

---

## Practical 4

**Aim:** Different actions on floating action button

A. Floating action button

B. Draggable floating action button

**Code:**

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: const MyHomePage(title: 'Floating Button demo'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});
  final String title;
  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  double _counter = 0;
  int display = 100;
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
  void _decrementCounter() {
    setState(() {
      _counter--;
    });
  }
  void _multiply() {
    setState(() {
      _counter = 2 * _counter;
    });
  }
}
```

---

---

```
}
void _divide() {
  setState(() {
    _counter = _counter / 2;
  });
}

void _reset() {
  setState(() {
    _counter = 0;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          Container(
            alignment: Alignment.center,
            height: 100,
            width: 200,
            decoration: BoxDecoration(border: Border.all(color:
Colors.black)),
            child: Text(
              '$_counter',
              style: Theme.of(context).textTheme.headline4,
            ),
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: <Widget>[
              FloatingActionButton(
                onPressed: _incrementCounter,
                tooltip: 'Increment',
                child: const Icon(Icons.add),
              ),
              FloatingActionButton(
                onPressed: _decrementCounter,
                tooltip: 'Decrement',
                child: const Icon(Icons.minimize_outlined),
              ),
              FloatingActionButton.extended(
                onPressed: _multiply,
                tooltip: 'Multiply by 2',
                label: Text("X 2"),
              ),
              FloatingActionButton.extended(
                onPressed: _divide,
                tooltip: 'Divide by 2',
              ),
            ],
          ),
        ],
      ),
    ),
  );
}
```

---

---

```

        label: Text("/ 2"),
      ),
    ],
  ),
  FloatingActionButton.extended(
    onPressed: _reset,
    tooltip: 'Reset',
    label: Text("Reset"),
  ),
],
)),
floatingActionButton: Stack(
  fit: StackFit.expand,
  children: [
    Align(
      alignment: Alignment.bottomRight,
      child: DraggableFab(
        child: FloatingActionButton(
          onPressed: _question,
          tooltip: 'Any Question',
          child: Icon(Icons.question_mark),
        )),
    ),
  ],
),
);
}
}

```

## Conclusions:

**Error/ Debug:** The draggable floating action button should be inside Stack and alignment for its proper working otherwise it disappears after dragging once.

## Learning:

- How Floating action button works and to change its shape
  - `FloatingActionButton.extended()` for text instead or along with icons.
  - Draggable floating action button `DraggableFab()`
-

---

# Practical 5

Aim: Firebase Connection Establishment

Code:

Main.dart File

```
import 'dart:math';

import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:firebase_core/firebase_core.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: const FirebaseOptions(
      apiKey: "AIzaSyDRCm8tQaTH1lDDtO9UNUIyWDqxHvw4GpE",
      authDomain:
        "practical-6.firebaseio.com", //project name
      databaseURL:
        "https://practical-6-8cb91-default-rtdb.firebaseio.com/",
      appId: "1:69678794187:android:47b1fc8531274f8b39624e",
      messagingSenderId: "69678794187", //cloud messaging
      projectId: "practical-6-8cb91"));
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  final Future<FirebaseApp> _fbApp = Firebase.initializeApp();
  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,
      ),
      home: FutureBuilder(
        future: _fbApp,
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            print("you have an error ${snapshot.error.toString()}");
            return const Text("something went wrong");
          } else if (snapshot.hasData) {
            return const MyHomePage(title: "Firebase Demo");
          } else {

```

```

        return const Center(
          child: CircularProgressIndicator(),
        );
      },
    ),
    //home: const MyHomePage(title: 'Firebase demo'),
  );
}

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

  final String title;

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

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

  void _incrementCounter() {
    DatabaseReference mTestref =
    FirebaseDatabase.instance.ref().child("test");
    mTestref.set("Hello SYIT ${Random().nextInt(100)}");

    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headline4,
            ),
          ],
        ),
      ),
    ),
  ),
}

```

---

```
        floatingActionButton: FloatingActionButton(  
            onPressed: _incrementCounter,  
            tooltip: 'Increment',  
            child: const Icon(Icons.add),  
        ), // This trailing comma makes auto-formatting nicer for build  
methods.  
    );  
}  
}
```

Packages imported:

1. `Firebase_database`
2. `Firebase_core`

Steps:

1. Create a Firebase Project
2. Configure Google Analytics
3. Download and Paste the `google-services.json` file in the app-level
4. Copy the following commands in the respective `build.gradle` files

Root-level (project-level) Gradle file (`<project>/build.gradle`):

```
buildscript {  
  
    repositories {  
  
        google() // Google's Maven repository  
content_copy  
  
        mavenCentral() // Maven Central repository  
content_copy  
    }  
  
    dependencies {  
  
        classpath 'com.google.gms:google-services:4.3.15'  
content_copy  
    }  
}
```

---

---

Module (app-level) Gradle file (<project>/<app-module>/build.gradle):

```
plugins {  
  id 'com.android.application'  
  content_copy  
  
  id 'com.google.gms.google-services'  
  content_copy  
}  
  
dependencies {  
  implementation platform('com.google.firebase:firebase-bom:31.2.2')  
  content_copy  
  
  implementation 'com.google.firebase:firebase-analytics-ktx'  
  content_copy  
}
```

## Conclusion:

### Error/Debug:

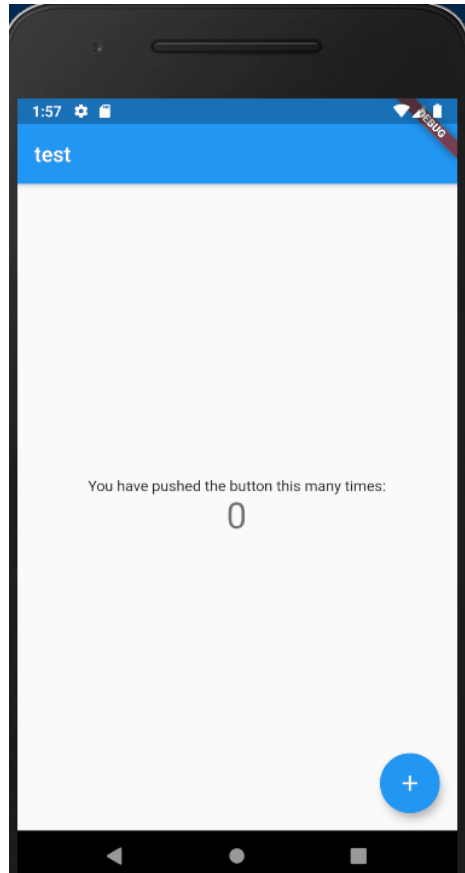
No Error

### Learning:

Establishing a real-time firebase database connection in a flutter project

## Output:





## Practical 6

Aim: Firebase Crud Operations(creating UI for crud operations)

Code:

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

void main() {
  runApp(MaterialApp(
    theme: ThemeData(
      brightness: Brightness.light,
      primaryColor: Colors.pink,
      colorScheme: ColorScheme.fromSwatch().copyWith(secondary: Colors.cyan),
```



```
    ),
    home: const MyApp(),
  ));
}

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

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  late String studentName, studentID, studyProgramID;
  late double studentGPA;

  getStudentName(name){
    this.studentName = name;
  }

  getStudentID(id){
    this.studentID = id;
  }

  getStudyProgramID(programID){
    this.studyProgramID = programID;
  }

  getStudentGPA(gpa){
    this.studentGPA = double.parse(gpa);
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Airbase inc."),
      ),
      body: Padding(padding: const EdgeInsets.all(16.0),
        child: Column(
          children: <Widget>[
            Padding(
              padding: const EdgeInsets.only(bottom:8.0),
              child: TextFormField(
                decoration: const InputDecoration(
                  labelText: "Student Name",
                  fillColor: Colors.white,
                  focusedBorder: OutlineInputBorder(
                    borderSide: BorderSide(color: Colors.blue, width: 2.0))),
                onChanged: (String name){
                  getStudentName(name);
                }
              )
            )
          ],
        ),
      ),
    );
  }
}
```

```
    },
  ),
),

Padding(
  padding: const EdgeInsets.only(bottom:8.0),
  child: TextFormField(
    decoration: const InputDecoration(
      labelText: "Student ID",
      fillColor: Colors.white,
      focusedBorder: OutlineInputBorder(
        borderSide: BorderSide(color: Colors.blue, width: 2.0))),
    onChanged: (String id){
      getStudentID(id);
    },
  ),
),

Padding(
  padding: const EdgeInsets.only(bottom:8.0),
  child: TextFormField(
    decoration: const InputDecoration(
      labelText: "Course code",
      fillColor: Colors.white,
      focusedBorder: OutlineInputBorder(
        borderSide: BorderSide(color: Colors.blue, width: 2.0))),
    onChanged: (String programID){
      getStudyProgramID(programID);
    },
  )
),

Padding(
  padding: const EdgeInsets.only(bottom:8.0),
  child: TextFormField(
    decoration: const InputDecoration(
      labelText: "GPA",
      fillColor: Colors.white,
      focusedBorder: OutlineInputBorder(
        borderSide: BorderSide(color: Colors.blue, width: 2.0))),
    onChanged: (String gpa){
      getStudentGPA(gpa);
    },
  )
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    ElevatedButton(
      child: Text('CREATE'),
```

```

        style: ElevatedButton.styleFrom(
          primary: Colors.green,
          //onPrimary: Colors.black,
        ),
        onPressed: () {
          createdata();
        },
      ),
      ElevatedButton(
        child: Text('READ'),
        style: ElevatedButton.styleFrom(
          primary: Colors.blue,
          //onPrimary: Colors.black,
        ),
        onPressed: () {
          readdata();
        },
      ),
      ElevatedButton(
        child: Text('UPDATE'),
        style: ElevatedButton.styleFrom(
          primary: Colors.deepOrangeAccent,
          //onPrimary: Colors.black,
        ),
        onPressed: () {
          updatedata();
        },
      ),
      ElevatedButton(
        child: Text('DELETE'),
        style: ElevatedButton.styleFrom(
          primary: Colors.red,
          //onPrimary: Colors.black,
        ),
        onPressed: () {
          deletedata();
        },
      ),
    ]),
  ),
  //TextFormField
  , //Widget
), //Column
);
}
void createdata() {
  print("created");
}
void readdata() {
  print("read");
}

```

---

```
}  
void updatedata() {  
  print("updated");  
}  
void deletedata() {  
  print("deleted");  
}  
}
```

**Conclusion:**

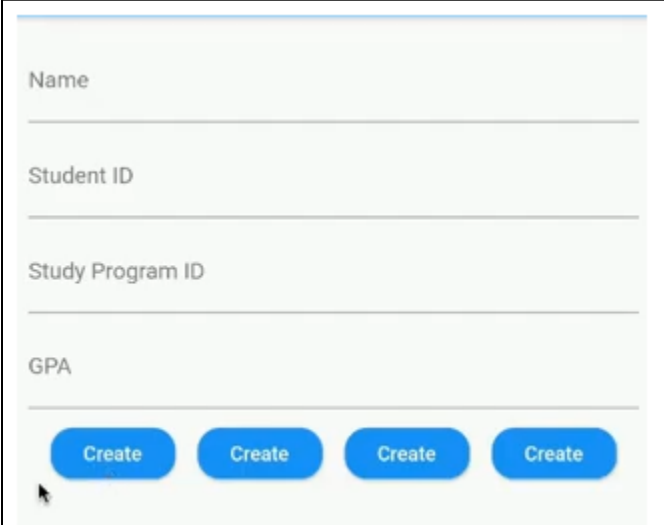
**Error/Debug:**

When attempting to retrieve data from a database in a Flutter app, an error occurs.

**Learning:**

Using crud operations to create, read, update, and delete data from a database.

**Output:**



A screenshot of a Flutter application interface for entering student data. The form consists of four text input fields stacked vertically, each with a label above it: "Name", "Student ID", "Study Program ID", and "GPA". Below the "GPA" field, there are four blue, rounded rectangular buttons, each labeled "Create". A mouse cursor is visible near the first "Create" button.

---

Airbase inc.

DEBUG

Student Name

Student ID

Course code

GPA

CREATE

READ

UPDATE

DELETE

## Practical 7 (a)

**Aim:** Curved Navigation Bar

**Code:**

```
import 'package:curved_navigation_bar/curved_navigation_bar.dart';
import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/container.dart';
import 'package:flutter/src/widgets/framework.dart';

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

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

```

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.pink.shade100,
      appBar: AppBar(
        title: Text(
          'Bottom Nav',
        ),
        backgroundColor: Colors.purple,
      ),

      //Bottom Nav bar
      bottomNavigationBar: CurvedNavigationBar(
        backgroundColor: Colors.pink.shade100,
        color: Colors.purple,
        //the time it takes to float
        //higher the no. more slow
        animationDuration: Duration(milliseconds: 300),
        items: <Widget>[
          //we are going to add icon

          Icon(
            Icons.home,
            color: Colors.white,
            size: 30,
          ),
          Icon(
            Icons.security,
            color: Colors.white,
            size: 30,
          ),
          Icon(
            Icons.add,
            color: Colors.white,
            size: 30,
          ),
          Icon(
            Icons.notifications,
            color: Colors.white,
            size: 30,
          ),
        ],

        onTap: (index) {
          print(index);

        },
      ),
    );
  }
}

```

---

**Conclusion:****Error/Debug:**

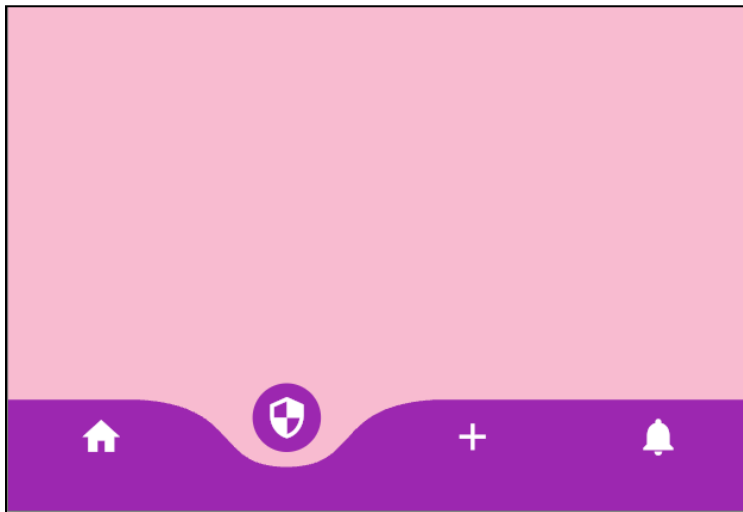
The curved Flutter navigation bar does not switch between pages sometimes.

**Learning:**

Adding a curved navigation bar to a flutter project

**Packages imported:**

curved\_navigation\_bar

**Output:**

## Practical 7 (b)

**Aim:** Stepper

**Code:**

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

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

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

  static const String _title = 'Flutter Code Sample';
```

```

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

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

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

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _index = 0;

  @override
  Widget build(BuildContext context) {
    return Stepper(
      currentStep: _index,
      onStepCancel: () {
        if (_index > 0) {
          setState(() {
            _index -= 1;
          });
        }
      },
      onStepContinue: () {
        if (_index <= 0) {
          setState(() {
            _index += 1;
          });
        }
      },
      onStepTapped: (int index) {
        setState(() {
          _index = index;
        });
      },
      steps: <Step>[
        Step(
          title: const Text('Step 1 title'),

```



---

```

        content: Container(
          alignment: Alignment.centerLeft,
          child: const Text('Content for Step 1')),
      ),
      const Step(
        title: Text('Step 2 title'),
        content: Text('Content for Step 2'),
      ),
    ],
  );
}
}

```

Code:

```

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: 'Stepper demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Stepper Demo'),
    );
  }
}

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

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

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

  List<Step> getSteps() => [
    Step(

```

---

```

        content: Text("Content 1"),
        title: const Text("Step 1"),
        isActive: currentStep >= 0,
    ),
    Step(
        content: Text("Content 2"),
        title: const Text("Step 2"),
        isActive: currentStep >= 1,
    ),
    Step(
        content: Text("Content 3"),
        title: const Text("Step 3"),
        isActive: currentStep >= 2,
    ),
    Step(
        content: Text("Content 4"),
        title: const Text("Step 4"),
        isActive: currentStep >= 3,
    ),
];

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: SafeArea(
      child: SingleChildScrollView(
        child: Stepper(
          steps: getSteps(),
          currentStep: currentStep,
          onStepCancel: () {
            setState(() {
              if(currentStep == 0){
                currentStep=currentStep;
              }
              else{
                currentStep --;
              }
            });
          },
          onStepContinue: () {
            setState(() {
              //the no. gives access to steps n-1
              if(currentStep == 3){
                currentStep = currentStep;
              }else{
                currentStep ++;
              }
            });
          },
        ),
      ),
    ),
  );
}

```

---

```
    }  
    });  
  },  
  
  onStepTapped: (int currentStep) {  
    setState(() {  
      currentStep=currentStep;  
    });  
  },  
),  
))  
// This trailing comma makes auto-formatting nicer for build methods.  
);  
}  
}
```

## Conclusion:

### Error/Debug:

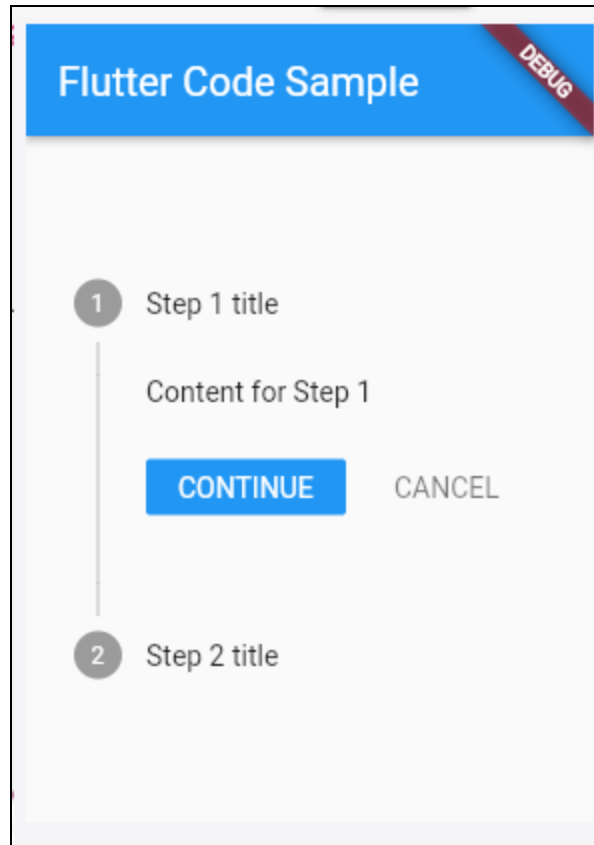
Stepper crashes on hot reload, and horizontal change not always working

### Learning:

Make a stepper widget that shows progress through a series of steps.

## Output:

---



---

## Practical 8

Aim: Actual firebase connection for CRUD operations

Code:

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_core/firebase_core.dart';
// import 'package:firebase_auth/firebase_auth.dart';
// Practical 8
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: const FirebaseOptions(
      apiKey: "AlzaSyC3GJGnXlg1vgNxLu2jlvzM5tP7EmSb5bY",
      appId: "1:11682018483:android:5c0773c5b130e2d7436cc6",
      messagingSenderId: "11682018483",
      projectId: "crud-pracs9",
    ),
  );
  runApp(MaterialApp(
    theme: ThemeData(
      brightness: Brightness.light,
      primaryColor: Colors.blue,
      colorScheme: ColorScheme.fromSwatch().copyWith(secondary: Colors.cyan),
    ),
    home: MyApp(),
  ));
}

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

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  late String studentName, studentID, studyProgramID;
  late String studentGPA;

  getStudentName(name) {
    this.studentName = name;
  }

  getStudentID(id) {
    this.studentID = id;
  }
}
```

---

```
getStudyProgramID(programID) {
  this.studyProgramID = programID;
}

getStudentGPA(gpa) {
  this.studentGPA = gpa;
}

createData() async {
  print("create");

  CollectionReference colref =
    FirebaseFirestore.instance.collection('MyStudents');
  //create map
  Map<String, dynamic> student_map = {
    "studentName": studentName,
    "studentID": studentID,
    "studentGPA": studentGPA,
    "studyProgramID": studyProgramID,
  };

  colref
    .add(student_map)
    .then((value) => print("User Added"))
    .catchError((error) => print("Failed to add user: $error"));
}

readData() {
  print("Read");
}

updateData() async {
  print("Update");

  CollectionReference colref =
    FirebaseFirestore.instance.collection('MyStudents');

  Map<String, dynamic> student_map = {
    "studentName": studentName,
    "studentID": studentID,
    "studentGPA": studentGPA,
    "studyProgramID": studyProgramID,
  };

  colref
    .doc('mtTt8BzAL3mbW2coYRIP')
    .set(student_map)
    .then((value) => print("User updated"))
    .catchError((error) => print("Failed to update user: $error"));
```

```

}

deleteData() {
  print("Delete");

  CollectionReference colref =
    FirebaseFirestore.instance.collection('MyStudents');

  colref.doc('8KD1GRXfxFSRH3MoWnqu').delete().then((colref) => print("Document
Deleted"),
    onError: (e)=>print("Error deleting the document=$e")
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Airbase inc."),
    ),
    body: Padding(
      padding: const EdgeInsets.all(8.0),
      child: Column(
        children: <Widget>[
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: TextFormField(
              decoration: const InputDecoration(
                labelText: "Student Name",
                fillColor: Colors.white,
                focusedBorder: OutlineInputBorder(
                  borderSide:
                    BorderSide(color: Colors.blue, width: 2.0))),
              onChanged: (String name) {
                getStudentName(name);
              },
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: TextFormField(
              decoration: const InputDecoration(
                labelText: "Student ID",
                fillColor: Colors.white,
                focusedBorder: OutlineInputBorder(
                  borderSide:
                    BorderSide(color: Colors.blue, width: 2.0))),
              onChanged: (String id) {
                getStudentID(id);
              },
            ),
          ),
        ],
      ),
    ),
  );
}

```

```

    ),
  ),
  Padding(
    padding: const EdgeInsets.all(16.0),
    child: TextFormField(
      decoration: const InputDecoration(
        labelText: "Course code",
        fillColor: Colors.white,
        focusedBorder: OutlineInputBorder(
          borderSide:
            BorderSide(color: Colors.blue, width: 2.0))),
      onChanged: (String programID) {
        getStudyProgramID(programID);
      },
    ),
  ),
  Padding(
    padding: const EdgeInsets.all(16.0),
    child: TextFormField(
      decoration: const InputDecoration(
        labelText: "GPA",
        fillColor: Colors.white,
        focusedBorder: OutlineInputBorder(
          borderSide:
            BorderSide(color: Colors.blue, width: 2.0))),
      onChanged: (String gpa) {
        getStudentGPA(gpa);
      },
    ),
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
      ElevatedButton(
        onPressed: (() {
          createData();
        })),
      child: Text("CREATE"),
      style: ElevatedButton.styleFrom(
        primary: Colors.green,
      ),
    ),
    ElevatedButton(
      onPressed: (() {
        readData();
      })),
      child: Text("Read"),
      style: ElevatedButton.styleFrom(
        primary: Colors.blue,
      ),
    ),
  ),

```



```

ElevatedButton(
  onPressed: () {
    updateData();
  },
  child: Text("Update"),
  style: ElevatedButton.styleFrom(
    primary: Colors.amber,
  )),
ElevatedButton(
  onPressed: () {
    deleteData();
  },
  child: Text("Delete"),
  style: ElevatedButton.styleFrom(
    primary: Colors.red,
  )),
),
),
),
);
}
}
}

```

Cloud Firestore

NEW

[Data](#)
[Rules](#)
[Indexes](#)
[Usage](#)

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

Configure App Check

Panel view

Query builder

Home

>

MyStudents


>

13wrKTbtR4IIBu.

crud-pracs9	MyStudents	13wrKTbtR4IIBuJMYOje
+ Start collection	+ Add document	+ Start collection
MyStudents >	13wrKTbtR4IIBuJMYOje >	+ Add field
	Jenis Winsta PLYIM0HljASBvame91oE gS3mMTdcD1is27SUAPfj hFaumSbkzn1YBxHjUNEi	studentGPA: "2.5" studentID: "124" studentName: "jenis" studyProgramID: "125"


# Cloud Firestore NEW

[Data](#) [Rules](#) [Indexes](#) [Usage](#)

 Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing [Configure App Check](#)


[Panel view](#) [Query builder](#)

 > MyStudents > zB6ANxe0OoWl.

 crud-pracs9

[+ Start collection](#)

MyStudents >

 MyStudents

[+ Add document](#)

13wrKTbtr4IIBuJMY0je

K4qUfrNpvC2oNg7P0NYN

PLyIM0HljASBvame91oE

gS3mMTdcDiis27SUAPfj

hFaumSbkzn1YBxHjUNEi

zB6ANxe0OoWIamI6qFAW >

 zB6ANxe0OoWIamI6qFAW

[+ Start collection](#)

[+ Add field](#)

studentGPA: "5"

studentID: "241587"

studentName: "Ram"

studyProgramID: "7"

Read

```
{studentGPA: 5, studentID: 241587, studyProgramID: 7, studentName: Ram}
```

---

## Practical 9

Aim: Drawer for navigation between pages

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'package:prac9/Acadimics.dart';
import 'package:prac9/ProjectPage.dart';

import 'AOI.dart';
import 'AboutPage.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: 'Practical 9',
      theme: ThemeData(
        primarySwatch: Colors.purple,
      ),
      home: const MyHomePage(title: 'Profile Page'),
    );
  }
}

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

  final String title;

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

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
```

---

```

    title: Text(widget.title),
  ),
  drawer: Drawer(
    child: ListView(
      padding: EdgeInsets.zero,
      children: <Widget>[
        DrawerHeader(
          decoration: BoxDecoration(
            gradient: LinearGradient(
              colors: [Colors.pink, Colors.purple],
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
            ),
          ),
        ),
        child: Container(
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: const <Widget>[
                CircleAvatar(
                  foregroundImage:
                    NetworkImage("https://tinyurl.com/2bhssn68"),
                  radius: 30,
                ),
                Text("Rhea Yadav",
                  style: TextStyle(
                    fontSize: 18,
                    color: Colors.white,
                    fontWeight: FontWeight.bold)),
              ],
            )),
        ),
      ),
    ),
  ),
  Container(
    // decoration: BoxDecoration(
    //   color: Colors.pink.shade300,
    //   // gradient: LinearGradient(
    //   //   colors: [Colors.purple, Colors.white, Colors.pink],
    //   //   begin: Alignment.topLeft,
    //   //   end: Alignment.bottomRight,
    //   //   ),
    //   // ),
    child: Center(
      child: Card(
        child: Column(children: <Widget>[
          ListTile(
            leading: Icon(Icons.note_alt_outlined),
            title: Text('ABOUT' ,style: TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
            trailing: Icon(Icons.arrow_circle_right),
            tileColor: Colors.pink.shade300,

```

```

onTap: () {
  Navigator.of(context).push(
    MaterialPageRoute(
      builder: (context) => const About(),
    ),
  );
},
),
Divider(),
ListTile(
  leading: Icon(Icons.group),
  title: Text('PROJECTS',style: TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
  trailing: Icon(Icons.arrow_circle_right),
  tileColor: Colors.pink.shade300,
  onTap: () {
    Navigator.of(context).push(
      MaterialPageRoute(
        builder: (context) => const Project(),
      ),
    );
  },
),
Divider(),
ListTile(
  leading: Icon(Icons.book),
  title: Text('ACADIMICS',style: TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
  trailing: Icon(Icons.arrow_circle_right),
  tileColor: Colors.pink.shade300,
  onTap: () {
    Navigator.of(context).push(
      MaterialPageRoute(
        builder: (context) => const Acadimics(),
      ),
    );
  },
),
Divider(),
ListTile(
  leading: Icon(Icons.interests),
  title: Text('AOI',style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white)),
  trailing: Icon(Icons.arrow_circle_right),
  tileColor: Colors.pink.shade300,
  onTap: () {
    Navigator.of(context).push(
      MaterialPageRoute(
        builder: (context) => const AOI(),
      ),
    );
  },
),

```

---

```
    },
  ),
  ],
)
))
],
),
),
body: SafeArea(
  child: SingleChildScrollView(
    child: Container(
      child: Center(
        child: Text("This is my home page"),
      ),
    ),
  ),
),
);
}
```

aboutpage.dart:

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

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

  @override
  State<About> createState() => _AboutState();
}

class _AboutState extends State<About> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Text("About PAGE"),
    );
  }
}
```

---

---

Academics.dart:

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

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

  @override
  State<About> createState() => _AboutState();
}

class _AboutState extends State<About> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Text("About PAGE"),
    );
  }
}
```

ProjectPage.dart:

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

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

  @override
  State<Project> createState() => _ProjectState();
}

class _ProjectState extends State<Project> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Text("Project PAGE"),
    );
  }
}
```

---

---

AOI.dart:

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

import 'package:flutter/material.dart';

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

  @override
  State<AOI> createState() => _AOIState();
}

class _AOIState extends State<AOI> {
  @override
  Widget build(BuildContext context) {
    return const Scaffold(
      body: Text("AOI PAGE"),
    );
  }
}
```

## Conclusion:

### