Ex No: 1

1. Develop an application that uses GUI components, Font, and Colors.

Aim:

To develop an application in android studio using GUI components, fonts and colors.

Software and Hardware Requirements:

Software:
Android Studio
Hardware:
Preferably 8GB+ RAM Laptop/Desktop

Procedure:

- Fonts:
  The google_fonts package will automatically use matching font files in your pubspec.yaml's assets.Open Sans,Montserrat are some of the fonts available.Style property can be used to add TextStyle like fontSize, color.

- Colour:
  A color used on interactive elements of the theme. This color is generally used on text and icons in buttons and tappable elements.Color property can be used to specify the color using the Colors class.

- GUI COMPONENTS:
  Container():Helps to create a rectangular visual element.Decoration can be used to give shape, backgroundColor etc to a container.

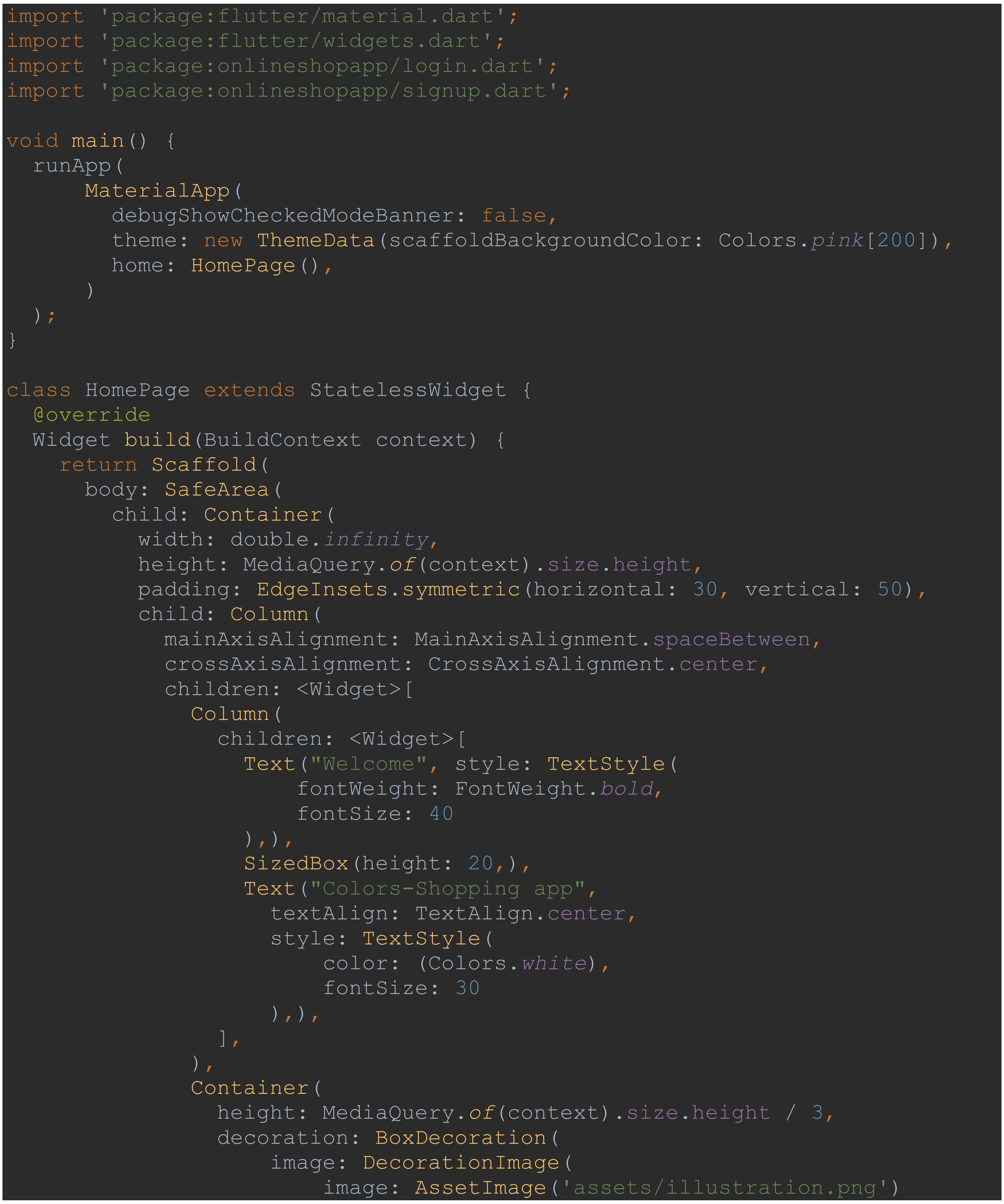
- Scaffold()
  Creates a visual scaffold for Material Design widgets appBar() id used to specify the title and background of the top bar.body() is used to contain the primary content of the scaffold.

- MaterialApp()
  o contains widgets that are used for the material design of an application.
  o theme property is used to set the theme of the application to dark or light.
  o Home property defines the starting point of the application. It usually contains Scaffold.
  - Text():
  o specify the string to be displayed, withing quotes inside Text().
  o Style property can be used to add TextStyle like fontSize, color.

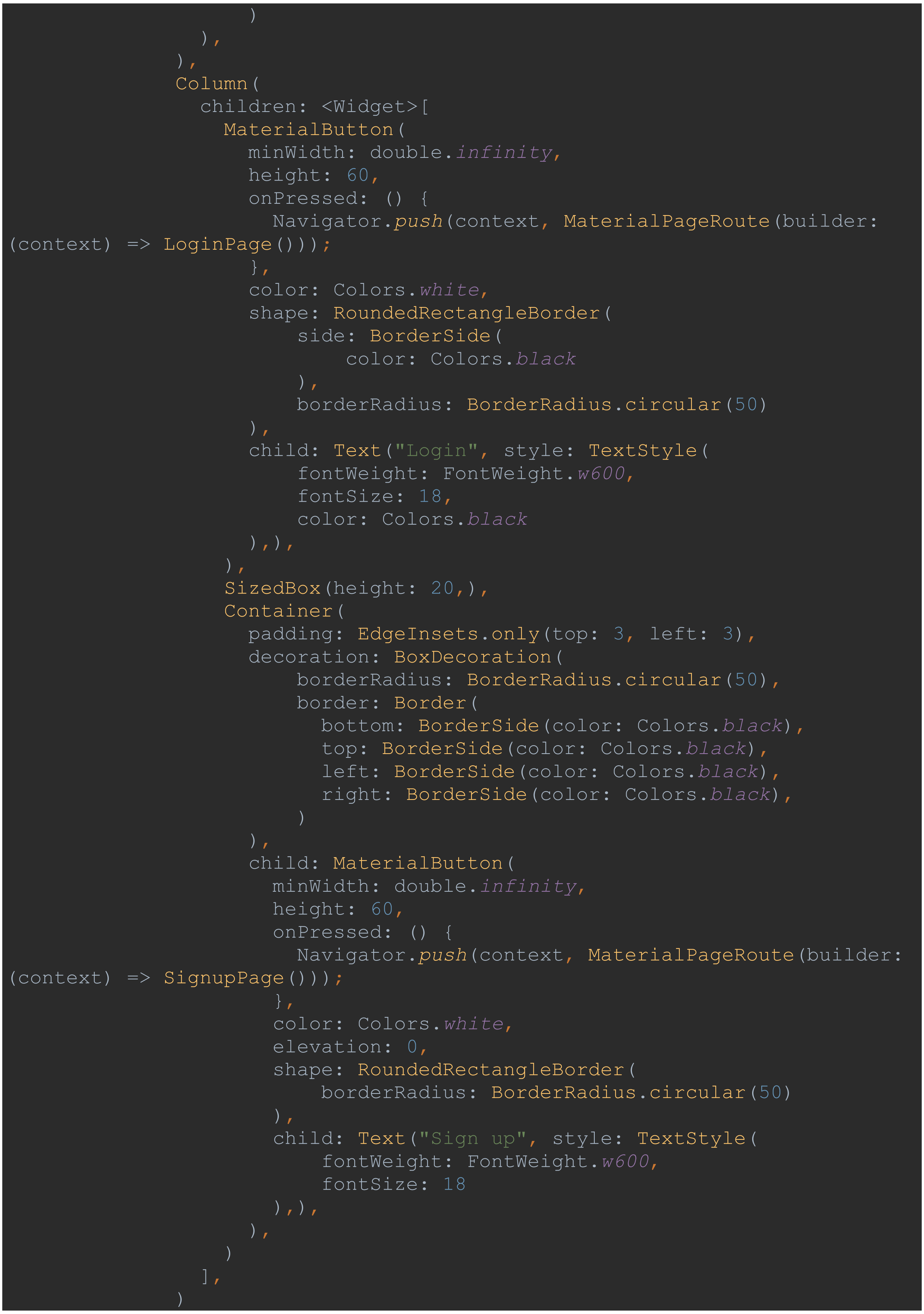
o textAlign property can be used for alignment of specified text

Code:

Main.dart:



```
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:onlineshopapp/login.dart';
import 'package:onlineshopapp/signup.dart';

void main() {
  runApp(
      MaterialApp(
        debugShowCheckedModeBanner: false,
        theme: new ThemeData(scaffoldBackgroundColor: Colors.pink[200]),
        home: HomePage(),
      )
  );
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Container(
          width: double.infinity,
          height: MediaQuery.of(context).size.height,
          padding: EdgeInsets.symmetric(horizontal: 30, vertical: 50),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: <Widget>[
              Column(
                children: <Widget>[
                  Text("Welcome", style: TextStyle(
                      fontWeight: FontWeight.bold,
                      fontSize: 40
                  ),),
                  SizedBox(height: 20,),
                  Text("Colors-Shopping app",
                    textAlign: TextAlign.center,
                    style: TextStyle(
                        color: (Colors.white),
                        fontSize: 30
                    ),),
                ],
              ),
              Container(
                height: MediaQuery.of(context).size.height / 3,
                decoration: BoxDecoration(
                    image: DecorationImage(
                        image: AssetImage('assets/illustration.png')
```

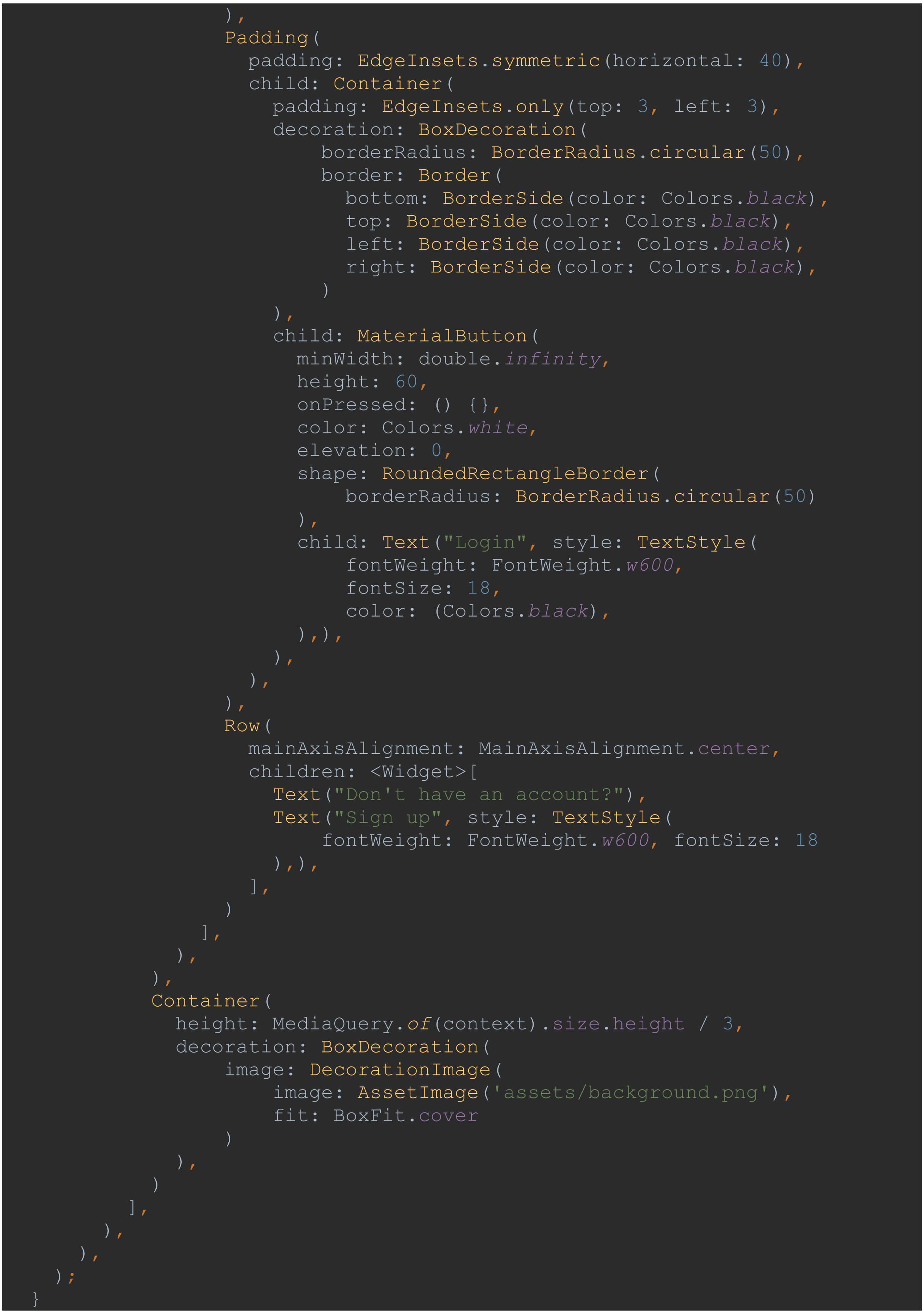


```
                    )
                ),
              ),
              Column(
                children: <Widget>[
                  MaterialButton(
                    minWidth: double.infinity,
                    height: 60,
                    onPressed: () {
                      Navigator.push(context, MaterialPageRoute(builder:
(context) => LoginPage()));
                    },
                    color: Colors.white,
                    shape: RoundedRectangleBorder(
                        side: BorderSide(
                            color: Colors.black
                        ),
                        borderRadius: BorderRadius.circular(50)
                    ),
                    child: Text("Login", style: TextStyle(
                        fontWeight: FontWeight.w600,
                        fontSize: 18,
                        color: Colors.black
                    ),),
                  ),
                  SizedBox(height: 20,),
                  Container(
                    padding: EdgeInsets.only(top: 3, left: 3),
                    decoration: BoxDecoration(
                        borderRadius: BorderRadius.circular(50),
                        border: Border(
                          bottom: BorderSide(color: Colors.black),
                          top: BorderSide(color: Colors.black),
                          left: BorderSide(color: Colors.black),
                          right: BorderSide(color: Colors.black),
                        )
                    ),
                    child: MaterialButton(
                      minWidth: double.infinity,
                      height: 60,
                      onPressed: () {
                        Navigator.push(context, MaterialPageRoute(builder:
(context) => SignupPage()));
                      },
                      color: Colors.white,
                      elevation: 0,
                      shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(50)
                      ),
                      child: Text("Sign up", style: TextStyle(
                          fontWeight: FontWeight.w600,
                          fontSize: 18
                      ),),
                    ),
                  )
                ],
              )
```

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

Login.dart:

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

class LoginPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      backgroundColor: Colors.pink[200],
      appBar: AppBar(
        elevation: 0,
        brightness: Brightness.light,
        backgroundColor: Colors.white,
        leading: IconButton(
          onPressed: () {
            Navigator.pop(context);
          },
          icon: Icon(Icons.arrow_back_ios, size: 20, color: Colors.black,),
        ),
      ),
      body: Container(
        height: MediaQuery.of(context).size.height,
        width: double.infinity,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            Expanded(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  Column(
                    children: <Widget>[
                      Text("Login", style: TextStyle(
                          fontSize: 30,
                          fontWeight: FontWeight.bold
                      ),),
                    ],
                  ),
                  Padding(
                    padding: EdgeInsets.symmetric(horizontal: 40),
                    child: Column(
                      children: <Widget>[
                        makeInput(label: "Email"),
                        makeInput(label: "Password", obscureText: true),
                      ],
                    ),
```



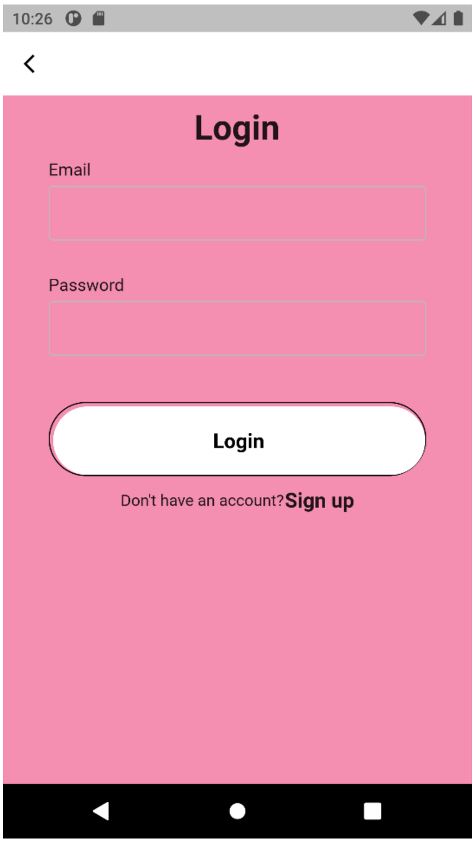
```
                ),
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 40),
                  child: Container(
                    padding: EdgeInsets.only(top: 3, left: 3),
                    decoration: BoxDecoration(
                        borderRadius: BorderRadius.circular(50),
                        border: Border(
                          bottom: BorderSide(color: Colors.black),
                          top: BorderSide(color: Colors.black),
                          left: BorderSide(color: Colors.black),
                          right: BorderSide(color: Colors.black),
                        )
                    ),
                    child: MaterialButton(
                      minWidth: double.infinity,
                      height: 60,
                      onPressed: () {},
                      color: Colors.white,
                      elevation: 0,
                      shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(50)
                      ),
                      child: Text("Login", style: TextStyle(
                          fontWeight: FontWeight.w600,
                          fontSize: 18,
                          color: (Colors.black),
                      ),),
                    ),
                  ),
                ),
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    Text("Don't have an account?"),
                    Text("Sign up", style: TextStyle(
                        fontWeight: FontWeight.w600, fontSize: 18
                    ),),
                  ],
                )
              ],
            ),
          ),
          Container(
            height: MediaQuery.of(context).size.height / 3,
            decoration: BoxDecoration(
                image: DecorationImage(
                    image: AssetImage('assets/background.png'),
                    fit: BoxFit.cover
                )
            ),
          )
        ],
      ),
    ),
  );
}
```

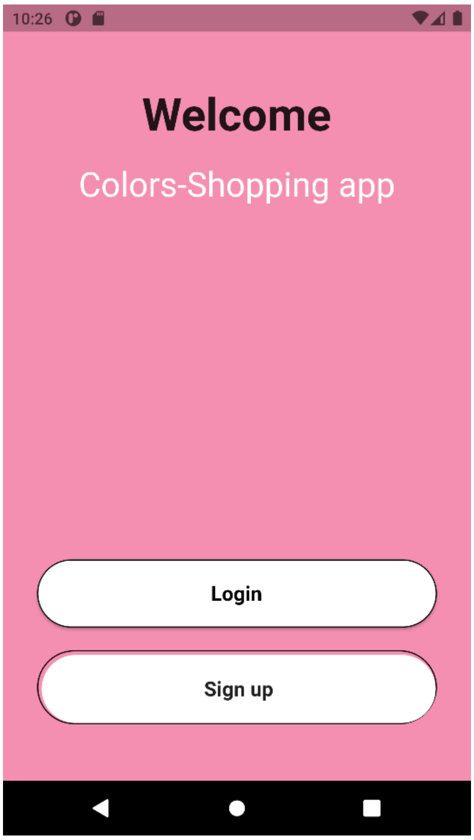
```
  Widget makeInput({label, obscureText = false}) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        Text(label, style: TextStyle(
            fontSize: 15,
            fontWeight: FontWeight.w400,
            color: Colors.black87
        ),),
        SizedBox(height: 5,),
        TextField(
          obscureText: obscureText,
          decoration: InputDecoration(
            contentPadding: EdgeInsets.symmetric(vertical: 0, horizontal:
10),
            enabledBorder: OutlineInputBorder(
                borderSide: BorderSide(color: (Colors.grey[400])!)
            ),
            border: OutlineInputBorder(
                borderSide: BorderSide(color: (Colors.grey[400])!)
            ),
          ),
        ),
        SizedBox(height: 30,),
      ],
    );
  }
}
```

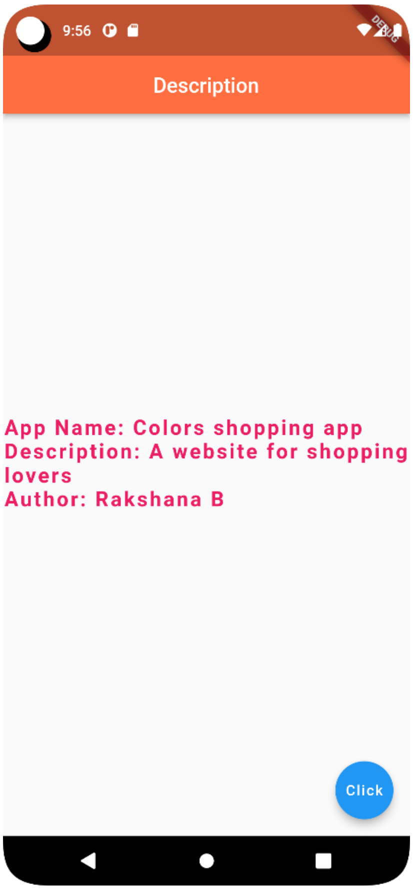
Main.dart:

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

void main() => runApp(MaterialApp(
  home: Scaffold(

    appBar: AppBar(
      title: Text('Description'),
      centerTitle: true,
      backgroundColor: Colors.deepOrangeAccent,
    ),
    body: Center(
      child: Text(
        'App Name: Colors shopping app\n'
            'Description: A website for shopping lovers\n'
            'Author: Rakshana B\n',
        style: TextStyle(
          fontSize: 20.0,
          fontWeight: FontWeight.bold,
          letterSpacing: 2.0,
          color: Colors.pink,
          fontFamily: 'Patrickhand',
        ),),
    ),
    floatingActionButton: FloatingActionButton(onPressed: (){},
        child: Text('Click')
    ),
  ),
));
```

Sample I/O:







App Name: Colors shopping app
Description: A website for shopping lovers
Author: Rakshana B

Result:

Thus, application was successfully implemented in Android Studio using GUI Components, fonts and colours.

