears RTS should remain silent to avoid conflict with CTS; a node that hears CTS should keep silent until the data transmission is complete.

Multiple Access with Collision Avoidance (MACA) for Wireless LAN's

WLAN data transmission collisions may still occur, and the MACA for Wireless (MACAW) is introduced to extend the function of MACA. It requires nodes sending acknowledgements after
each successful frame transmission, as well as the additional function of Carrier sense **Carrier Sense Multiple Access/Collision Avoidance**
In CSMA/CA, as soon as a node receives a packet that is to be sent, it checks to be sure the channel is clear (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear. This period of time is called the backoff factor, and is counted down by a backoff counter. If the channel is clear when the backoff counter reaches zero, the node transmits the packet. If the channel is not clear when the backoff counter reaches zero, the backoff factor is set again, and the process is repeated.

6. ETHERNET

Ethernet refers to the family of Local-Area Network (LAN) covered by the IEEE 802.3 standard that defines what is commonly known as the CSMA/CD protocol.Four data rates are currently defined for operation over optical fiber and twisted-pair cables : 10 Mbps-10 Base-T Ethernet, 100 Mpbs-Fast Ethernet, 1000 Mbps-Gigabit Ethernet and 10,000 Mbps-10 Gigabit Ethernet.Ethernet uses a communication concept called datagrams to get message across the network. Ethernet uses a CSMA/CD multiple access algorithm. **Carrier Sense Multiple Access with Collision Detection (CSMA/ CD)**
When node has data to transmit, the node first listens to the cable to see if a carrier(signal) is being transmitted by another node. This may be achieved by monitoring whether a current is flowing in the cable.The individual bits are sent by encoding them with a 10 clock using Manchester encoding. Data is only sent when no carrier is observed(i.e no current present) and the physical medium is therefore idle.
Any node which does not need to transmit, listens to see if other nodes have started to transmit information to it.The collision will result in the corruption of the frame being sent, which will subsequently be discarded by the receiver since a corrupted Ethernet frame will not have a valid 32- bit MAC CRC at the end.
If two or more stations have messages to send at the same time and they are separated by significant distances on the bus/channel, each may begin transmitting at roughly the same time without being aware of the other station.The signals from each node will superimpose on the channel and is garbled beyond the decoding ability of the receiving station. This is termed as "collision".
When there is data waiting to be sent, each transmitting node also monitors its own transmission. If it observes a collision, it stops transmission immediately and instead transmits a 32-bit jam sequence.The purpose of this sequence is to ensure that any other node which may currently be receiving this frame will receive the jam signal in place of the correct 32-bit MAC
CRC, this causes the other receivers to discard the frame due to a CRC error. When two or more transmitting nodes each detect a corruption of their own data (i.e a collision), each responds in the same way by transmitting the jam sequence.

MAC addresses

- Every device connected to an Ethernet network has a unique MAC address, assigned by the manufacturer of the network card. Its function is like that of an IP address, since it serves as a unique identifier that enables devices to talk to each other.

Preamble (7 bytes)	**Start Frame Delimiter (1 byte)**	**Destination Address (6 bytes)**	**Source Address (2 bytes)**	**Length or Type (2 bytes)**	**Data and Padding**	**CRC (4 bytes)**

- **Preamble** allows the receiver to synchronize with the signal. It is a sequence of alternating 0 and 1.

- **Start Frame Delimiter (SFD):** The sequence 10101011, which indicates the actual start of the frame and enables the receiver to locate the first bit of the rest of the frame.
- Both the source and destination host address are identified with a 48 bits address.
- **Type** field serves as the de-multiplexing key, i.e. it identifies to which of possibly many higher level protocols this frame should be delivered.
- **Data**: It is a minimum of 46 bytes and a maximum of 1500 bytes. The reason for minimum frame size is that the frame must be long enough to detect a collision.

- **CRC**: This field contains error detection information.
- Ethernet is a bit oriented protocol. Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet.

7. WiFi

- WiFi means "Wireless Fidelity". It is a wireless technology that uses radio frequency to transmit data through the air. The standard for Wireless Local Area Networks (WLANs). It's actually IEEE 802.11, a family of standards. WiFi is based on the 802.11 standard: 802.11a and 802.11g.WiFi systems are the half duplex shared media configuration, where all stations transmit and receive on the same radio channel.

- WiFi combines concepts found in CSMA/CD and MACAW, but also offers features to preserve energy.The developers of the 802.11 specifications develop a collision avoidance mechanism called the Distributed Control Functions(DCF). According to DCF, a WiFi station will transmit only when the channel is clear. All transmissions are acknowledged, so if a station does not receive an acknowledgement, it assumes a collision occurred and retires.

ISM Band

- ISM stands for industrial, scientific and medical. ISM bands are set aside for equipment that is related to industrial or scientific processes or is used by medical equipment.Perhaps the most familiar ISM-band device is the microwave oven, which operates in the 2.4-Ghz ISM band.The ISM bands are license-free, provided that devices are low-power. You don't need a license to set up and operate a wireless network.

- WLAN Architecture: **Ad-Hoc mode** : Peer-to-peer setup where clients can connect to each other directly. Generally not used for business networks.Mobile stations communicate to each other directly. It's set up for a special purpose and for a short period of time. For example, the participants of a meeting in a conference room may create an ad hoc network at the beginning of the meeting and dissolve it when the meeting ends.

WiFi networks services are as follows:

Distribution: This service is used by mobile stations in an infrastructure network every time they send data. Once a frame has been accepted by an access point, it uses the distribution service to deliver the frame to its destination. Any communication that uses an access point travels through the distribution service, including communications between two mobile stations associated with the same access point.

Integration: is a service provided by the distribution system, it allows the connection of the distribution system to a non-IEEE 802.11 network. The integration function is specific to the distribution system used and therefore is not specified by 802.11, except in terms of the services it must offers.

Authentication/Deauthentication : Physical security is a major component of a wired LAN security solution. Wired network's equipment can be locked inside offices. Wireless network cannot offer the same level of physical security, however, and therefore must depend on additional authentication routines to ensure that users accessing the network are authorized to do so.

Deauthentication: terminates an authenticated relationship. Because authentication is needed before network use is authorized, a side effect of deauthentication is termination of any current association.

Association: Delivery of frames to mobile stations is made possible because mobile stations register, or associate, with access points. The distribution system can then use the registration information to determine which access point to use for any mobile stations.

Reassociations: When a mobile station moves between basic services areas within a single extended service area, it must evaluate signal strength and perhaps switch the access point with which it is associated.

8. ROUTING

Routing is the process of transferring the packets from one network to another network and delivering the packets to the hosts. The traffic is routed to all the networks in the internetwork by the routers. In the routing process a router must know following things:

- Destination device address.
- Neighbor routers for learning about remote networks.
- Possible routes to all remote networks.
- The best route with the shortest path to each remote network.
- How the routing information can be verified and maintained.

Definition of Static Routing

Static routing does not involve any change in routing table unless the network administrator alters or modify them manually. Static routing algorithms function well where the network traffic is predictable. This is simple to design and easy to implement. There is no requirement of complex routing protocols.

Static routing is also known as **non-adaptive** routing which enables a pre-computed route to be fed into the routers offline. The administrative distance is a metric to measure the trustworthiness of the information received from a router. The default administrative distance for static route is 1, consequently the static routes will only be included in the routing table when there is a direct connection to that network. Static routes can be considered as an efficient method for a small and simple network that does not change frequently.

