An Application that uses GUI components, Fonts, Colors

Ex.No:1 **Date: 30/08/2022**

Aim:

To create a mobile application that uses GUI components, fonts, and colors.

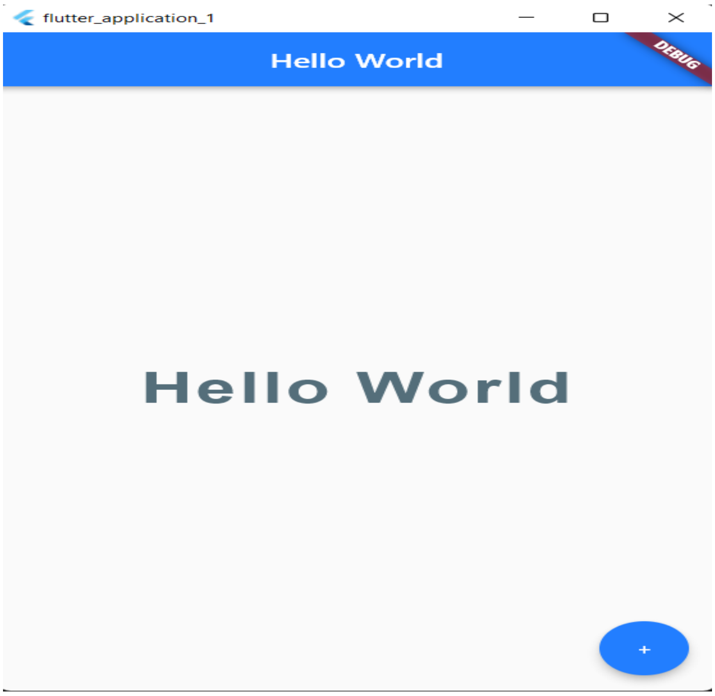
Procedure:

- 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, withing quotes inside Text().
  - Style property can be used to add TextStyle like fontSize, color.
  - textAlign property can be used for alignment of specified text

Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: Home(),
  ));}
class Home extends StatelessWidget {
  const Home({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
          title: Text("Hello World"),
          centerTitle: true,
          backgroundColor: Color.fromARGB(255, 34, 126, 255)),
      body: Center(
        child: Text(
          "Hello World",
style: TextStyle(
            fontSize: 45.0,
            fontWeight: FontWeight.bold,
            letterSpacing: 2.0,
            color: Colors.blueGrey[600],
            fontFamily: 'Arial',
          ),
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {},
        child: Text("+"),
        backgroundColor: Color.fromARGB(255, 34, 126, 255),
      ),);}}
```

Output:



Hello World

Hello World

Result:

A mobile application which uses GUI components, fonts, and colors has been implementedsuccessfully.

An application that uses Layout Managers and Event Listeners

Ex.No:2 **Date: 06/09/2022**

Aim:

To create a mobile application that uses Layout Managers and Event Listeners

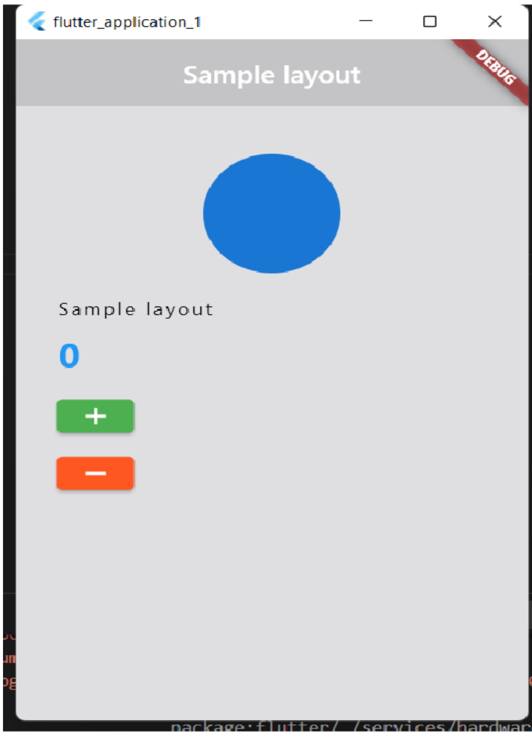
Procedure:

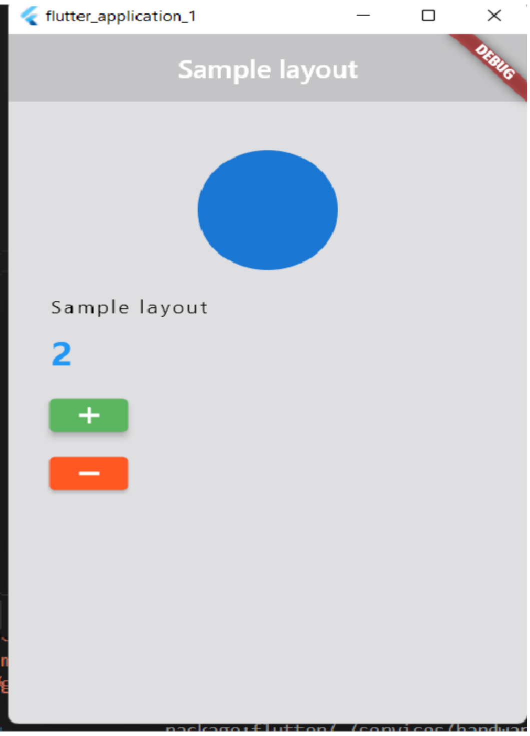
- Layout managers:
  - Column() class is used to display its children in a vertical way.
  - Children property is used to specify its descendants.
  - ListTile is a fixed-height row that typically contains some text as well asleading or trailing icon.
  - The icons (or other widgets) for the tile are defined withthe leading and trailing parameters.
- Event listeners:
  - onPressed() property is used to assign a callback function to the button oricon.
  - The application executes this function whenever the user presses taps thechip.
  - If onPressed() is null, then it denotes disabled.

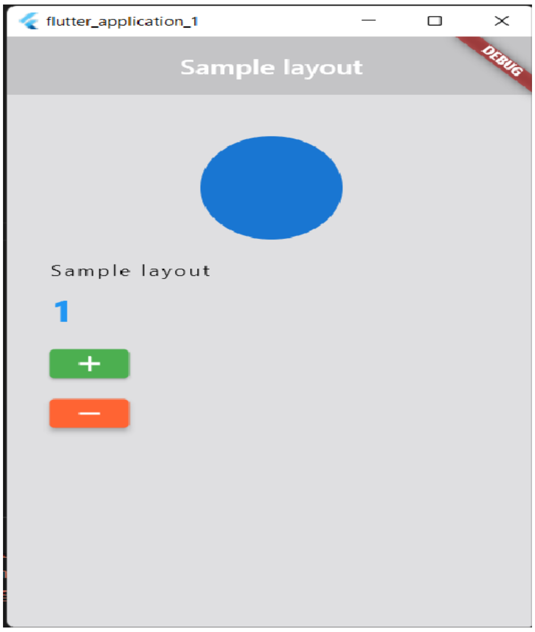
Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MaterialApp(
    home: Home(),
  ));
}
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);
  @override
  State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
  int projects = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color.fromARGB(255, 223, 223, 225),
appBar: AppBar(
        title: Text("Sample layout"),
        backgroundColor: Colors.black12,
        centerTitle: true,
        elevation: 0.0,
      ),
      body: Padding(
        padding: EdgeInsets.fromLTRB(30.0, 40.0, 30.0, 0.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Center(
              child: CircleAvatar(
                backgroundImage: AssetImage('assets/flutter.png'),
                radius: 50.0,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            Text(
              "Sample layout",
              style: TextStyle(
                color: Colors.black,
                letterSpacing: 2.0,
              ),
            ),
            SizedBox(
              height: 10.0,
            ),
            Text(
              "$projects",
              style: TextStyle(
                color: Colors.blue,
                letterSpacing: 2.0,
                fontSize: 28.0,
                fontWeight: FontWeight.bold,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            ElevatedButton(
              onPressed: () {
                setState(() {
                  projects++;
                });
              },
 onLongPress: () {
                setState(() {
                  projects *= 2;
                });
              },
              child: Icon(
                Icons.add,
              ),
              style: ElevatedButton.styleFrom(
                primary: Colors.green,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            ElevatedButton(
              onPressed: () {
                setState(() {
                  if (projects > 0) projects--;
                });
              },
              onLongPress: () {
                setState(() {
                  if (projects > 0) projects ~/= 2;
                });
              },
              child: Icon(
                Icons.remove,
              ),
              style: ElevatedButton.styleFrom(
                primary: Colors.deepOrange,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            Row(
              children: [
                SizedBox(
                  width: 20.0,
                ),],)],),),),);
              }
            }
```

Output:







Result:

An application that uses layout managers and event listeners has been implementedsuccessfully

Creation of Calculator Application

Ex.No:3 **Date: 13/09/2022**

Aim:

To create a mobile calculator application

Procedure:

- Initialize num1, num2 and res (result) as 0
- Declare a function for each of the basic arithmetic operations (+ , - , * , /) whichtakes 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.

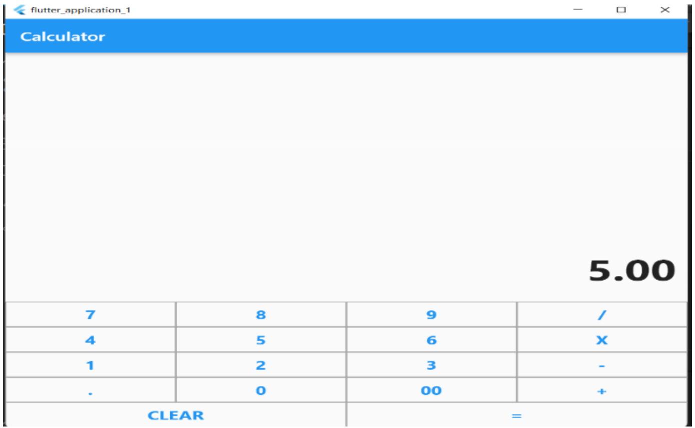
Code:

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Calculator',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      debugShowCheckedModeBanner: false,
      home: const MyHomePage(),
    );}}
class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key}) : super(key: key);
  @override
  _MyHomePageState createState() => _MyHomePageState();
}
```

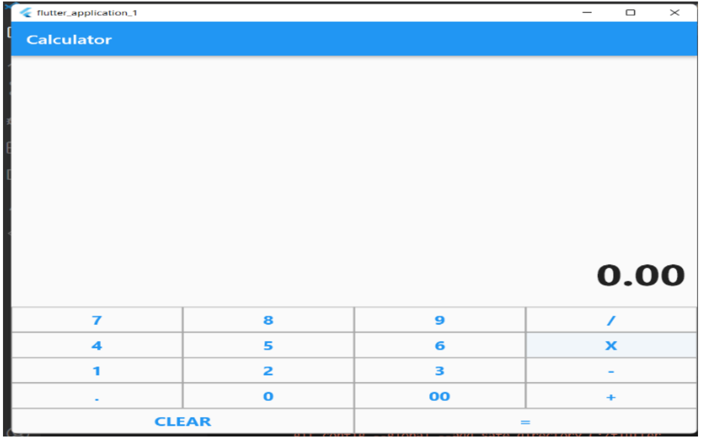
```
class _MyHomePageState extends State<MyHomePage> {
  String output = "0";

  String _output = "0";
  double num1 = 0.0;
  double num2 = 0.0;
  String operand = "";
  buttonPressed(String buttonText) {
    if (buttonText == "CLEAR") {
      _output = "0";
      num1 = 0.0;
      num2 = 0.0;
      operand = "";
    } else if (buttonText == "+" ||
        buttonText == "-" ||
        buttonText == "/" ||
        buttonText == "X") {
      num1 = double.parse(output);
      operand = buttonText;
      _output = "0";
    } else if (buttonText == ".") {
      if (_output.contains(".")) {
        return;
      } else {
        _output = _output + buttonText;
      }
    } else if (buttonText == "=") {
      num2 = double.parse(output);
      if (operand == "+") {
        _output = (num1 + num2).toString();}
      if (operand == "-") {
        _output = (num1 - num2).toString();}
      if (operand == "X") {
        _output = (num1 * num2).toString();}
      if (operand == "/") {
        _output = (num1 / num2).toString();}
      num1 = 0.0;
      num2 = 0.0;
      operand = "";
    } else {
      _output = _output + buttonText;
    }
    setState(() {
      output = double.parse(_output).toStringAsFixed(2);
    });
  }
```

```
buildButton(String buttonText) {
    return Expanded(
        child: OutlinedButton(
      style: OutlinedButton.styleFrom(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(0.0),
        ),

        side: const BorderSide(width: 1, color: Colors.grey),
        minimumSize: const Size.fromHeight(
            50.0), // Set this padding: EdgeInsets.zero, // and this
      ),
      child: Text(
        buttonText,
        style: const TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold),
      ),
      onPressed: () => buttonPressed(buttonText),
    ));}
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: const Text("Calculator"),
        ),
        body: Column(
          children: <Widget>[
            const Expanded(
              child: Divider(
                color: Colors.white,
              ),),
            Column(children: [
              Container(
                  alignment: Alignment.centerRight,
                  padding: const EdgeInsets.symmetric(
                      vertical: 24.0, horizontal: 12.0),
                  child: Text(output,
                      style: const TextStyle(
                        fontSize: 48.0,
                        fontWeight: FontWeight.bold,
                      ))),
              Row(children: [
                buildButton("7"),
                buildButton("8"),
                buildButton("9"),
                buildButton("/")
              ]),
Row(children: [
                buildButton("4"),
                buildButton("5"),
                buildButton("6"),
                buildButton("X")]),
              Row(children: [
                buildButton("1"),
                buildButton("2"),
                buildButton("3"),
                buildButton("-")]),
              Row(children: [
                buildButton("."),
                buildButton("0"),
                buildButton("00"),
                buildButton("+")]),
              Row(children: [
                buildButton("CLEAR"),
                buildButton("="),
              ])])],
        ));}}
```

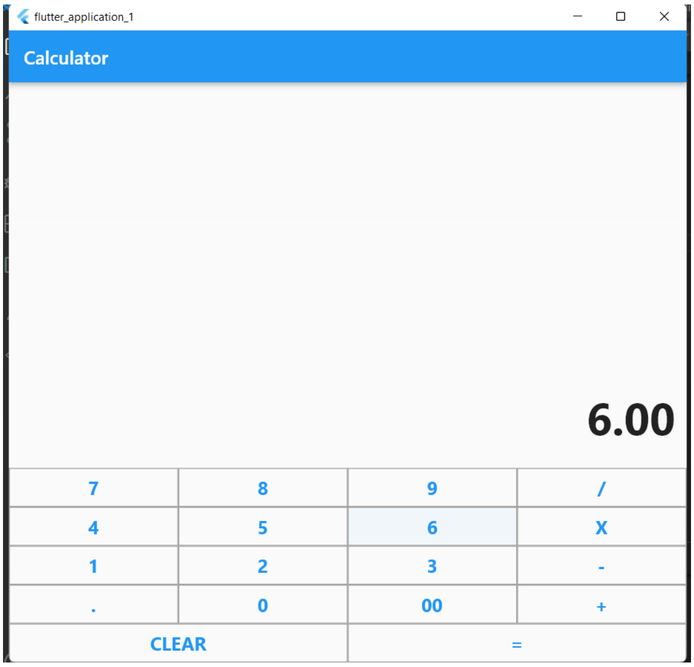
Output:



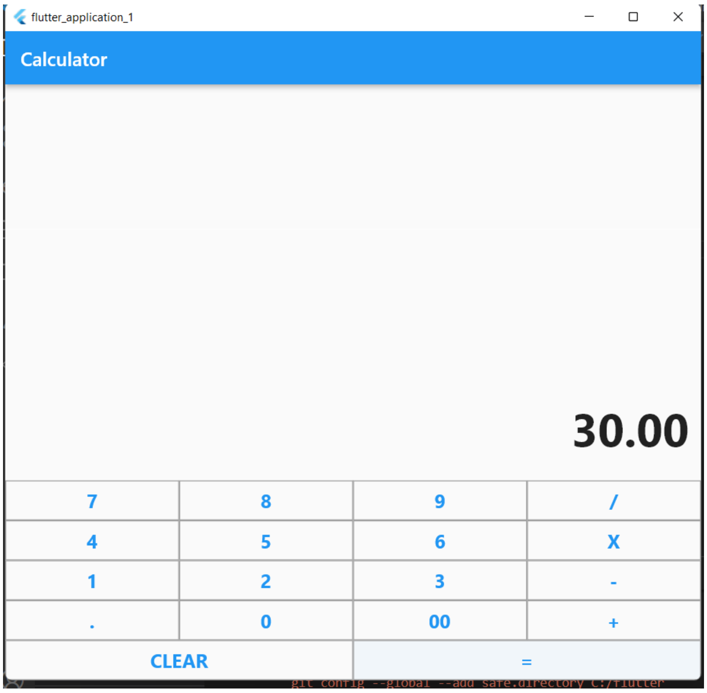
5.00



0.00



6.00



30.00

Result:

A calculator application for mobiles has been implemented successfully.

An application that draws basic graphical primitives on screen

Ex.No:4 **Date: 20/09/2022**

Aim:

To create a mobile application that draws basic graphical primitives on screen.

Procedure:

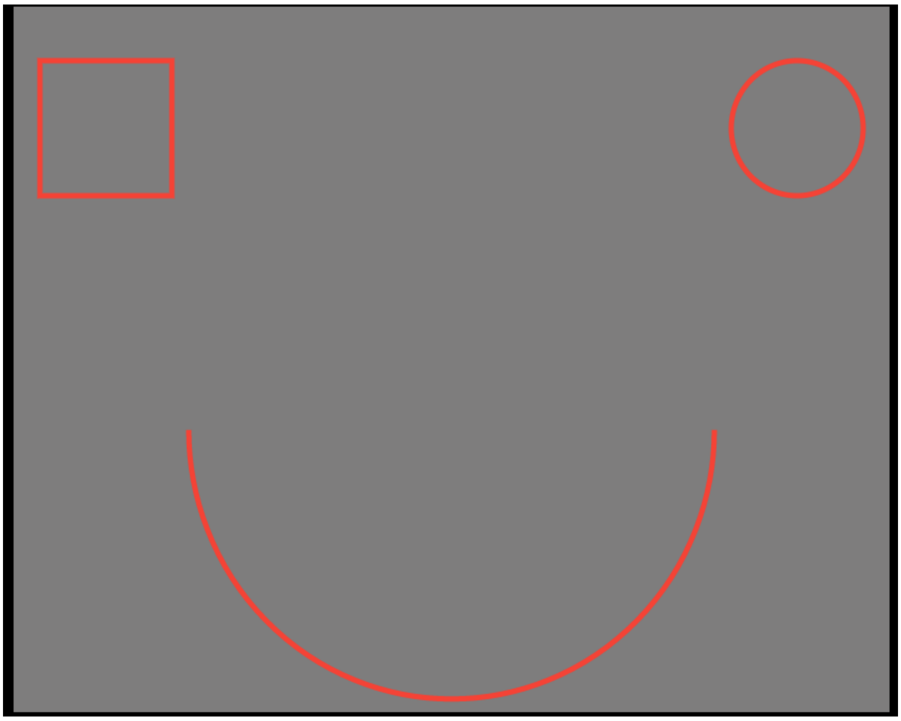
- Declare a class for each graphical primitive.
- The CustomPainter class is used.
- The paint method takes canvas and size as parameters.
- Create an instance of Paint() class.
- canvas.drawRect() is used to draw a rectangle.
- Similarly, for line drawLine() is used.
- For circle and arc, drawCircle()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

Code:

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

final Color darkBlue = Color.fromARGB(255, 18, 32, 47);
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark().copyWith(scaffoldBackgroundColor: darkBlue),
      debugShowCheckedModeBanner: false,
      home: Scaffold(
// Outer white container with padding
        body: Container(
          color: Colors.black,
          padding: EdgeInsets.symmetric(horizontal: 40, vertical: 80),
// Inner yellow container
          child: Container(
// pass double.infinity to prevent shrinking of the painter area to 0.
```

```
  width: double.infinity,
            height: double.infinity,
            color: Color.fromARGB(255, 126, 125, 125),
            child: CustomPaint(painter: FaceOutlinePainter()),
          ),
        ),
      ),
    );
  }
}

class FaceOutlinePainter extends CustomPainter {
  @override
  void paint(Canvas canvas, Size size) {
    final paint = Paint();
    paint.style = PaintingStyle.stroke;
    paint.strokeWidth = 4.0;
    paint.color = Color.fromARGB(255, 244, 67, 54);

    canvas.drawOval(
      Rect.fromLTWH(size.width - 120, 40, 100, 100),
      paint,
    );

    canvas.drawRect(
      Rect.fromLTWH(20, 40, 100, 100),
      paint,
    );

    final mouth = Path();
    mouth.moveTo(size.width * 0.8, size.height * 0.6);
    mouth.arcToPoint(
      Offset(size.width * 0.2, size.height * 0.6),
      radius: Radius.circular(150),
    );
    canvas.drawPath(mouth, paint);
  }

  bool shouldRepaint(FaceOutlinePainter oldDelegate) => false;
}
```

