**Practical No 20**

**Aim: Write a Dart program using Android studio**

To start the dart practical use online editor click on following link <https://dartpad.dev/>

1. **Introduction to Variable, Array, Control Flow Source Code:**

void main()

{

// introduction to variable and array var fname='Uday';

var lname='Futak';

var fruits=['Mango','Apple','Jack-fruit','Pineapple']; var assoArray= {

'friend': 'Rudra',

'url': 'https://vivaSchoolofmca.com'

};

print("Hello world"); print(fname+' '+lname); print(fruits); print(fruits[3]); print(assoArray); print(assoArray['friend']); for(var fruit in fruits)

{

print(fruit);

}

for(int i=1;i<=10;i++)

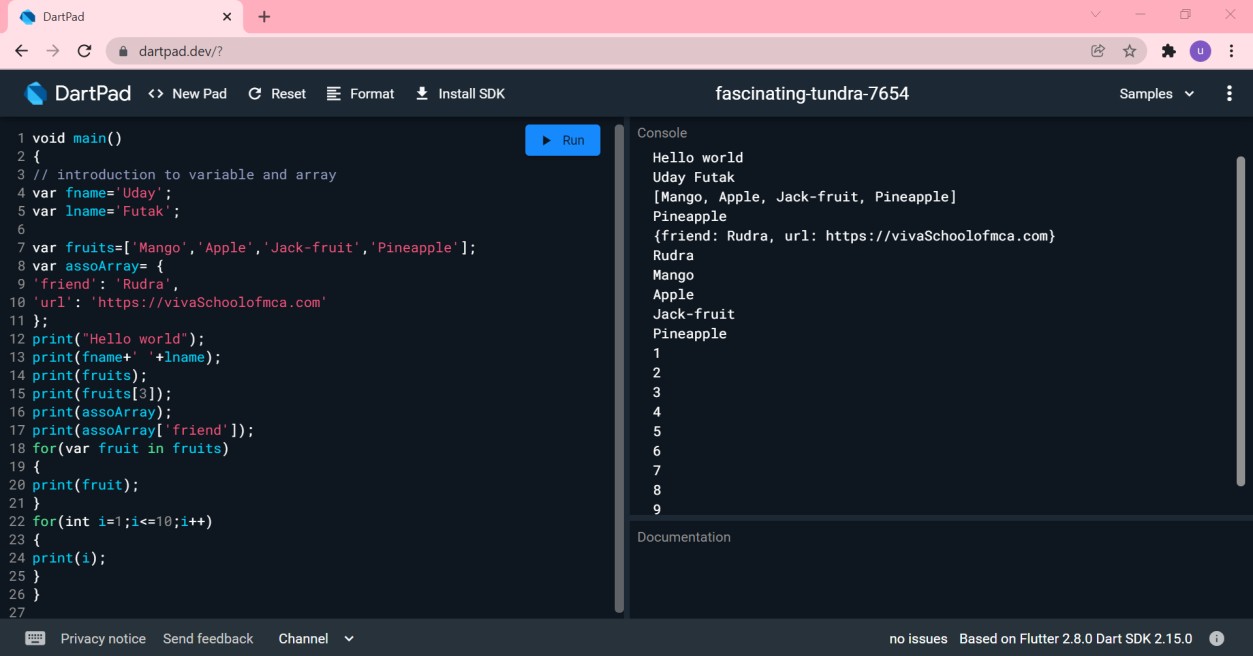
{

print(i);

}

}

**Output:**



1. **Introduction to Function Source Code:**

//introduction to function void main()

{

int result=squareit(6);

print('Square of number is = $result');

}

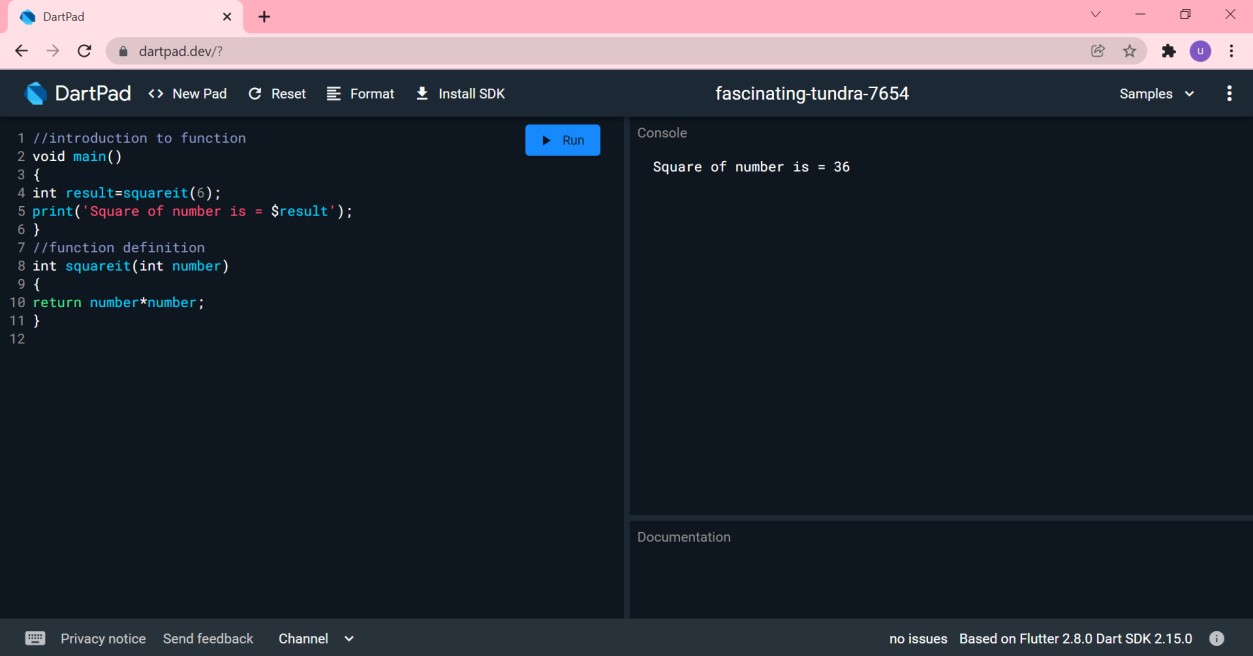
//function definition

int squareit(int number)

{

return number\*number;

}

**Output:**

1. **Introduction to Class, Constructor,Name Constructor, Getter and Setter methods. Source Code:**

class Student { String name; int age=0;

//constructor Student(this.name, this.age);

Student.construtor1(this.name); //named constructor with single parameter Student.construtor2(this.name,this.age);//named constructor with TWO parameter

String get studName { return name;

}

set studName(String name)

{

this.name = name;

}

set studAge(int age)

{

if(age<= 0)

{

print("Age should be greater than 5");

}

else

{

this.age = age;

}

}

int get studAge { return age;

}

}

void main() {

Student s1 = Student("Ronil Patil",50);

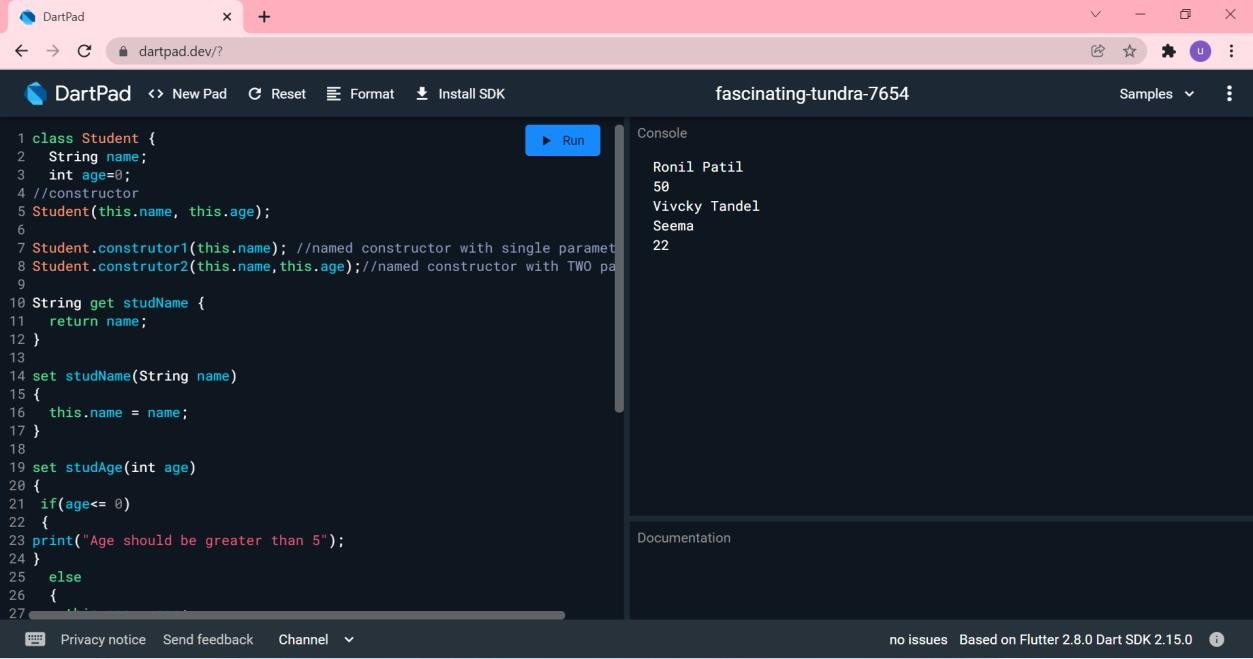
Student s2= Student.construtor1("Vivcky Tandel"); Student s3=Student.construtor2("Seema", 22); print(s1.studName);

print(s1.studAge); print(s2.studName);

print(s3.studName); print(s3.studAge);

}

**Output:**



1. **Single Inheritance Source Code:**

class PersonInfo

{

String name; int age=0;

//constructor PersonInfo(this.name, this.age);

//setter and getter String get personName

{

return name;

}

set personName(String name)

{

this.name = name;

}

int get personAge{ return age;

}

set personAge(int age)

{

this.age=age;

}

//method

void printData()

{

print("Name=$personName"); print("Age=$personAge");

}

}

class StudentInfo extends PersonInfo

{

int rollno=0; double marks=0;

StudentInfo(this.rollno,this.marks,name,age) : super(name, age); int get studentRollno{ return rollno;

}

set studentRollno(int rollno){ this.rollno=rollno;

}

double get studentMarks{ return marks;

}

set studentMarks(double marks){ this.marks=marks;

}

void printData()

{

super.printData();

print("Rollno=$studentRollno"); print("Marks=$studentMarks");