Definition of Dynamic Routing

Dynamic routing is a superior routing technique which alters the routing information according to the changing network circumstances by examining the incoming routing update messages. When the network change occurs, it sends out a message to the router to indicate that change, the routing software recalculates routes and sends the new routing update message. These messages pervade the network, enabling the router to change their routing tables accordingly.

The technique uses routing protocols to disseminate knowledge such as RIP, OSPF, BGP, etc. Unlike static routing, it does not require manual updation instead its automatic in manner and updates the routing table information periodically relying upon network conditions. For doing so, it requires extra resources for storing the information.

Dynamic routing is also referred to as **adaptive routing**. These algorithms change their routing decisions to reflect the changes in the topology or traffic.

Basis for comparison	**Static Routing**	**Dynamic Routing**
Configuration	Manual	Automatic
Routing table building	Routing locations are hand-typed	Locations are dynamically filled in the table.
Routes	User defined	Routes are updated according to change in topology.
Routing algorithms	Doesn't employ complex routing algorithms.	Uses complex routing algorithms to perform routing operations.
Implemented in	Small networks	Large networks
Link failure	Link failure obstructs the rerouting.	Link failure doesn't affect the rerouting.
Security	Provides high security.	Less secure due to sending broadcasts and multicasts.
Routing protocols	No routing protocols are indulged in the process.	Routing protocols such as RIP, EIGRP, OSPF, BGP etc are involved in the routing process.
Additional resources	Not required	Needs additional resources to store the information.

Advantages and Disadvantages Static Routing

Advantages	Disadvantages
Easily implemented in a small network.	Unsuitable for complex topologies and large networks.
No overheads are produced on router CPU.	Large networks increase configuration complexity and time consumption.
Secure because the routes are managed statically.	Link failure can hinder traffic rerouting.
It is predictable as the route to the destination is fixed.	The administrator must be extra careful while configuring the routes.
Extra resources (such as CPU and memory) are not required as update mechanisms are not needed.	
Bandwidth usage is not required between routers.	

Advantages and Disadvantages of Dynamic Routing

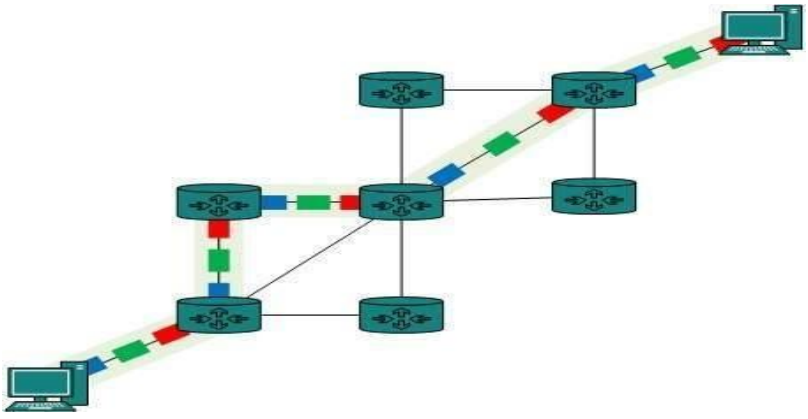
Advantages	Disadvantages
• Suitable for all the topologies. • Network size doesn't affect the router operations. • Topologies are adapted automatically to reroute the traffic. • Additional resources are required such as CPU, memory and link bandwidth.	• Initially, it could be complicated to implement. • The broadcasting and multicasting of routing updates make it less secure. • Routes rely on current topologies.

9. SWITCHING

Switching is process to forward packets coming in from one port to a port leading towards the destination. Circuit Switching

When two nodes communicate with each other over a dedicated communication path, it is called circuit switching.There 'is a need of pre-specified route from which data will travel and no other data is permitted. In circuit switching, to transfer the data, circuit must be established so that the data transfer can take place.Circuits can be permanent or temporary. Applications which use circuit switching may have to go through three phases:

- Establish a circuit
- Transfer the data
- Disconnect the circuit

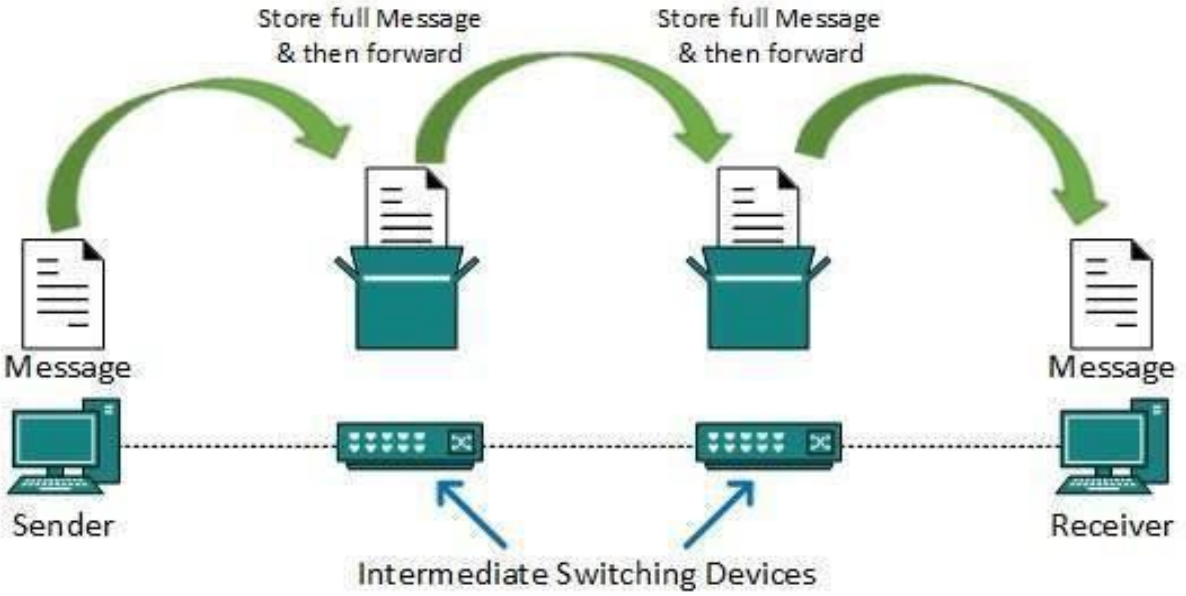


Circuit switching was designed for voice applications. Telephone is the best suitable example of circuit switching. Before a user can make a call, a virtual path between caller and callee is established over the network.

Message Switching

This technique was somewhere in middle of circuit switching and packet switching. In message switching, the whole message is treated as a data unit and is switching / transferred in its entirety.A switch working on message switching, first receives the whole message and buffers it until there are resources available to

transfer it to the next hop. If the next hop is not having enough resource to accommodate large size message, the message is stored and switch waits.

