**Practical 1**

**AIM: Introduction to Android and Create “Custom Message” application. That will**

**display “Custom Message” in the middle of the screen in the Black color with the**

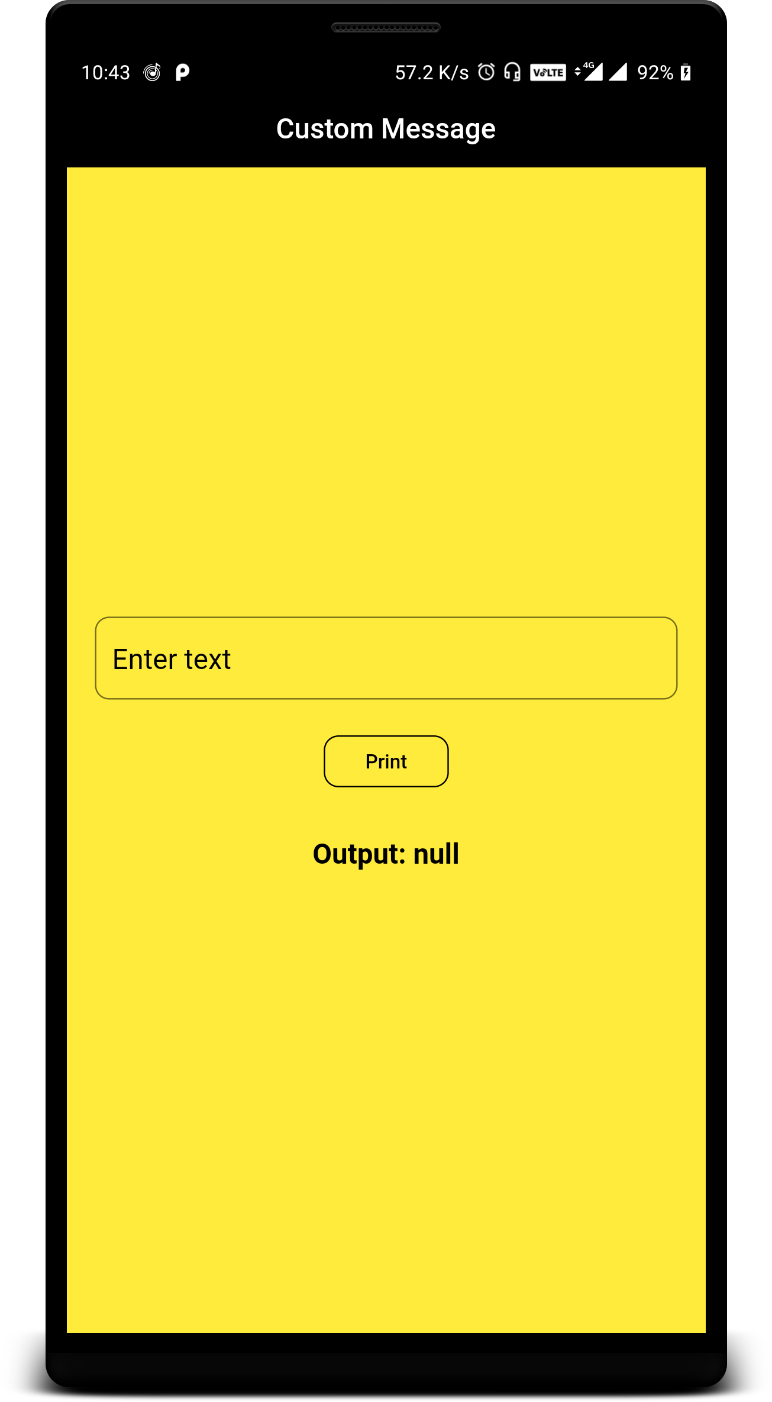
**Yellow background.**

**Source Code:**

**main.dart:**

|  |
| --- |
| import 'package:flutter/material.dart';  void main() => runApp(MyApp());  class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  debugShowCheckedModeBanner: false,  title: 'Custom Message',  theme: ThemeData(primaryColor: Colors.black, cursorColor: Colors.black),  home: MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {  State createState() => new HomePageState();  }  class HomePageState extends State<MyHomePage> {  final TextEditingController textEditingController =  new TextEditingController();  String statement;  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: new AppBar(  title:  new Text("Custom Message", style: TextStyle(color: Colors.black)),  centerTitle: true,  elevation: 0.0,  backgroundColor: Colors.yellow,  ),  body: Container(  color: Colors.yellow,  child: Column(children: <Widget>[  Padding(  padding: EdgeInsets.all(20.0),  child: TextField(  keyboardType: TextInputType.text,  decoration: new InputDecoration(  labelText: "Enter text",  labelStyle:  new TextStyle(color: Colors.black, fontSize: 20.0),  border: OutlineInputBorder(  borderRadius: new BorderRadius.circular(10.0)),  ),  controller: textEditingController,  ),  ),  new OutlineButton(  child: new Text("Print"),  borderSide: new BorderSide(color: Colors.black),  shape: new RoundedRectangleBorder(  borderRadius: new BorderRadius.circular(10.0)),  textColor: Colors.black,  padding: EdgeInsets.all(10),  onPressed: doPrint,  ),  new Padding(  padding: EdgeInsets.all(30.0),  child: Text("Output: $statement",  style: new TextStyle(  fontSize: 20.0,  color: Colors.black,  fontFamily: 'RobotMono',  fontWeight: FontWeight.bold)),  ),  ]),  ));  }  void doPrint() {  setState(() {});  statement = textEditingController.text;  }  } |

**Output:-**

****

**Conclusion: By performing this experiment, one can have the knowledge of basic widgets of flutter.**

**Practical 2**

**AIM: Create an android application to calculate sum of two numbers and gives result in**

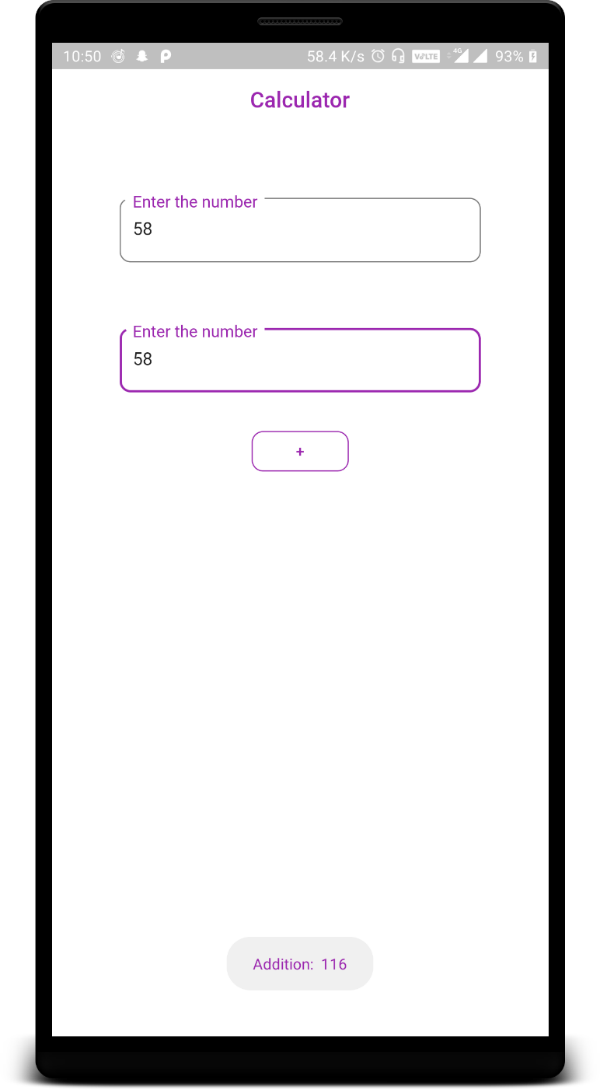
**Toast Message.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:fluttertoast/fluttertoast.dart';  void main() => runApp(MyApp());  class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  debugShowCheckedModeBanner: false,  title: 'Flutter Demo',  theme: ThemeData(primaryColor: Colors.purple, cursorColor: Colors.purple),  home: MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {  @override  \_MyHomePageState createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {  final TextEditingController textEditingController =  new TextEditingController();  final TextEditingController textEditingControllerOne =  new TextEditingController();  int sum, num1, num2;  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: new AppBar(  title: new Text("Calculator", style: TextStyle(color: Colors.purple)),  centerTitle: true,  elevation: 0.0,  backgroundColor: Colors.white,  ),  body: new Container(  color: Colors.white,  child: new Padding(  padding: EdgeInsets.all(32.0),  child: new Column(  children: <Widget>[  new Padding(  padding: EdgeInsets.all(30.0),  child: new TextField(  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Enter the number",  labelStyle:  new TextStyle(color: Colors.purple, fontSize: 20.0),  border: OutlineInputBorder(  borderRadius: new BorderRadius.circular(10.0)),  ),  controller: textEditingController,  ),  ),  new Padding(  padding: EdgeInsets.all(30.0),  child: new TextField(  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Enter the number",  labelStyle:  new TextStyle(color: Colors.purple, fontSize: 20.0),  border: OutlineInputBorder(  borderRadius: new BorderRadius.circular(10.0)),  ),  controller: textEditingControllerOne,  ),  ),  new Row(  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  children: <Widget>[  new OutlineButton(  child: new Text("+"),  borderSide: new BorderSide(color: Colors.purple),  shape: new RoundedRectangleBorder(  borderRadius: new BorderRadius.circular(10.0)),  textColor: Colors.purple,  padding: EdgeInsets.all(10),  onPressed: doAddition,  ),  ],  ),  ],  ),  ),  ),  );  }  void doAddition() {  num1 = int.parse(textEditingController.text);  num2 = int.parse(textEditingControllerOne.text);  sum = num1 + num2;  Fluttertoast.showToast(  msg: "Addition: $sum",  toastLength: Toast.LENGTH\_LONG,  gravity: ToastGravity.BOTTOM,  backgroundColor: Colors.black,  textColor: Colors.yellow);  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can able to implement the basic calculation app in android as well as flutter.**

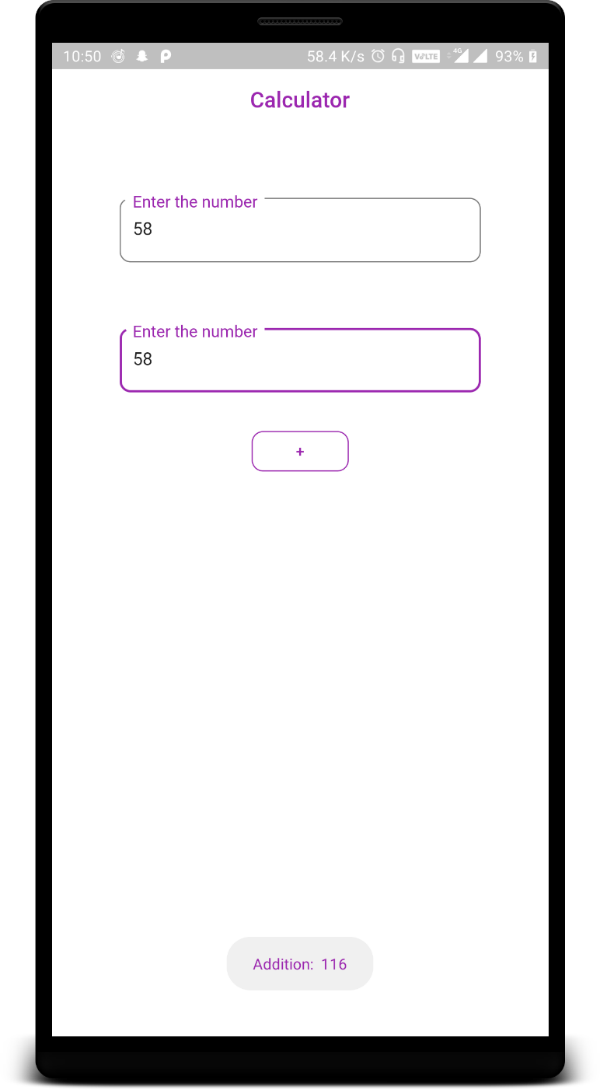
**Practical 3**

**AIM: Create an application that will display Toast (Message) on specific interval of time.Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:fluttertoast/fluttertoast.dart';  void main() => runApp(MyApp());  class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  debugShowCheckedModeBanner: false,  title: 'Flutter Demo',  theme: ThemeData(primaryColor: Colors.purple, cursorColor: Colors.purple),  home: MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {  @override  \_MyHomePageState createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {  final TextEditingController textEditingController =  new TextEditingController();  final TextEditingController textEditingControllerOne =  new TextEditingController();  int sum, num1, num2;  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: new AppBar(  title: new Text("Calculator", style: TextStyle(color: Colors.purple)),  centerTitle: true,  elevation: 0.0,  backgroundColor: Colors.white,  ),  body: new Container(  color: Colors.white,  child: new Padding(  padding: EdgeInsets.all(32.0),  child: new Column(  children: <Widget>[  new Padding(  padding: EdgeInsets.all(30.0),  child: new TextField(  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Enter the number",  labelStyle:  new TextStyle(color: Colors.purple, fontSize: 20.0),  border: OutlineInputBorder(  borderRadius: new BorderRadius.circular(10.0)),  ),  controller: textEditingController,  ),  ),  new Padding(  padding: EdgeInsets.all(30.0),  child: new TextField(  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Enter the number",  labelStyle:  new TextStyle(color: Colors.purple, fontSize: 20.0),  border: OutlineInputBorder(  borderRadius: new BorderRadius.circular(10.0)),  ),  controller: textEditingControllerOne,  ),  ),  new Row(  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  children: <Widget>[  new OutlineButton(  child: new Text("+"),  borderSide: new BorderSide(color: Colors.purple),  shape: new RoundedRectangleBorder(  borderRadius: new BorderRadius.circular(10.0)),  textColor: Colors.purple,  padding: EdgeInsets.all(10),  onPressed: doAddition,  ),  ],  ),  ],  ),  ),  ),  );  }  void doAddition() {  num1 = int.parse(textEditingController.text);  num2 = int.parse(textEditingControllerOne.text);  sum = num1 + num2;  Fluttertoast.showToast(  msg: "Addition: $sum",  toastLength: Toast.LENGTH\_LONG,  gravity: ToastGravity.BOTTOM,  backgroundColor: Colors.black,  textColor: Colors.yellow);  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can able to implement that shows the length of toast with specific interval of time.**

