S

Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH

Estd.2001

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

UIDESIGN-FLUTTER LAB (R22CSE3121)

LAB MANUAL

III Year I Semester

DEPARTMENT OF INFORMATION TECHNOLOGY

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

B. TECH –INFORMATION TECHNOLOGY

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competency, values and social consciousness.

INSTITUTION MISSION

IM$_1$ Provide high quality academic programs, training activities and research facilities.

IM$_2$ Promote Continuous Industry-Institute Interaction for Employability, Entrepreneurship, Leadership and Research aptitude among stakeholders.

IM$_3$ Contribute to the Economical and technological development of the region, state and nation.

DEPARTMENT VISION

To be a recognized knowledge centre in the field of Information Technology with self - motivated, employable engineers to society.

DEPARTMENT MISSION

The Department has following Missions:

DM$_1$ To offer high quality student centric education in Information Technology.

DM$_2$ To provide a conducive environment towards innovation and skills.

DM$_3$ To involve in activities that provide social and professional solutions.

DM$_4$ To impart training on emerging technologies namely cloud computing and IOT with involvement of stake holders.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: **Higher Studies:** Graduates with an ability to apply knowledge of Basic sciences and programming skills in their career and higher education.

PEO2: **Lifelong Learning:** Graduates with an ability to adopt new technologies for ever changing IT industry needs through Self-Study, Critical thinking and Problem solving skills.

PEO3: **Professional skills:** Graduates will be ready to work in projects related to complex problems involving multi-disciplinary projects with effective analytical skills.

PEO4: **Engineering Citizenship:** Graduates with an ability to communicate well and exhibitsocial, technical and ethical responsibility in process or product.

PROGRAM OUTCOMES (POs) & PROGRAM SPECIFIC OUTCOMES (PSOs)

PO	Description
PO 1	**Engineering Knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO 2	**Problem Analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO 3	**Design/Development of Solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO 4	**Conduct Investigations of Complex Problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO 5	**Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO 6	**The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO 7	**Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO 8	**Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO 10	**Project Management and Finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO 11	**Life-Long Learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)
Program Specific Outcomes	
PSO 1	**Software Development:** To apply the knowledge of Software Engineering, Data Communication, Web Technology and Operating Systems for building IOT and Cloud Computing applications.
PSO 2	**Industrial Skills Ability:** Design, develop and test software systems for world-wide network of computers toprovide solutions to real world problems.
PSO 3	**Project implementation:** Analyze and recommend the appropriate IT Infrastructure required for the implementationof a project.

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time),those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with thesynopsis / program / experiment details.
3. Student should enter into the laboratory with:
   a) Laboratory observation notes with all the details (Problem statement, Aim, Algorithm,Procedure, Program, Expected Output, etc.,) filled in for the lab session.
   b) Laboratory Record updated up to the last session experiments and other utensils (if any)needed in the lab.
   c) Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation notebook, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain thediscipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the labafter completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Head of the Department **Principal**

TABLEOFCONTENTS

1.DIMENSIONSOFTHELAB		
AreaofthelabinSqmts	:	66Sqm
2.CAPACITYOFTHELAB	:	60Students
3.EQUIPMENTS		
ComputerSystems(Clients)	:	60
CPU	:	60
Monitors	:	60
KeyBoard	:	60
Mouse	:	60
4. SYSTEMCONFIGURATION	**:**	**Intel®Core™I3**
Speed	:	3.10GHz,2GBRAM
HardDisk	:	500GB
HCLLEDMonitorSize	:	18.5
5. SOFTWARE	**:**	**Android Studio**
6. AMBIENCE		
Printers	:	01
Projector	:	01
ComputerTables	:	60
StudentChairs	:	60
Charts	:	02
PhotoFrames	:	03
Switch/Hub	:	03
WhiteBoards	:	01
A/Cs	:	02
PowerBackup	:	UPS

R22CSE3121: UIDESIGN-FLUTTER

B.Tech.IIIYearISem. **LTPC**
0021

CourseObjectives:

- LearnstoImplementFlutterWidgetsand Layout
- UnderstandsResponsiveUI DesignandwithNavigationinFlutter
- KnowledgeonWidgesandcustomizewidgetsforspecificUIelements, Themes
- Understandtoincludeanimationapartfromfetchingdata

CourseOutcomes:

- ➢ ImplementsFlutterWidgetsandLayouts
- ➢ ResponsiveUI DesignandwithNavigationin Flutter
- ➢ Create custom widgets for specific UI elements and also Apply styling using themes and custom styles.
- ➢ Designa formwithvarious input fields, alongwithvalidationanderror handling
- ➢ Fetches data and write code for unit Test for UI components and also animation

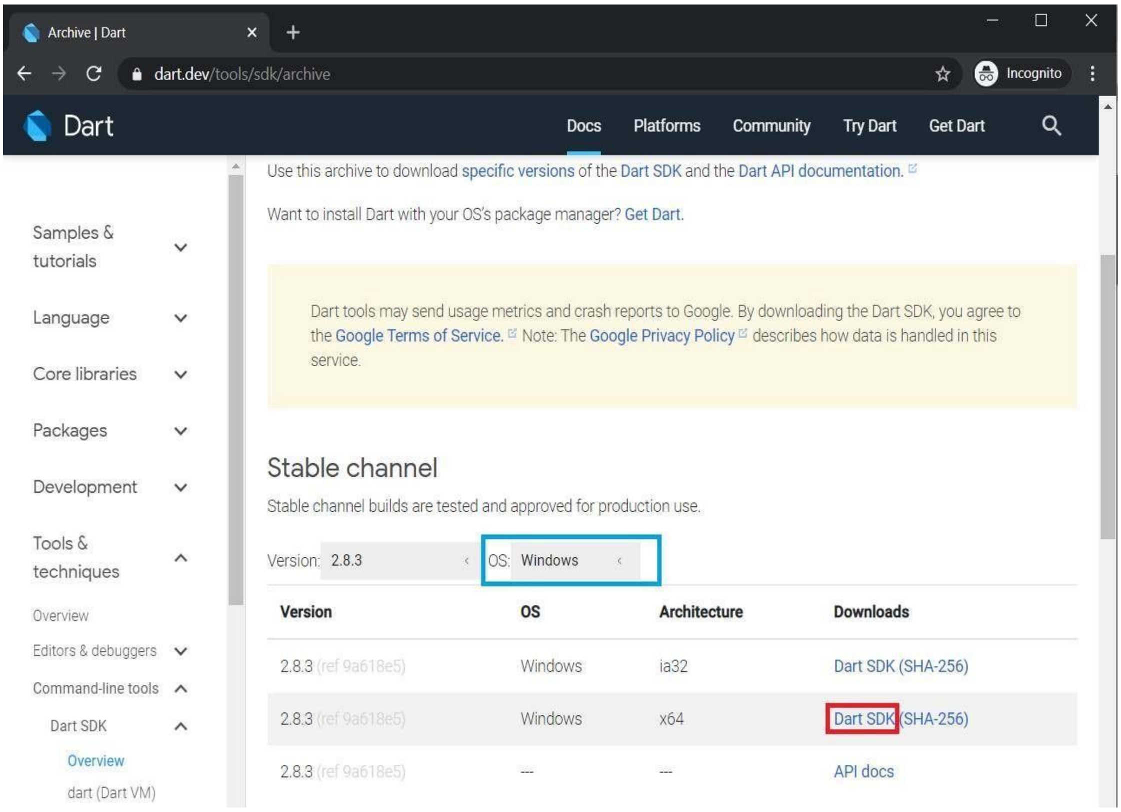
ListofExperiments:Studentsneedtoimplementthefollowing experiments

1. InstallFlutterandDartSDK.
2. WriteasimpleDartprogramtounderstandthelanguagebasics.
3. ExplorevariousFlutterwidgets(Text,Image,Container,etc.).
4. Implementdifferent layoutstructuresusingRow,Column,andStackwidgets.
5. Setupnavigationbetweendifferent screensusingNavigator.
6. Implementnavigationwithnamedroutes
7. UseFlutter'sdebuggingtoolstoidentifyandfixissues.
8. DesignaresponsiveUI thatadaptstodifferent screensizes.
9. Implementmediaqueriesandbreakpointsforresponsiveness.
10. Createcustomwidgetsfor specificUI elements.
11. Applystylingusingthemesandcustomstyles.
12. AddanimationstoUIelementsusingFlutter'sanimation framework.

TEXTBOOK:
1.MarcoL.Napoli,BeginningFlutter:AHands-onGuidetoAppDevelopment.

1.InstallFlutterandDartSDK.

Ans) Dart SDK is a pre-compiled version so we have todownloadand extractitonly. For this follow the below-given instructions: **Step 1:** Download Dart SDK. DownloadDartSDKfromtheDartSDKarchivepage.TheURL is: https://dart.dev/tools/sdk/archive



Use this archive to download specific versions of the Dart SDK and the Dart API documentation.

Want to install Dart with your OS's package manager? Get Dart.

Dart tools may send usage metrics and crash reports to Google. By downloading the Dart SDK, you agree to the Google Terms of Service. Note: The Google Privacy Policy describes how data is handled in this service.