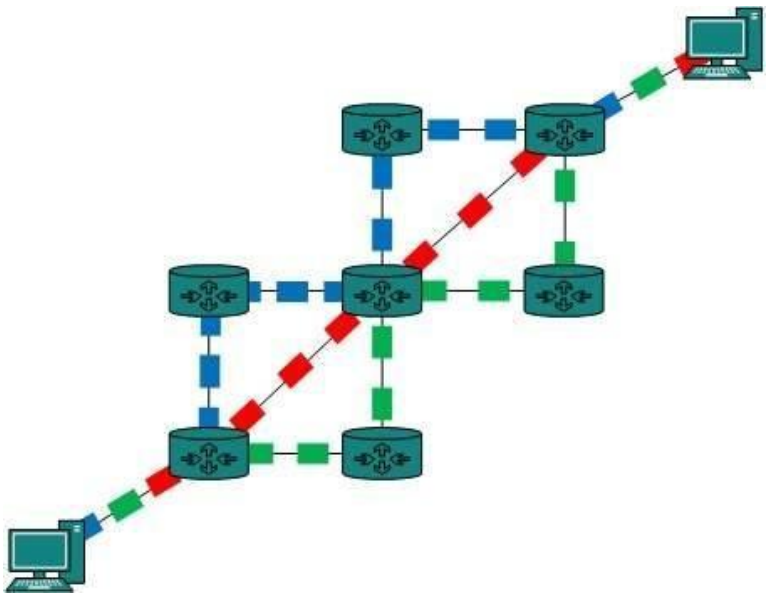


This technique was considered substitute to circuit switching. As in circuit switching the whole path is blocked for two entities only. Message switching is replaced by packet switching.
Message switching has the following drawbacks:

- Every switch in transit path needs enough storage to accommodate entire message.
- Because of store-and-forward technique and waits included until resources are available, message switching is very slow.
- Message switching was not a solution for streaming media and real-time applications.

Packet Switching

Shortcomings of message switching gave birth to an idea of packet switching. The entire message is broken down into smaller chunks called packets. The switching information is added in the header of each packet and transmitted independently.It is easier for intermediate networking devices to store small size packets and they do not take much resources either on carrier path or in the internal memory of switches.



Packet switching enhances line efficiency as packets from multiple applications can be multiplexed over the carrier. The internet uses packet switching technique. Packet switching enables the user to differentiate data streams based on priorities. Packets are stored and forwarded according to their priority to provide quality of service.

10. NETWORK COMPONENTS

Computer network is a group of two or more computers that connect with each other to share a resource. **Sharing of devices and resources is the purpose of computer network.** You can share printers, fax machines, scanners, network connection, local drives, copiers and other resources.

Major computer network components include:

- **Repeater –**Repeater operate at the physical layer. It forwards bits from one LAN segment to another. The basic purpose of a repeater is to extend the distance of LAN. A repeater is a network device that is used to regenerate or replicate signals that are weakend or disorted by transmission over long distance and through areas with high levels of electromagnetic interference. Repeaters do not have physical addresses on the network and do not translate anything.

- **Bridge** – Bridge operates at the Data link layer. It has a single input and single output port. A bridge extends the maximum distance of network by connecting separate network segments. A bridge simply passes on all the signals it receives. It uses MAC addresses to handle traffic flow. Bridge performs data link functions such as error detection, frame formatting and frame routing.

  Advantages : Simple to use and install, transparent to users, additional software is not required, it forms single logical networks.

  Limitations : It suffers from broadcast storms, fault isolation is not provided.

- **Router –**When we talk about computer network components, the other device that used to **connect a LAN with an internet connection is called Router**. When you have **two distinct networks** (LANs) or want to share a single internet connection to multiple computers, we use a Router**.**

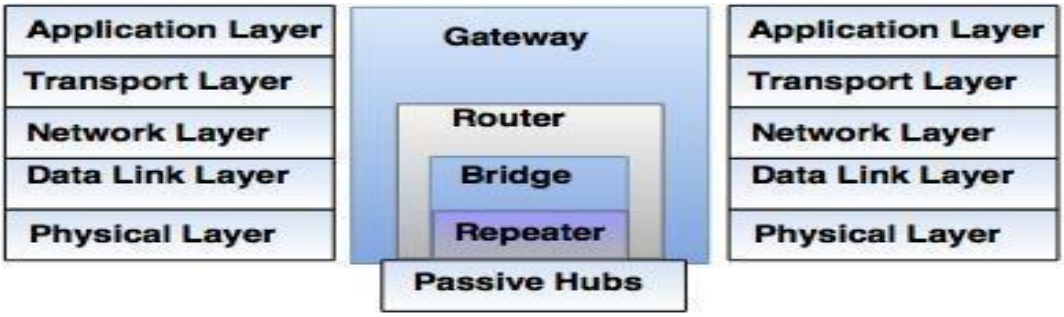
- **Gateway** – Gateway is combination of networking hardware and software that connects two dissimilar kinds of networks. A gateway is a protocol converter and operates on all seven layers of the OSI model. A gateway can accept a packet formatted for one protocol (TCP /IP) and convert it to a packet formatted for another protocol (Apple Talk) before forwarding it.

- **Network Interface Card (NIC) – Network adapter** is a device that enables a computer to talk with other computer/network. Using unique **hardware addresses (MAC address)**encoded on the card chip, the data-link protocol employs these addresses to discover other systems on the network so that it can transfer data to the right destination.

  There are **two types of network cards**: **wired and wireless**. The wired NIC uses cables and connectors as a medium to transfer data, whereas in the wireless card, the connection is made using antenna that employs radio wave technology.

- **Hub** –Hub is a device that splits a network connection into multiple computers. It is like a distribution center. When a computer request information from a network or a specific computer, it sends the request to the hub through a cable. The hub will receive the request and transmit it to the entire network. Each computer in the network should then figure out whether the broadcast data is for them or not.
- **Advantages** : most economic way of expanding network, simple, inexpensive network.
- **Limitations**: cannot connect different Ethernet types, hubs do not isolate collision domains

- **Switches** –Switch is a telecommunication device grouped as one of computer network components. Switch is like a Hub but built in with advanced features. It uses **physical device addresses** in each incoming messages so that it can deliver the message to the right destination or port. Like Hub, switch don't broadcast the received message to entire network, rather before sending it checks to which system or port should the message be sent.



Types of Connecting Devices

UNIT- V
APPLICATION ESSENTIALS

Creation of simple interactive applications – Simple database applications – Multimedia applications – Design and development of information systems – Personal Information System – Information retrieval system – Social networking applications

I. CREATION OF SIMPLE INTERACTIVE APPLICATION:

Definition of Interactive Application:

An interactive application is a collection of objects intended for performing certain task when user triggers the command. The non-interactive applications operate without human involvement. For example- Compiler and batch processing applications are non-interactive applications. The compiler is a program that converts the high level programs in the machine language. Similarly the batch processing is the execution of a series of jobs in a program on a computer without manual intervention.

The interactive application share a Graphical user interface (GUI). The interactive application consists of forms on which the components such as buttons, text fields, and radio buttons are present.

In the interactive applications one page may get linked with other pages dynamically. Especially in case of interactive web application this linking of the pages is through hyperlinks. This linking is based on user input. Typical examples of interactive web applications are online course registration system, online shopping system and so on.

Advantages of Creating Interactive Applications:

It increases the engagement of users in the collaborative productions. Instead of spending lot of time on one way presentation being shared across a PowerPoint presentation, an interactive application allows the user to engage with the information used. Files can be easily shared accessed, edited and saved. The interactive applications boost the communication. The information can be shared using email, print and files. The intention of the user to handle that application can be identified by the interactive session he makes with the application. This helps in boosting the communication.

The documents can be annotated effective changes can be made while handling the interactive application. The tools such as 3D modeling, hyperlinks, video links and other applications can be embedded within an interactive application.

The interconnectivity among the interactive applications allows the greater availability. Users can connect to interactive applications using iOS and Android smart devices. The information can be made available to the fingertips by using touch screen technologies. This helps a novice or illiterate user to handle that application with ease and comfort.