**Practical 4**

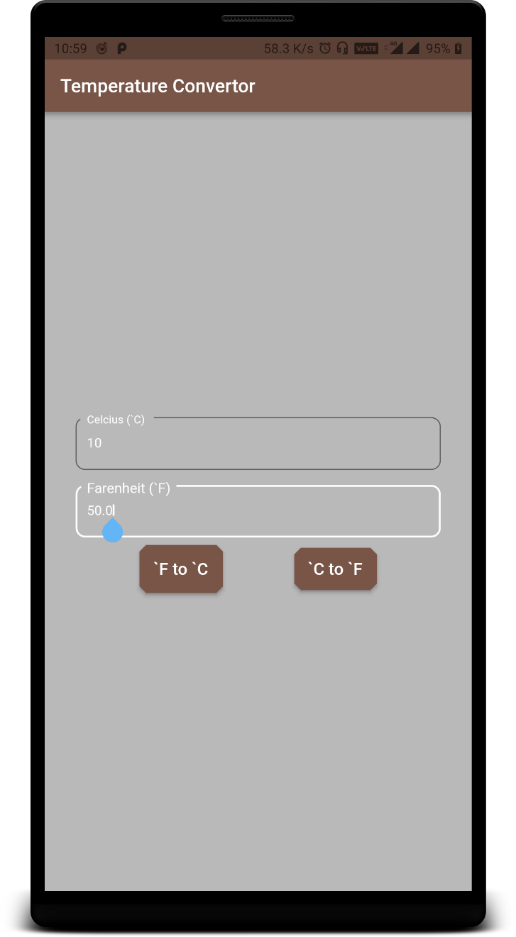
**AIM: Create a temperature converter Application. (Fahrenheit-Celsius)**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:flutter/services.dart';  void main() {  SystemChrome.setSystemUIOverlayStyle(  SystemUiOverlayStyle(statusBarIconBrightness: Brightness.light));  runApp(MyApp());  }  class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  debugShowCheckedModeBanner: false,  title: 'Temperature Convertor',  theme: ThemeData(  primaryColor: Colors.white,  cursorColor: Colors.white,  ),  home: MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {  @override  \_MyHomePageState createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {  final TextEditingController textEditingController =  new TextEditingController();  final TextEditingController textEditingControllerOne =  new TextEditingController();  double celcius, farenheit;  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(  title: Text(  "Temperature Convertor",  style: TextStyle(  color: Colors.white,  ),  textAlign: TextAlign.center,  ),  backgroundColor: Colors.brown,  ),  body: new Container(  color: Colors.black26,  child: Padding(  padding: EdgeInsets.all(25.0),  child: Column(  children: <Widget>[  Padding(  padding: EdgeInsets.all(8.0),  child: TextField(  style: TextStyle(color: Colors.white),  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Celcius (`C) ",  labelStyle: new TextStyle(  color: Colors.white, fontStyle: FontStyle.normal),  border: OutlineInputBorder(  borderRadius: BorderRadius.circular(10.0),  ),  ),  controller: textEditingController,  ),  ),  Padding(  padding: EdgeInsets.all(8.0),  child: TextField(  style: TextStyle(color: Colors.white),  keyboardType: TextInputType.number,  decoration: new InputDecoration(  labelText: "Farenheit (`F)",  labelStyle: new TextStyle(  fontStyle: FontStyle.normal,  fontSize: 20.0,  color: Colors.white),  border: new OutlineInputBorder(  borderRadius: BorderRadius.circular(10.0))),  controller: textEditingControllerOne,  ),  ),  Row(  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  children: <Widget>[  RaisedButton(  onPressed: fareToCel,  child: Text(  "`F to `C",  style: TextStyle(fontSize: 18.0),  ),  elevation: 3.0,  padding: EdgeInsets.all(15.0),  color: Colors.brown,  splashColor: Colors.blueGrey,  shape: BeveledRectangleBorder(  borderRadius: BorderRadius.circular(7.0)),  textColor: Colors.white,  ),  RaisedButton(  onPressed: celToFare,  child: Text(  "`C to `F",  style: TextStyle(fontSize: 18.0),  ),  elevation: 3.0,  shape: BeveledRectangleBorder(  borderRadius: BorderRadius.circular(7.0)),  splashColor: Colors.blueGrey,  padding: EdgeInsets.all(12.0),  color: Colors.brown,  textColor: Colors.white,  )  ],  )  ],  ),  ),  ));  }  void fareToCel() {  farenheit = double.parse(textEditingControllerOne.text);  farenheit = ((32 \* farenheit - 32) \* (5 / 9));  setState(() {  textEditingController.text = farenheit.toString();  });  }  void celToFare() {  celcius = double.parse(textEditingController.text);  celcius = (((9 / 5) \* celcius) + 32);  setState(() {  textEditingControllerOne.text = celcius.toString();  });  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can able to make any formula containing application in android as well as flutter.**

**Practical 5**

**AIM: Create a login application with following features:**

**1. Successful Login message in TextView with Green background if Username & password is correct**

**2. Failure message in TextView with Red background if Username or password is incorrect.**

**3. Disable Login Button after three wrong login attempts.**

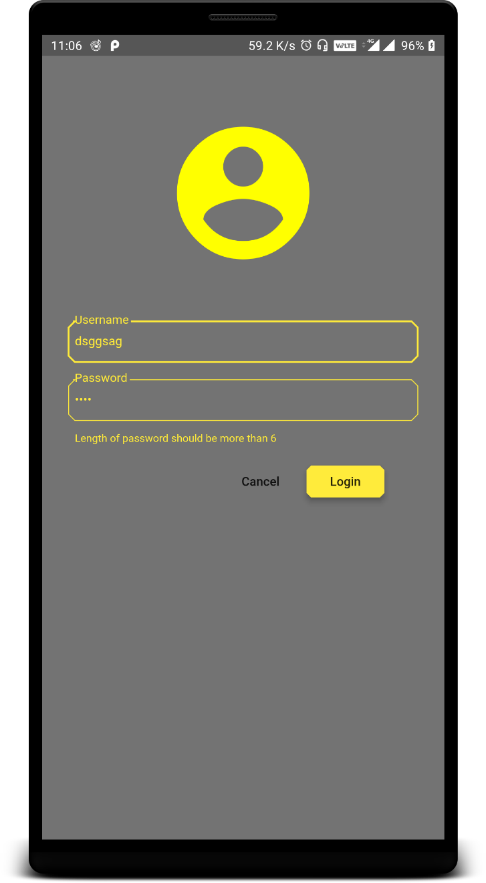
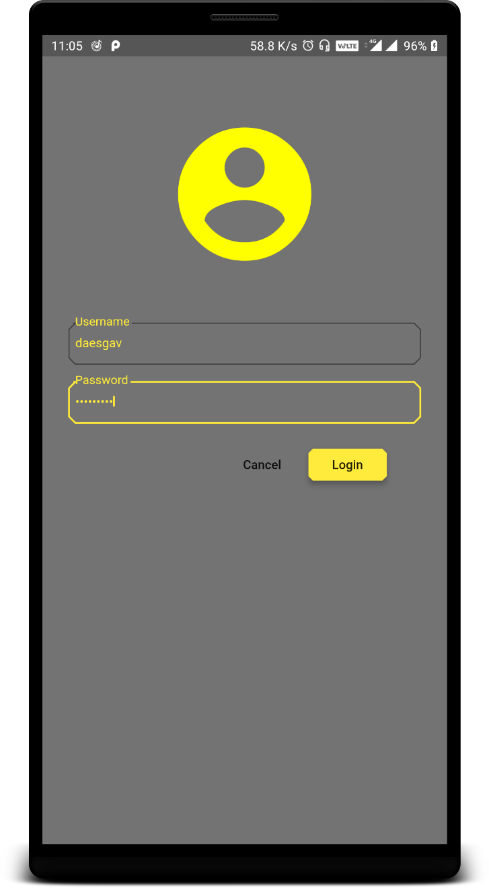
**4. Close application if user selects Cancel Button.**

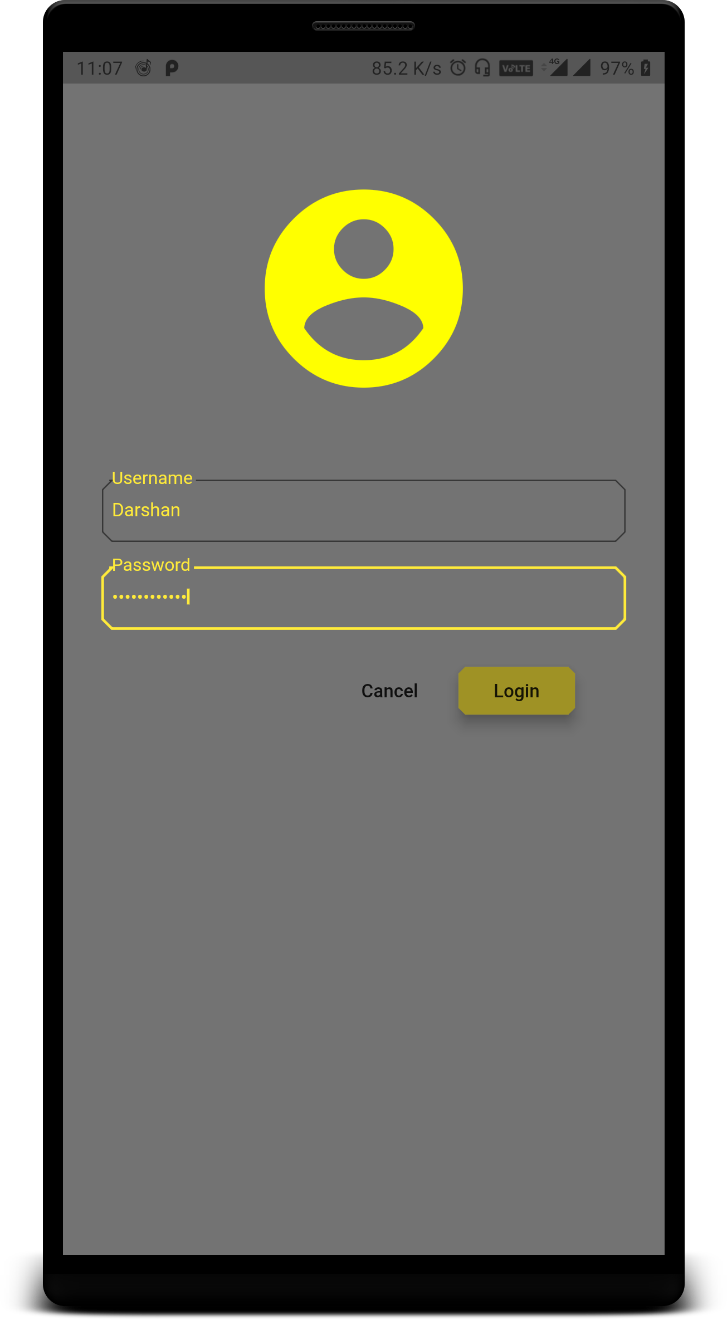
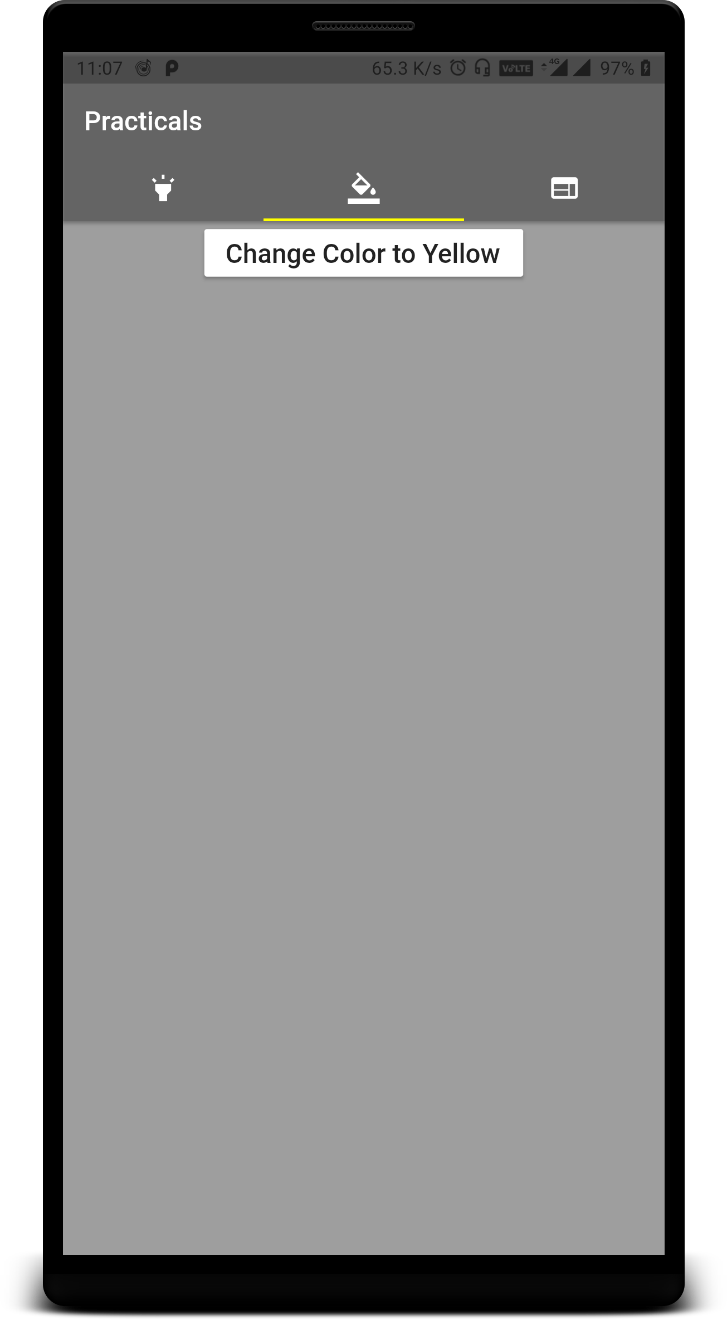
**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'cut\_corner\_border.dart';  import 'first\_screen.dart';  import 'second\_screen.dart';  import 'package:flutter/services.dart';  void main() {  SystemChrome.setPreferredOrientations([  DeviceOrientation.portraitUp,  ]);  runApp(MyApp());  }  class MyApp extends StatelessWidget {  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  debugShowCheckedModeBanner: false,  title: 'Flutter Demo',  theme: ThemeData(  accentColor: Colors.yellowAccent,  primaryColor: Colors.yellow,  cursorColor: Colors.yellow,  errorColor: Colors.yellow),  home: MyHomePage(title: 'Login'),  routes: <String, WidgetBuilder>{  '/first\_screen': (context) => new FirstClass('First'),  '/second\_screen': (context) => new SecondClass('Second')  },  );  }  }  class MyHomePage extends StatefulWidget {  MyHomePage({Key key, this.title}) : super(key: key);  final String title;  @override  \_MyHomePageState createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {  final TextEditingController textEditingController =  new TextEditingController();  final TextEditingController textEditingControllerOne =  new TextEditingController();  final \_formKey = GlobalKey<FormState>();  @override  Widget build(BuildContext context) {  return Scaffold(  body: Container(  color: Colors.black54,  child: ListView(  scrollDirection: Axis.vertical,  children: <Widget>[  Padding(  padding: EdgeInsets.all(15.0),  child: Form(  key: \_formKey,  child: Column(  mainAxisSize: MainAxisSize.max,  children: <Widget>[  Padding(  padding: EdgeInsets.fromLTRB(0, 50, 0, 0),  child: Icon(Icons.account\_circle,  color: Colors.yellowAccent, size: 180.0),  ),  Padding(  padding: EdgeInsets.fromLTRB(10, 50, 10, 0),  child: TextFormField(  style: TextStyle(color: Colors.yellow),  keyboardType: TextInputType.text,  decoration: new InputDecoration(  labelText: "Username",  labelStyle: new TextStyle(  color: Colors.yellow, fontSize: 18.0),  border: CutCornersBorder(  borderRadius: BorderRadius.circular(5.0)),  ),  controller: textEditingController,  validator: (value) {  if (value.isEmpty &&  textEditingController.text != "Darshan") {  return 'Enter valid email';  }  },  ),  ),  Padding(  padding: EdgeInsets.fromLTRB(10, 10, 10, 0),  child: TextFormField(  style: TextStyle(color: Colors.yellow),  keyboardType: TextInputType.text,  obscureText: true,  controller: textEditingControllerOne,  decoration: new InputDecoration(  labelText: "Password",  labelStyle: new TextStyle(  color: Colors.yellow, fontSize: 18.0),  border: CutCornersBorder(  borderRadius: BorderRadius.circular(5.0))),  validator: (value) {  if (value.length < 6 &&  textEditingControllerOne.text != "HelloDarshan") {  return 'Length of password should be more than 6';  }  },  ),  ),  Padding(  padding: EdgeInsets.fromLTRB(180, 10, 0, 0),  child: Row(  children: <Widget>[  Padding(  padding: EdgeInsets.all(8.0),  child: FlatButton(  child: Text("Cancel"),  onPressed: () {},  ),  ),  RaisedButton(  child: Text("Login",  style: new TextStyle(color: Colors.black87)),  color: Colors.yellow,  splashColor: Colors.black26,  shape: BeveledRectangleBorder(  borderRadius:  BorderRadius.all(Radius.circular(5.0))),  elevation: 5.0,  onPressed: () {  if (\_formKey.currentState.validate()) {  if ((textEditingController.text == "Darshan") &&  (textEditingControllerOne.text ==  "HelloDarshan")) {  Navigator.of(context)  .pushNamed('/first\_screen');  }  }  },  )  ],  ),  ),  ],  ),  ),  ),  ],  ),  ),  );  }  } |

**Output:**

****

** **

**Conclusion: By performing this experiment, one can able to make any login screen with its validation technique using full material design components in android as well as flutter.**

**Practical 6**

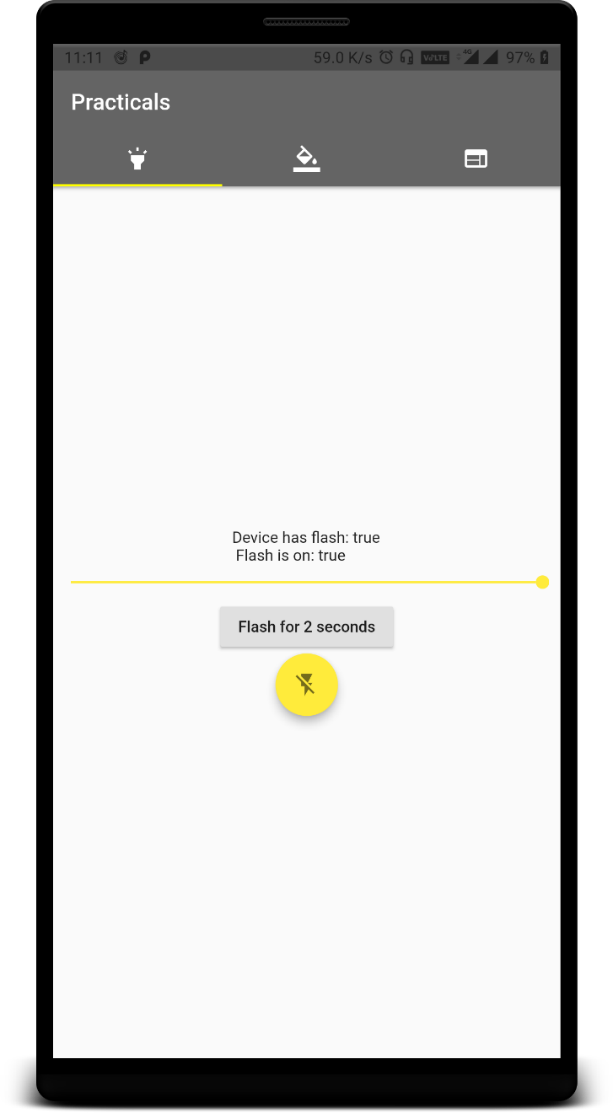
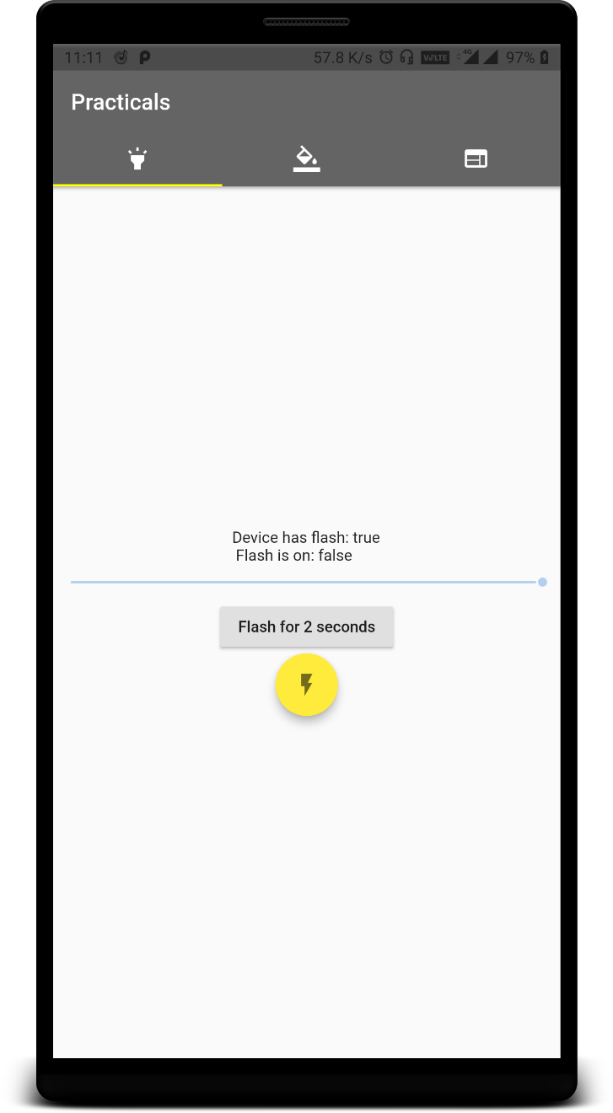
**AIM: Create an application which turns ON or OFF Torch/Flashlight of Camera.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:lamp/lamp.dart';  import 'package:webview\_flutter/webview\_flutter.dart';  import 'dart:async';  class FirstClass extends StatefulWidget {  String title;  @override  State<StatefulWidget> createState() {  return FirstPageTest(title);  }  FirstClass(String s);  }  class FirstPageTest extends State<FirstClass> {  bool \_hasFlash = false;  bool \_isOn = false;  double \_intensity = 1.0;  bool pressed = true;  FirstPageTest(String title);  @override  initState() {  super.initState();  initPlatformState();  }  initPlatformState() async {  bool hasFlash = await Lamp.hasLamp;  print("Device has flash ? $hasFlash");  setState(() {  \_hasFlash = hasFlash;  });  }  @override  Widget build(BuildContext context) {  \_intensityChanged(double intensity) {  Lamp.turnOn(intensity: intensity);  setState(() {  \_intensity = intensity;  });  }  Future \_turnFlash() async {  \_isOn ? Lamp.turnOff() : Lamp.turnOn(intensity: \_intensity);  var f = await Lamp.hasLamp;  setState(() {  \_hasFlash = f;  \_isOn = !\_isOn;  });  }  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  bottom: TabBar(  tabs: <Widget>[  Tab(icon: Icon(Icons.highlight, color: Colors.white)),  Tab(  icon: Icon(  Icons.format\_color\_fill,  color: Colors.white,  )),  Tab(  icon: Icon(  Icons.web,  color: Colors.white,  ))  ],  ),  title: Text("Practicals", style: TextStyle(color: Colors.white)),  backgroundColor: Colors.black45,  automaticallyImplyLeading: false,  elevation: 2.0,  ),  body: TabBarView(children: [  new Container(  child: Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: <Widget>[  new Text(  'Device has flash: $\_hasFlash\n Flash is on: $\_isOn'),  new Slider(  value: \_intensity,  onChanged: \_isOn ? \_intensityChanged : null),  new RaisedButton(  onPressed: () async =>  await Lamp.flash(new Duration(seconds: 2)),  child: new Text("Flash for 2 seconds")),  new FloatingActionButton(  child: Icon(\_isOn ? Icons.flash\_off : Icons.flash\_on),  backgroundColor: Colors.yellow,  foregroundColor: Colors.black54,  onPressed: \_turnFlash,  ),  ],  ),  ),  ),  ]),  ),  );  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can able to operate the hardware such as flashlight with an app developed in android or flutter.**