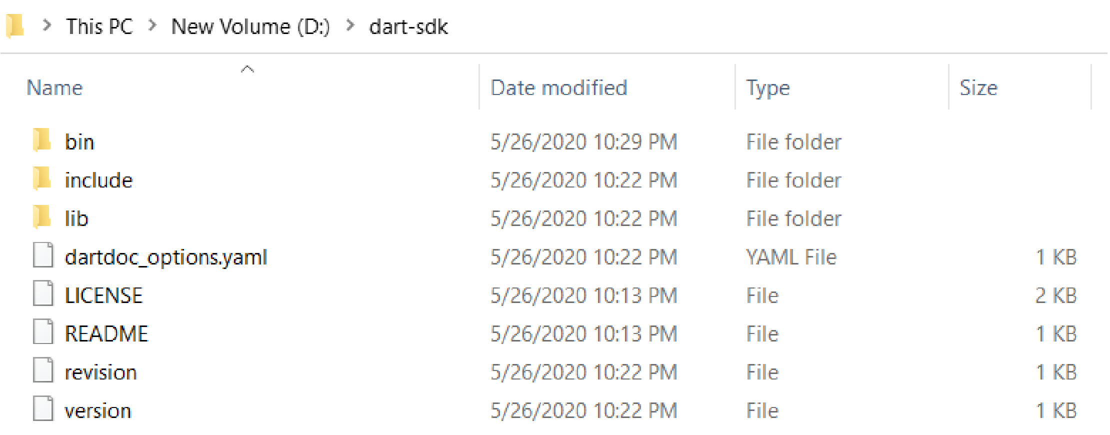
Stable channel

Stable channel builds are tested and approved for production use.

Version: 2.8.3 OS: Windows

Version	OS	Architecture	Downloads
2.8.3 (ref 9a618e5)	Windows	ia32	Dart SDK (SHA-256)
2.8.3 (ref 9a618e5)	Windows	x64	Dart SDK (SHA-256)
2.8.3 (ref 9a618e5)	---	---	API docs

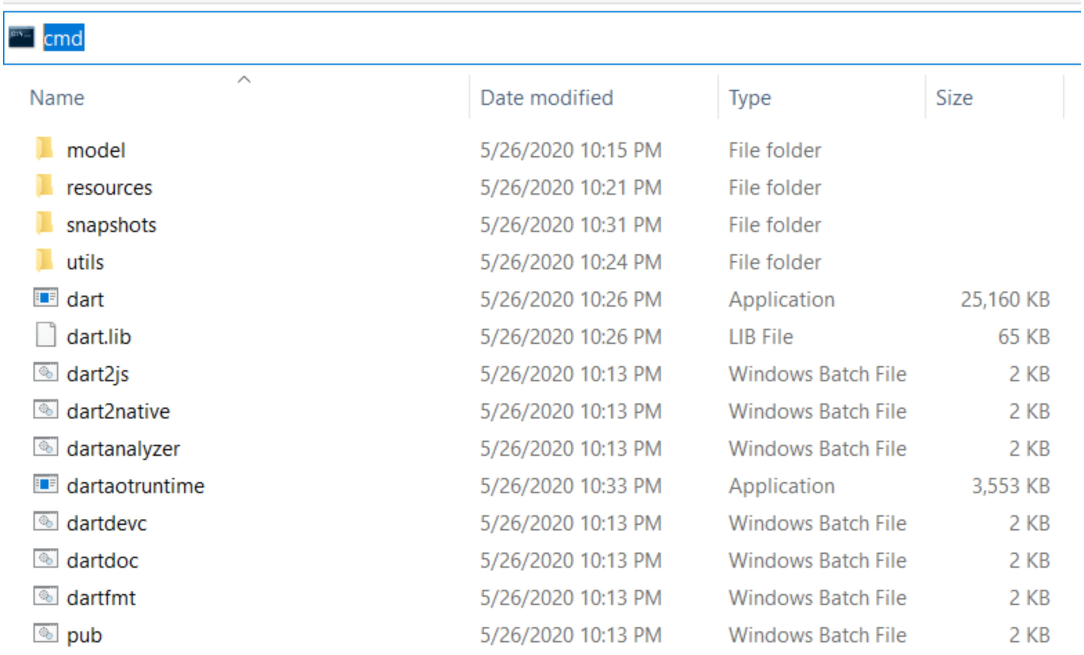
Click on DART SDK to download SDK for Windows 64-Bit Architecture. The download will start and a zip file will be downloaded. **Note:** To download SDK for any other OS select OS of your choice. **Step 2:** Extract the downloaded zip file. Extract the contents of downloaded zip file and after extracting contentsof zip file will be as shown:



This PC > New Volume (D:) > dart-sdk

Name	Date modified	Type	Size
bin	5/26/2020 10:29 PM	File folder	
include	5/26/2020 10:22 PM	File folder	
lib	5/26/2020 10:22 PM	File folder	
dartdoc_options.yaml	5/26/2020 10:22 PM	YAML File	1 KB
LICENSE	5/26/2020 10:13 PM	File	2 KB
README	5/26/2020 10:13 PM	File	1 KB
revision	5/26/2020 10:22 PM	File	1 KB
version	5/26/2020 10:22 PM	File	1 KB

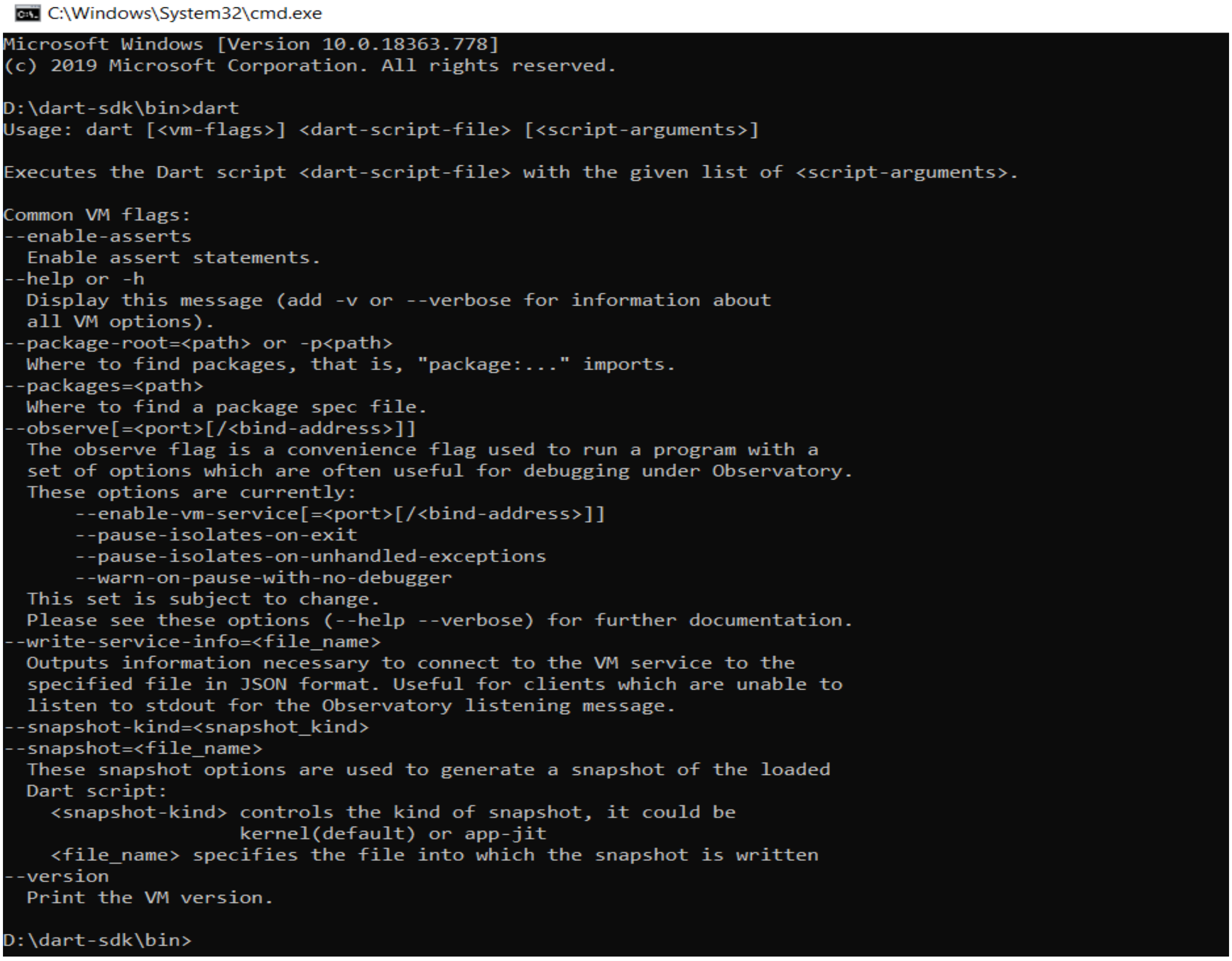
Step3: RunningDart.Nowopenbinfolderandtype“cmd”asgivenbelow:



cmd

Name	Date modified	Type	Size
model	5/26/2020 10:15 PM	File folder	
resources	5/26/2020 10:21 PM	File folder	
snapshots	5/26/2020 10:31 PM	File folder	
utils	5/26/2020 10:24 PM	File folder	
dart	5/26/2020 10:26 PM	Application	25,160 KB
dart.lib	5/26/2020 10:26 PM	LIB File	65 KB
dart2js	5/26/2020 10:13 PM	Windows Batch File	2 KB
dart2native	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartanalyzer	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartaotruntime	5/26/2020 10:33 PM	Application	3,553 KB
dartdevc	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartdoc	5/26/2020 10:13 PM	Windows Batch File	2 KB
dartfmt	5/26/2020 10:13 PM	Windows Batch File	2 KB
pub	5/26/2020 10:13 PM	Windows Batch File	2 KB