User's Perspective about the Interactive Applications:

The users use the interactive application for getting required information. The users always want that the desired information must be neatly arranged on the application. There are two kinds of browsing the information – known item searching and casual browsing. Known item Searching - The known item searching is done by the users who know what they are looking for . They know what is the particular label, how to proceed for getting particular information and from where to leave. They just quickly find the required information and leave. Casual Browsing type of users does not know what they are looking for. Sometimes they do not know anything about the product or the services on the corresponding website.

Producer's Perspective about the interactive Application:

Cost is a major issue while building the information architecture from producer's perspective. One cannot predict the exact cost of the information architect. However, depending upon the goals and nature of the application. While building the interactive application architecture by using producer's perspective:

1. The organizational goals and vision must be clear to information architect.
2. The decision of the contents to be placed on the main page should be taken prior to implementation of the architecture.
3. The information architecture must be kept away from organizational politics.
4. The site should be designed so that the intended audience will get the satisfaction of using it.
5. All the controversial issues must be resolved during the design process, before the application gets built.

Steps for creating Simple Interactive Web Applications

1. **Understanding Data Items and the Data dictionary:**

Data item identifies the unit of information. It defines how the item can be used. A data dictionary is a collection of descriptions of data objects or items in a data model. The data dictionary is dynamic; any changes in the data item are effective immediately for all applications that include the data item. Applications access the data dictionary at runtime and immediately reflect modifications to data item. The data dictionary is created for modeling the data items in the interactive applications.

2. **Understanding the Table Design:**

A relational database table stores the data that an application uses in columns and rows. Each column is a data item, and each row is a record. One or more tables can be used in an application. To create a table, the required data items are selected. These data items must already exist in the data dictionary. Then assign the key fields as indices for retrieving and updating the data belonging to the tables. Various operations that can be performed on this database table's are- create, insert record and delete particular record, and update some record.

3. Understanding Business View Design:

A business view is a selection of data items from one or more tables. After you create a table, use Business View Design to choose only the data items that are required for your application. Use appropriate SQL statements to retrieve data from any of the supported database.

4. **Understanding Form Design:**

Form design is an important part of the interactive application. Applications are composed of forms and a form is the interface between a user and a table. This interface should present the data logically and contain the functions that are necessary to enter and manipulate data. The form design task can be accomplished by placing various components such as text boxes, Labels, Buttons, Check Buttons and so on the form at appropriate locations.

5. **Understanding Report Design:**

Report Design is used to present the business data stored in the database. Report is basically some kind of template. The data is typically presented using batch applications that access the data through business views. Each report is comprised of sections which are the building blocks of all reports. Within the template, you can add, hide, remove and rearrange sections as needed. One can create variations of reports in the interactive applications.

6. **Understanding Data Structure Design:**

Data structure is a key element of any programming language or environment. A data structure is a list of parameters that passes data among applications and tables or forms.

7. **Understanding Event Rules Design:**

Events are the activities that occur on the form of an interactive application. Event can be initiated by user or an application. The event rules can be created for following purposes.

1. Perform mathematical calculation.
2. Pass data from one field in the form to another field in another form.
3. Interconnect two forms.
4. Hide and display the controls using system functions.
5. Assign the value or an expression to particular field.
6. Creation of variables or programmer defined field at run time.
7. Process table input and output, validate data and retrieve record.

The areas at which the event rules are applicable are:

1. **Controls:** Controls are reusable components that can be placed on the form. For eg: Radio buttons, text boxes and so on. The event can be triggered from these controls.
2. **Form Processing:** Form processing means application of business logic associated with each form. Form Processing depends on the occurrence of specific events.
3. **Table:** You can create database triggers, or rules that you attach to a table by using Table Design Event Rules. The logic that is attached to a table is run whenever any application initiates that database event.

8. Understanding system Functions:

System Functions are procedures provided by the tool and are usually specific to the type of component being used. For example there are system functions to hide and show fields on an application.

II. SIMPLE DATABASE APPLICATIONS:

Definition: Database is an organized collection of data. A database management system (DBMS) is a computer software application that interacts with the user, other applications and the database itself to capture and analyze data. **Examples:** MS-Access, Oracle, MySQL

Characteristics of Database Applications

1. Consistency: DBMS provide greater consistency to the forms of data storage.
2. Support for Query Language – To retrieve and manipulate the data efficiently
3. Multiuser Environment: Simultaneous access of the database without creating conflicts.
4. Less Data Redundancy:
5. Relationship among Data
6. Security

Advantages of DBMS

1. **Reduced Data redundancy** – The database approach removes the redundancy by integrating the files. But this approach cannot eliminate redundancy completely but can control the data redundancy.
2. **Data consistency** - In this approach all copies of data are kept consistent.
3. **Sharing of data** - The centralized stored database can be accessed by multiple users or application programs.
4. **Centralized database**- Data is stored in a single repository and an authorized access to this data is allowed. Only the administrator can give appropriate access rights to the authentic users.
5. **Data Integrity** – This provides validity and consistency of stored data.
6. **Improved Security** – The database approach prevents unauthorized access to data by means of user name, password, and access rights.
7. **Use of Standards** - Certain standards can be enforced to database approach for data formats, naming conventions, documentation standards, access rules and so on.
8. **Backup and Recovery** - The Backup and Recovery – The backup and recovery actions can be taken in its current consistent state from inconsistent state
9. **Increased productivity** - The database approach allows programmer to concentrate on specific functionality required by the user.
10. **Increased Concurrency**- The database approach handles the concurrent data effectively. The loss of information or integrity is avoided in this approach.
11. **Improved Maintenance**

Disadvantages of DBMS:

- Complexity – Difficult to implement
- Size - Large storage space
- Cost - The multiuser database management system is very expensive.

Data Models

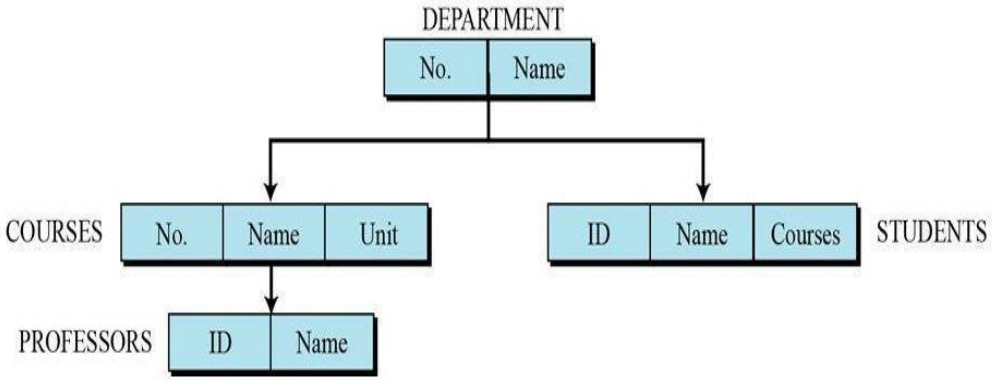
They describe the logical structure of a database, relationship between the database stored in database and various constraints on data. **Importance of Data Model**

1. End users have different view for data.
2. Data model organizes data for different users.

Types of Data Model:

1. **Hierarchical Model:**

In this model, each entity has only one parent but can have several children. At the top of the hierarchy there is only one node called root. This model represents the relationship in 1:N types.