**Practical 7**

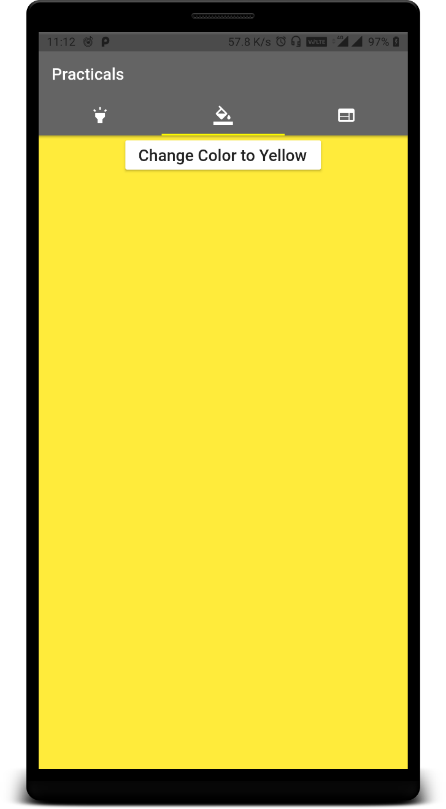
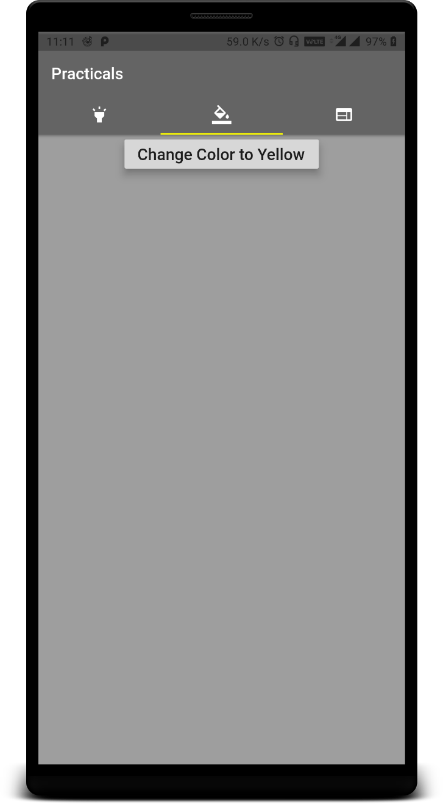
**AIM: Create an application that will change color of the screen, based on selected options from the menu.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  class FirstClass extends StatefulWidget {  String title;  @override  State<StatefulWidget> createState() {  return FirstPageTest(title);  }  FirstClass(String s);  }  class FirstPageTest extends State<FirstClass> {  FirstPageTest(String title);  @override  Widget build(BuildContext context) {  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  bottom: TabBar(  tabs: <Widget>[  Tab(icon: Icon(Icons.highlight, color: Colors.white)),  Tab(  icon: Icon(  Icons.format\_color\_fill,  color: Colors.white,  )),  Tab(  icon: Icon(  Icons.web,  color: Colors.white,  ))  ],  ),  title: Text("Practicals", style: TextStyle(color: Colors.white)),  backgroundColor: Colors.black45,  automaticallyImplyLeading: false,  elevation: 2.0,  ),  body: TabBarView(children: [  new Container(  color: pressed ? Colors.grey : Colors.yellow,  child: Column(  children: <Widget>[  MaterialButton(  child: Text(  "Change Color to Yellow",  style: TextStyle(fontSize: 20.0),  ),  color: Colors.white,  onPressed: () {  setState(() {  pressed = !pressed;  });  })  ],  ),  ),  ]),  ),  );  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can perform the dynamic layout changes in flutter or an android app.**

**Practical 8**

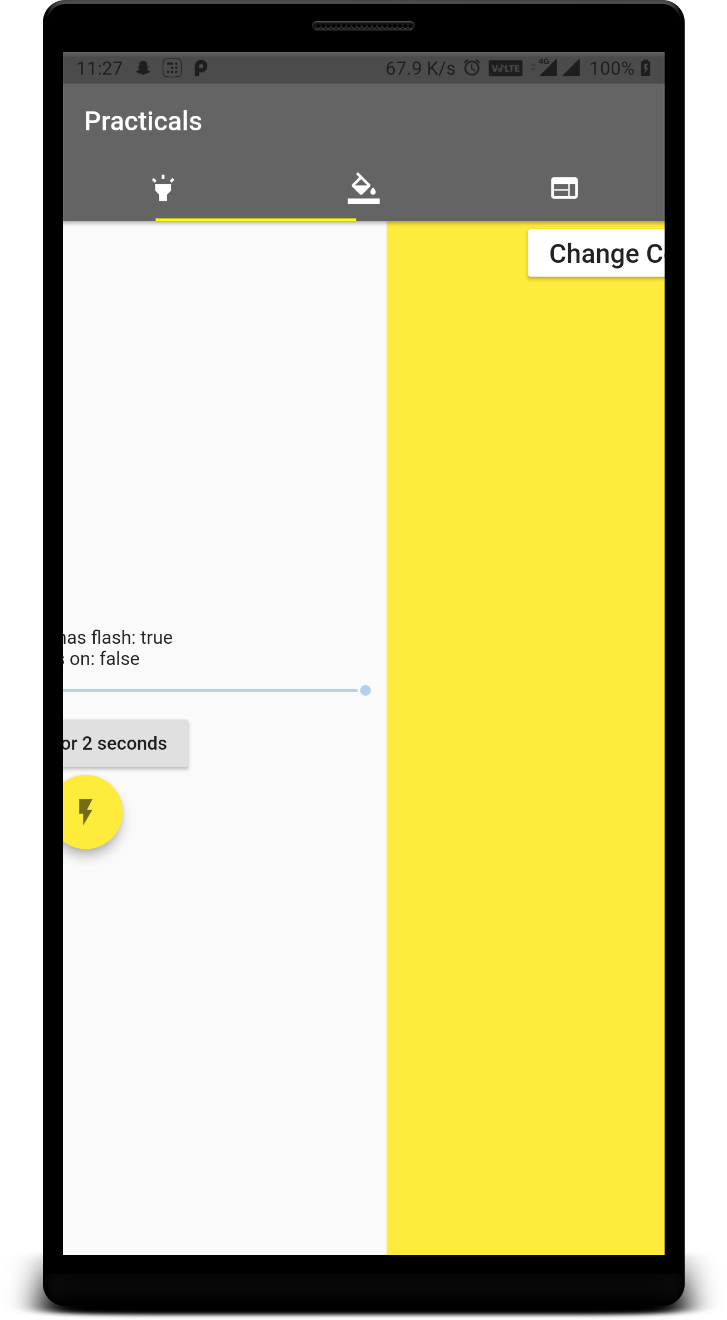
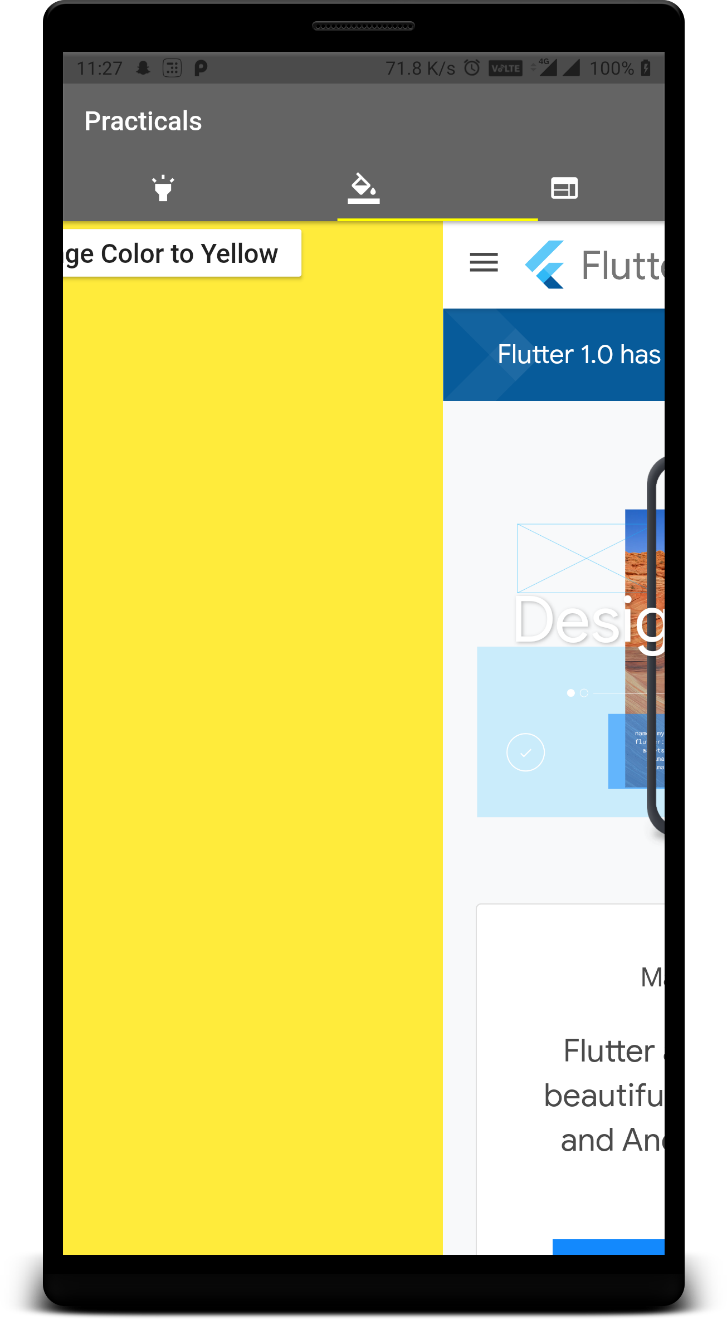
**AIM: Create an application with the help of fragment.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:lamp/lamp.dart';  import 'package:webview\_flutter/webview\_flutter.dart';  import 'dart:async';  class FirstClass extends StatefulWidget {  String title;  @override  State<StatefulWidget> createState() {  return FirstPageTest(title);  }  FirstClass(String s);  }  class FirstPageTest extends State<FirstClass> {  bool \_hasFlash = false;  bool \_isOn = false;  double \_intensity = 1.0;  bool pressed = true;  FirstPageTest(String title);  @override  initState() {  super.initState();  initPlatformState();  }  initPlatformState() async {  bool hasFlash = await Lamp.hasLamp;  print("Device has flash ? $hasFlash");  setState(() {  \_hasFlash = hasFlash;  });  }  @override  Widget build(BuildContext context) {  \_intensityChanged(double intensity) {  Lamp.turnOn(intensity: intensity);  setState(() {  \_intensity = intensity;  });  }  Future \_turnFlash() async {  \_isOn ? Lamp.turnOff() : Lamp.turnOn(intensity: \_intensity);  var f = await Lamp.hasLamp;  setState(() {  \_hasFlash = f;  \_isOn = !\_isOn;  });  }  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  bottom: TabBar(  tabs: <Widget>[  Tab(icon: Icon(Icons.highlight, color: Colors.white)),  Tab(  icon: Icon(  Icons.format\_color\_fill,  color: Colors.white,  )),  Tab(  icon: Icon(  Icons.web,  color: Colors.white,  ))  ],  ),  title: Text("Practicals", style: TextStyle(color: Colors.white)),  backgroundColor: Colors.black45,  automaticallyImplyLeading: false,  elevation: 2.0,  ),  body: TabBarView(children: [  new Container(  child: Center(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: <Widget>[  new Text(  'Device has flash: $\_hasFlash\n Flash is on: $\_isOn'),  new Slider(  value: \_intensity,  onChanged: \_isOn ? \_intensityChanged : null),  new RaisedButton(  onPressed: () async =>  await Lamp.flash(new Duration(seconds: 2)),  child: new Text("Flash for 2 seconds")),  new FloatingActionButton(  child: Icon(\_isOn ? Icons.flash\_off : Icons.flash\_on),  backgroundColor: Colors.yellow,  foregroundColor: Colors.black54,  onPressed: \_turnFlash,  ),  ],  ),  ),  ),  new Container(  color: pressed ? Colors.grey : Colors.yellow,  child: Column(  children: <Widget>[  MaterialButton(  child: Text(  "Change Color to Yellow",  style: TextStyle(fontSize: 20.0),  ),  color: Colors.white,  onPressed: () {  setState(() {  pressed = !pressed;  });  })  ],  ),  ),  new Container(  child: WebView(  initialUrl: "https://flutter.io",  ),  ),  ]),  ),  );  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can know the concept of the fragments in android and can be able to develop the application that uses the fragment concept.**