}

}

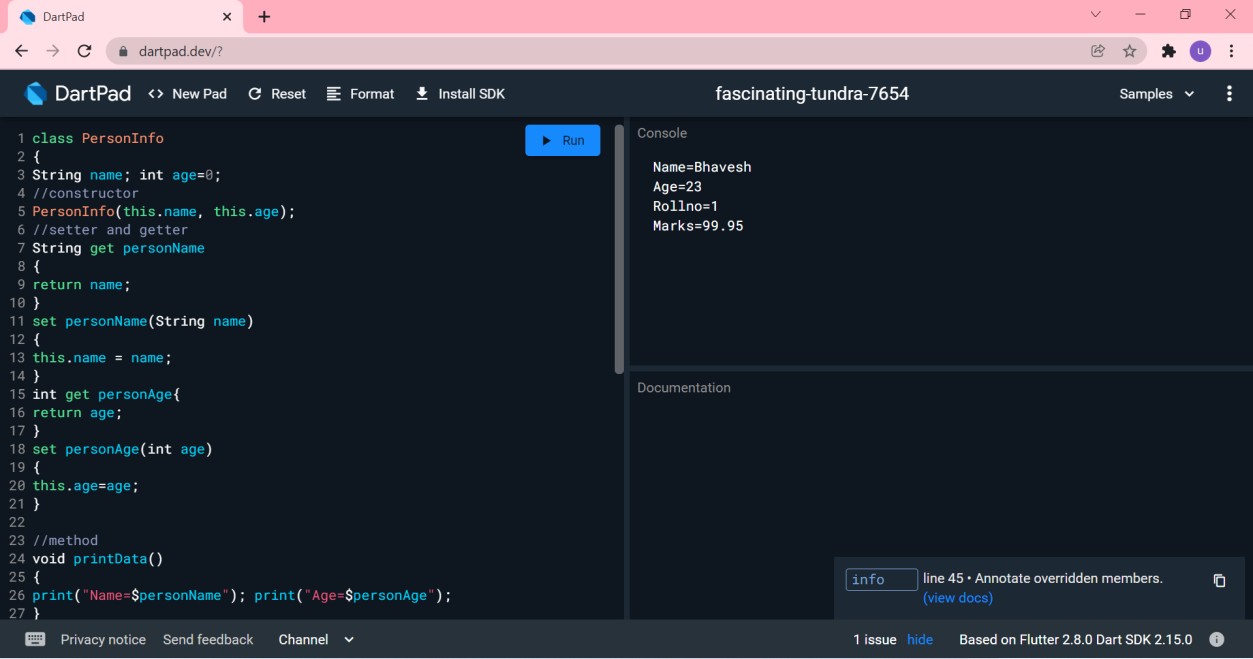
void main()

{

StudentInfo s=StudentInfo(1,99.95,"Bhavesh",23); s.printData();

}

**Output:**



1. **Abstract class and Interfaces Source Code:**

abstract class shape

{

//abstract method void draw(){}

}

class line extends shape

{

void draw(){ print('Line is drawing');

}

}

class circle extends shape

{

void draw(){ print('Circle is drawing');

}

}

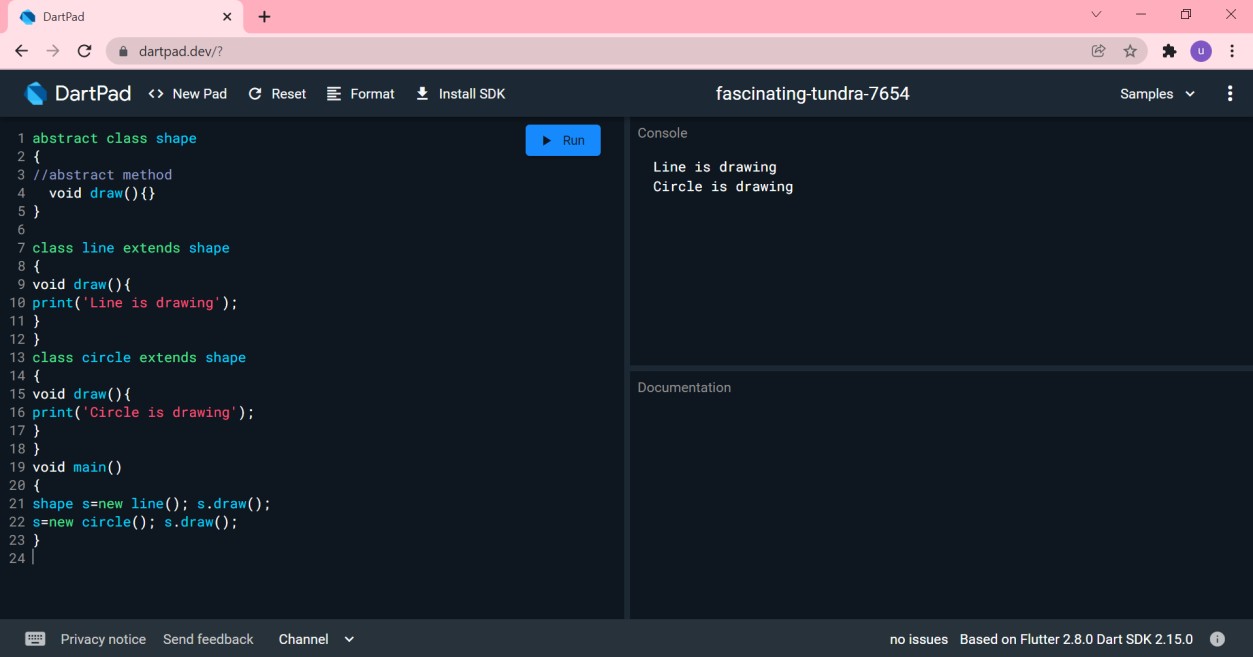
void main()

{

shape s=new line(); s.draw(); s=new circle(); s.draw();

}

**Output:**



**Practical No 21**

**Aim: Write a program to understand the Basic structure of Flutter framework**

1. **Directionality**

**File Name: main.dart**

Import ‘package:flutter/material.dart’;

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

}

class MyApp extends StatelessWidget {

const MyApp({Key? Key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) {

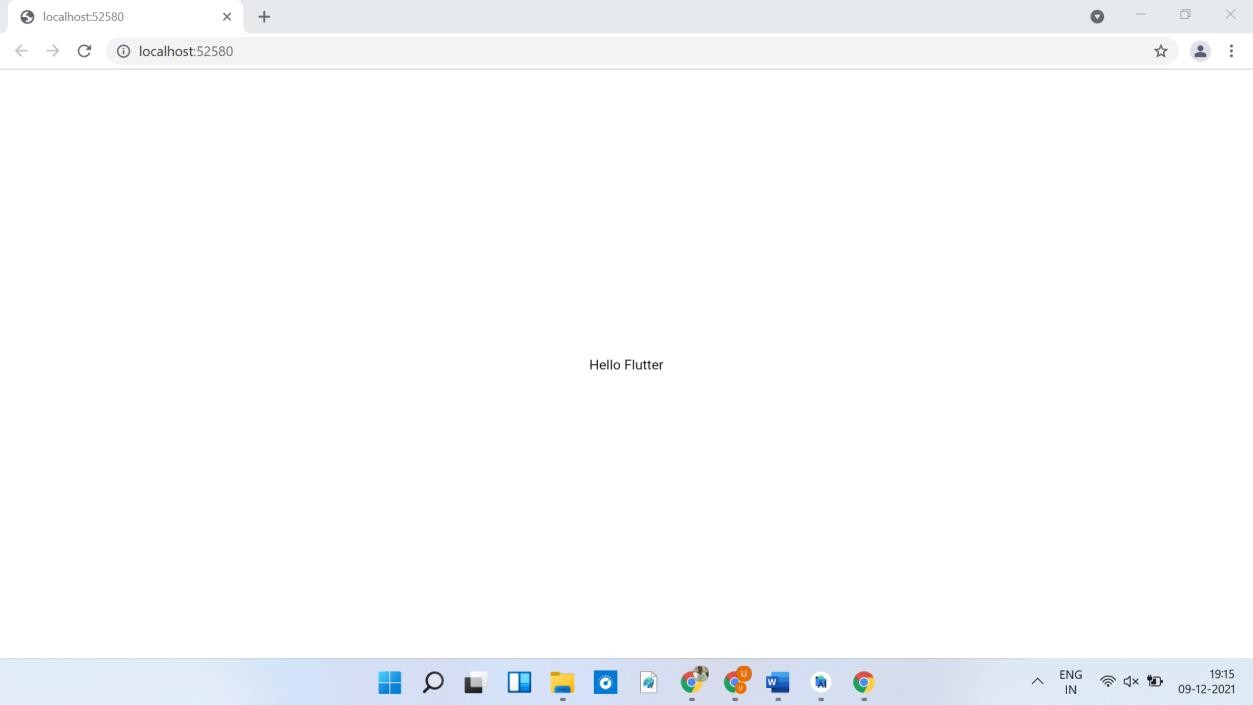
return Directionality(textDirection: TextDirection.ltr, child: Center(child: Text(“Hello Flutter”) )

);

}

}

**Output: Chrome Web**



**Mobile View**



1. **Material Design File Name: main.dart**

import 'package:flutter/material.dart';

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

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) {

return MaterialApp(title: "Demo2-Material Design", home: Scaffold(

appBar:AppBar(

title:Text('Demo 2-Material Design'),

),

body: Center(

child: Text('Hello World'), ),

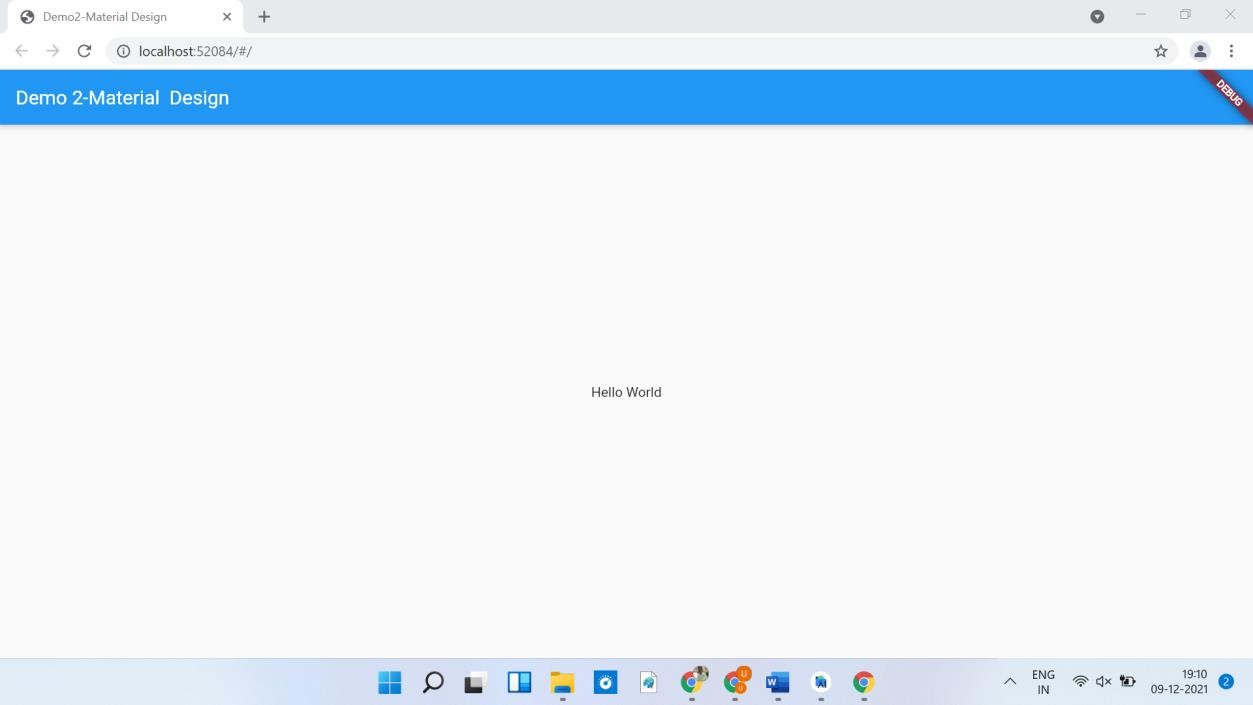
),

);