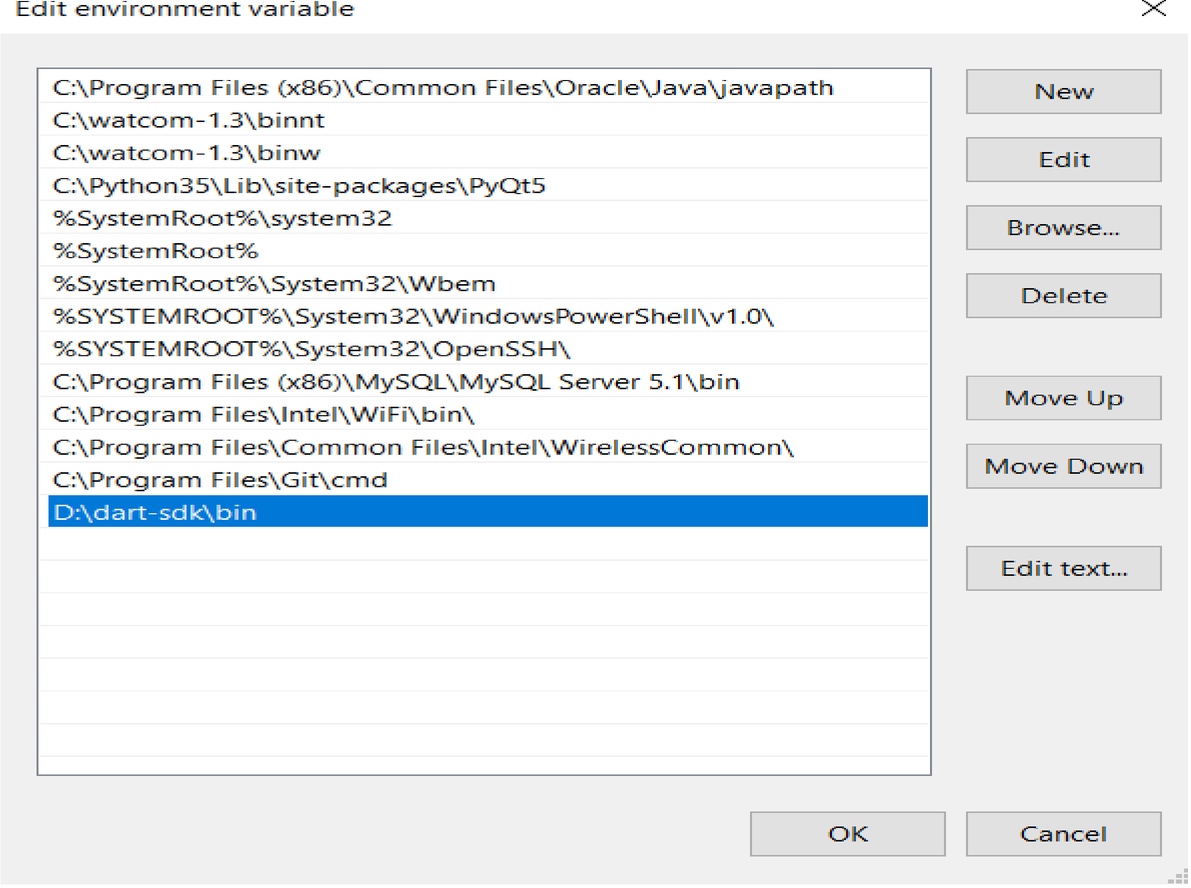
CommandPromptwillopenwithourdesiredpathofbinfolderandnowtypedart”.



```
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\dart-sdk\bin>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]

Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about
  all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:..." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a
  set of options which are often useful for debugging under Observatory.
  These options are currently:
      --enable-vm-service[=<port>[/<bind-address>]]
      --pause-isolates-on-exit
      --pause-isolates-on-unhandled-exceptions
      --warn-on-pause-with-no-debugger
  This set is subject to change.
  Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the
  specified file in JSON format. Useful for clients which are unable to
  listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
--snapshot=<file_name>
  These snapshot options are used to generate a snapshot of the loaded
  Dart script:
    <snapshot-kind> controls the kind of snapshot, it could be
                    kernel(default) or app-jit
    <file_name> specifies the file into which the snapshot is written
--version
  Print the VM version.

D:\dart-sdk\bin>
```

And now we are readyto use dart throughbin folder but setting up the path in environment variables will ease our task of Step3 and we can run dart from anywhere in the file system using command prompt.

Step 4: Setting up path in environment variables. OpenEnvironment Variables from advanced system settings and add Path in System Variables as depicted in image:

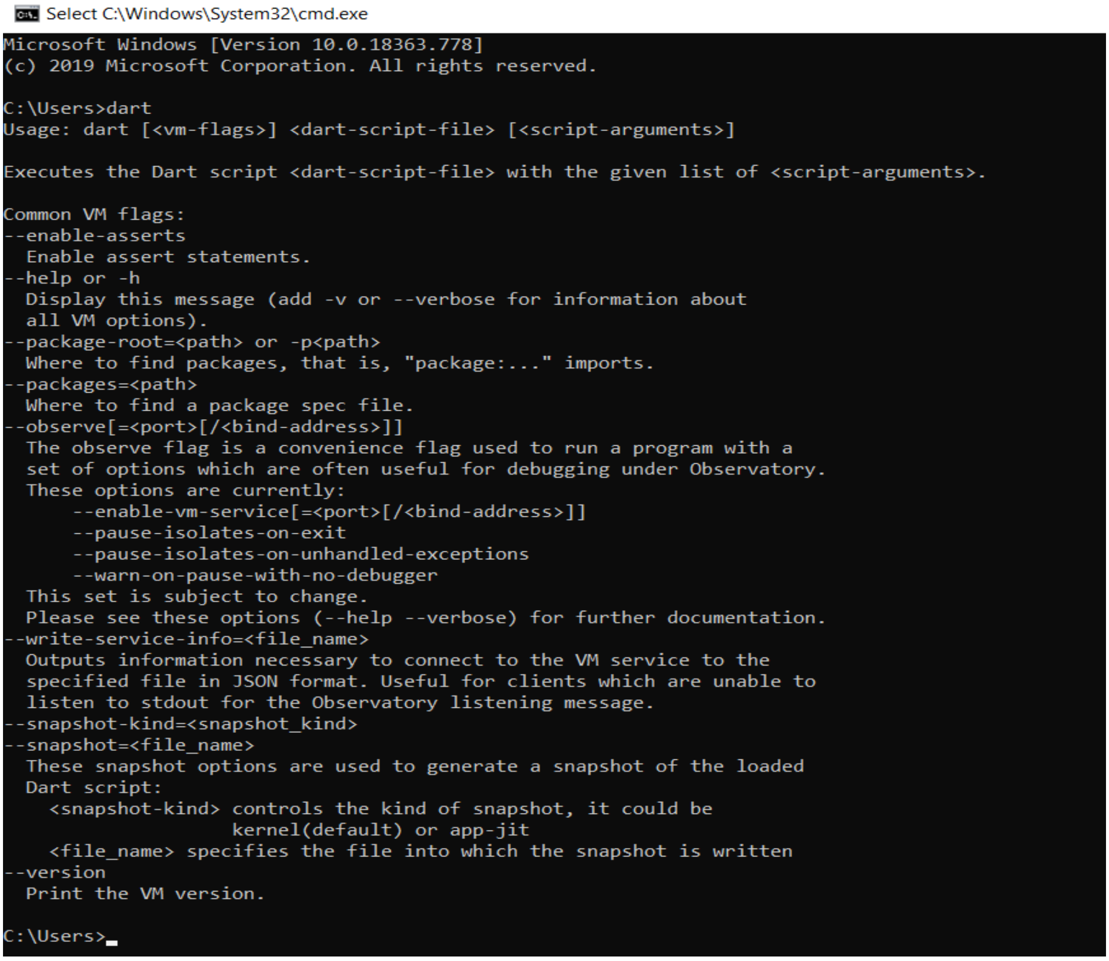


Edit environment variable

```
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\watcom-1.3\binnt
C:\watcom-1.3\binw
C:\Python35\Lib\site-packages\PyQt5
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin
C:\Program Files\Intel\WiFi\bin\
C:\Program Files\Common Files\Intel\WirelessCommon\
C:\Program Files\Git\cmd
D:\dart-sdk\bin
```

NowwearedonetouseDartfromanywherein thefilesystem.

Step5: Run DartUsingcmd



Select C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.18363.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users>dart
Usage: dart [<vm-flags>] <dart-script-file> [<script-arguments>]

Executes the Dart script <dart-script-file> with the given list of <script-arguments>.

Common VM flags:
--enable-asserts
  Enable assert statements.
--help or -h
  Display this message (add -v or --verbose for information about
  all VM options).
--package-root=<path> or -p<path>
  Where to find packages, that is, "package:..." imports.
--packages=<path>
  Where to find a package spec file.
--observe[=<port>[/<bind-address>]]
  The observe flag is a convenience flag used to run a program with a
  set of options which are often useful for debugging under Observatory.
  These options are currently:
      --enable-vm-service[=<port>[/<bind-address>]]
      --pause-isolates-on-exit
      --pause-isolates-on-unhandled-exceptions
      --warn-on-pause-with-no-debugger
  This set is subject to change.
  Please see these options (--help --verbose) for further documentation.
--write-service-info=<file_name>
  Outputs information necessary to connect to the VM service to the
  specified file in JSON format. Useful for clients which are unable to
  listen to stdout for the Observatory listening message.
--snapshot-kind=<snapshot_kind>
--snapshot=<file_name>
  These snapshot options are used to generate a snapshot of the loaded
  Dart script:
    <snapshot-kind> controls the kind of snapshot, it could be
                    kernel(default) or app-jit
    <file_name> specifies the file into which the snapshot is written
--version
  Print the VM version.