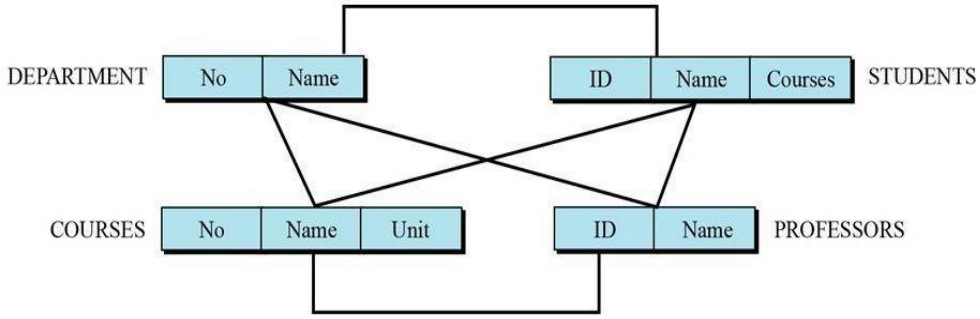
i.e One university can have multiple courses. One course can have multiple projects and so on. **Advantages:**
This model groups the data into tables and defines the relationship between the tables.

Disadvantages

- For searching any data, we have to start from the root and move downwards and visit each child node. Thus traversing through each node is required.
- For addition of some information about child node, sometimes the parent information needs to be modified.
- It fails to handle many to many relationships. It causes duplication and data redundancy.

2. Network Model:

This is enhanced version of hierarchical model. It addresses the M:N relationship . i.e No single parent concept. So, any child in this model can have any number of parents.

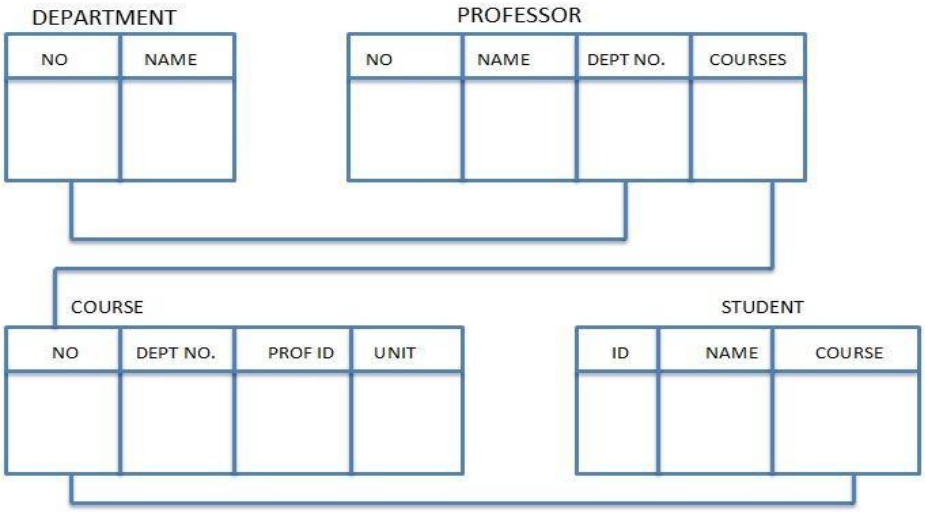


Advantages:

- Capability to handle more relationships: Since the network model allows many to many relationship, it helps in modeling real life situations.
- Ease of data access
- Data Integrity **Disadvantages:**
- Complex to implement
- Complicated Operations
- Difficult to change structure

3. Relational Model:

In a relational model, the data is stored in the form of tables. The database systems based on it are called relational database systems. These tables are related to each other.



Advantages:

- The database design is simple to implement and manage.
- In this model, the table is an entity which is primarily used. Hence insertion, deletion and retrieval of data becomes easy.
- The data can be easily linked with other table, using the table attributes.
- It facilitates the support for SQL languages.
- It ensures the structural independence.

Disadvantages:

- It has substantial hardware and software overhead.
- It can facilitate to poor design and implementation.
- It cannot handle the data in the form of images, audio or video.

4. E-R Model:

This model is basically used to design the relational database systems; the primary purpose of E-R Model is to show the relationship between various data objects. The object relationship pair can be graphically represented by a diagram called Entity Relationship Diagram (ERD).

Components of ERD:

Entity:

An entity is an object that exists and is distinguishable. It is similar to a record in a programming language with attributes and entities are represented by rectangle.

Relationship:

An association among several entities. It is represented in diamond. Relationship may have attributes and have cardinality (e.g. One-to-many).

Attribute:

It is drawn in ellipses. It is similar to record fields in a programming language. Each attribute has a set of permitted values called domain.

Notations used in ER Diagram:

Entity: It is an object and distinguishable it is similar to record.

Weak Entity: When this entity is dependent upon some another entity then it is called weak entity.

Attribute: The attributes are properties or characteristics of an entity.

Derived attribute:

It is a kind of attribute which is based on another attribute.

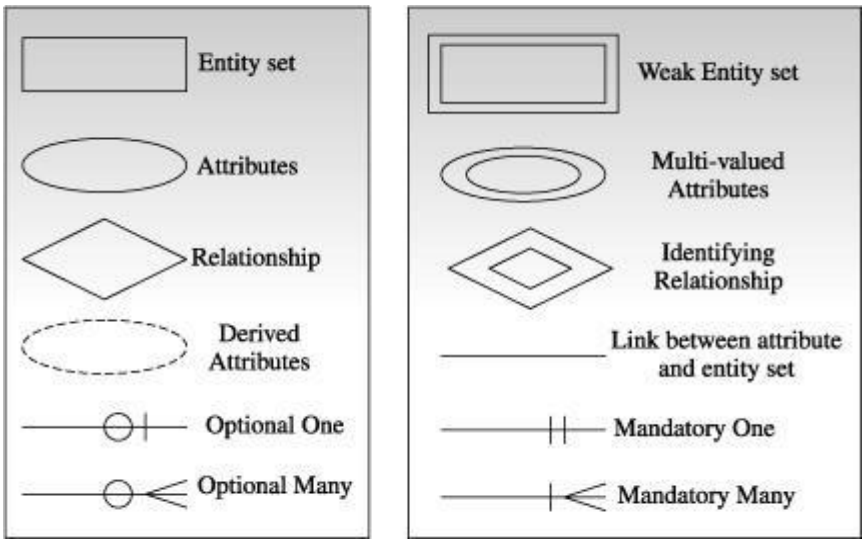
Key attributes:

A key attribute is an unique attribute representing distinguishing characteristics of entity. Typically primary key of record is a key attribute.