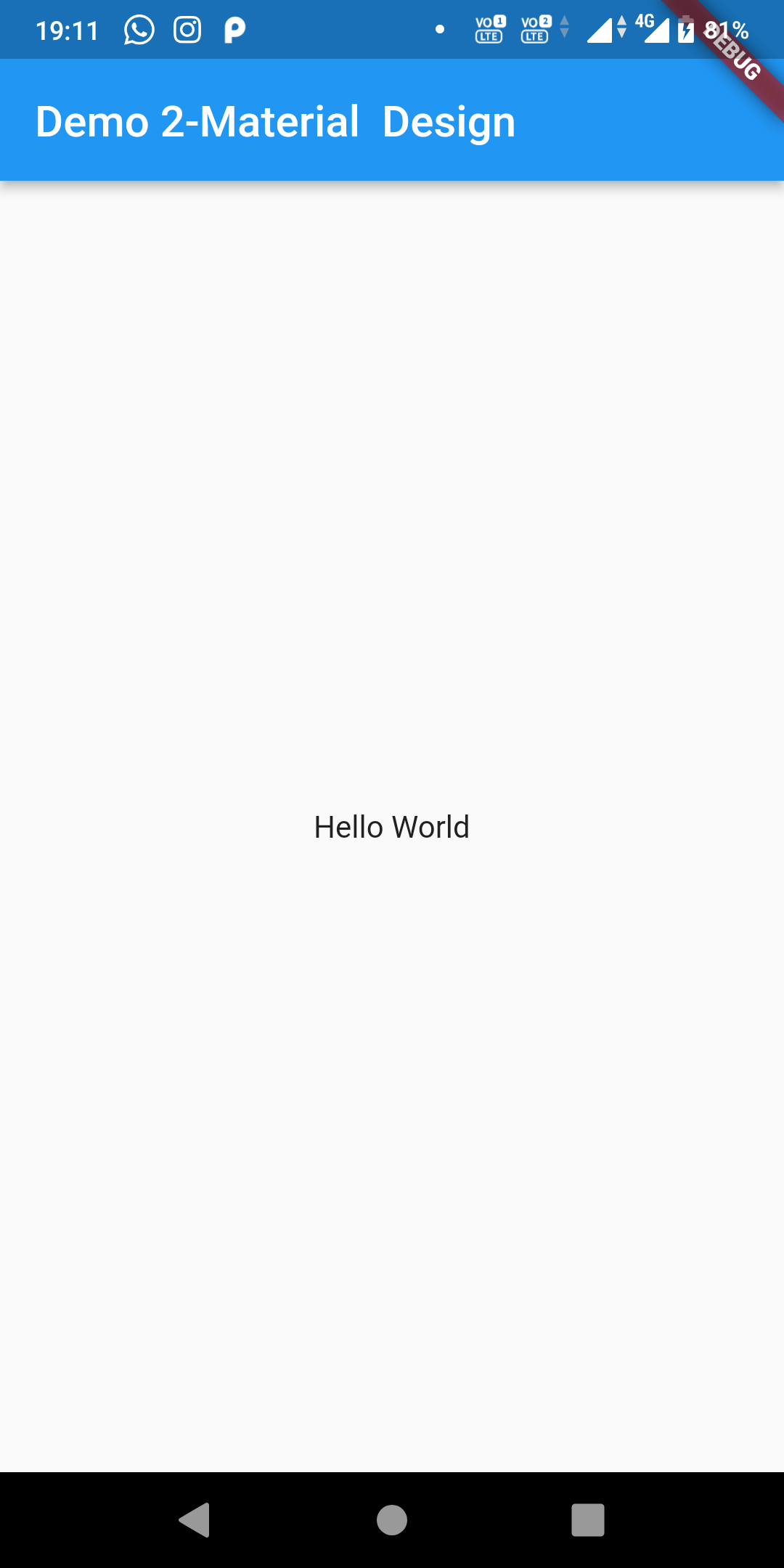
}

}

## Output: Chrome Web



**Mobile View**



1. **Row widget**

**File Name: main.dart**

import 'package:flutter/material.dart';

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

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) { return MaterialApp(

title: "Demo2-Material Design", home: Scaffold(

appBar:AppBar(

title:Text('Demo 3- Row Widget'),

),

body: Center(

child: Row(

mainAxisAlignment: MainAxisAlignment.center, children:<Widget>[ Text('Child1'), Text('Child2'),

],

)

)

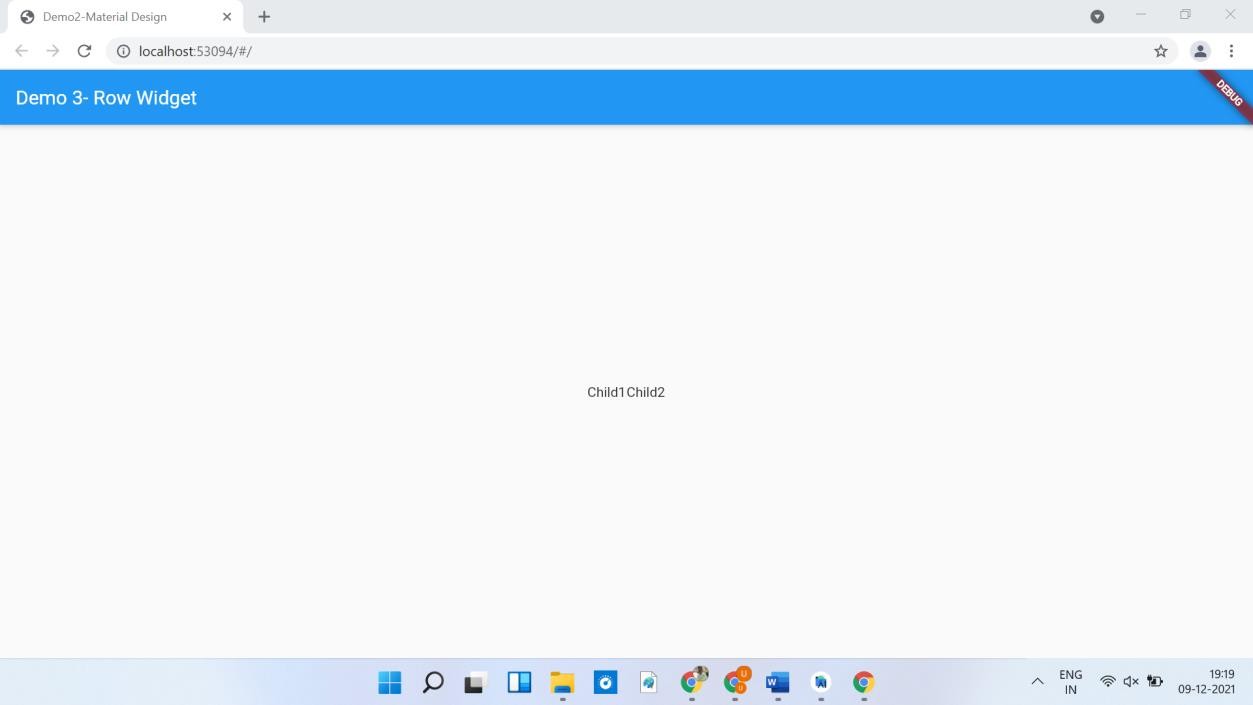
),

);

}

}

## Output: Chrome Web



**Mobile View**



**Practical No 22**

**Aim: Write a program to understand the use of basic Widget**

1. **Text**
2. **Row, Column**
3. **Stack**

**4 . Container**

**File Name: main.dart**

import 'package:flutter/material.dart';

void main() { runApp(

const MaterialApp(

title: 'Flutter Tutorial', home: MyApp(),

),

);

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

leading: const IconButton( icon: Icon(Icons.*menu*), tooltip: 'Navigation menu', onPressed: null,

),

title: const Text('Example title'), actions: const [

IconButton(

icon: Icon(Icons.*search*), tooltip: 'Search', onPressed: null,

),

],

),

*// body is the majority of the screen.*

body: const Center(

child: Text('Hello, world!'),

),

floatingActionButton: const FloatingActionButton( tooltip: 'Add', *// used by assistive technologies child: Icon(Icons.add),*