C:\Users>_
```

2)WriteasimpleDartprogramtounderstandthelanguagebasics. Ans)

```
        voidmain(){
          varfirstName="John";
```

```
  varlastName="Doe";
  print("Fullnameis$firstName$lastName");
}
```

Output:FullnameisJohnDoe

```
voidmain(){
intnum1=10;//declaringnumber1 int
num2 = 3; //declaring number2

//Calculation
intsum=num1+num2;


int diff= num1 - num2;
intmul=num1*num2;
doublediv=num1/num2;//Itisdoublebecauseitoutputsnumberwith decimal.

// displaying the output
print("Thesumis$sum");
print("The diff is $diff");
print("The mul is $mul");
print("The div is $div");
}
```

Output:

```
Thesumis13
The diff is 7
Themulis30
Thedivis 3.3333333333333335
```

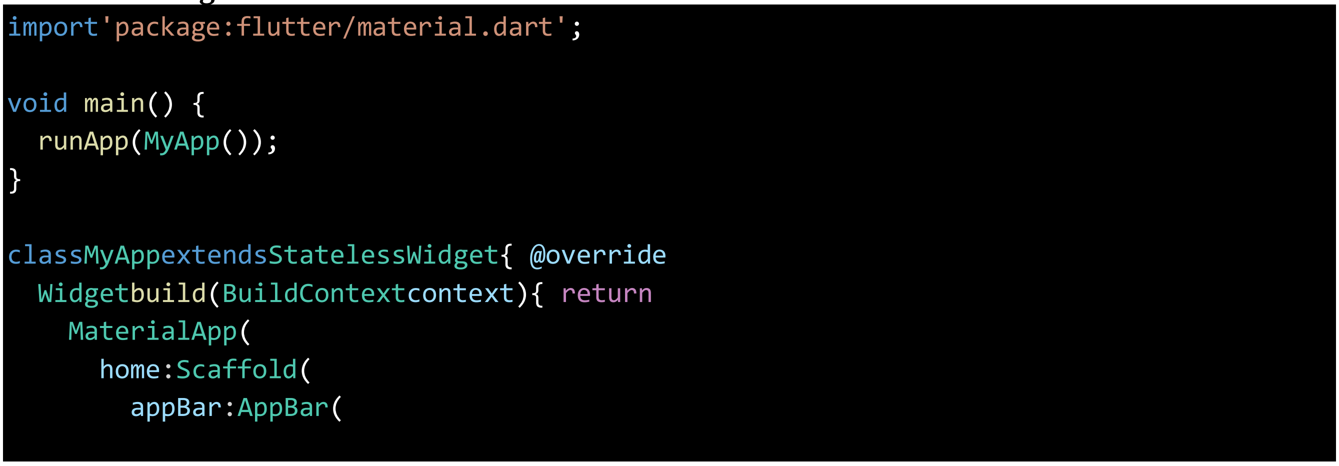
```
import'dart:io';

void main() {
  print("Enternumber:");
  int?number=int.parse(stdin.readLineSync()!);
  print("The entered number is ${number}");
}
```

Output:

```
Enternumber:
50
Theentered numberis50
```

3.ExplorevariousFlutterwidgets(Text,Image,Container,etc.).

TextWidget:



```
import'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

classMyAppextendsStatelessWidget{ @override
  Widgetbuild(BuildContextcontext){ return
    MaterialApp(
      home:Scaffold(
        appBar:AppBar(
          title:Text('TextWidgetExample'),
        ),
        body: Center(
          child:Text(
             'Hello,Flutter!',
             style: TextStyle(
              fontSize: 24.0,
               fontWeight:FontWeight.bold,
               color: Colors.blue,
             ),
           ),
        ),
      ),
    );
  }
}
```

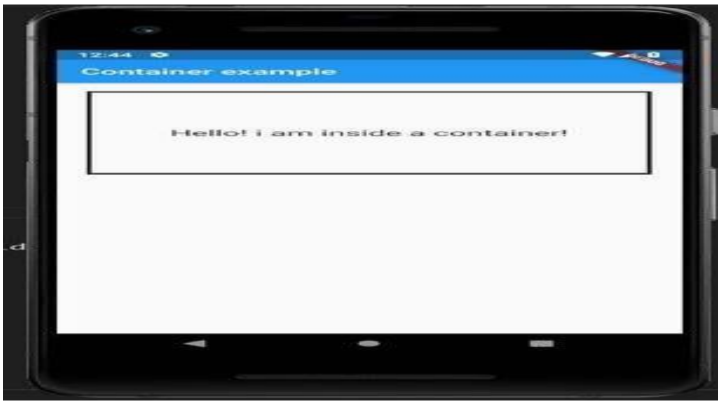
ImageWidget:

```
import'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

classMyAppextendsStatelessWidget{ @override
  Widgetbuild(BuildContextcontext){ return
    MaterialApp(
      home: Scaffold(
        appBar:AppBar(
          title:Text('ImagefromAssetsExample'),
        ),
        body:Center(
          child:Image.asset('images/ab.png'),
        ),
      ),
    );
  }
}
```

ContainterWidget:

```
import'package:flutter/material.dart';

voidmain()=>runApp(const MyApp());

class MyApp extends StatelessWidget {
constMyApp({Key?key}):super(key:key);

@override
Widgetbuild(BuildContextcontext){ return
        MaterialApp(
        home:Scaffold(
                appBar:AppBar(
                title:constText("Containerexample"),
                ),
                body:Container(
                height: 200,
                width:double.infinity,
                //color:Colors.purple, alignment:
                Alignment.center, margin:const
                EdgeInsets.all(20),
                padding:constEdgeInsets.all(30),
                decoration: BoxDecoration(
                        border:Border.all(color:Colors.black,width:3),
                ),
                child:constText("Hello!iaminsideacontainer!", style:
                        TextStyle(fontSize: 20)),
                ),

        ),
        );
}
}
```

Output:

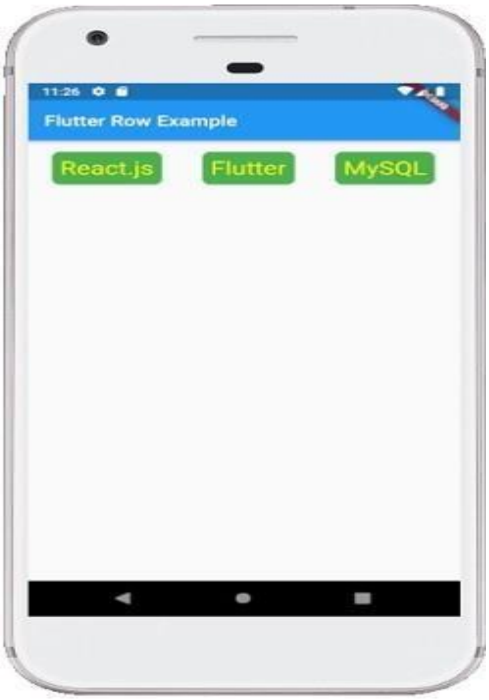


4) Implementdifferentlayoutstructuresusing Row,Column,andStackwidgets

Row Widget

```
import'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

classMyAppextendsStatelessWidget{ @override
  Widget build(BuildContext context) {
    returnMaterialApp(home:MyHomePage());
  }
}

classMyHomePageextendsStatefulWidget{
  @override
  _MyHomePageStatecreateState()=>_MyHomePageState();
}

