EX. No.1 GUI COMPONENTS, FONTS AND COLORS

Aim: To develop an application that uses GUI components, Font, and Colors.

Procedure:

GUI components:

- 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()
  - contains widgets that are used for the material design of an application.
  - theme property is used to set the theme of the application to dark or light.
  - Home property defines the starting point of the application. It usually contains Scaffold.
- Text():
  - import 'package:flutter/material.dart';
  - specify the string to be displayed, within quotes inside Text().
  - Style property can be used to add TextStyle like fontSize, color.
  - textAlign property can be used for alignment of specified text
- GridView.count()
  - creates a layout with a fixed number of tiles in the cross axis children property is used to specify the widgets to be included in the layout. (Eg:
  - containers)
  - To set spacing between items along main axis or cross axis, set the required double values for mainAxisSpacing property and crossAxisSpacing property respectively
- Container()
  - Helps to create a rectangular visual element.
  - The margin property uses EdgeInsets to set the margin for the four directions (LTRB).
  - Image or icon or text can be included placed inside the container using child parameter:
  - Decoration (BoxDecoration) can be used to give shape, backgroundColor etc. to a container.

Font:

- Style property can be used to add TextStyle like fontSize, color.
- To use google fonts,
  - Install using ‘flutter pub add google_fonts’
  - import 'package:google_fonts/google_fonts.dart';
  - Specify the font name in the style property of Text().
  - textStyle attribute can be used to format the text.

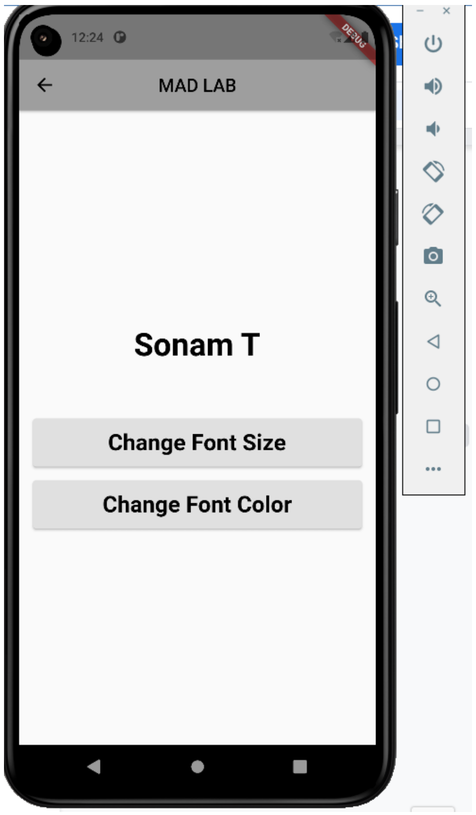
- style:GoogleFonts.rockSalt(textStyle:constTextStyle(color: Colors.black,fontSize: 20)

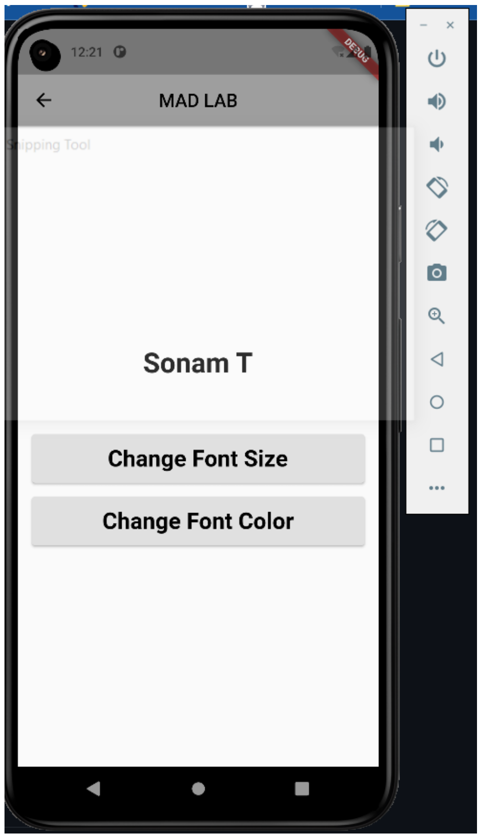
__**Colors:**__
- Color property can be used to specify the color using the Colors class.
- It can also be represented in the format of #RRGGBB where RR represents Red color, GG represents the Green color and BB represents the Blue color.

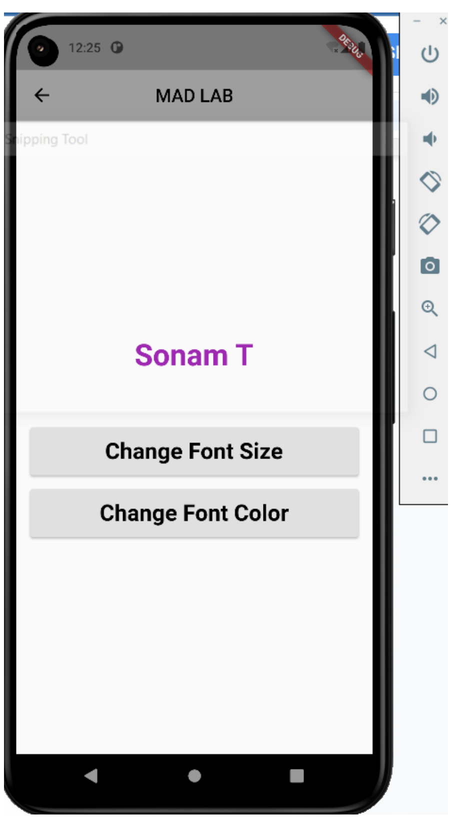
Code :

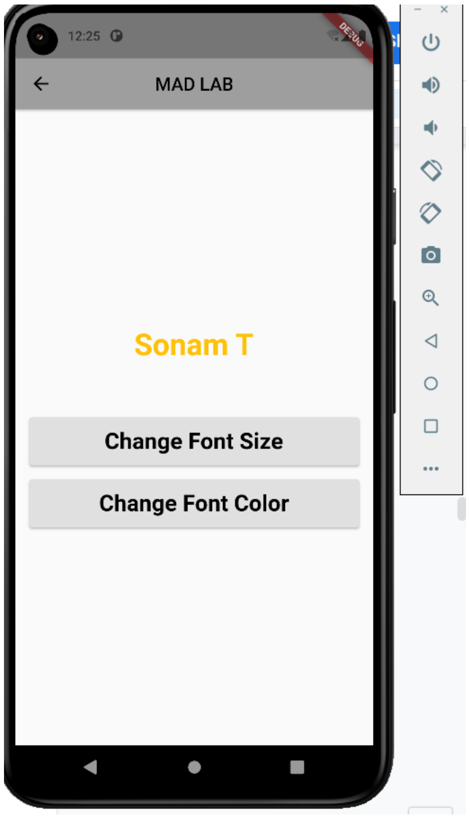
```
import 'dart:math';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
void main() {
  runApp(changefont());
}
class changefont extends StatefulWidget {
  const changefont({Key? key}) : super(key: key);
  @override
  _changefontState createState() => _changefontState();
}

class _changefontState extends State<changefont> {
  double size = 30.0;
  List<Color> colors = [
    Colors.black,
    Colors.blue,
    Colors.green,
    Colors.amber,
    Colors.red,
    Colors.purple,
    Colors.cyan
  ];
  int col = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        title: const Text(
          'MAD LAB',
          textAlign: TextAlign.center,
        ),
        elevation: 2.0,
      ),
      body: Padding(
padding: EdgeInsets.all(15.0),
child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      Text(
        'Sonam T',
        textAlign: TextAlign.center,
        style: TextStyle(
          fontSize: size,
          fontWeight: FontWeight.bold,
          color: colors[col],
        ),
      ),
      SizedBox(height: 60.0),
      ElevatedButton(
        style: ElevatedButton.styleFrom(
          primary: Colors.grey[300],
          onSurface: Colors.black,
          padding: EdgeInsets.all(12.0),
        ),
        child: Text(
          'Change Font Size',
          style: TextStyle(
            fontSize: 25.0,
            fontWeight: FontWeight.bold,
          ),
        ),
        onPressed: () {
          setState(() {
            size = Random().nextInt(8) + 30;
          });
        },
      ),
      SizedBox(height: 15.0),
      ElevatedButton(
        style: ElevatedButton.styleFrom(
          primary: Colors.grey[300],
          onSurface: Colors.black,
          padding: EdgeInsets.all(12.0),
        ),
        child: Text(
          'Change Font Color',
          style: TextStyle(
            fontSize: 25.0,
            fontWeight: FontWeight.bold,
                ),
              ),
              onPressed: () {
                setState(() {
                  col = Random().nextInt(7);
                });
              },      ),
          ]),
      ), );
  }}
```

Output:









Result: The application has been developed successfully and output verified

EX. No.2 LAYOUT MANAGERS AND EVENT LISTENERS

Aim: To develop an application that uses Layout Managers and event listeners.

PROCEDURE:

<u>- Layout managers:</u>

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

<u>- Event listeners:</u>

- onPressed() property is used to assign a callback function to the button or icon.
- The application executes this function whenever the user presses taps the chip.

- If onPressed() is null, then it denotes disabled.

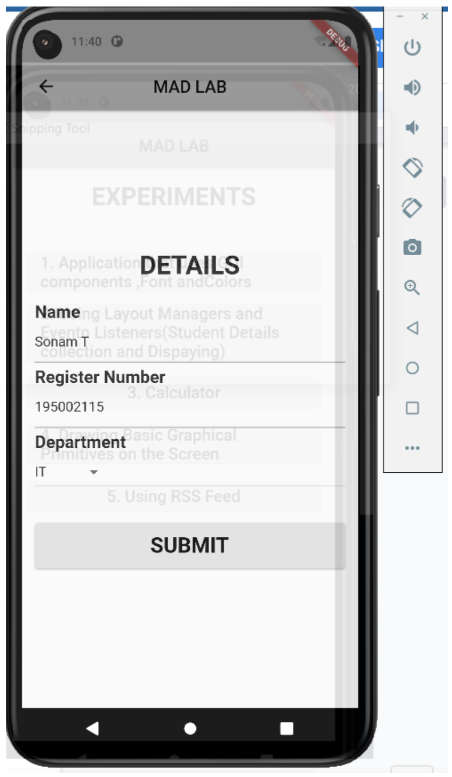
Code :

Details.dart:

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
void main() {
  runApp(details());
}
class details extends StatefulWidget {
  const details({Key? key}) : super(key: key);
  @override
  _detailsState createState() => _detailsState();
}
class _detailsState extends State<details> {
  int submitted = 0;
  String name = '';
  String reg = '';
  String dept = "SELECT";
  var items = [
    "SELECT",
    "IT",
    "CSE",
    "BME",
    "ECE",
    "EEE",
    "CHEM",
    "CIVIL",
    "MECH"
  ];
  @override
  Widget build(BuildContext context) {
    if (submitted == 0) {
      return Scaffold(
        appBar: AppBar(
          centerTitle: true,
          title: const Text(
            'MAD LAB',
            textAlign: TextAlign.center,
          ),
          elevation: 2.0,
        ),
        body: Padding(
          padding: EdgeInsets.all(15.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              Text(
                'DETAILS',
                textAlign: TextAlign.center,
                style: TextStyle(
                  fontSize: 30.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              SizedBox(height: 25.0),
              Text(
                'Name',
                style: TextStyle(
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              TextField(
                decoration: InputDecoration(
                  hintText: 'Enter your name',
                ),
                onChanged: (String? newValue) {
                  setState(() {
```

```
      name = newValue!;
    });
  },
),
SizedBox(height: 5.0),
Text(
  'Register Number',
  style: TextStyle(
    fontSize: 20.0,
    fontWeight: FontWeight.bold,
  ),  ),
TextField(
  keyboardType: TextInputType.number,
  inputFormatters: [FilteringTextInputFormatter.digitsOnly],
  decoration: InputDecoration(
    hintText: 'Enter your register number',
  ),
  onChanged: (String? newValue) {
    setState(() {
      reg = newValue!;
    });
  },
),
SizedBox(height: 5.0),
Text(
  'Department',
  style: TextStyle(
    fontSize: 20.0,
    fontWeight: FontWeight.bold,
  ),
),
DropdownButton(
  value: dept,
  items: items.map((String items) {
    return DropdownMenuItem(value: items, child: Text(items));
  }).toList(),
  onChanged: (String? newValue) {
    setState(() {
      dept = newValue!;
    });
  },   ),
SizedBox(height: 35.0),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    primary: Colors.grey[300],
    onSurface: Colors.black,
           padding: EdgeInsets.all(12.0),
          ),
          child: Text(
            'SUBMIT',
            style: TextStyle(
              fontSize: 25.0,
              fontWeight: FontWeight.bold,
            ),
          ),
          onPressed: () {
            setState(() {
              submitted = 1;
            });
          },
        ),
      ], ),),
);} else {
return Scaffold(
  body: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      Text(
        'DETAILS',
        textAlign: TextAlign.center,
        style: TextStyle(
          fontSize: 30.0,
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(height: 45.0),
      Text(
        ' Name : ' + '$name',
        style: TextStyle(
          fontSize: 20.0,
          fontWeight: FontWeight.bold,
        ), ),
      SizedBox(height: 15.0),
      Text(
        ' Register Number : ' + '$reg',
        style: TextStyle(
          fontSize: 20.0,
          fontWeight: FontWeight.bold,
        ),
      ),
      SizedBox(height: 15.0),
            Text(
              ' Department : ' + '$dept',
              style: TextStyle(
                fontSize: 20.0,
                fontWeight: FontWeight.bold,
              ),
            ),
            SizedBox(height: 35.0),
            ElevatedButton(
              style: ElevatedButton.styleFrom(
                primary: Colors.grey[300],
                onSurface: Colors.black,
                padding: EdgeInsets.all(12.0),
              ),
              child: Text(
                'BACK',
                style: TextStyle(
                  fontSize: 25.0,
                  fontWeight: FontWeight.bold,
                ),
              ),
              onPressed: () {
                setState(() {
                  dept = "SELECT";
                  submitted = 0;
                });
              },
            ),
          ],
        ),
      );
    }}}
```

Output:

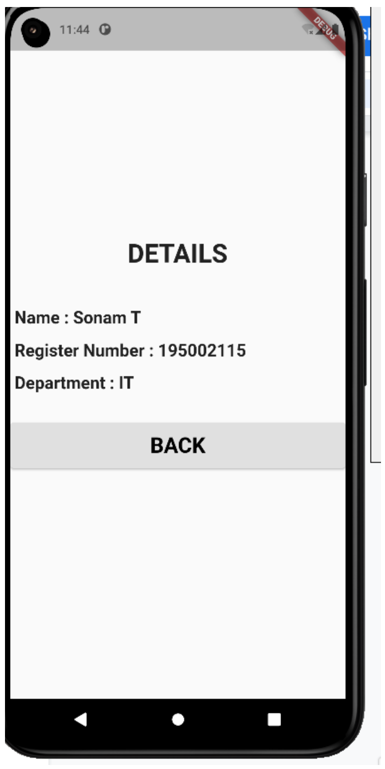

MAD LAB

DETAILS

Name
Sonam T

Register Number
195002115

Department
IT


DETAILS

Name : Sonam T

Register Number : 195002115

Department : IT

Result: Thus, an application that uses layout managers and event listeners has been implemented using Flutter.

EX. No.3 **SIMPLE CALCULATOR**

Aim: To develop a naive calculator application.

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';
import 'homePage.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Simple Calculator',
      debugShowCheckedModeBanner: false,
      theme: ThemeData.light(),
      home: HomePage(),
    );
  }
}
```