**Practical 9**

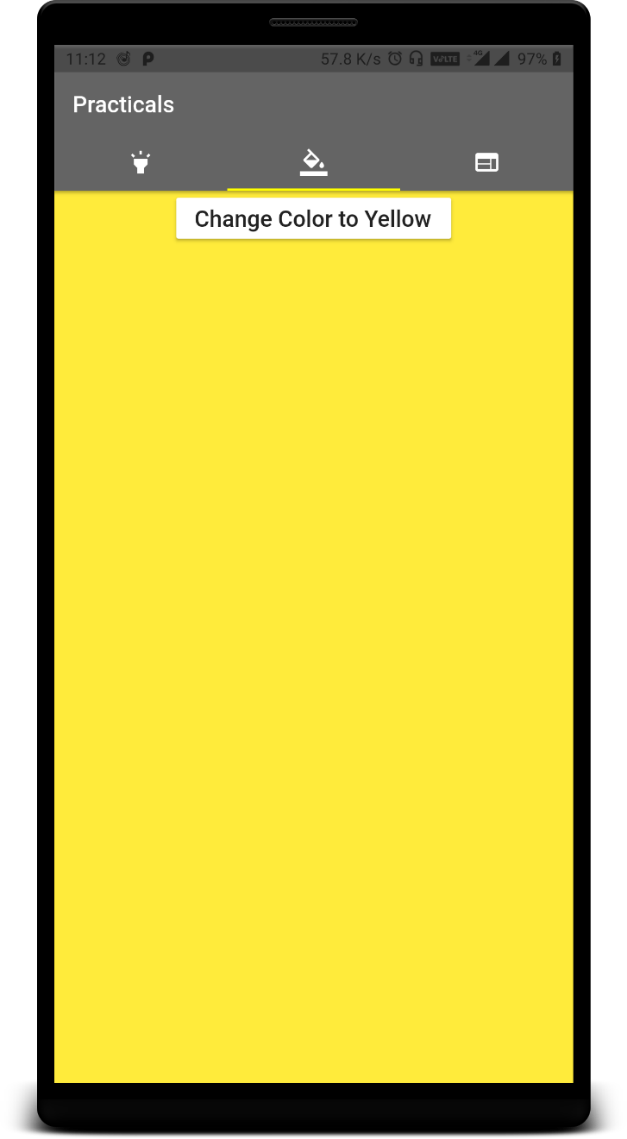
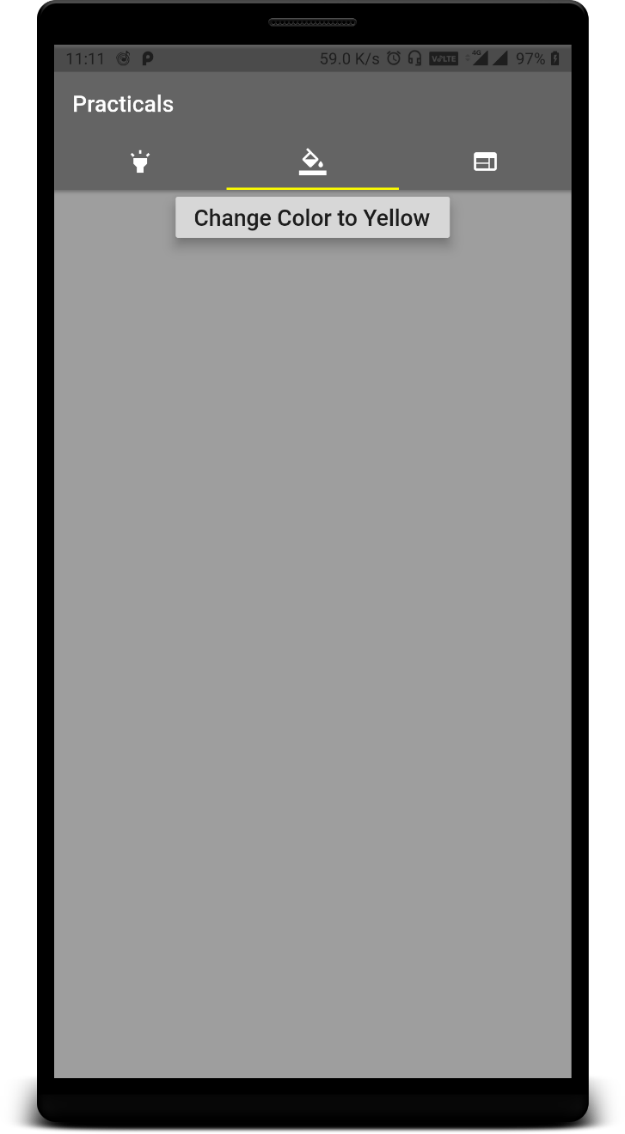
**AIM: Create an application that will change color of the screen, based on selected options from the menu.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  class FirstClass extends StatefulWidget {  String title;  @override  State<StatefulWidget> createState() {  return FirstPageTest(title);  }  FirstClass(String s);  }  class FirstPageTest extends State<FirstClass> {  FirstPageTest(String title);  @override  Widget build(BuildContext context) {  return DefaultTabController(  length: 3,  child: Scaffold(  appBar: AppBar(  bottom: TabBar(  tabs: <Widget>[  Tab(icon: Icon(Icons.highlight, color: Colors.white)),  Tab(  icon: Icon(  Icons.format\_color\_fill,  color: Colors.white,  )),  Tab(  icon: Icon(  Icons.web,  color: Colors.white,  ))  ],  ),  title: Text("Practicals", style: TextStyle(color: Colors.white)),  backgroundColor: Colors.black45,  automaticallyImplyLeading: false,  elevation: 2.0,  ),  body: TabBarView(children: [  new Container(  child: WebView(  initialUrl: "https://flutter.io",  ),  ],  ),  ),  ]),  ),  );  }  } |

**Output:**

****

**Conclusion: By performing this experiment, one can perform the dynamic layout changes in flutter or an android app.**

**Practical 10**

**AIM: Create an application with the help of database.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:flutter\_database/pages/home.dart';  import 'package:flutter\_database/pages/login/login.dart';  void main() => runApp(new MyApp());  final routes = {  '/login': (BuildContext context) => new LoginPage(),  '/home': (BuildContext context) => new HomePage(),  '/': (BuildContext context) => new LoginPage(),  };  class MyApp extends StatelessWidget {  @override  Widget build(BuildContext context) {  return new MaterialApp(  title: 'Sqflite App',  theme: new ThemeData(primarySwatch: Colors.teal),  routes: routes,  );  }  } |

**login.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:flutter\_database/data/database\_helper.dart';  import 'package:flutter\_database/model/user.dart';  import 'package:flutter\_database/pages/login/login\_presenter.dart';  class LoginPage extends StatefulWidget {  @override  \_LoginPageState createState() => new \_LoginPageState();  }  class \_LoginPageState extends State<LoginPage> implements LoginPageContract {  BuildContext \_ctx;  bool \_isLoading = false;  final formKey = new GlobalKey<FormState>();  final scaffoldKey = new GlobalKey<ScaffoldState>();  String \_username, \_password;  LoginPagePresenter \_presenter;  \_LoginPageState() {  \_presenter = new LoginPagePresenter(this);  }  void \_submit() {  final form = formKey.currentState;  if (form.validate()) {  setState(() {  \_isLoading = true;  form.save();  \_presenter.doLogin(\_username, \_password);  });  }  }  void \_showSnackBar(String text) {  scaffoldKey.currentState.showSnackBar(new SnackBar(  content: new Text(text),  ));  }  @override  Widget build(BuildContext context) {  \_ctx = context;  var loginBtn = new RaisedButton(  onPressed: \_submit,  child: new Text("Login"),  color: Colors.green,  );  var loginForm = new Column(  crossAxisAlignment: CrossAxisAlignment.center,  children: <Widget>[  new Text(  "Sqflite App Login",  textScaleFactor: 2.0,  ),  new Form(  key: formKey,  child: new Column(  children: <Widget>[  new Padding(  padding: const EdgeInsets.all(10.0),  child: new TextFormField(  onSaved: (val) => \_username = val,  decoration: new InputDecoration(labelText: "Username"),  ),  ),  new Padding(  padding: const EdgeInsets.all(10.0),  child: new TextFormField(  onSaved: (val) => \_password = val,  decoration: new InputDecoration(labelText: "Password"),  ),  )  ],  ),  ),  loginBtn  ],  );  return new Scaffold(  appBar: new AppBar(  title: new Text("Login Page"),  ),  key: scaffoldKey,  body: new Container(  child: new Center(  child: loginForm,  ),  ),  );  }  @override  void onLoginError(String error) {  // TODO: implement onLoginError  \_showSnackBar(error);  setState(() {  \_isLoading = false;  });  }  @override  void onLoginSuccess(User user) async {  // TODO: implement onLoginSuccess  \_showSnackBar(user.toString());  setState(() {  \_isLoading = false;  });  var db = new DatabaseHelper();  await db.saveUser(user);  Navigator.of(context).pushNamed("/home");  }  } |

**login\_presenter.dart**

|  |
| --- |
| import 'package:flutter\_database/data/rest\_data.dart';  import 'package:flutter\_database/model/user.dart';  abstract class LoginPageContract {  void onLoginSuccess(User user);  void onLoginError(String error);  }  class LoginPagePresenter {  LoginPageContract \_view;  RestData api = new RestData();  LoginPagePresenter(this.\_view);  doLogin(String username, String password) {  api  .login(username, password)  .then((user) => \_view.onLoginSuccess(user))  .catchError((onError) => \_view.onLoginError(onError.toString()));  }  } |

**database\_helper.dart**

|  |
| --- |
| import 'dart:io';  import 'package:flutter\_database/model/user.dart';  import 'package:path/path.dart';  import 'dart:async';  import 'package:path\_provider/path\_provider.dart';  import 'package:sqflite/sqflite.dart';  class DatabaseHelper {  static final DatabaseHelper \_instance = new DatabaseHelper.internal();  factory DatabaseHelper() => \_instance;  static Database \_db;  Future<Database> get db async {  if (\_db != null) {  return \_db;  }  \_db = await initDb();  return \_db;  }  DatabaseHelper.internal();  initDb() async {  Directory documentDirectory = await getApplicationDocumentsDirectory();  String path = join(documentDirectory.path, "main.db");  var ourDb = await openDatabase(path, version: 1, onCreate: \_onCreate);  return ourDb;  }  void \_onCreate(Database db, int version) async {  await db.execute(  "CREATE TABLE User(id INTEGER PRIMARY KEY, username TEXT, password TEXT)");  print("Table is created");  }  //insertion  Future<int> saveUser(User user) async {  var dbClient = await db;  int res = await dbClient.insert("User", user.toMap());  return res;  }  //deletion  Future<int> deleteUser(User user) async {  var dbClient = await db;  int res = await dbClient.delete("User");  return res;  }  } |

**rest\_data.dart**

|  |
| --- |
| import 'dart:async';  import 'package:flutter\_database/model/user.dart';  import 'package:flutter\_database/utils/network\_util.dart';  class RestData {  NetworkUtil \_netUtil = new NetworkUtil();  static final BASE\_URL = "";  static final LOGIN\_URL = BASE\_URL + "/";  Future<User> login(String username, String password) {  return new Future.value(new User(username, password));  }  } |

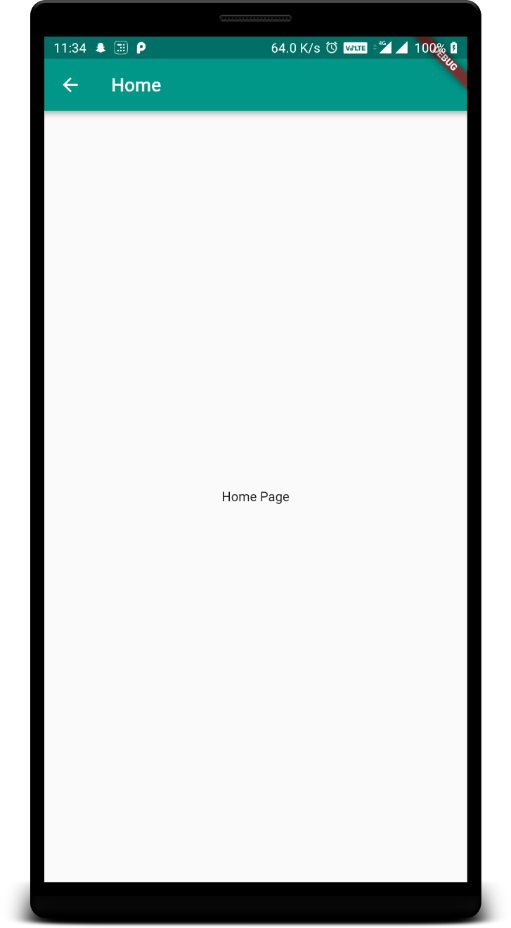
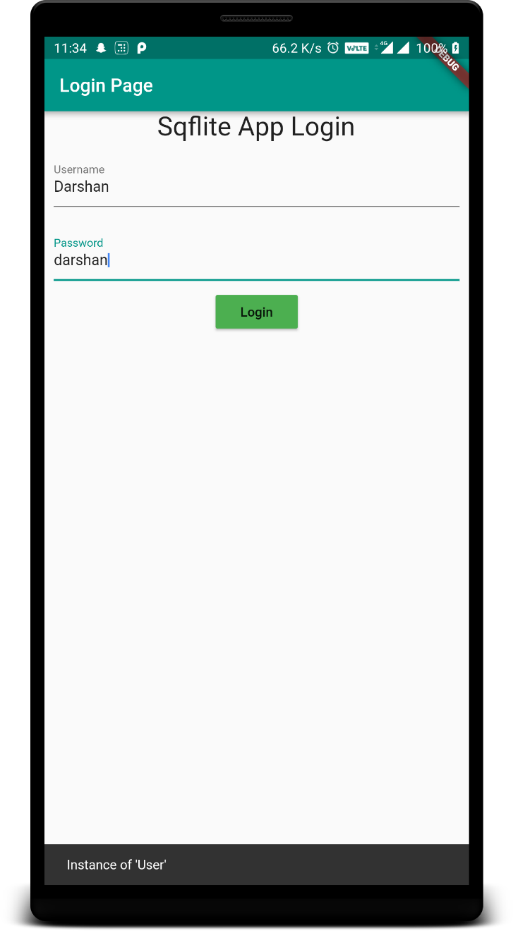
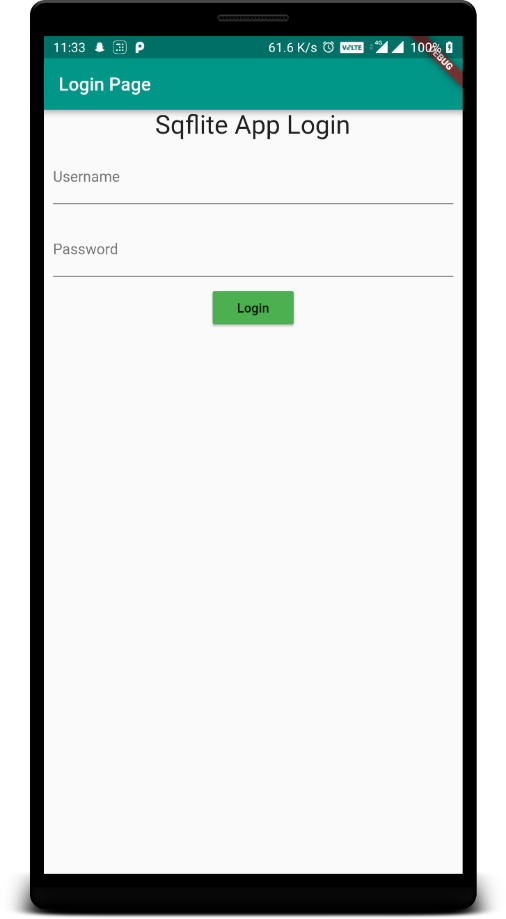
**user.dart**

|  |
| --- |
| class User {  String \_username;  String \_password;  User(this.\_username, this.\_password);  User.map(dynamic obj) {  this.\_username = obj['username'];  this.\_password = obj['password'];  }  String get username => \_username;  String get password => \_password;  Map<String, dynamic> toMap() {  var map = new Map<String, dynamic>();  map["username"] = \_username;  map["password"] = \_password;  return map;  }  } |