Ex No: 2

2. Develop an application that uses Layout Managers and event listeners.

Aim:

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

Software and Hardware Requirements:

Software:
Android Studio

Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

Layout managers:

o Column() class is used to display its children in a vertical way.
o Children property is used to specify its descendants.
o ListTile is a fixed-height row that typically contains some text as well as leading or trailing icon.
o The icons (or other widgets) for the tile are defined with the leading and trailing parameters.

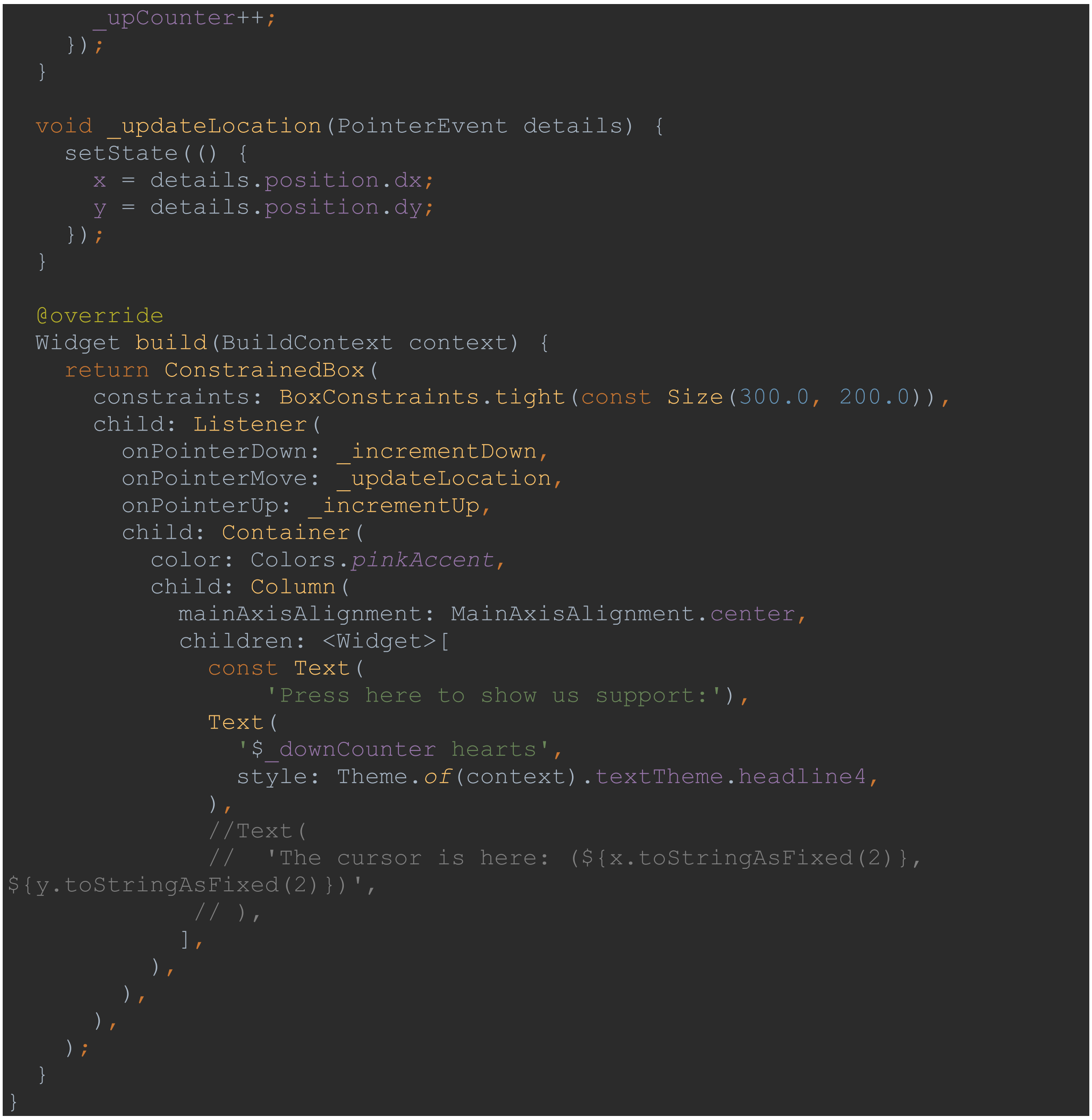
Event listeners:

o onPressed() property is used to assign a callback function to the button or icon.
o The application executes this function whenever the user presses taps the chip.
o If onPressed() is null, then it denotes disabled.

Code:

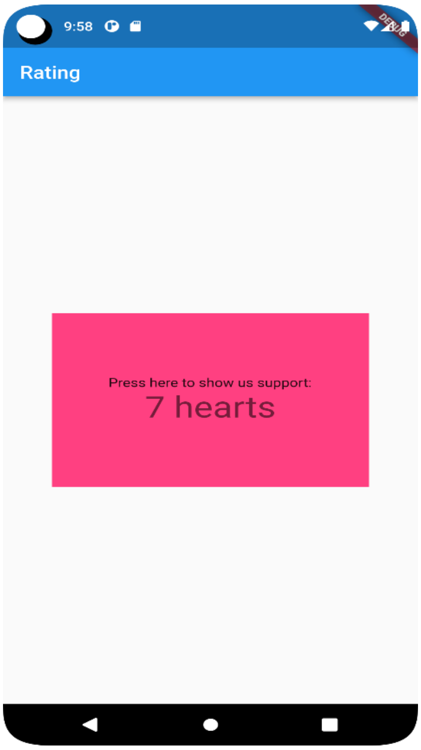
Main.dart:

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

void main() => runApp(const MyApp());

/// This is the main application widget.
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  static const String _title = 'Rating';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      home: Scaffold(
        appBar: AppBar(title: const Text(_title)),
        body: const Center(
          child: MyStatefulWidget(),
        ),
      ),
    );
  }
}

/// This is the stateful widget that the main application instantiates.
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

/// This is the private State class that goes with MyStatefulWidget.
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _downCounter = 0;
  int _upCounter = 0;
  double x = 0.0;
  double y = 0.0;

  void _incrementDown(PointerEvent details) {
    _updateLocation(details);
    setState(() {
      _downCounter++;
    });
  }

  void _incrementUp(PointerEvent details) {
    _updateLocation(details);
    setState(() {
```



```
      _upCounter++;
    });
  }

  void _updateLocation(PointerEvent details) {
    setState(() {
      x = details.position.dx;
      y = details.position.dy;
    });
  }

  @override
  Widget build(BuildContext context) {
    return ConstrainedBox(
      constraints: BoxConstraints.tight(const Size(300.0, 200.0)),
      child: Listener(
        onPointerDown: _incrementDown,
        onPointerMove: _updateLocation,
        onPointerUp: _incrementUp,
        child: Container(
          color: Colors.pinkAccent,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              const Text(
                  'Press here to show us support:'),
              Text(
                '$_downCounter hearts',
                style: Theme.of(context).textTheme.headline4,
              ),
              //Text(
              //  'The cursor is here: (${x.toStringAsFixed(2)},
${y.toStringAsFixed(2)})',
             // ),
            ],
          ),
        ),
      ),
    );
  }
}
```

Sample I/O:



Press here to show us support:

7 hearts

Result:

Thus a Simple Android Application that uses Layout Managers and Event Listeners is developed and executed successfully.

Ex No: 3

3) Develop a native calculator application.

Aim:

To implement a simple calculator in android studio.

Software and Hardware Requirements:

Software:

Android Studio

Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

Initialize num1, num2 and res (result) as 0

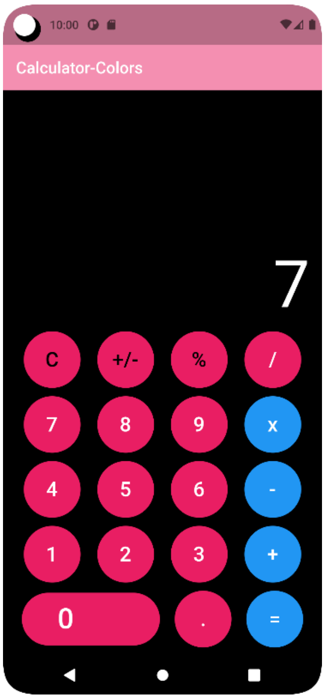
- Declare a function for each of the basic arithmetic operations (+ , - , * , /) which takes two operands as parameters and returns the result.
- Use the TextField, to get num1 and num2 as input.
- TextEditingController is used to retrieve the values of the TextField(s).
- Use another non-editable TextField to display the result.
- Use MaterialButton to perform the labelled arithmetic operation.

Code:

Main.dart:

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.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
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter
run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the applica-
tion
        // is not restarted.
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // 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
      // changed in this State, which causes it to rerun the build method
below
      // 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),
      ),
      body: Center(
        // Center is a layout widget. It takes a single child and positions it
        // in the middle of the parent.
        child: Column(
          // Column is also a layout widget. It takes a list of children and
          // arranges them vertically. By default, it sizes itself to fit its
          // 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
          // Studio, or the "Toggle Debug Paint" command in Visual Studio Code)
          // to see the wireframe for each widget.
          //
          // Column has various properties to control how it sizes itself and
          // how it positions its children. Here we use mainAxisAlignment to
          // center the children vertically; the main axis here is the vertical
          // 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.
    );
  }
}
```

Sample I/O:



Result:

Thus a Simple Android Application for Native Calculator is developed and executed successfully.

Expt. No: 4

4) Write an application that draws basic graphical primitives on the screen.

Aim:

To implement basic graphical primitives in android studio.

Software and Hardware Requirements:

Software:
Android Studio
Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

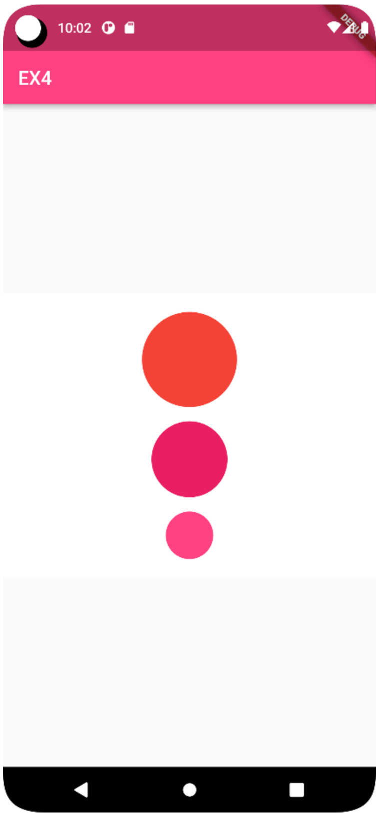
- Initialize numbutton and provide an suitable colour
- Initialize firstnumber and second number along with result text and operation
- Declare a function for each of the basic arithmetic operations (+ , - , * , /) which takes two operands as parameters and returns the result.

Code:

Main.dart:

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


void main() {
  runApp(new MaterialApp(
    home: Traffic(),
  ));
}
class Traffic extends StatelessWidget {
  const Traffic({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('EX4'),
          backgroundColor: Colors.pinkAccent,
        ),

        body:Center(
          child: Container(
            width: 500,
            height: 300,
            color: Colors.white,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                Container(
                  width: 100,
                  height: 100,
                  decoration: BoxDecoration(
                      shape: BoxShape.circle, color: Colors.red
                  ),
                ),
                SizedBox(
                    height:15
                ),
                Container(
                  width: 80,
                  height: 80,
                  decoration: BoxDecoration(
                      shape: BoxShape.circle, color: Colors.pink
                  ),
                ),
                SizedBox(
                    height:15
                ),
                Container(
                  width: 50,
                  height: 50,
                  decoration: BoxDecoration(
                      shape: BoxShape.circle,color: Colors.pinkAccent
                  ),
                ),
              ],
            ),
          ),
        )
    );
  }
}
```

Sample I/O:



Result:

Thus a Simple Android Application that draws basic Graphical Primitives on the screen is developed and executed successfully.

Expt. No: 5

5)Develop an application that makes use of database.

Aim:

To develop a simple Android Application using a database.

Software and Hardware Requirements:

Software:

Android Studio

Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

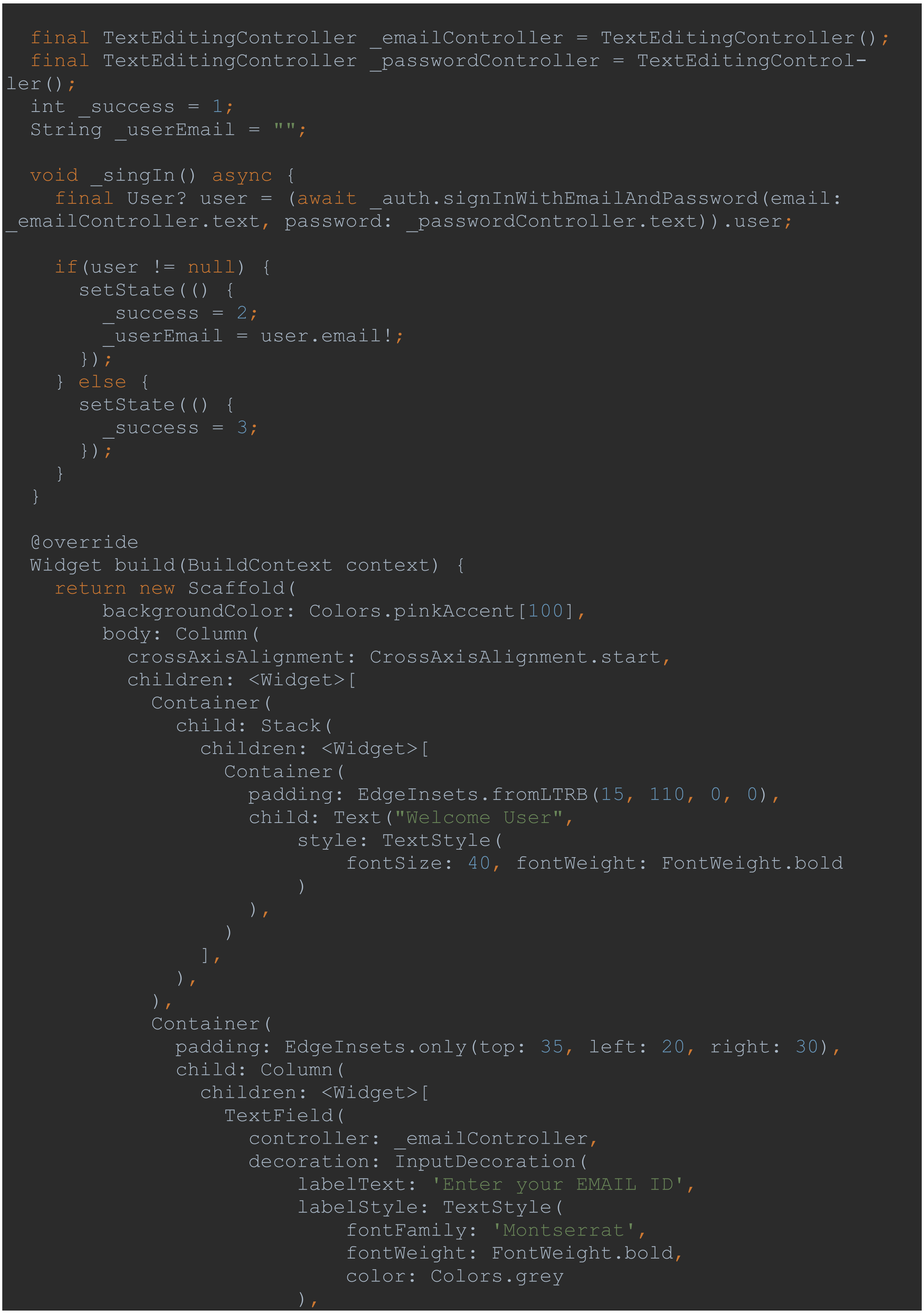
- In pubspec.yaml add these :
- firebase_core
- firebase_auth
- And then click pub get.
  - Use ‘firebase login’ command to login to google account
  - Use ‘flutterfire configure’ to add a firebase project to the application.
  - Run main.dart file
  - FirebaseAuth.instance.currentUser is used to get the current user object
  - Under authenthication in firbase the users database will be visible.

Code:

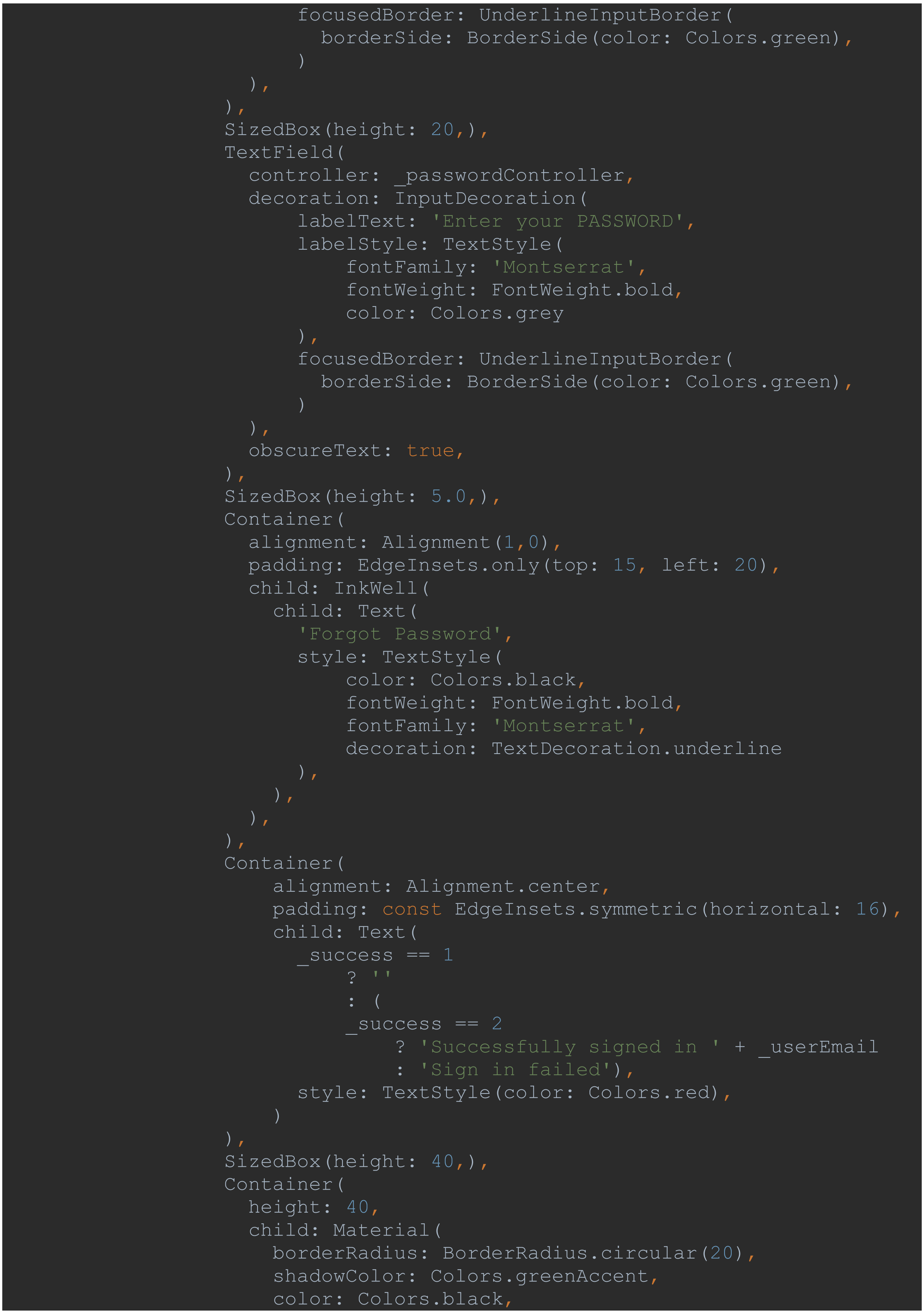
Main.dart:

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

import 'signup.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}
```