onPressed: null,

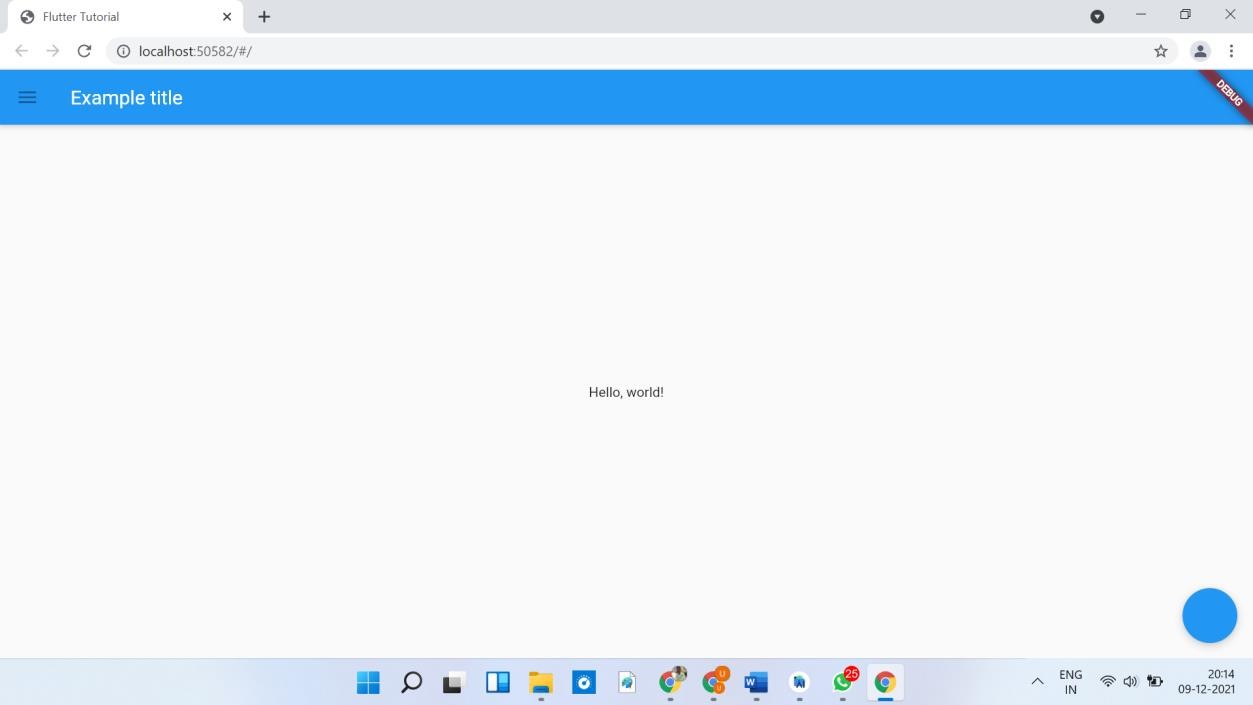
),

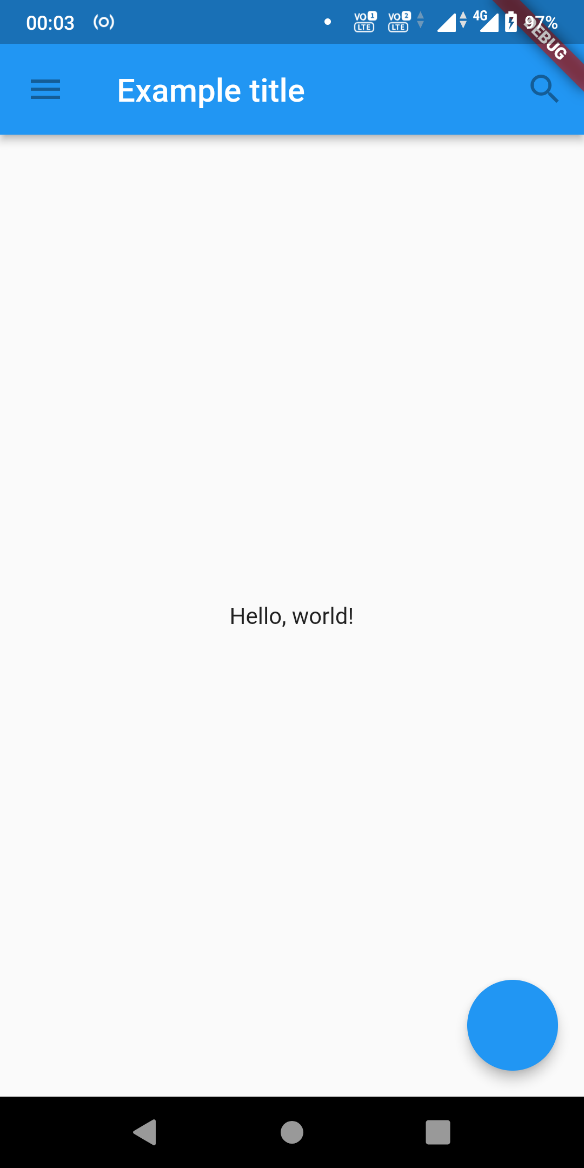
);

}

}

## Output: Chrome Web



**Mobile View**

**Practical No 23**

**Aim: Write a program to understand the working of checkboxes.**

**File Name: main.dart**

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

runApp(MyApp(

TextInput:Text("Paid/Unpaid"), ));

}

class MyApp extends StatefulWidget {

MyApp({required this.TextInput});

final Widget TextInput;

MyAppState createState() => new MyAppState();

}

class MyAppState extends State<MyApp> {

bool checkBoxValue = false; String actionText="Default"; @override Widget build(BuildContext ctxt) {

return MaterialApp( title:

"MySampleApplication", home: Scaffold(

appBar: AppBar(

title: Text("Hello Flutter App"),

),

body: Center( child:Column(

children: <Widget>[ widget.TextInput, Text(actionText), Checkbox(value:

checkBoxValue, onChanged: (bool ? newValue)

{

setState(() { checkBoxValue=newValue!;

if(newValue==true) { actionText = "Checked";

}

else {

actionText="unChecked";

}

});

}

)

],

),

)

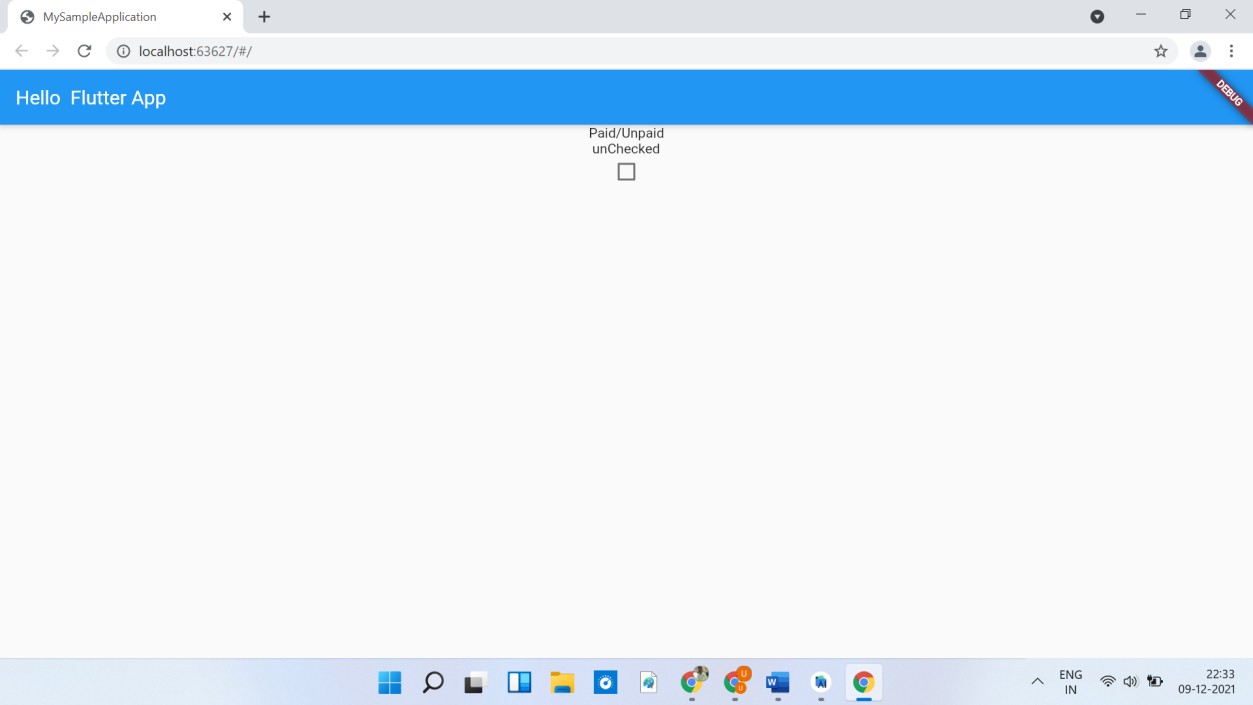
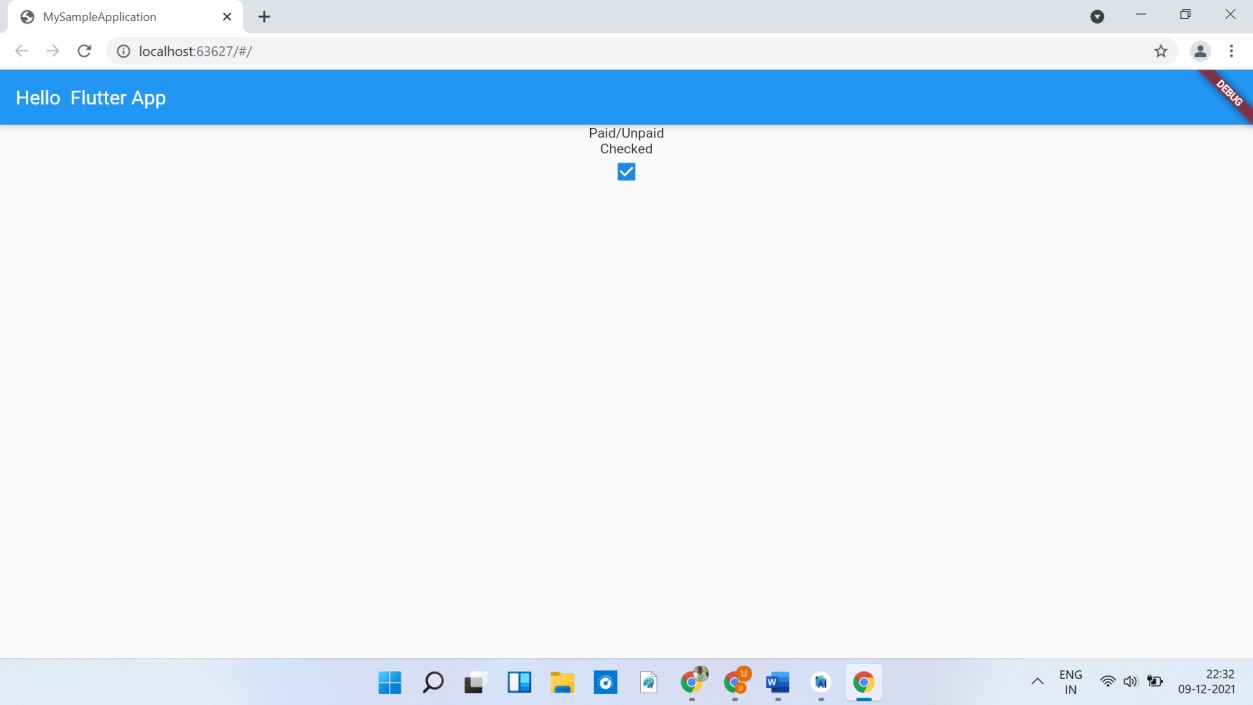
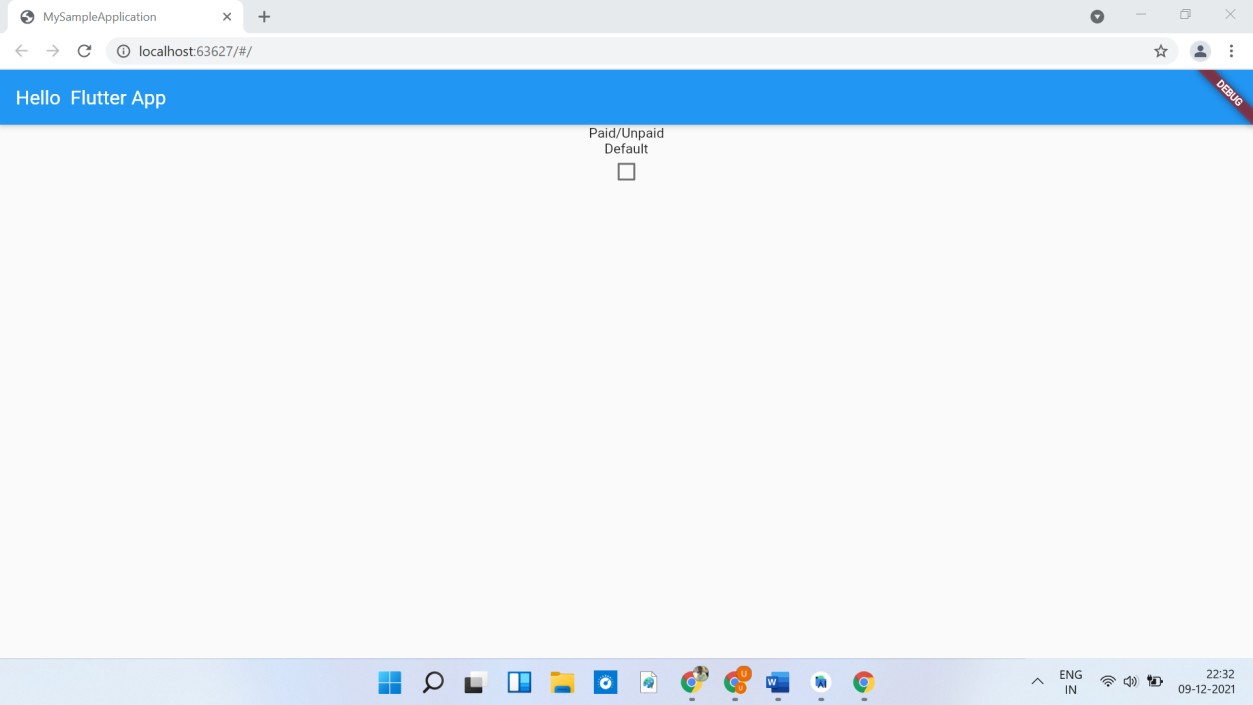
),