**network\_util.dart**

|  |
| --- |
| import 'dart:async';  class NetworkUtil {  static NetworkUtil \_instance = new NetworkUtil.internal();  NetworkUtil.internal();  factory NetworkUtil() => \_instance;  Future<dynamic> get() {  return null;  }  } |

**Output:**

****

**Conclusion: By performing above experiment, one can able to create the application using database in android and flutter.**

**Practical 11**

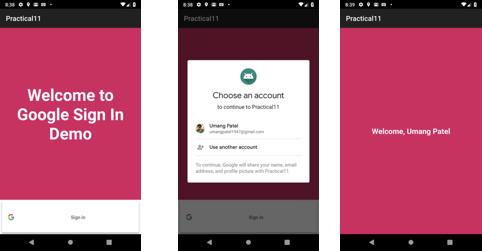
**AIM: Create an application that will change color of the screen, based on selected options from the menu.**

**Source Code:**

**main.dart**

|  |
| --- |
| ***activity\_login.xml :***  <?xml version="1.0" encoding="utf-8"?>  <androidx.appcompat.widget.LinearLayoutCompat xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:orientation="vertical"      tools:context=".LoginActivity">        <androidx.appcompat.widget.AppCompatTextView          android:layout\_width="match\_parent"          android:layout\_height="0dp"          android:layout\_weight="5"          android:background="@color/colorAccent"          android:gravity="center"          android:text="@string/welcome\_to\_google\_sign\_in\_demo\_text"          android:textColor="@android:color/white"          android:textSize="48sp"          android:textStyle="bold"          app:layout\_constraintBottom\_toBottomOf="parent"          app:layout\_constraintLeft\_toLeftOf="parent"          app:layout\_constraintRight\_toRightOf="parent"          app:layout\_constraintTop\_toTopOf="parent" />      <com.google.android.gms.common.SignInButton          android:id="@+id/login\_button"          android:layout\_width="match\_parent"          android:layout\_height="0dp"          android:layout\_weight="1" />  </androidx.appcompat.widget.LinearLayoutCompat>  ***activity\_home.xml :***  <?xml version="1.0" encoding="utf-8"?>  <androidx.appcompat.widget.LinearLayoutCompat xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      android:background="@color/colorAccent"      android:orientation="vertical"      tools:context=".HomeActivity">      <androidx.appcompat.widget.AppCompatTextView          android:id="@+id/display\_name\_text\_view"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent"          android:gravity="center"          android:textAppearance="?android:textAppearanceLarge"          android:textColor="@android:color/white"          android:textStyle="bold"          tools:text="Hello, Umang" />  </androidx.appcompat.widget.LinearLayoutCompat>  **Java files :**  ***LoginActivity.java :***  package com.android.example.practical11;  import android.content.Intent;  import android.os.Bundle;  import android.util.Log;  import android.view.View;  import android.widget.Toast;  import com.google.android.gms.auth.api.signin.GoogleSignIn;  import com.google.android.gms.auth.api.signin.GoogleSignInAccount;  import com.google.android.gms.auth.api.signin.GoogleSignInClient;  import com.google.android.gms.auth.api.signin.GoogleSignInOptions;  import com.google.android.gms.common.api.ApiException;  import com.google.android.gms.tasks.OnCompleteListener;  import com.google.android.gms.tasks.Task;  import com.google.firebase.auth.AuthCredential;  import com.google.firebase.auth.AuthResult;  import com.google.firebase.auth.FirebaseAuth;  import com.google.firebase.auth.FirebaseUser;  import com.google.firebase.auth.GoogleAuthProvider;  import java.util.Objects;  import androidx.annotation.NonNull;  import androidx.annotation.Nullable;  import androidx.appcompat.app.AppCompatActivity;  public class LoginActivity extends AppCompatActivity implements View.OnClickListener {      private static final int RC\_SIGN\_IN = 10;      private GoogleSignInClient mGoogleSignInClient;      private FirebaseAuth mAuth;      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_login);          GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)                  .requestIdToken(getString(R.string.default\_web\_client\_id))                  .requestEmail()                  .build();          mGoogleSignInClient = GoogleSignIn.getClient(this, gso);          mAuth = FirebaseAuth.getInstance();          findViewById(R.id.login\_button).setOnClickListener(this);      }      @Override      protected void onStart() {          super.onStart();          // Check for existing Google Sign In account, if the user is already signed in          // the GoogleSignInAccount will be non-null.          FirebaseUser currentUser = mAuth.getCurrentUser();          updateUI(currentUser);      }      @Override      public void onClick(View v) {          signIn();      }      private void signIn() {          Intent signInIntent = mGoogleSignInClient.getSignInIntent();          startActivityForResult(signInIntent, RC\_SIGN\_IN);      }      @Override      protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {          super.onActivityResult(requestCode, resultCode, data);          // Result returned from launching the Intent from GoogleSignInApi.getSignInIntent(...);          if (requestCode == RC\_SIGN\_IN) {              Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);              try {                  // Google Sign In was successful, authenticate with Firebase                  GoogleSignInAccount account = task.getResult(ApiException.class);                  firebaseAuthWithGoogle(Objects.requireNonNull(account));              } catch (ApiException e) {                  // Google Sign In failed, update UI appropriately                  Log.e("GSignIn", "Google sign in failed", e);              }          }      }      private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {          Log.d("GsignIn", "firebaseAuthWithGoogle:" + acct.getId());          AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);          mAuth.signInWithCredential(credential)                  .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {                      @Override                      public void onComplete(@NonNull Task<AuthResult> task) {                          if (task.isSuccessful()) {                              // Sign in success, update UI with the signed-in user's information                              Log.d("GSignIn", "signInWithCredential:success");                              FirebaseUser user = mAuth.getCurrentUser();                              updateUI(user);                          } else {                              // If sign in fails, display a message to the user.                              Log.e("GSignIn", "signInWithCredential:failure", task.getException());                              Toast.makeText(LoginActivity.this, "Authentication Failed", Toast.LENGTH\_SHORT).show();                              updateUI(null);                          }                      }                  });      }      private void updateUI(FirebaseUser user) {          if (user != null) {              startActivity(HomeActivity.newIntent(LoginActivity.this, user));              finish();          }      }  }  ***HomeActivity.java :***  package com.android.example.practical11;  import android.content.Context;  import android.content.Intent;  import android.os.Bundle;  import com.google.firebase.auth.FirebaseUser;  import androidx.appcompat.app.AppCompatActivity;  import androidx.appcompat.widget.AppCompatTextView;  public class HomeActivity extends AppCompatActivity {      private static final String EXTRA\_DISPLAY\_NAME = "user\_display\_name";      public static Intent newIntent(Context packageContext, FirebaseUser user) {          Intent intent = new Intent(packageContext, HomeActivity.class);          String userDisplayName = user.getDisplayName();          intent.putExtra(EXTRA\_DISPLAY\_NAME, userDisplayName);          return intent;      }      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_home);          AppCompatTextView displayNameTextView = findViewById(R.id.display\_name\_text\_view);          displayNameTextView.setText("Welcome, " + getIntent().getStringExtra(EXTRA\_DISPLAY\_NAME));      }  } |

**Output:**

****

**Conclusion :-**

Hence, we have successfully demonstrated the use of Google Sign-in functionality while building a great Android app.

**Practical 12**

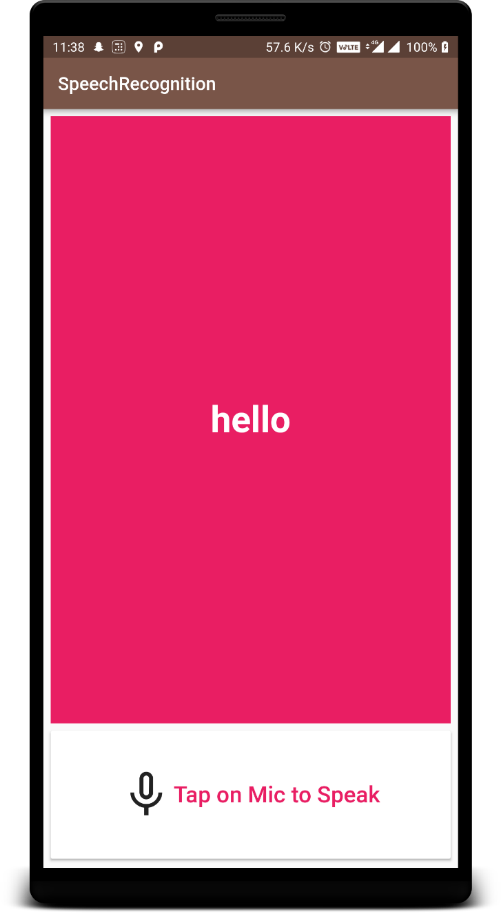
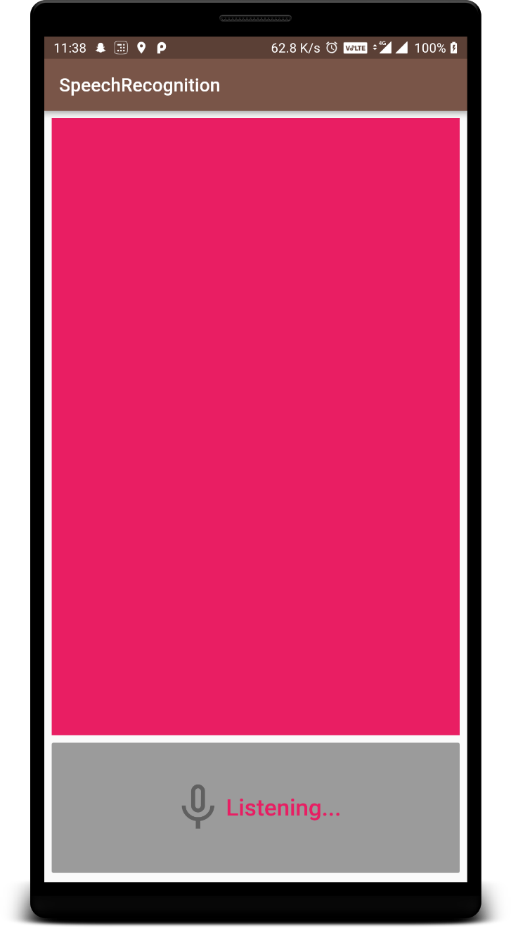
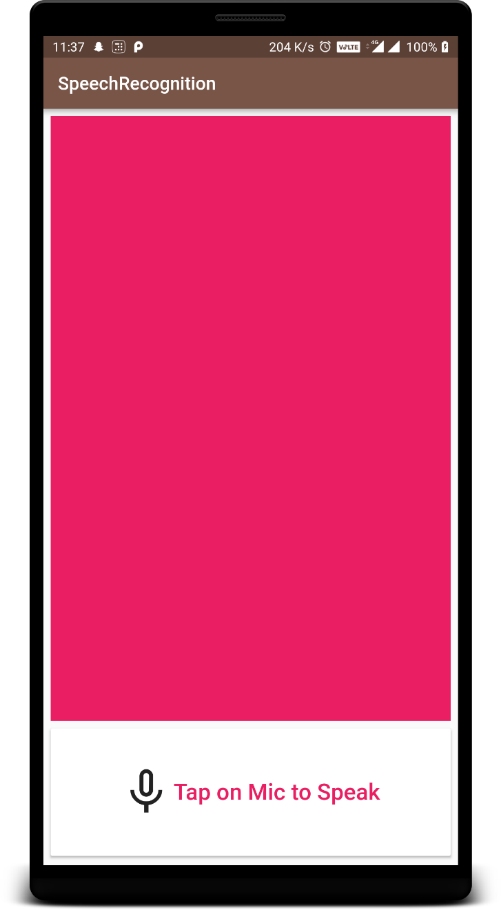
**AIM: Create an application to handle support voice interaction.**