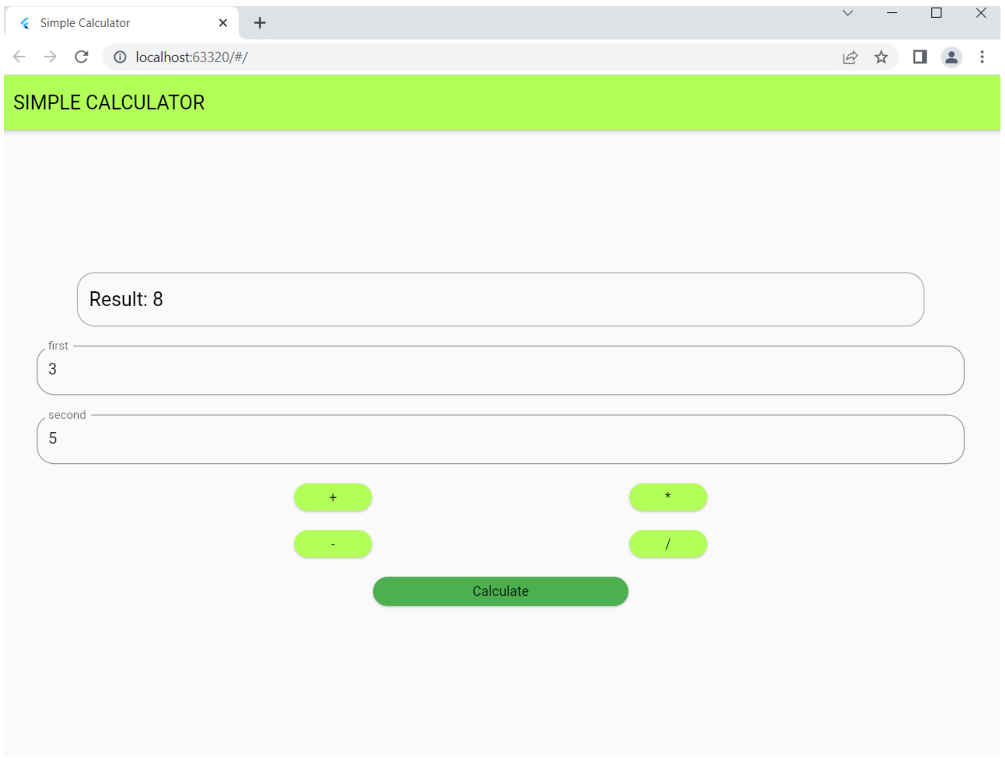
homepage.dart

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

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

```
}

class _HomePageState extends State<HomePage> {
  double num1 = 0, num2 = 0, res = 0;
  TextEditingController t1 = TextEditingController(text: '');

  TextEditingController t2 = TextEditingController(text: '');
  doAddition() {
    setState(() {
      num2 = double.parse(t2.text);
      num1 = double.parse(t1.text);
      res = num1 + num2;
    });
  }

  doSub() {
    setState(() {
      num2 = double.parse(t2.text);
      num1 = double.parse(t1.text);
      res = num1 - num2;
    });
  }

  doMul() {
    setState(() {
      num2 = double.parse(t2.text);
      num1 = double.parse(t1.text);
      res = num1 * num2;
    });
  }

  doDiv() {
    setState(() {
      num2 = double.parse(t2.text);
      num1 = double.parse(t1.text);
      res = (num1 / num2);
    });
  }

  doClear() {
    setState(() {
      t1 = TextEditingController(text: '');
      t2 = TextEditingController(text: '');
      res = 0;
    });
  }
```

```
doDecimal() {
  setState(() {
    //TODO:.............................
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: false,
    appBar: AppBar(
      title: Text('SIMPLE CALCULATOR',
          style: TextStyle(fontSize: 20.0, color: Colors.black)),
      backgroundColor: Colors.lightGreenAccent,
    ),
    body: Container(
      padding:
          EdgeInsets.only(bottom: 40.0, top: 15.0, left: 40.0, right: 40.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 42.0),
            child: TextField(
              keyboardType: TextInputType.number,
              cursorColor: Colors.indigo,
              enabled: false,
              //enableInteractiveSelection: false,
              decoration: InputDecoration(
                fillColor: Colors.white,
                hintText: 'Result: $res',
                hintStyle: TextStyle(fontSize: 20.0, color: Colors.black),
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(18.0),
                ),
              ),
            ),
          ),
          Padding(
            padding: EdgeInsets.only(top: 20.0),
          ),
          //The Text field for the First number
          TextField(
            keyboardType: TextInputType.number,
            cursorColor: Colors.tealAccent,
  controller: t1,
  decoration: InputDecoration(
    labelText: 'first',
    fillColor: Colors.white,
    hintText: 'Enter your First number',
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(18.0),
    ),
  ),
),
Padding(
  padding: EdgeInsets.only(top: 20.0),
),
//The Text field for the second number

TextField(
  keyboardType: TextInputType.number,
  cursorColor: Colors.tealAccent,
  controller: t2,
  decoration: InputDecoration(
    labelText: 'second',
    fillColor: Colors.white,
    hintText: 'Enter your Second number',
    border: OutlineInputBorder(
      borderRadius: BorderRadius.circular(18.0),
    ),
  ),
),
Padding(
  padding: EdgeInsets.only(top: 20.0),
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    MaterialButton(
      child: Text('+'),
      shape: StadiumBorder(),
      color: Colors.lightGreenAccent,
      onPressed: () {
        //TODO:
        doAddition();
      },
    ),
    MaterialButton(
      child: Text('*'),
      shape: StadiumBorder(),
      color: Colors.lightGreenAccent,
      onPressed: () {
        //TODO:
        doMul();
      },
    ),
  ],
),
Padding(
  padding: EdgeInsets.only(top: 20.0),
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    MaterialButton(
      child: Text('-'),
      color: Colors.lightGreenAccent,
      shape: StadiumBorder(),
      onPressed: () {
        //TODO:
        doSub();
      },
    ),
    MaterialButton(
      child: Text('/'),
      shape: StadiumBorder(),
      color: Colors.lightGreenAccent,
      onPressed: () {
        //TODO:
        doDiv();
      },
    ),
  ],
),
Padding(
  padding: EdgeInsets.only(top: 20.0),
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    MaterialButton(
      child: Text(
        'Calculate',
        textAlign: TextAlign.center,
      ),
      color: Colors.green,
                    shape: StadiumBorder(),
                    padding: EdgeInsets.only(
                        left: 110.0, right: 110.0, top: 15.0, bottom: 15.0),
                    onPressed: () {
                      //TODO:
                      doClear();
                    },
                  ),
                ],
              ),
            ],
          ),
        ),
      );
    }
  }
```

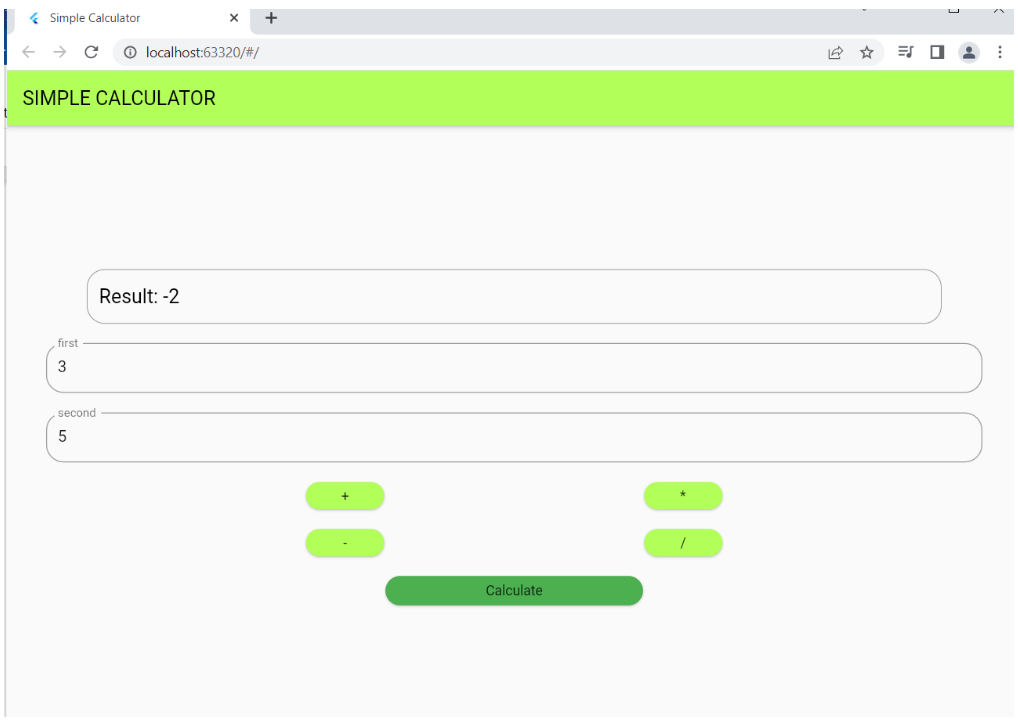
Output:

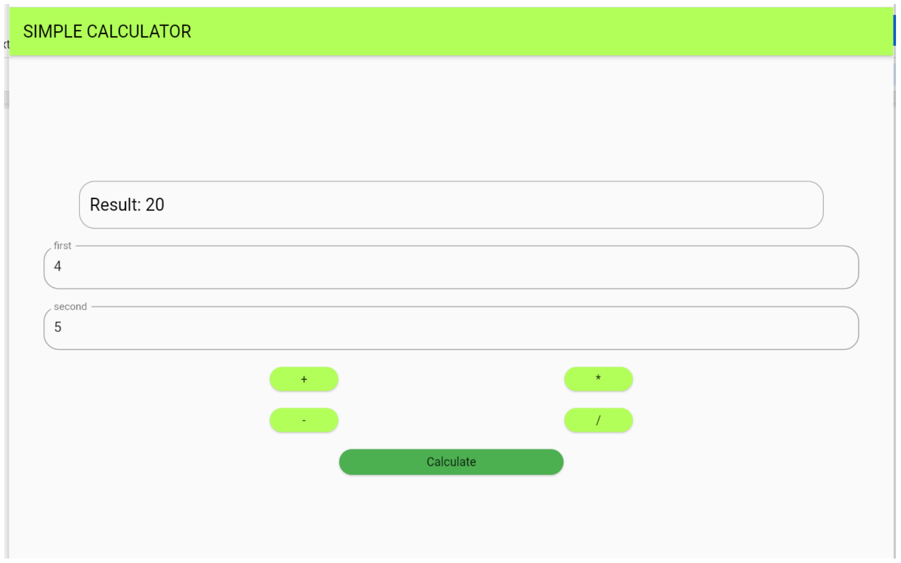


Result: 8

first
3

second
5





Result: Thus, a simple naive calculator application is developed using Flutter.

EX. No.4 **BASIC GRAPHICAL PRIMITIVES**

Aim: To write an application that draws basic graphical primitives on the screen.

PROCEDURE:

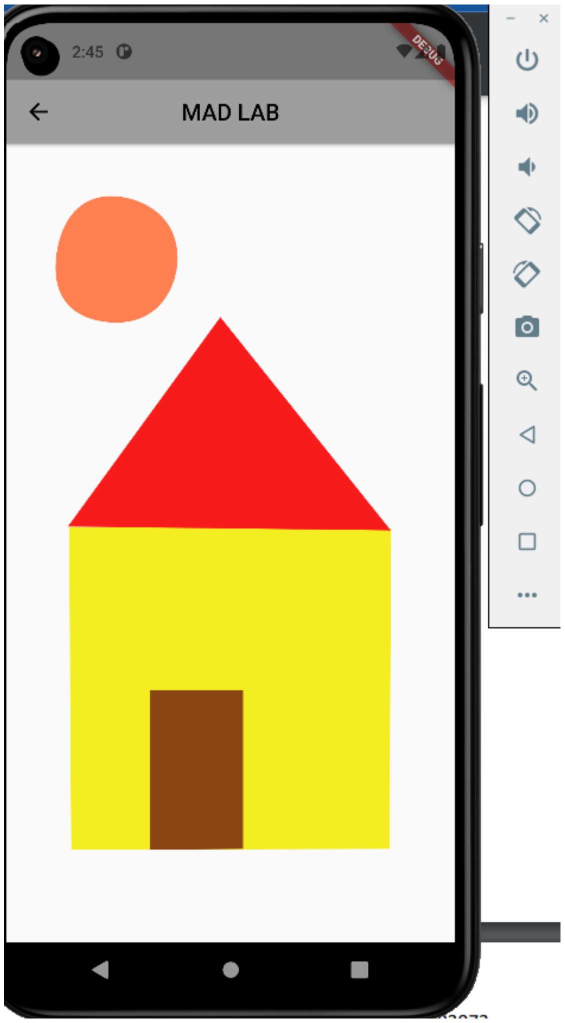
- Declare a class for each graphical primitive.
- The CustomPainter class is used.
- The paint method takes canvas and size as parameters.
- Create an instance of Paint() class.
- canvas.drawRect() is used to draw a rectangle.
- Similarly, for line drawLine() is used.
- For circle and arc, drawCircle() and drawArc() are used respectively.
- Inside the scaffold, the required class is called by specifying it as the painter of CustomPaint class.

Code :

```
import 'package:flutter/material.dart';
void main() {
  runApp(drawing());
}
class drawing extends StatelessWidget {
  const drawing({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        title: const Text(
          'MAD LAB',
          textAlign: TextAlign.center,
        ),
        elevation: 2.0,
      ),
      body: CustomPaint(
        size: Size(
            MediaQuery.of(context).size.width,
            (MediaQuery.of(context).size.width * 1.7142857142857142)
                .toDouble()),
        painter: Painter(),
      ),
    );
  }
```

```
}

class Painter extends CustomPainter {
  @override
  void paint(Canvas canvas, Size size) {
    Paint paint0 = Paint()
      ..color = const Color.fromARGB(255, 243, 237, 33)
      ..style = PaintingStyle.fill
      ..strokeWidth = 1;
    Path path0 = Path();
    path0.moveTo(size.width * 0.1400000, size.height * 0.4973333);
    path0.lineTo(size.width * 0.8577143, size.height * 0.4990000);
    path0.lineTo(size.width * 0.8542857, size.height * 0.9150000);
    path0.lineTo(size.width * 0.1445714, size.height * 0.9160000);
    path0.quadraticBezierTo(size.width * 0.1434286, size.height * 0.8133333,
        size.width * 0.1400000, size.height * 0.4973333);
    path0.close();
    canvas.drawPath(path0, paint0);
    Paint paint1 = Paint()
      ..color = const Color.fromARGB(255, 255, 127, 80)
      ..style = PaintingStyle.fill
      ..strokeWidth = 1;
    Path path1 = Path();
    path1.moveTo(size.width * 0.2342857, size.height * 0.0680000);
    path1.cubicTo(
        size.width * 0.2914286,
        size.height * 0.0680000,
        size.width * 0.3817143,
        size.height * 0.0886667,
        size.width * 0.3817143,
        size.height * 0.1486667);
    path1.cubicTo(
        size.width * 0.3817143,
        size.height * 0.1820000,
        size.width * 0.3457143,
        size.height * 0.2320000,
        size.width * 0.2457143,
        size.height * 0.2320000);
    path1.cubicTo(
        size.width * 0.1885714,
        size.height * 0.2320000,
        size.width * 0.1097143,
        size.height * 0.2190000,
        size.width * 0.1097143,
        size.height * 0.1606667);
    path1.cubicTo(
        size.width * 0.1097143,
        size.height * 0.1273333,
        size.width * 0.1342857,
        size.height * 0.0680000,
        size.width * 0.2342857,
        size.height * 0.0680000);
    path1.close();
    canvas.drawPath(path1, paint1);
    Paint paint2 = Paint()
      ..color = const Color.fromARGB(255, 248, 27, 27)
      ..style = PaintingStyle.fill
      ..strokeWidth = 1;
    Path path2 = Path();
    path2.moveTo(size.width * 0.1371429, size.height * 0.4966667);
    path2.lineTo(size.width * 0.4771429, size.height * 0.2250000);
    path2.lineTo(size.width * 0.8571429, size.height * 0.5016667);
    canvas.drawPath(path2, paint2);
    Paint paint3 = Paint()
      ..color = const Color.fromARGB(255, 139, 69, 19)
      ..style = PaintingStyle.fill
      ..strokeWidth = 1;
    Path path3 = Path();
    path3.moveTo(size.width * 0.3200000, size.height * 0.7090000);
    path3.lineTo(size.width * 0.5277143, size.height * 0.7090000);
    path3.lineTo(size.width * 0.5277143, size.height * 0.9150000);
    path3.lineTo(size.width * 0.3200000, size.height * 0.9160000);
    path3.close();
    canvas.drawPath(path3, paint3);
  }

  @override
  bool shouldRepaint(CustomPainter old) {
    return true;
  }
 }
```

Output:



Result: The application that draws basic graphical primitives on the screen has been implemented using Flutter.

EX. No.5 **DATABASE CONNECTION**

Aim: To develop an application that makes use of a database.

PROCEDURE:

- Include package for sqlite and path
-Have column for Student name and roll no
-CRUD operations carried out
-Delete and Update done in terms of roll no

Code :

Main.dart

```
// ignore_for_file: prefer_const_constructors
import 'db.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
void main() {
runApp(const MyApp());
}
class MyApp extends StatelessWidget {
const MyApp({Key? key}) : super(key: key);
@override
Widget build(BuildContext context) {
return MaterialApp(
home: Home(),
);
}
}
class Home extends StatefulWidget{
const Home({Key? key}) : super(key: key);
@override
State<StatefulWidget> createState() {
return _HomeState();
}
}
class _HomeState extends State<Home>{
getdataview(){
Future.delayed(Duration(milliseconds: 500),() async {
slist = await mydb.db.rawQuery('SELECT * FROM students ORDER BY
roll_no;');
setState(() {view=1;});
});
}
```

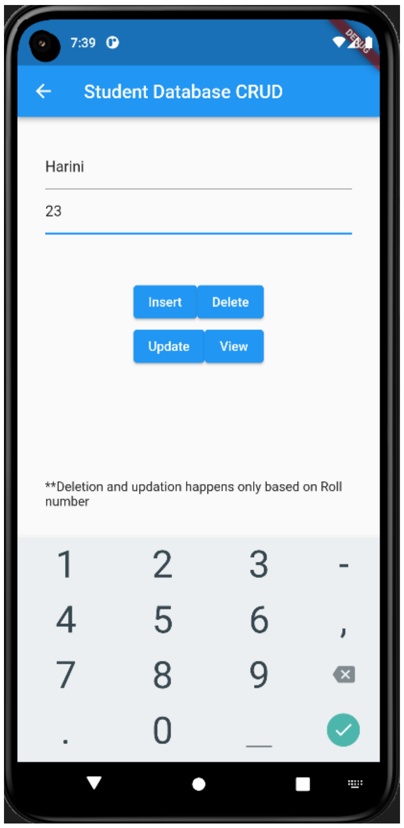
```
TextEditingController name = TextEditingController();
TextEditingController rollno = TextEditingController();
List<Map> slist = [];
MyDb mydb = MyDb();
int view=0;
@override
void initState() {
mydb.open();
super.initState();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text(
"Student Database CRUD",
textAlign: TextAlign.center,),
leading: view==1?
BackButton(
color: Colors.white,
onPressed: () {
setState(() {
view=0;
});
},
):
null
),
body:Container(
padding: EdgeInsets.all(30),
child: view==0 ?
Column(
mainAxisAlignment: MainAxisAlignment.start,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
TextField(
controller: name,
decoration: InputDecoration(
hintText: "Student Name",
),
),
TextField(
keyboardType: TextInputType.number,
inputFormatters: [
FilteringTextInputFormatter.digitsOnly
],
controller: rollno,
decoration: InputDecoration(
hintText: "Roll No.",
),
),
SizedBox(height: 50),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children :[
ElevatedButton(
onPressed: (){
Future.delayed(Duration(milliseconds: 500),
() async {
var data = await
mydb.getStudent(int.parse(rollno.text));
if(data != null){
ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("Student Already present with roll_no :
"+rollno.text)));
}else{
mydb.db.rawInsert("INSERT INTO students
(name, roll_no) VALUES (?, ?);",
[name.text, rollno.text]);
ScaffoldMessenger.of(context).showSnackBar
(SnackBar(content: Text("New Student Added")));
name.text = "";
rollno.text = "";
}
});
},
child: Text("Insert")
),
ElevatedButton(onPressed: (){
Future.delayed(Duration(milliseconds: 500), ()
async {
var data = await
mydb.getStudent(int.parse(rollno.text));
if(data == null){
ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("No student found with roll_no :
"+rollno.text)));
}else{
mydb.db.rawDelete("DELETE FROM students
where roll_no=?;",[rollno.text]);
//mydb.db.rawDelete("DELETE * FROM
students");
ScaffoldMessenger.of(context).showSnackBar
(SnackBar(content: Text("Student Successfully removed")));
name.text = "";
rollno.text = "";
}
});
}, child: Text("Delete")),
],),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children :[
ElevatedButton(onPressed: (){
Future.delayed(Duration(milliseconds: 500), ()
async {
var data = await
mydb.getStudent(int.parse(rollno.text));
if(data != null){
mydb.db.rawInsert("UPDATE students SET name
= ?, roll_no = ? WHERE roll_no = ?",
[name.text, rollno.text, rollno.text]);
ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("Student Data Updated")));
name.text = "";
rollno.text = "";
}else{
ScaffoldMessenger.of(context).showSnackBar(
SnackBar(content: Text("No student found with roll_no :
"+rollno.text)));
}
});
}, child: Text("Update")),
ElevatedButton(onPressed: (){
getdataview();
}, child: Text("View")),
],),
Expanded(child: Align(
alignment: Alignment.bottomLeft,
child: Text("**Deletion and updation happens only
based on Roll number",),
)),
],
):
SingleChildScrollView(
child: Container(
child: slist.isEmpty?Text("No students to show."):
Column(
children: slist.map((stuone){
return Card(
child: ListTile(
leading: Icon(Icons.people),
title:
Text(stuone["roll_no"].toString()),
subtitle: Text("Name:" +
stuone["name"]),
),
);
}).toList(),
),),)
));}}
```