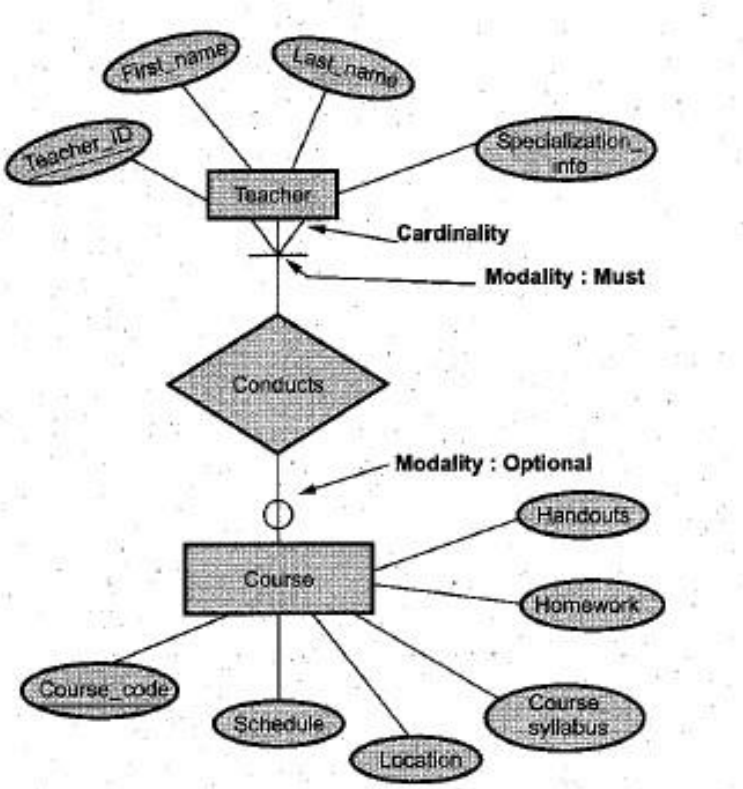
Multivalued attribute: A multivalued attribute have more than one value.

Relationship:

When two entities share some information then it is denoted by relationship.



Example: Draw an ER diagram for the relationship of teacher, and courses. Also specify the association, cardinality and modality.



Association:

In the above ER diagram, a relationship conducts is introduced. "Teacher" is associated with "Course" by conducting it.

Cardinality:

Many Teachers can conduct the single course.

Modality:

For conduction of a course, there must be a Teacher. There may be a situation that a Teacher is not conducting any course.

Advantages:

- Simple to design.
- Visual Representation.
- Easy Conversion to tables. **Disadvantages:**
- Limited Specification.
- High level design.
- No industry standard.

Architecture of Database Systems:

The database system architecture represents the structure and layout of the data stored in it. The architecture of database system can be from single tier to multitier. The multi-tiered system divides the whole system into multiple modules. Each of these individual modules can be independently altered or modified.

Database Tier:

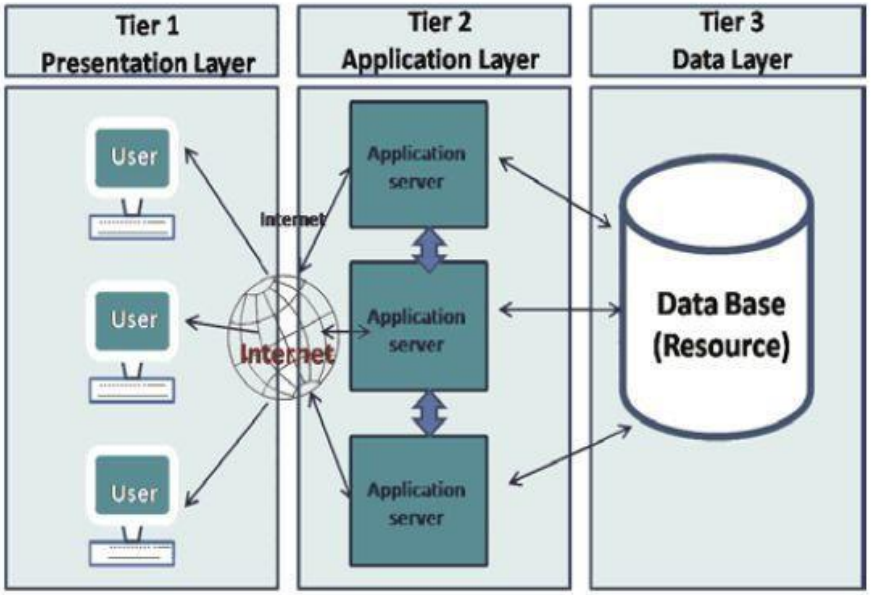
This is the layer at which the actual database resides. In this layer, all the tables, their mappings and the actual data present. When you save you details from the front end, it will be inserted into the respective tables in the database layer, by using the programs in the application layer. When the user wants to retrieve the data from this database, the database layer fires the queries to get the data from the tables present in the database**.**

Application Tier:

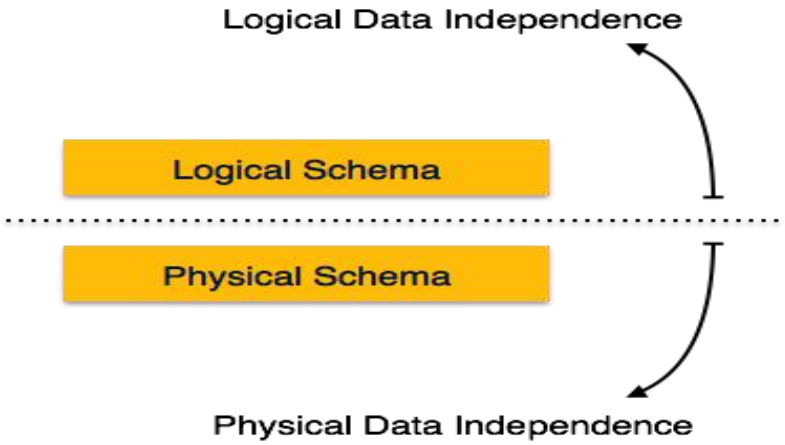
This layer sends the requests made by the users of presentation tier to the database tier and returns the response from database tier to presentation tier. The layer has all the business logics like validation, calculations and manipulations of data. If this layer sees that the request is invalid, it sends back the message to presentation layer. Hence, the application layer sits in the middle and acts as the mediator between the users and the database.

Presentation Tier:

This layer is made up of the users who use the database. These users have no knowledge of underlying database. At this layer multiple views of the database are provided.



Data Independence:



Definition: Data independence is an ability by which one can change the data at one level without affecting the data at another level. Here level can be **physical, conceptual or external.** Data independence is one of the most important characteristics of database management system. By this property, the structure of the database can be easily modified by without changing the application programs. There are two types of independence:

1. **Physical Independence:**

This is the kind of data independence which allows the modification of physical schema without requiring any

change to the conceptual schema. For example – if there is any change in memory size of database server then it will not affect the logical structure of any data object.

2. Logical Independence:

This kind of data independence which allows the modification of conceptual schema without requiring any change to the external schema. For example – Any change in the table structure such as addition or deletion of some columns does not affect user views.

By these data independence the time and cost acquired by changes in any level can be reduced and abstract view of data can be provided to the user.

Data Dictionary:

Concept: Data dictionary contains information about database itself. The data dictionary thus contains the **metadata** i.e. data about data. Following types of information is stored in data dictionary.

- Definition of database objects such as tables, views, constraints, clusters, procedures, functions, triggers
- Column name
- Data type information
- Amount of space required to store the data object
- Default field values
- Access rights
- Database usernames – Schema information
- Last updated or accessed information

All this information can be stored in tables , XML files or in spreadsheet.The data dictionary is updated automatically by the database systems when user issues the corresponding queries.

Keys used in Database Applications

Keys are important in relational database in order to establish the relationship between the tables. The keys are also used to access the records in the database table. Following are the various keys that are used in database system.

1. Primary Key:

This is the most important key in database which uniquely identifies the record. It can be a single attribute or combination of attributes. The database designer has to specially assign one of the candidate keys as primary key so that the record can be uniquely identified with the help of it. For example – **Stud_RollNo** is a primary key from the student database.