);

}

}

## Output: Chrome Web



**Mobile View**



**Practical No 24**

**Aim: Write a program to understand the working of buttons.**

**File Name: main.dart**

import 'package:flutter/material.dart';

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

TextInput:Text("Default Text");

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) { return MaterialApp(

title: 'Flutter Demo', theme: ThemeData(

primarySwatch: Colors.*blue*,

),

home: const MyHomePage(title: 'Demo-5 StateFul Example'),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key, required this.title}) : super(key: key);

final String title;

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> { int \_counter = 0;

void \_incrementCounter() { setState(() {

\_counter++;

});

}

void \_decrementCounter() { setState(() {

\_counter--;

});

}

@override

Widget build(BuildContext context) { return Scaffold(

appBar: AppBar(

title: Text(widget.title),

),

body: Center( child: Column(

mainAxisAlignment: MainAxisAlignment.center, children: <Widget>[

Text(

'$\_counter',

style: Theme.*of*(context).textTheme.headline1,

),

Row(

mainAxisAlignment: MainAxisAlignment.center, children: [ Padding(

padding: const EdgeInsets.all(8), child:

FlatButton (

onPressed: \_incrementCounter, child: Text('Add', style: TextStyle(fontSize: 30)), padding:

EdgeInsets.all(20), textColor: Colors.*white*,

color: Colors.*red*,),

),

Padding(

padding: const EdgeInsets.all(8), child: FlatButton

(

onPressed: \_decrementCounter, child: Text('Sub', style: TextStyle(fontSize: 30)), padding:

EdgeInsets.all(20), textColor: Colors.*white*, color: Colors.*blue*,)

),

],

),

],

),

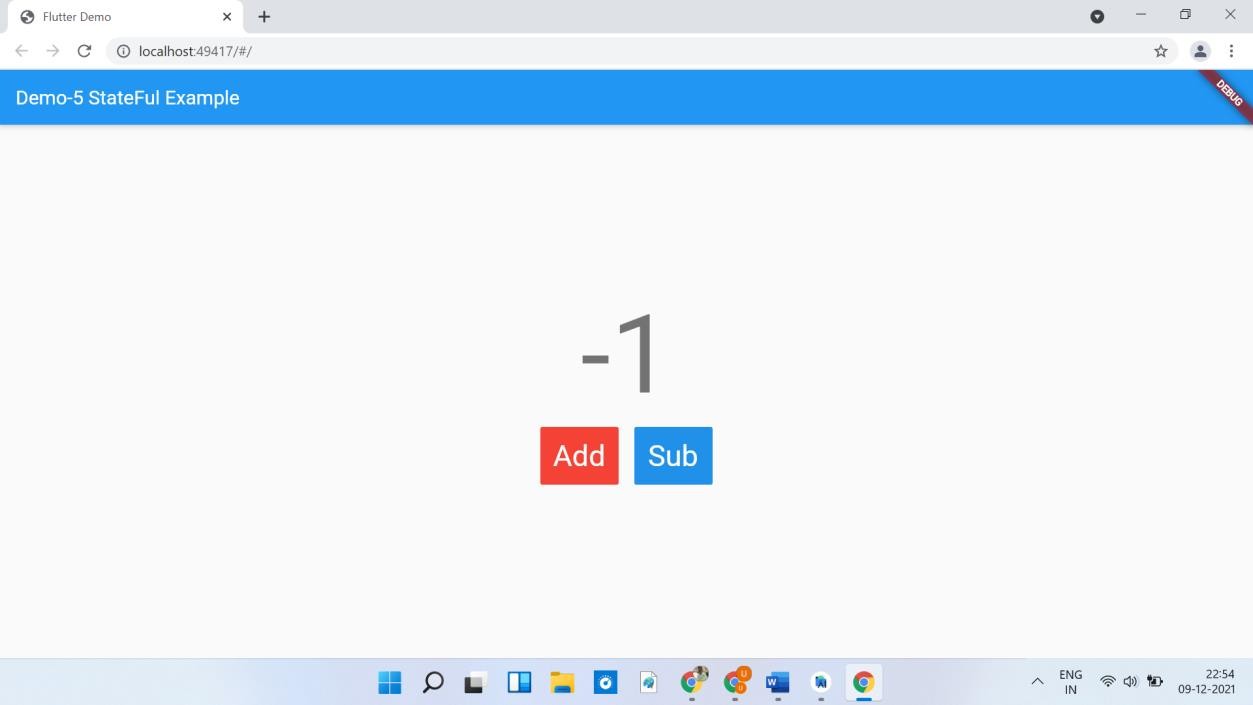
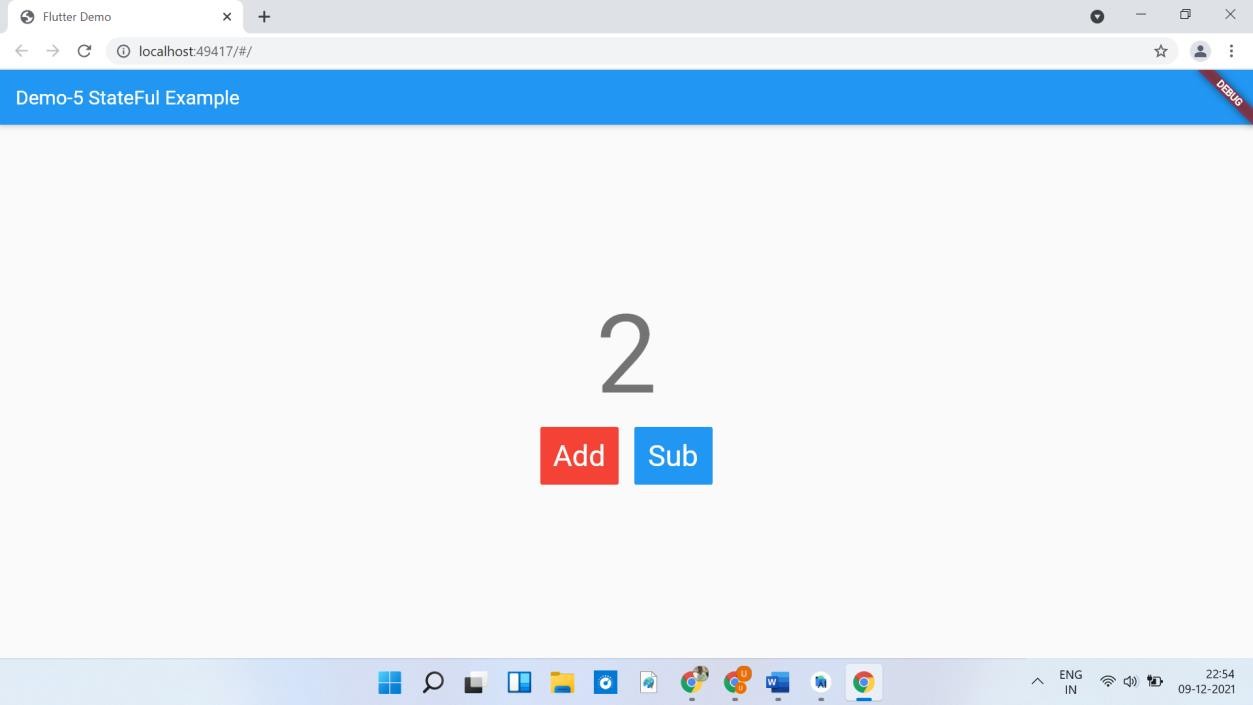
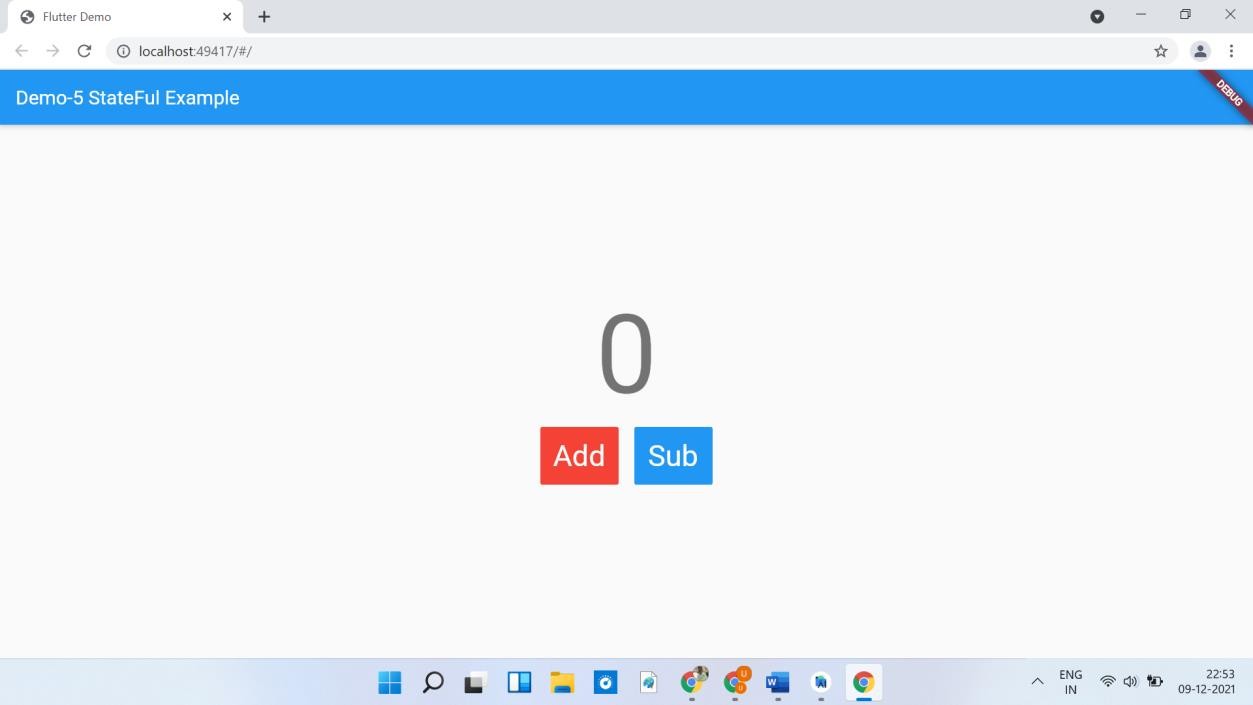
),