Db.dart

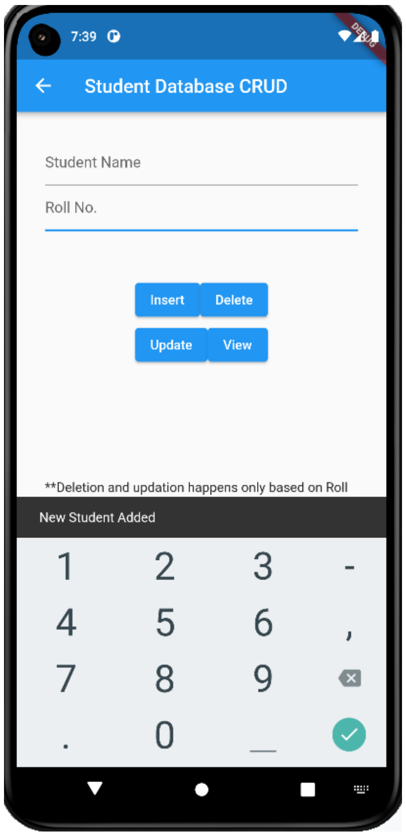
```
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';
class MyDb{
late Database db;
Future open() async {
// Get a location using getDatabasesPath
var databasesPath = await getDatabasesPath();
String path = join(databasesPath,'demo.db');
//join is from path package
//output /data/user/0/com.dbapp.flutter.dbapp/databases/demo.db
db = await openDatabase(path, version: 1,
onCreate: (Database db, int version) async {
// When creating the db, create the table
await db.execute('''
CREATE TABLE IF NOT EXISTS students(
id primary key,
name varchar(255) not null,
roll_no int unique not null
);
//create more table here
''');
//table students will be created if there is no table
'students'
});
}
Future<Map<dynamic, dynamic>?> getStudent(int rollno) async {
List<Map> maps = await db.query('students',
where: 'roll_no = ?',
whereArgs: [rollno]);
//getting student data with roll no.
if (maps.isNotEmpty) {
```

```
return maps.first;
}
return null;
}
}
```

Output:

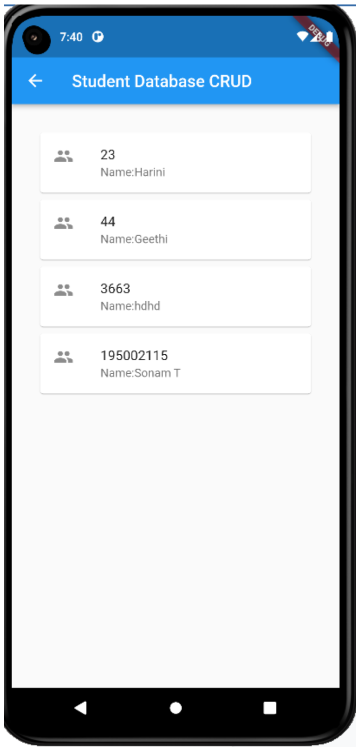


**Deletion and updation happens only based on Roll number



**Deletion and updation happens only based on Roll

New Student Added


7:40

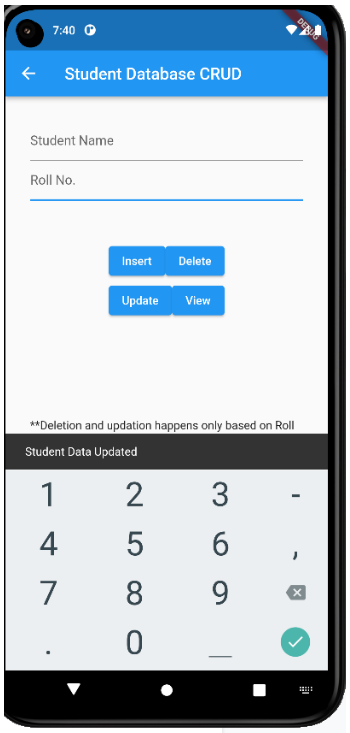
Student Database CRUD

23
Name:Harini

44
Name:Geethi

3663
Name:hdhd

195002115
Name:Sonam T


7:40

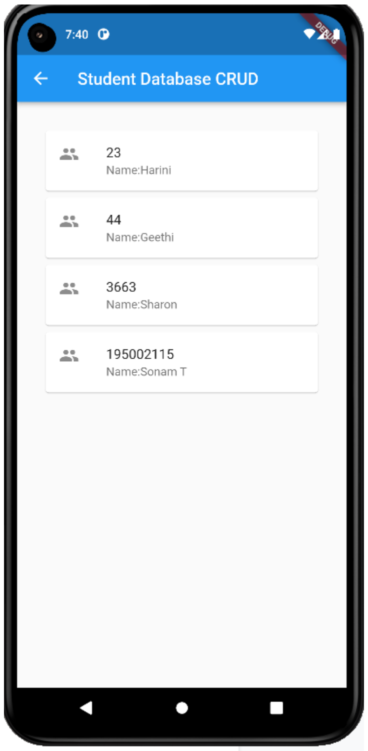
Student Database CRUD

Student Name

Roll No.

**Deletion and updation happens only based on Roll

Student Data Updated


7:40

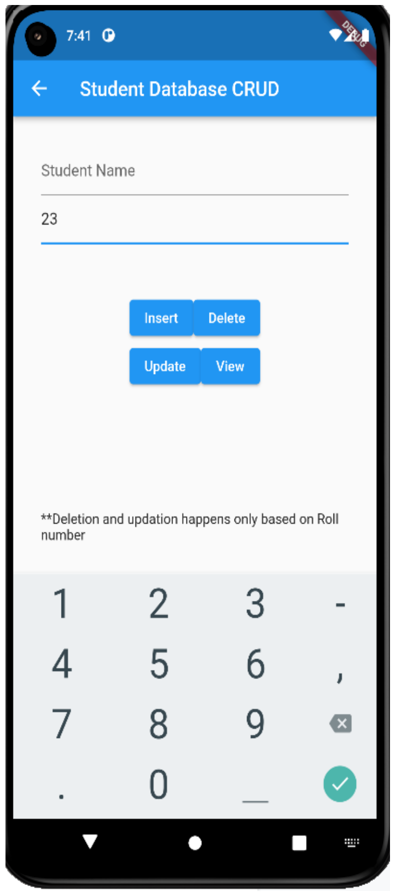
Student Database CRUD

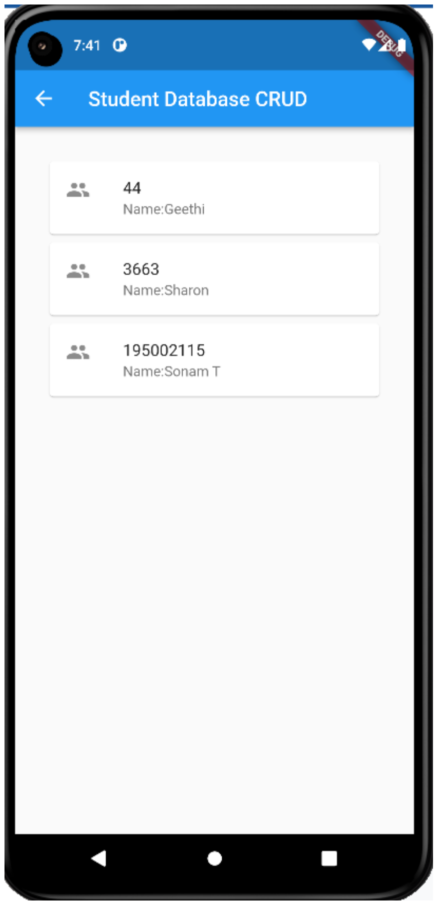
23
Name:Harini

44
Name:Geethi

3663
Name:Sharon

195002115
Name:Sonam T





Result: The application has been developed successfully and output verified

EX. No.6 **RSS FEED**

Aim: To develop an application that makes use of RSS Feed.

PROCEDURE:

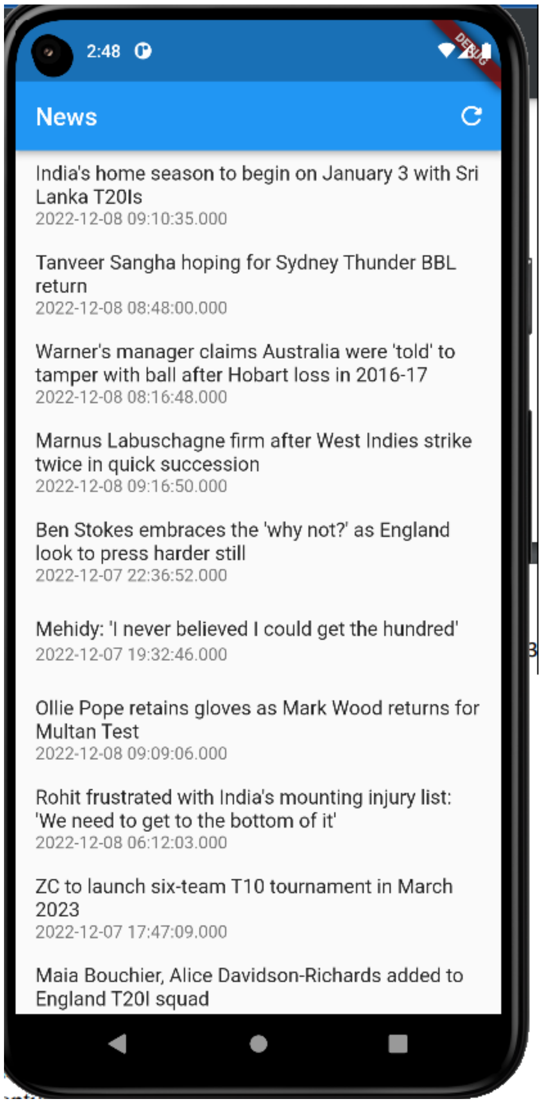
- Import packages.
import 'package:webfeed/webfeed.dart';
import 'package:http/http.dart' as http;
import 'package:url_launcher/url_launcher.dart';
- Define RSS Feed URL (FEED_URL)
- Create a variable to hold our RSS feed data. (_feed)
- Create a place holder for our title (_title)
- Create a method to navigate to the selected RSS item (openFeed)
- Use RssFeed.parse(response.body)to grab the RSS data from the provided URL.
- Create the UI for the ListView and plug in the retrieved RSS data

Code :

```
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:webfeed/webfeed.dart';
import 'package:http/http.dart' as http;
import 'package:url_launcher/url_launcher.dart';
void main() {
  runApp(const RSSDemo());
}

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(title: "RSS Feed", home: RSSMainPicture());
  }
}

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

  @override
  State<RSSMainPicture> createState() => _RSSMainPictureState();
}
```

```
class _RSSMainPictureState extends State<RSSMainPicture> {
  late Future<RssFeed> result;
  Future<RssFeed> giver() async {
    var response =
        await http.get(Uri.parse("https://www.espncricinfo.com/rss/content/story/feeds/0.xml"));
    var channel = RssFeed.parse(response.body);
    return channel;
  }

  @override
  void initState() {
    super.initState();
    result = giver();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("News"),
        actions: [
          IconButton(onPressed: ()=>result=giver(), icon: const Icon(Icons.refresh_rounded)),
        ],
      ),
      body: FutureBuilder<RssFeed?>(
        future: result,
        builder: (context,snapshot){
          if(snapshot.hasError){
            if(kDebugMode){
              print("Error");
            }
            return Container();
          }
          else if(snapshot.connectionState==ConnectionState.waiting){
            return const Center(
              child: CircularProgressIndicator(),
            );
          }
          else if(snapshot.hasData){
            var feed=snapshot.data!;
            var items=feed.items;
            return ListView.builder(
              itemCount: items?.length,
              itemBuilder: (context,index){
                var item=items![index];
            return GestureDetector(
              onTap: () async{
                if (!await launchUrl(Uri.parse(item.link!))) {
                  throw 'Could not launch ${item.link}';
                }
              },
              child: ListTile(
                // leading: CachedNetworkImage(
                //   imageUrl: mediaImage!,
                //   progressIndicatorBuilder: (context, url, downloadProgress) =>
                //       CircularProgressIndicator(value: downloadProgress.progress),
                //   errorWidget: (context, url, error) => const Icon(Icons.error),
                // ),
                title: Text(item.title!),
                subtitle: Text("${item.pubDate!}"),
              ),);
          },);}
      return Container();
    },
  ),); }}
```

Output:


2:48

News

India's home season to begin on January 3 with Sri Lanka T20Is
2022-12-08 09:10:35.000

Tanveer Sangha hoping for Sydney Thunder BBL return
2022-12-08 08:48:00.000

Warner's manager claims Australia were 'told' to tamper with ball after Hobart loss in 2016-17
2022-12-08 08:16:48.000

Marnus Labuschagne firm after West Indies strike twice in quick succession
2022-12-08 09:16:50.000

Ben Stokes embraces the 'why not?' as England look to press harder still
2022-12-07 22:36:52.000

Mehidy: 'I never believed I could get the hundred'
2022-12-07 19:32:46.000

Ollie Pope retains gloves as Mark Wood returns for Multan Test
2022-12-08 09:09:06.000

Rohit frustrated with India's mounting injury list: 'We need to get to the bottom of it'
2022-12-08 06:12:03.000

ZC to launch six-team T10 tournament in March 2023
2022-12-07 17:47:09.000

Maia Bouchier, Alice Davidson-Richards added to England T20I squad

Result: The application has been developed successfully and output verified

EX. No.7 MULTI-THREADING

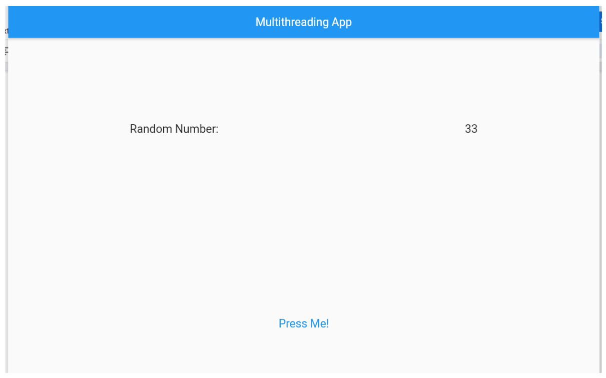
Aim: To write an application that implements multi-threading.

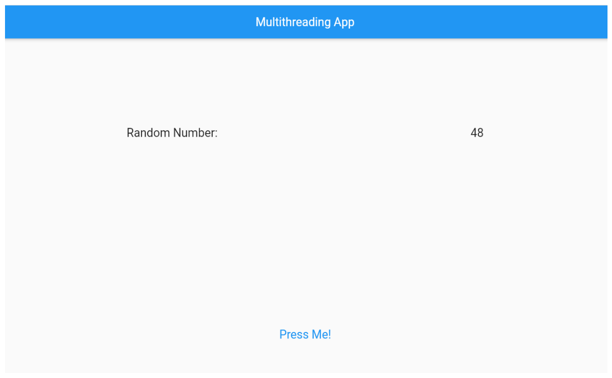
PROCEDURE:

-Implementing random number generation using multithreading
-import package for material.dart and foundation.dart
-Have a button that helps replace number generated each time
Code :

```
import 'dart:async';
import 'dart:math';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
void main() => runApp(Home());
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);
  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  int randint = 99;
  static FutureOr<int> randGen(int cal) {
    var rng = Random();
    return rng.nextInt(100);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Multithreading App",
        ),
        centerTitle: true,
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              Text(
                "Random Number: ",
                style: TextStyle(
                  fontSize: 20.0,
                ),
              ),
              Text(
                "${randint}",
                style: TextStyle(
                  fontSize: 20.0,
                ),),  ], ),
          SizedBox(
            height: 20.0,
          ),
          TextButton(
            onPressed: () async {
              int result = await compute(randGen, randint);
              setState(() {
                randint = result;
              });},
            child: Text(
              "Press Me!",
              style: TextStyle(
                fontSize: 20.0,
              ),),),],),);}}
```

Output:





Result: Thus, an application that implements multithreading is implemented using Flutter

EX. No.8 GPS LOCATION INFORMATION

Aim: To develop a native application that uses GPS location information.

PROCEDURE:

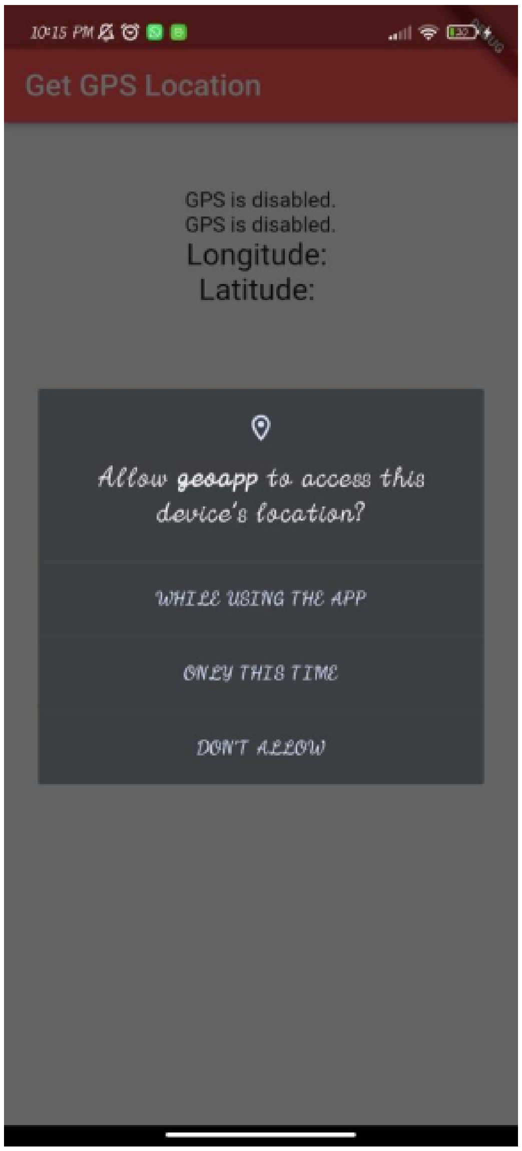
- Install the following packages: geolocator & geocoding
- Import them using,
    - import 'package:geocoding/geocoding.dart';
    - import 'package:geolocator/geolocator.dart';
- Get current location of the device, by creating an instance of Geolocator and calling getCurrentPosition.
- Convert latitude and longitude values into address using placemarkFromCoordinates().