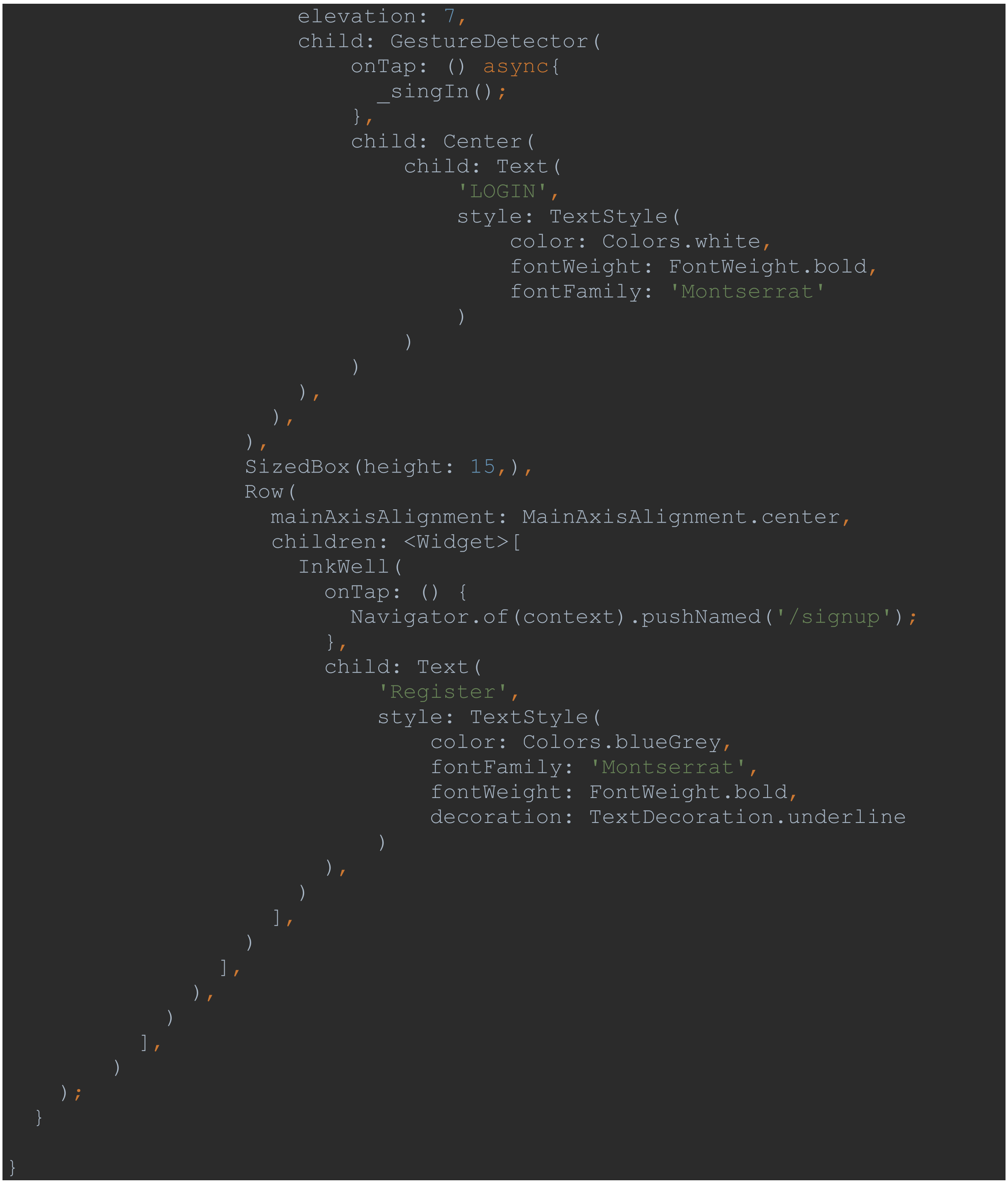
```
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // 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
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter
run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the applica-
tion
        // is not restarted.
        backgroundColor: Colors.pinkAccent[200],
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
      debugShowCheckedModeBanner: false,
      routes: <String, WidgetBuilder>{
        '/signup': (BuildContext context) => new SignupPage()
      },
    );
  }
}

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> {
```



```
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  int _success = 1;
  String _userEmail = "";

  void _singIn() async {
    final User? user = (await _auth.signInWithEmailAndPassword(email:
_emailController.text, password: _passwordController.text)).user;

    if(user != null) {
      setState(() {
        _success = 2;
        _userEmail = user.email!;
      });
    } else {
      setState(() {
        _success = 3;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
        backgroundColor: Colors.pinkAccent[100],
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Container(
              child: Stack(
                children: <Widget>[
                  Container(
                    padding: EdgeInsets.fromLTRB(15, 110, 0, 0),
                    child: Text("Welcome User",
                        style: TextStyle(
                            fontSize: 40, fontWeight: FontWeight.bold
                        )
                    ),
                  )
                ],
              ),
            ),
            Container(
              padding: EdgeInsets.only(top: 35, left: 20, right: 30),
              child: Column(
                children: <Widget>[
                  TextField(
                    controller: _emailController,
                    decoration: InputDecoration(
                        labelText: 'Enter your EMAIL ID',
                        labelStyle: TextStyle(
                            fontFamily: 'Montserrat',
                            fontWeight: FontWeight.bold,
                            color: Colors.grey
                        ),
```



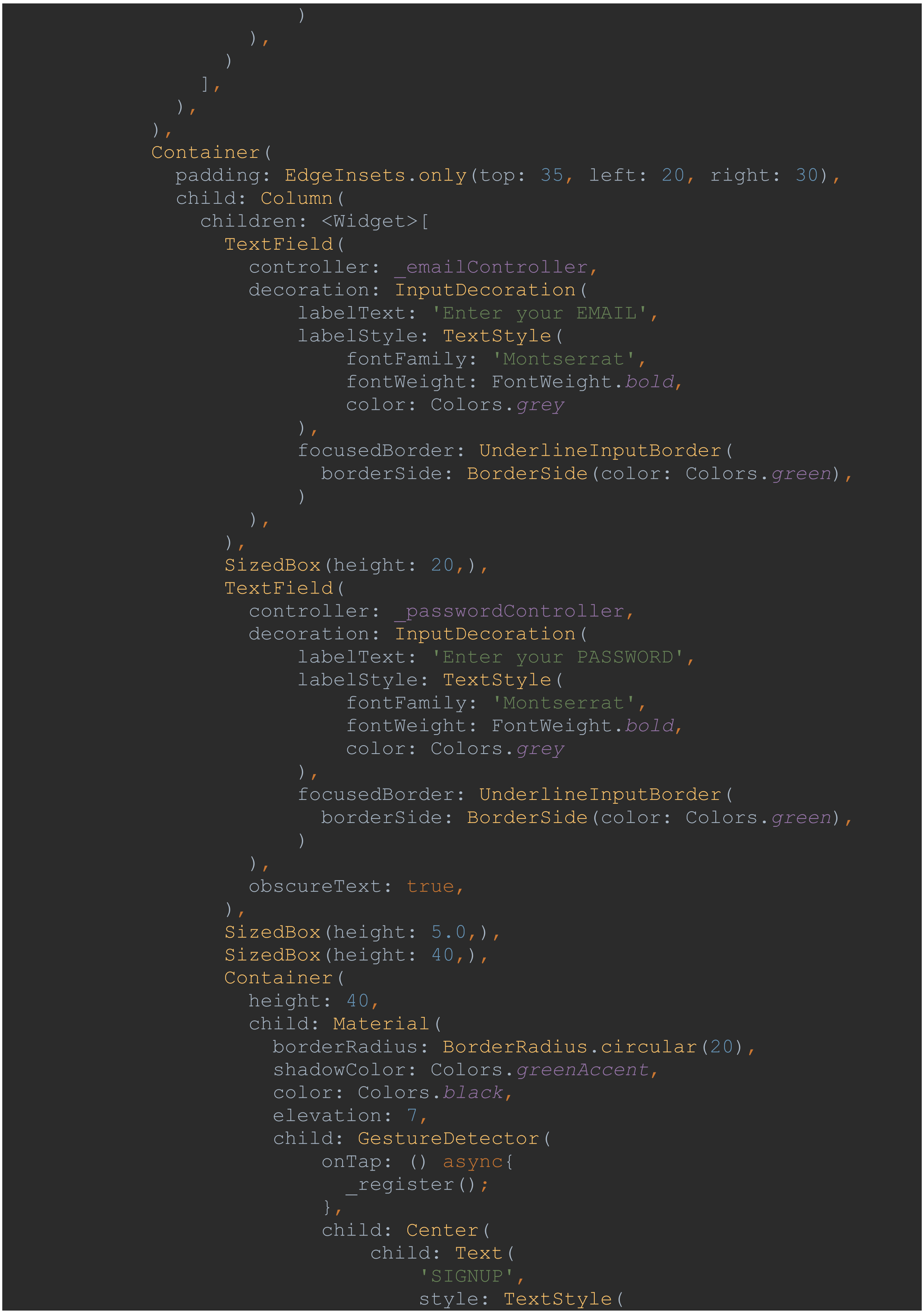
```
      focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.green),
      )
  ),
),
SizedBox(height: 20,),
TextField(
  controller: _passwordController,
  decoration: InputDecoration(
      labelText: 'Enter your PASSWORD',
      labelStyle: TextStyle(
          fontFamily: 'Montserrat',
          fontWeight: FontWeight.bold,
          color: Colors.grey
      ),
      focusedBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.green),
      )
  ),
  obscureText: true,
),
SizedBox(height: 5.0,),
Container(
  alignment: Alignment(1,0),
  padding: EdgeInsets.only(top: 15, left: 20),
  child: InkWell(
    child: Text(
      'Forgot Password',
      style: TextStyle(
          color: Colors.black,
          fontWeight: FontWeight.bold,
          fontFamily: 'Montserrat',
          decoration: TextDecoration.underline
      ),
    ),
  ),
),
Container(
    alignment: Alignment.center,
    padding: const EdgeInsets.symmetric(horizontal: 16),
    child: Text(
      _success == 1
          ? ''
          : (
          _success == 2
              ? 'Successfully signed in ' + _userEmail
              : 'Sign in failed'),
      style: TextStyle(color: Colors.red),
    )
),
SizedBox(height: 40,),
Container(
  height: 40,
  child: Material(
    borderRadius: BorderRadius.circular(20),
    shadowColor: Colors.greenAccent,
    color: Colors.black,
```



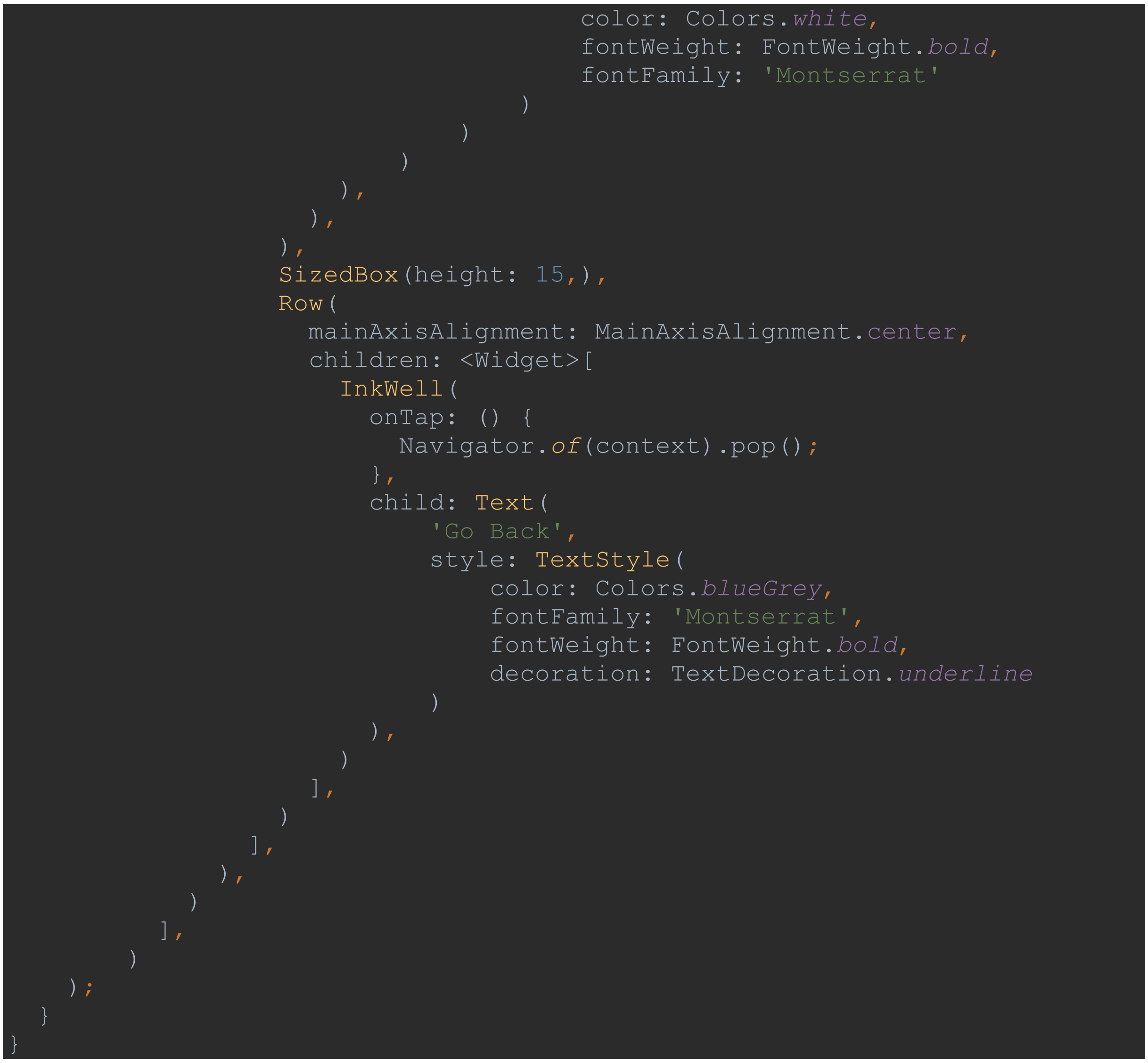
```
                      elevation: 7,
                      child: GestureDetector(
                          onTap: () async{
                            _singIn();
                          },
                          child: Center(
                              child: Text(
                                  'LOGIN',
                                  style: TextStyle(
                                      color: Colors.white,
                                      fontWeight: FontWeight.bold,
                                      fontFamily: 'Montserrat'
                                  )
                              )
                          )
                      ),
                    ),
                  ),
                  SizedBox(height: 15,),
                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                      InkWell(
                        onTap: () {
                          Navigator.of(context).pushNamed('/signup');
                        },
                        child: Text(
                            'Register',
                            style: TextStyle(
                                color: Colors.blueGrey,
                                fontFamily: 'Montserrat',
                                fontWeight: FontWeight.bold,
                                decoration: TextDecoration.underline
                            )
                        ),
                      )
                    ],
                  )
                ],
              ),
            )
          ],
        )
    );
  }

}
```

Signup.dart:

```dart
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

class SignupPage extends StatefulWidget {
  @override
  _SignupPageState createState() => _SignupPageState();
}

class _SignupPageState extends State<SignupPage> {

  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  late bool _sucess;
  late String _userEmail;

  void _register() async {
    final User? user = (
        await _auth.createUserWithEmailAndPassword(email: _emailController.text, password: _passwordController.text)
    ).user;

    if(user != null) {
      setState(() {
        _sucess = true;
        _userEmail = user.email!;
      });
    } else {
      setState(() {
        _sucess = false;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
        backgroundColor: Colors.pinkAccent[100],
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Container(
              child: Stack(
                children: <Widget>[
                  Container(
                    padding: EdgeInsets.fromLTRB(15, 110, 0, 0),
                    child: Text("Register",
                        style: TextStyle(
                            fontSize: 40, fontWeight: FontWeight.bold
```