);

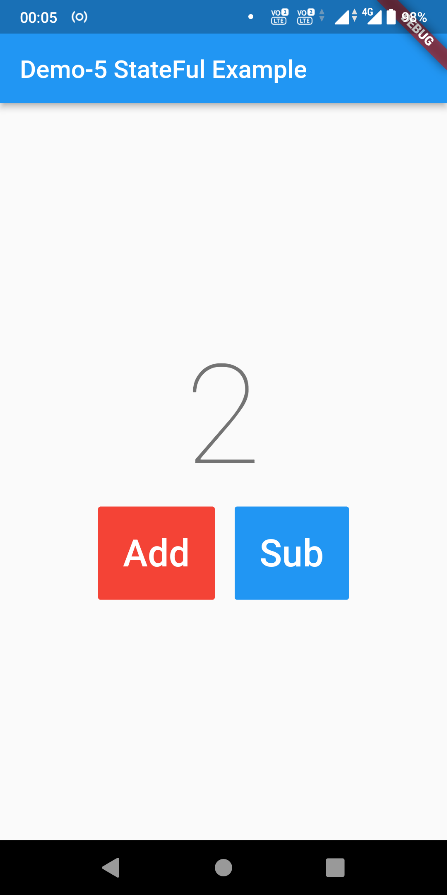
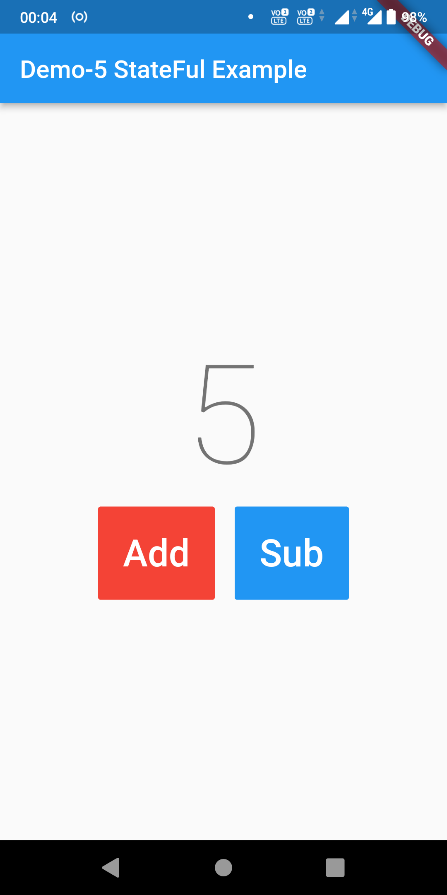
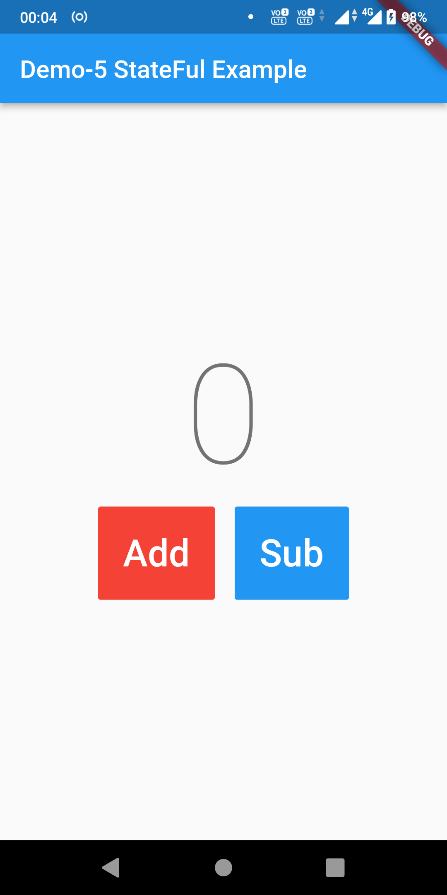
}

}

## Output: Chrome Web



**Mobile View**



**Practical No 25**

**Aim: Write a program to show the working of the counter.**

**File Name: main.dart**

import 'package:flutter/material.dart';

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

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) { return MaterialApp(

title: 'Flutter Demo', theme: ThemeData(

*// This is the theme of your application.*

*//*

*// Try running your application with "flutter run". You'll see the*

*// application has a blue toolbar. Then, without quitting the app,*

*try run",*

*// changing the primarySwatch below to Colors.green and then invoke*

*// "hot reload" (press "r" in the console where you ran "flutter*

*// or simply save your changes to "hot reload" in a Flutter IDE).*

*// Notice that the counter didn't reset back to zero; the*

*application*

*// is not restarted.*

primarySwatch: Colors.*blue*,

),

home: const MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key, required this.title}) : super(key: key);

*// This widget is the home page of your application. It is stateful, meaning*

*// that it has a State object (defined below) that contains fields that affect*

*// how it looks.*

*// This class is the configuration for the state. It holds the values (in this*

*// case the title) provided by the parent (in this case the App widget) and*

*// used by the build method of the State. Fields in a Widget subclass are*

*// always marked "final".*

final String title; @override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> { int \_counter = 0;

void \_incrementCounter() { setState(() {

*// This call to setState tells the Flutter framework that something*

*has below*

*// changed in this State, which causes it to rerun the build method*

*// so that the display can reflect the updated values. If we changed*

*// \_counter without calling setState(), then the build method would*

*not be*

*// called again, and so nothing would appear to happen.*

\_counter++;

});