Code :

```
// ignore_for_file: prefer_const_constructors, avoid_print
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:geolocator/geolocator.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget{
@override
Widget build(BuildContext context) {
return MaterialApp(
home: Home()
);
}
}
class Home extends StatefulWidget {
@override
State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
bool servicestatus = false;
bool haspermission = false;
late LocationPermission permission;
late Position position;
String long = "", lat = "";
late StreamSubscription<Position> positionStream;
@override
void initState() {
```

```
checkGps();
super.initState();
}
checkGps() async {
servicestatus = await Geolocator.isLocationServiceEnabled();
if(servicestatus){
permission = await Geolocator.checkPermission();
if (permission == LocationPermission.denied) {
permission = await Geolocator.requestPermission();
if (permission == LocationPermission.denied) {
print('Location permissions are denied');
}else if(permission == LocationPermission.deniedForever){
print("'Location permissions are permanently denied");
}else{
haspermission = true;
print("Has location permission");
}
}else{
haspermission = true;
print("Has location permission");
}
if(haspermission){
setState(() {
//refresh the UI
});
getLocation();
}
}else{
print("GPS Service is not enabled, turn on GPS location");
}
setState(() {
//refresh the UI
});
}
getLocation() async {
position = await Geolocator.getCurrentPosition(desiredAccuracy:
LocationAccuracy.high);
print(position.longitude); //Output: 80.24599079
print(position.latitude); //Output: 29.6593457
long = position.longitude.toString();
lat = position.latitude.toString();
setState(() {
//refresh UI
});
LocationSettings locationSettings = LocationSettings(
accuracy: LocationAccuracy.high, //accuracy of the location data
distanceFilter: 100, //minimum distance (measured in meters) a
//device must move horizontally before an
update event is generated;
);
StreamSubscription<Position> positionStream =
Geolocator.getPositionStream(
locationSettings: locationSettings).listen((Position position) {
print(position.longitude); //Output: 80.24599079
print(position.latitude); //Output: 29.6593457
long = position.longitude.toString();
lat = position.latitude.toString();
setState(() {
//refresh UI on update
});
});
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Get GPS Location"),
backgroundColor: Colors.redAccent
),
body: Container(
alignment: Alignment.center,
padding: EdgeInsets.all(50),
child: Column(
children: [
Text(servicestatus? "GPS is Enabled": "GPS is
disabled."),
Text(haspermission? "GPS is Enabled": "GPS is
disabled."),
Text("Longitude: $long", style:TextStyle(fontSize: 20)),
Text("Latitude: $lat", style: TextStyle(fontSize: 20),)
]
)
)
);
}
}
```

Output:



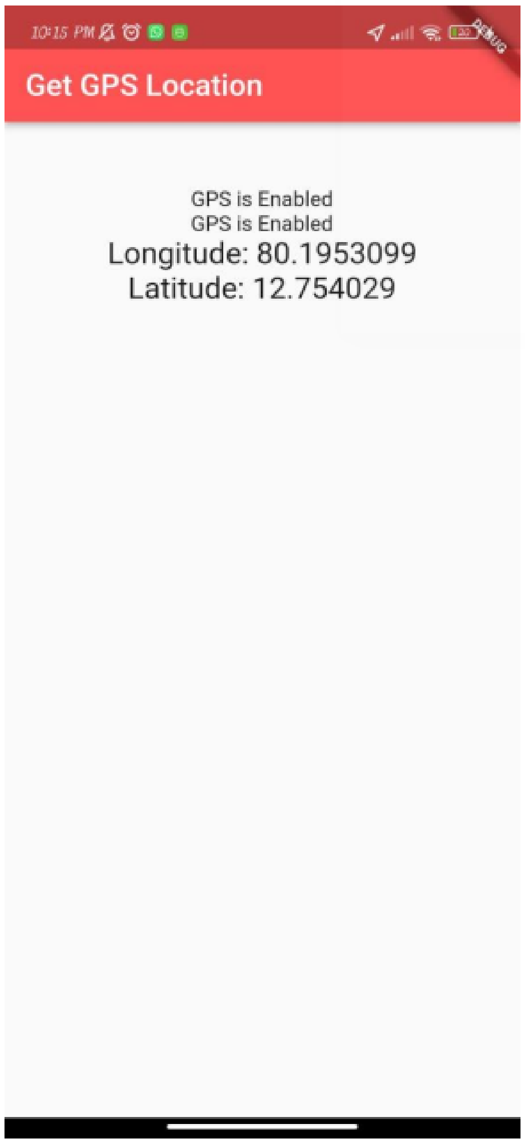
GPS is disabled.
GPS is disabled.
Longitude:
Latitude:

Allow geoapp to access this device's location?

WHILE USING THE APP

ONLY THIS TIME

DON'T ALLOW



GPS is Enabled
GPS is Enabled
Longitude: 80.1953099
Latitude: 12.754029

Result: The application has been developed successfully and output verified

EX. No.9 **WRITING TO SD CARD**

Aim: To implement an application that writes to SD card.

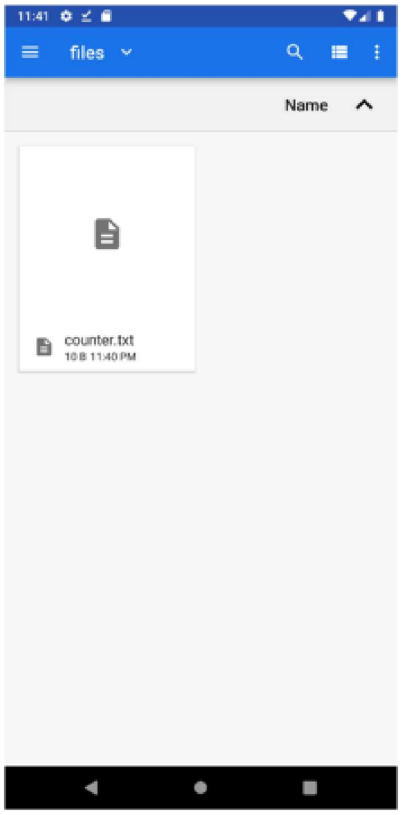
PROCEDURE:

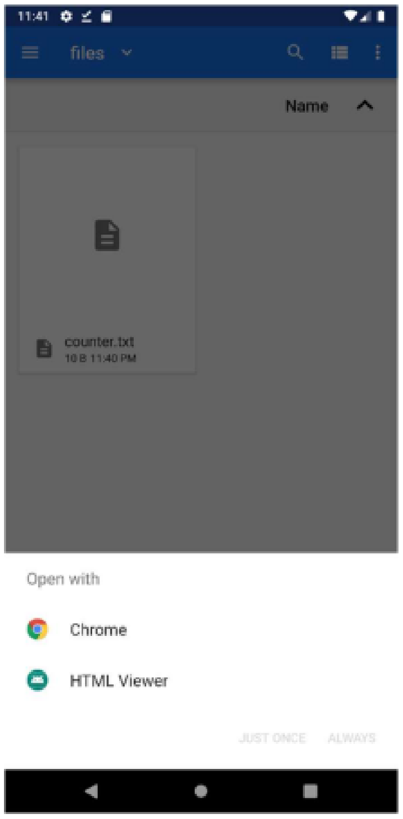
- Install path_provider package
- The path where is file is to be written is obtained using getExternalStorageDirectory() function.
- writeAsString(<String>) is used to write contents into a text file.
- readAsString() is used to read the contents of the file

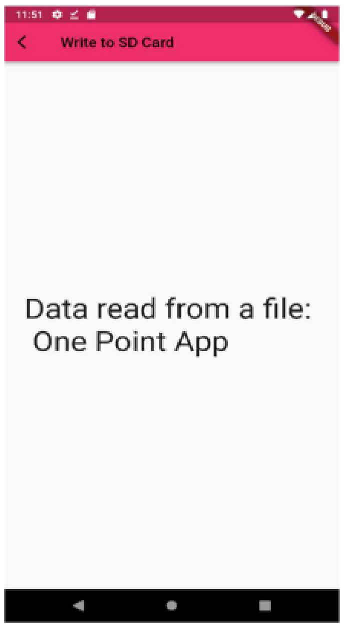
Code :

```
import 'dart:async';
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:path_provider/path_provider.dart';
class SDcard extends StatefulWidget {
 @override
 _AppState createState() => _AppState();
}
class _AppState extends State<SDcard> {
 String data='';
 Future<String> get _localPath async {
 final directory = await getExternalStorageDirectory();
 print(directory?.path);
 return directory!.path;
 }
 Future<File> get _localFile async {
 final path = await _localPath;
 return File('$path/counter.txt');
 }
 Future<String> readContent() async {
 try {
 final file = await _localFile;
 // Read the file
 String contents = await file.readAsString();
 // Returning the contents of the file
 return contents;
 } catch (e) {
 // If encountering an error, return
 return 'Error!';
 }
 }
 Future<File> writeContent() async {
 final file = await _localFile;
 // Write the file
 return file.writeAsString('One Point App');
 }
 @override
 void initState() {
 super.initState();
 writeContent();
 readContent().then((String value) {
 setState(() {
 data = value;
 });
 });
 }
 @override
 Widget build(BuildContext context) {
 return Scaffold(
 appBar: AppBar(
 title: Text("Write to SD Card", style: TextStyle(color: Colors.black)),
 leading: GestureDetector(
 child: Icon( Icons.arrow_back_ios, color: Colors.black, ),
 onTap: () {
 Navigator.pop(context);
 } ,
 ),
 backgroundColor: Color(0xffef2e6c),
 ),
 body: Center(
 child: Text(
 'Data read from a file: \n $data',style:TextStyle(fontSize: 40)
 ),),);}}
```

Output:







Data read from a file:
One Point App

Result: Thus an application that writes to SD card has been implemented using Flutter

EX. No.10 **ALERT BOX**

Aim: To implement an application that creates an alert upon receiving a message.

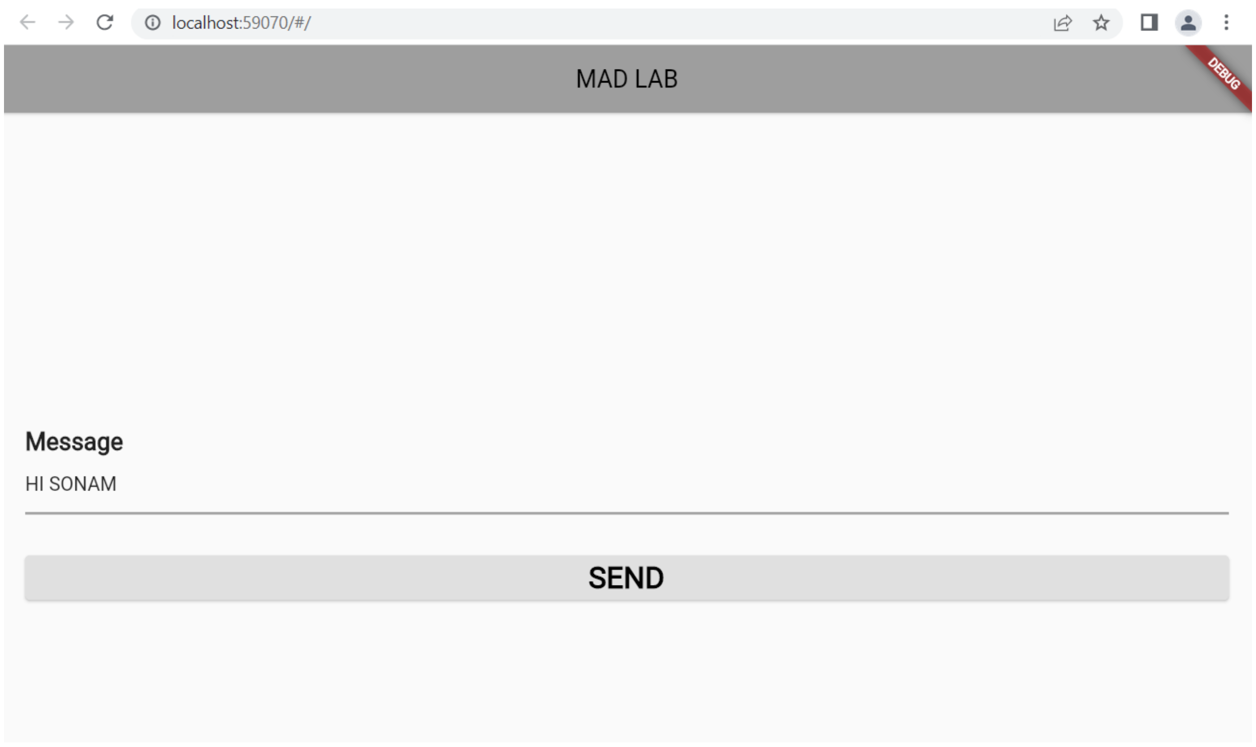
PROCEDURE:

- On the page,write a message and click send.
- In the onPressed() property, use showDialog to specify the alert box contents.
- AlertDialog() is used to create the alert message box.
    - o The message specified pops up as a alert.

Code :

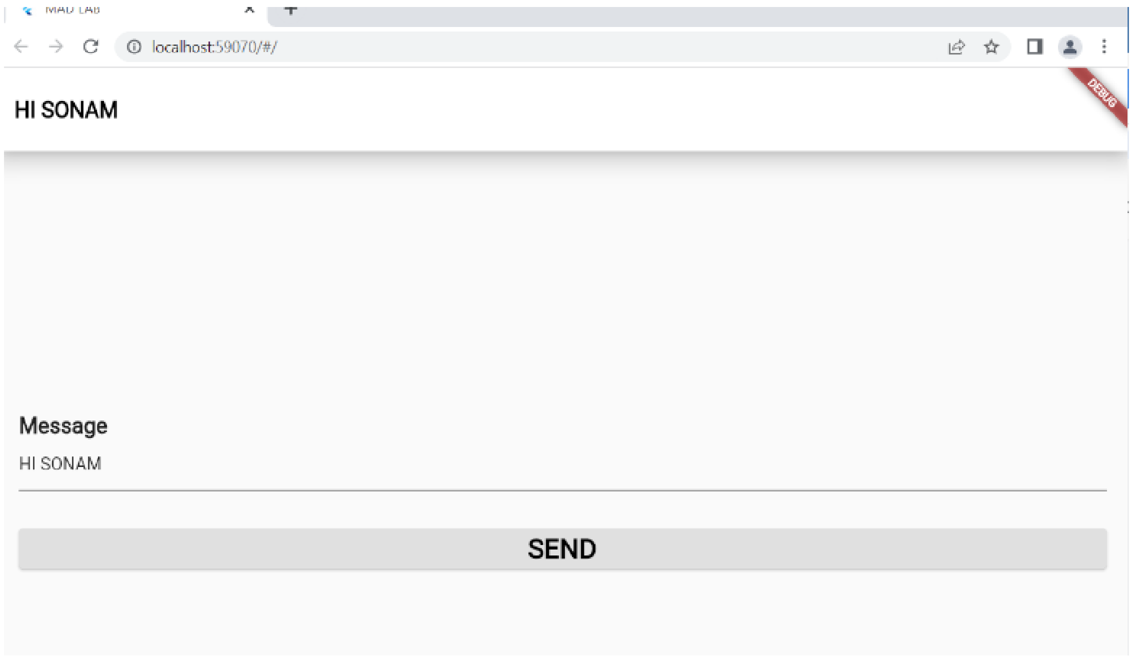
```
import 'package:flutter/material.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:overlay_support/overlay_support.dart';
void main() {
  runApp(Notifs);
}

class Notifs extends StatefulWidget {
const Notifs({ Key? key }) : super(key: key);
@override
State<Notifs> createState() => _NotifsState();
}
class _NotifsState extends State<Notifs> {
String msg="";
@override
Widget build(BuildContext context) {
return OverlaySupport.global(
child: MaterialApp(
title: 'MAD LAB',
theme: ThemeData(
primarySwatch: Colors.grey,
),
home: Scaffold(
appBar: AppBar(
centerTitle: true,
title: const Text(
'MAD LAB',
textAlign: TextAlign.center,
),
elevation: 2.0,
),
body: Padding(
padding: EdgeInsets.all(20.0),
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
Text(
'Message',
style: TextStyle(
fontSize: 20.0,
fontWeight: FontWeight.bold,
),),
TextField(
decoration: InputDecoration(
hintText: 'Enter your notification message',
),
onChanged: (String? newValue) {
setState(() {
msg = newValue!;
});},),
SizedBox(height: 35.0),
ElevatedButton(
style: ElevatedButton.styleFrom(
primary: Colors.grey[300],
onSurface: Colors.black,
padding: EdgeInsets.all(12.0),),
child: Text(
'SEND',
style: TextStyle(
fontSize: 25.0,
fontWeight: FontWeight.bold,
),),
onPressed: () {
showSimpleNotification(
Text(
"\n"+msg+"\n",
style: TextStyle(
fontSize: 20.0,
fontWeight: FontWeight.bold,
),),
background: Colors.white,
);},),],),),),
```

Output:



Message

HI SONAM



HI SONAM

Message

HI SONAM

Result: Thus, an application that creates an alert upon receiving a message is implemented using Flutter.

EX. No.11 **ALARM CLOCK**

Aim: To write a mobile application that creates an alarm clock.

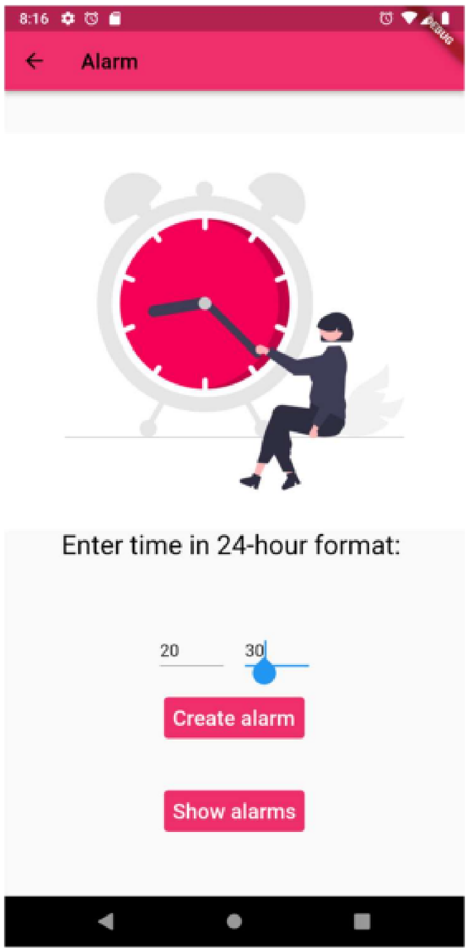
PROCEDURE:

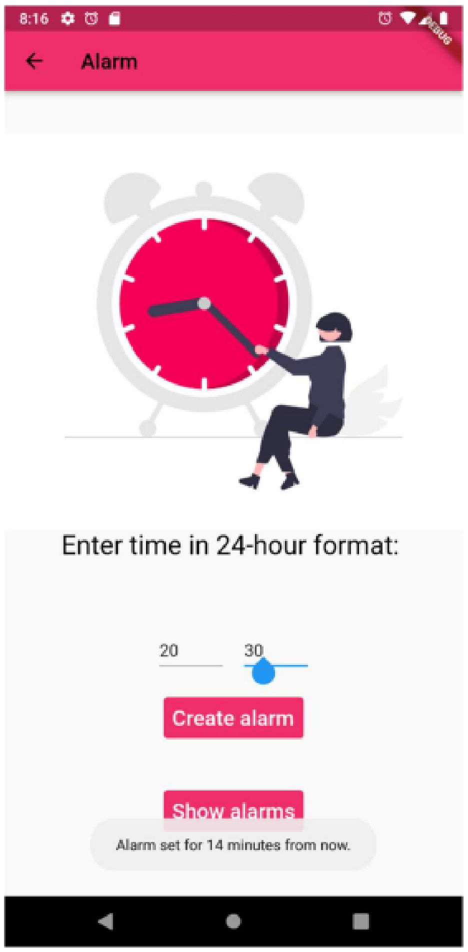
- Install the flutter_alarm_clock package using
  flutter pub add flutter_alarm_clock
- Import it using
  import 'package:flutter_alarm_clock/flutter_alarm_clock.dart';
- The FlutterAlarmClock.createAlarm() that takes hours and minutes as parameters.
- Hours and minutes are taken as input from user, using TextField().
- On clicking on “Create Alarm” button, a snackbar is displayed which appears when an alarm is set.
- The “Show Alarms” button, opens the clock application of the device which shows the created alarms.