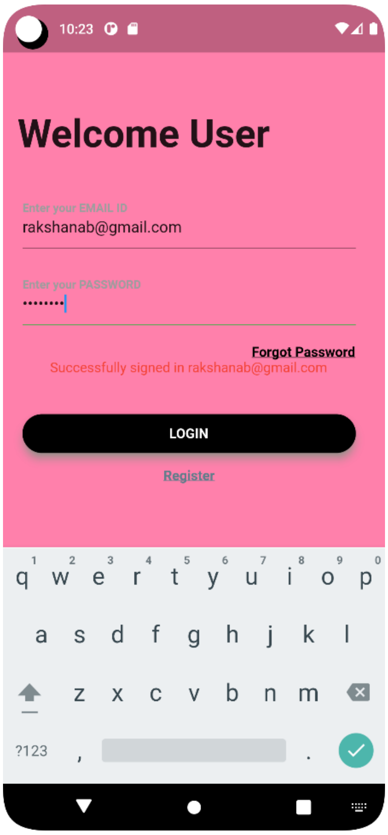


```
            )
        ),
      )
    ],
  ),
),
Container(
  padding: EdgeInsets.only(top: 35, left: 20, right: 30),
  child: Column(
    children: <Widget>[
      TextField(
        controller: _emailController,
        decoration: InputDecoration(
            labelText: 'Enter your EMAIL',
            labelStyle: TextStyle(
                fontFamily: 'Montserrat',
                fontWeight: FontWeight.bold,
                color: Colors.grey
            ),
            focusedBorder: UnderlineInputBorder(
              borderSide: BorderSide(color: Colors.green),
            )
        ),
      ),
      SizedBox(height: 20,),
      TextField(
        controller: _passwordController,
        decoration: InputDecoration(
            labelText: 'Enter your PASSWORD',
            labelStyle: TextStyle(
                fontFamily: 'Montserrat',
                fontWeight: FontWeight.bold,
                color: Colors.grey
            ),
            focusedBorder: UnderlineInputBorder(
              borderSide: BorderSide(color: Colors.green),
            )
        ),
        obscureText: true,
      ),
      SizedBox(height: 5.0,),
      SizedBox(height: 40,),
      Container(
        height: 40,
        child: Material(
          borderRadius: BorderRadius.circular(20),
          shadowColor: Colors.greenAccent,
          color: Colors.black,
          elevation: 7,
          child: GestureDetector(
              onTap: () async{
                _register();
              },
              child: Center(
                  child: Text(
                      'SIGNUP',
                      style: TextStyle(
```



```
                                      color: Colors.white,
                                      fontWeight: FontWeight.bold,
                                      fontFamily: 'Montserrat'
                                  )
                              )
                          )
                      ),
                    ),
                  ),
                  SizedBox(height: 15,),
                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                      InkWell(
                        onTap: () {
                          Navigator.of(context).pop();
                        },
                        child: Text(
                            'Go Back',
                            style: TextStyle(
                                color: Colors.blueGrey,
                                fontFamily: 'Montserrat',
                                fontWeight: FontWeight.bold,
                                decoration: TextDecoration.underline
                            )
                        ),
                      )
                    ],
                  )
                ],
              ),
            )
          ],
        )
    );
  }
}
```

Sample I/O:



Welcome User

Enter your EMAIL ID
rakshanab@gmail.com

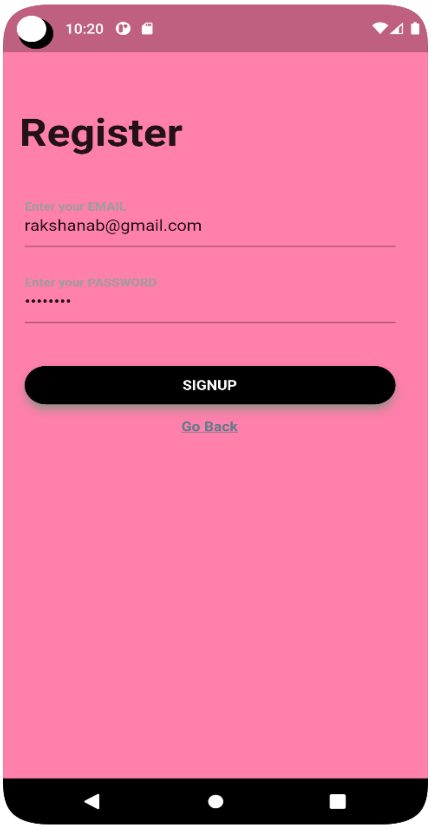
Enter your PASSWORD
••••••••

Forgot Password

Successfully signed in rakshanab@gmail.com

LOGIN

Register



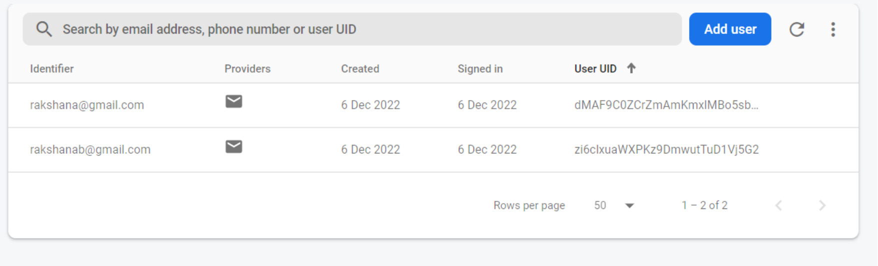
Register

Enter your EMAIL
rakshanab@gmail.com

Enter your PASSWORD
••••••••

SIGNUP

Go Back



Identifier	Providers	Created	Signed in	User UID ↑
rakshana@gmail.com	✉	6 Dec 2022	6 Dec 2022	dMAF9C0ZCrZmAmKmxIMBo5sb...
rakshanab@gmail.com	✉	6 Dec 2022	6 Dec 2022	zi6clxuaWXPKz9DmwutTuD1Vj5G2

Rows per page 50 1 – 2 of 2

Result:

Thus, a simple application using a database has been successfully implemented using android studio.

Expt. No: 6

6)Develop an application that makes use of RSS Feed.

Aim:

To develop an Android application for RSS (Really Simple Syndication) Feed using Android Studio.

Software and Hardware Requirements:

Software:
Android Studio

Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

1. Open Android Studio and create a new project.
2. Select Empty Activity.
3. The Main.java file makes use of XMLPullParser that parses through the RSS XML file.
4. Add the following in android manifest xml file
   <uses-permission android:name="android.permission.INTERNET"/>

Code:

MainActivity.java

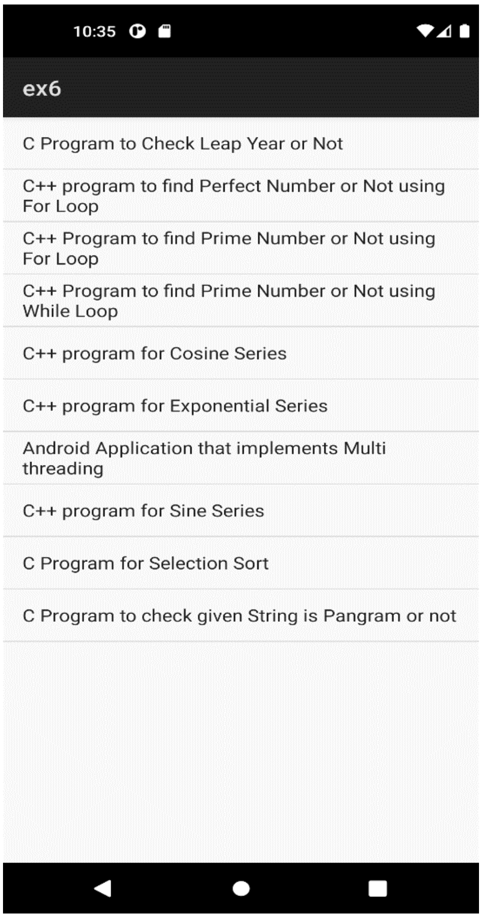
```
package com.example.rss;


import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
  List headlines;
  List links;

  @Override
  protected void onCreate(Bundle savedInstanceState)
  {
     super.onCreate(savedInstanceState);
     new MyAsyncTask().execute();
  }

  class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
  {
     @Override
     protected ArrayAdapter doInBackground(Object[] params)
     {
        headlines = new ArrayList();
        links = new ArrayList();
        try
        {
           URL url = new URL("https://codingconnect.net/feed");
           XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
           factory.setNamespaceAware(false);
           XmlPullParser xpp = factory.newPullParser();

           // We will get the XML from an input stream
           xpp.setInput(getInputStream(url), "UTF_8");
           boolean insideItem = false;

           // Returns the type of current event: START_TAG, END_TAG, etc..
           int eventType = xpp.getEventType();
           while (eventType != XmlPullParser.END_DOCUMENT)
           {
              if (eventType == XmlPullParser.START_TAG)
              {
                 if (xpp.getName().equalsIgnoreCase("item"))
                 {
                    insideItem = true;
                 }
                 else if (xpp.getName().equalsIgnoreCase("title"))
                 {
                    if (insideItem)
                       headlines.add(xpp.nextText()); //extract the headline
                 }
                 else if (xpp.getName().equalsIgnoreCase("link"))
                 {
                    if (insideItem)
                       links.add(xpp.nextText()); //extract the link of article
                 }
              }
              else if(eventType==XmlPullParser.END_TAG && xpp.getName().equalsIgnoreCase("item"))
```

```
                {
                    insideItem=false;
                }
                eventType = xpp.next(); //move to next element
            }

        }
        catch (MalformedURLException e)
        {
            e.printStackTrace();
        }
        catch (XmlPullParserException e)
        {
            e.printStackTrace();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        return null;
    }
    protected void onPostExecute(ArrayAdapter adapter)
    {
        adapter = new ArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1, headlines);
        setListAdapter(adapter);
    }
}

@Override
protected void onListItemClick(ListView l, View v, int position, long id)
{
    Uri uri = Uri.parse((links.get(position)).toString());
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}

public InputStream getInputStream(URL url)
{
    try
    {
        return url.openConnection().getInputStream();
    }
    catch (IOException e)
    {
        return null;
    }
}
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >
  <ListView
     android:id="@+id/listView"
     android:layout_width="match_parent"
     android:layout_height="wrap_content" />

</LinearLayout>
```

SAMPLE I/O:



10:35

ex6

C Program to Check Leap Year or Not

C++ program to find Perfect Number or Not using For Loop

C++ Program to find Prime Number or Not using For Loop

C++ Program to find Prime Number or Not using While Loop

C++ program for Cosine Series

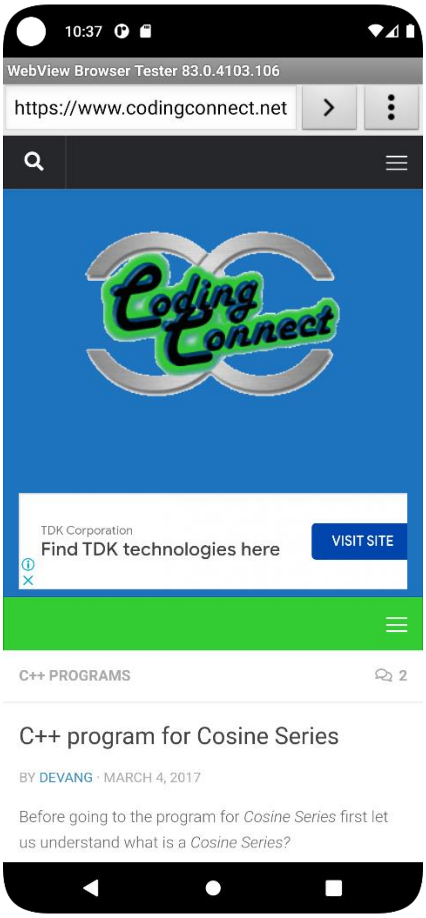
C++ program for Exponential Series

Android Application that implements Multi threading

C++ program for Sine Series

C Program for Selection Sort

C Program to check given String is Pangram or not



10:37

WebView Browser Tester 83.0.4103.106

https://www.codingconnect.net

TDK Corporation

Find TDK technologies here

VISIT SITE

C++ PROGRAMS 2

C++ program for Cosine Series

BY DEVANG · MARCH 4, 2017

Before going to the program for *Cosine Series* first let us understand what is a *Cosine Series?*

Result:

Thus, an application for RSS (Really Simple Syndication) Feed using Android Studio is implemented successfully.

Expt.No: 7

7)Implement an application that implements multi-threading.

Aim:

To develop an Android Application that implements Multithreading.

Software and Hardware Requirements:

Software:

Android Studio

Hardware:

Preferably 8GB+ RAM Laptop/Desktop

Procedure:

Multithreading:

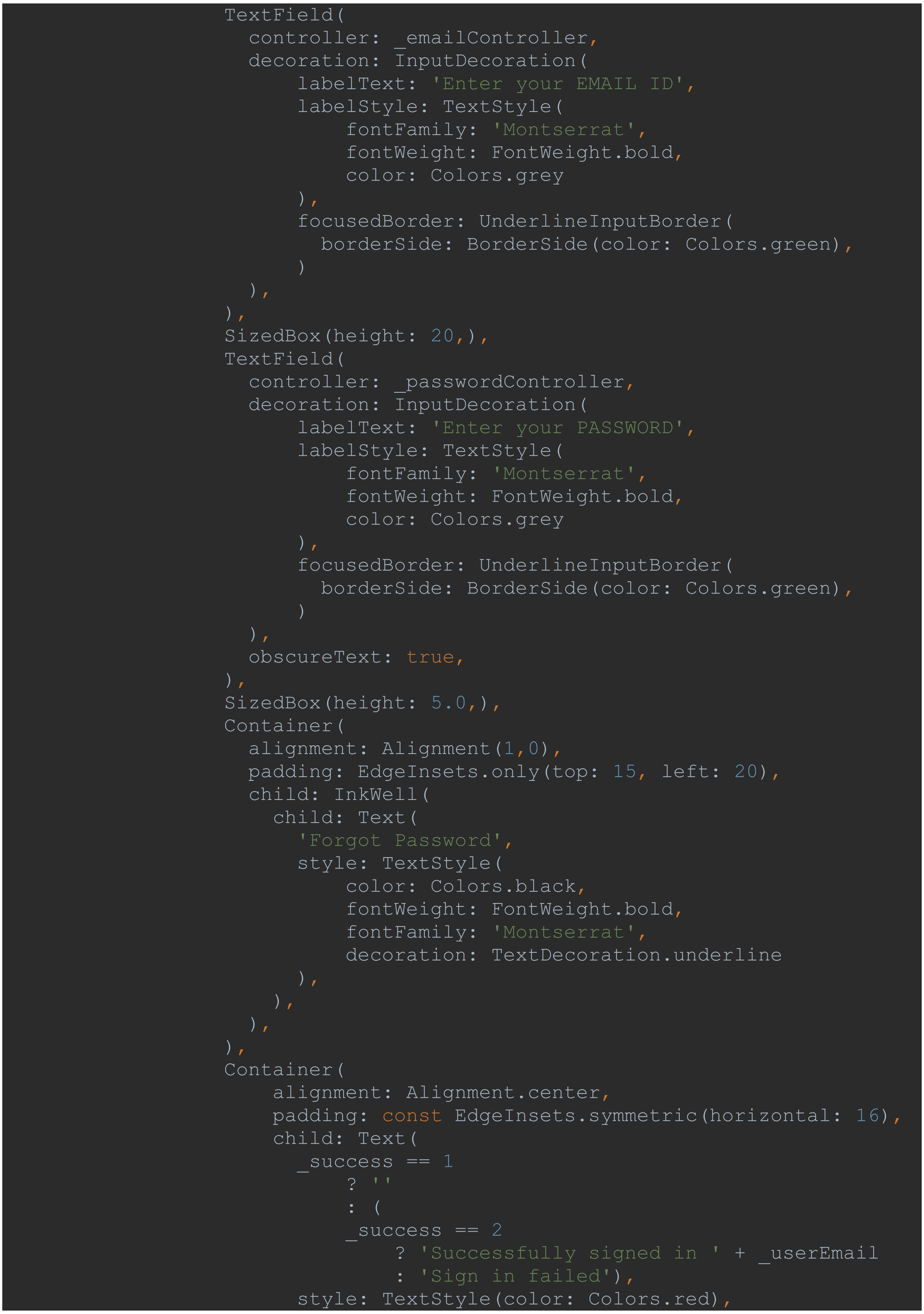
Multithreading is the ability of a program or an operating system to enable more than one user at a time without requiring multiple copies of the program running on the computer. Multithreading can also handle multiple requests from the same user.

- In pubspec.yaml add these :
- firebase_core
- firebase_auth
- And then click pub get.
  - Use ‘firebase login’ command to login to google account
  - Use ‘flutterfire configure’ to add a firebase project to the application.
  - Run main.dart file
  - FirebaseAuth.instance.currentUser is used to get the current user object
  - Under authenthication in firbase the users database will be visible.

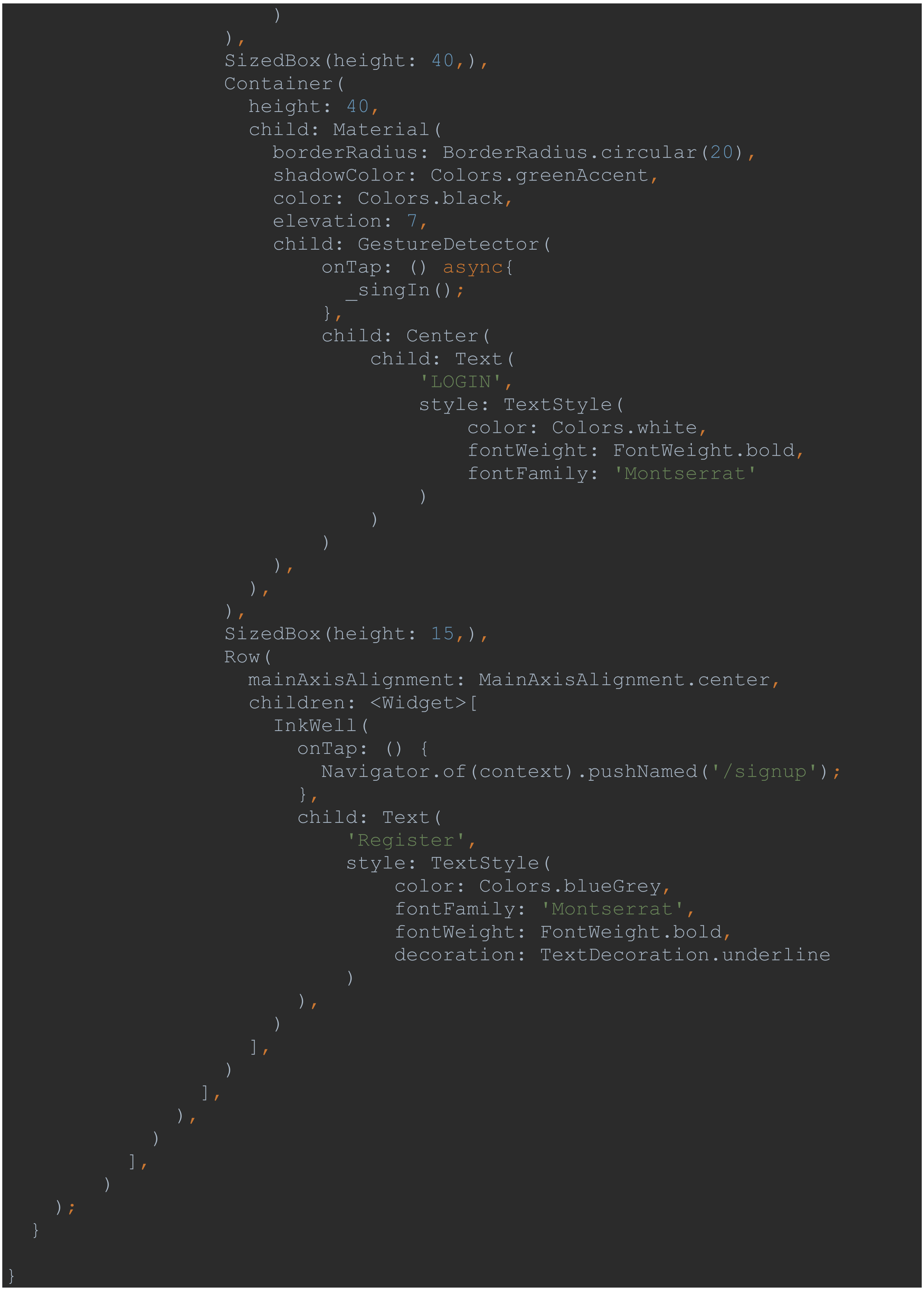
Code:

Main.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'signup.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  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
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter
run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the applica-
tion
        // is not restarted.
        backgroundColor: Colors.pinkAccent[200],
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
      debugShowCheckedModeBanner: false,
      routes: <String, WidgetBuilder>{
        '/signup': (BuildContext context) => new SignupPage()
      },
    );
  }
}