**Source Code:**

**main.dart**

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:speech\_recognition/speech\_recognition.dart';  import 'package:flutter/services.dart';  void main() {  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp]);  runApp(new MyApp());  }  const languages = const [  const Language('English', 'en\_US'),  ];  class Language {  final String name;  final String code;  const Language(this.name, this.code);  }  class MyApp extends StatefulWidget {  @override  \_MyAppState createState() => new \_MyAppState();  }  class \_MyAppState extends State<MyApp> {  SpeechRecognition \_speech;  bool \_speechRecognitionAvailable = false;  bool \_isListening = false;  String transcription = '';  //String \_currentLocale = 'en\_US';  Language selectedLang = languages.first;  @override  initState() {  super.initState();  activateSpeechRecognizer();  }  // Platform messages are asynchronous, so we initialize in an async method.  void activateSpeechRecognizer() {  print('\_MyAppState.activateSpeechRecognizer... ');  \_speech = new SpeechRecognition();  \_speech.setAvailabilityHandler(onSpeechAvailability);  \_speech.setCurrentLocaleHandler(onCurrentLocale);  \_speech.setRecognitionStartedHandler(onRecognitionStarted);  \_speech.setRecognitionResultHandler(onRecognitionResult);  \_speech.setRecognitionCompleteHandler(onRecognitionComplete);  \_speech  .activate()  .then((res) => setState(() => \_speechRecognitionAvailable = res));  }  @override  Widget build(BuildContext context) {  return new MaterialApp(  debugShowCheckedModeBanner: false,  home: new Scaffold(  appBar: new AppBar(  title: new Text('SpeechRecognition'),  backgroundColor: Colors.brown,  ),  body: new Padding(  padding: new EdgeInsets.all(8.0),  child: new Center(  child: new Column(  mainAxisSize: MainAxisSize.min,  crossAxisAlignment: CrossAxisAlignment.stretch,  children: [  new Expanded(  child: Container(  width: 500,  height: 500,  color: Colors.pink,  child: Center(  child: new Text(transcription,  style: TextStyle(  fontSize: 40.0,  color: Colors.white,  fontWeight: FontWeight.bold,  ),  textAlign: TextAlign.center)),  )),  \_buildButton(  onPressed: \_speechRecognitionAvailable && !\_isListening  ? () => start()  : null,  label:  \_isListening ? 'Listening...' : 'Tap on Mic to Speak',  ),  ],  ),  )),  ),  );  }  Widget \_buildButton({String label, VoidCallback onPressed}) => new Padding(  padding: new EdgeInsets.only(top: 8, bottom: 2, left: 0, right: 0),  child: new RaisedButton(  color: Colors.white,  onPressed: onPressed,  padding: EdgeInsets.all(40.0),  child: Row(  mainAxisAlignment: MainAxisAlignment.center,  children: <Widget>[  Icon(  Icons.mic\_none,  size: 60.0,  ),  Text(  label,  style: const TextStyle(color: Colors.pink, fontSize: 25.0),  ),  ],  ),  ));  void start() => \_speech  .listen(locale: selectedLang.code)  .then((result) => print('\_MyAppState.start => result ${result}'));  void onSpeechAvailability(bool result) =>  setState(() => \_speechRecognitionAvailable = result);  void onCurrentLocale(String locale) {  print('\_MyAppState.onCurrentLocale... $locale');  setState(  () => selectedLang = languages.firstWhere((l) => l.code == locale));  }  void onRecognitionStarted() => setState(() => \_isListening = true);  void onRecognitionResult(String text) => setState(() => transcription = text);  void onRecognitionComplete() => setState(() => \_isListening = false);  } |

**Output:**

****

**Conclusion: By performing above experiment, one can able to make the voice recognition app in android as well as flutter.**

**Practical 13**

**AIM: Create Check-Out application to find the hottest places people are leaving. Use the Google Places API for Android to manage nearby locations and Firebase to store and synchronize data across devices in real time.**

**Source Code:**

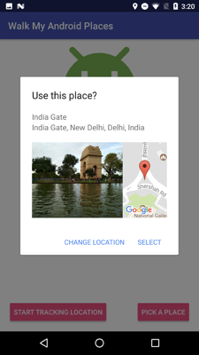
**activity\_main.dart**

|  |
| --- |
| <FrameLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent">      <com.google.android.material.button.MaterialButton          android:id="@+id/checkout\_button"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:onClick="checkOut"          android:text="@string/check\_out" />      <fragment xmlns:android="<http://schemas.android.com/apk/res/android>"          xmlns:tools="<http://schemas.android.com/tools>"          android:id="@+id/map"          android:name="com.google.android.gms.maps.SupportMapFragment"          android:layout\_width="match\_parent"          android:layout\_height="match\_parent"          tools:context=".MapsActivity" />  </FrameLayout>  ***AndroidManifest.xml :***  <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:tools="<http://schemas.android.com/tools>"      package="com.android.example.practical13">      <!--           The ACCESS\_COARSE/FINE\_LOCATION permissions are not required to use           Google Maps Android API v2, but you must specify either coarse or fine           location permissions for the 'MyLocation' functionality.      -->      <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />      <application          android:allowBackup="true"          android:icon="@mipmap/ic\_launcher"          android:label="@string/app\_name"          android:roundIcon="@mipmap/ic\_launcher\_round"          android:supportsRtl="true"          android:theme="@style/AppTheme"          tools:ignore="GoogleAppIndexingWarning">            <meta-data              android:name="com.google.android.geo.API\_KEY"              android:value="@string/google\_maps\_key" />          <activity              android:name=".MapsActivity"              android:label="@string/title\_activity\_maps">              <intent-filter>                  <action android:name="android.intent.action.MAIN" />                  <category android:name="android.intent.category.LAUNCHER" />              </intent-filter>          </activity>      </application>  </manifest> |

**MapsActivity.java**

|  |
| --- |
| package com.android.example.practical13;  import android.app.Activity;  import android.content.Intent;  import android.location.Location;  import android.os.Bundle;  import android.view.View;  import android.view.ViewTreeObserver;  import android.widget.Button;  import android.widget.Toast;  import com.firebase.client.DataSnapshot;  import com.firebase.client.Firebase;  import com.firebase.client.FirebaseError;  import com.firebase.client.ServerValue;  import com.google.android.gms.common.GoogleApiAvailability;  import com.google.android.gms.common.GooglePlayServicesNotAvailableException;  import com.google.android.gms.common.GooglePlayServicesRepairableException;  import com.google.android.gms.common.api.GoogleApiClient;  import com.google.android.gms.common.api.ResultCallback;  import com.google.android.gms.location.places.Place;  import com.google.android.gms.location.places.PlaceBuffer;  import com.google.android.gms.location.places.Places;  import com.google.android.gms.location.places.ui.PlacePicker;  import com.google.android.gms.maps.CameraUpdateFactory;  import com.google.android.gms.maps.GoogleMap;  import com.google.android.gms.maps.OnMapReadyCallback;  import com.google.android.gms.maps.SupportMapFragment;  import com.google.android.gms.maps.model.LatLng;  import com.google.android.gms.maps.model.LatLngBounds;  import com.google.android.gms.maps.model.MarkerOptions;  import com.google.firebase.database.ChildEventListener;  import java.util.HashMap;  import java.util.Map;  import androidx.fragment.app.FragmentActivity;    public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,          ChildEventListener {        private static final String FIREBASE\_ROOT\_NODE = "checkouts";      private static final int REQUEST\_PLACE\_PICKER = 1;      private GoogleMap mMap;      private Firebase mFirebase;      private GoogleApiClient mGoogleApiClient;      private LatLngBounds.Builder mBounds = new LatLngBounds.Builder();      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_maps);          // Set up Google Maps          SupportMapFragment mapFragment = (SupportMapFragment)                  getSupportFragmentManager().findFragmentById(R.id.map);          mapFragment.getMapAsync(this);          // Set up the API client for Places API          mGoogleApiClient = new GoogleApiClient.Builder(this)                  .addApi(Places.GEO\_DATA\_API)                  .build();          mGoogleApiClient.connect();          // Set up Firebase          Firebase.setAndroidContext(this);          mFirebase = new Firebase(FIREBASE\_URL);          mFirebase.child(FIREBASE\_ROOT\_NODE).addChildEventListener(this);      }      /\*\*       \* Prompt the user to check out of their location. Called when the "Check Out!" button       \* is clicked.       \*/      public void checkOut(View view) {          try {              PlacePicker.IntentBuilder intentBuilder = new PlacePicker.IntentBuilder();              Intent intent = intentBuilder.build(this);              startActivityForResult(intent, REQUEST\_PLACE\_PICKER);          } catch (GooglePlayServicesRepairableException e) {              GoogleApiAvailability.getInstance().getErrorDialog(this, e.getConnectionStatusCode(),                      REQUEST\_PLACE\_PICKER);          } catch (GooglePlayServicesNotAvailableException e) {              Toast.makeText(this, "Please install Google Play Services!", Toast.LENGTH\_LONG).show();          }      }      @Override      protected void onActivityResult(int requestCode, int resultCode, Intent data) {          if (requestCode == REQUEST\_PLACE\_PICKER) {              if (resultCode == Activity.RESULT\_OK) {                  Place place = PlacePicker.getPlace(data, this);                  Map<String, Object> checkoutData = new HashMap<>();                  checkoutData.put("time", ServerValue.TIMESTAMP);                  mFirebase.child(FIREBASE\_ROOT\_NODE).child(place.getId()).setValue(checkoutData);              } else if (resultCode == PlacePicker.RESULT\_ERROR) {                  Toast.makeText(this, "Places API failure! Check the API is enabled for your key",                          Toast.LENGTH\_LONG).show();              }          } else {              super.onActivityResult(requestCode, resultCode, data);          }      }      /\*\*       \* Map setup. This is called when the GoogleMap is available to manipulate.       \*       \* @param googleMap       \*/      @Override      public void onMapReady(GoogleMap googleMap) {          mMap = googleMap;          mMap.setMyLocationEnabled(true);          mMap.setOnMyLocationChangeListener(new GoogleMap.OnMyLocationChangeListener() {              @Override              public void onMyLocationChange(Location location) {                  LatLng ll = new LatLng(location.getLatitude(), location.getLongitude());                  addPointToViewPort(ll);                  // we only want to grab the location once, to allow the user to pan and zoom freely.                  mMap.setOnMyLocationChangeListener(null);              }          });          // Pad the map controls to make room for the button - note that the button may not have          // been laid out yet.          final Button button = (Button) findViewById(R.id.checkout\_button);          button.getViewTreeObserver().addOnGlobalLayoutListener(                  new ViewTreeObserver.OnGlobalLayoutListener() {                      @Override                      public void onGlobalLayout() {                          mMap.setPadding(0, button.getHeight(), 0, 0);                      }                  }          );      }      /\*\*       \* Act upon new check-outs when they appear.       \*/      @Override      public void onChildAdded(DataSnapshot dataSnapshot, String s) {          String placeId = dataSnapshot.getKey();          if (placeId != null) {              Places.GeoDataApi                      .getPlaceById(mGoogleApiClient, placeId)                      .setResultCallback(new ResultCallback<PlaceBuffer>() {                                             @Override                                             public void onResult(PlaceBuffer places) {                                                 LatLng location = places.get(0).getLatLng();                                                 addPointToViewPort(location);                                                 mMap.addMarker(new MarkerOptions().position(location));                                                 places.release();                                             }                                         }                      );          }      }      @Override      public void onChildChanged(DataSnapshot dataSnapshot, String s) {          // TODO      }      @Override      public void onChildRemoved(DataSnapshot dataSnapshot) {          // TODO      }      @Override      public void onChildMoved(DataSnapshot dataSnapshot, String s) {          // TODO      }      @Override      public void onCancelled(FirebaseError firebaseError) {          // TODO      }      private void addPointToViewPort(LatLng newPoint) {          mBounds.include(newPoint);          mMap.animateCamera(CameraUpdateFactory.newLatLngBounds(mBounds.build(),                  findViewById(R.id.checkout\_button).getHeight()));      }  } |

**Output:**

****

**Conclusion: By performing above experiment, one can develop an app in android using GOOGLE CLOUD MAPS API which shows the nearby places.**

**Practical 14**

**AIM: Create an application using In-app Billing API to make it easy for users to buy digital products and subscriptions.**

**Source Code:**

**activity\_main.xml**

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <android.support.constraint.ConstraintLayout xmlns:android="<http://schemas.android.com/apk/res/android>"      xmlns:app="<http://schemas.android.com/apk/res-auto>"      xmlns:tools="<http://schemas.android.com/tools>"      android:layout\_width="match\_parent"      android:layout\_height="match\_parent"      tools:context=".MainActivity">      <Button          android:id="@+id/button"          android:layout\_width="wrap\_content"          android:layout\_height="wrap\_content"          android:text="@string/purchase"/>  </android.support.constraint.ConstraintLayout> |

**MapsActivity.java**

|  |
| --- |
| package com.example.practical14;  import android.content.Intent;  import android.support.annotation.NonNull;  import android.support.annotation.Nullable;  import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.view.Menu;  import android.view.MenuItem;  import android.view.View;  import android.widget.Button;  import android.widget.Toast;  import com.anjlab.android.iab.v3.BillingProcessor;  import com.anjlab.android.iab.v3.TransactionDetails;  public class MainActivity extends AppCompatActivity implements BillingProcessor.IBillingHandler {      BillingProcessor bp;      @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.*activity\_main*);          bp = new BillingProcessor(this, "YOUR LICENSE KEY FROM GOOGLE PLAY CONSOLE HERE", this);          bp.initialize();          Button btnPurchase = findViewById([R.id](http://r.id/).*button*);          btnPurchase.setOnClickListener(new View.OnClickListener() {              @Override              public void onClick(View v) {                  bp.purchase(MainActivity.this,"android.test.purchased");              }          });      }      @Override      public boolean onOptionsItemSelected(MenuItem item) {          return super.onOptionsItemSelected(item);      }      @Override      public boolean onCreateOptionsMenu(Menu menu) {          return super.onCreateOptionsMenu(menu);      }      @Override      public void onProductPurchased(@NonNull String productId, @Nullable TransactionDetails details) {          Toast.*makeText*(this,"You've purchased something",Toast.*LENGTH\_LONG*).show();      }      @Override      public void onPurchaseHistoryRestored() {      }      @Override      public void onBillingError(int errorCode, @Nullable Throwable error) {      }      @Override      public void onBillingInitialized() {      }      @Override      protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {          if(!bp.handleActivityResult(requestCode,resultCode,data)){              super.onActivityResult(requestCode,resultCode,data);          }      }      @Override      protected void onDestroy() {          if (bp!=null){              bp.release();          }          super.onDestroy();      }  } |

**Output:**

****

**Conclusion: By performing above experiment, one can develop an app in android using GOOGLE CLOUD Billing API which is useful for creating bills and payment gateway.**

**Practical 15**

**Aim: Introduction to I-phone & installation of x-code on MAC.**

**Introduction to I-phone:**

iPhones are used to make phone calls and send text messages but they can also be used for accessing the internet to check your emails, surfing the web and much more. An iPhone is a type of smartphone. iPhones are made by Apple. Android phones are made by Samsung, HTC, and other companies. Microsoft manufactures Windows phones.