Output:



Result:

A mobile application that draws basic graphical primitives on screen has been implementedsuccessfully.

An application that uses database for persistent storage

Ex.No:5 **Date: 27/09/2022**

Aim:

To create a mobile application that draws basic graphical primitives on screen.

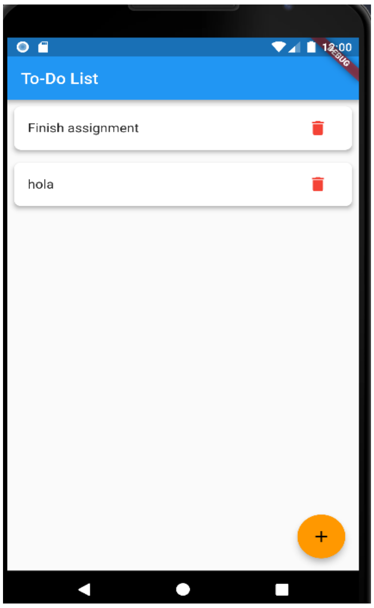
Procedure:

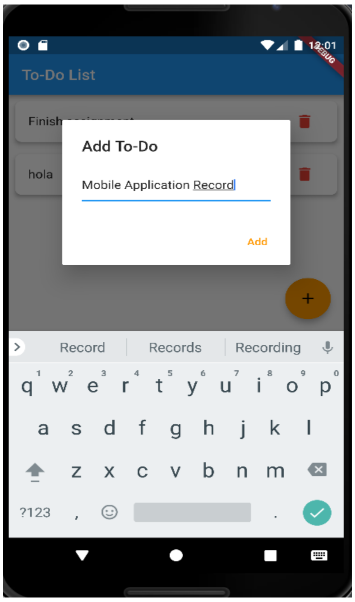
- Install the following packages:
    - npm install firebase-tools
    - flutter pub add firebase_core
    - flutter pub add firebase_auth
- Use ‘firebase login’ command to login to google account
- Use ‘flutterfire configure’ to add a firebase project to the application.
- Import the generated ‘firebase options’ file to main.dart file.
- FirebaseAuth.*instance*.currentUser is used to get the current user object
- Use FilePicker to select files from the device.
- storage.ref().child() is used to store the chosen file to Firebase storage.

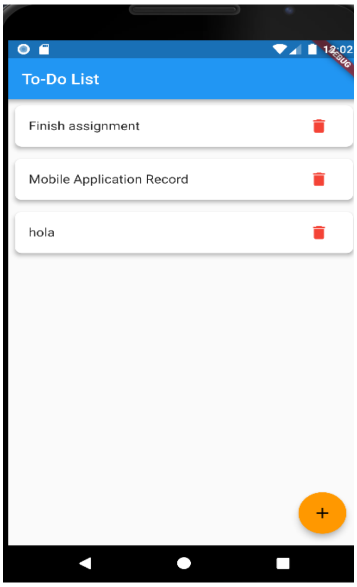
Code:

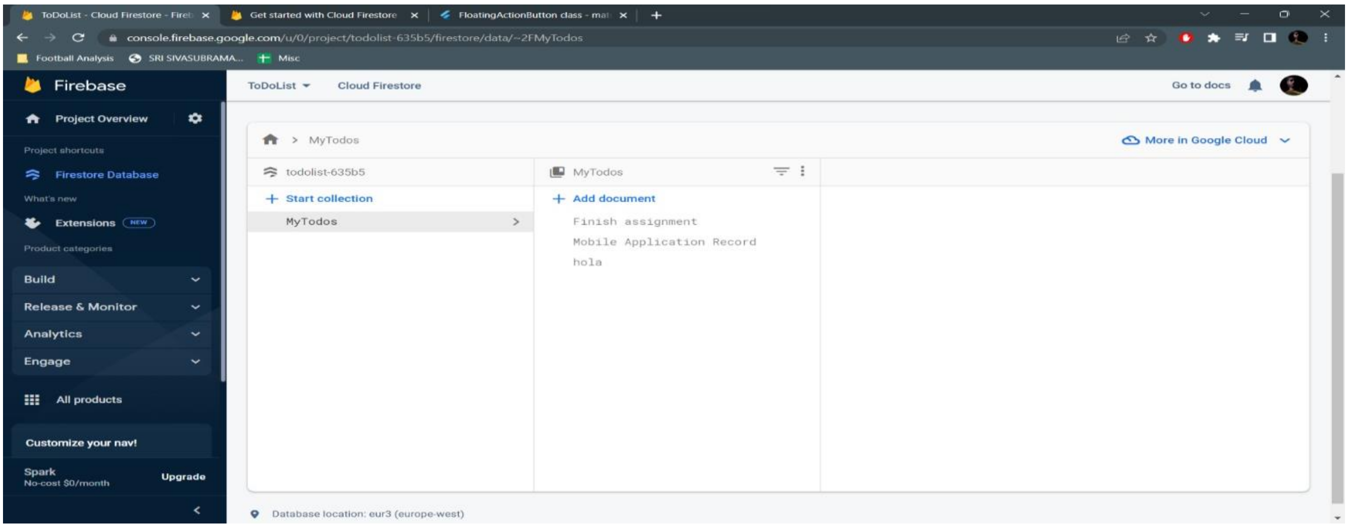
```
import 'package:flutter/material.dart';

import 'package:cloud_firestore/cloud_firestore.dart';
void main() => runApp(
      MaterialApp(

        theme:  ThemeData(
            brightness: Brightness.light,
            primaryColor: Colors.blue,
            accentColor: Colors.orange),
        home: MyApp(),

      ),

    );
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();

}
class _MyAppState extends State<MyApp> {
  List todos = List();
  String input = '';
  createTodos() {
    DocumentReference documentReference =
        Firestore.instance.collection('MyTodos').document(input);
    Map<String, String> todos = {'todoTitle': input};
    documentReference.setData(todos).whenComplete(() {

    print('$input created');
    });
  }
  deleteTodos() {}
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('To-Do List'),
        ),
        floatingActionButton: FloatingActionButton(
          child: Icon(Icons.add),
          onPressed: () {
            showDialog(
            context: context,
              builder: (BuildContext context) {
                return AlertDialog(
                  title: Text('Add To-Do'),
                  content: TextField(
                  onChanged: (String value) {
                      input = value;
                    },
                  ),
                  actions: <Widget>[
                    FlatButton(
                    onPressed: () {
                        createTodos();
                        Navigator.of(context).pop();
                      },
                      child: Text('Add'),
                    ),
                  ],
                );
              },
            );
          },
        ),
        body: StreamBuilder(

            stream: Firestore.instance.collection('MyTodos').snapshots(),
            builder: (context, snapshots) {
              return ListView.builder(
                shrinkWrap: true,
                itemCount: snapshots.data.documents.length,
                itemBuilder: (BuildContext context, int index) {
                DocumentSnapshot documentSnapshot =
                      snapshots.data.documents[index];
                  return Dismissible(
                      key: Key(index.toString()),
                      child: Card(
                        elevation: 4.0,

                        margin: EdgeInsets.all(8.0),
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(8),

                        ),

                        child: ListTile(

                          title: Text(documentSnapshot['todoTitle']),
                          trailing: IconButton(
                              icon: Icon(
                                Icons.delete,
                                color: Colors.red,
                              ),

                              onPressed: () {

                                setState(() {
                                  todos.removeAt(index);
                         });

                     }),
                   ),

                ));

              },);

            }));
```

Output:









Result:

A mobile application that draws uses databases has been implemented successfully.

An application that makes use of RSS feed

Ex.No:6 **Date:11/10/2022**

Aim:

To create a mobile application that uses RSS feed.

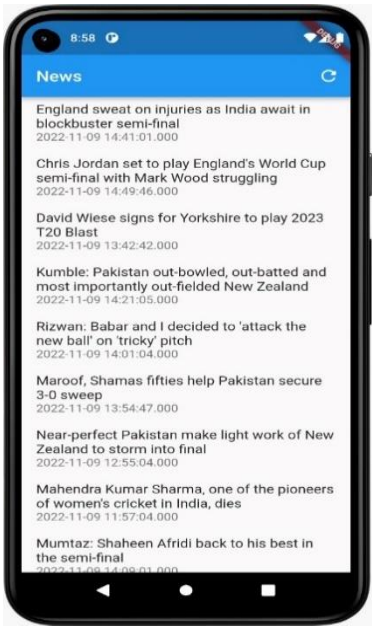
Procedure:

- 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:
main.dart

```
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:



News

England sweat on injuries as India await in blockbuster semi-final
2022-11-09 14:41:01.000

Chris Jordan set to play England's World Cup semi-final with Mark Wood struggling
2022-11-09 14:49:46.000

David Wiese signs for Yorkshire to play 2023 T20 Blast
2022-11-09 13:42:42.000

Kumble: Pakistan out-bowled, out-batted and most importantly out-fielded New Zealand
2022-11-09 14:21:05.000

Rizwan: Babar and I decided to 'attack the new ball' on 'tricky' pitch
2022-11-09 14:01:04.000

Maroof, Shamas fifties help Pakistan secure 3-0 sweep
2022-11-09 13:54:47.000

Near-perfect Pakistan make light work of New Zealand to storm into final
2022-11-09 12:55:04.000

Mahendra Kumar Sharma, one of the pioneers of women's cricket in India, dies
2022-11-09 11:57:04.000

Mumtaz: Shaheen Afridi back to his best in the semi-final
2022-11-09 14:09:01.000

Result:

RSS feed has been successfully integrated with the mobile app.

An application that implements multithreading

Ex.No:7 **Date:18/10/2022**

Aim:

To create a mobile application that implements multithreading.

Procedure:

- Install the following packages:
  - npm install firebase-tools
  - flutter pub add firebase_core
  - flutter pub add firebase_auth
- Use ‘firebase login’ command to login to google account
- Use ‘flutterfire configure’ to add a firebase project to the application.
- Import the generated ‘firebase options’ file to main.dart file.
- FirebaseAuth.*instance*.currentUser is used to get the current user object
- Use FilePicker to select files from the device.
- storage.ref().child() is used to store the chosen file to Firebase storage.
- ‘async’ enables your program to start a potentially long-running task and still be ableto be responsive to other events while that task runs, rather than having to wait until that task has finished.
- ‘await’ keyword is used before a call to a function that returns a promise. This makesthe code wait at that point until the promise is settled, at which point the fulfilled value of the promise is treated as a return value, or the rejected value is thrown.

Code:

main.dart

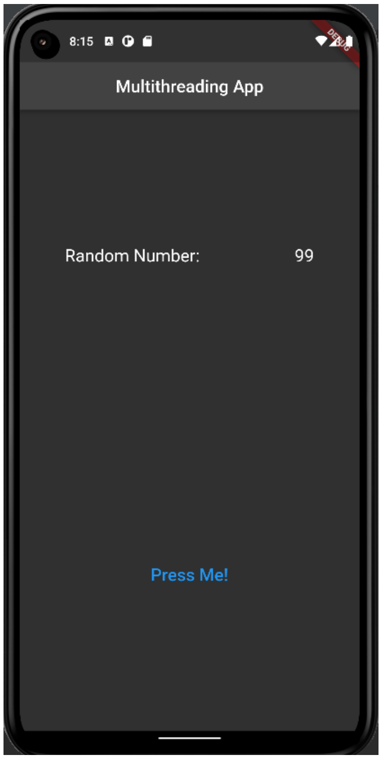
```
import 'package:expt7/pages/home.dart';
import 'package:flutter/material.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        brightness: Brightness.dark,
      ),
      home: const Home(),
    );
  }
}
```

home.dart

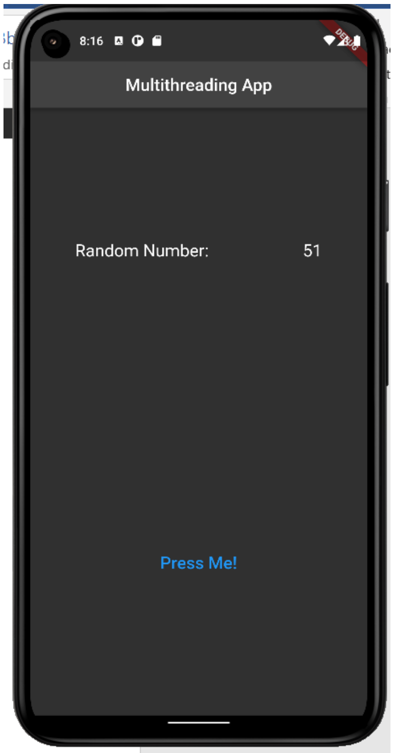
```
import 'dart:async';
import 'dart:math';

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

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:



Multithreading App

Random Number: 99



Multithreading App

Random Number: 51

Result:

An android application that implements multithreading has been developed and executed successfully.

An application that uses GPS location information

Ex.No:8 **Date: 01/11/2022**

Aim:

To create a mobile application that uses GPS location information.