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> {

  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  int _success = 1;
  String _userEmail = "";

  void _singIn() async {
    final User? user = (await _auth.signInWithEmailAndPassword(email:
_emailController.text, password: _passwordController.text)).user;

    if(user != null) {
      setState(() {
        _success = 2;
        _userEmail = user.email!;
      });
    } else {
      setState(() {
        _success = 3;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
        backgroundColor: Colors.pinkAccent[100],
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Container(
              child: Stack(
                children: <Widget>[
                  Container(
                    padding: EdgeInsets.fromLTRB(15, 110, 0, 0),
                    child: Text("Welcome User",
                        style: TextStyle(
                            fontSize: 40, fontWeight: FontWeight.bold
                        )
                    ),
                  )
                ],
              ),
            ),
            Container(
              padding: EdgeInsets.only(top: 35, left: 20, right: 30),
              child: Column(
                children: <Widget>[
```



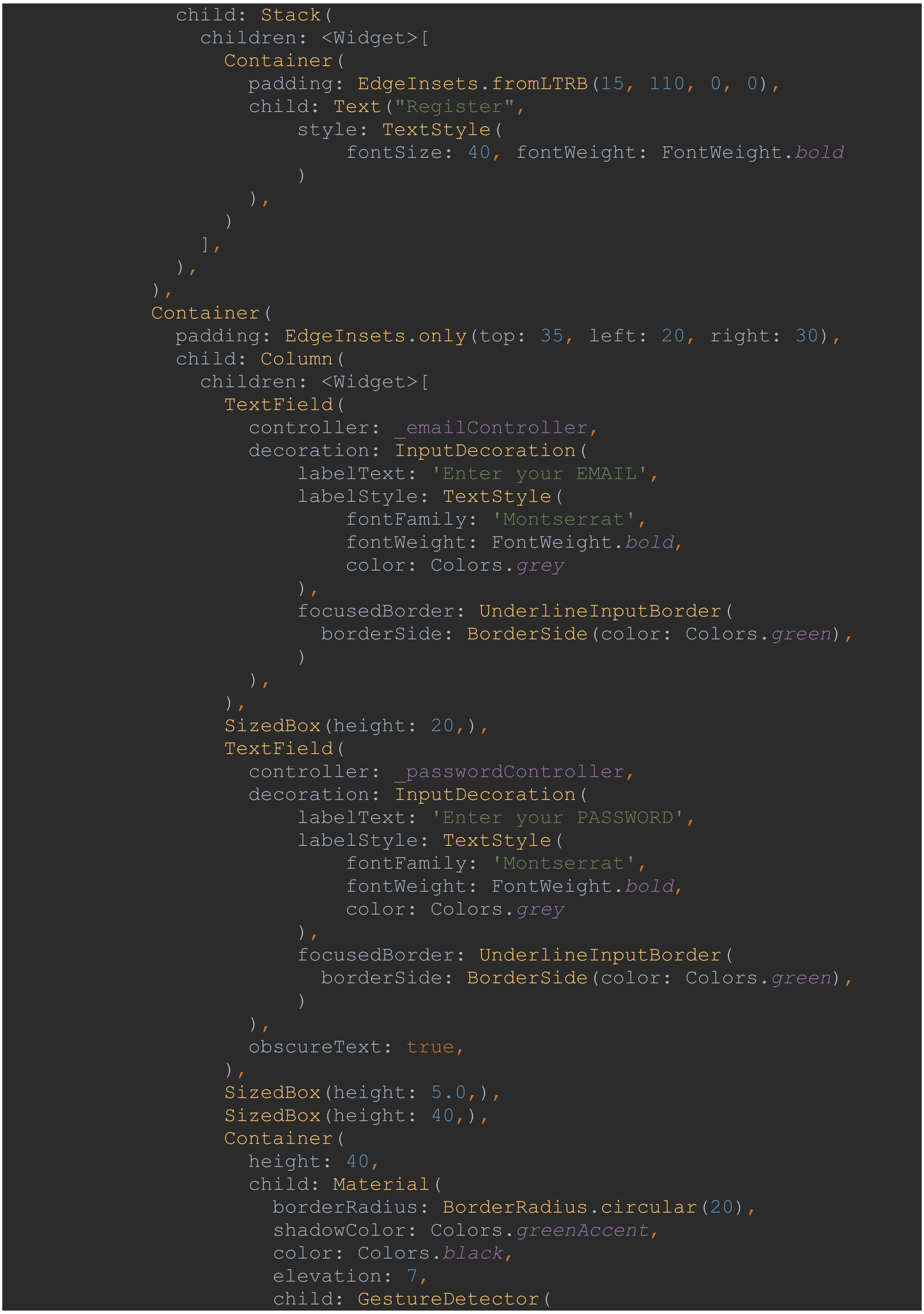
```
                  TextField(
                    controller: _emailController,
                    decoration: InputDecoration(
                        labelText: 'Enter your EMAIL ID',
                        labelStyle: TextStyle(
                            fontFamily: 'Montserrat',
                            fontWeight: FontWeight.bold,
                            color: Colors.grey
                        ),
                        focusedBorder: UnderlineInputBorder(
                          borderSide: BorderSide(color: Colors.green),
                        )
                    ),
                  ),
                  SizedBox(height: 20,),
                  TextField(
                    controller: _passwordController,
                    decoration: InputDecoration(
                        labelText: 'Enter your PASSWORD',
                        labelStyle: TextStyle(
                            fontFamily: 'Montserrat',
                            fontWeight: FontWeight.bold,
                            color: Colors.grey
                        ),
                        focusedBorder: UnderlineInputBorder(
                          borderSide: BorderSide(color: Colors.green),
                        )
                    ),
                    obscureText: true,
                  ),
                  SizedBox(height: 5.0,),
                  Container(
                    alignment: Alignment(1,0),
                    padding: EdgeInsets.only(top: 15, left: 20),
                    child: InkWell(
                      child: Text(
                        'Forgot Password',
                        style: TextStyle(
                            color: Colors.black,
                            fontWeight: FontWeight.bold,
                            fontFamily: 'Montserrat',
                            decoration: TextDecoration.underline
                        ),
                      ),
                    ),
                  ),
                  Container(
                      alignment: Alignment.center,
                      padding: const EdgeInsets.symmetric(horizontal: 16),
                      child: Text(
                        _success == 1
                            ? ''
                            : (
                            _success == 2
                                ? 'Successfully signed in ' + _userEmail
                                : 'Sign in failed'),
                        style: TextStyle(color: Colors.red),
```



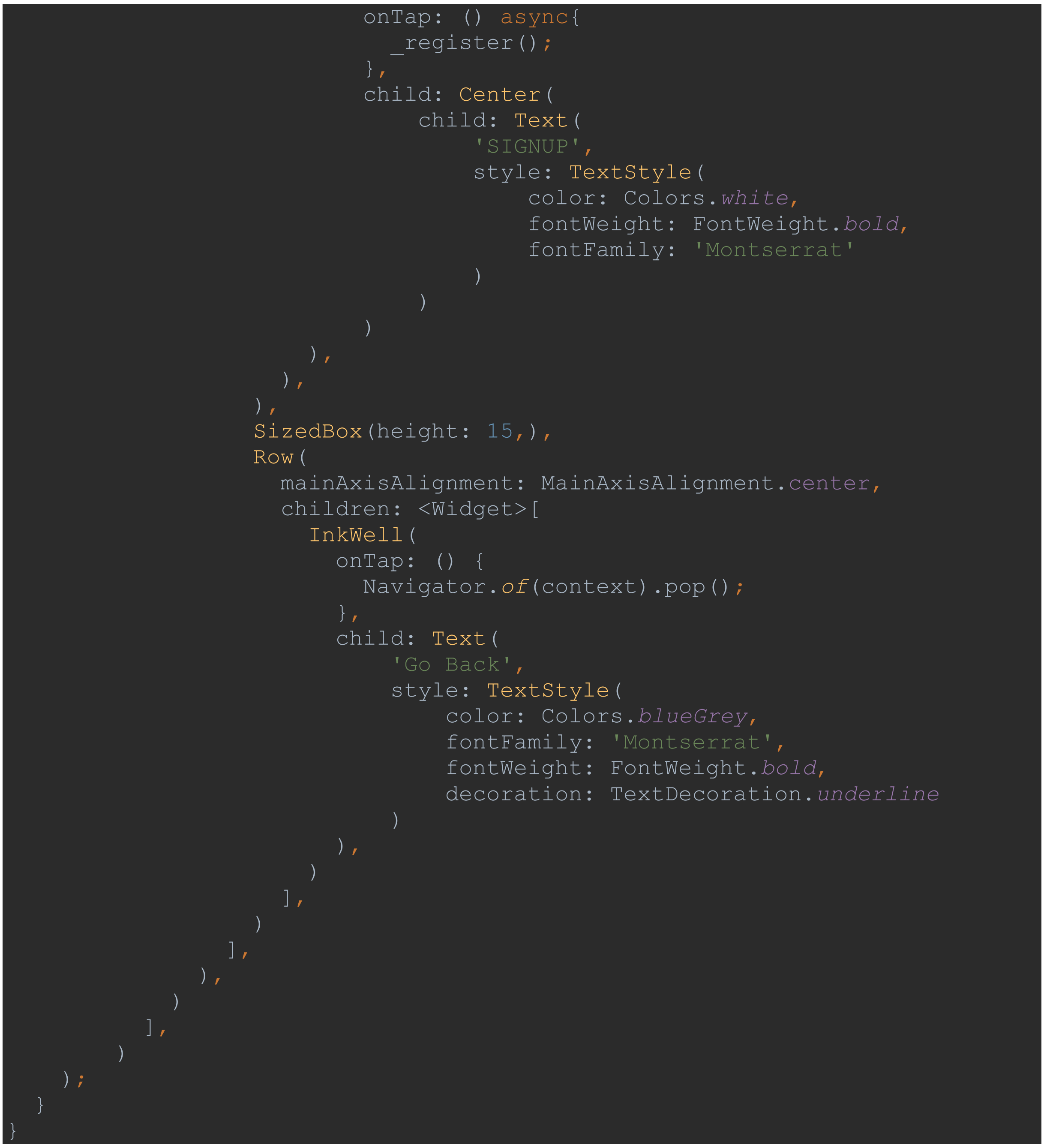
```
                      )
                  ),
                  SizedBox(height: 40,),
                  Container(
                    height: 40,
                    child: Material(
                      borderRadius: BorderRadius.circular(20),
                      shadowColor: Colors.greenAccent,
                      color: Colors.black,
                      elevation: 7,
                      child: GestureDetector(
                          onTap: () async{
                            _singIn();
                          },
                          child: Center(
                              child: Text(
                                  'LOGIN',
                                  style: TextStyle(
                                      color: Colors.white,
                                      fontWeight: FontWeight.bold,
                                      fontFamily: 'Montserrat'
                                  )
                              )
                          )
                      ),
                    ),
                  ),
                  SizedBox(height: 15,),
                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                      InkWell(
                        onTap: () {
                          Navigator.of(context).pushNamed('/signup');
                        },
                        child: Text(
                            'Register',
                            style: TextStyle(
                                color: Colors.blueGrey,
                                fontFamily: 'Montserrat',
                                fontWeight: FontWeight.bold,
                                decoration: TextDecoration.underline
                            )
                        ),
                      )
                    ],
                  )
                ],
              ),
            )
          ],
        )
    );
  }

}
```

Signup.dart:

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

class SignupPage extends StatefulWidget {
  @override
  _SignupPageState createState() => _SignupPageState();
}

class _SignupPageState extends State<SignupPage> {

  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  late bool _sucess;
  late String _userEmail;

  void _register() async {
    final User? user = (
        await _auth.createUserWithEmailAndPassword(email: _emailController.text, password: _passwordController.text)
    ).user;

    if(user != null) {
      setState(() {
        _sucess = true;
        _userEmail = user.email!;
      });
    } else {
      setState(() {
        _sucess = false;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
        backgroundColor: Colors.pinkAccent[100],
        body: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Container(
```

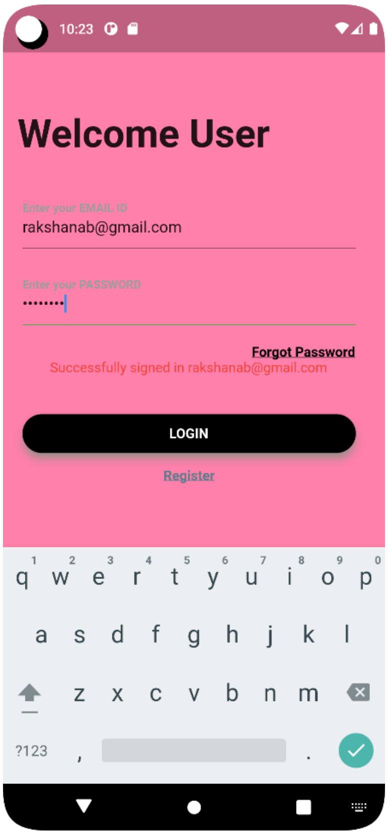


```
  child: Stack(
    children: <Widget>[
      Container(
        padding: EdgeInsets.fromLTRB(15, 110, 0, 0),
        child: Text("Register",
            style: TextStyle(
                fontSize: 40, fontWeight: FontWeight.bold
            )
        ),
      )
    ],
  ),
),
Container(
  padding: EdgeInsets.only(top: 35, left: 20, right: 30),
  child: Column(
    children: <Widget>[
      TextField(
        controller: _emailController,
        decoration: InputDecoration(
            labelText: 'Enter your EMAIL',
            labelStyle: TextStyle(
                fontFamily: 'Montserrat',
                fontWeight: FontWeight.bold,
                color: Colors.grey
            ),
            focusedBorder: UnderlineInputBorder(
              borderSide: BorderSide(color: Colors.green),
            )
        ),
      ),
      SizedBox(height: 20,),
      TextField(
        controller: _passwordController,
        decoration: InputDecoration(
            labelText: 'Enter your PASSWORD',
            labelStyle: TextStyle(
                fontFamily: 'Montserrat',
                fontWeight: FontWeight.bold,
                color: Colors.grey
            ),
            focusedBorder: UnderlineInputBorder(
              borderSide: BorderSide(color: Colors.green),
            )
        ),
        obscureText: true,
      ),
      SizedBox(height: 5.0,),
      SizedBox(height: 40,),
      Container(
        height: 40,
        child: Material(
          borderRadius: BorderRadius.circular(20),
          shadowColor: Colors.greenAccent,
          color: Colors.black,
          elevation: 7,
          child: GestureDetector(
```



```
                          onTap: () async{
                            _register();
                          },
                          child: Center(
                              child: Text(
                                  'SIGNUP',
                                  style: TextStyle(
                                      color: Colors.white,
                                      fontWeight: FontWeight.bold,
                                      fontFamily: 'Montserrat'
                                  )
                              )
                          )
                      ),
                    ),
                  ),
                  SizedBox(height: 15,),
                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: <Widget>[
                      InkWell(
                        onTap:  () {
                          Navigator.of(context).pop();
                        },
                        child: Text(
                            'Go Back',
                            style: TextStyle(
                                color: Colors.blueGrey,
                                fontFamily: 'Montserrat',
                                fontWeight: FontWeight.bold,
                                decoration: TextDecoration.underline
                            )
                        ),
                      )
                    ],
                  )
                ],
              ),
            )
          ],
        )
    );
  }
}
```

Sample I/O:

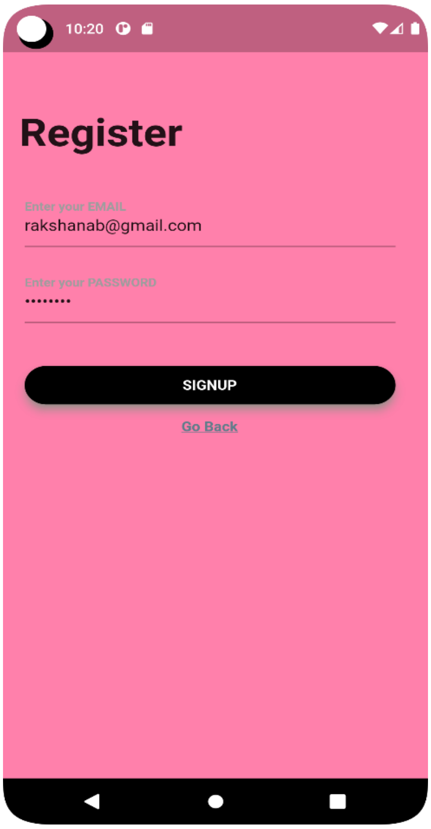


Welcome User

Enter your EMAIL ID
rakshanab@gmail.com

Enter your PASSWORD
••••••••

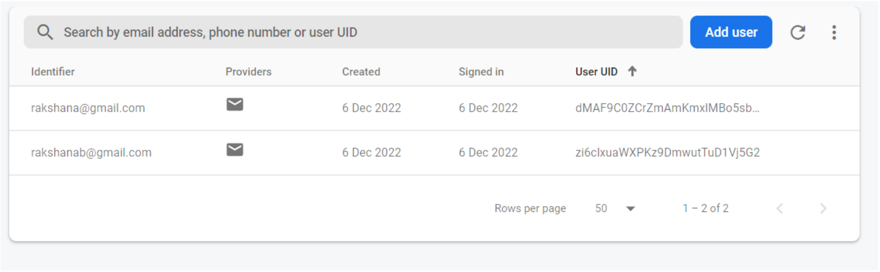
Successfully signed in rakshanab@gmail.com



Register

Enter your EMAIL
rakshanab@gmail.com

Enter your PASSWORD
••••••••



Identifier	Providers	Created	Signed in	User UID ↑
rakshana@gmail.com	✉	6 Dec 2022	6 Dec 2022	dMAF9C0ZCrZmAmKmxlMBo5sb...
rakshanab@gmail.com	✉	6 Dec 2022	6 Dec 2022	zi6clxuaWXPKz9DmwutTuD1Vj5G2

Result:

Thus, an Android Application that implements Multithreading has been successfully implemented.

Expt. No: 8

8)Develop a native application that uses GPS location information.

Aim:

To develop an Android Application that uses GPS location information.

Procedure:

GPS Location:

GPS coordinates are a unique identifier of a precise geographic location on the earth, Coordinates, in this context, are points of intersection in a grid system. GPS coordinates are usually expressed as the combination of latitude and longitude.