Stud_RollNo	FirstName	LastName	Address	CourseId
1001	AAA	BBB	Mumbai	C101
1002	XXX	YYY	Pune	M201
1003	PPP	QQQ	Bhopal	E303

2. Candidate Key:

A candidate key is a single field or the least combination of fields that uniquely identifies each record in the table. But the individual attribute cannot identify the record uniquely. For example – **FirstName, LastName, Address** in combination can uniquely identify the record, but individually FirstName, LastName or Address cannot uniquely identify the record.

Stud_RollNo	FirstName	LastName	Address	CourseId
1001	AAA	BBB	Mumbai	C101
1002	XXX	YYY	Pune	M201
1003	PPP	QQQ	Bhopal	E303

3. Foreign Key:

A Foreign Key is generally a primary key for one table that appears as a field in another where the first table has relationship to the second. For example: consider the following student table in which CourseID is a foreign key.

Stud_RollNo	FirstName	LastName	Address	CourseId
1001	AAA	BBB	Mumbai	C101
1002	XXX	YYY	Pune	M201
1003	PPP	QQQ	Bhopal	E303

CourseId	Name
C101	Computer Engineering
M201	Mechanical Engineering
E303	Environmental Engineering
V505	Civil Engineering

4. Composite Key:

The key that consists of two or more attributes that uniquely identifies an entity occurrence is called Composite key. For example to identify the Student taking particular course we can uniquely identify such type of record by combining two or more columns from the same table. Hence, **Stud_RollNo** and **CourseId** together form a composite key.

Stud_RollNo	CourseId
1001	C101
1002	M201
1003	E303

III.MULTIMEDIA APPLICATIONS:

The word multimedia means more than one media for conveying information. The multimedia can be defined as:

Definition: Computer- based techniques of text, images, audio, video, graphics, animation , and any other medium where every type of information can be represented, processed, stored, transmitted, produced and presented digitally.

Examples:

Some of the important programs are listed below in some categories. They are:

- Maya, Flash, Blender, comes mainly under graphics category.
- Interactivity category basically includes MySQL, AJAX, Flash and Flex and PHP.
- Audio category is of sound slides, Pro-tools, Adobe Auditions and more.
- Similarly programs in video category are Canopus Edius, i Movie, Flash Video Encoder, Final Cut Pro.
- Text programs are like Word press, InDesign, and Dream Weaver.

Components of Multimedia:

1. Text can be added for giving emphasis.
2. Graphics are added for visual impact. A picture is worth a thousand words. Graphics enhance presentation.
3. Voice or audio enhance presentation by adding persuasion.
4. The animation is for attracting attention. A chart can be focused more quickly by adding animation to it.
5. Video is multimedia can be used for providing clear instructions.

Uses of Multimedia:

1. Education:

Multimedia is extensively used in the fields of education and training. Even in conventional method we use audio visual for imparting education, where charts, model etc. were used. Now a days the classroom need is not limited to that conventional method rather it needs audio and visual media. The multimedia integrates all of them in one system. For the use of multimedia as an education aid the PC contains a high quality display. The software package named computer aided instruction is available that provides a friendly interactive method of learning.

2. Training:

There are various systems and intelligent tutoring systems available to train the students in many areas starting from the mathematics of a primary student to a difficult surgical process for a medical student.

3. Business:

The business application of multimedia includes, product demos, instant messaging. One of the excellent applications is voice and live conferencing. A multimedia can make audience come live. It is widely used in programs. The quality og business communication can be enhanced by multimedia. Product promotion, customer information, communication to employee can be done by multimedia.

4. Games and Entertainment:

Real life games can be created by multimedia. Developers use sound, animation, graphics of multimedia to create

games. Flight simulator creates real life imaging.

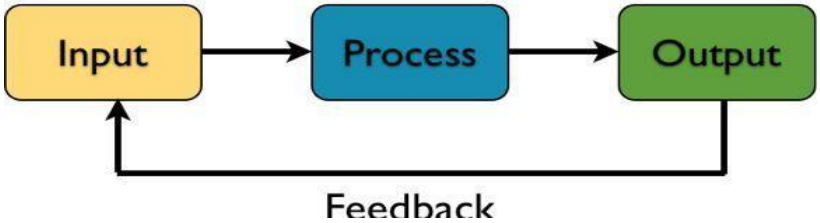
IV. INFORMATION SYSTEM

Definition: An Information System (IS) is a set of interrelated components that collect, manipulate, store and disseminate data and information and provide a feedback mechanism to meet an objective.

Examples:

- **1. Supply Chain Management:** This system is for managing the flow of goods and services that involves the movement and storage of raw materials, work-in-progress inventory and finished goods.
- **2. Customer Relationship Management:** Manages communication and marketing initiatives directed at customers.
- **3. Geographic Positioning System (GPS)**: it provides driving directions and desired locations.
- **4. Enterprise Resource Planning (ERP):** It is an information system used to integrate the management of all internal and external information across an entire organization. The parts of information system are:

1. **Input**
2. **Processing**
3. **Output**
4. **Feedback**



1. **Input:**

Input is an activity of gathering and capturing raw data. For example- in online student information system, instructor has to submit the detailed information students the summary of students information can be compiled.

2. **Processing:**

Processing means converting or transforming data into useful output.

Processing can involve making calculations, comparing data and taking alternatives actions, and storing data for future use. For example – in tax management system – the tax needs to be calculated from using the data such as gross salary, insurances, other deductions and so on. Processing data into useful information is critical in business settings. Processing can be done manually or with computer assistance. After processing the data results are typically stored in storage.

3. **Output:**

Output involves producing useful information in the form of **documents or reports**.

For example – in online banking system, the report is getting generated on the current transaction and can be presented to the customer in the form of bank statement. This statement shows the details of all the transactions and the current available balance in the account.

4. **Feedback:**

In information system, feedback is information from the system which is used to make changes to input or processing activities. For example, errors or problems might make it necessary to correct input data or change a process. Consider a payroll example. Perhaps the number of hours an employee worked was entered as 400 instead of 40. Feedback is very important to managers or decision makers in order to modify the system.

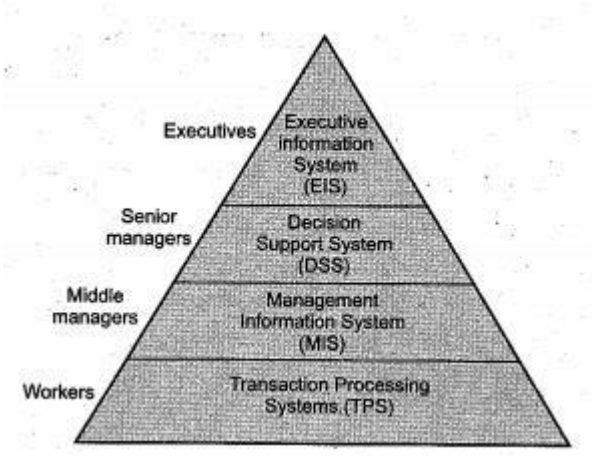
Characteristics of Information System:

- **Accessibility** : Information present in the information should be easily accessible by authorized users.
- **Accurate**: The information must be accurate and error free.
- **Complete**: Information present in the information system must be complete so that it wil satisfy all the queries of its users.
- **Relevant**: The relevant information is important for the decision maker.
- **Reliable**: This is very important characteristics of the information system. The reliable information is trusted by its users. The reliability of information depends upon the sources of data collection for the information systems.
- **Secure**: The information systems must be secure and prevent any unauthorized access to it.
- **Simple**: The information system must be very simple to handle and not very complex. It should not present the information system with too many details whereby the decision maker is unable to determine what is really important.