Procedure:

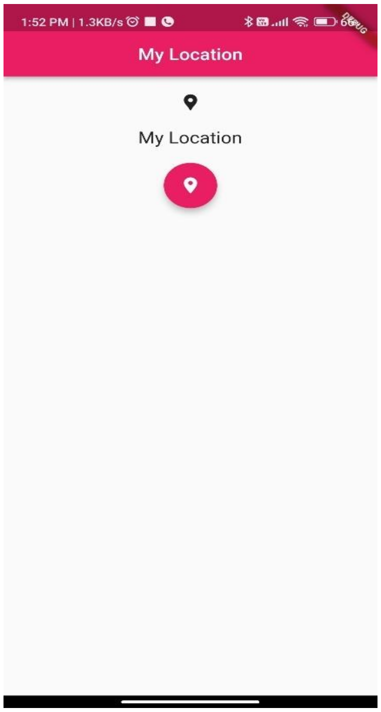
- Install the following packages: geolocator & geocoding
- Import them using,
  - import 'package:geocoding/geocoding.dart';
  - import 'package:geolocator/geolocator.dart';
- Get current location of the device, by creating an instance of Geolocator and callinggetCurrentPosition.
- Convert latitude and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

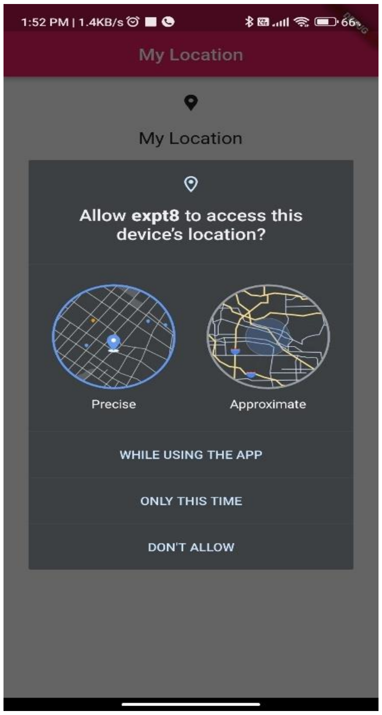
Code:

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

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: const Home(),
    );
  }}
```

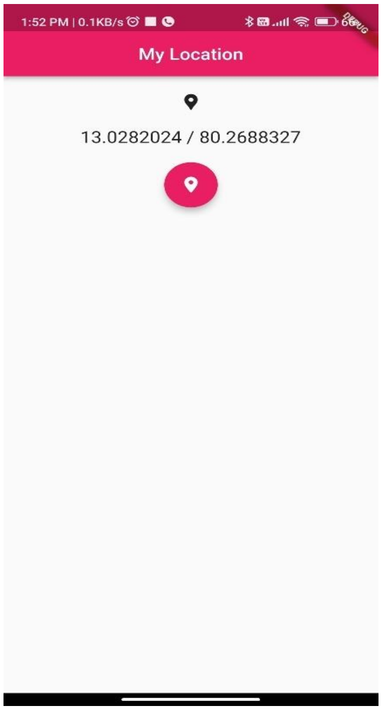
```
class Home extends StatelessWidget {
  const Home({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          "My Location"
        ),
        centerTitle: true,
      ),
      body: const LocationInfo(

      ),
      floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
    );
  }
}

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

  @override
  State<LocationInfo> createState() => _LocationInfoState();
}

class _LocationInfoState extends State<LocationInfo> {
  String _myLoc ="My Location";
  Location location=new Location();
  late bool _serviceEnabled;
  late PermissionStatus _permissionGranted;
  late LocationData _locationData;
  bool _isListenLocation =false, _isGetLocation = false;
  @override
  Widget build(BuildContext context) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: <Widget>[
        const SizedBox(
          height: 20.0,
        ),
        const Icon(
          Icons.location_pin,
        ),
        const SizedBox(
          height: 20.0,
        ),
  Center(
          child: Text(
            "$_myLoc",
            style: TextStyle(
              fontSize: 20.0,
            ),
          ),
        ),
        const SizedBox(
          height: 20.0,
        ),
        FloatingActionButton(
            child: Icon(
              Icons.location_on_sharp,
            ),
            onPressed: updateLoc,
        ),
      ],
    );
  }
  void updateLoc() async{
    _serviceEnabled = await location.serviceEnabled();
    if(!_serviceEnabled){
      _serviceEnabled = await location.requestService();
      if(_serviceEnabled)
        return;
    }
    _permissionGranted = await location.hasPermission();
    if(_permissionGranted == PermissionStatus.denied){
      _permissionGranted = await location.requestPermission();
      if(_permissionGranted != PermissionStatus.granted)
        return;
    }
    _locationData = await location.getLocation();
    setState(() {
      _isGetLocation = true;
    });
    if(_isGetLocation){
      _myLoc="${_locationData.latitude} / ${_locationData.longitude}";
    }
  }
}
```

Output:





Allow **expt8** to access this device's location?



```
13.0282024 / 80.2688327
```

Result:

A native application that uses GPS location has been developed and executed successfully.

An application that takes advantage of rich gesture-based UI handling

Ex.No:9 **Date: 08/11/2022**

Aim:

To create a mobile application that will take advantage of underlying phone functionality including rich gesture-based UI handling

Procedure:

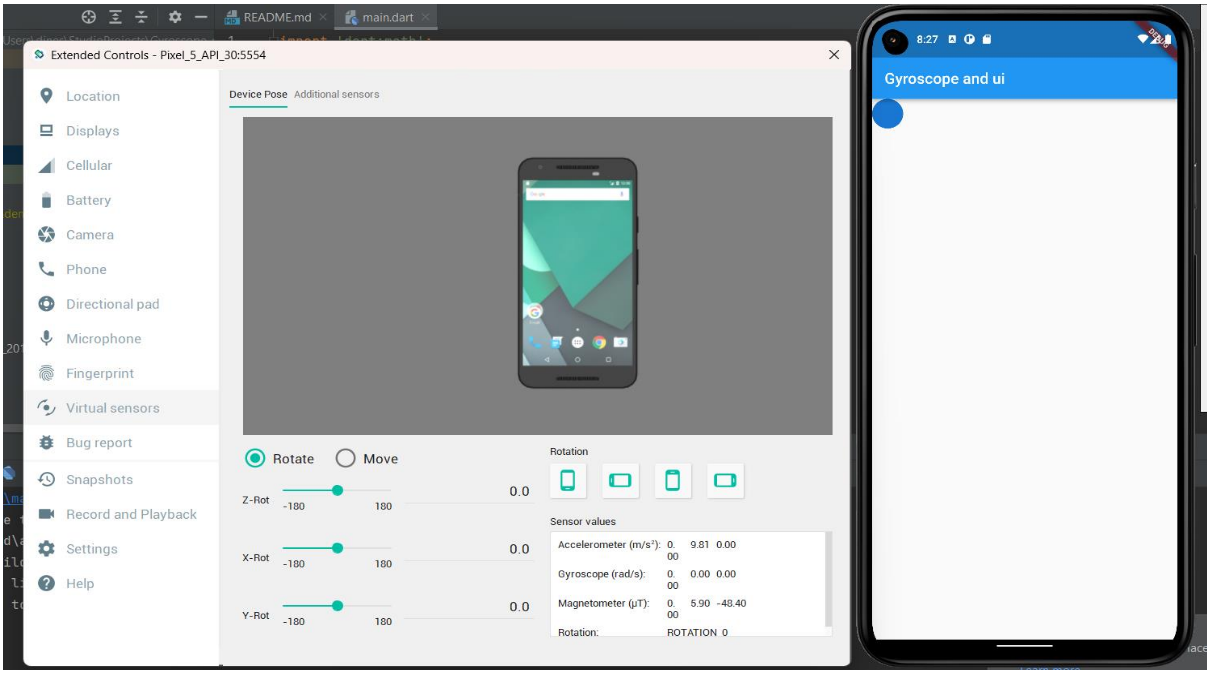
- Install path_provider package
- The path where is file is to be written is obtained using getExternalStorageDirectory() function.
- writeAsString(<String>) is used to write contents into a text file.
- readAsString() is used to read the contents of the file.

Code:

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

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
// "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Gyroscope and ui'),
    );
  }
}

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

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  double _dx = 0,
      _dy = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: StreamBuilder<GyroscopeEvent>(
        stream: SensorsPlatform.instance.gyroscopeEvents,
        builder: (context, snapshot) {
```

```
if (snapshot.hasData) {
          _dy = _dy + snapshot.data!.y * 10;
          _dx = _dx + snapshot.data!.x * 10;
        }
        return Stack(
          children: [
            Positioned(
              top: _dy,
              left: _dx,
              child: GestureDetector(
                onPanUpdate: (details) {
                  setState(() {
                    _dy = max(0, _dy + details.delta.dy);
                    _dx = max(0, _dx + details.delta.dx);
                  });
                },
                child: const CircleAvatar(),
              ),
            )
          ],
        );
      },),);}}
```

Output:



● Rotate ○ Move

Z-Rot -180 180 0.0

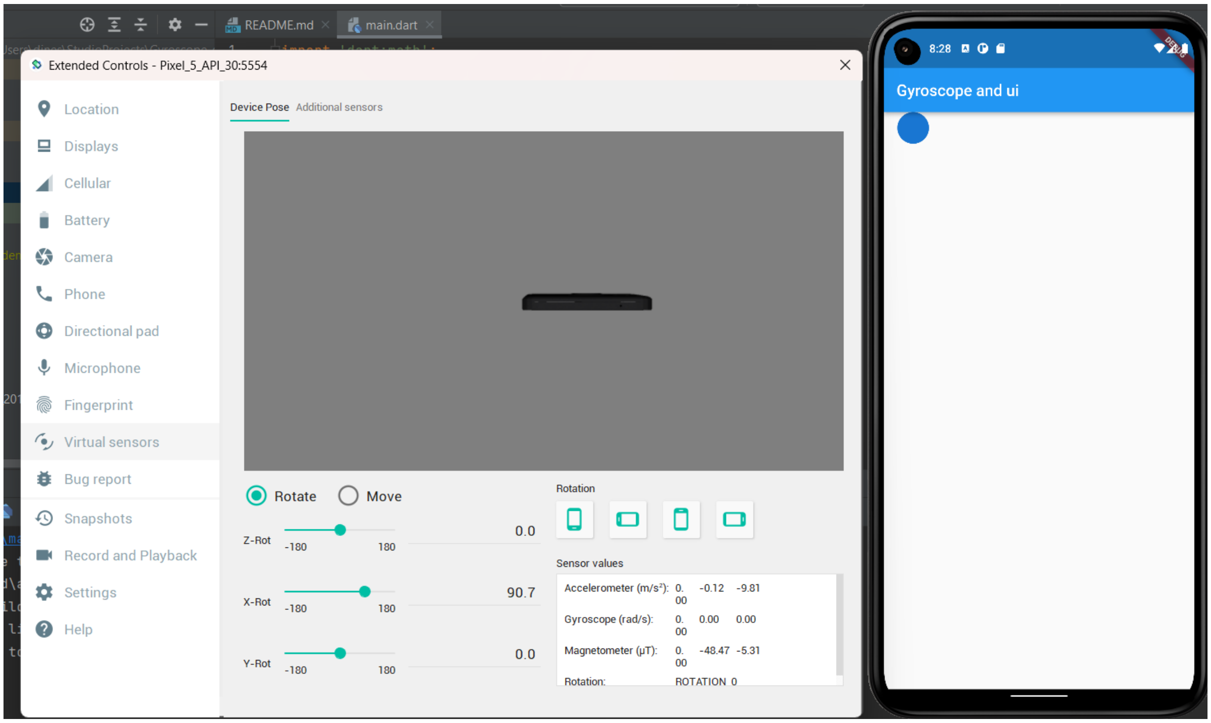
X-Rot -180 180 0.0

Y-Rot -180 180 0.0

Rotation

Sensor values

```
Accelerometer (m/s²):  0.    9.81  0.00
                       00
Gyroscope (rad/s):     0.    0.00  0.00
                       00
Magnetometer (μT):     0.    5.90  -48.40
                       00
Rotation:              ROTATION_0
```



◉ Rotate ○ Move

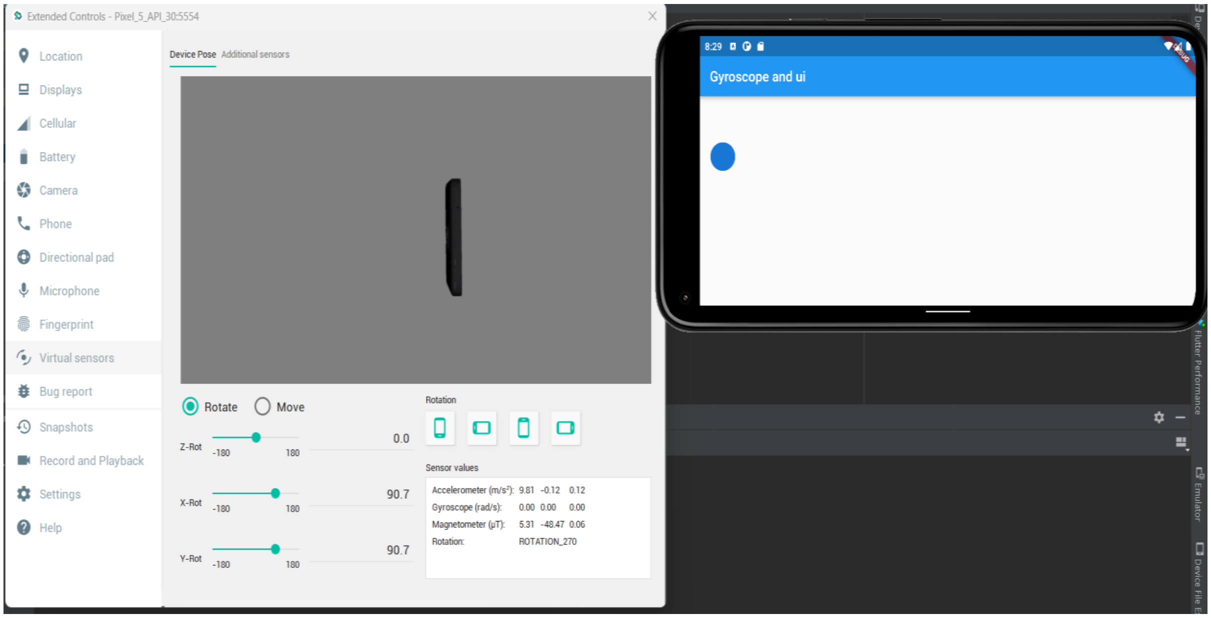
Z-Rot -180 180 0.0

X-Rot -180 180 90.7

Y-Rot -180 180 0.0

Sensor values

Accelerometer (m/s²): 0.00 -0.12 -9.81
Gyroscope (rad/s): 0.00 0.00 0.00
Magnetometer (μT): 0.00 -48.47 -5.31
Rotation: ROTATION_0



◉ Rotate ○ Move

Z-Rot -180 180 0.0

X-Rot -180 180 90.7

Y-Rot -180 180 90.7

Sensor values

Accelerometer (m/s²): 9.81 -0.12 0.12
Gyroscope (rad/s): 0.00 0.00 0.00
Magnetometer (μT): 5.31 -48.47 0.06
Rotation: ROTATION_270

Result:

A mobile application that uses rich gestures to handle UI was developed and executed successfully.

An application that creates an alert upon user action

Ex.No:10 **Date: 15/11/2022**

Aim:

To create an application that sends an alert upon user action.

Procedure:

- On the To-do list page, create a TextButton labelled 'ADD' to add a new task.
- In the onPressed() property, use showDialog to specify the alert box contents.
- AlertDialog() is used to create the alert message box.
    - The content property is used to specify the message using Text(). In thiscase, the message displayed is "Task added".
    - The action property is used to specify the buttons in the alert box usingTextButton().