- Install the flutter_sensors and the location dependencies.
- Import the following:

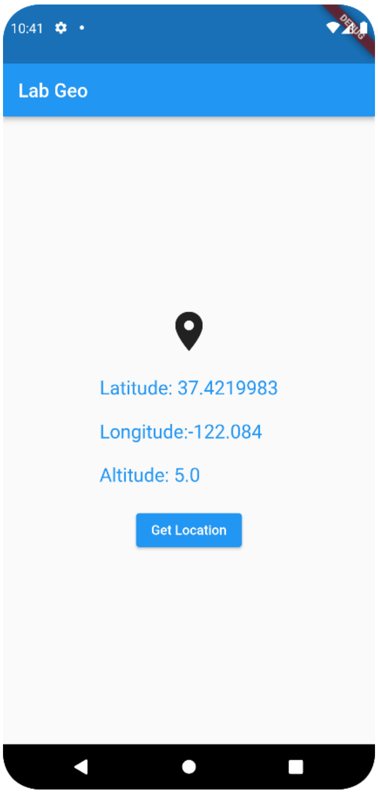
import 'package:flutter_sensors/flutter_sensors.dart';
import 'package:location/location.dart';
-Ask for the permission to retrieve the location using location.requestPermission()
-Then get the location using the following method : location.getLocation()

Code:

Main.dart:

```
import 'package:flutter_sensors/flutter_sensors.dart';
import 'package:flutter/material.dart';
import 'dart:math';
import 'package:location/location.dart';

void main() {


  return runApp(const location_wid());
}
class location_wid extends StatefulWidget {


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

  @override
  _location_widState createState() => _location_widState();
}

class _location_widState extends State<location_wid> {
  Location location = Location();
  bool _isServiceEnabled = false;
  PermissionStatus _permissionGranted = PermissionStatus.denied;
  LocationData _locationData = LocationData.fromMap({});

  @override
  void initState() {

    super.initState();
  }
  @override
  Widget build(BuildContext context) {

    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title : Text("Lab Geo")),
        body: Center(child : Column(mainAxisAlignment : MainAxisAlignment.center,children : [
          Icon(Icons.location_on, size: 50,),
          Text('\nLatitude: ' + _locationData.latitude.toString() + '\n\nLongitude:'  + _locationData.longitude.toString() + '\n\nAltitude: ' + _locationData.altitude.toString() + '\n', style: TextStyle(fontSize: 20, color: Colors.blue))
          ,ElevatedButton(onPressed: ()async{
            _isServiceEnabled = await location.serviceEnabled();
            if(!_isServiceEnabled)
              _isServiceEnabled = await location.requestService();
            print(_isServiceEnabled);
            PermissionStatus permission = await location.hasPermission();
            if(permission==PermissionStatus.denied)
              permission = await location.requestPermission();
            print(permission==PermissionStatus.granted);
            _locationData = await location.getLocation();
            // print(_locationData.latitude);
            setState(() {
            });
          },child:  Text('Get Location'),
          ),

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

Sample I/O:



Latitude: 37.4219983

Longitude:-122.084

Altitude: 5.0

Result:

Thus, an Android Application that uses GPS location was successfully implemented.

Expt. No: 9

9) Implement an application that writes data to the SD card

Aim:

To develop an Android Application that writes data to the SD card.

Procedure:

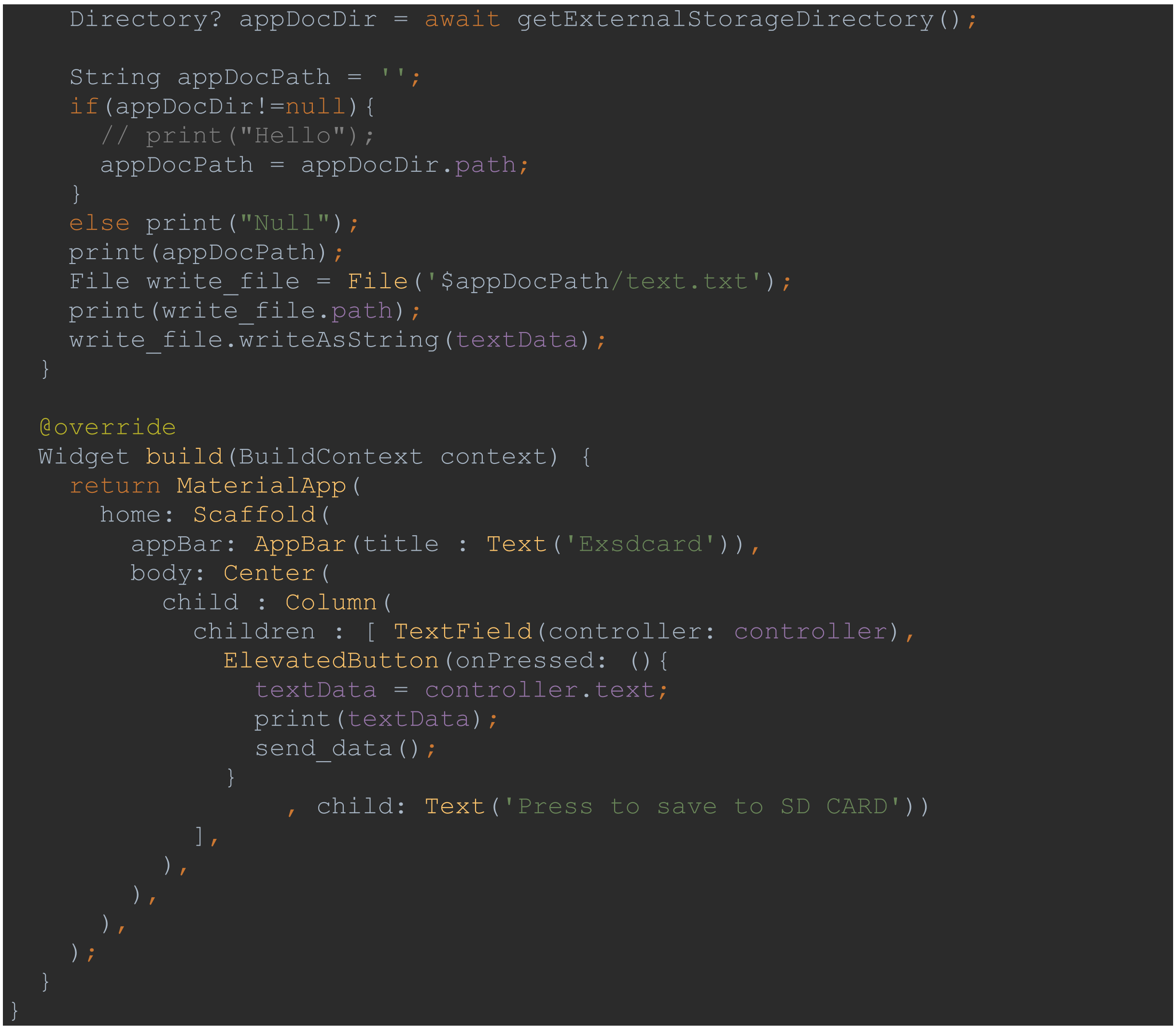
SD card:
A Secure Digital (SD) card is a tiny flash memory card designed for high-capacity memory and various portable devices, such as car navigation systems, cellular phones, e-books, PDAs, smartphones, digital cameras, music players, digital video camcorders and personal computers.

-Install the services and the path_provider dependencies.
- Import the following:
import 'package:flutter/services.dart';
import 'package:path_provider/path_provider.dart';
-Get the external directory path using
getExternalStorageDirectory()
-Then check if the path is null or not. If not then write to a file when the button is clicked

Code:

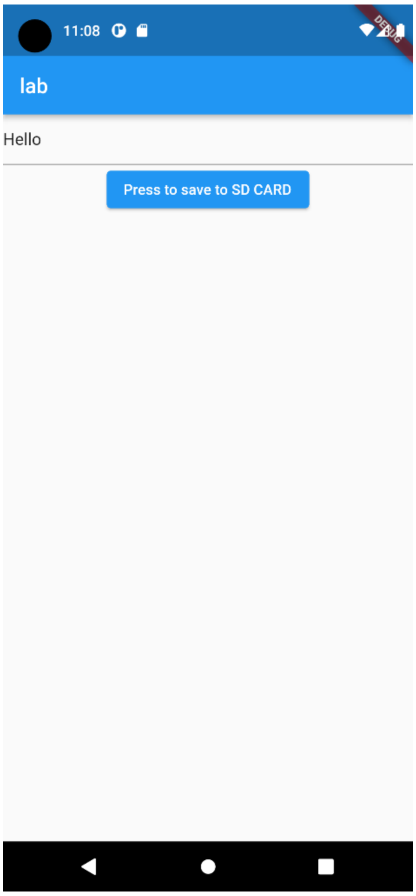
Main.dart:

```
import 'dart:io';
import 'dart:typed_data';

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'dart:math';
import 'package:path_provider/path_provider.dart';
void main() => runApp(const MyApp());
class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  _MyAppState createState() => _MyAppState();
}


class _MyAppState extends State<MyApp> {
  @override
  String textData = '';
  TextEditingController controller = TextEditingController();
  void send_data() async{
```



```
    Directory? appDocDir = await getExternalStorageDirectory();

    String appDocPath = '';
    if(appDocDir!=null){
      // print("Hello");
      appDocPath = appDocDir.path;
    }
    else print("Null");
    print(appDocPath);
    File write_file = File('$appDocPath/text.txt');
    print(write_file.path);
    write_file.writeAsString(textData);
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title : Text('Exsdcard')),
        body: Center(
          child : Column(
            children : [ TextField(controller: controller),
              ElevatedButton(onPressed: (){
                textData = controller.text;
                print(textData);
                send_data();
              }
                  , child: Text('Press to save to SD CARD'))
            ],
          ),
        ),
      ),
    );
  }
}
```

Sample I/O:



Result:

Thus, an Android Application that writes data to the SD Card was successfully implemented.

Expt.No: 10

10)Implement an application that creates an alert upon receiving a message.

Aim:

To develop an Android Application that creates an alert upon receiving a message.

Procedure:

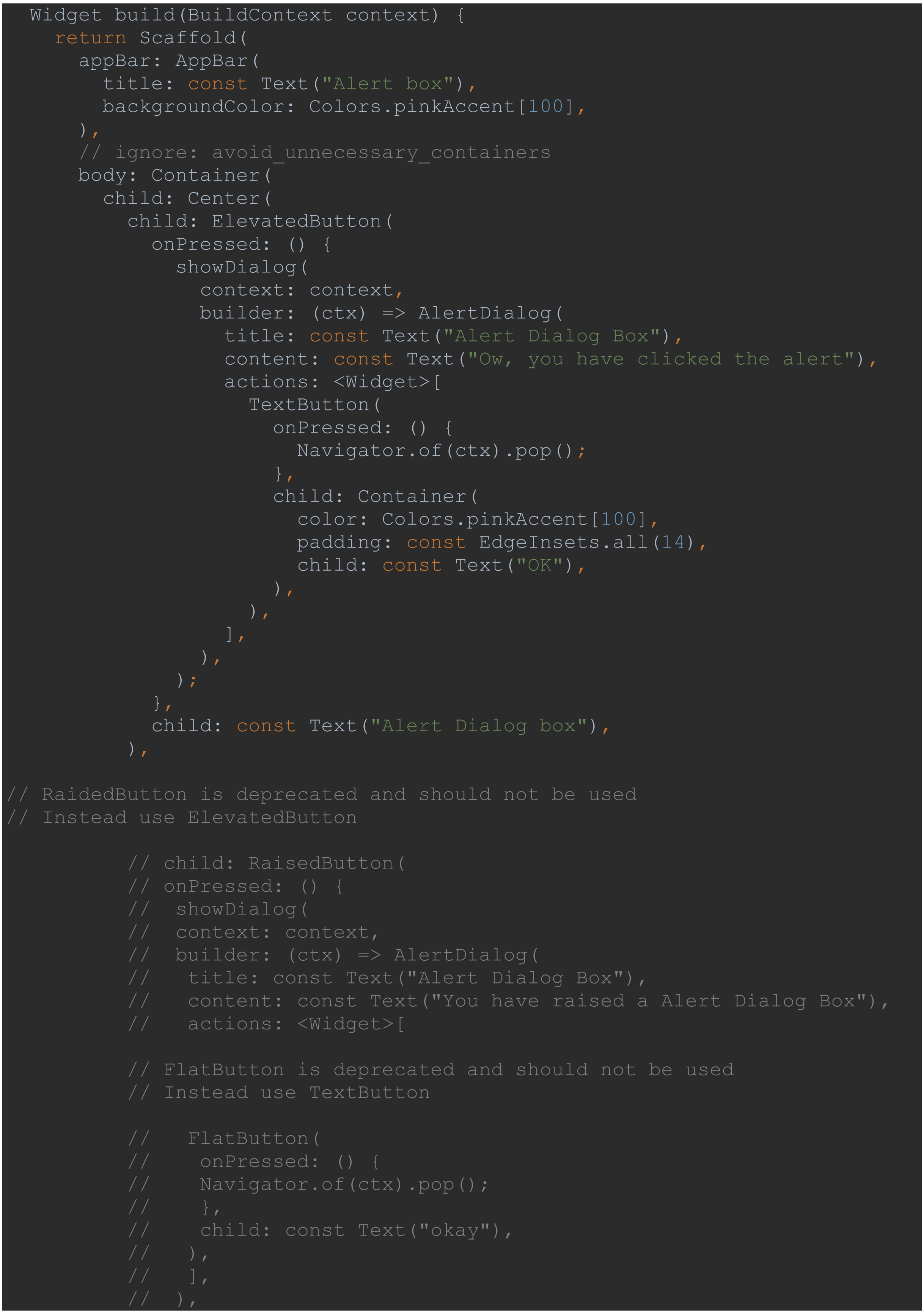
Alert Dialog box:
Alert Dialog box informs the user about the situation that requires acknowledgment. Alert Box is a prompt that takes user confirmation.

- Import the following:
import 'package:flutter/material.dart';
-A button is created, to trigger the alert dialog box
- When the button is pressed showDialog widget is used
- This makes us to implement alert dialog box.

Code:

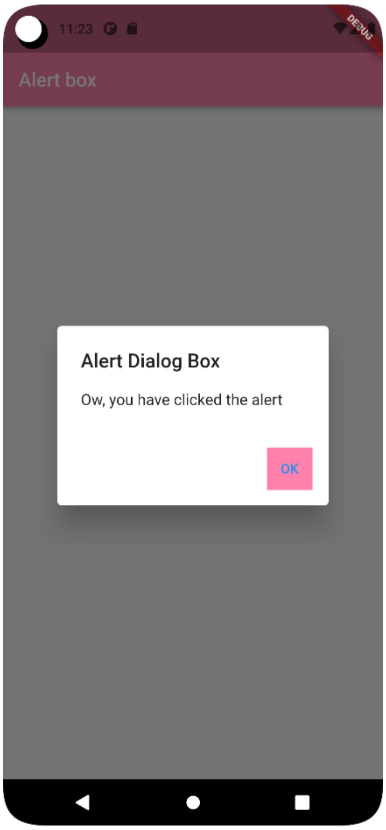
Main.dart:

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

void main() => runApp(const MyApp());

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
// ignore: library_private_types_in_public_api
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
```



```
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Alert box"),
        backgroundColor: Colors.pinkAccent[100],
      ),
      // ignore: avoid_unnecessary_containers
      body: Container(
        child: Center(
          child: ElevatedButton(
            onPressed: () {
              showDialog(
                context: context,
                builder: (ctx) => AlertDialog(
                  title: const Text("Alert Dialog Box"),
                  content: const Text("Ow, you have clicked the alert"),
                  actions: <Widget>[
                    TextButton(
                      onPressed: () {
                        Navigator.of(ctx).pop();
                      },
                      child: Container(
                        color: Colors.pinkAccent[100],
                        padding: const EdgeInsets.all(14),
                        child: const Text("OK"),
                      ),
                    ),
                  ],
                ),
              );
            },
            child: const Text("Alert Dialog box"),
          ),

// RaidedButton is deprecated and should not be used
// Instead use ElevatedButton

          // child: RaisedButton(
          // onPressed: () {
          //  showDialog(
          //  context: context,
          //  builder: (ctx) => AlertDialog(
          //   title: const Text("Alert Dialog Box"),
          //   content: const Text("You have raised a Alert Dialog Box"),
          //   actions: <Widget>[

          // FlatButton is deprecated and should not be used
          // Instead use TextButton

          //   FlatButton(
          //    onPressed: () {
          //    Navigator.of(ctx).pop();
          //    },
          //    child: const Text("okay"),
          //   ),
          //   ],
          //  ),
          //  );
          // },
          // child: const Text("Show alert Dialog box"),
          // ),
        ),
      ),
    );
  }
}
```

OUTPUT:



Alert box

Alert Dialog Box

Ow, you have clicked the alert

RESULT:

Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

Expt.No: 11

11)Write a mobile application that creates an alarm clock.

Aim:

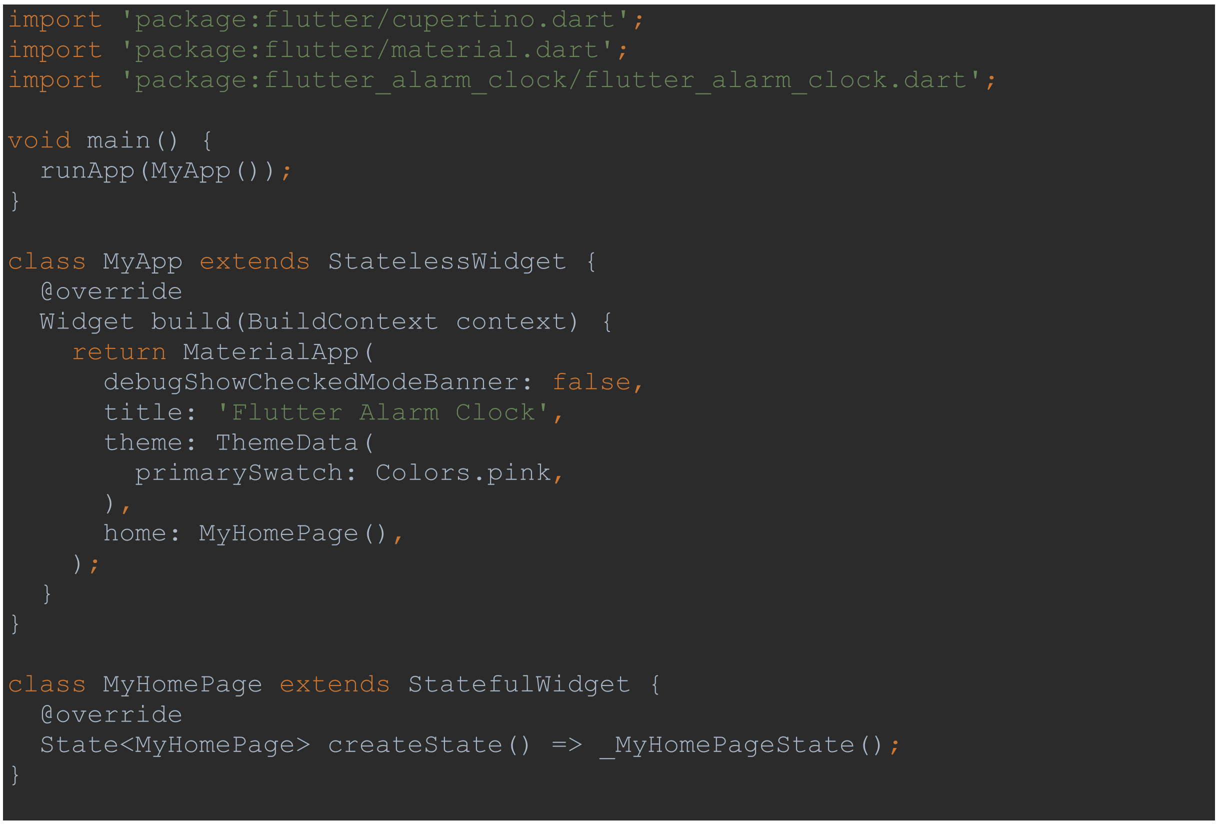
To develop an Android Application that creates an alert upon receiving a message.