class_MyHomePageStateextendsState<MyHomePage>{ @override
  Widgetbuild(BuildContextcontext){
    return Scaffold(
      appBar:AppBar(
        title:Text("FlutterRowExample"),
      ),
      body:Row(
          mainAxisAlignment:MainAxisAlignment.spaceEvenly,
          children: <Widget>[
            Container(
              margin:EdgeInsets.all(12.0),
              padding:EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                  borderRadius:BorderRadius.circular(8),color:Colors.green),
              child: Text(
                "React.js",
                style:TextStyle(color:Colors.yellowAccent,fontSize:25),
              ),
            ),
            Container(
              margin:EdgeInsets.all(15.0),
              padding:EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                  borderRadius:BorderRadius.circular(8),color:Colors.green),
              child: Text(
                "Flutter",
                style:TextStyle(color:Colors.yellowAccent,fontSize:25),
              ),
            ),
            Container(
              margin:EdgeInsets.all(12.0),
              padding:EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                  borderRadius:BorderRadius.circular(8),color:Colors.green),
              child: Text(
                "MySQL",
                style:TextStyle(color:Colors.yellowAccent,fontSize:25),
              ),
            )
          ]),
    );
  }
}
```

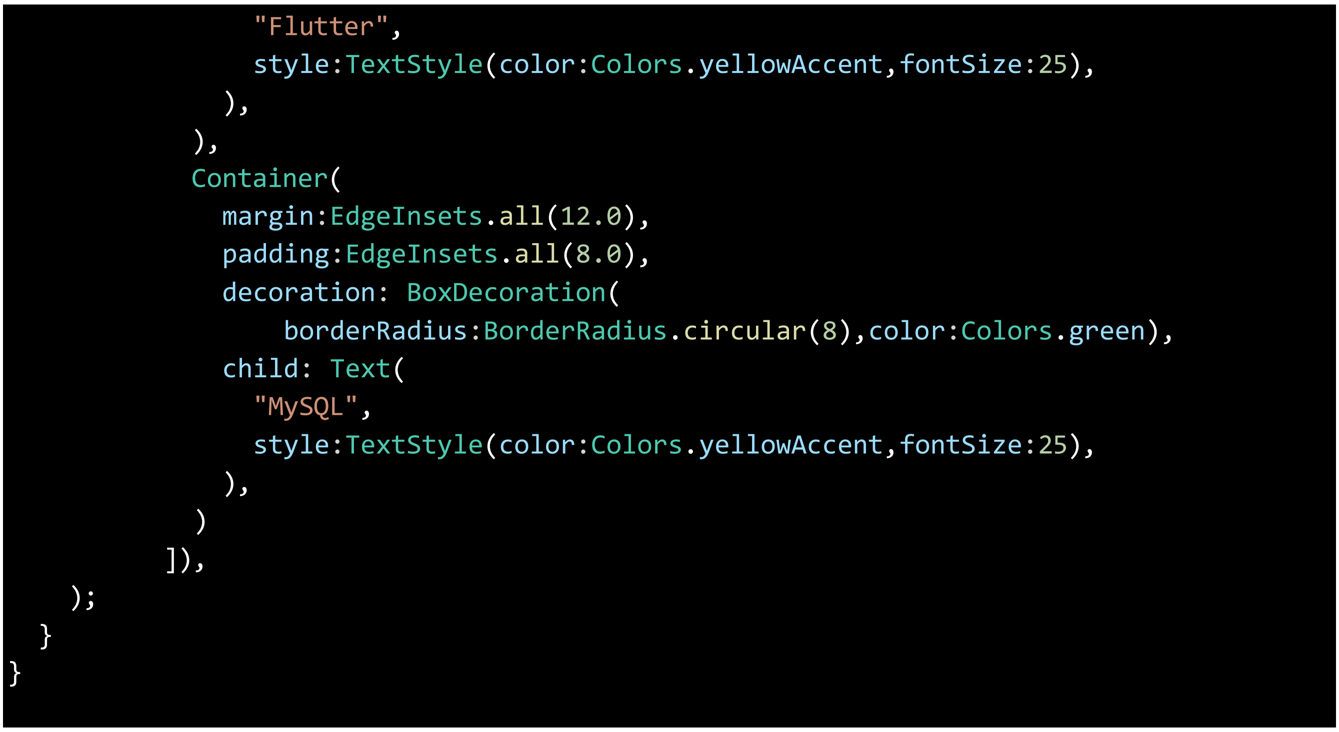
Output:



11:26
Flutter Row Example
React.js
Flutter
MySQL

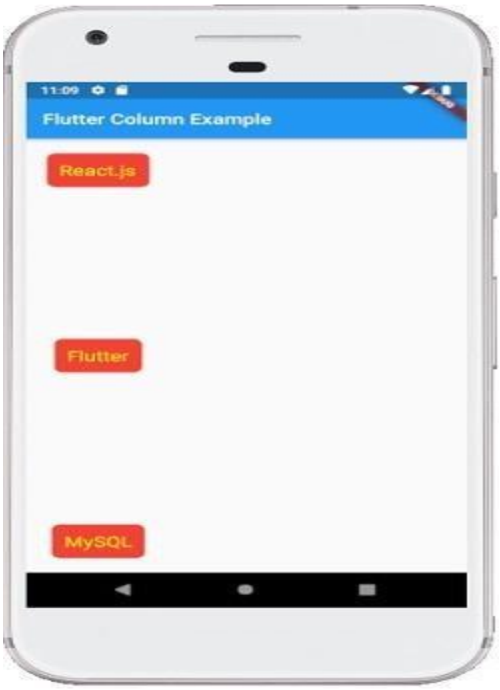
ColumnWidget:

```
import'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

classMyAppextendsStatelessWidget{
  @override
  Widget build(BuildContext context) {
    returnMaterialApp(home:MyHomePage());
  }
}

classMyHomePageextendsStatefulWidget{
  @override
  _MyHomePageStatecreateState()=>_MyHomePageState();
}

class_MyHomePageStateextendsState<MyHomePage>{
  @override
  Widgetbuild(BuildContextcontext){
    return Scaffold(
      appBar:AppBar(
        title:Text("FlutterColumnExample"),
      ),
      body:Column(
          mainAxisAlignment:MainAxisAlignment.spaceEvenly,
          children: <Widget>[
            Container(
              margin:EdgeInsets.all(12.0),
              padding:EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                  borderRadius:BorderRadius.circular(8),color:Colors.green),
              child: Text(
                "React.js",
                style:TextStyle(color:Colors.yellowAccent,fontSize:25),
              ),
            ),
            Container(
              margin:EdgeInsets.all(15.0),
              padding:EdgeInsets.all(8.0),
              decoration: BoxDecoration(
                  borderRadius:BorderRadius.circular(8),color:Colors.green),
              child: Text(
```



```
        "Flutter",
        style:TextStyle(color:Colors.yellowAccent,fontSize:25),
       ),
      ),
      Container(
       margin:EdgeInsets.all(12.0),
       padding:EdgeInsets.all(8.0),
       decoration: BoxDecoration(
         borderRadius:BorderRadius.circular(8),color:Colors.green),
       child: Text(
        "MySQL",
        style:TextStyle(color:Colors.yellowAccent,fontSize:25),
       ),
      )
     ]),
  );
 }
}
```

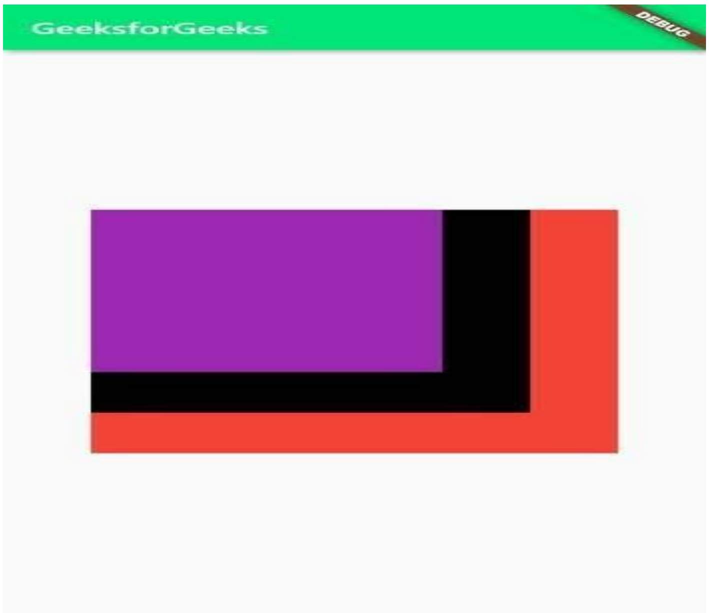
Output:



StackWidget:
import'package:flutter/material.dart';
void main() {
runApp(MaterialApp(

```
        home:Scaffold(
            appBar:AppBar(
                title: Text('GeeksforGeeks'),
                backgroundColor:Colors.greenAccent[400],
            ),
            //Appar
            body:center(
                child:SizedBox(
                width:300,
                height:300,
                child:Center(
                    child: Stack(
                    children:<Widget>[
                        Conainer(
                        width:300,
                        height: 300,
                        color:Colors.red,
                        ),//Container
                        Container(
                        width: 250,
                        height:250,
                        color:Colors.black,
                        ),//Container

                        Container(
                        height:200,
                        width:200,
                        color:Colors.purple,
                        ),//Container
                    ],//<Widget>[]
                    ),//Stack
                ),//Center
                ),//SizedBox
            )//Center
            )//Scaffold
        )//MaterialApp
    );
}
```

Output:

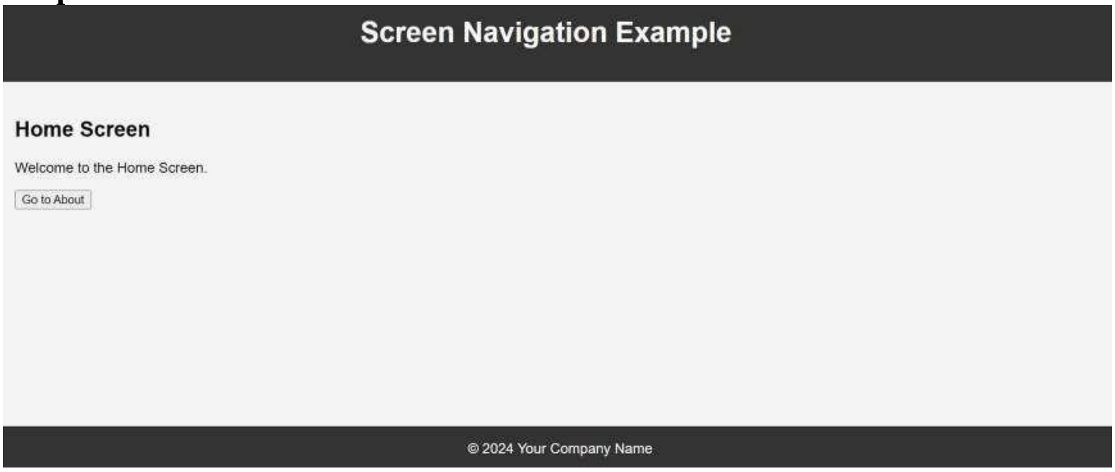


5) SetupnavigationbetweendifferentscreensusingNavigator.

Ans)

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<title>ScreenNavigationExample</title>
<style>
        body{
            font-family:Arial,sans-serif;
            margin: 0;
            padding:0;
            background-color:#f4f4f4;
        }

        header {
            background-color:#333;
            color: #fff;
            text-align:center;
            padding: 1em;
        }
        main {
            max-width:1200px;
            margin: 0 auto;
            padding:20px;
        }

        section {
            display:none;
        }

        footer{
            background-color:#333;
            color: #fff;
            text-align:center;
            padding: 1em;
            position: fixed;
            bottom: 0;
            width:100%;
        }

        .active{
            display:block;
        }
</style>
</head>
<body>
<headerclass="jumbotrontext-center">
<h1>ScreenNavigationExample</h1>
</header>

<main>
<sectionid="home"class="active">
<h2>HomeScreen</h2>
<p>WelcometotheHomeScreen.</p>
<buttononclick="navigateTo('about')">GotoAbout</button>
</section>

<sectionid="about">
<h2>About Screen</h2>
<p>ThisistheAbout Screen.</p>
<buttononclick="navigateTo('home')">GotoHome</button>
</section>
</main>

<footerclass="bg-darktext-lighttext-centerpy-3mt-
5">&copy; 2024 Your Company Name
</footer>

<script>
        functionnavigateTo(screenId){
            // Hide all sections
            document.querySelectorAll('section').forEach(section=>{
                section.classList.remove('active');
            });
            // Show the selected section
            document.getElementById(screenId).classList.add('active');
        }
</script>
</body>
</html>
```