Code:

main.dart

```
import 'package:expt10/pages/home.dart';
import 'package:flutter/material.dart';

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Experiment 10',
      theme: ThemeData.dark(),
      home: const Home(),
    );
  }
}
```

home.dart

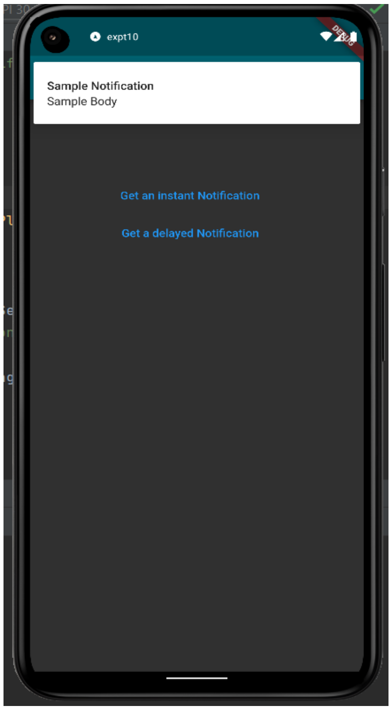
```
import 'package:expt10/services/local_notification_service.dart';
import 'package:flutter/material.dart';
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  late final LocalNotificationService service;
  @override
  void initState(){
    service = LocalNotificationService();
    service.initialize();
    super.initState();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          "Local Notifications Expt"
        ),
        backgroundColor: const Color(0xff006473),
        centerTitle: true,
      ),
      body: Padding(
        padding: EdgeInsets.all(MediaQuery.of(context).size.width*0.25),
        child: Column(
          children: <Widget>[
            TextButton(
                onPressed: () async {
                  await service.showNotification(
                      id: 0,
                      title: "Sample Notification",
                      body: "Sample Body"
                  );
                },
                child: const Text(
                  "Get an instant Notification"
                ),
            ),
            TextButton(
              onPressed: () async {
```

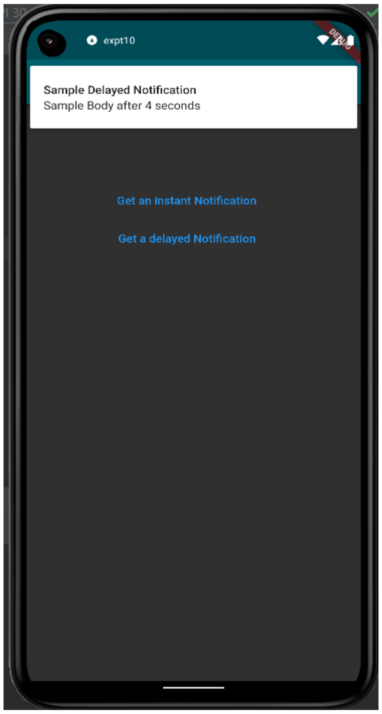
```
await service.showScheduledNotification(
                id: 0,
                title: "Sample Notification",
                body: "Sample Body",
                seconds: 4,
             );
            },
            child: const Text(
               "Get a delayed Notification"
            ),),],),),);
  }
}
```

local_notification_service.dart

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:timezone/timezone.dart' as tz;
import 'package:timezone/data/latest.dart' as tz;

class LocalNotificationService {
  LocalNotificationService();

  final _localNotificationService = FlutterLocalNotificationsPlugin();

  Future<void> initialize() async{
    tz.initializeTimeZones();
    const AndroidInitializationSettings androidInitializationSettings =
    AndroidInitializationSettings('ic_stat_assistant_navigation');

    const DarwinInitializationSettings iosInitializationSettings =
        DarwinInitializationSettings(
          requestAlertPermission: true,
          requestBadgePermission: true,
          requestSoundPermission: true,
        );
    const InitializationSettings settings = InitializationSettings(
        android: androidInitializationSettings,
        iOS: iosInitializationSettings
    );

    await _localNotificationService.initialize(settings);
  }
  Future<NotificationDetails> _notificationDetails() async{
    const AndroidNotificationDetails androidNotificationDetails = AndroidNotificationDetails(
 "channel_id", "channel_name",
      channelDescription: "Description",
      importance: Importance.max,
      priority: Priority.max,
      playSound: true,
    );
    const DarwinNotificationDetails darwinNotificationDetails = DarwinNotificationDetails();
    return const NotificationDetails(android: androidNotificationDetails,iOS:
darwinNotificationDetails);
  }
  Future<void> showNotification({
    required int id,
    required String title,
    required String body}) async{
      final details = await _notificationDetails();
      await _localNotificationService.show(id, title, body, details);
  }
  Future<void> showScheduledNotification({
    required int id,
    required String title,
    required String body,
    required int seconds
  }) async{
    final details = await _notificationDetails();
    await _localNotificationService.zonedSchedule(
      id,
      title,
      body,
      tz.TZDateTime.from(DateTime.now().add(Duration(seconds: seconds)), tz.local,),
      details,
      androidAllowWhileIdle: true,
      uiLocalNotificationDateInterpretation: UILocalNotificationDateInterpretation.absoluteTime
    );
  }
}
```

Output:


Sample Notification
Sample Body


Sample Delayed Notification
Sample Body after 4 seconds

Result:

An application that sends an alert upon user action was developed and executed successfully.

An application that creates an alarm clock

Ex.No:11 **Date: 22/11/2022**

Aim:

To create an 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 whenan alarm is set.
- The “Show Alarms” button, opens the clock application of the device which showsthe created alarms.

Code:

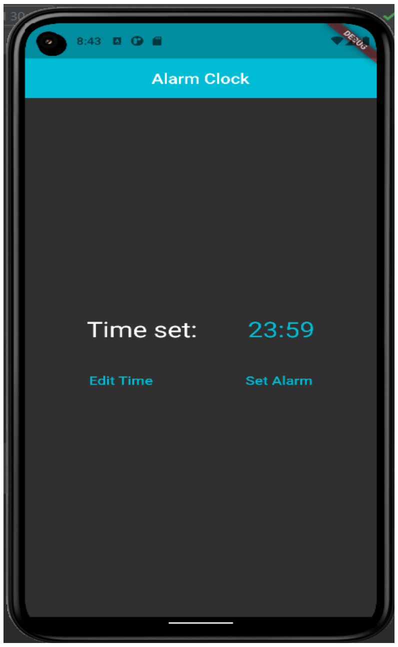
main.dart

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
  theme: ThemeData(
        primarySwatch: Colors.cyan,
        brightness: Brightness.dark,
      ),
      home: const Home(),
    );
  }
}
```

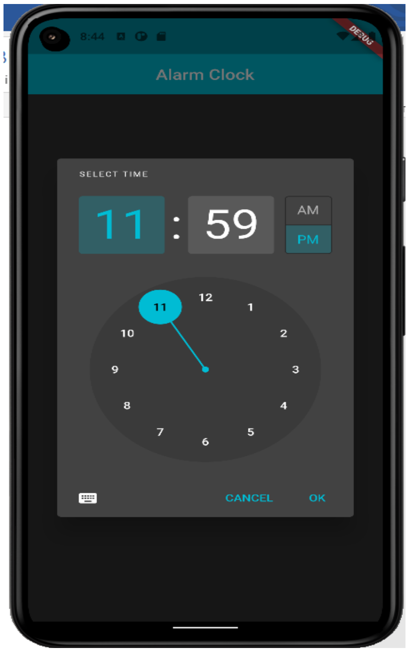
home.dart

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

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

class _HomeState extends State<Home> {
  TimeOfDay time= TimeOfDay(hour: 23, minute: 59);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Alarm Clock",
        ),
        centerTitle: true,
        elevation: 0.0,
        backgroundColor: Colors.cyan,
      ),
      body: Padding(
        padding: EdgeInsets.all(20),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
Text(
                    "Time set: ",
                    style: TextStyle(
                      fontSize: 30.0,
                    ),
                  ),
                  Text(
                    "${time.hour.toString().padLeft(2,'0')}:${time.minute.toString().padLeft(2,'0')}",
                    style: TextStyle(
                      fontSize: 30.0,
                      color: Colors.cyan,
                    ),
                  )
                ],
              ),
              SizedBox(
                height: 30.0,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [
                  TextButton(
                    onPressed: () async{
                      TimeOfDay? newTime = await showTimePicker(
                          context: context,
                          initialTime: time,
                      );
                      if(newTime == null) return;
                      setState(() {
                        time = newTime;
                      });
                    },
                    child: Text(
                      "Edit Time",
                      style: TextStyle(
                        fontSize: 17.0,
                      ),
                    ),
                  ),
                  TextButton(
                    onPressed: () {
                      FlutterAlarmClock.createAlarm(time.hour,time.minute);
                    },
child: Text(
                      "Set Alarm",
                      style: TextStyle(
                        fontSize: 17.0,
                      ),),),],),],),),),);
                    }
                  }
```

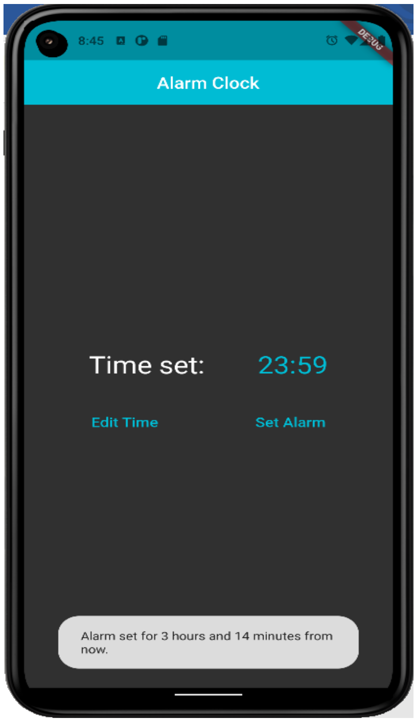
Output:



Time set: 23:59

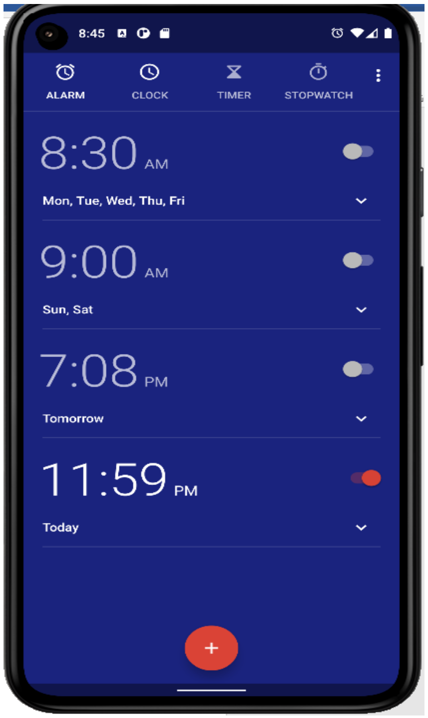


11 : 59 PM



Time set: 23:59

Alarm set for 3 hours and 14 minutes from now.



8:30 AM
Mon, Tue, Wed, Thu, Fri

9:00 AM
Sun, Sat

7:08 PM
Tomorrow

11:59 PM
Today

Result:
An application that creates an alarm clock is developed and tested successfully.

Simple Game With Multimedia Support

Ex.No:12 **Date:27/11/2022**

Aim:

To implement a simple gaming application with multimedia support.

Procedure:

- Create a class TileModel for each tile, which has the following as members
  - ImageAssetPath
  - 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.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();
    //3
    tileModel.setImageAssetPath("assets/horse.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();
    //4
    tileModel.setImageAssetPath("assets/monkey.pn
    g"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();
    //5
    tileModel.setImageAssetPath("assets/panda.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();
    //6
    tileModel.setImageAssetPath("assets/parrot.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();

    //7
    tileModel.setImageAssetPath("assets/rabbit.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();

    //8
    tileModel.setImageAssetPath("assets/zoo.pn
    g"); 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.p
    ng"); tileModel.setIsSelected(false);
    pairs.add(tileModel);
    pairs.add(tileModel);
    tileModel = new
    TileModel();
```

```
 //3
 tileModel.setImageAssetPath("assets/question.p
 ng"); tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new
 TileModel();
 //4
 tileModel.setImageAssetPath("assets/question.p
 ng"); tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new
 TileModel();
 //5
 tileModel.setImageAssetPath("assets/question.p
 ng"); tileModel.setIsSelected(false);
 pairs.add(tileModel);
 pairs.add(tileModel);
 tileModel = new
 TileModel();

 //6
 tileModel.setImageAssetPath("assets/question.p
 ng"); tileModel.setIsSelected(false);
 pairs.add(tileModel);

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

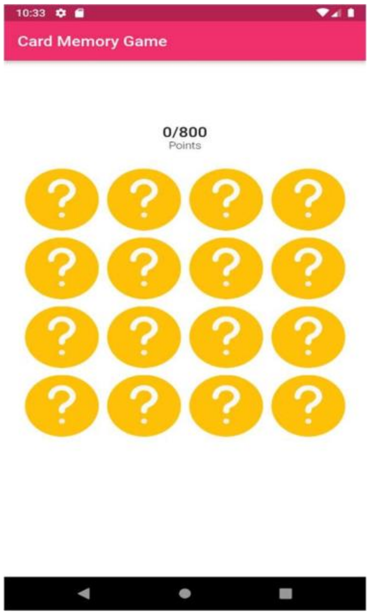
```
import 'dart:async';


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;
        });
      });
    }
```

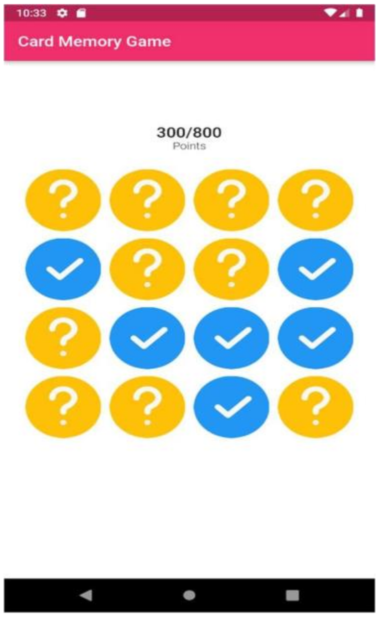
```
@override
Widget build(BuildContext context)
 {return Scaffold(
  appBar: AppBar(
   title: Text('Card Memory Game'),
   backgroundColor:Color(0xffef2e6c)  ,
  ),
  backgroundColor: Colors.white,
  body: SingleChildScrollView(
  child: Container(
    padding: EdgeInsets.symmetric(horizontal: 20, 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();
}
```

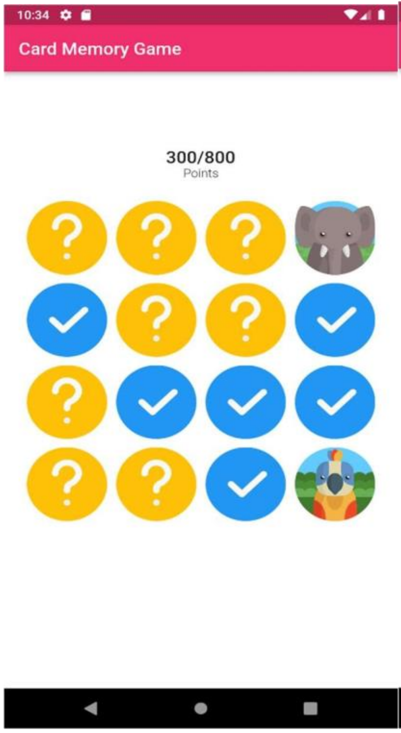
```
class _TileState extends
  State<Tile> {@override
  Widget build(BuildContext context)
    {return GestureDetector(
      onTap: () {
        if (!selected) {
          setState(() {
            myPairs[widget.tileIndex].setIsSelected(true);
          });
          if (selectedTile != "") {
            /// 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;
                });

              });
```

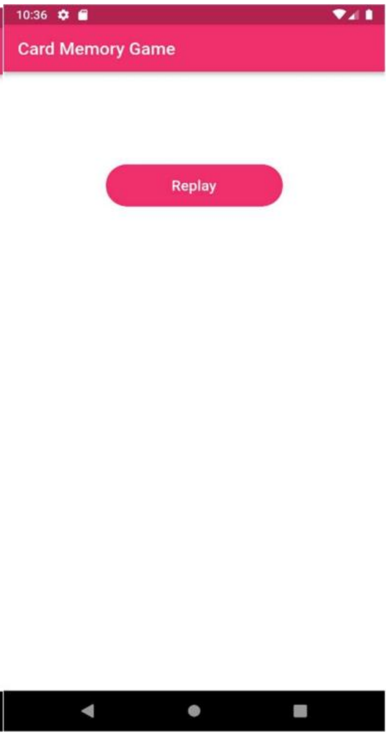
```
       selectedTile = "";
      }
     } else {
      setState(() {
       selectedTile = myPairs[widget.tileIndex].getImageAssetPath();
       selectedIndex = widget.tileIndex;
      });

      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:









Result:

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

Connectivity Via SOAP Or REST

Ex.No:13 **Date:29/11/2022**

Aim:

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

Procedure:

- Import,
  - http.dart
  - 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')}",
  };}
```

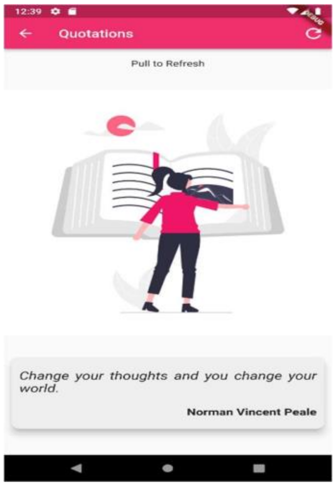
api.dart

```
  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,
    ),);}
Future<Null> getQuotes() async {
data = await Api.getQuotes();
setState(() {});
  }}
```

Output:



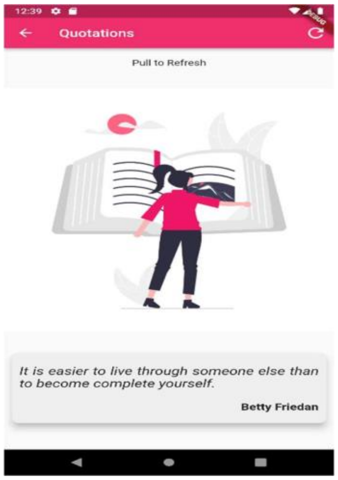
12:39

Quotations

Pull to Refresh

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:

Hence, a mobile application for data handling and connectivity via SOAP or REST tobackend services potentially hosted in a cloud environment.

Geo-Positioning, Accelerometer And Rich Gesture Based UI

Ex.No:14 **Date:03/12/2022**

Aim:

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

Procedure:

Geo-positioning:

- 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().

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:

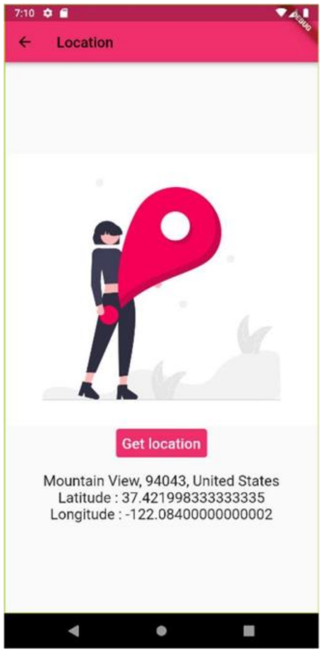
- In the onTap() property of the GestureDetector(), pass the function to be performed.
- In this case, it reverses the boolean value isLightsOn.
- This is used to switch the theme of the screen as dark or light.
- The child property of GestureDetector() is used to specify icon, on clicking which the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

Geo-positioning:

Code:

```
  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.pn
              g'), 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);
  }
 }
}
```

Output:



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

Accelerometer:

Code:

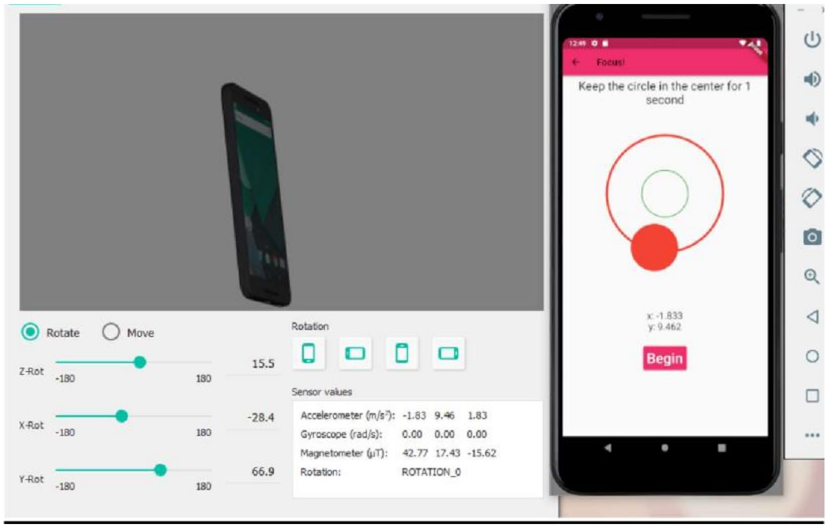
```
import 'dart:async';

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 movea 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 ben created so just resume it
    accel?.resume();
   }

   // Accelerometer events come faster than we need them so a timer is used to only
 proccessthem 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
      eventsetColor(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,
      ),
     )
    ],
   ),
  );
 }
}
```

Output:



Focus!

Keep the circle in the center for 1 second

x: -1.833
y: 9.462

Begin

● Rotate ○ Move

Z-Rot -180 180 15.5

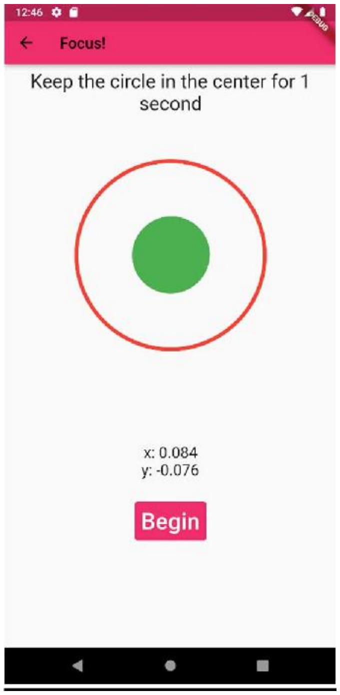
X-Rot -180 180 -28.4

Y-Rot -180 180 66.9

Rotation

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
```



Focus!

Keep the circle in the center for 1 second

x: 0.084
y: -0.076

Begin

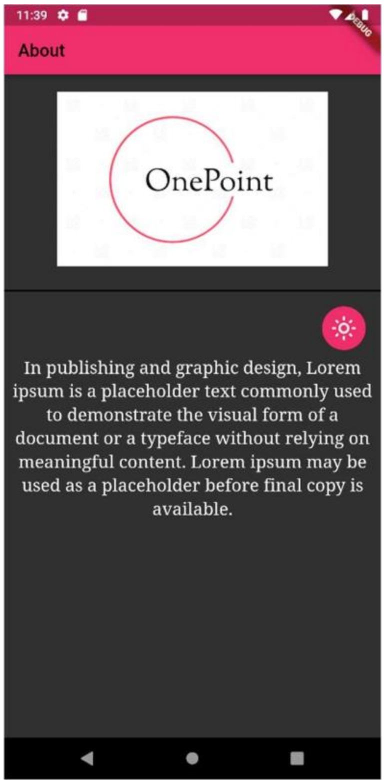
Gesture based UI:

Code:

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

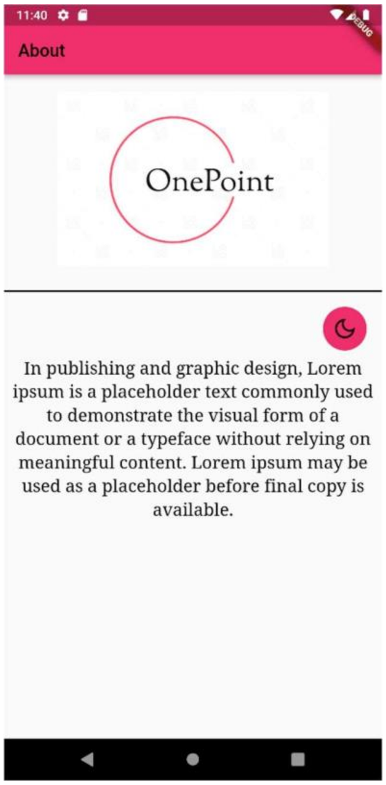
Output:

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 beenimplemented using Flutter.

Social Media Integration

Ex.No:15 **Date:06/12/2022**

Aim:

To write an application for integrating mobile applications in the market, includingsocial networking software integration with Google.

Procedure:

- Download the following packages using flutter pub add.
  - firebase_auth
  - firebase_core
  - google_sign_in
- In the firebase console, enable Google as a provider under Authentication-> Sign Inmethod.
- Get SHA key, by using the command gradlew signingReport at the android directoryof 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:
```

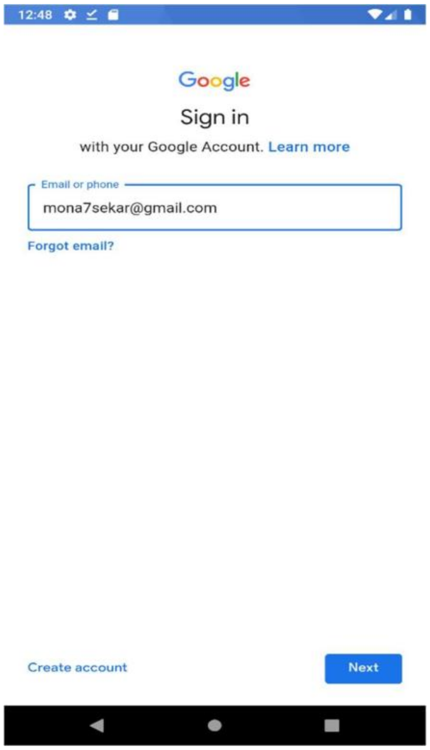
```
      password);return null;
    } on FirebaseAuthException catch (e)
      {return e.message;
    }
  }

  //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(SnackB
           ar( 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),
       ),
      ),
     ),
    ],
   ),
  );
 }
}
```

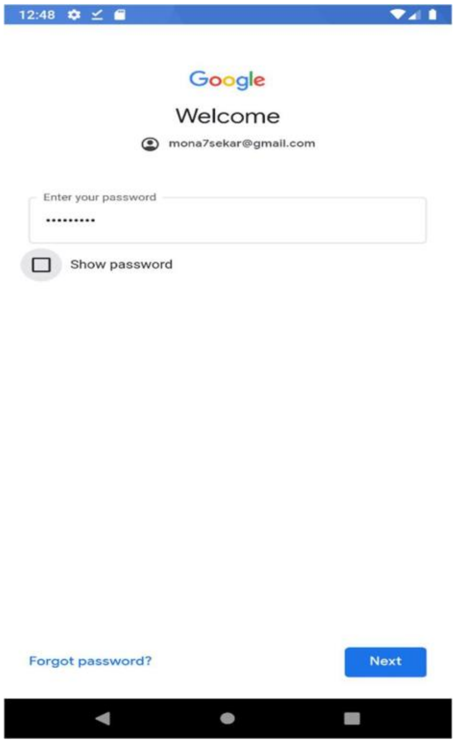
Output:



Sign in

with your Google Account. Learn more

Email or phone: mona7sekar@gmail.com

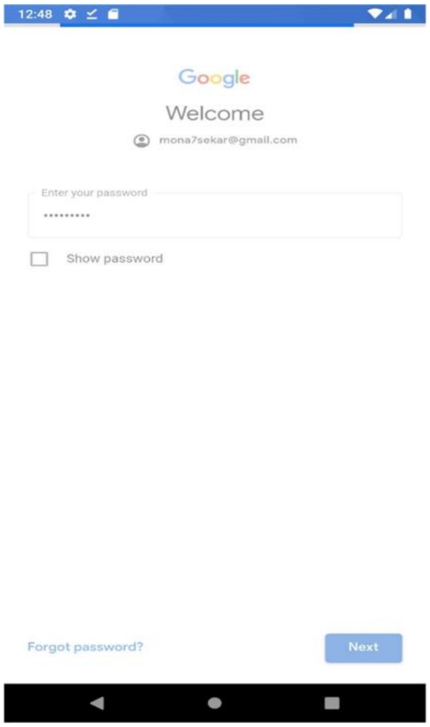


Welcome

mona7sekar@gmail.com

Enter your password: •••••••••

□ Show password

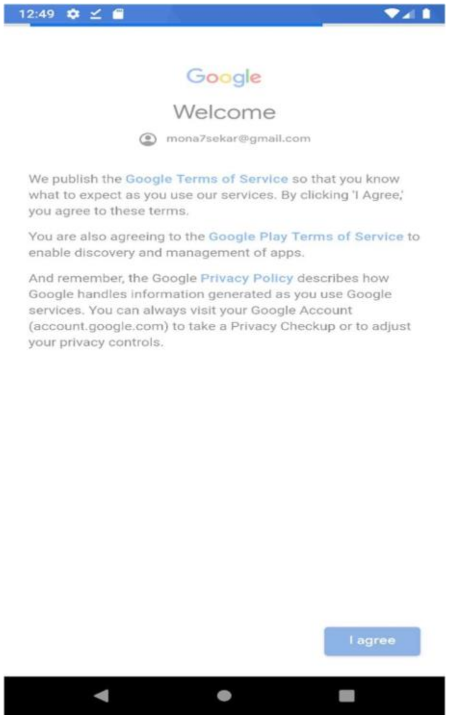


Welcome

mona7sekar@gmail.com

Enter your password: •••••••••

□ Show password



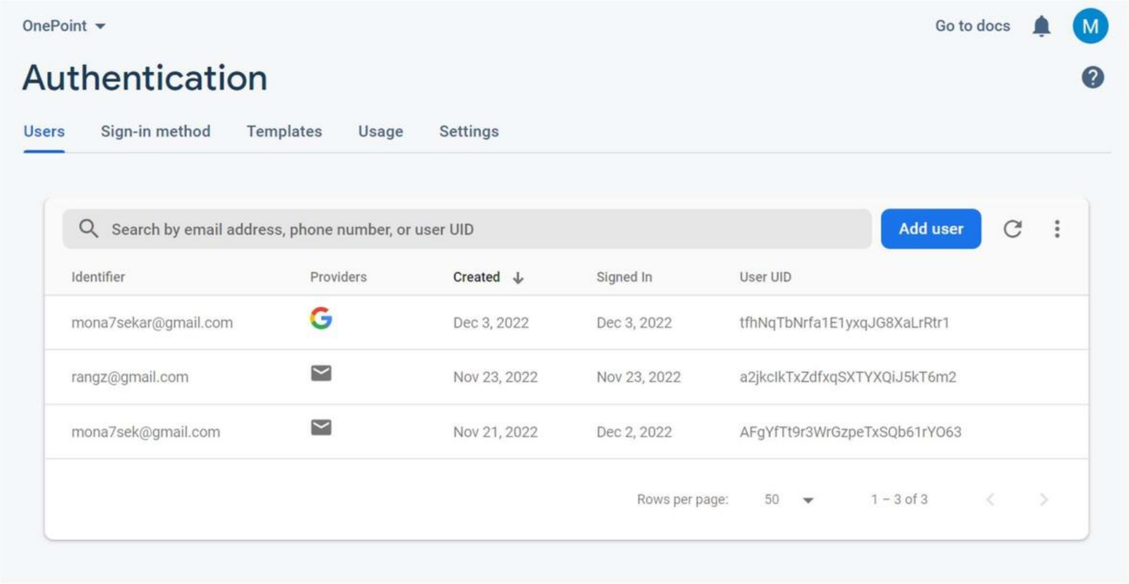
Welcome

mona7sekar@gmail.com

We publish the Google Terms of Service so that you know what to expect as you use our services. By clicking 'I Agree,' you agree to these terms.

You are also agreeing to the Google Play Terms of Service to enable discovery and management of apps.

And remember, the Google Privacy Policy describes how Google handles information generated as you use Google services. You can always visit your Google Account (account.google.com) to take a Privacy Checkup or to adjust your privacy controls.



Identifier	Providers	Created ↓	Signed In	User UID
mona7sekar@gmail.com	G	Dec 3, 2022	Dec 3, 2022	tfhNqTbNrfa1E1yxqJG8XaLrRtr1
rangz@gmail.com	✉	Nov 23, 2022	Nov 23, 2022	a2jkcIkTxZdfxqSXTYXQiJ5kT6m2
mona7sek@gmail.com	✉	Nov 21, 2022	Dec 2, 2022	AFgYfTt9r3WrGzpeTxSQb61rYO63

Result:

Thus, an application that uses social networking software (Google) for authenticationhas been implemented.

UIT 1711 Mobile Application Development Lab
Mini Project

Project Title:

Ceramic Industry and Products

Problem Statement:

Mobile Application for Ceramic industry and products.

Problem Description:

Ceramic Industry is growing rapidly. It came into existence about a century ago and the innovations have added freshness to this. Our solutions are meant to understand the requirements of the current scenario. Ceramics have made their way to our regular lives in the form of utensils, tiles, artwork, etc. The ceramic industry is facing several challenges these days and that has created an urge for creating an online identity.

From procurement to delivering the final product, there are many stages to complete. Ceramics are growing useful every day. Our App is a powerful management system. This is system aims to generate a strict control of business activities. These systems should be highly organized as well as rigorous to ensure complete business activities. Lots of daily operations take place in the ceramic industry. From purchasing, production, transportation to inventory management, there is much more to explore with these operations.

Source Code:

Dependencies:

```
cupertino_icons: ^1.0.2
firebase_core: ^2.4.0
firebase_auth: ^4.2.0
cloud_firestore: ^4.2.0
fluttertoast: ^8.1.1
google_fonts: 2.1.0
provider: ^6.0.4
```

Main.dart

```
import 'package:flutter/material.dart';
import 'login.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:login/model/cart_model.dart';
import 'package:provider/provider.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => CartModel(),
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        home: login(),
      ),
    );
  }
}
```

Login.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:login/pages/intro_screen.dart';
import 'homescreen.dart';

class login extends StatefulWidget {
  const login({super.key});

  @override
  State<login> createState() => _loginState();
}

class _loginState extends State<login> {
  final _formKey = GlobalKey<FormState>();

  final emailEditingController = new TextEditingController();
  final passwordEditingController = new TextEditingController();

  final _auth = FirebaseAuth.instance;
  @override
  Widget build(BuildContext context) {
    //emailfield
    final emailfield = TextFormField(
      autofocus: false,
      controller: emailEditingController,
      keyboardType: TextInputType.emailAddress,
      validator: (value) {
        if (value!.isEmpty) {
          return ("Please enter the username");
        }
        if (!RegExp("^[a-zA-Z0-9+_.-]+@[a-zA-Z0-9.-]+.[a-z]").hasMatch(value)) {
          return ("Please Enter a valid email");
        }
      },
      onSaved: (value) {
        emailEditingController.text = value!;
      },
      textInputAction: TextInputAction.next,
      decoration: InputDecoration(
          prefixIcon: Icon(Icons.mail),
          hintText: "USERNAME",
   contentPadding: EdgeInsets.fromLTRB(20, 15, 20, 15),
   border: OutlineInputBorder(borderRadius: BorderRadius.circular(20))),
);

final passwordfield = TextFormField(
 autofocus: false,
 controller: passwordEditingController,
 obscureText: true,
 validator: (value) {
  RegExp regex = new RegExp(r'^.{6,}$');
  if (value!.isEmpty) {
   return ("Please enter the Password");
  }
  if (!regex.hasMatch(value)) {
   return ("Enter Valid Password(Min. 6 Character)");
  }
 },
 onSaved: (value) {
  passwordEditingController.text = value!;
 },
 textInputAction: TextInputAction.done,
 decoration: InputDecoration(
   prefixIcon: Icon(Icons.vpn_key),
   contentPadding: EdgeInsets.fromLTRB(20, 15, 20, 15),
   hintText: "PASSWORD",
   border: OutlineInputBorder(borderRadius: BorderRadius.circular(20))),
);

final loginButton = Material(
 elevation: 5,
 borderRadius: BorderRadius.circular(30),
 child: MaterialButton(
  color: Color(0xFF4C53A5),
  padding: EdgeInsets.fromLTRB(20, 15, 20, 15),
  minWidth: MediaQuery.of(context).size.width,
  onPressed: () {
   signIn(emailEditingController.text, passwordEditingController.text);
  },
  child: const Text(
   "LOGIN",
   textAlign: TextAlign.center,
   style: TextStyle(
     fontWeight: FontWeight.bold, fontSize: 20, color: Colors.white),
  ),
 ),
);
return Scaffold(
 backgroundColor: Colors.white,
 body: Center(
  child: SingleChildScrollView(
   child: Container(
    child: Padding(
     padding: const EdgeInsets.all(30.0),
      child: Form(
        key: _formKey,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Padding(
              padding: const EdgeInsets.only(
                left: 80.0,
                right: 80.0,
                top: 20,
                bottom: 20,
              ),
              child: Image.asset('lib/images/login.jpg'),
            ),
            SizedBox(
              height: 70,
              child: Text(
                "LOGIN ",
                style: TextStyle(
                  fontSize: 30,
                  color: Color(0xFF4C53A5),
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
            emailfield,
            SizedBox(
              height: 30,
            ),
            passwordfield,
            SizedBox(
              height: 30,
            ),
            loginButton,
            SizedBox(
              height: 30,
            ),
            Text("DONT HAVE A ACCOUNT!!"),
            GestureDetector(
              onTap: () {},
              child: Text(
                "SIGNUP",
                style:
                    TextStyle(color: Color(0xFF4C53A5), fontSize: 16),
              ),
            )
          ],
        ),
      ),
    ),
  ),
),
      ),
    );
  }

  void signIn(String email, String password) async {
    if (_formKey.currentState!.validate()) {
      await _auth
          .signInWithEmailAndPassword(email: email, password: password)
          .then((uid) => {
                Fluttertoast.showToast(msg: "Login Successful"),
                Navigator.of(context).pushReplacement(
                    MaterialPageRoute(builder: (context) => IntroScreen())),
              })
          .catchError((e) => {Fluttertoast.showToast(msg: e!.message)});
    }
  }
}
```

Pages:

Intro_screen.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[50],
      body: SafeArea(
        child: Column(
          children: [
            // big logo
            Padding(
              padding: const EdgeInsets.only(
                left: 80.0,
                right: 80.0,
                top: 30,
                bottom: 20,
              ),
              child: Image.asset('lib/images/img5.jpg'),
            ),

            // we deliver groceries at your doorstep
            Padding(
              padding: const EdgeInsets.all(28.0),
              child: Text(
                'We deliver Ceramic Products at your doorstep',
                textAlign: TextAlign.center,
                style: GoogleFonts.notoSerif(
                    color: Color(0xFF4C53A5),
                    fontSize: 28,
                    fontWeight: FontWeight.bold),
              ),
            ),

            // groceree gives you fresh vegetables and fruits
            Text(
              'New items everyday',
              textAlign: TextAlign.center,
              style: TextStyle(
                fontSize: 16,
                color: Color(0xFF4C53A5),
              ),
            ),

            const SizedBox(height: 24),

            const Spacer(),

            // get started button
            GestureDetector(
              onTap: () => Navigator.pushReplacement(
                context,
                MaterialPageRoute(
                  builder: (context) {
                    return HomePage();
                  },
                ),
              ),
              child: Container(
                padding: const EdgeInsets.all(24),
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(16),
                  color: Color(0xFF4C53A5),
                ),
                child: const Text(
                  "Get Started",
                  style: TextStyle(
                    color: Colors.white,
                    // fontWeight: FontWeight.bold,
                    fontSize: 16,
                  ),
                ),
              ),
            ),

            const Spacer(),
          ],
        ),
      ),
    );
  }
}
```

Home_screen.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';
import '../components/grocery_item_tile.dart';
import '../model/cart_model.dart';
import 'cart_page.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0,
        leading: Padding(
          padding: const EdgeInsets.only(left: 24.0),
          child: Icon(
            Icons.location_on,
            color: Color(0xFF4C53A5),
          ),
        ),
        title: Text(
          'Chennai, India',
          style: TextStyle(
            fontSize: 16,
            color: Color(0xFF4C53A5),
          ),
        ),
        centerTitle: false,
        actions: [
          Padding(
            padding: const EdgeInsets.only(right: 24.0),
            child: Container(
              padding: EdgeInsets.all(16),
              decoration: BoxDecoration(
                color: Colors.grey[200],
                borderRadius: BorderRadius.circular(12),
              ),
              child: Icon(
                Icons.person,
                color: Colors.grey,
              ),
            ),
          ),
        ],
      ),
      floatingActionButton: FloatingActionButton(
        backgroundColor: Colors.black,
        onPressed: () => Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) {
              return CartPage();
            },
          ),
        ),
        child: const Icon(Icons.shopping_bag),
      ),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const SizedBox(height: 48),

          // good morning bro
          const Padding(
            padding: EdgeInsets.symmetric(horizontal: 24.0),
            child: Text(
              'Welcome Buyers,',
              style: TextStyle(color: Color(0xFF4C53A5)),
            ),
          ),

          const SizedBox(height: 4),

          // Let's order fresh items for you
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 24.0),
            child: Text(
              "Let's order Ceramic Industry Products for you",
              style: GoogleFonts.notoSerif(
                color: Color(0xFF4C53A5),
                fontSize: 20,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),

          const SizedBox(height: 24),

          const Padding(
            padding: EdgeInsets.symmetric(horizontal: 24.0),
            child: Divider(),
          ),

          const SizedBox(height: 24),
          // categories -> horizontal listview
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 24.0),
            child: Text(
              "New Arrivals",
              style: GoogleFonts.notoSerif(
                color: Color(0xFF4C53A5),
                fontWeight: FontWeight.bold,
                fontSize: 18,
              ),
            ),
          ),

          // recent orders -> show last 3
          Expanded(
            child: Consumer<CartModel>(
              builder: (context, value, child) {
                return GridView.builder(
                  padding: const EdgeInsets.all(12),
                  // physics: const NeverScrollableScrollPhysics(),
                  itemCount: value.shopItems.length,
                  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
                    crossAxisCount: 2,
                    childAspectRatio: 1 / 1.2,
                  ),
                  itemBuilder: (context, index) {
                    return GroceryItemTile(
                      itemName: value.shopItems[index][0],
                      itemPrice: value.shopItems[index][1],
                      imagePath: value.shopItems[index][2],
                      color: value.shopItems[index][3],
                      onPressed: () =>
                          Provider.of<CartModel>(context, listen: false)
                              .addItemToCart(index),
                    );
                  },
                );
              },
            ),
          ),
        ],
      ),
    );
  }
}
```

Cart_page.dart

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

import '../model/cart_model.dart';
```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0,
        iconTheme: IconThemeData(
          color: Colors.grey[800],
        ),
      ),
      body: Consumer<CartModel>(
        builder: (context, value, child) {
          return Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Let's order fresh items for you
              Padding(
                padding: EdgeInsets.symmetric(horizontal: 24.0),
                child: Text(
                  "My Cart",
                  style: GoogleFonts.notoSerif(
                    fontSize: 36,
                    fontWeight: FontWeight.bold,
                    color: Color(0xFF4C53A5),
                  ),
                ),
              ),

              // list view of cart
              Expanded(
                child: Padding(
                  padding: const EdgeInsets.all(12.0),
                  child: ListView.builder(
                    itemCount: value.cartItems.length,
                    padding: EdgeInsets.all(12),
                    itemBuilder: (context, index) {
                      return Padding(
                        padding: const EdgeInsets.all(12.0),
                        child: Container(
                          decoration: BoxDecoration(
                              color: Colors.grey[200],
                              borderRadius: BorderRadius.circular(8)),
                          child: ListTile(
                            leading: Image.asset(
                              value.cartItems[index][2],
                              height: 36,
                            ),
                            title: Text(
                              value.cartItems[index][0],
                              style: const TextStyle(fontSize: 18),
                            ),
                            subtitle: Text(
                              '\$' + value.cartItems[index][1],
                              style: const TextStyle(fontSize: 12),
                            ),
                            trailing: IconButton(
                              icon: const Icon(Icons.cancel),
                              onPressed: () =>
                                  Provider.of<CartModel>(context, listen: false)
                                      .removeItemFromCart(index),
                            ),
                          ),
                        ),
                      );
                    },
                  ),
                ),
              ),

              // total amount + pay now

              Padding(
                padding: const EdgeInsets.all(36.0),
                child: Container(
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(8),
                    color: Colors.green,
                  ),
                  padding: const EdgeInsets.all(24),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text(
                            'Total Price',
                            style: TextStyle(color: Colors.green[200]),
                          ),

                          const SizedBox(height: 8),
                          // total price
                          Text(
                            '\$${value.calculateTotal()}',
                            style: const TextStyle(
                              fontSize: 18,
                              fontWeight: FontWeight.bold,
                              color: Colors.white,
                            ),
                          ),
                        ],
                      ),
                      // pay now
                      Container(
                        decoration: BoxDecoration(
                          border: Border.all(color: Colors.green.shade200),
                          borderRadius: BorderRadius.circular(28),
                        ),
                        padding: const EdgeInsets.all(12),
                        child: Row(
                          children: const [
                            Text(
                              'Pay Now',
                              style: TextStyle(color: Colors.white),
                            ),
                            Icon(
                              Icons.arrow_forward_ios,
                              size: 16,
                              color: Colors.white,
                            ),
                          ],
                        ),
                      ),
                    ],
                  ),
                ),
              )
            ],
          );
        },
      ),
    );
  }
}
```

Model:

Cart_model.dart

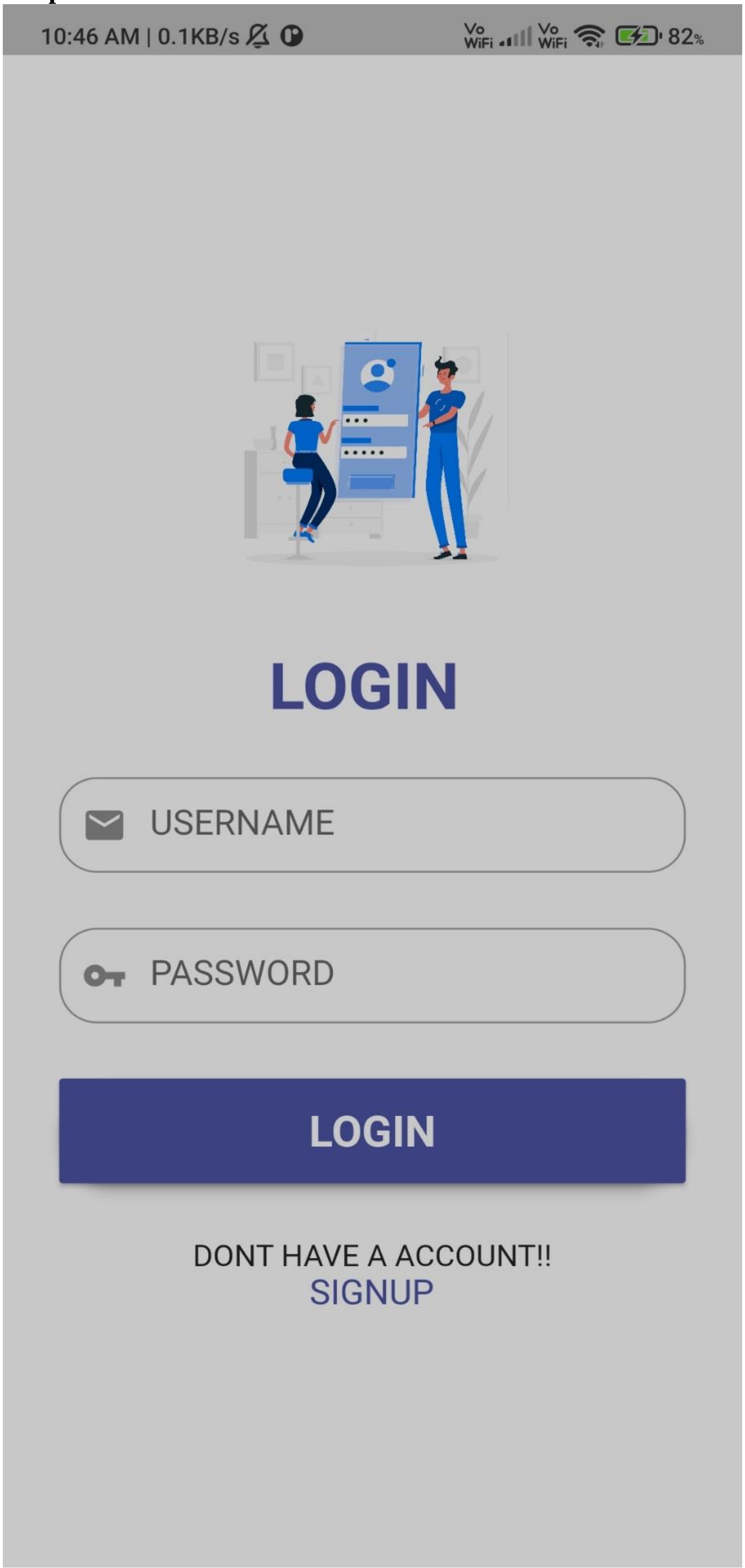
```
import 'package:flutter/material.dart';

class CartModel extends ChangeNotifier {
  // list of items on sale
  final List _shopItems = const [
    // [ itemName, itemPrice, imagePath, color ]
    ["Stylish Pots", "100.00", "lib/images/img1.jpg", Colors.green],
    ["Plates and Cups", "200.00", "lib/images/imgFc5.JPG", Colors.yellow],
    ["Floor Tiles", "150.00", "lib/images/imgFc1.jpg", Colors.brown],
    ["Toiletary", "400.00", "lib/images/imgFc2.jpg", Colors.blue],
    ["Pipes", "50.00", "lib/images/imgFc6.JPG", Colors.lime],
    ["Gas Fire Radient", "500.00", "lib/images/img6.jpg", Colors.orange],
    ["Roof Tiles", "250.00", "lib/images/imgFc7.jpg", Colors.lightGreen],
    ["Bricks", "330.00", "lib/images/imgFc9.JPG", Colors.blueGrey],
    ["TableWare", "600.00", "lib/images/img7.PNG", Colors.indigo],
    ["Green Vase", "90.00", "lib/images/img2.jpg", Colors.brown],
    ["Staring Set", "350.00", "lib/images/ss1.jpg", Colors.yellow],
    ["CookWare", "420.00", "lib/images/img8.jpg", Colors.purple],
    ["3 Combo Pots", "280.00", "lib/images/splashscreen2.jpg", Colors.green],
    ["Full Set Combo", "400.00", "lib/images/imgFc4.png", Colors.blue],
    ["Bluish Cups", "140.00", "lib/images/img3.jpg", Colors.deepOrange],
  ];

  // list of cart items
  List _cartItems = [];

  get cartItems => _cartItems;

  get shopItems => _shopItems;

  // add item to cart
  void addItemToCart(int index) {
    _cartItems.add(_shopItems[index]);
    notifyListeners();
  }

  // remove item from cart
  void removeItemFromCart(int index) {
    _cartItems.removeAt(index);
    notifyListeners();
  }

  // calculate total price
  String calculateTotal() {
    double totalPrice = 0;
    for (int i = 0; i < cartItems.length; i++) {
      totalPrice += double.parse(cartItems[i][1]);
    }
    return totalPrice.toStringAsFixed(2);
  }
}
```

Components:

ceramic_item_tile.dart

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

class GroceryItemTile extends StatelessWidget {
  final String itemName;
  final String itemPrice;
  final String imagePath;
  final color;
  void Function()? onPressed;

  GroceryItemTile({
    super.key,
    required this.itemName,
    required this.itemPrice,
    required this.imagePath,
    required this.color,
    required this.onPressed,
  });

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(12),
      child: Container(
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(12),
          color: color[100],
        ),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            // item image
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 40.0),
              child: Image.asset(
                imagePath,
                height: 64,
              ),
            ),

            // item name
            Text(
              itemName,
              style: TextStyle(
                fontSize: 16,
              ),
            ),

            MaterialButton(
              onPressed: onPressed,
              color: color,
              child: Text(
                '\$' + itemPrice,
                style: TextStyle(
                  color: Colors.white,
                  fontWeight: FontWeight.bold,
                ),
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

Output:



10:46 AM | 0.1KB/s

82%

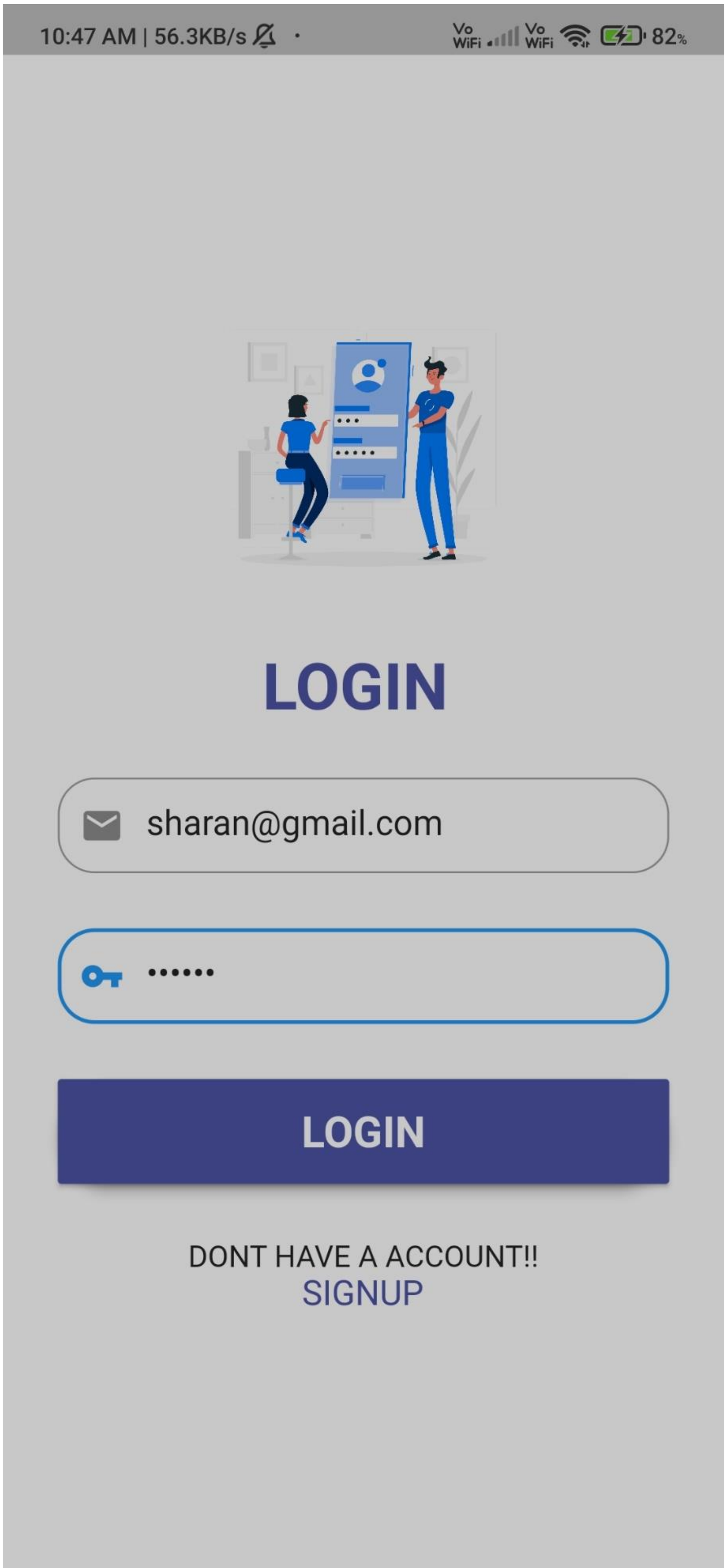
LOGIN

USERNAME

PASSWORD

LOGIN

DONT HAVE A ACCOUNT!!
SIGNUP



10:47 AM | 56.3KB/s

82%

LOGIN

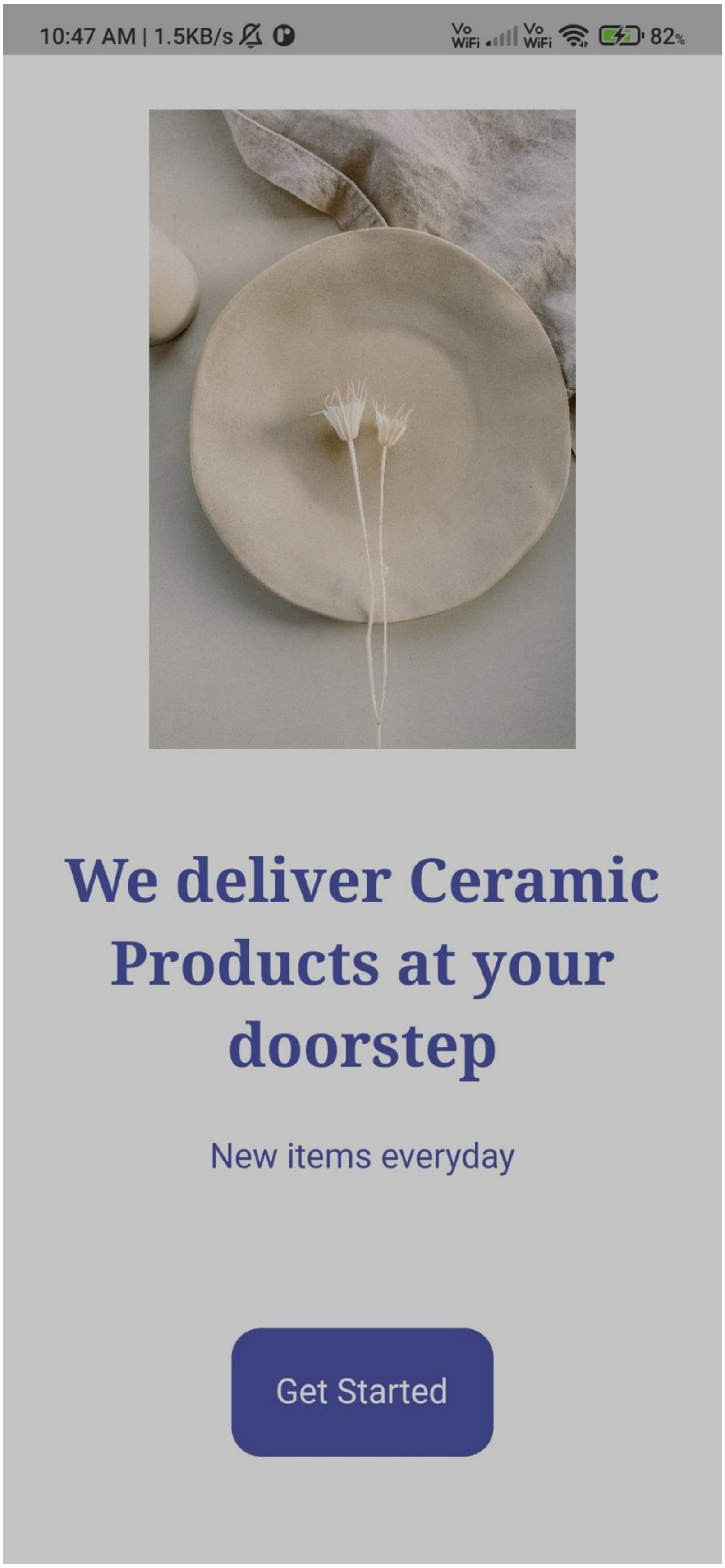
sharan@gmail.com

••••••

LOGIN

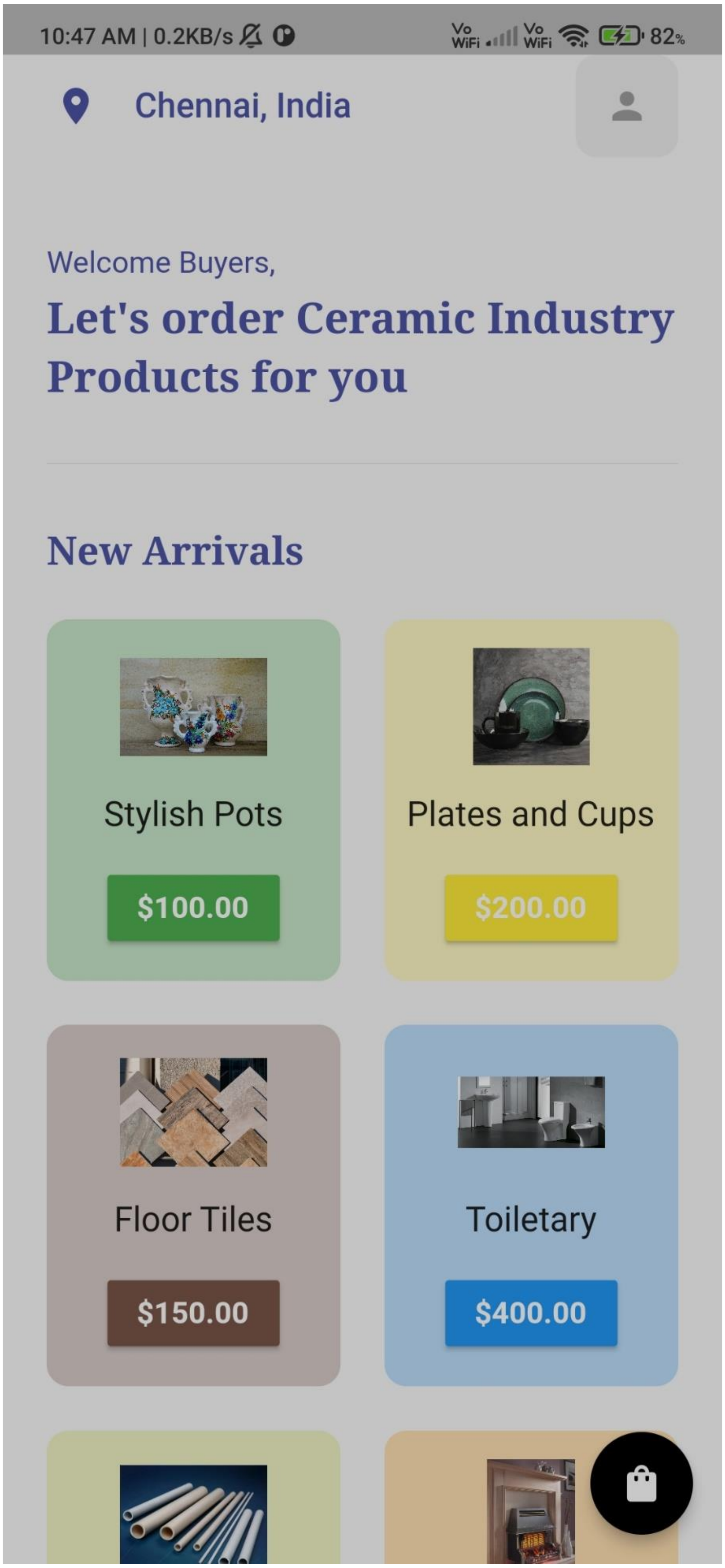
DONT HAVE A ACCOUNT!!

SIGNUP



We deliver Ceramic Products at your doorstep

New items everyday



Chennai, India

Welcome Buyers,

Let's order Ceramic Industry Products for you

New Arrivals

Stylish Pots

$100.00

Plates and Cups

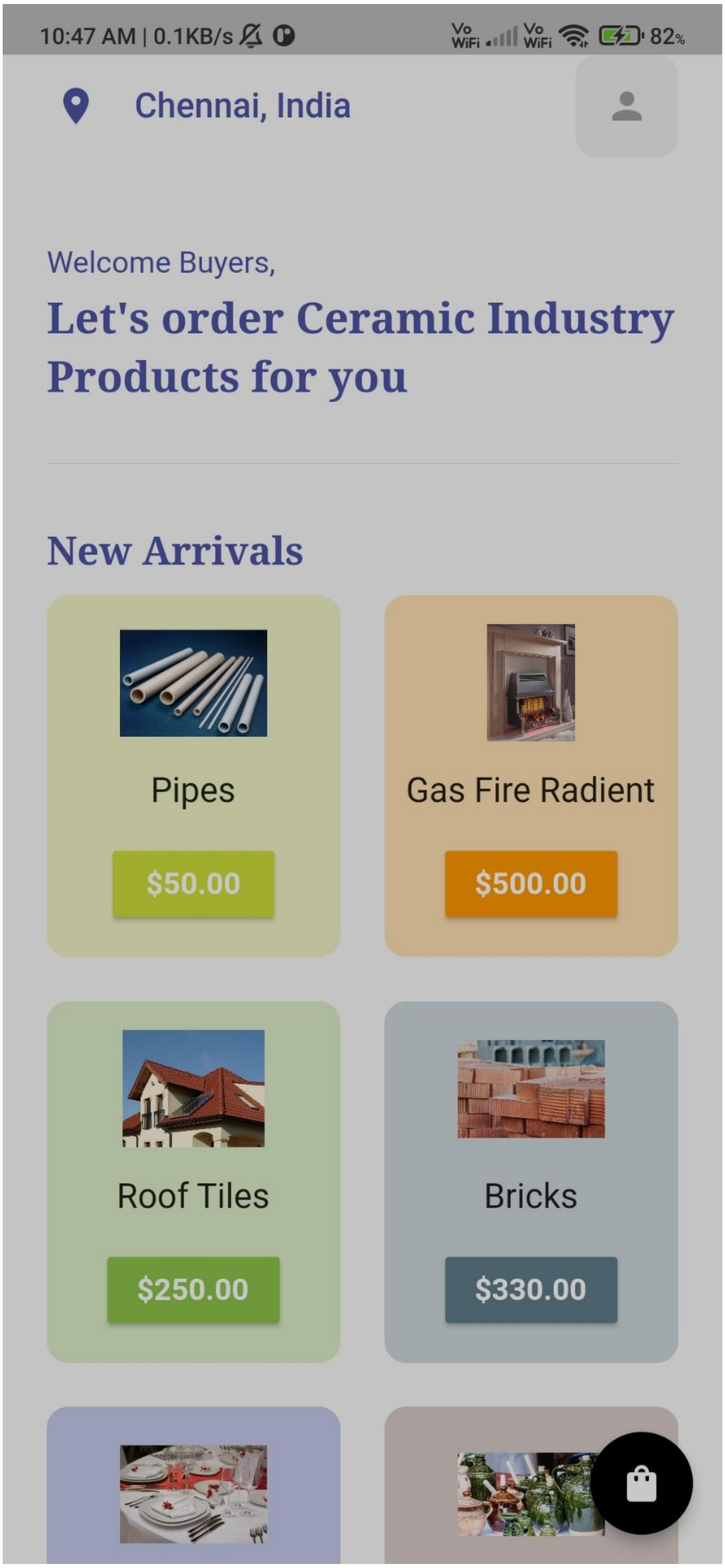
$200.00

Floor Tiles

$150.00

Toiletary

$400.00



Chennai, India

Welcome Buyers,

Let's order Ceramic Industry Products for you

New Arrivals

Pipes

$50.00

Gas Fire Radient

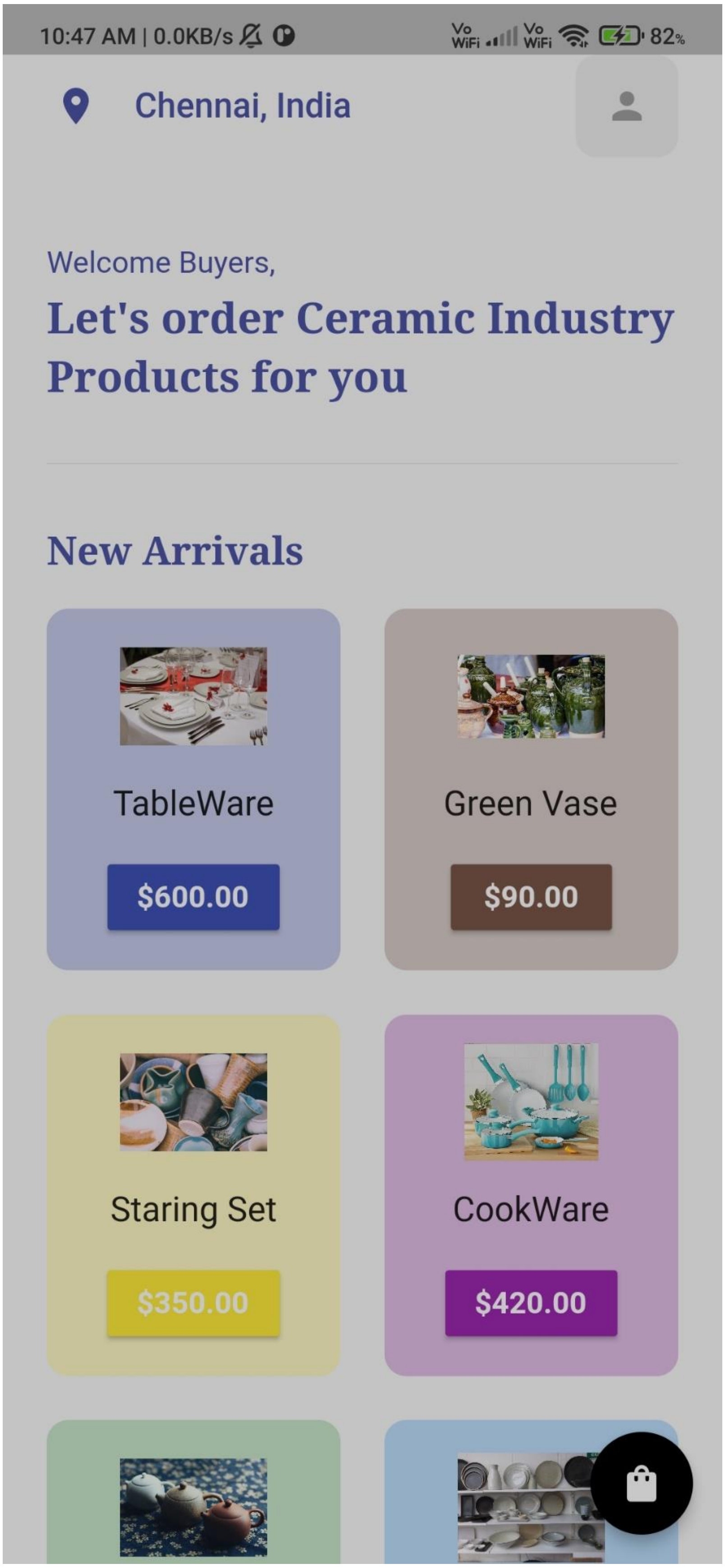
$500.00

Roof Tiles

$250.00

Bricks

$330.00



Chennai, India

Welcome Buyers,

Let's order Ceramic Industry Products for you

New Arrivals

TableWare

$600.00

Green Vase

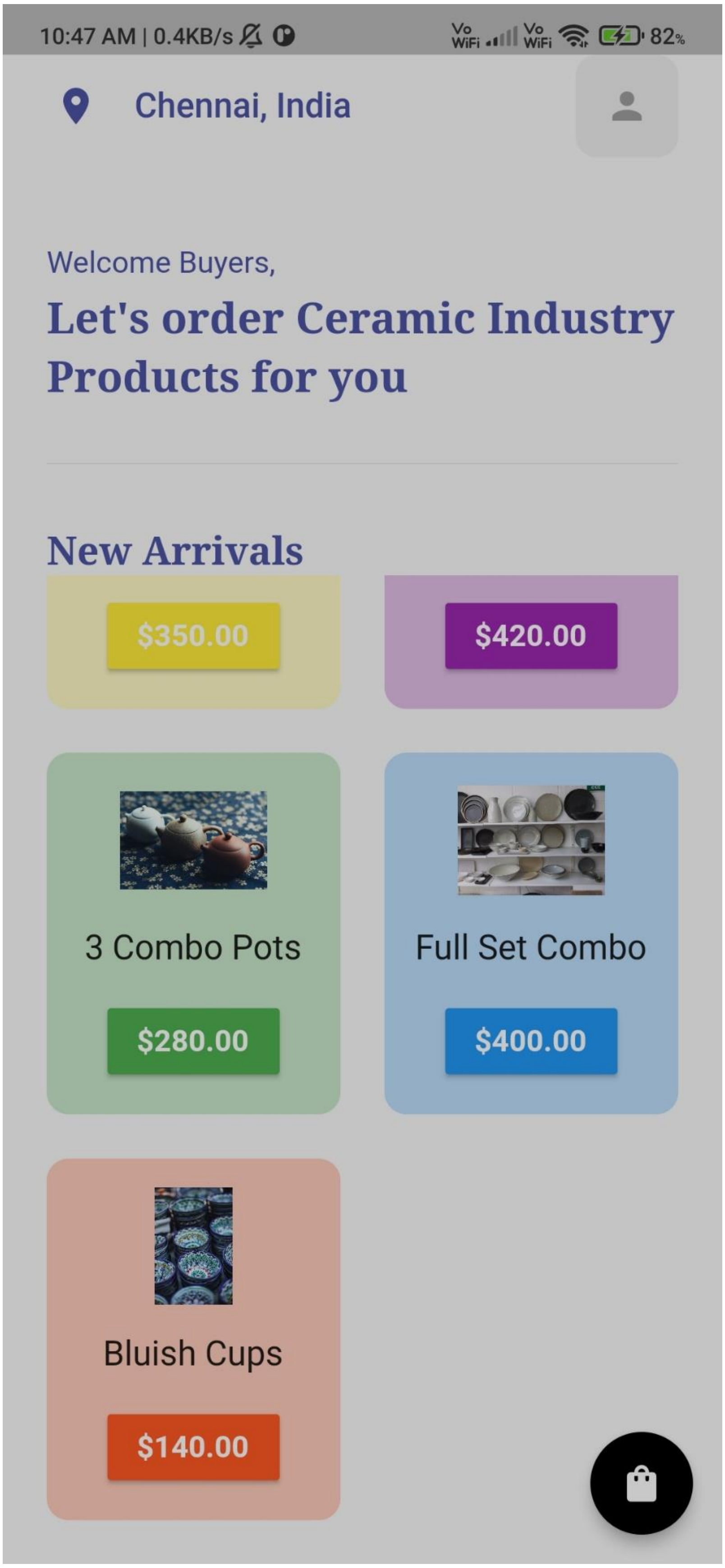
$90.00

Staring Set

$350.00

CookWare

$420.00



10:47 AM | 0.4KB/s

Vo WiFi Vo WiFi 82%

Chennai, India

Welcome Buyers,

Let's order Ceramic Industry Products for you

New Arrivals

$350.00

$420.00

3 Combo Pots

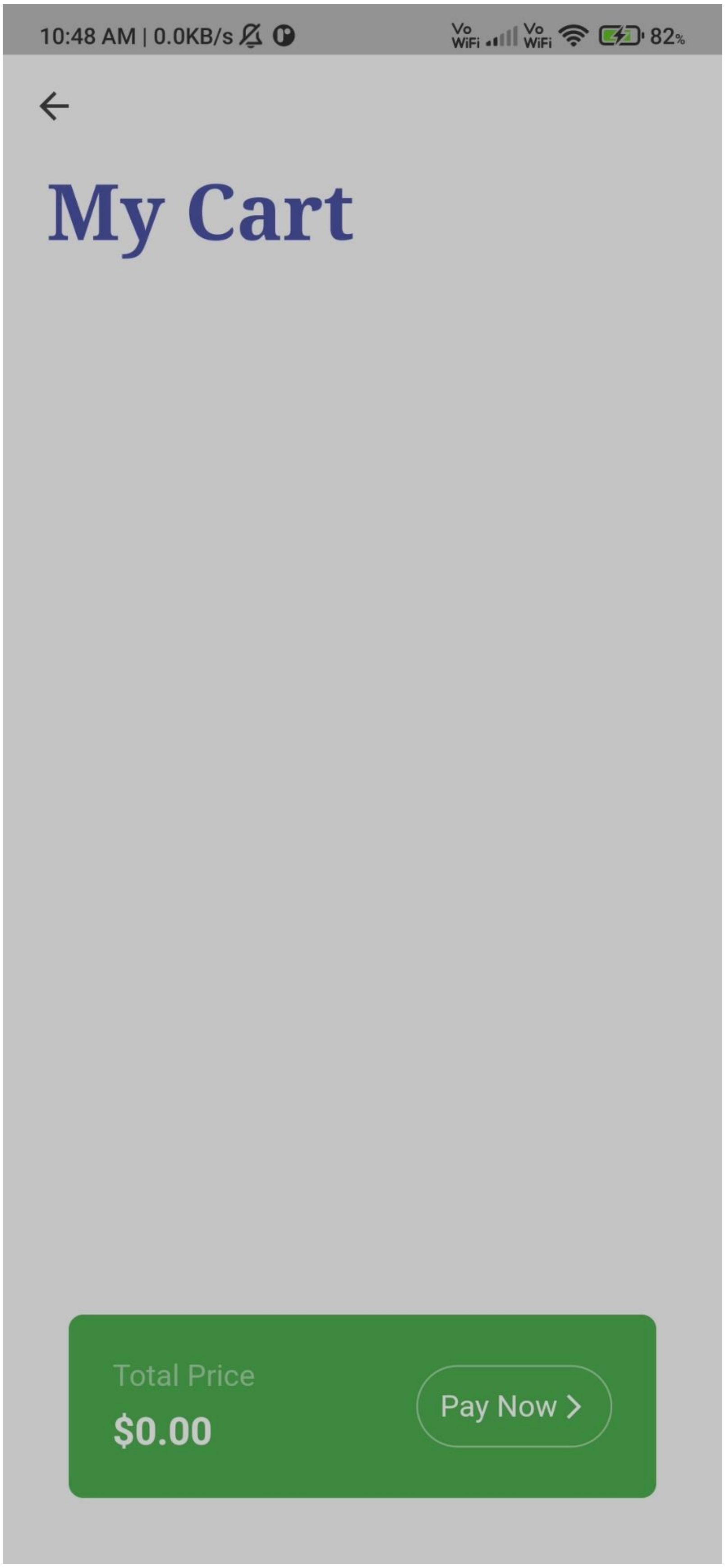
$280.00

Full Set Combo

$400.00

Bluish Cups

$140.00



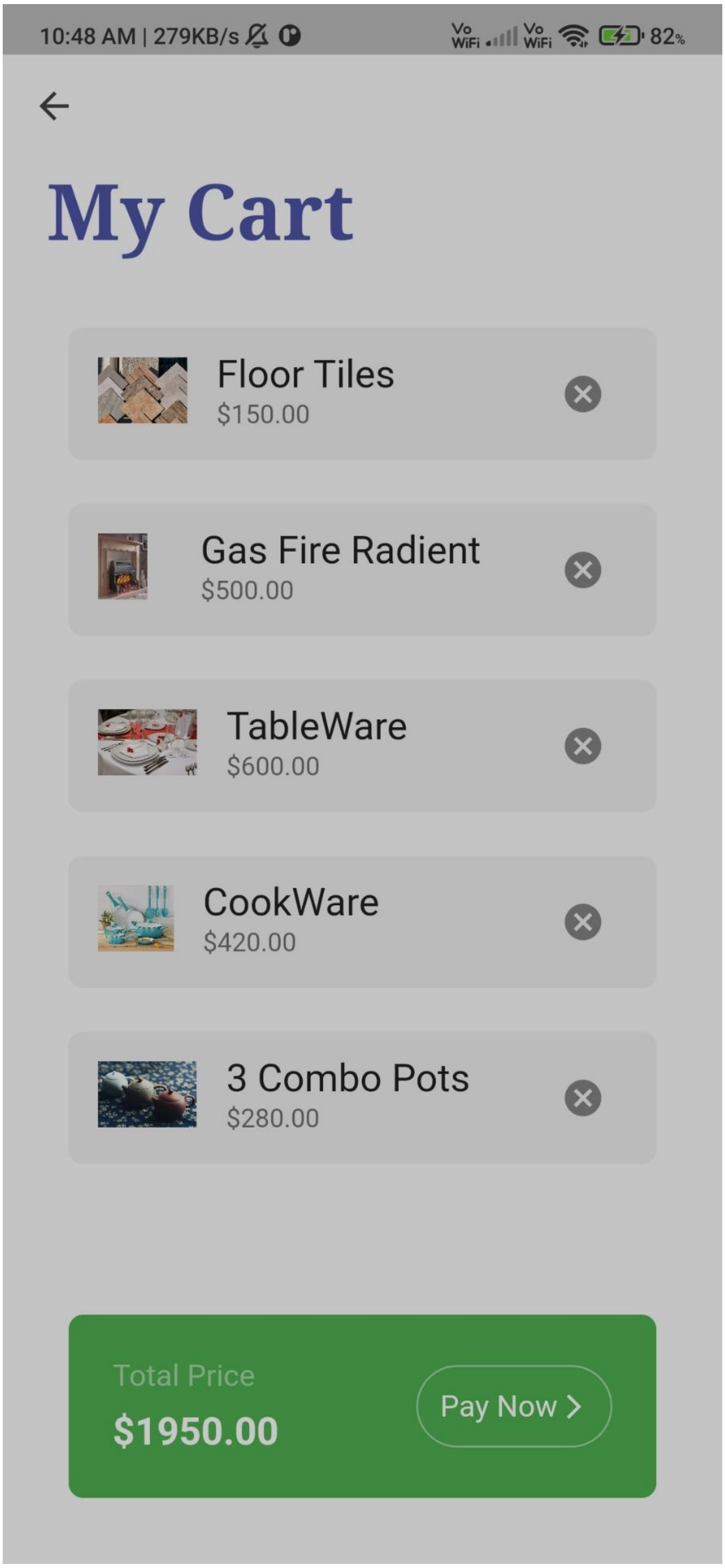
10:48 AM | 0.0KB/s

82%

My Cart

Total Price

$0.00

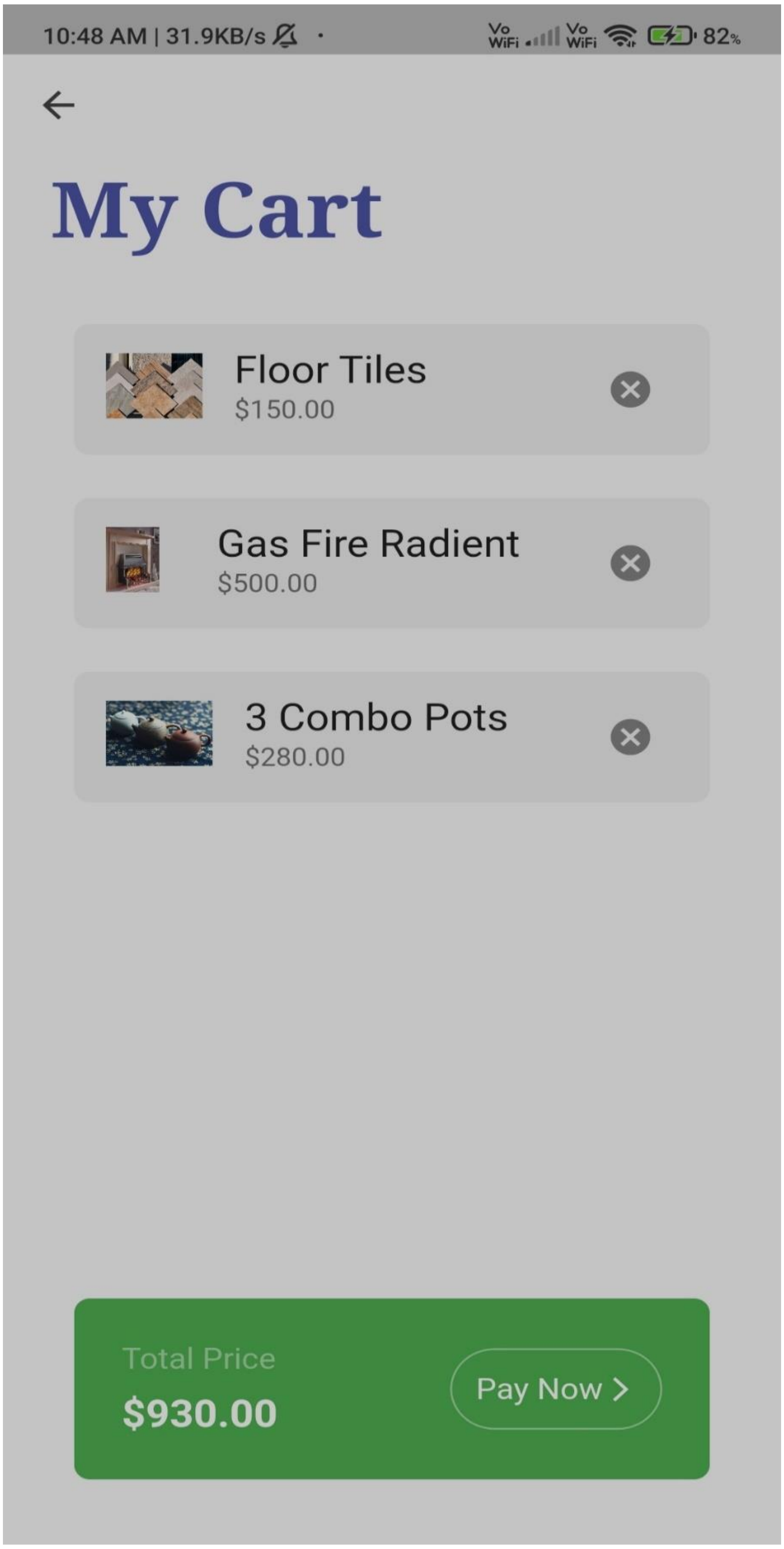


My Cart

Floor Tiles	$150.00
Gas Fire Radient	$500.00
TableWare	$600.00
CookWare	$420.00
3 Combo Pots	$280.00

Total Price

$1950.00



10:48 AM | 31.9KB/s

82%

My Cart

Floor Tiles
$150.00

Gas Fire Radient
$500.00

3 Combo Pots
$280.00

Total Price
$930.00

Result:
The project has been tested and it has run successfully.