Procedure:

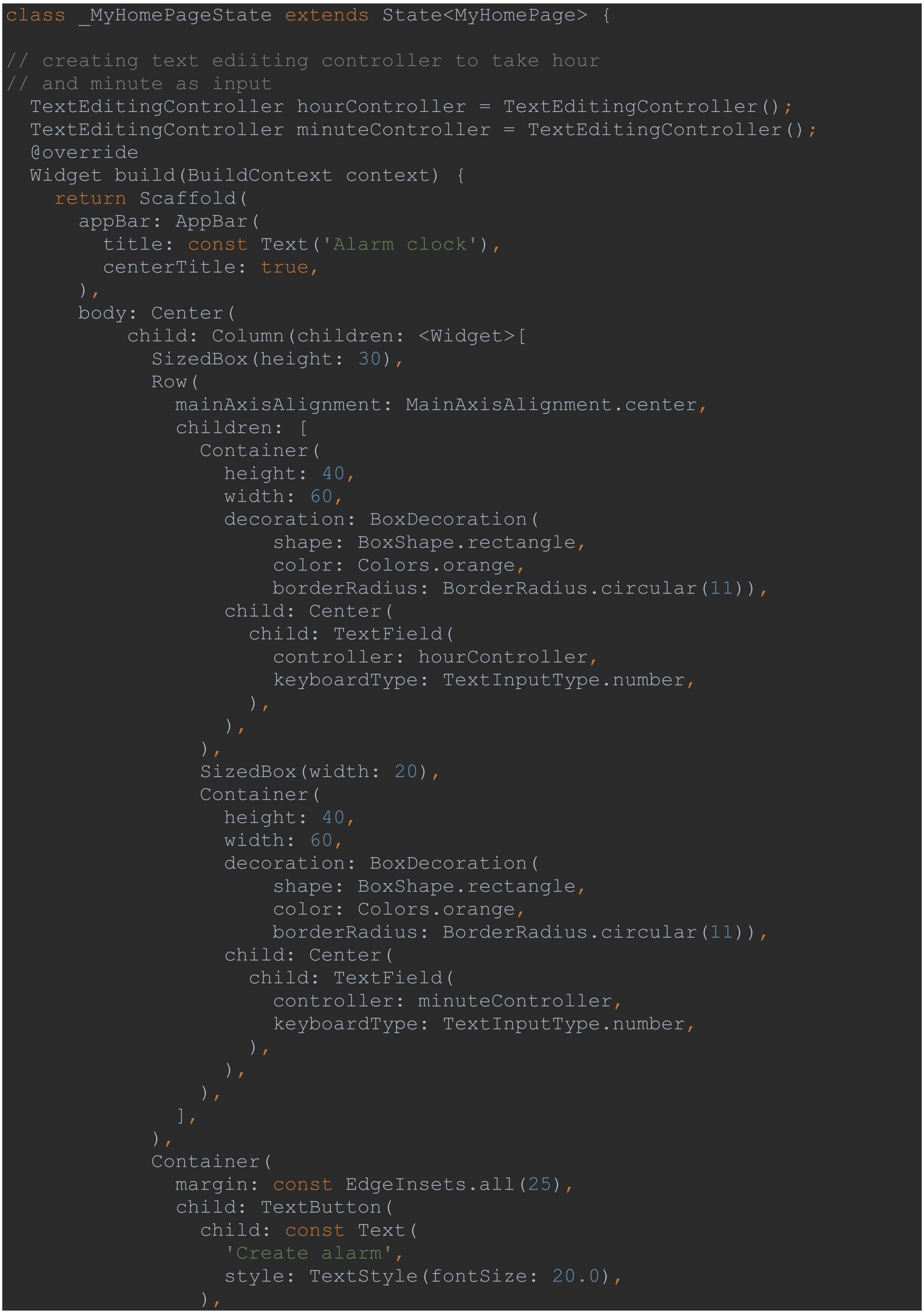
- Install the flutter_alarm_clock dependency.
-Import the following:
import 'package:flutter_alarm_clock/flutter_alarm_clock.dart';
-Two texteditingcontrollers are created, one for hour and the other for minute.
- Then the button create alarm is created.
- A snackbar will be shown to show that the alarm has been created.

Code:

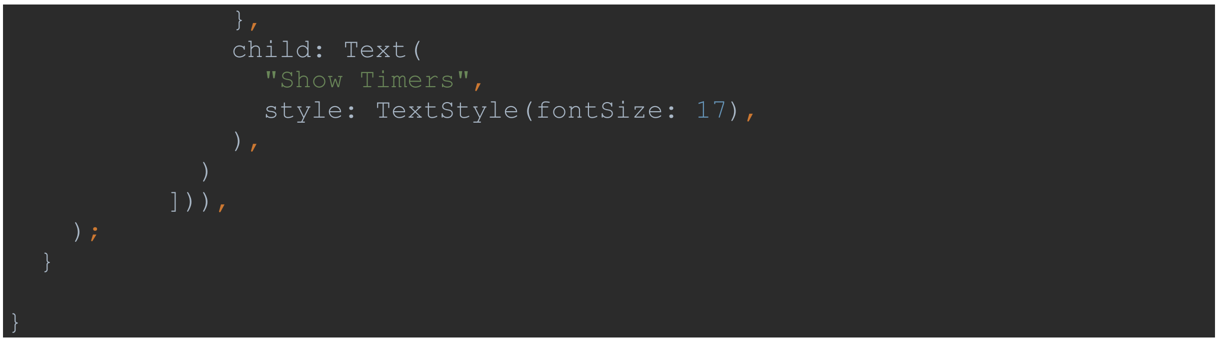
Main.dart:



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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Alarm Clock',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```

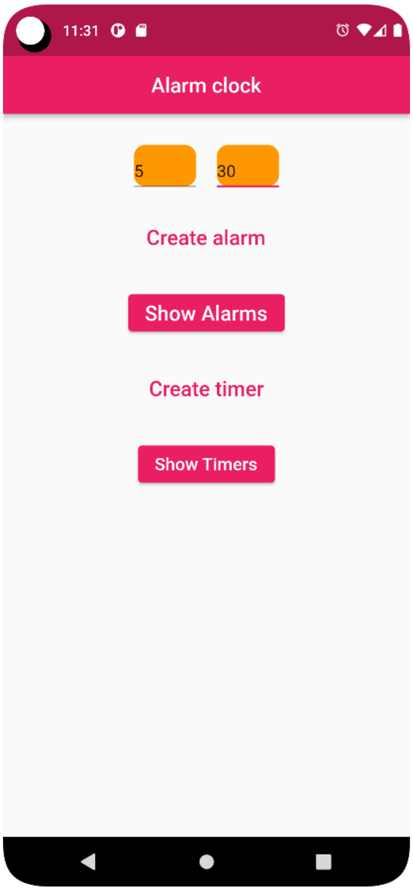


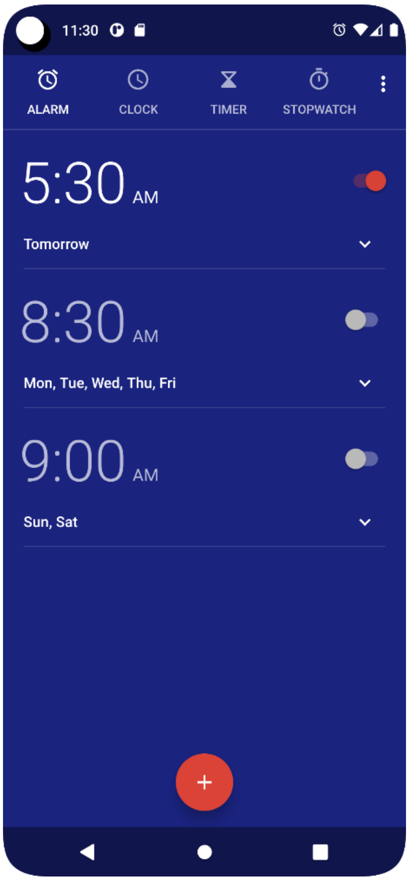
```
class _MyHomePageState extends State<MyHomePage> {

// creating text ediiting controller to take hour
// and minute as input
  TextEditingController hourController = TextEditingController();
  TextEditingController minuteController = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Alarm clock'),
        centerTitle: true,
      ),
      body: Center(
          child: Column(children: <Widget>[
            SizedBox(height: 30),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Container(
                  height: 40,
                  width: 60,
                  decoration: BoxDecoration(
                      shape: BoxShape.rectangle,
                      color: Colors.orange,
                      borderRadius: BorderRadius.circular(11)),
                  child: Center(
                    child: TextField(
                      controller: hourController,
                      keyboardType: TextInputType.number,
                    ),
                  ),
                ),
                SizedBox(width: 20),
                Container(
                  height: 40,
                  width: 60,
                  decoration: BoxDecoration(
                      shape: BoxShape.rectangle,
                      color: Colors.orange,
                      borderRadius: BorderRadius.circular(11)),
                  child: Center(
                    child: TextField(
                      controller: minuteController,
                      keyboardType: TextInputType.number,
                    ),
                  ),
                ),
              ],
            ),
            Container(
              margin: const EdgeInsets.all(25),
              child: TextButton(
                child: const Text(
                  'Create alarm',
                  style: TextStyle(fontSize: 20.0),
                ),
                onPressed: () {
                  int hour;
                  int minutes;
                  hour = int.parse(hourController.text);
                  minutes = int.parse(minuteController.text);

                  // creating alarm after converting hour
                  // and minute into integer
                  FlutterAlarmClock.createAlarm(hour, minutes);
                },
              ),
            ),
            ElevatedButton(
              onPressed: () {

                // show alarm
                FlutterAlarmClock.showAlarms();
              },
              child: const Text(
                'Show Alarms',
                style: TextStyle(fontSize: 20.0),
              ),
            ),
            Container(
              margin: const EdgeInsets.all(25),
              child: TextButton(
                  child: const Text(
                    'Create timer',
                    style: TextStyle(fontSize: 20.0),
                  ),
                  onPressed: () {
                    int minutes;
                    minutes = int.parse(minuteController.text);

                    // create timer
                    FlutterAlarmClock.createTimer(minutes);
                    showDialog(
                        context: context,
                        builder: (context) {
                          return AboutDialog(
                            children: [
                              Center(
                                child: Text("Timer is set",
                                    style: TextStyle(
                                        fontSize: 20, fontWeight: Font-
Weight.bold)),
                              )
                            ],
                          );
                        });
                  }),
            ),
            ElevatedButton(
              onPressed: () {

                // show timers
                FlutterAlarmClock.showTimers();
```



```
              },
              child: Text(
                "Show Timers",
                style: TextStyle(fontSize: 17),
              ),
            )
          ])),
    );
  }
}
```

OUTPUT:





RESULT:

Thus Android Application that creates Alarm Clock is developed and executed successfully.

Expt. No: 12

12)Develop a simple gaming application with multimedia support.

Aim:

To develop a simple gaming application with multimedia support

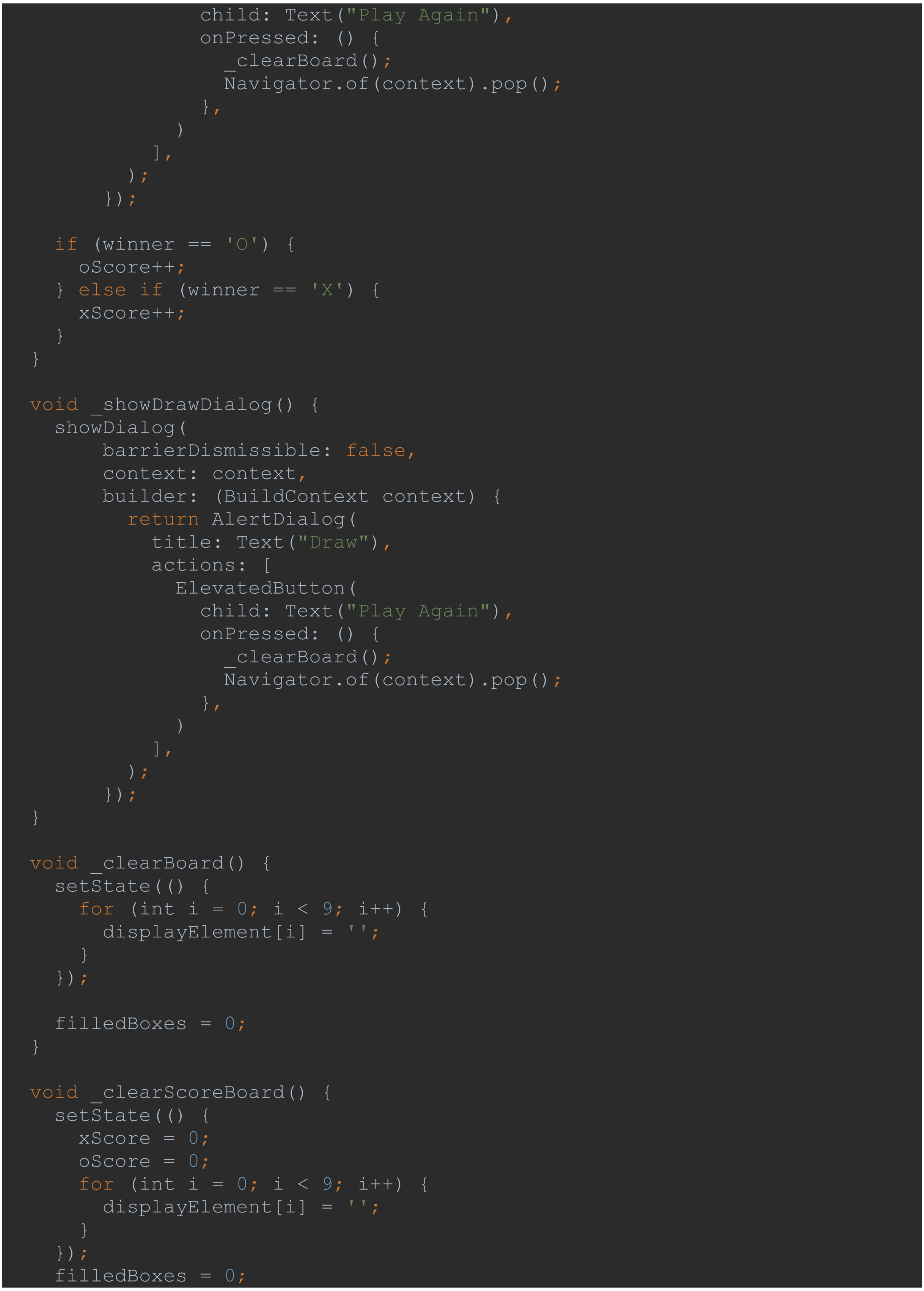
Procedure:

-Import the following:
import 'package:flutter/material.dart';
-Create a function _checkwinner that checks for the winner in the game
- Create a function _showwindialog that shows the dialog box when a user wins or draws the match.
-Create a function _clearboard that clears the board once the game is over.

Code:

Main.dart:

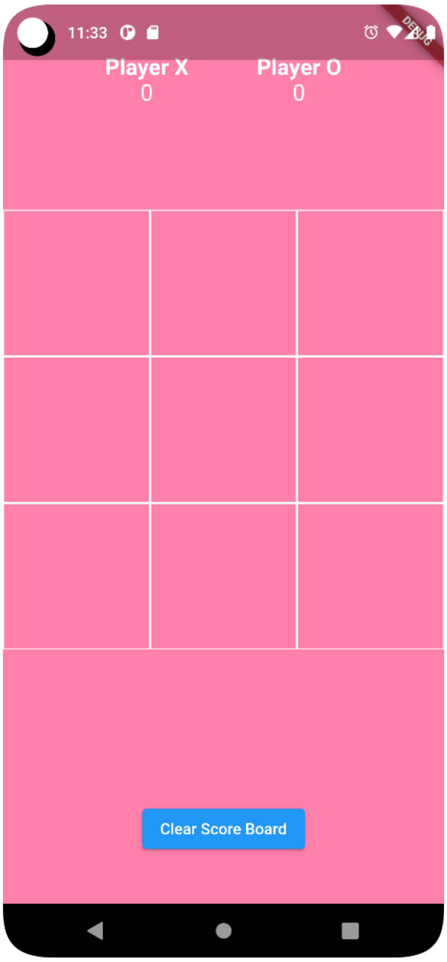
```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  bool oTurn = true;

// 1st player is O
  List<String> displayElement = ['', '', '', '', '', '', '', '', ''];
  int oScore = 0;
  int xScore = 0;
  int filledBoxes = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.pinkAccent[100],
      body: Column(
        children: <Widget>[
          Expanded(
            child: Container(
              child: Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
                  Padding(
                    padding: const EdgeInsets.all(30.0),
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: <Widget>[
                        Text(
                          'Player X',
                          style: TextStyle(fontSize: 20,
                              fontWeight: FontWeight.bold,
                              color: Colors.white),
                        ),
                        Text(
                          xScore.toString(),
                          style: TextStyle(fontSize: 20,color: Colors.white),
                        ),
                      ],
                    ),
                  ),
                  Padding(
                    padding: const EdgeInsets.all(30.0),
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: <Widget>[
                        Text('Player O', style: TextStyle(fontSize: 20,
                            fontWeight: FontWeight.bold,
                            color: Colors.white)
                        ),
                        Text(
                          oScore.toString(),
                          style: TextStyle(fontSize: 20,color: Colors.white),
                        ),
                      ],
                    ),
                  ),
                ],
              ),
            ),
          ),
          Expanded(
            flex: 4,
            child: GridView.builder(
                itemCount: 9,
                gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                    crossAxisCount: 3),
                itemBuilder: (BuildContext context, int index) {
                  return GestureDetector(
                    onTap: () {
                      _tapped(index);
                    },
                    child: Container(
                      decoration: BoxDecoration(
                          border: Border.all(color: Colors.white)),
                      child: Center(
                        child: Text(
                          displayElement[index],
                          style: TextStyle(color: Colors.white, fontSize:
35),
                        ),
                      ),
                    ),
                  );
                }),
          ),
          Expanded(
              child: Container(
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    ElevatedButton(
                      onPressed: _clearScoreBoard,
                      child: Text("Clear Score Board"),
                    ),
                  ],
                ),
              ))
        ],
      ),
    );
  }

  void _tapped(int index) {
    setState(() {
      if (oTurn && displayElement[index] == '') {
        displayElement[index] = 'O';
        filledBoxes++;
      } else if (!oTurn && displayElement[index] == '') {
        displayElement[index] = 'X';
        filledBoxes++;
      }

      oTurn = !oTurn;
      _checkWinner();
    });
  }

  void _checkWinner() {

    // Checking rows
    if (displayElement[0] == displayElement[1] &&
        displayElement[0] == displayElement[2] &&
        displayElement[0] != '') {
      _showWinDialog(displayElement[0]);
    }
    if (displayElement[3] == displayElement[4] &&
        displayElement[3] == displayElement[5] &&
        displayElement[3] != '') {
      _showWinDialog(displayElement[3]);
    }
    if (displayElement[6] == displayElement[7] &&
        displayElement[6] == displayElement[8] &&
        displayElement[6] != '') {
      _showWinDialog(displayElement[6]);
    }

    // Checking Column
    if (displayElement[0] == displayElement[3] &&
        displayElement[0] == displayElement[6] &&
        displayElement[0] != '') {
      _showWinDialog(displayElement[0]);
    }
    if (displayElement[1] == displayElement[4] &&
        displayElement[1] == displayElement[7] &&
        displayElement[1] != '') {
      _showWinDialog(displayElement[1]);
    }
    if (displayElement[2] == displayElement[5] &&
        displayElement[2] == displayElement[8] &&
        displayElement[2] != '') {
      _showWinDialog(displayElement[2]);
    }

    // Checking Diagonal
    if (displayElement[0] == displayElement[4] &&
        displayElement[0] == displayElement[8] &&
        displayElement[0] != '') {
      _showWinDialog(displayElement[0]);
    }
    if (displayElement[2] == displayElement[4] &&
        displayElement[2] == displayElement[6] &&
        displayElement[2] != '') {
      _showWinDialog(displayElement[2]);
    } else if (filledBoxes == 9) {
      _showDrawDialog();
    }
  }

  void _showWinDialog(String winner) {
    showDialog(
        barrierDismissible: false,
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text("\" " + winner + " \" is Winner!!!"),
            actions: [
              ElevatedButton(
```

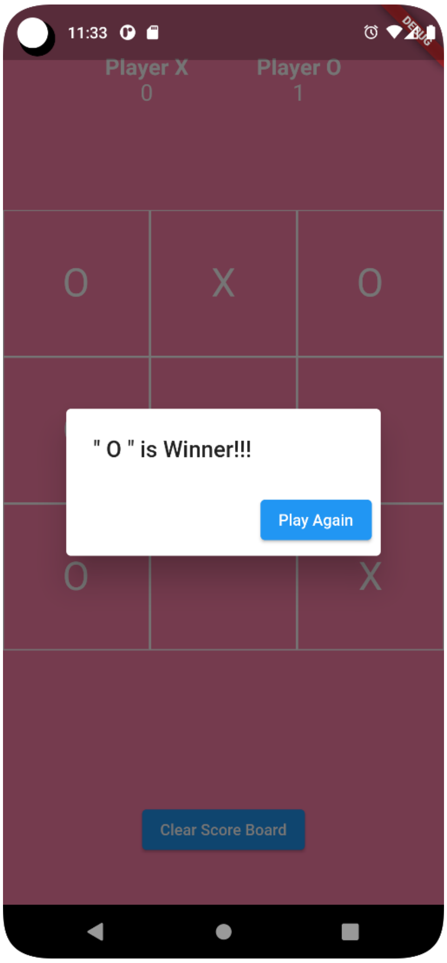


```
              child: Text("Play Again"),
              onPressed: () {
                _clearBoard();
                Navigator.of(context).pop();
              },
            )
          ],
        );
      });

  if (winner == 'O') {
    oScore++;
  } else if (winner == 'X') {
    xScore++;
  }
}

void _showDrawDialog() {
  showDialog(
      barrierDismissible: false,
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text("Draw"),
          actions: [
            ElevatedButton(
              child: Text("Play Again"),
              onPressed: () {
                _clearBoard();
                Navigator.of(context).pop();
              },
            )
          ],
        );
      });
}

void _clearBoard() {
  setState(() {
    for (int i = 0; i < 9; i++) {
      displayElement[i] = '';
    }
  });

  filledBoxes = 0;
}

void _clearScoreBoard() {
  setState(() {
    xScore = 0;
    oScore = 0;
    for (int i = 0; i < 9; i++) {
      displayElement[i] = '';
    }
  });
  filledBoxes = 0;
```

```
    }
}
```

SAMPLE I/O :





" O " is Winner!!!