Output:

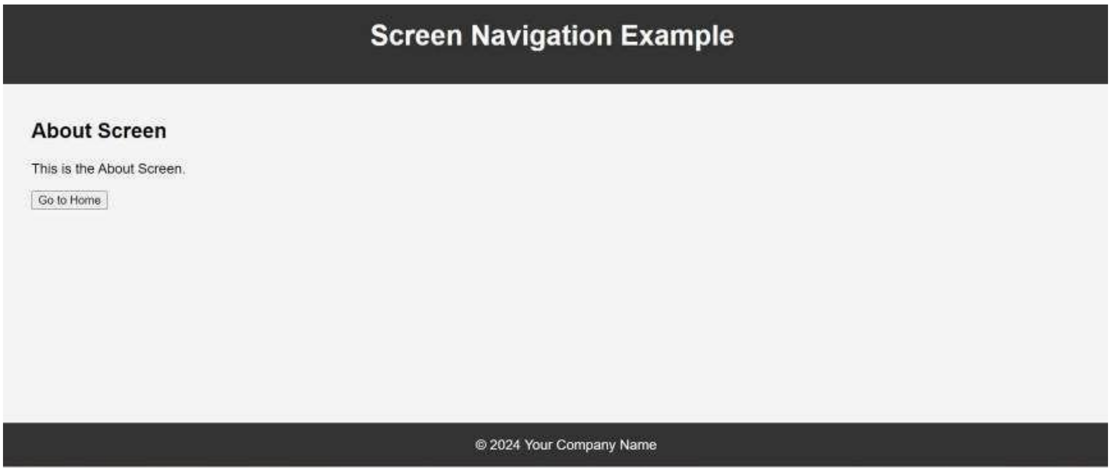


Screen Navigation Example

Home Screen

Welcome to the Home Screen.

© 2024 Your Company Name



Screen Navigation Example

About Screen

This is the About Screen.

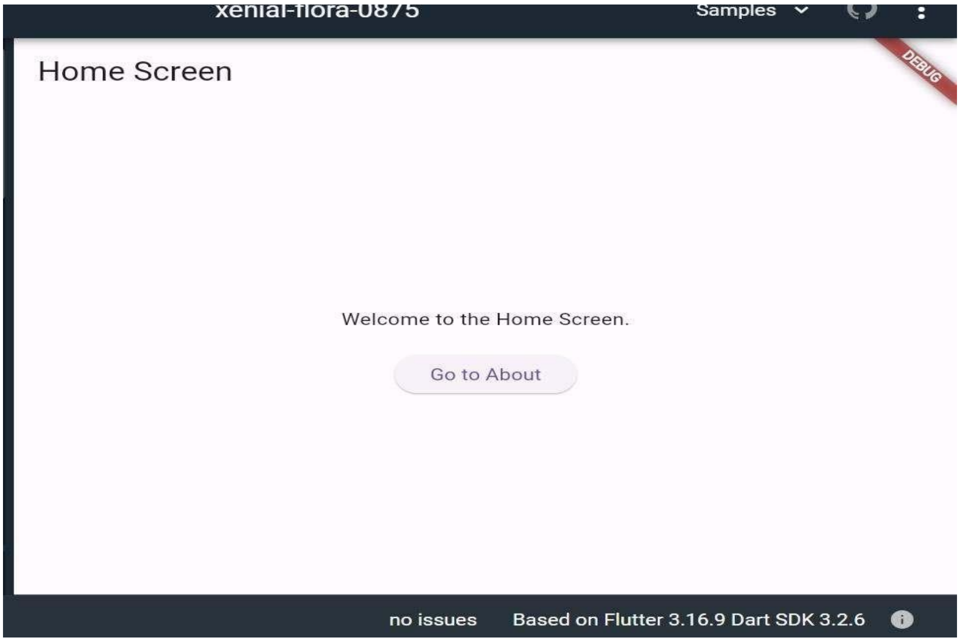
© 2024 Your Company Name

6) Implementnavigationwithnamedroutes.

Ans)

```
import'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}

classMyAppextendsStatelessWidget{
  @override
  Widgetbuild(BuildContextcontext){
    return MaterialApp(
      title:'NamedRoutesNavigationExample',
      initialRoute: '/',
      routes:{
        '/': (context) => HomeScreen(),
        '/about':(context)=>AboutScreen(),
      },
    );
  }
}

classHomeScreenextendsStatelessWidget{
  @override
  Widgetbuild(BuildContextcontext){
    return Scaffold(
      appBar:AppBar(
        title: Text('HomeScreen'),
      ),
      body: Center(
        child:Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'WelcometotheHomeScreen.',
            ),
            SizedBox(height:20),
            ElevatedButton(
            onPressed: () {
                Navigator.pushNamed(context,'/about');
              },
              child:Text('GotoAbout'),
          ),],
        ),
      ),
    );
  }
}

classAboutScreenextendsStatelessWidget{
  @override
  Widgetbuild(BuildContextcontext){
    return Scaffold(
      appBar:AppBar(
        title:Text('AboutScreen'),
      ),
      body: Center(
        child:Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'ThisistheAboutScreen.',
            ),
            SizedBox(height:20),
            ElevatedButton(
            onPressed: () {
              Navigator.pop(context);
             },
             child:Text('GobacktoHome'),
            ),
          ],
        ),
      ),
    );
  }
}
```

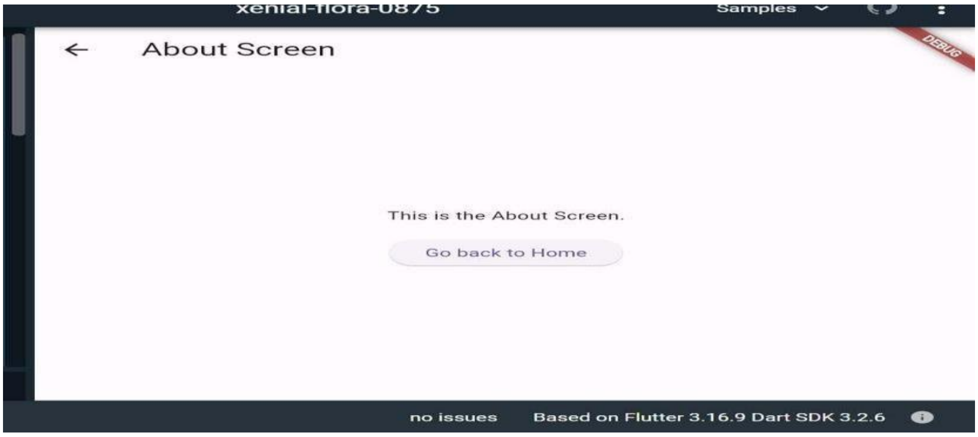
Output:



Home Screen

Welcome to the Home Screen.

Go to About



About Screen

This is the About Screen.

Go back to Home

7) UseFlutter'sdebuggingtoolstoidentifyandfixissues.

Ans) Flutterprovidesaset ofdebuggingtoolsthatcanhelp you identifyand fix issues in your app. Here's a step-by-step guide on how to use these tools:

1. FlutterDevTools:
Run yourapp withtheflutterruncommand.
OpenDevToolsbyrunningthefollowingcommandinyourterminal: bash

```
flutterpubglobalactivatedevtools
flutter pub global run devtools
```

Openyour appin a Chrome browserand connectit toDevTools by clicking on the "Open DevTools" button in the terminal or by navigating to http://127.0.0.1:9100/.
DevToolsprovidestabslikeInspector,Timeline,Memory,andmore.

2. Flutter Inspector:
UsetheFlutterInspectorinyourintegrateddevelopmentenvironment(IDE)like Android Studio or Visual Studio Code.
ToggletheInspectorinAndroidStudiowiththeshortcutAlt+Shift+D
(Windows/Linux) or Option + Shift + D (Mac).
Inspectthewidgettree,modifywidgetproperties,andobservewidget relationships.

3. HotReload:
Leverage Hot Reload to see the immediate effect of code changes without restarting the entire app.
PressRintheterminalorusethe"Hot Reload"buttonin yourIDE.

4. DebuggingwithBreakpoints:
Set breakpoints in your codeto pause executionand inspect variables.
Usethedebugger inyourIDEtostepthroughcodeandidentifyissues.

5. Logging:
Utilizetheprintfunctiontologmessagestotheconsole.

```
print('Debuggingmessage');
```

Viewlogsintheterminalorthe"Logs"tabinDevTools.

6. DebugPaint:
Enable debug paint to visualize the layout and rendering of widgets.
UsethedebugPaintSizeEnabledanddebugPaintBaselinesEnabledflags.

```
voidmain(){
  debugPaintSizeEnabled=true;//Showsboundingboxesofwidgets
  runApp(MyApp());
}
```

7. MemoryProfiling:
Use the "Memory" tab in DevTools to analyze memory usage and identify potential memory leaks.
Monitorobjectallocationsanddeallocations.