}

@override

Widget build(BuildContext context) {

*// This method is rerun every time setState is called, for instance as done*

*// by the \_incrementCounter method above.*

*//*

*// The Flutter framework has been optimized to make rerunning build methods*

*// fast, so that you can just rebuild anything that needs updating rather*

*// than having to individually change instances of widgets.*

return Scaffold( appBar: AppBar(

*// Here we take the value from the MyHomePage object that was created by*

*// the App.build method, and use it to set our appbar title.*

title: Text(widget.title),

*it*

*and its*

),

body: Center(

*// Center is a layout widget. It takes a single child and positions*

*// in the middle of the parent.*

child: Column(

*// Column is also a layout widget. It takes a list of children*

*// arranges them vertically. By default, it sizes itself to fit*

*// children horizontally, and tries to be as tall as its parent.*

*//*

*// Invoke "debug painting" (press "p" in the console, choose the*

*// "Toggle Debug Paint" action from the Flutter Inspector in*

*Android Code)*

*and to*

*vertical*

*// Studio, or the "Toggle Debug Paint" command in Visual Studio*

*// to see the wireframe for each widget.*

*//*

*// Column has various properties to control how it sizes itself*

*// how it positions its children. Here we use mainAxisAlignment*

*// center the children vertically; the main axis here is the*

*// axis because Columns are vertical (the cross axis would be*

*// horizontal).*

mainAxisAlignment: MainAxisAlignment.center, children: <Widget>[

const Text(

'You have pushed the button this many times:',

),

Text(

'$\_counter',

style: Theme.*of*(context).textTheme.headline4,

),

],

),

),

floatingActionButton: FloatingActionButton( onPressed: \_incrementCounter,

tooltip: 'Increment',

child: const Icon(Icons.*add*),

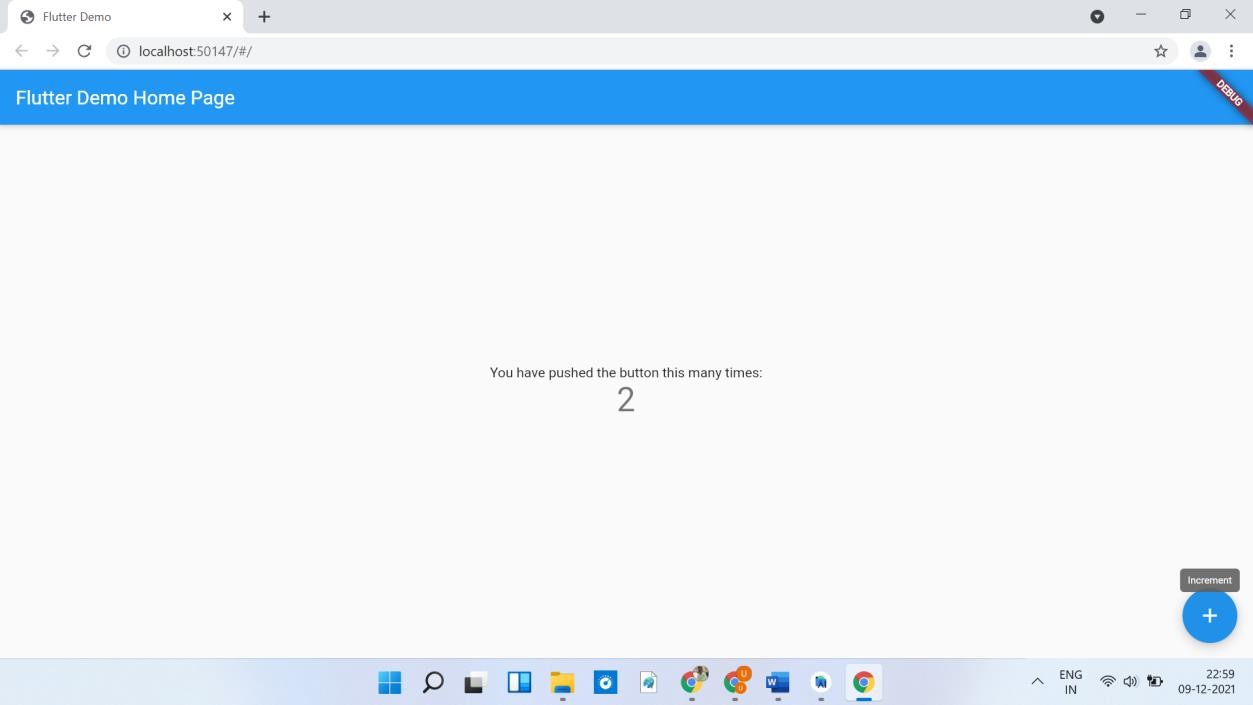
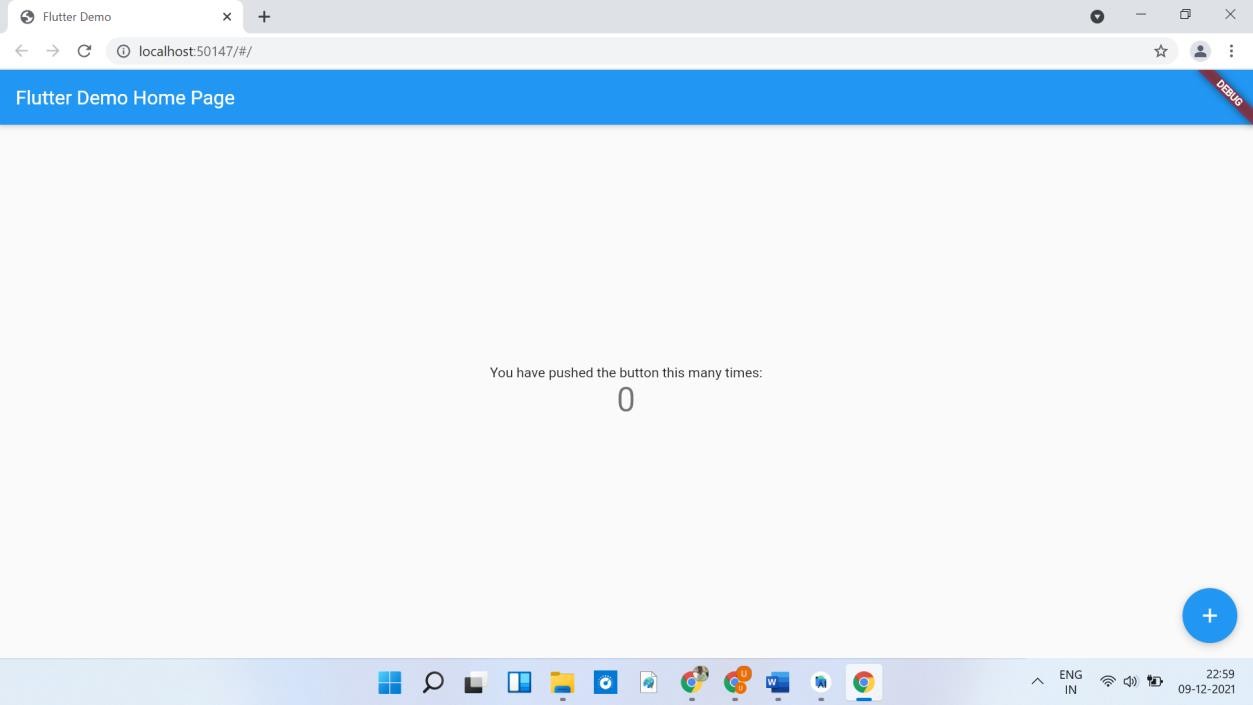
), *// This trailing comma makes auto-formatting nicer for build methods.*

);

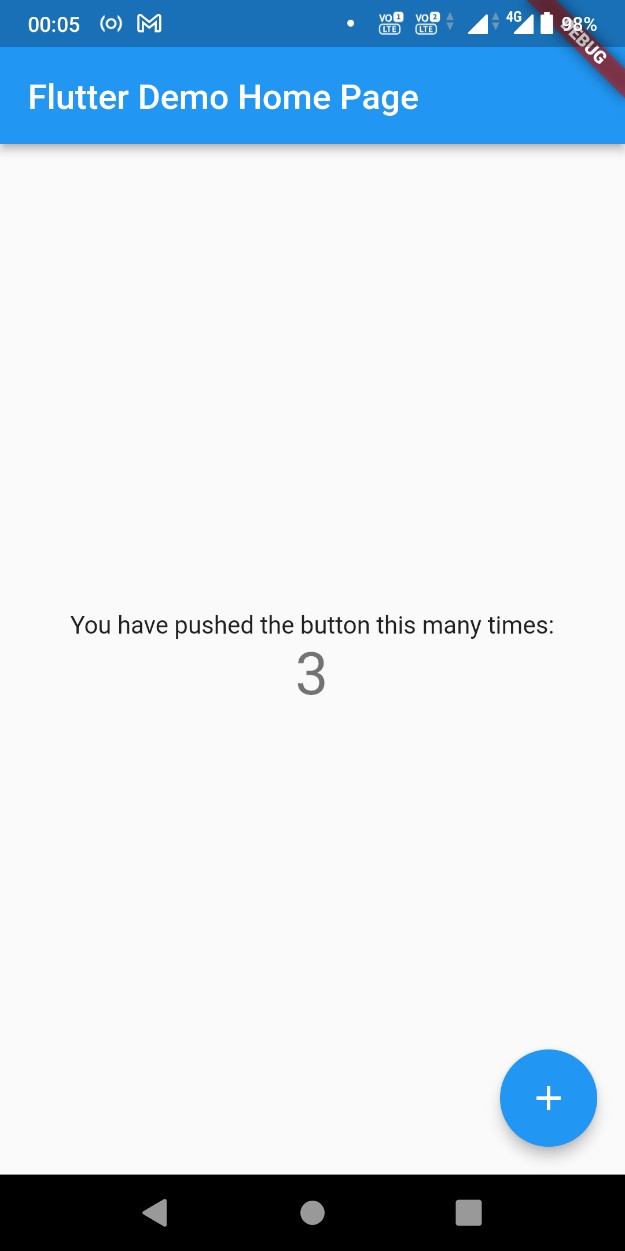
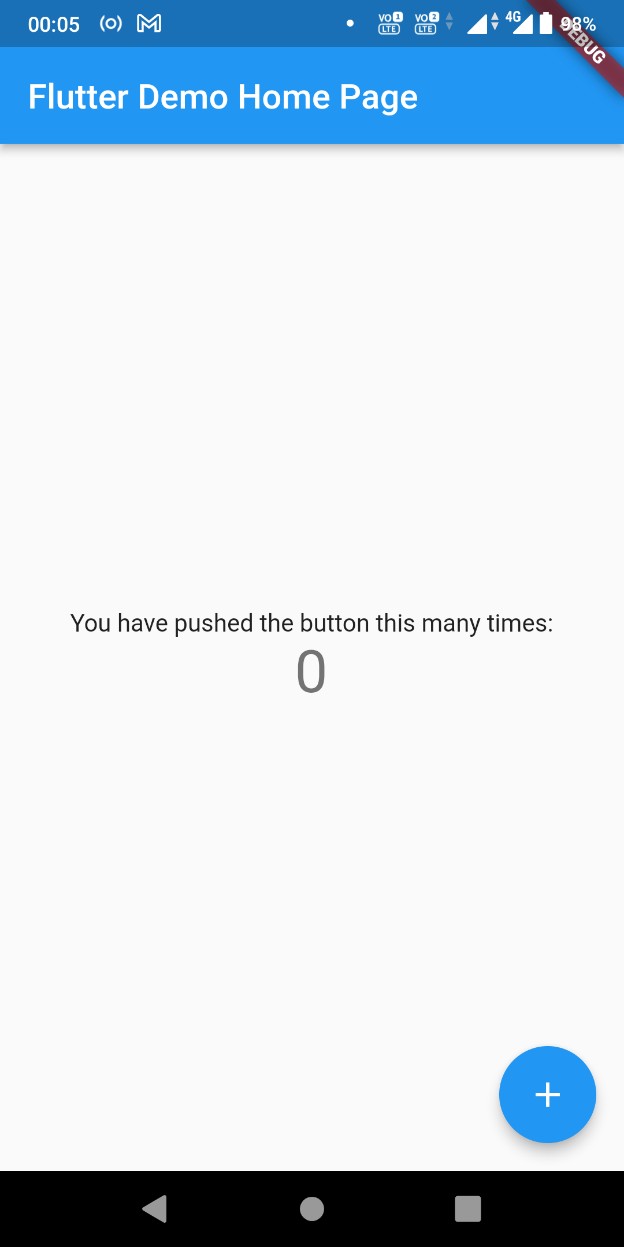
}

}

## Output: Chrome Web



**Mobile View**



**Practical No 26**

**Aim: Write a flutter program using android studio: Flutter Navigation File Name: main.dart**

import 'package:flutter/material.dart'; import 'package:navigation/Home.dart'; import 'package:navigation/Dashboard.dart'; void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

*// This widget is the root of your application.*

@override

Widget build(BuildContext context) { return MaterialApp(

initialRoute: '/Home.dart', routes: { '/home':(context)=>Home(), '/Dashboard':(context)=>Dashboard(),

},

home:Home());

}

}

class MyHomePage extends StatefulWidget {

const MyHomePage({Key? key, required this.title}) : super(key: key);

*// This widget is the home page of your application. It is stateful, meaning*

*// that it has a State object (defined below) that contains fields that affect*

*// how it looks.*

*// This class is the configuration for the state. It holds the values (in this*

*// case the title) provided by the parent (in this case the App widget) and*

*// used by the build method of the State. Fields in a Widget subclass are*

*// always marked "final".*

final String title;

@override

State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> { int \_counter = 0;

void \_incrementCounter() { setState(() {

*// This call to setState tells the Flutter framework that something*

*has below*

*// changed in this State, which causes it to rerun the build method*

*// so that the display can reflect the updated values. If we changed*

*// \_counter without calling setState(), then the build method would*

*not be*

*// called again, and so nothing would appear to happen.*

\_counter++;

});

}

@override

Widget build(BuildContext context) {

*// This method is rerun every time setState is called, for instance as done*

*// by the \_incrementCounter method above.*

*//*

*// The Flutter framework has been optimized to make rerunning build methods*

*// fast, so that you can just rebuild anything that needs updating rather*

*// than having to individually change instances of widgets.*

return Scaffold( appBar: AppBar(

*// Here we take the value from the MyHomePage object that was created by*

*// the App.build method, and use it to set our appbar title.*

title: Text(widget.title),

*it*

*and its*

),

body: Center(

*// Center is a layout widget. It takes a single child and positions*

*// in the middle of the parent.*

child: Column(

*// Column is also a layout widget. It takes a list of children*

*// arranges them vertically. By default, it sizes itself to fit*

*// children horizontally, and tries to be as tall as its parent.*

*//*

*// Invoke "debug painting" (press "p" in the console, choose the*

*// "Toggle Debug Paint" action from the Flutter Inspector in*

*Android Code)*

*and to*

*vertical*

),

*// Studio, or the "Toggle Debug Paint" command in Visual Studio*

*// to see the wireframe for each widget.*

*//*

*// Column has various properties to control how it sizes itself*

*// how it positions its children. Here we use mainAxisAlignment*

*// center the children vertically; the main axis here is the*

*// axis because Columns are vertical (the cross axis would be*

*// horizontal).*

mainAxisAlignment: MainAxisAlignment.center, children: <Widget>[

const Text(

'You have pushed the button this many times:',

),

Text(

'$\_counter',

style: Theme.*of*(context).textTheme.headline4,

),

],

),

floatingActionButton: FloatingActionButton( onPressed: \_incrementCounter,

tooltip: 'Increment',

child: const Icon(Icons.*add*),

), *// This trailing comma makes auto-formatting nicer for build methods.*

);