RESULT:

Thus a simple gaming application with multimedia support has been build and executed successfully .

Expt. No: 13

13)Write a mobile application for data handling and connectivity via SOAP or REST to backend services potentially hosted in a cloud environment

Aim:

To develop a mobile application for data handling and connectivity via SOAP or REST to backend services potentially hosted in a cloud environment.

Procedure:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “My Application″ and click Next.
- •Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.It will take some time to build and load the project.
- Click on app -> res -> layout -> activity_main.xml.
- Now click on Text as shown below.Then delete the code which is thereand type the code as given below. • Click on app -> manifests -> AndroidManifest.xml.
- Now include the INTERNET permissions in the AndroidManifest.xml file.
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Then delete the code which is there and type the code as given below.

.

Code:

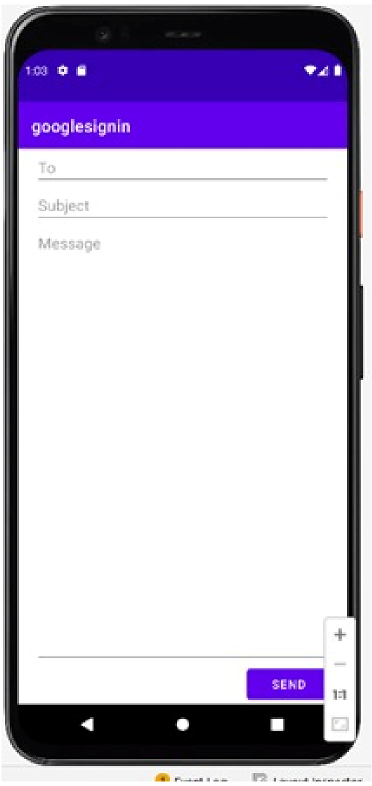
MainActivity.java:

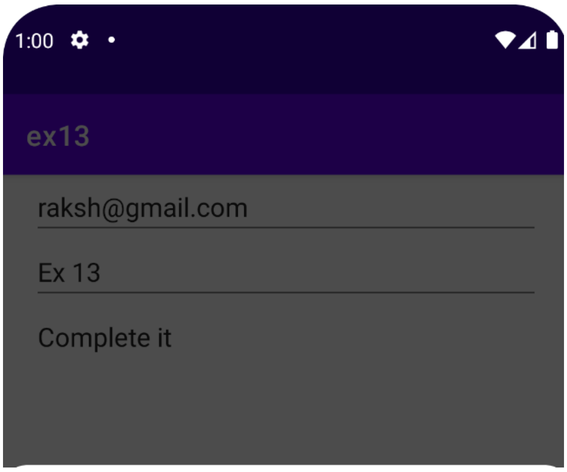
```
package com.example.ex13;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity; public class
MainActivity extends AppCompatActivity {
    private EditText eTo; private
    EditText eSubject;private
    EditText eMsg; private Button
            btn; @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); eTo =
                (EditText)findViewById(R.id.txtTo);
        eSubject = (EditText)findViewById(R.id.txtSub);eMsg =
                (EditText)findViewById(R.id.txtMsg); btn =
                (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(Intent.ACTION_SEND); it.putExtra(Intent.EXTRA_EMAIL, new
                        String[]{eTo.getText().toString()});
                it.putExtra(Intent.EXTRA_SUBJECT,eSubject.getText().toString());
                it.putExtra(Intent.EXTRA_TEXT,eMsg.getText()); it.setType("message/rfc822");
                startActivity(Intent.createChooser(it,"ChooseMail App"));
            }});
    }
}
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
    <EditText android:id="@+id/txtTo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To"/>
    <EditText android:id="@+id/txtSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"/>
    <EditText
        android:id="@+id/txtMsg"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="Message"/>
    <Button android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Send"
        android:id="@+id/btnSend"/>
</LinearLayout>
```

SAMPLE I/O :





raksh@gmail.com

Ex 13

Complete it

RESULT:

Thus an application has been build and executed successfully .

Expt. No: 14

14) Write a mobile application that will take advantage of underlying phone functionality including GEO positioning, accelerometer, and rich gesture-based UI handling.

Aim:

To develop an Android Application that uses GPS location information, accelerometer, and rich gesture-based UI handling.

Procedure:

GPS Location:

GPS coordinates are a unique identifier of a precise geographic location on the earth, Coordinates, in this context, are points of intersection in a grid system. GPS coordinates are usually expressed as the combination of latitude and longitude.

- Install the flutter_sensors and the location dependencies.
- Import the following:

import 'package:flutter_sensors/flutter_sensors.dart';
import 'package:location/location.dart';
-Ask for the permission to retrieve the location using location.requestPermission()
-Then get the location using the following method : location.getLocation()

Accelerometer:

- Install the sensors package.
- Import it using, ‘import 'package:sensors/sensors.dart';’
- accelerometer readings tell if the device is moving in a particular direction.

Gesture-based UI:

- In the onTap() property of the GestureDetector(), pass the function to be performed.
- In this case, it reverses the boolean value isLightsOn.
- This is used to switch the theme of the screen as dark or light.
- The child property of GestureDetector() is used to specify icon, on clicking which the action is to be performed.

Code:

1)Main.dart:

```
import 'package:flutter_sensors/flutter_sensors.dart';
import 'package:flutter/material.dart';
import 'dart:math';
import 'package:location/location.dart';

void main() {


  return runApp(const location_wid());
}
class location_wid extends StatefulWidget {


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

  @override
  _location_widState createState() => _location_widState();
}

class _location_widState extends State<location_wid> {
  Location location = Location();
  bool _isServiceEnabled = false;
  PermissionStatus _permissionGranted = PermissionStatus.denied;
  LocationData _locationData = LocationData.fromMap({});

  @override
  void initState() {

    super.initState();
  }
  @override
  Widget build(BuildContext context) {

    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title : Text("Lab Geo")),
        body: Center(child : Column(mainAxisAlignment : MainAxisAlignment.center,children : [
          Icon(Icons.location_on, size: 50,),
          Text('\nLatitude: ' + _locationData.latitude.toString() +
'\n\nLongitude:'  + _locationData.longitude.toString() + '\n\nAltitude: ' +
_locationData.altitude.toString() + '\n', style: TextStyle(fontSize: 20,
color: Colors.blue))
          ,ElevatedButton(onPressed: ()async{
            _isServiceEnabled = await location.serviceEnabled();
            if(!_isServiceEnabled)
              _isServiceEnabled = await location.requestService();
            print(_isServiceEnabled);
            PermissionStatus permission = await location.hasPermission();
            if(permission==PermissionStatus.denied)
              permission = await location.requestPermission();
            print(permission==PermissionStatus.granted);
            _locationData = await location.getLocation();
            // print(_locationData.latitude);
            setState(() {
            });
          },child:  Text('Get Location'),
          ),

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

2)main.dart:

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


void main() {
  runApp(
    HomeApp(),
  );
}
class HomeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: FirebaseMessagingDemo(),
    );
  }
}


/*
void main() {
  runApp(
    ChangeNotifierProvider<AppStateNotifier>(
      builder: (context) => AppStateNotifier(),
      child: HomeApp(),
    ),
  );
}


class HomeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Consumer<AppStateNotifier>(
      builder: (context, appState, child) {
        return MaterialApp(
          title: 'Flutter Tutorials',
          debugShowCheckedModeBanner: false,
          theme: AppTheme.lightTheme,
          darkTheme: AppTheme.darkTheme,
          themeMode: appState.isDarkModeOn ? ThemeMode.dark :
ThemeMode.light,
          home: ThemeDemo(),
        );
      },
    );
  }
}

*/
```

```
/*
// Wrap main widget inside the StreamProvider
class HomeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return StreamProvider<ConnectivityResult>(
      builder: (context) =>
ConnectivityService().connectionStatusController,
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'Flutter Tutorials',
        home: new ConnectivityDemo(),
      ),
    );
  }
}
*/

/*
class HomeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      //home: FirstScreen(),
      routes: {
        FirstScreen.routeId: (context) => FirstScreen(),
        SecondScreen.routeId: (context) => SecondScreen(),
      },
      initialRoute: FirstScreen.routeId,
    );
  }
}
*/
```

AppstateNotifier.dart:

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

class AppStateNotifier extends ChangeNotifier {
  //
  bool isDarkModeOn = false;

  void updateTheme(bool isDarkModeOn) {
    this.isDarkModeOn = isDarkModeOn;
    notifyListeners();
  }
}
```

AppTheme.dart:

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

class AppTheme {
  //
  AppTheme._();

  static final ThemeData lightTheme = ThemeData(
    scaffoldBackgroundColor: Colors.teal,
    appBarTheme: AppBarTheme(
      color: Colors.teal,
      iconTheme: IconThemeData(
        color: Colors.white,
      ),
    ),
    cardTheme: CardTheme(
      color: Colors.teal,
    ),
    iconTheme: IconThemeData(
      color: Colors.white54,
    ),
    textTheme: TextTheme(
      title: TextStyle(
        color: Colors.white,
        fontSize: 20.0,
      ),
      subtitle: TextStyle(
        color: Colors.white70,
        fontSize: 18.0,
      ),
    ),
  );

  static final ThemeData darkTheme = ThemeData(
    scaffoldBackgroundColor: Colors.black,
    appBarTheme: AppBarTheme(
      color: Colors.black,
      iconTheme: IconThemeData(
        color: Colors.white,
      ),
    ),
    cardTheme: CardTheme(
      color: Colors.black,
    ),
    iconTheme: IconThemeData(
      color: Colors.white54,
    ),
    textTheme: TextTheme(
      title: TextStyle(
        color: Colors.white,
        fontSize: 20.0,
      ),
      subtitle: TextStyle(
        color: Colors.white70,
        fontSize: 18.0,
      ),
    ),
  );
}
```

ThemeDemo.dart:

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

class ThemeDemo extends StatefulWidget {
  @override
  State<StatefulWidget> createState() => ThemeDemoState();
}

class ThemeDemoState extends State<ThemeDemo> {
  //
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        elevation: 0,
        title: Text('Flutter Themes'),
        leading: Icon(Icons.menu),
        actions: <Widget>[
          Switch(
            value: Provider.of<AppStateNotifier>(context).isDarkModeOn,
            onChanged: (boolVal) {
              Provider.of<AppStateNotifier>(context).updateTheme(boolVal);
            },
          )
        ],
      ),
      body: Container(
        child: ListView.builder(
          itemCount: 10,
          itemBuilder: (context, pos) {
            return Card(
              elevation: 0,
              child: ListTile(
                title: Text(
                  'Title $pos',
                  style: Theme.of(context).textTheme.title,
                ),
                subtitle: Text(
                  'Subtitle $pos',
                  style: Theme.of(context).textTheme.subtitle,
                ),
                leading: Icon(
                  Icons.alarm,
                  color: Theme.of(context).iconTheme.color,
                ),
                trailing: Icon(
                  Icons.chevron_right,
                  color: Theme.of(context).iconTheme.color,
                ),
              ),
            );
          },
        ),
      ),
    );
  }
}
```

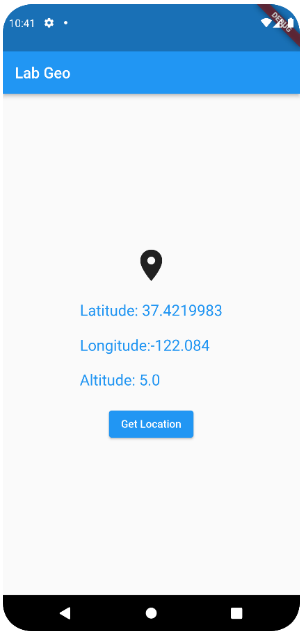
3)activity_main.xml

3)
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="213dp"
        android:layout_height="324dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="92dp"
        android:layout_marginTop="114dp"
        android:rotationX="25"
        android:text="TextView" />

</LinearLayout>
```

Main_activity.java

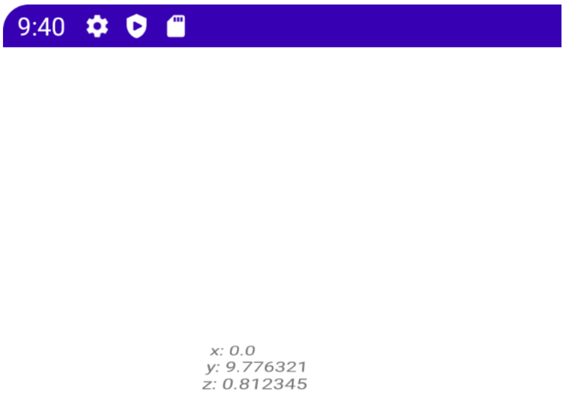
```
package com.example.sen;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;
import android.hardware.SensorEvent;
import android.hardware.Sensor;
import java.util.List;
public class MainActivity extends Activity {
    SensorManager sm = null;
    TextView textView1 = null;
    List list;

    SensorEventListener sel = new SensorEventListener(){
        public void onAccuracyChanged(Sensor sensor, int accuracy) {}
        public void onSensorChanged(SensorEvent event) {
            float[] values = event.values;
            textView1.setText("x: "+values[0]+"\ny: "+values[1]+"\nz: "+values[2]);
        }
    };

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        /* Get a SensorManager instance */
        sm = (SensorManager)getSystemService(SENSOR_SERVICE);

        textView1 = (TextView)findViewById(R.id.textView1);

        list = sm.getSensorList(Sensor.TYPE_ACCELEROMETER);
        if(list.size()>0){
            sm.registerListener(sel, (Sensor) list.get(0), SensorManager.SENSOR_DELAY_NORMAL);
        }else{
            Toast.makeText(getBaseContext(), "Error: No Accelerometer.",
Toast.LENGTH_LONG).show();
        }
    }

    @Override
    protected void onStop() {
        if(list.size()>0){
            sm.unregisterListener(sel);
        }
        super.onStop();
    }
}
```

Sample I/O:

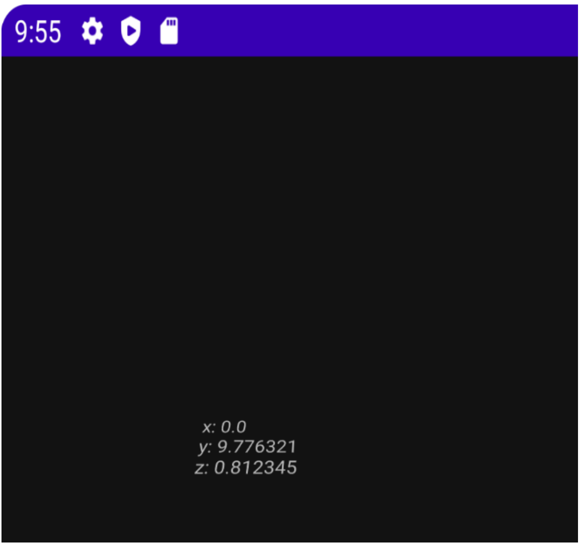


Lab Geo

```
Latitude: 37.4219983
Longitude:-122.084
Altitude: 5.0
```



```
x: 0.0
y: 9.776321
z: 0.812345
```



```
x: 0.0
y: 9.776321
z: 0.812345
```

Result:

Thus, an Android Application that uses GPS location was successfully implemented.

Expt. No: 15

15) Write an application for integrating mobile applications in the market, including social networking software integration .

Aim:

To develop a mobile application for_integrating mobile applications in the market, including social networking software integration.

Procedure:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “My Application″ and click Next.
- •Then select the Minimum SDK as shown below and click Next.
- Then select the Empty Activity and click Next.
- Finally click Finish.It will take some time to build and load the project.
- Click on app -> res -> layout -> activity_main.xml.
- Now click on Text as shown below.Then delete the code which is thereand type the code as given below. • Click on app -> manifests -> AndroidManifest.xml.
- Now include the INTERNET permissions in the AndroidManifest.xml file.
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Then delete the code which is there and type the code as given below.

.

Code:

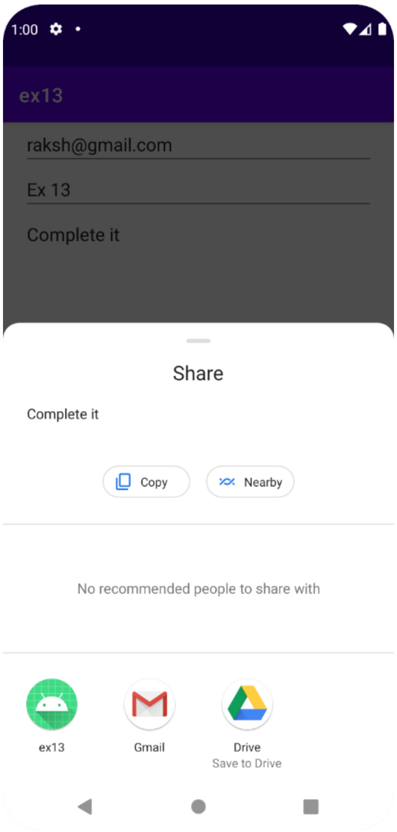
MainActivity.java:

```
package com.example.ex13;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity; public class
MainActivity extends AppCompatActivity {
    private EditText eTo; private
    EditText eSubject;private
    EditText eMsg; private Button
            btn; @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); eTo =
                (EditText)findViewById(R.id.txtTo);
        eSubject = (EditText)findViewById(R.id.txtSub);eMsg =
                (EditText)findViewById(R.id.txtMsg); btn =
                (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(Intent.ACTION_SEND); it.putExtra(Intent.EXTRA_EMAIL, new
                        String[]{eTo.getText().toString()});
                it.putExtra(Intent.EXTRA_SUBJECT,eSubject.getText().toString());
                it.putExtra(Intent.EXTRA_TEXT,eMsg.getText()); it.setType("message/rfc822");
                startActivity(Intent.createChooser(it,"ChooseMail App"));
            }});
    }
}
```

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
    <EditText android:id="@+id/txtTo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To"/>
    <EditText android:id="@+id/txtSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"/>
    <EditText
        android:id="@+id/txtMsg"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="Message"/>
    <Button android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Send"
        android:id="@+id/btnSend"/>
</LinearLayout>
```

SAMPLE I/O :



1:00

ex13

raksh@gmail.com

Ex 13

Complete it

Share

Complete it

No recommended people to share with

RESULT:

Thus an application has been build and executed successfully .