8. PerformanceProfiling(Timeline):
Analyzeappperformanceusingthe"Timeline"tabinDevTools. Identify UI jank, slow frames, and performance bottlenecks.

9. FlutterDriverTests:
WriteautomatedUItestsusingFlutterDriver.
Simulateuser interactionsandvalidatethecorrectnessofyour UI.

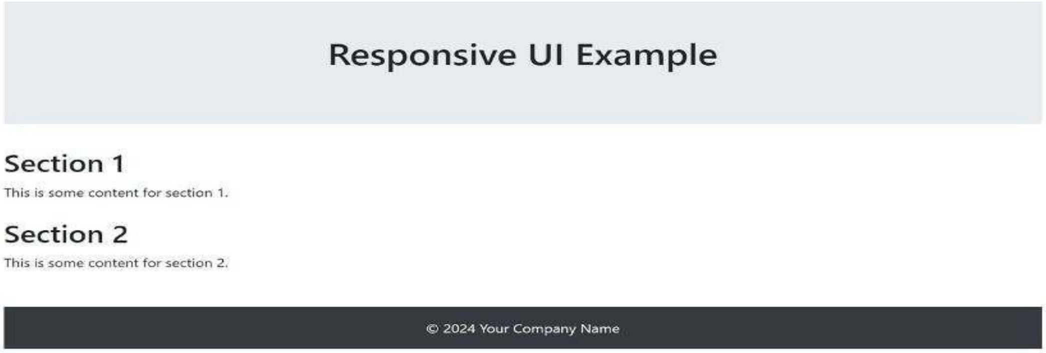
8) DesignaresponsiveUIthatadaptstodifferentscreensizes.

Ans)

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<title>ResponsiveUIExample</title>
</head>
<body>
<divclass="container">
<headerclass="jumbotrontext-center">
<h1>ResponsiveUIExample</h1>
</header>

<main>
<sectionclass="mb-4">
<h2>Section1</h2>
<p>Thisissomecontentforsection1.</p>
</section>

<sectionclass="mb-4">
<h2>Section2</h2>
<p>Thisissomecontentforsection2.</p>
</section>
</main>

<footerclass="bg-darktext-lighttext-centerpy-3mt-
5">&copy; 2024 Your Company Name
</footer>
</div>
```

```
<!--BootstrapJSanddependencies(jQuery) -->
<scriptsrc="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"></
script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></
script>
</body>
</html>
```

Output:



Responsive UI Example

Section 1

This is some content for section 1.

Section 2

This is some content for section 2.

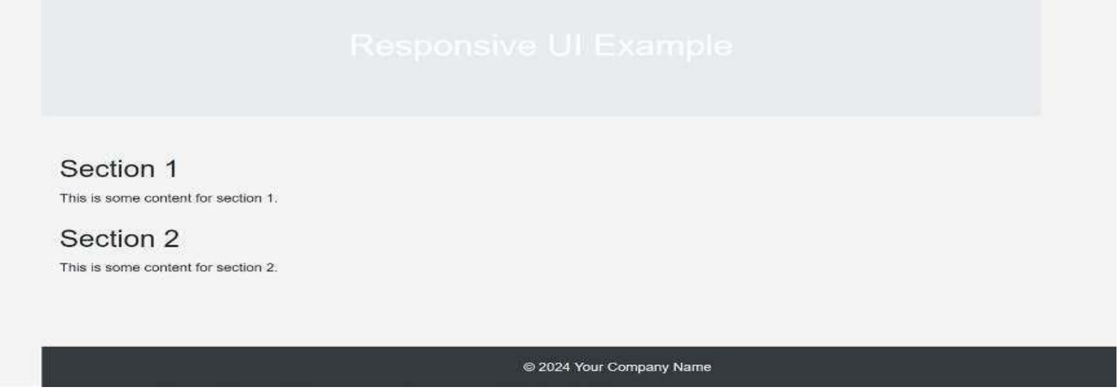
© 2024 Your Company Name

9) Implementmediaqueriesandbreakpointsforresponsiveness.

```
Ans)<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width,initial-scale=1.0">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
    body{
      font-family: Arial,
      sans-serif;margin:0;
      padding:0;
      background-color:#f4f4f4;
    }

    header {
      background-color:
      #333;color: #fff;
      text-align:
      center;
      padding:
      1em;
    }

    main {
      max-width:
      1200px;
      margin: 0
      auto;padding:
      20px;
    }

    section{
      margin-bottom:20px;
    }

    footer{
      background-color:
      #333;color: #fff;
      text-align:
      center;
      padding:
      1em;
      position:
      fixed;
      bottom:0;
      width:100%;
    }

    @mediaonlyscreenand(max-width: 768px)
      {main {
        padding:10px;
      }

      footer{
        position:static;
      }
    }
</style>
<title>ResponsiveUIExample</title>
</head>
<body>
<divclass="container">
<headerclass="jumbotrontext-center">
<h1>ResponsiveUIExample</h1>
</header>

<main>
<sectionclass="mb-4">
<h2>Section1</h2>
<p>Thisissomecontentforsection1.</p>
</section>

<sectionclass="mb-4">
<h2>Section2</h2>
<p>Thisissomecontentforsection2.</p>
```

```
</section>
</main>

<footerclass="bg-darktext-lighttext-centerpy-3
mt-5">&copy; 2024 Your Company Name
</footer>
</div>

<!--BootstrapJSanddependencies(jQuery) -->
<scriptsrc="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"></s
cript>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js
"></ script>
</body>
</html>
```

Output:



Responsive UI Example

Section 1

This is some content for section 1.

Section 2

This is some content for section 2.

© 2024 Your Company Name

10) Createcustomwidgetsforspecific UIelements.

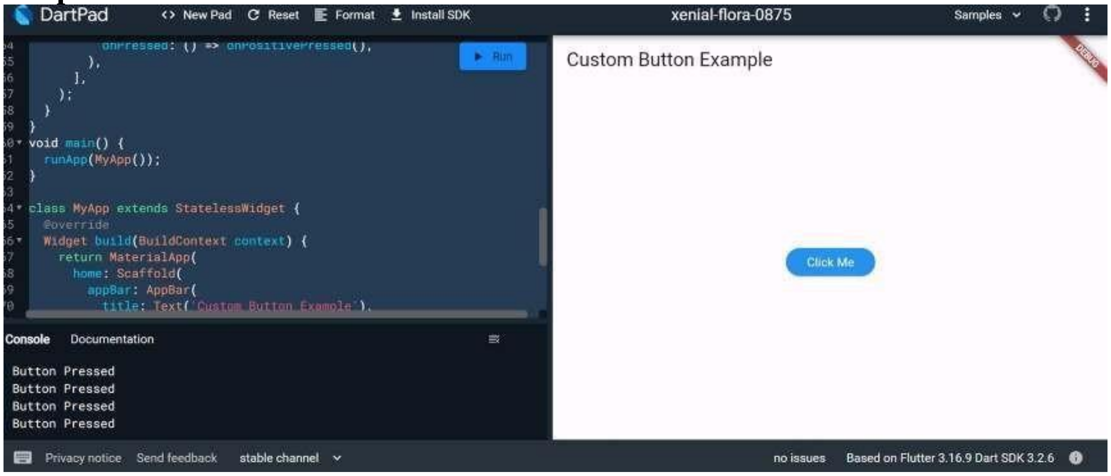
Ans)

```
import 'package:flutter/material.dart';
class CustomButton extends
StatelessWidget {final String text;
final Function
  onPressed;final
  ColorbuttonColor;
  final Color
  textColor;
  CustomButton({
       requiredthis.text,
       required
       this.onPressed,
this.buttonColor=Colors.blue,this.textColor=Colors.white,
      });
      @override
      Widgetbuild(BuildContext
       context) {return
       ElevatedButton(
       onPressed: ()
       =>onPressed(),style:
        ButtonStyle(
         backgroundColor: MaterialStateProperty.all<Color>(buttonColor),
         foregroundColor: MaterialStateProperty.all<Color>(textColor),
        ),
        child:Text(text),
       );
      }
     }


     class CustomAlertDialog extends
      StatelessWidget{finalStringtitle;
      final String message;
      final String
      positiveButtonText; final
      StringnegativeButtonText;
      final Function
      onPositivePressed; final
      Function
      onNegativePressed;
      CustomAlertDialog({
      required
       this.title,
       required
       this.message,
       required
       this.positiveButtonText,
       required
       this.negativeButtonText,
       required
       this.onPositivePressed,
       required
       this.onNegativePressed,
      });
      @override
```

```
  Widget build(BuildContext
   context){returnAlertDialog(
   title: Text(title),
    content:
    Text(message),
    actions:
    <Widget>[
    CustomButton(
      text:negativeButtonText,
      onPressed:()=>onNegativePressed(),
     ),
     CustomButton(
      text:positiveButtonText,
      onPressed:()=>onPositivePressed(),
     ),
    ],
   );


  }
}
voidmain(){
  runApp(My
  App());
}

class MyApp extends
  StatelessWidget {@override
  Widget build(BuildContext
  context) {return MaterialApp(
    home:
     Scaffold(
     appBar:
     AppBar(
      title:Text('CustomButtonExample'),
     ),
     body: Center(
      child:
      CustomButton
      (text: 'Click
      Me',
       onPressed:()
       {
        //Handlebutton
        press
        print('Button
        Pressed');
```

```
            },
          ),
        ),
      ),
    );
  }
}
```

Output:



```
            onPressed: () => onPositivePressed(),
          ),
        ],
      );
    }
}
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Custom Button Example'),
```

```
Button Pressed
Button Pressed
Button Pressed
Button Pressed
```

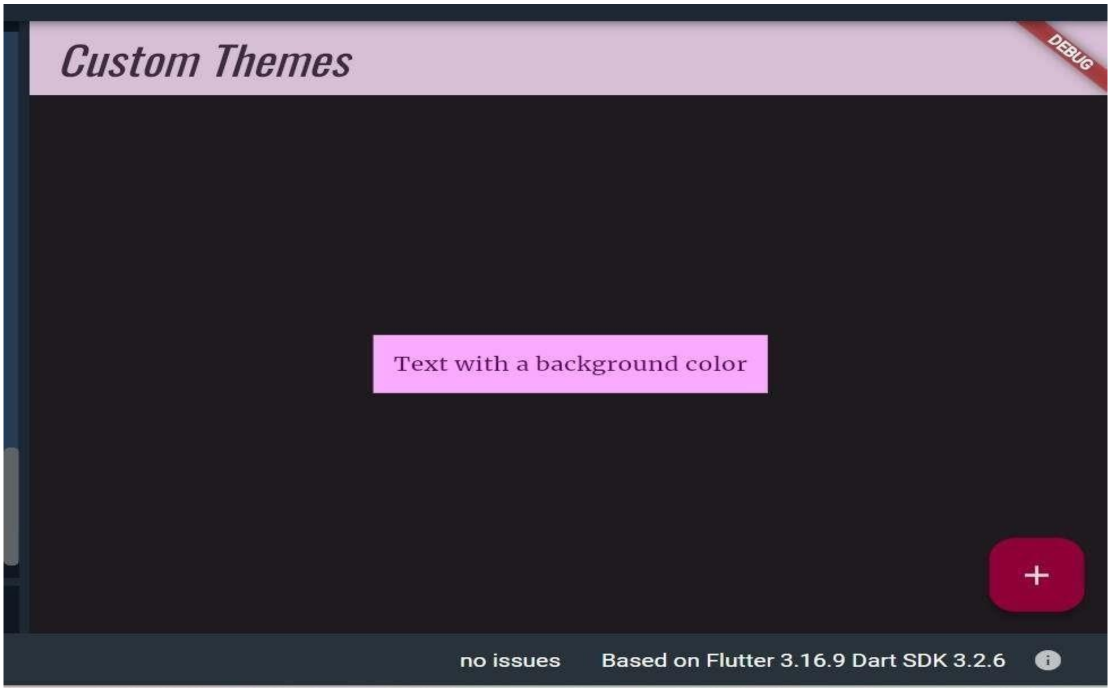
11) Applystylingusingthemesandcustom

styles.

Ans)

```
import'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

void main() {
  runApp(const
  MyApp());
}

class MyApp extends
  StatelessWidget{const
  MyApp({super.key});

  @override
  Widget build(BuildContext
    context){constappName=
    'CustomThemes';

    return
      MaterialApp(
      title:appName,
      theme:
      ThemeData(
      useMaterial3:
      true,

        // Define the default brightness and
        colors.colorScheme:
        ColorScheme.fromSeed(seedColor:
        Colors.purple,
          //TRYTHIS:Changeto"Brightness.light"
          //          andseethatallcolorschange
          //          tobettercontrastalight
          background.brightness:
          Brightness.dark,
        ),

        //Definethedefault`TextTheme`.Usethistospecifythedefault
        //textstylingforheadlines,titles,bodiesoftext,and
        more.textTheme: TextTheme(
          displayLarge:constTextStyle(
            fontSize: 72,
            fontWeight:FontWeight.bold,
          ),
          //TRYTHIS:ChangeoneoftheGoogleFonts
          //          to"lato", "poppins",or "lora".
          //          Thetitleuses"titleLarge"
          //          andthemiddletextuses
          "bodyMedium".titleLarge:
          GoogleFonts.oswald(
            fontSize:30,
            fontStyle:FontStyle.italic,
          ),
          bodyMedium:
          GoogleFonts.merriweather(),
          displaySmall:
          GoogleFonts.pacifico(),
        ),
      ),
      home: const
        MyHomePage(title:
        appName,
      ),
    );
  }
}

class MyHomePage extends
  StatelessWidget{finalStringtitle;

  const MyHomePage({super.key, required

  this.title});@override
  Widgetbuild(BuildContext
    context){returnScaffold(
    appBar:
        AppBar(
        title:
        Text(title,
            style:
                  Theme.of(context).textTheme.titleLarge!.copyWi
                  th(color:
                  Theme.of(context).colorScheme.onSecondary,
                )),
        backgroundColor:Theme.of(context).colorScheme.secondary,
      ),
      body:
        Center(
        child:
        Container(
          padding:const
            EdgeInsets.symmetric(
            horizontal: 12,
            vertical: 12,
          ),
          color:
          Theme.of(context).colorScheme.primary,
          child:Text(
            'Textwithabackgroundcolor',
            //TRYTHIS:ChangetheTextvalue
            //          orchangetheTheme.of(context).textTheme
            //          to"displayLarge"or"displaySmall".
            style:
                  Theme.of(context).textTheme.bodyMedium!.copyWith(
                  color: Theme.of(context).colorScheme.onPrimary,
                ),
          ),
        ),
      ),
      floatingActionButton:Theme(
        data:Theme.of(context).copyWith(
          //TRYTHIS:ChangetheseedColorto"Colors.red" or
          //          "Colors.blue".


          colorScheme:
            ColorScheme.fromSeed(
            seedColor: Colors.pink,
            brightness:Brightness.dark,
          ),
        ),
        child:
          FloatingActionButton
          (onPressed: () {},
          child:const Icon(Icons.add),
        ),
      ),
    );
  }
}
```

Output:



Custom Themes

Text with a background color

no issues Based on Flutter 3.16.9 Dart SDK 3.2.6

12) AddanimationstoUIelementsusingFlutter'sanimationframework.

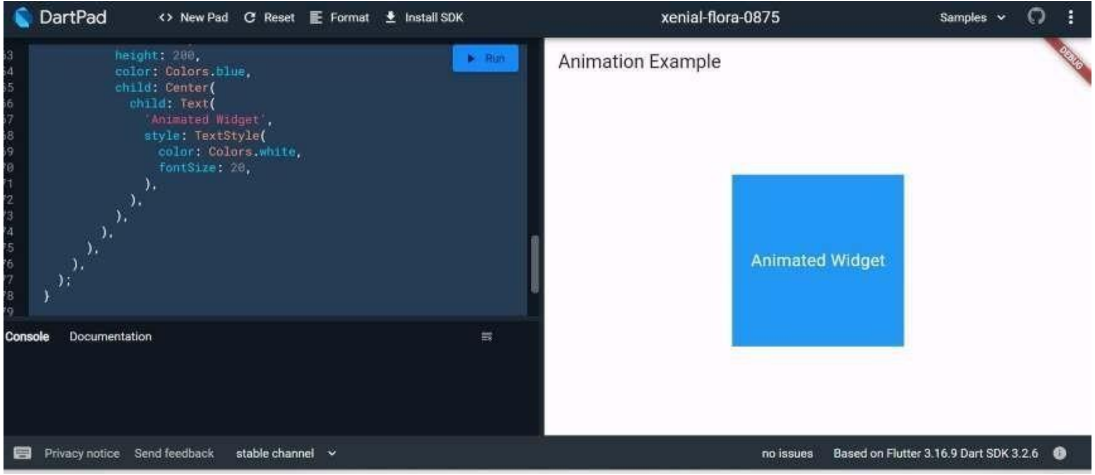
Ans)

```
 import'package:flutter/material.dart';

voidmain(){
  runApp(My
  App());
}

class MyApp extends
  StatelessWidget {@override
  Widget build(BuildContext
  context) {return MaterialApp(
      title: 'Animation
      Example',theme:
      ThemeData(
      primarySwatch:
      Colors.blue,
      ),
      home:MyAnimatedWidget(),
    );
  }
}

classMyAnimatedWidgetextends
```

```
 StatefulWidget{@override
 _MyAnimatedWidgetStatecreateState()=>_MyAnimatedWidgetState();
}

class_MyAnimatedWidgetStateextendsState<MyAnimatedWidget>
  with SingleTickerProviderStateMixin {
 late AnimationController
 _animationController;late
 Animation<double>_opacityAnimation;

 @override
 void
 initState()
  {
  super.initS
  tate();

  //CreateanAnimationControllerwithadurationof1second
  _animationController =
   AnimationController(vsync:this,
   duration: Duration(seconds: 1),
  );

  //CreateaTweentoanimateopacityfrom0.0to1.0
  _opacityAnimation=Tween<double>(begin:0.0,end:1.0).animate( CurvedAnimation(
    parent:
    _animationController,
    curve:
    Curves.easeInOut,
   ),
  );

  //Startthe animation
  _animationController.forward();
 }

 @override
 Widgetbuild(BuildContext
  context){returnScaffold(
  appBar:AppBar(
    title:Text('AnimationExample'),
   ),
   body:Center(
    child:FadeTransition(
     opacity:
```

```
     _opacityAnimation,
     child: Container(
     width: 200,
       height:200,
       color:
       Colors.blue,
       child:
       Center(
       child: Text(
           'Animated
           Widget',style:
           TextStyle(
           color:
           Colors.white,
           fontSize: 20,
           ),
         ),
       ),
     ),
   ),
  ),
 );
}

@override
voiddispose(){
 _animationController.dispose();
 super.dispose();
}
}
```

Output:



```
            height: 200,
            color: Colors.blue,
            child: Center(
              child: Text(
                'Animated Widget',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 20,
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }
```