}

}

## File Name: Home.dart

import 'package:flutter/material.dart'; import 'Dashboard.dart'; class Home extends StatefulWidget{

\_HomeState createState()=>\_HomeState();

}

class \_HomeState extends State<Home> { @override Widget build(BuildContext context) {

*// TODO: implement build*

return Scaffold( appBar: AppBar( title:Text("Home Screen"),

),

body:Column( mainAxisAlignment:MainAxisAlignment.center, children:<Widget>[

Text("Welcome to Home Screen", style:TextStyle( fontSize:20

),

),

SizedBox( height:10,

),

Center( child:RaisedButton( onPressed: () { Navigator.*push*(context,MaterialPageRoute(

builder: (context)=>Dashboard()));

},

child:Text('Move to Dashboard Screen',style:

TextStyle(fontSize: 20,

fontWeight: FontWeight.*bold*),)

)

)

],

)

);

throw UnimplementedError();

}

}

## File Name: Dashboard.dart

import 'package:flutter/material.dart'; import 'Home.dart'; class Dashboard extends StatefulWidget{

\_DashboardState createState()=>\_DashboardState();

}

class \_DashboardState extends State<Dashboard> { @override Widget build(BuildContext context) {

*// TODO: implement build*

return Scaffold( appBar: AppBar( title:Text("Dashboard Screen"),

),

body:Column( mainAxisAlignment:MainAxisAlignment.center, children:<Widget>[

Text("Welcome to Dashboard Screen", style:TextStyle(

fontSize:20

),

),

SizedBox( height:10,

),

Center(

child:RaisedButton(

onPressed:(){Navigator.*push*(context,MaterialPageRoute( builder: (context)=>Home()));

20,

)

],

)

);

},

child:Text('Move to Home Screen',style: TextStyle(fontSize:

fontWeight: FontWeight.*bold*),)

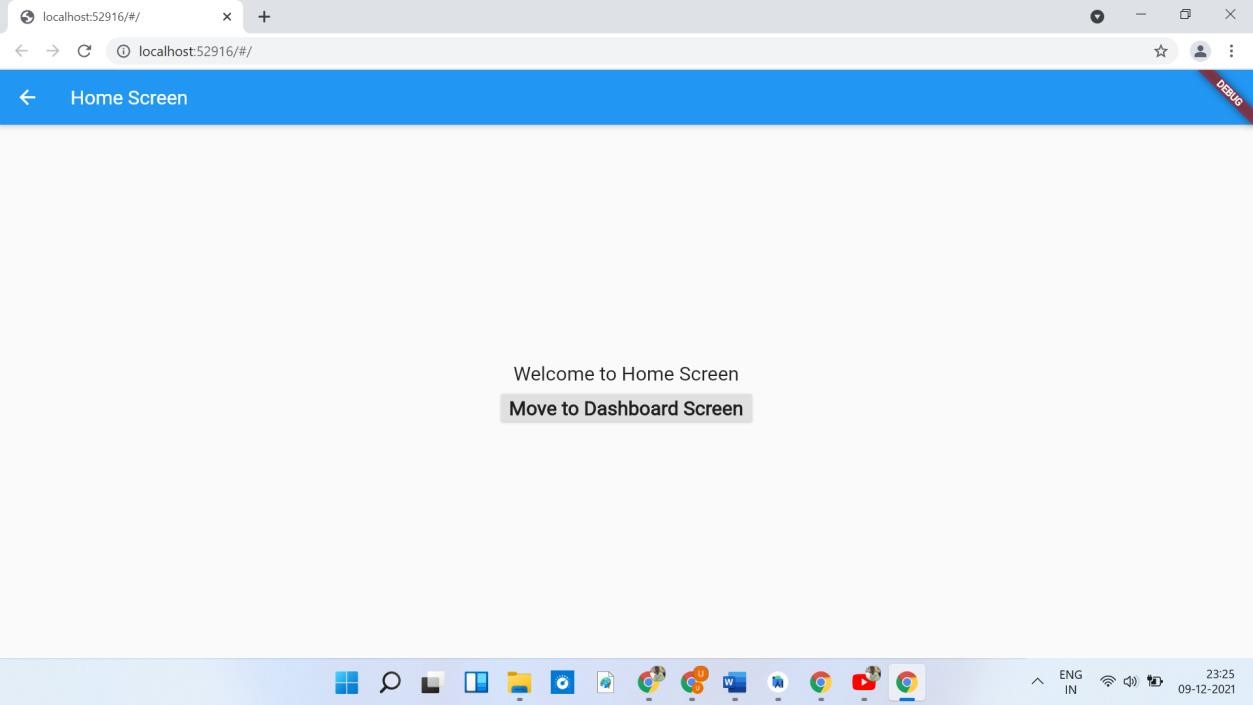
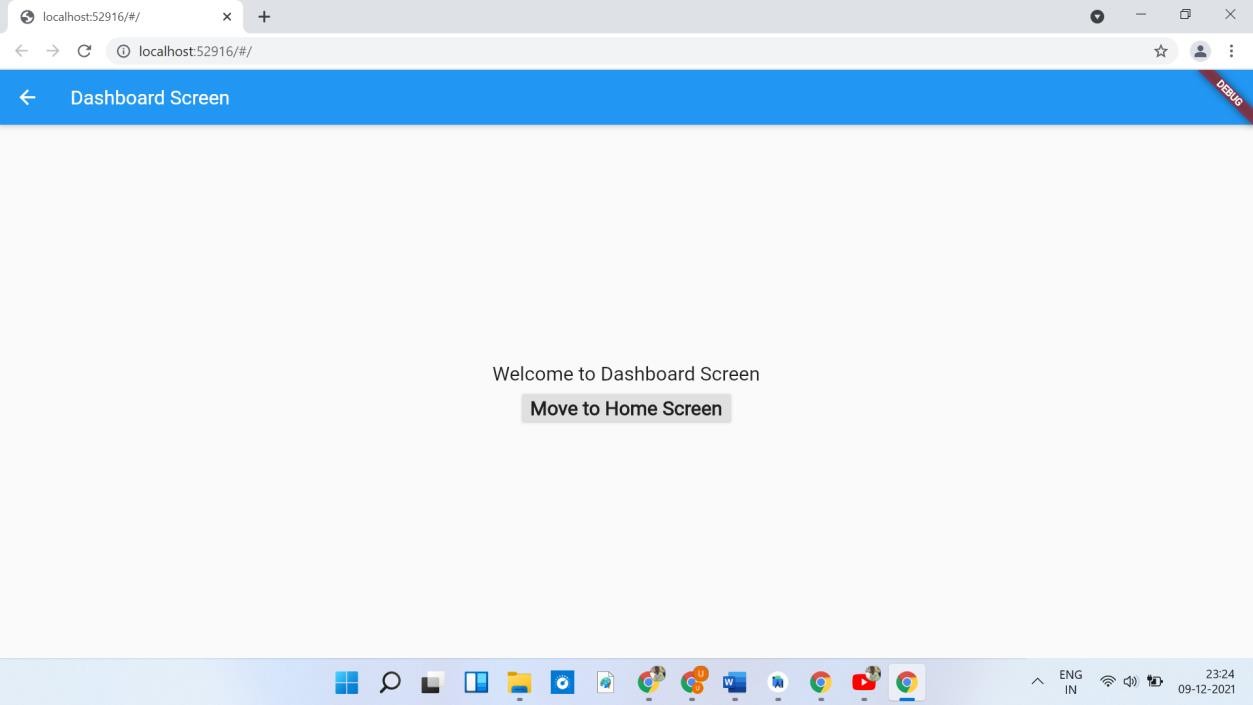
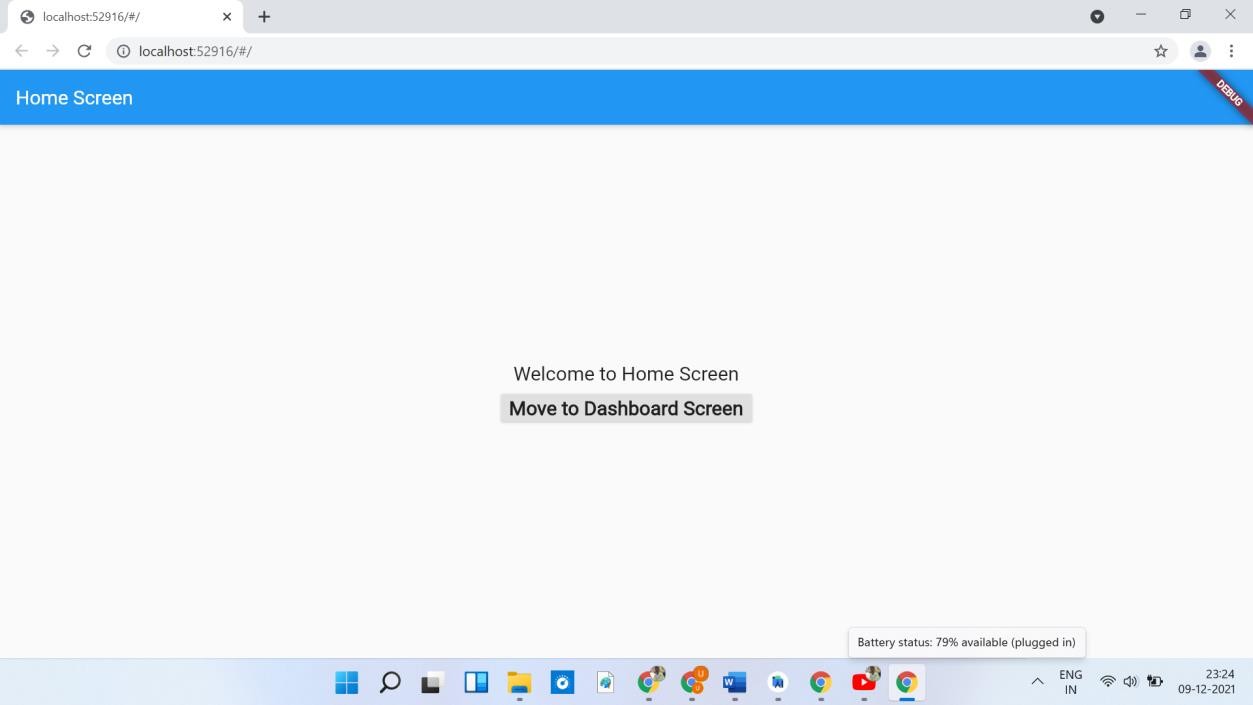
)

throw UnimplementedError();

}

}

## Output: Chrome Web



**Mobile View**