Code :

```
import 'package:flutter/material.dart';
import 'package:flutter_alarm_clock/flutter_alarm_clock.dart';
class AlarmPage extends StatefulWidget {
 @override
 _AlarmPageState createState() => _AlarmPageState();
}
class _AlarmPageState extends State<AlarmPage> {
 TextEditingController hourController = TextEditingController();
 TextEditingController minuteController = TextEditingController();
 @override
 Widget build(BuildContext context) {
 return Scaffold(
 appBar: AppBar(
 iconTheme: IconThemeData(
 color: Colors.black, //change your color here
 ),
 backgroundColor: Color(0xffef2e6c),
 title: Text("Alarm", style: TextStyle(color: Colors.black)),
 ),
 body: Center(
 child: Column(
 mainAxisAlignment: MainAxisAlignment.center,
 children: <Widget>[
 Image.asset('assets/images/undraw_Time_management_re_tk5w.png'),
 Text('Enter time in 24-hour format: \n',style:TextStyle(fontSize:
25,color:Colors.black)),
 SizedBox(height: 30),
 Row(
 mainAxisAlignment: MainAxisAlignment.center,
 children: [
 Container(
 height: 40,
 width: 60,
 child: Center(
 child: TextField(
 controller: hourController,
 keyboardType: TextInputType.number,
 ),
 ),
 ),
 SizedBox(width: 20),
 Container(
 height: 40,
 width: 60,
 child: Center(
 child: TextField(
 controller: minuteController,
 keyboardType: TextInputType.number,
 ),),),], ),
 Container(
 margin: const EdgeInsets.all(25),
 child: TextButton(
 style: ButtonStyle(backgroundColor:
MaterialStateProperty.all(Color(0xffef2e6c))),
 child: const Text(
 'Create alarm',
 style: TextStyle(fontSize: 20.0,color:Colors.white),
 ),
 onPressed: () {
 int hour;
 int minutes;
 hour = int.parse(hourController.text);
 minutes = int.parse(minuteController.text);
 FlutterAlarmClock.createAlarm(hour, minutes);
 },
 ),
 ),
 Container(
 margin: const EdgeInsets.all(15),
 child: TextButton(
 style: ButtonStyle(backgroundColor:
MaterialStateProperty.all(Color(0xffef2e6c))),

 child: const Text(
 'Show alarms',
 style: TextStyle(fontSize: 20.0,color:Colors.white),
 ),
 onPressed: () {
 FlutterAlarmClock.showAlarms();
 },),),],),),);}}
```

Output:





Alarm set for 14 minutes from now.

Result: The application has been developed successfully and output verified

EX. No.12 SIMPLE GAME WITH MULTIMEDIA SUPPORT

Aim: To implement a simple gaming application with multimedia support.

PROCEDURE:

- Create a class TileModel for each tile, which has the following as members
o ImageAssetPath
o IsSelected
- Create a list called 'pairs' which contains a pair of each tile of a specific image.
- Use GridView to display the tiles as a 4x4 grid.
- Initialize points as 0 using setState().
- For every matched tile, increment points by 100.
- Play until points == 800.
- Click on replay to restart the game

Code :

data.dart

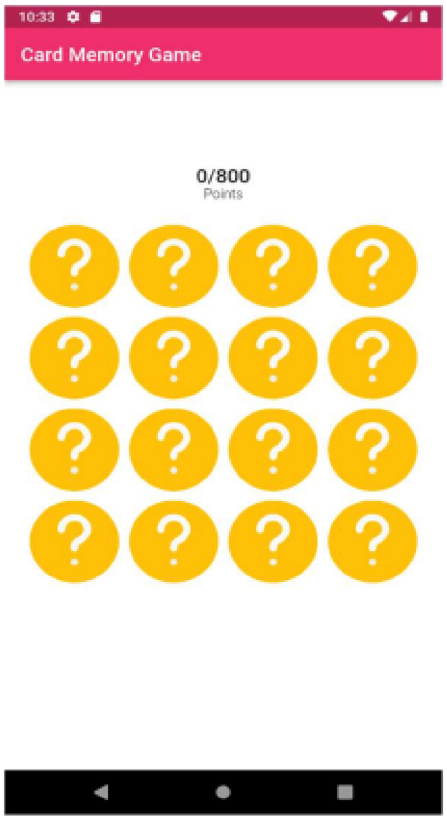
```
import 'package:memory_game/models/TileModel.dart';
String selectedTile = "";
int selectedIndex ;
bool selected = true;
int points = 0;
List<TileModel> myPairs = new List<TileModel>();
List<bool> clicked = new List<bool>();
List<bool> getClicked(){
 List<bool> yoClicked = new List<bool>();
 List<TileModel> myairs = new List<TileModel>();
 myairs = getPairs();
 for(int i=0;i<myairs.length;i++){
 yoClicked[i] = false;
 }
 return yoClicked;
}
List<TileModel> getPairs(){
 List<TileModel> pairs = new List<TileModel>();
 TileModel tileModel = new TileModel();
 //1
 tileModel.setImageAssetPath("assets/fox.png");

 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //2
 tileModel.setImageAssetPath("assets/hippo.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //3
 tileModel.setImageAssetPath("assets/horse.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //4
 tileModel.setImageAssetPath("assets/monkey.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //5
 tileModel.setImageAssetPath("assets/panda.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //6
 tileModel.setImageAssetPath("assets/parrot.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //7
 tileModel.setImageAssetPath("assets/rabbit.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 //8
 tileModel.setImageAssetPath("assets/zoo.png");
 tileModel.setIsSelected(false);
 pairs.add(tileModel);

 pairs.add(tileModel);
 tileModel = new TileModel();
 return pairs;
}
List<TileModel> getQuestionPairs(){
 List<TileModel> pairs = new List<TileModel>();
 TileModel tileModel = new TileModel();
```

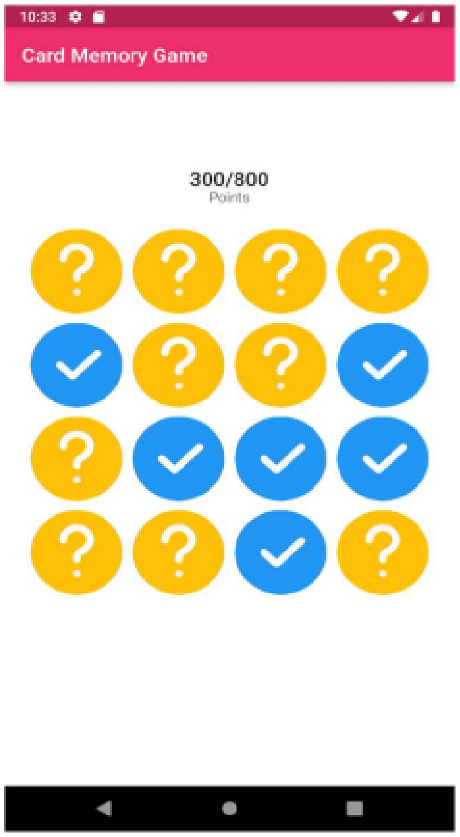
```
//1
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//2
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//3
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//4
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//5
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//6
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);

pairs.add(tileModel);
tileModel = new TileModel();
//7
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
pairs.add(tileModel);
pairs.add(tileModel);
tileModel = new TileModel();
//8
tileModel.setImageAssetPath("assets/question.png");
tileModel.setIsSelected(false);
```

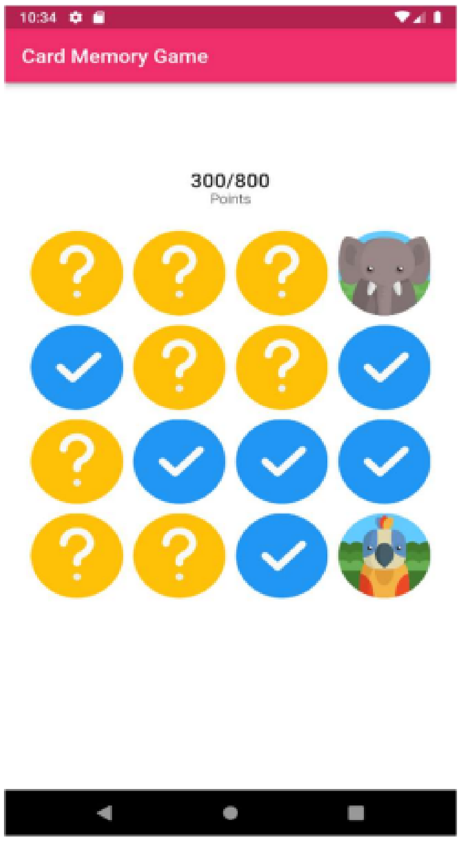
```
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new TileModel();
 return pairs;
}
TileModel.dart
class TileModel{
 String imageAssetPath;
 bool isSelected;
 TileModel({this.imageAssetPath, this.isSelected});
 void setImageAssetPath(String getImageAssetPath){
 imageAssetPath = getImageAssetPath;
 }
 String getImageAssetPath(){
 return imageAssetPath;
 }
 void setIsSelected(bool getIsSelected){
 isSelected = getIsSelected;
 }
 bool getIsSelected(){
 return isSelected;
 }
}
main.dart
import 'dart:async';

import 'package:flutter/material.dart';
import 'package:memory_game/data/data.dart';
import 'package:memory_game/models/TileModel.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
 // This widget is the root of your application.
 @override
 Widget build(BuildContext context) {
 return MaterialApp(
 title: 'Card Memory Game',
 debugShowCheckedModeBanner: false,
 theme: ThemeData(
 // primaryColor: Color(0xffef2e6c),
 primarySwatch: Colors.red,
 ),
 home: Home(),
 );
 }
}
class Home extends StatefulWidget {
 @override
 _HomeState createState() => _HomeState();
}
class _HomeState extends State<Home> {
 List<TileModel> gridViewTiles = new List<TileModel>();
 List<TileModel> questionPairs = new List<TileModel>();
 @override
 void initState() {
 // TODO: implement initState
 super.initState();
 reStart();
 }
 void reStart() {
 myPairs = getPairs();
 myPairs.shuffle();
 gridViewTiles = myPairs;
 Future.delayed(const Duration(seconds: 5), () {
// Here you can write your code
 setState(() {
 print("2 seconds done");
 // Here you can write your code for open new view
 questionPairs = getQuestionPairs();
 gridViewTiles = questionPairs;

 selected = false;
 });
 });
 }
 @override
 Widget build(BuildContext context) {
 return Scaffold(
 appBar: AppBar(
 title: Text('Card Memory Game'),
 backgroundColor:Color(0xffef2e6c) ,
 ),
 backgroundColor: Colors.white,
 body: SingleChildScrollView(
 child: Container(
 padding: EdgeInsets.symmetric(horizontal: 20, vertical: 50),
 child: Column(
 children: <Widget>[
 SizedBox(
 height: 40,
 ),
 points != 800 ? Column(
 crossAxisAlignment: CrossAxisAlignment.center,
 children: <Widget>[
 Text(
 "$points/800",
 style: TextStyle(
 fontSize: 20, fontWeight: FontWeight.w500),
 ),
 Text(
 "Points",
 textAlign: TextAlign.start,
 style: TextStyle(
 fontSize: 14, fontWeight: FontWeight.w300),
 ),
 ],
 ) : Container(),
 SizedBox(
 height: 20,
 ),
 points != 800 ? GridView(
 shrinkWrap: true,
 //physics: ClampingScrollPhysics(),
 scrollDirection: Axis.vertical,
 gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(
 mainAxisSpacing: 0.0, maxCrossAxisExtent: 100.0),
 children: List.generate(gridViewTiles.length, (index) {
 return Tile(
 imagePathUrl: gridViewTiles[index].getImageAssetPath(),
 tileIndex: index,
 parent: this,
 );
 }),
 ) : Container(
 child: Column(
 children: <Widget>[
 GestureDetector(
 onTap: (){
 setState(() {
 points = 0;
 reStart();
 });
 },
 child: Container(
 height: 50,
 width: 200,
 alignment: Alignment.center,
 decoration: BoxDecoration(
 color: Color(0xffef2e6c),
  borderRadius: BorderRadius.circular(24),
  ),
  child: Text("Replay", style: TextStyle(
  color: Colors.white,
  fontSize: 17,
  fontWeight: FontWeight.w500
  ),),),),
  SizedBox(height: 20,),
  ],) )],),),),),);}}
 class Tile extends StatefulWidget {
  String imagePathUrl;
  int tileIndex;
  _HomeState parent;
  Tile({this.imagePathUrl, this.tileIndex, this.parent});

  @override
  _TileState createState() => _TileState();
 }
 class _TileState extends State<Tile> {
  @override
  Widget build(BuildContext context) {
  return GestureDetector(
  onTap: () {
  if (!selected) {
  setState(() {
  myPairs[widget.tileIndex].setIsSelected(true);
  });
  if (selectedTile != "") {
  /// testing if the selected tiles are same
  if (selectedTile == myPairs[widget.tileIndex].getImageAssetPath()) {
  print("add point");
  points = points + 100;
  print(selectedTile + " thishis" + widget.imagePathUrl);
  TileModel tileModel = new TileModel();
  print(widget.tileIndex);
  selected = true;
  Future.delayed(const Duration(seconds: 2), () {
  tileModel.setImageAssetPath("");
  myPairs[widget.tileIndex] = tileModel;
  print(selectedIndex);
  myPairs[selectedIndex] = tileModel;
  this.widget.parent.setState(() {});
  setState(() {
  selected = false;
  });
  selectedTile = "";
```

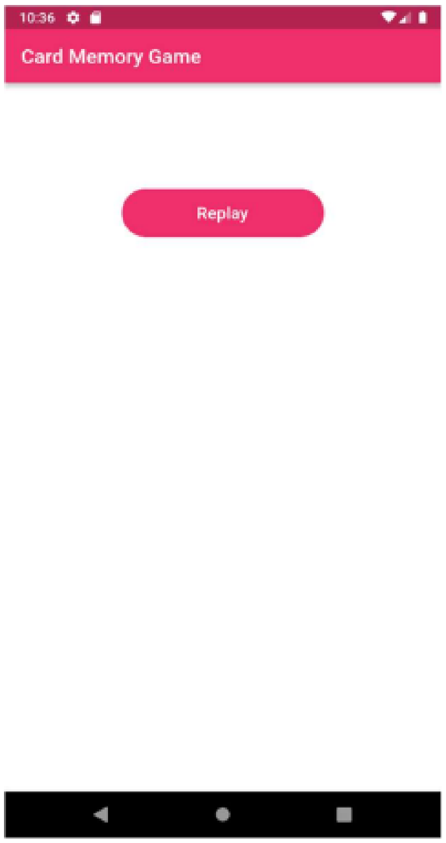
```
});
} else {
print(selectedTile +
" thishis " +
myPairs[widget.tileIndex].getImageAssetPath());
print("wrong choice");
print(widget.tileIndex);
print(selectedIndex);
selected = true;
Future.delayed(const Duration(seconds: 2), () {
this.widget.parent.setState(() {
myPairs[widget.tileIndex].setIsSelected(false);
myPairs[selectedIndex].setIsSelected(false);
});
setState(() {
selected = false;
}); });
selectedTile = "";
}
} else {
setState(() {
selectedTile = myPairs[widget.tileIndex].getImageAssetPath();
selectedIndex = widget.tileIndex;
});
print(selectedTile);
print(selectedIndex);
}}},
child: Container(
margin: EdgeInsets.all(5),
child: myPairs[widget.tileIndex].getImageAssetPath() != ""
? Image.asset(myPairs[widget.tileIndex].getIsSelected()
? myPairs[widget.tileIndex].getImageAssetPath()
: widget.imagePathUrl)
: Container(
color: Colors.white,
child: Image.asset("assets/correct.png"),
),),);}}
```

Output:


10:33
Card Memory Game
0/800
Points


10:33
Card Memory Game
300/800
Points





Result:

Thus, a simple gaming application that supports multimedia is implemented using Flutter.

EX. No.13 **CONNECTIVITY VIA SOAP OR REST**

Aim: To develop an application that uses GUI components, Font, and Colors. To a mobile application for data handling and connectivity via SOAP or REST to backend services potentially hosted in a cloud environment.

PROCEDURE:

- Import,
o http.dart
o dart:convert
- Specify the URL of the API within "Uri.parse(<>)"
- http.get() is used to fetch url contents.

Code :

quotes.dart:

```
// To parse this JSON data, do
//
// final quotes = quotesFromJson(jsonString);
import 'dart:convert';
Quotes quotesFromJson(String str) => Quotes.fromJson(json.decode(str));
String quotesToJson(Quotes data) => json.encode(data.toJson());
class Quotes {
 Quotes({
 this.id,
 this.tags,
 this.content = '',
 this.author = '',
 this.authorSlug,
 this.length,
 this.dateAdded,
 this.dateModified,
 });
 String? id;
 List<String>? tags;
 String content;
 String author;
 String? authorSlug;
 int? length;
 DateTime? dateAdded;
 DateTime? dateModified;
 factory Quotes.fromJson(Map<String, dynamic> json) => Quotes(
 id: json["_id"],
 tags: List<String>.from(json["tags"].map((x) => x)),
 content: json["content"],
 author: json["author"],
 authorSlug: json["authorSlug"],
 length: json["length"],
 dateAdded: DateTime.parse(json["dateAdded"]),
 dateModified: DateTime.parse(json["dateModified"]),
 );
 Map<String, dynamic> toJson() => {
 "_id": id,
 "tags": List<dynamic>.from(tags!.map((x) => x)),
 "content": content,
 "author": author,
 "authorSlug": authorSlug,
 "length": length,
 "dateAdded":
 "${dateAdded!.year.toString().padLeft(4, '0')}-${dateAdded!.month.toString().padLeft(2,
'0')}-${dateAdded!.day.toString().padLeft(2, '0')}",
 "dateModified":
 "${dateModified!.year.toString().padLeft(4, '0')}-
${dateModified!.month.toString().padLeft(2, '0')}-${dateModified!.day.toString().padLeft(2,
'0')}",
 };
}
```

api.dart

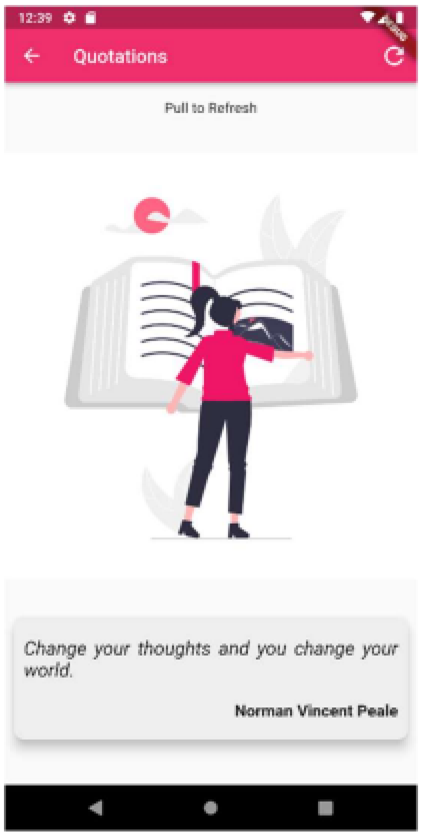
```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'quotes.dart';
class Api {
 static Future<Quotes?> getQuotes() async {
 Uri url = Uri.parse('http://api.quotable.io/random');
 http.Response response = await http.get(url);
 if (response.statusCode == 200) {
 print("success");
 return Quotes.fromJson(jsonDecode(response.body));
 } else {
 print("error in getting data");
 }
 }
}
```

quotes_page.dart:

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'quotes.dart';
import 'api.dart';
class QuotesScreen extends StatefulWidget {
 QuotesScreen({Key? key}) : super(key: key);
 @override
 State<QuotesScreen> createState() => _QuotesScreenState();
}
class _QuotesScreenState extends State<QuotesScreen> {
 var size, height, width;
 Quotes? data;
 @override
 Widget build(BuildContext context) {
 size = MediaQuery.of(context).size;
 height = size.height;
 width = size.width;
 return Scaffold(
 appBar: AppBar(
 backgroundColor: Color(0xffef2e6c),
 title: Text("Quotations"),
 actions: [
 IconButton(
 icon: Icon(
 Icons.refresh_outlined,
 ),
 iconSize: 30,
 onPressed: () {
 print("icon refresh");
 getQuotes();
 },
 ),
 ],
 ),
 body: RefreshIndicator(
 onRefresh: getQuotes,
 child: ListView(
 children: [
 Padding(
 padding: const EdgeInsets.all(18.0),
 child: Text(
 "Pull to Refresh",
 textAlign: TextAlign.center,
 style: TextStyle(
 fontSize: 15,
 ),
 ),
 ),
 SizedBox(height: 20),
 Image.asset('assets/images/undraw_Bibliophile_re_xarc.png'),
 SizedBox(height: 20),
 Container(
 padding: EdgeInsets.symmetric(
 horizontal: 10,
 ),
 width: width / 2,
 child: Card(
 margin: EdgeInsets.only(top: 20),
 color: Color(0XFFeeeeee),
 shape: RoundedRectangleBorder(
 borderRadius: BorderRadius.circular(10.0),
 ),
 elevation: 10,
 child: Padding(
 padding: EdgeInsets.symmetric(horizontal: 10, vertical: 20),
 child: Column(
 mainAxisAlignment: MainAxisAlignment.center,
 children: [
 Text(
 '${data?.content ?? "Don't talk about what you have done or what you are
going to do."}',
 textAlign: TextAlign.justify,
 style: TextStyle(
 fontSize: 20,
 fontStyle: FontStyle.italic,
 ),
 ),
 SizedBox(height: 22),
 Align(
 alignment: Alignment.bottomRight,
 child: Text(
 data?.author ?? "Thomas Jefferson",
 textAlign: TextAlign.justify,
 style: TextStyle(
 fontSize: 17,
 fontWeight: FontWeight.bold,
 ),))],
 ),),), )],),
),);}
 Future<Null> getQuotes() async {
```

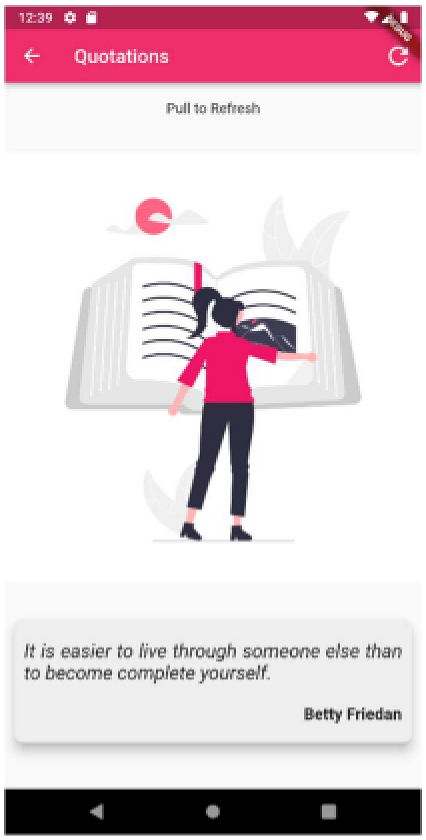
```
 data = await Api.getQuotes();
 setState(() {});
 }
}
```

Output:



Change your thoughts and you change your world.

Norman Vincent Peale



It is easier to live through someone else than to become complete yourself.

Betty Friedan

Result: The application for data handling and connectivity via SOAP or REST to backend services is potentially hosted in a cloud environment

EX. No.14

GEO-POSITIONING, ACCELEROMETER AND RICH GESTURE BASED UI

Aim: To write a mobile application that will implement GEO positioning, accelerometer, and rich gesture-based UI handling.

PROCEDURE:

Geo-positioning:
- Install the following packages: geolocator & geocoding
- Import them using,
o import 'package:geocoding/geocoding.dart';
o import 'package:geolocator/geolocator.dart';
- Get current location of the device, by creating an instance of Geolocator and calling getCurrentPosition.
- Convert latitude and longitude values into address using placemarkFromCoordinates().

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 :

Geo-positioning:

```
import 'package:flutter/material.dart';
import 'package:geocoding/geocoding.dart';
import 'package:geolocator/geolocator.dart';
class LocationPage extends StatefulWidget {
 @override
 _LocationPageState createState() => _LocationPageState();
}
class _LocationPageState extends State<LocationPage> {
 Position? _currentPosition;
 String _currentAddress = '';

 @override
 Widget build(BuildContext context) {
 return Scaffold(
 appBar: AppBar(
 iconTheme: IconThemeData(
 color: Colors.black, //change your color here
 ),
 backgroundColor: Color(0xffef2e6c),
 title: Text("Location",style:TextStyle(color:Colors.black)),
 ),
 body: Center(
 child: Column(
 mainAxisAlignment: MainAxisAlignment.center,
 children: <Widget>[
 Image.asset('assets/images/undraw_Current_location_re_j130.png'),
 TextButton(
 style: ButtonStyle(backgroundColor: MaterialStateProperty.all(Color(0xffef2e6c))),
 child: Text("Get location",style:TextStyle(fontSize: 20,color:Colors.white)),
 onPressed: () {
 _getCurrentLocation();
 },
 ),
 Divider(color:Colors.transparent,thickness: 150),
 if (_currentAddress != null) Text(
 _currentAddress,style: TextStyle(fontSize: 20),
 ),
 if (_currentPosition != null) Text( 'Latitude : ' +
 _currentPosition!.latitude.toString(),style: TextStyle(fontSize: 20),
 ),
 if (_currentPosition != null) Text( 'Longitude : ' +
 _currentPosition!.longitude.toString(),style: TextStyle(fontSize: 20),
 ),],),),); }
 _getCurrentLocation() {
 Geolocator
 .getCurrentPosition(desiredAccuracy: LocationAccuracy.best,
forceAndroidLocationManager: true)
 .then((Position position) {
 setState(() {
 _currentPosition = position;
 _getAddressFromLatLng();
 });
 }).catchError((e) {
 print(e);
 });
}
 _getAddressFromLatLng() async {
 try {
 List<Placemark> placemarks = await placemarkFromCoordinates(
 _currentPosition!.latitude,
 _currentPosition!.longitude
 );
 Placemark place = placemarks[0];
 setState(() {
 _currentAddress = "${place.locality}, ${place.postalCode}, ${place.country}";
 });
 } catch (e) {
 print(e);
 }
 }
}
```

Accelerometer:

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

class FocusPage extends StatefulWidget {
 final String title='Focus!';
 @override
 FocusPageState createState() => FocusPageState();
}
class FocusPageState extends State<FocusPage> {
 // color of the circle
 Color color = Colors.greenAccent;
 // event returned from accelerometer stream
 AccelerometerEvent? event;
 // hold a refernce to these, so that they can be disposed
 Timer? timer;
 StreamSubscription? accel;
 // positions and count
 double top = 125;
 double? left;
 int count = 0;
 // variables for screen size
 double? width;
 double? height;
 setColor(AccelerometerEvent event) {
 // Calculate Left
 double x = ((event.x * 12) + ((width! - 100) / 2));
 // Calculate Top
 double y = event.y * 12 + 125;
 // find the difference from the target position
 var xDiff = x.abs() - ((width! - 100) / 2);
 var yDiff = y.abs() - 125;
```

```
// check if the circle is centered, currently allowing a buffer of 3 to make centering easier
if (xDiff.abs() < 3 && yDiff.abs() < 3) {
// set the color and increment count
setState(() {
color = Colors.greenAccent;
count += 1;
});
} else {
// set the color and restart count
setState(() {
color = Colors.red;

count = 0;
});
}
}
setPosition(AccelerometerEvent event) {
if (event == null) {
return;
}
// When x = 0 it should be centered horizontally
// The left positin should equal (width - 100) / 2
// The greatest absolute value of x is 10, multipling it by 12 allows the left position to move
a total of 120 in either direction.
setState(() {
left = ((event.x * 12) + ((width! - 100) / 2));
});
// When y = 0 it should have a top position matching the target, which we set at 125
setState(() {
top = event.y * 12 + 125;
});
}
startTimer() {
// if the accelerometer subscription hasn't been created, go ahead and create it
if (accel == null) {
accel = accelerometerEvents.listen((AccelerometerEvent eve) {
setState(() {
event = eve;
});});
} else {
// it has already been created so just resume it
accel?.resume();
}
// Accelerometer events come faster than we need them so a timer is used to only proccess
them every 200 milliseconds
if (timer == null || !timer!.isActive) {
```

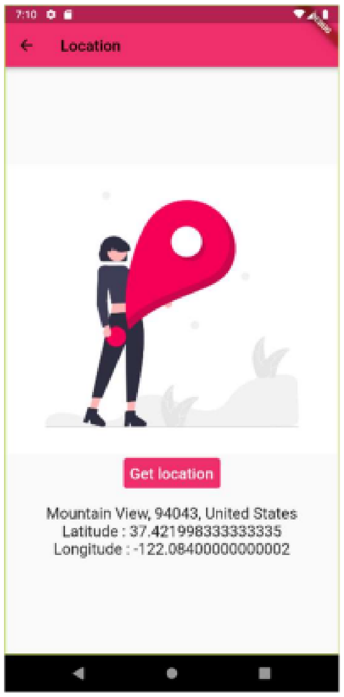
```
timer = Timer.periodic(Duration(milliseconds: 200), (_) {
// if count has increased greater than 3 call pause timer to handle success
if (count > 3) {
pauseTimer();
} else {
// proccess the current event
setColor(event!);
setPosition(event!);
}
});}}
pauseTimer() {
// stop the timer and pause the accelerometer stream
timer?.cancel();
accel?.pause();
// set the success color and reset the count
setState(() {
count = 0;
color = Colors.green;
});
}
@override
void dispose() {
timer?.cancel();
accel?.cancel();
super.dispose();
}
@override
Widget build(BuildContext context) {
// get the width and height of the screen
width = MediaQuery.of(context).size.width;
height = MediaQuery.of(context).size.height;
return Scaffold(
appBar: AppBar(
iconTheme: IconThemeData(
color: Colors.black, //change your color here
),
title: Text(widget.title,style:TextStyle(color:Colors.black)),
backgroundColor : Color(0xffef2e6c),
),
body: Column(
children: [
Padding(
padding: const EdgeInsets.all(8.0),
child: Text('Keep the circle in the center for 1 second',textAlign:
TextAlign.center,style: TextStyle(fontSize:25)),
),
Stack(
children: [
// This empty container is given a width and height to set the size of the stack
Container(
height: height! / 2,
width: width,

),
// Create the outer target circle wrapped in a Position
Positioned(
// positioned 50 from the top of the stack
// and centered horizontally, left = (ScreenWidth - Container width) / 2
top: 50,
left: (width! - 250) / 2,
child: Container(
height: 250,
width: 250,
decoration: BoxDecoration(
border: Border.all(color: Colors.red, width: 5.0),
borderRadius: BorderRadius.circular(125),
),),),
// This is the colored circle that will be moved by the accelerometer
// the top and left are variables that will be set
Positioned(
top: top,
left: left ?? (width! - 100) / 2,
// the container has a color and is wrappeed in a ClipOval to make it round
child: ClipOval(
child: Container(
width: 100,
height: 100,
color: color,
),),),
// inner target circle wrapped in a Position
Positioned(
top: 125,
left: (width! - 100) / 2,
child: Container(
height: 100,
width: 100,
decoration: BoxDecoration(
border: Border.all(color: Colors.green, width: 2.0),
borderRadius: BorderRadius.circular(50),
),),),],),
Text('x: ${(event?.x ?? 0).toStringAsFixed(3)}',style:TextStyle(fontSize: 20)),
Text('y: ${(event?.y ?? 0).toStringAsFixed(3)}',style:TextStyle(fontSize: 20)),
Padding(
padding: EdgeInsets.symmetric(horizontal: 16.0, vertical: 30.0),

child: TextButton(
style: ButtonStyle(backgroundColor: MaterialStateProperty.all(Color(0xffef2e6c))),
onPressed: startTimer,
child: Text('Begin.!!',style: TextStyle(fontSize: 30.0,color:Colors.white),),
// color: Theme.of(context).primaryColor,
// textColor: Colors.white,
), )],),); }}
```

Gesture based UI:

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
class AboutPage extends StatefulWidget {
 @override
 _AboutPageState createState() => _AboutPageState();
}
class _AboutPageState extends State<AboutPage> {
 bool _lightIsOn = false;
 @override
 void dispose() {
 super.dispose();
 }
 @override
 void initState() {
 super.initState();
 }
 @override
 Widget build(BuildContext context) {
 return MaterialApp(
 theme: _lightIsOn ? ThemeData.dark() : ThemeData.light(),
 home: Scaffold(
 appBar: AppBar(
 title: Text('About', style: TextStyle(color: Colors.black)),
 backgroundColor: Color(0xffef2e6c),
 ),
 body: Column(children: <Widget>[
 Container(
 margin: EdgeInsets.all(20),
 height: 200,
 width: 350,
 child: Image.asset('assets/images/logo.png'),
 ),
 Divider(color:Colors.black,thickness: 2,),
 Container(
 // alignment: FractionalOffset.center,
 child: Column(
 // mainAxisAlignment: MainAxisAlignment.center,
 children: <Widget>[
 GestureDetector(
 onTap: () {
 setState(() {

 // Toggle light when tapped.
 _lightIsOn = !_lightIsOn;
 });
 },
 child: Container(
 margin: EdgeInsets.fromLTRB(350, 10, 3, 6),
 width : 50,
 height:50,
 padding: const EdgeInsets.all(8),
 // Change button text when light changes state.
 decoration: BoxDecoration(
 shape : BoxShape.circle,
 color: Color(0xffef2e6c),
 ),
 child: Icon(
 _lightIsOn ? Icons.light_mode_outlined : Icons.dark_mode_outlined,
 size: 30),
 ),),],),
 ),
 Text('In publishing and graphic design, '
 'Lorem ipsum is a placeholder text commonly used to demonstrate '
 'the visual form of a document or a typeface without relying on '
 'meaningful content. Lorem ipsum may be used as a placeholder '
 'before final copy is available.',
 textAlign: TextAlign.center,
 softWrap: true,
 style: GoogleFonts.notoSerif(textStyle: TextStyle( color: _lightIsOn ? Colors.white :
Colors.black,fontSize: 20),)
 ),
 ]))); }}
```

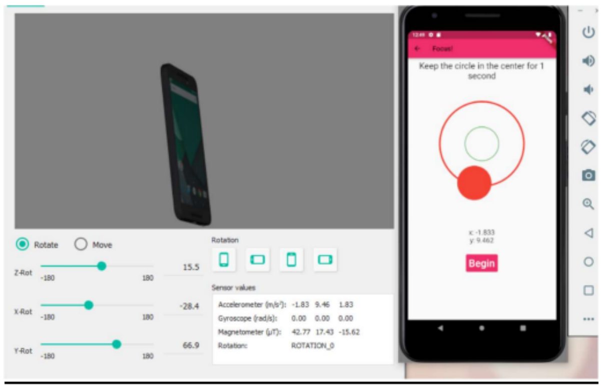
Output:

Geopositioning:

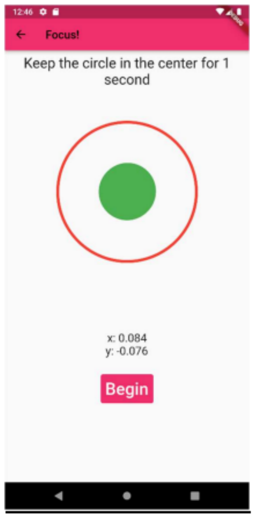


Mountain View, 94043, United States
Latitude : 37.421998333333335
Longitude : -122.08400000000002

Accelerometer:



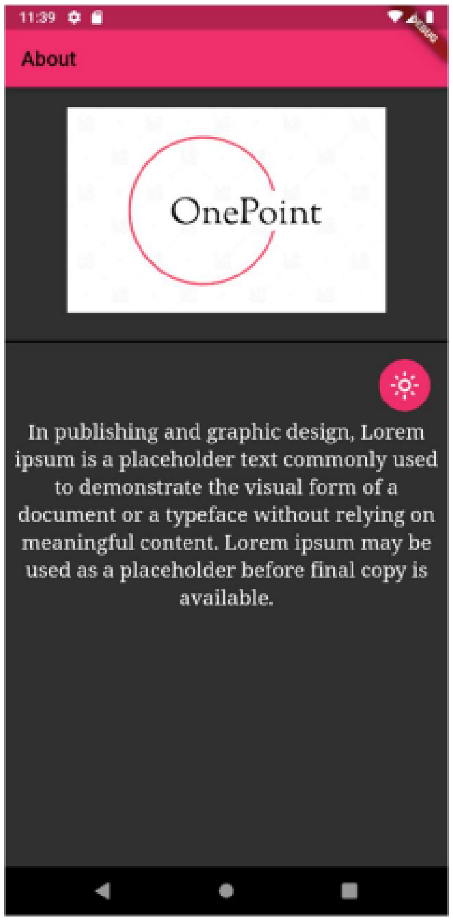
```
Z-Rot  -180  180   15.5
X-Rot  -180  180  -28.4
Y-Rot  -180  180   66.9