- **Timely**: The information in the information system must be delivered in timely manner whenever it is required.
- **Economical**: Information must be economical to produce.
- **Verifiable**: The information must be verifiable, that means one can check it to make it sure that information available in the information system is correct.

Components of Information Systems:
Hardware: Computer-based information systems use computer hardware, such as processors, monitors, keyboard and printers.
Software: These are the programs used to organize process and analyze data.
Databases: Information system work with data, organized into tables and files.
Network: Different elements need to be connected to each other, especially if menu different people in an organization use the same information system.
Procedures: These describe how specific data are processed and analyzed in order to get the answers for which the information system is designed.

Types:



Various types of information systems are:

1. **Transaction Processing System (TPS):**
   - The transaction processing system provides way to collect, process, store, display, modify or cancel the transactions.
   - This system allows multiple transactions to take place simultaneously.
   - The data collected by this system is typically stored in databases which can be used to produce reports such as billing, inventory summaries and manufacturing schedules. **Examples**- payroll system, order processing system and stock control systems.

Properties of Transaction Data (ACID PROPERTIES)
- **Atomicity**: The transaction must occur completely or not executed at all.
  Example: Withdrawal from one account and deposition in other account completes one complete transaction.
- **Consistency**: The data written in the dB must be valid data. This validity must be based on the constraints, triggers and other combinations.
- **Isolation**: If transaction is executed concurrently then it may come to some state. If transaction is executed sequentially then it may come to some other state. The isolation property ensures that these two states are equal.
- **Durability**: A transaction is said to be durable if it is committed and remains in the state even is case of power failure or system crash. To satisfy this property the transaction must be recorded in non- volatile memory.

2. **Management Information System (MIS):**
   - A management information system is information system that uses data collected by Transaction Processing System (TPS).

- This data is used for creating reports in such a way that managers can make use of it to make important business decisions.
- Some reports are created to present the summary. These activities are performed to increase the efficiency of managerial activity.

Examples-

- Sales management system
- Inventory control system
- Budgeting system
- Management reporting system
- Personnel system

3. Decision Support System (DSS):

- The decision support system helps make decision by working and analyzing data that can generate statistical projections and data models.
- This system gives support rather than replacing a managers judgment while improving the quality of a manager decision.
- The DSS helps solve problems while using external data.

Examples-

- Logistics systems
- Financial Planning systems
- Spreadsheet models

4. Executive Information Systems (EIS):

- It is strategic level of information system that is present at the top level of the pyramid.
- It helps executives and senior managers analyze the environment in which the organization operates, to identify long-term trends, and to plan appropriate courses and action.
- EIS organizes and present data and information from both external data sources and internal MIS or TPS in order to support and extend the inherent capabilities of senior executives.
- EIS emphasizes graphical displays and easy-to-use user interfaces.
- In recent years, the term EIS has lost popularity in favor of business intelligence.

Examples – Financial Analysis system

Design and Development of Information System

1. Feasibility study:

- The aim of a feasibility study is to see whether it is possible to develop a system at a reasonable cost. At the end of the feasibility study a decision is taken whether to proceed or not.
- A feasibility study contains the general requirements of the proposed systems.It may be that development of a new system is not needed instead an update of the existing is enough.

2. Requirement Analysis:

- This is very important part in the development of an Information System and involves looking at an organization or system and finding out how information is being handled at the moment.
- The stage where users and IT specialists work together to collect and comprehend the business requirements. Based on requirements, both will work on the design and discuss the tasks to be done.
- The requirement analysis document is prepared at the end of this stage.

3. Design:

- At, this stage the systems blueprint is created.
- The technical architecture is designed which includes telecommunications, hardware and software suited for the system.
- The design process include: Outputs, Inputs, File Design, Hardware, Software
- The system design should be done for : user interface, data design, process design

4. Development and Testing:

- Any new system needs to be thoroughly tested before being introduced.
- During this stage the building of the technical architecture, database and programs are executed.
- It is also the stage where the system is tested using the established test scripts and compare the expected outcomes to actual outcomes.

5. **Implementation:**
   - The stage where system is in place and is used by the actual workforce.
   - User guide manual and training are provided to users.

6. **Evaluation:**
   - During this stage system need to be evaluated for any bug from time to time.

7. **Maintenance:**
   - This is the stage where system needs to be enhanced or strengthened in order to meet the goals of the organization.

V.PERSONAL INFORMATION SYSTEM:

- Personal Information management is set of activities in which people perform in order to acquire, organize, maintain, retrieve and use personal information such as documents, web pages, email messages every day to accomplish the assigned tasks.
- Personal Information System (PIS) maintains the information about the employees in, department like personal, promotional, postings, qualifications, awards, incentives, leave etc. That assists an organization in many ways.
- There are various roles in the personal information system such as- employee, manager, customer, student and so on.
- Conceptually PIS is a collection information and methods that help the people to maintain the information of persons.
- This information system can be maintained offline. One can carry this information system in pen drive.

Example -

- Address book system
- Personal Notes
- Email notification
- Reminders and Alert system
- Lists
- Personal File collection system(document, music, photos)
- Instant messaging systems

Need for Personal Information System:

- This system saves times and efforts in locating the information.
- Information system is used for east retrieval of information.
- The information system organizes the entire information systematically.
- Using personal information system within an organization means better employee productivity and better team work in the near term.

Functionality of Personal Information System:

There are two modes of functionality of personal information system:

1. **User panel:** The user panel is for entering the personal information such as profile details, qualification details, and employment details.
2. **Administrator panel:** The administrator panels maintain following activities like user settings, profile master, qualification master, and document upload master, email settings, printer settings.

Benefits of Personal Information System:

- Personal information system contains the data of all its users.
- Users can easily search and locate data with personal information management system.
- Information stored in personal information system is transferrable to other locations and software programs.

Exercise: Design simple personal application that gives you reminders for each day. Identify the inputs to be taken, processing to be done, and the output to be produced . What multimedia components can be added to this application?

The personal application for reminder is a simple and effective application that can be used in busy schedule for reminding the day to day activities.

Features of this application:

- Users can set / update date / time of particular event.
- The history data can be cleared.
- Priority of task can be set or changed.
- One can feed to-do list to the application.
- The meeting schedule can be input to the application. The remaining application will display the schedule one hour prior to actual schedule.
- The birthdays, anniversaries or important dates can be reminded on particular dates by flashing images, messages and ringing alarm. Users can stop the alarm or press 'remind me' after sometime button.
- Email data via, name of person, email address, phone number and so on can be used by the application as input. This feature can be set if user permits to do so.
- The day / date / time can be set according to appropriate time zone of the country.

The GUI for simple personal application that gives you reminders is as follows:

- **Input** : Name of the person, birthdate, anniversary date, meeting time, purpose of meeting, allotted timing for meeting.
- **Processing**: It involves making calculations, matching data against system date, matching person name, storing data for future use.
- **Output**: Displaying reminding information on the device, displaying date, ringing alarm, flashing light.

Multimedia Components:

- **Text**: The text is used for typing the input to the system as well for displaying the name of the event, detailed information about some schedules, to-do list, name of the person and so on.
- **Graphics**: The attractive graphics flashing as output on matching with date or time of particular event.
- **Image**: The image / photo of the persons can be displayed on the app while reminding the birthdays and anniversaries.
- **Audio**: Melodious songs or ringtone will be ringing for the reminding alarm.
- **Animation**: Animated images or text can be displayed on the device for reminding app on particular event.