The difference between each phone is that they have a different operating system developed by the different manufacturers. You can use your fingers to control the touch screen of your phone or you can also use a pencil like device called a stylus which makes it easier to use. You will find a stylus for sale in electronic shops.

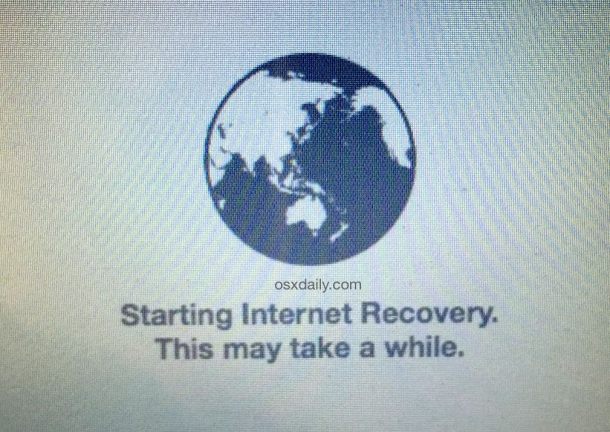
A word you will see a lot of: App: Is short for the word application. It is a shortcut that allows you to go directly to the programme or website you wish to access.

**Installation of x-code on MAC:**

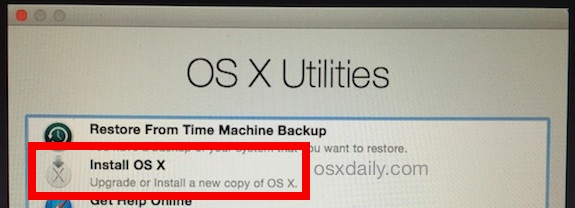
Internet Recovery requires internet access, that may be slightly obvious but it’s worth mentioning because if the Mac is unable to connect to a network then it will not be able to download the operating system. Whenever possible, you should [back up the Mac with Time Machine](http://osxdaily.com/2010/07/21/how-to-do-manual-backups-with-time-machine/) before attempting this.

You can start the internet recovery reinstall process from either a shutdown Mac, or by rebooting the Mac. This process will be the same on any new Mac, be it an iMac, MacBook Pro, MacBook Air, etc:

1. Immediately after hearing the Mac boot chime, **hold down Command+Option+R** – if you see the Apple logo you waited too long and need to reboot and try again\*
2. OPTIONAL: You may or may not see an option to join a wi-fi network, this depends on whether the Mac can access any saved networks from OS X or not
3. When you see a spinning globe icon, Internet Recovery mode has been entered with a message saying it can take a while, a progress bar appears as the recovery functions are downloaded



1. When finished downloading, you’ll see the familiar “OS X Utilities” screen, choose “Reinstall OS X” to begin the re-installation process of the Mac operating system



1. Select the destination and complete the re-installation (or installation) of OS X as usual

You’ll notice the version of OS X that can be reinstalled this way is shown in the icon or listed under the “Reinstall OS X” option, and that version will match whatever version of OS X came preinstalled on the Mac. For example, if the Mac shipped with OS X Mavericks but is now running OS X Yosemite, then OS X Mavericks would be the version that reinstalls through the Internet Recovery reinstall process.

For Macs that do not currently have an operating system found or installed, the option will show as “Install OS X” rather than “Reinstall OS X”.

Installing and re-installing OS X through Internet Recovery is pretty easy, but do be aware that because everything is coming from Apple servers, it can take quite a while as the system restore features are downloaded locally, and then the version of OS X to install are also downloaded locally as well.

When OS X has finished installing on the Mac, it will boot into a fresh install of OS X system software.

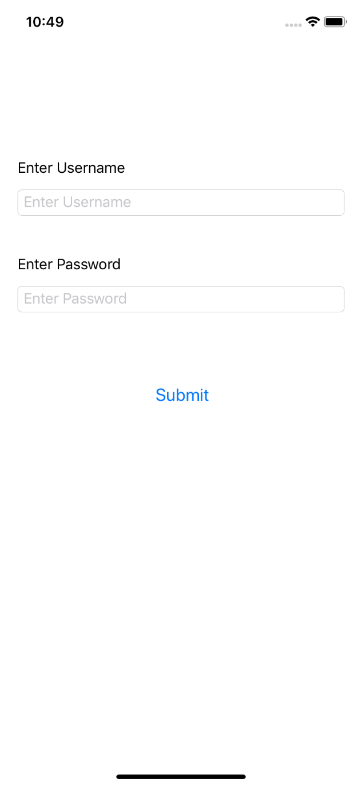
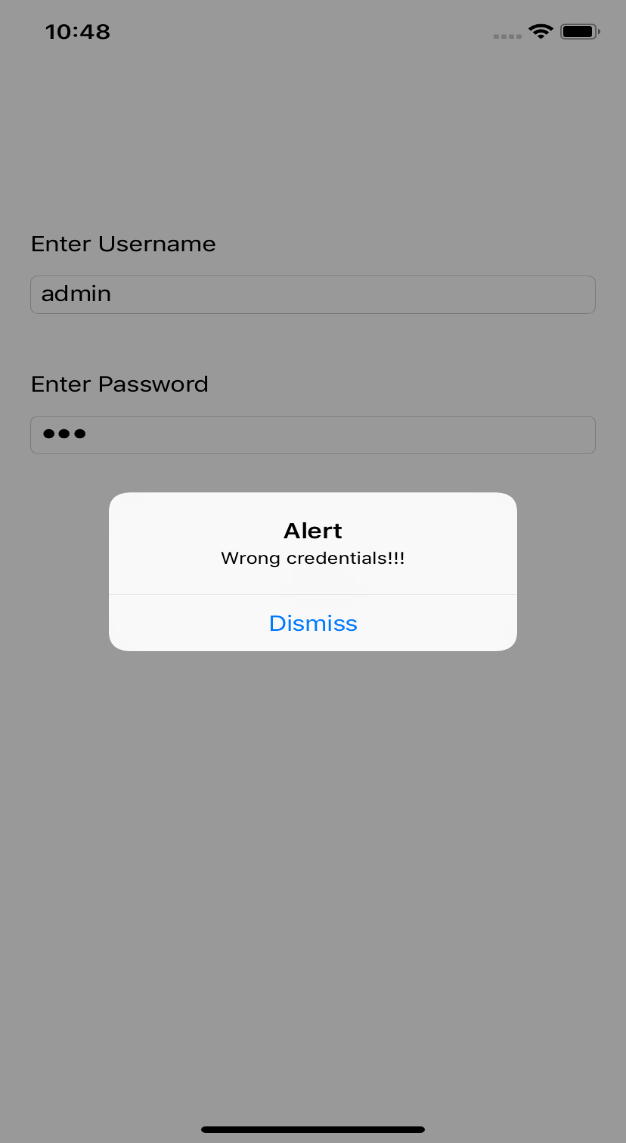
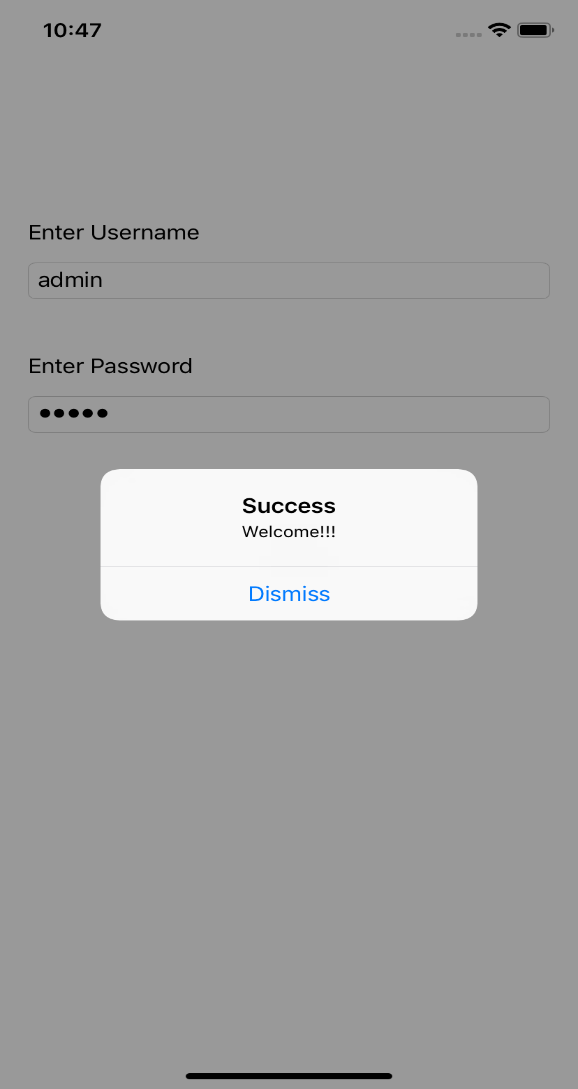
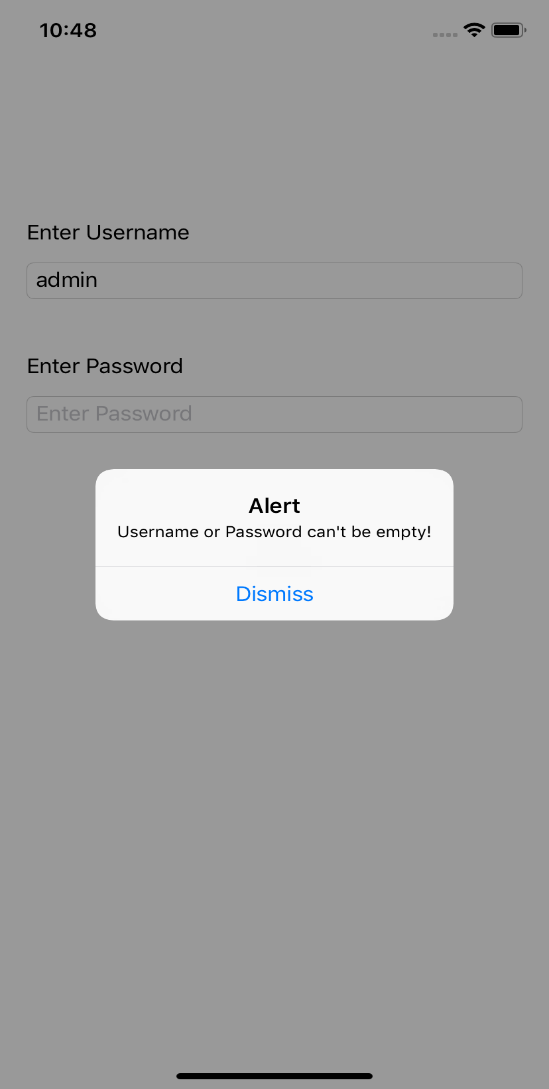
**Practical 16**

**Aim: Develop Login Application using XCode.**

**ViewController.Swift**

|  |
| --- |
| import UIKit  class ViewController: UIViewController {       @IBOutlet weak var username: UITextField!     @IBOutlet weak var password: UITextField!     override func viewDidLoad() {         super.viewDidLoad()  *// Do any additional setup after loading the view.*     }     @IBAction func loginPressed(\_ sender: Any) {         if (username.text == "admin" && password.text == "admin") {             let alertController = UIAlertController(title: "Success", message:                 "Welcome!!!", preferredStyle: .alert)             alertController.addAction(UIAlertAction(title: "Dismiss", style: .default))             self.present(alertController, animated: true, completion: nil)         }         else if(username.text == "" || password.text == ""){             let alertController = UIAlertController(title: "Alert", message:                 "Username or Password can't be empty!", preferredStyle: .alert)             alertController.addAction(UIAlertAction(title: "Dismiss", style: .default))             self.present(alertController, animated: true, completion: nil)         }         else{             let alertController = UIAlertController(title: "Alert", message:                 "Wrong credentials!!!", preferredStyle: .alert)             alertController.addAction(UIAlertAction(title: "Dismiss", style: .default))             self.present(alertController, animated: true, completion: nil)         }     }  } |

**Output:**

**   **

**Conclusion:**

Hence, we have successfully learnt to develop Login App on iOS Platform.