Sensor values
Accelerometer (m/s²):  -1.83  9.46   1.83
Gyroscope (rad/s):      0.00  0.00   0.00
Magnetometer (μT):     42.77 17.43 -15.62
Rotation:              ROTATION_0

x: -1.833
y: 9.462
```



```
x: 0.084
y: -0.076
```

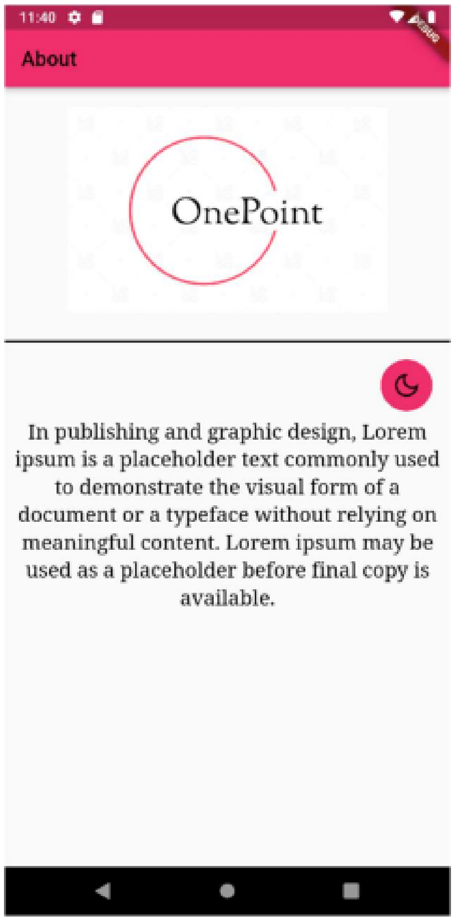
Gesture based UI:

Dark mode



In publishing and graphic design, Lorem ipsum is a placeholder text commonly used to demonstrate the visual form of a document or a typeface without relying on meaningful content. Lorem ipsum may be used as a placeholder before final copy is available.

Light mode



In publishing and graphic design, Lorem ipsum is a placeholder text commonly used to demonstrate the visual form of a document or a typeface without relying on meaningful content. Lorem ipsum may be used as a placeholder before final copy is available.

Result: Thus, GEO positioning, accelerometer, and rich gesture-based UI handling have been implemented using Flutter.

EX. No.15 **SOCIAL MEDIA INTEGRATION**

Aim: To write an application for integrating mobile applications in the market, including social networking software integration with Google.

PROCEDURE:

- Download the following packages using flutter pub add.
o firebase_auth
o firebase_core
o google_sign_in
- In the firebase console, enable Google as a provider under Authentication-> Sign In method.
- Get SHA key, by using the command gradlew signingReport at the android directory of the flutter application.
- Add SHA-1 fingerprint to the application.
- Now, get Google user credential using the await GoogleSignIn().signIn();
- Obtain the auth details from the request.
- Obtain the auth details from the request

Code :

authentication.dart:

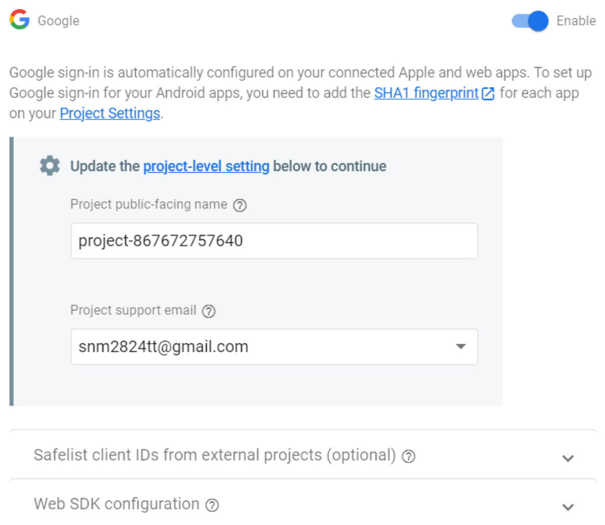
```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:google_sign_in/google_sign_in.dart';
class AuthenticationHelper {
 final FirebaseAuth _auth = FirebaseAuth.instance;
 get user => _auth.currentUser;
 Future<String?> signInWithGoogle() async {
 final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();
 final GoogleSignInAuthentication? googleAuth = await googleUser?.authentication;
 final credential = GoogleAuthProvider.credential(
 accessToken: googleAuth?.accessToken,
 idToken: googleAuth?.idToken,
 );
 await FirebaseAuth.instance.signInWithCredential(credential);
 return null;
 }

 Future<UserCredential> signInWithFacebook() async {
 // Trigger the sign-in flow
 final LoginResult loginResult = await FacebookAuth.instance.login();
 // Create a credential from the access token
 final OAuthCredential facebookAuthCredential =
FacebookAuthProvider.credential(loginResult.accessToken.token);
 // Once signed in, return the UserCredential
 return FirebaseAuth.instance.signInWithCredential(facebookAuthCredential);
 }
//SIGN UP METHOD
 Future<String?> signUp({required String email, required String password}) async {
 try {
 await _auth.createUserWithEmailAndPassword(
 email: email,
 password: password,
 );
 return null;
 } on FirebaseAuthException catch (e) {
 return e.message;
 }
 }
 //SIGN IN METHODJ
 Future<String?> signIn({required String email, required String password}) async {
 try {
 await _auth.signInWithEmailAndPassword(email: email, password: password);
 return null;
 } on FirebaseAuthException catch (e) {
 return e.message;
 }
 }
 //SIGN OUT METHOD
 Future<void> signOut() async {
 await _auth.signOut();
 print('signout');
 }
}
```

login.dart:

```
import 'package:flutter/material.dart';
import './authentication.dart';
import './home.dart';
import './signup.dart';

class Login extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
 return Scaffold(
 body: ListView(
 padding: EdgeInsets.all(8.0),
 children: <Widget>[
 SizedBox(height: 80),
 // logo
 Column(
 children: [
 Image.asset('assets/images/logo.png'),
 SizedBox(height: 50),
 Text(
 'Welcome back!',
 style: TextStyle(fontSize: 24),
 ),],),
 SizedBox(
 height: 50,
 ),
 Padding(
 padding: const EdgeInsets.all(16.0),
 child: LoginForm(),
 ),
 SizedBox(height: 20),
 Row(
 children: <Widget>[
 SizedBox(width: 30),
 Text('New here ? ',
 style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20)),
 GestureDetector(
 onTap: () {
 Navigator.pushReplacement(context,MaterialPageRoute(builder: (context) =>
Signup()));
 },
 child: Text('Get Registered Now..',
 style: TextStyle(fontSize: 20, color: Color(0xffef2e6c))),
 )],),
 Row(
 children: <Widget>[
 SizedBox(width: 30),
 GestureDetector(

 onTap: () {
 AuthenticationHelper()
 .signInWithGoogle()
 .then((result) {
 if (result == null) {
 Navigator.pushReplacement(context,
 MaterialPageRoute(builder: (context) => MyApp()));
 } else {
 ScaffoldMessenger.of(context).showSnackBar(SnackBar(
 content: Text(
 result,
 style: TextStyle(fontSize: 16),
 ),)), }}); },
 child: Text('Sign in with Google',
 style: TextStyle(fontSize: 20, color: Color(0xffef2e6c))),
 )],),],),);}}
class LoginForm extends StatefulWidget {
 LoginForm({Key? key}) : super(key: key);
 @override
 _LoginFormState createState() => _LoginFormState();
}
class _LoginFormState extends State<LoginForm> {
 final _formKey = GlobalKey<FormState>();
 String? email;
 String? password;
 bool _obscureText = true;
 @override
 Widget build(BuildContext context) {
 return Form(
 key: _formKey,
 child: Column(

 mainAxisAlignment: MainAxisAlignment.spaceAround,
 children: <Widget>[
 // email
 TextFormField(
 // initialValue: 'Input text',
 decoration: InputDecoration(
 prefixIcon: Icon(Icons.email_outlined,color:Colors.black),
 labelText: 'Email',
 labelStyle: TextStyle(
 color: Color(0xffef2e6c),
 ),
 enabledBorder: OutlineInputBorder(
 borderRadius: BorderRadius.all(
 const Radius.circular(100.0),
 ),
 ),
 focusedBorder: OutlineInputBorder(
 borderRadius: BorderRadius.all(
 const Radius.circular(100.0),
 ),
 borderSide: BorderSide(color: Color(0xffef2e6c) ),
 ),
 ),
 validator: (value) {
 if (value!.isEmpty) {
 return 'Please enter some text';
 }
 return null;
 },
 onSaved: (val) {
 email = val;
 },
 ),
 SizedBox(
 height: 20,
 ),
 // password
 TextFormField(
 // initialValue: 'Input text',
 decoration: InputDecoration(
 labelText: 'Password',
 labelStyle: TextStyle(
 color: Color(0xffef2e6c),
 ),
 prefixIcon: Icon(Icons.lock_outline,color:Colors.black),
 enabledBorder: OutlineInputBorder(
 borderRadius: BorderRadius.all(
 const Radius.circular(100.0),

 ),
 ),
 focusedBorder: OutlineInputBorder(
 borderRadius: BorderRadius.all(
 const Radius.circular(100.0),
 ),
 borderSide: BorderSide(color: Color(0xffef2e6c) ),
 ),
 suffixIcon: GestureDetector(
 onTap: () {
 setState(() {
 _obscureText = !_obscureText;
 });
 },
 child: Icon(
 _obscureText ? Icons.visibility_off : Icons.visibility,
 ),),),
 obscureText: _obscureText,
 onSaved: (val) {
```

```
password = val;
},
validator: (value) {
if (value!.isEmpty) {
return 'Please enter some text';
}
return null;
},
),
SizedBox(height: 30),
SizedBox(
height: 54,
width: 184,
child: ElevatedButton(
onPressed: () {
// Respond to button press
if (_formKey.currentState!.validate()) {
_formKey.currentState!.save();
AuthenticationHelper()
.signIn(email: email!, password: password!)
.then((result) {
if (result == null) {
Navigator.pushReplacement(context,
MaterialPageRoute(builder: (context) => MyApp()));

} else {
ScaffoldMessenger.of(context).showSnackBar(SnackBar(
content: Text(
result,
style: TextStyle(fontSize: 16),
),
));}}); } },
style: ElevatedButton.styleFrom(
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.all(Radius.circular(24.0))),
backgroundColor: Color(0xffef2e6c)),
child: Text(
'Login',
style: TextStyle(fontSize: 24),
), ), ), ],), ); }}
```

Output:



Google Enable

Google sign-in is automatically configured on your connected Apple and web apps. To set up Google sign-in for your Android apps, you need to add the SHA1 fingerprint for each app on your Project Settings.

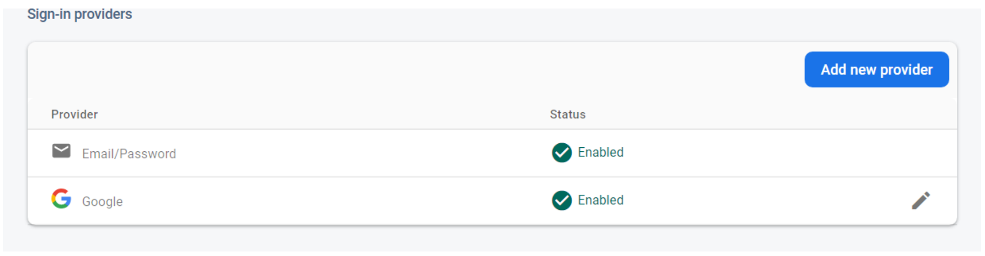
Update the project-level setting below to continue

Project public-facing name: project-867672757640

Project support email: snm2824tt@gmail.com

Safelist client IDs from external projects (optional)

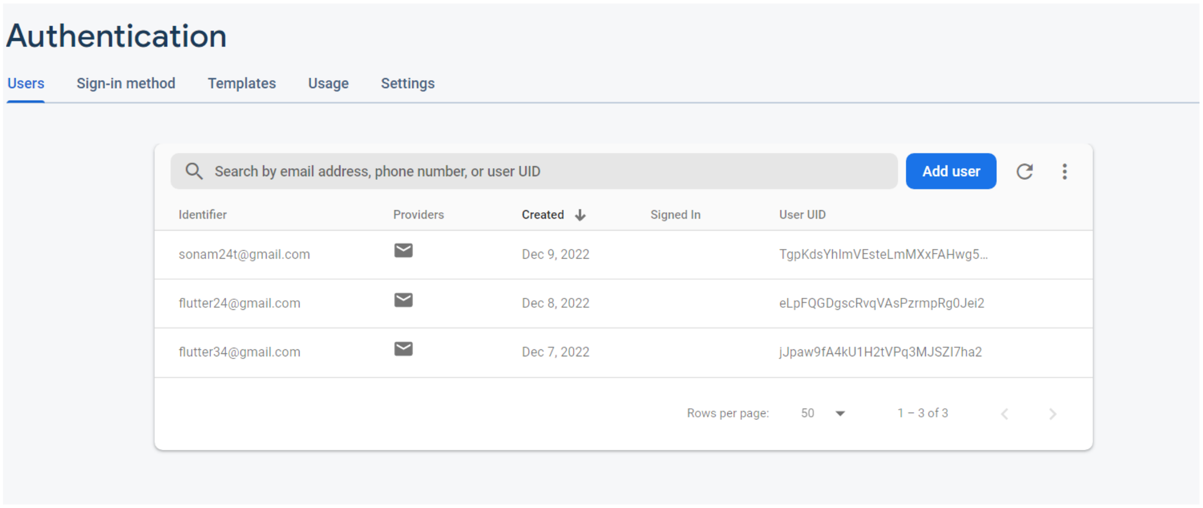
Web SDK configuration



Sign-in providers

Add new provider

Provider	Status
Email/Password	Enabled
Google	Enabled



Authentication

Users | Sign-in method | Templates | Usage | Settings

Search by email address, phone number, or user UID — Add user

Identifier	Providers	Created	Signed In	User UID
sonam24t@gmail.com	✉	Dec 9, 2022		TgpKdsYhlmVEsteLmMXxFAHwg5...
flutter24@gmail.com	✉	Dec 8, 2022		eLpFQGDgscRvqVAsPzrmpRg0Jei2
flutter34@gmail.com	✉	Dec 7, 2022		jJpaw9fA4kU1H2tVPq3MJSZI7ha2

Rows per page: 50 1 – 3 of 3

Result: Thus, an application that uses social networking software (Google) for authentication has been implemented.

MINI PROJECT

Team members:

Sonam T - 195002115

Thota Geethika Sree- 195002120

Title: EVENT BOOKING APP

Problem Description:

Upgrade your event and ticket booking business with a one-stop solution event bookings

App.

Procedure:

-Implement a front page for onboarding

- Display events in the next page with images and details about the event

-Implement a database using sqlite to store data of person booking for each event

Codes:

<u>Main.dart:</u>

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

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

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'GS App',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const SplashScreen(),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        body: Stack(
      children: <Widget>[
        Container(
          decoration: BoxDecoration(color: Color(0xff102733)),
        ),
        Container(
          padding: EdgeInsets.symmetric(horizontal: 50),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              const SizedBox(
                height: 18,
              ),
              Row(
                children: <Widget>[
                  const Text(
                    "Event Booking App",
                    style: TextStyle(
                        color: Color(0xffFFA700),
                        fontSize: 25,
                        fontWeight: FontWeight.w800),
                  )
                ],
              ),
              const SizedBox(
                height: 14,
              ),
              const Text(
                "An app by Sonam and Geethika!!",
                style:
                    TextStyle(color: Colors.white, fontWeight: FontWeight.w500),
              ),
              const SizedBox(
                height: 14,
              ),
              GestureDetector(
                onTap: () {
                  Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) => const HomeScreen()));
                },
                child: Container(
                  child: Row(
                    // ignore: prefer_const_literals_to_create_immutables
                    children: <Widget>[
                      Text(
                        "Get Started",
                        style: TextStyle(color: Colors.white, fontSize: 17),
                      ),
                      SizedBox(
                        width: 5,
                      ),
                      Icon(
                        Icons.arrow_forward,
                        color: Colors.white,
                      )
                    ],
                  ),
                ),
              )
            ],),),
      ],));
}}
```

home.dart:

```
// ignore_for_file: prefer_const_constructors
import 'package:flutter/material.dart';
import 'package:login/data/data.dart';
import 'package:login/models/event_model.dart';
import 'package:login/add.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);
  @override
  // ignore: library_private_types_in_public_api
  _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  List<EventsModel> events = <EventsModel>[];
```

```
String todayDateIs = "12";

@override
void initState() {
  // TODO: implement initState
  super.initState();

  events = getEvents();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      child: Stack(
        children: <Widget>[
          Container(
            decoration: BoxDecoration(color: Color(0xff102733)),
          ),
          SingleChildScrollView(
            child: Container(
              padding:
                  const EdgeInsets.symmetric(vertical: 60, horizontal: 30),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                  Row(
                    children: <Widget>[
                      Row(
                        // ignore: prefer_const_literals_to_create_immutables
                        children: const <Widget>[
                          Text(
                            "GS",
                            style: TextStyle(
                                color: Color(0xffFCCD00),
                                fontSize: 22,
                                fontWeight: FontWeight.w800),
                          )
                        ],), ],),
                  const SizedBox(
                    height: 20,
                  ),
                  Row(
                    children: <Widget>[
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: const <Widget>[
                          Text(
                            "Hello, there!",
                            style: TextStyle(
                                color: Colors.white,
                                fontWeight: FontWeight.w700,
                                fontSize: 21),
                          ),
                          SizedBox(
                            height: 6,
                          ),
                          Text(
                            "Let's explore what's happening nearby",
                            style:
                                TextStyle(color: Colors.white, fontSize: 15),
                          ) ], ),],  ),
                  const SizedBox(
                    height: 20,
                  ),

                  /// Popular Events

                  const Text(
                    "Popular Events",
                    style: TextStyle(color: Colors.white, fontSize: 20),
                  ),
                  GestureDetector(
                      //change here Sonam
                      onTap: () {
                        Navigator.push(
                            context,
                            MaterialPageRoute(
                                builder: (context) => const Home()));
                      },
                      child: Container(
                          child: PopularEventTile(
                        desc: events[0].desc,
                        imgeAssetPath: events[0].imgeAssetPath,
                        date: events[0].date,
                        address: events[0].address,
                      ))),
                  GestureDetector(
                      //change here Sonam
                      onTap: () {
                        Navigator.push(
                                      context,
                                      MaterialPageRoute(
                                          builder: (context) => const Home()));
                                },
                                child: Container(
                                    child: PopularEventTile(
                                  desc: events[1].desc,
                                  imgeAssetPath: events[1].imgeAssetPath,
                                  date: events[1].date,
                                  address: events[1].address,
                                ))),
                            GestureDetector(
                                onTap: () {
                                  Navigator.push(
                                      context,
                                      MaterialPageRoute(
                                          builder: (context) => const Home()));
                                },
                                child: Container(
                                    child: PopularEventTile(
                                  desc: events[2].desc,
                                  imgeAssetPath: events[2].imgeAssetPath,
                                  date: events[2].date,
                                  address: events[2].address,
                                ))),],),
                        ),  ), ], ),),);}}

                  class PopularEventTile extends StatelessWidget {
                    String desc;
                    String date;
                    String address;
                    String imgeAssetPath;

                    /// later can be changed with imgUrl
                    // ignore: use_key_in_widget_constructors
                    PopularEventTile(
                        {required this.address,
                        required this.date,
                        required this.imgeAssetPath,
                        required this.desc});

                    @override
                    Widget build(BuildContext context) {
                      return Container(
                        height: 100,
                        margin: EdgeInsets.only(bottom: 16),
                  decoration: BoxDecoration(
                      color: Color(0xff29404E), borderRadius: BorderRadius.circular(8)),
                  child: Row(
                    children: <Widget>[
                      Expanded(
                          child: Container(
                        padding: EdgeInsets.only(left: 16),
                        width: MediaQuery.of(context).size.width - 100,
                        child: Column(
                          mainAxisAlignment: MainAxisAlignment.center,
                          crossAxisAlignment: CrossAxisAlignment.start,
                          children: <Widget>[
                            Text(
                              desc,
                              style: TextStyle(color: Colors.white, fontSize: 18),
                            ),
                            const SizedBox(
                              height: 8,
                            ),
                            Row(
                              children: <Widget>[
                                Text(
                                  date,
                                  style: TextStyle(color: Colors.white, fontSize: 10),
                                )
                              ],
                            ),
                            SizedBox(
                              height: 4,
                            ),
                            Row(
                              children: <Widget>[
                                Text(
                                  address,
                                  style: TextStyle(color: Colors.white, fontSize: 10),
                                )
                              ],
                            ),
                          ],
                        ),
                      )),
                      ClipRRect(
                          borderRadius: BorderRadius.only(
                              topRight: Radius.circular(8),
                              bottomRight: Radius.circular(8)),
                          child: Image.asset(
                                  imgeAssetPath,
                                  height: 100,
                                  width: 120,
                                  fit: BoxFit.cover,
                                )),
                          ],),);}}
                  // ignore: non_constant_identifier_names
```

Add.dart:

```
// ignore_for_file: prefer_const_constructors
import 'package:login/db.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';

class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);
  @override
  State<StatefulWidget> createState() {
    return _HomeState();
  }
}

class _HomeState extends State<Home> {
  getdataview() {
    Future.delayed(Duration(milliseconds: 500), () async {
      slist =
          await mydb.db.rawQuery('SELECT * FROM students ORDER BY roll_no;');
      setState(() {
        view = 1;
      });
    });
  }

  TextEditingController name = TextEditingController();
  TextEditingController rollno = TextEditingController();
  List<Map> slist = [];
  MyDb mydb = MyDb();
  int view = 0;
  @override
  void initState() {
    mydb.open();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
```

```
return Scaffold(
  appBar: AppBar(
    title: Text(
      "Book events",
      textAlign: TextAlign.center,
    ),
    leading: view == 1
      ? BackButton(
          color: Colors.white,
          onPressed: () {
            setState(() {
              view = 0;
            });
          },
        )
      : null),
  body: Container(
    padding: EdgeInsets.all(30),
    child: view == 0
      ? Column(
          mainAxisAlignment: MainAxisAlignment.start,
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextField(
              controller: name,
              decoration: InputDecoration(
                hintText: "Name",
              ),
            ),
            TextField(
              keyboardType: TextInputType.number,
              inputFormatters: [
                FilteringTextInputFormatter.digitsOnly
              ],
              controller: rollno,
              decoration: InputDecoration(
                hintText: "Aadhar no",
              ),
            ),
            SizedBox(height: 50),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                ElevatedButton(
                    onPressed: () {
                      Future.delayed(Duration(milliseconds: 500),
                    () async {
                  var data = await mydb
                      .getStudent(int.parse(rollno.text));
                  if (data != null) {
                    ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(
                            content: Text(
                                "Member Already present with given aadar_no : " +
                                    rollno.text)));
                  } else {
                    mydb.db.rawInsert(
                        "INSERT INTO students (name, roll_no) VALUES (?, ?);",
                        [name.text, rollno.text]);
                    ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(
                            content: Text("New Member Added")));
                    name.text = "";
                    rollno.text = "";
                  }
                });
              },
              child: Text("Insert")),
          ElevatedButton(
              onPressed: () {
                Future.delayed(Duration(milliseconds: 500),
                    () async {
                  var data = await mydb
                      .getStudent(int.parse(rollno.text));
                  if (data == null) {
                    ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(
                            content: Text(
                                "No member found with aadar_no : " +
                                    rollno.text)));
                  } else {
                    mydb.db.rawDelete(
                        "DELETE FROM students where roll_no=?;",
                        [rollno.text]);
//mydb.db.rawDelete("DELETE * FROM students");
                    ScaffoldMessenger.of(context).showSnackBar(
                        SnackBar(
                            content: Text(
                                "Member Successfully removed")));
                    name.text = "";
                    rollno.text = "";
                  }
          });
        },
        child: Text("Delete")),
  ],
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    ElevatedButton(
        onPressed: () {
          Future.delayed(Duration(milliseconds: 500),
              () async {
            var data = await mydb
                .getStudent(int.parse(rollno.text));
            if (data != null) {
              mydb.db.rawInsert(
                  "UPDATE students SET name = ?, roll_no = ? WHERE roll_no = ?",
                  [name.text, rollno.text, rollno.text]);
              ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(
                      content:
                          Text("Members Data Updated")));
              name.text = "";
              rollno.text = "";
            } else {
              ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(
                      content: Text(
                          "No one found with aadar no : " +
                              rollno.text)));
            }
          });
        },
        child: Text("Update")),
    ElevatedButton(
        onPressed: () {
          getdataview();
        },
        child: Text("View")),
  ],
),
Expanded(
    child: Align(
  alignment: Alignment.bottomLeft,
  child: Text(
    "**Deletion and updation happens only based on Aadhar number",
                        ),
                      )),
                    ],
                  )
                : SingleChildScrollView(
                    child: Container(
                      child: slist.isEmpty
                          ? Text("No one booked yet.")
                          : Column(
                              children: slist.map((stuone) {
                                return Card(
                                  child: ListTile(
                                    leading: Icon(Icons.people),
                                    title: Text(stuone["roll_no"].toString()),
                                    subtitle: Text("Name:" + stuone["name"]),
                                  ),
                                );
                              }).toList(),
                            ),
                    ),
                  )));}}
```

Database stored here:

<u>Db.dart:</u>

```
import 'package:path/path.dart';
import 'package:sqflite/sqflite.dart';

class MyDb {
  late Database db;
  Future open() async {
// Get a location using getDatabasesPath
    var databasesPath = await getDatabasesPath();
    String path = join(databasesPath, 'demo.db');
//join is from path package
//output /data/user/0/com.dbapp.flutter.dbapp/databases/demo.db
    db = await openDatabase(path, version: 1,
        onCreate: (Database db, int version) async {
// When creating the db, create the table
      await db.execute('''
CREATE TABLE IF NOT EXISTS students(
id primary key,
name varchar(255) not null,
roll_no int unique not null

age int  not null
);
//create more table here

''');
//table students will be created if there is no table 'students'
    });
  }

  Future<Map<dynamic, dynamic>?> getStudent(int rollno) async {
    List<Map> maps =
        await db.query('students', where: 'roll_no = ?', whereArgs: [rollno]);
//getting student data with roll no.
    if (maps.isNotEmpty) {
      return maps.first;
    }
    return null;
  }
}
```

Data:

Data.dart:

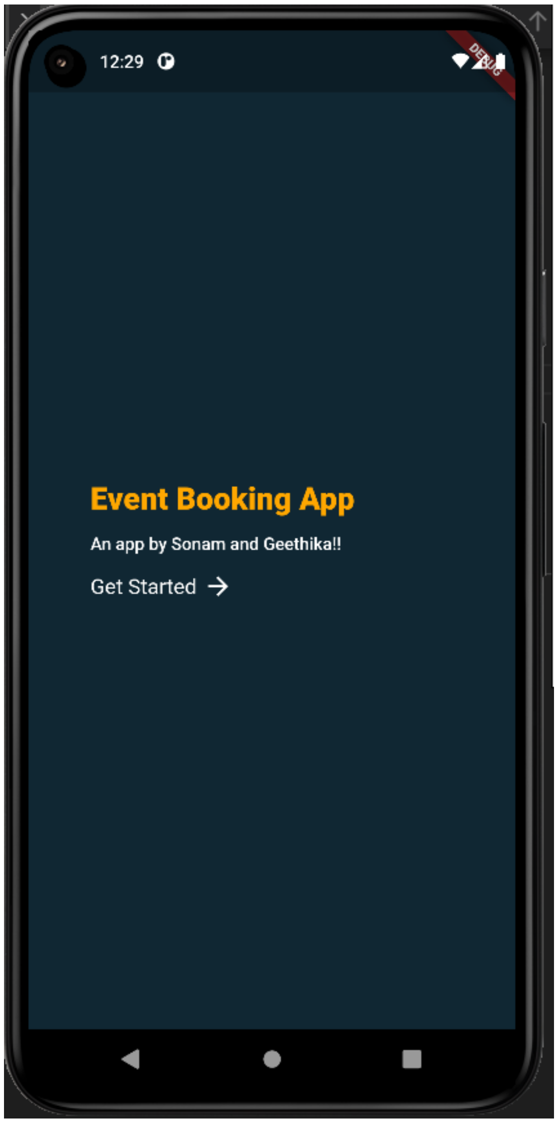
```
import 'dart:math';

import 'package:login/models/event_model.dart';

List<EventsModel> getEvents() {
  // ignore: prefer_collection_literals, deprecated_member_use
  List<EventsModel> events = <EventsModel>[];
  EventsModel eventsModel = new EventsModel();
  //1
  eventsModel.imgeAssetPath = "assets/tileimg.png";
  eventsModel.date = "Jan 05, 2023";
  eventsModel.desc = "Sports Meet in Galaxy Field";
  eventsModel.address = "Greenfields,  Bangalore";

  events.add(eventsModel);

  eventsModel = new EventsModel();

  //2
  eventsModel.imgeAssetPath = "assets/second.png";
  eventsModel.date = "Jan 24, 2023";
  eventsModel.desc = "Art & Meet in Socials";
  eventsModel.address = "Express Avenue,Adayar,Chennai";
  events.add(eventsModel);

  eventsModel = new EventsModel();
  //3
  eventsModel.imgeAssetPath = "assets/music_event.png";
  eventsModel.date = "Jan 12, 2023";
  eventsModel.address = "Phoenix Mall,Velacherry, Chennai";
  eventsModel.desc = "Youth Music Event";
  events.add(eventsModel);

  eventsModel = new EventsModel();

  return events;
}
```

Model:

event_model.dart:

```
class EventsModel {
  late String desc;
  late String date;
  late String address;
  late String imgeAssetPath;
}
```

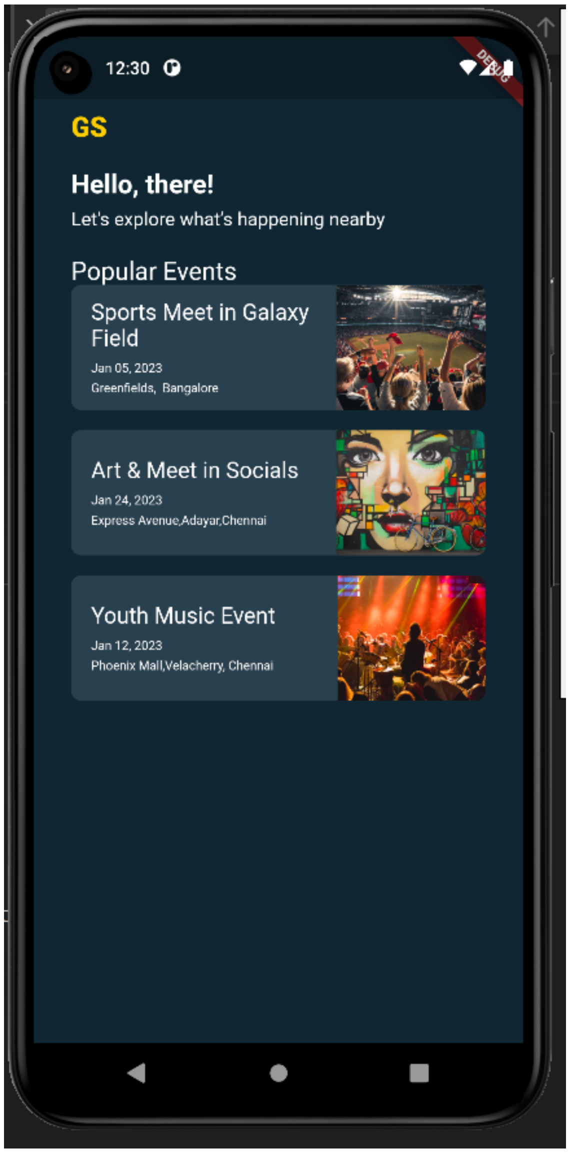
Output:


12:29

Event Booking App

An app by Sonam and Geethika!!

Get Started →


12:30

GS

Hello, there!

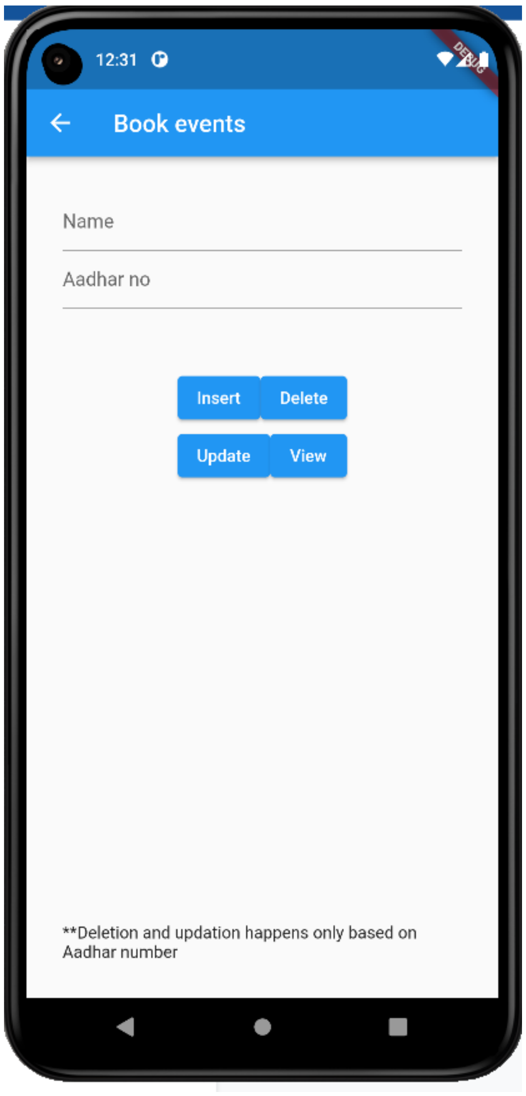
Let's explore what's happening nearby

Popular Events

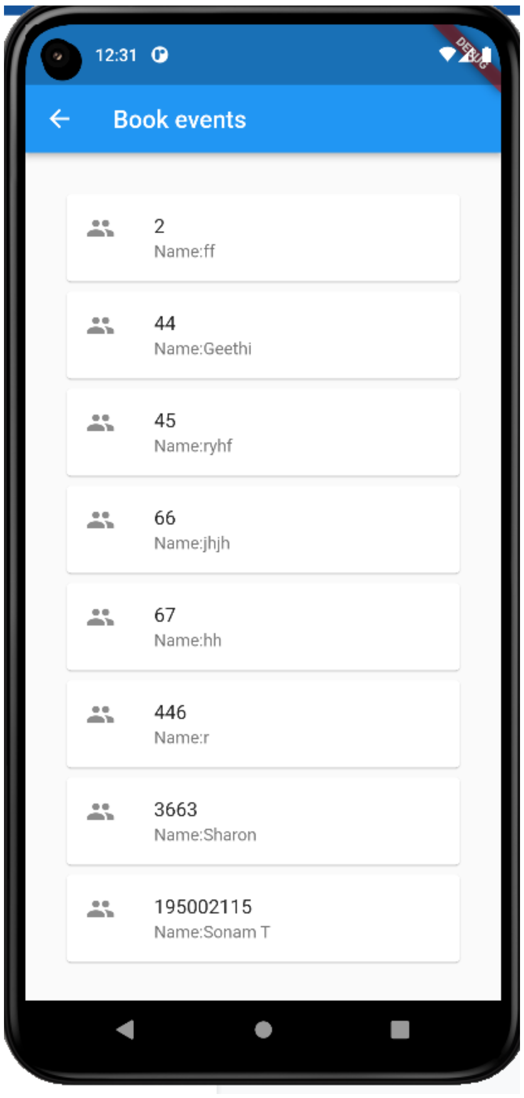
Sports Meet in Galaxy Field
Jan 05, 2023
Greenfields, Bangalore

Art & Meet in Socials
Jan 24, 2023
Express Avenue,Adayar,Chennai

Youth Music Event
Jan 12, 2023
Phoenix Mall,Velacherry, Chennai



**Deletion and updation happens only based on Aadhar number



2
Name:ff

44
Name:Geethi

45
Name:ryhf

66
Name:jhjh

67
Name:hh

446
Name:r

3663
Name:Sharon

195002115
Name:Sonam T

Result: The application has been executed successfully and output verified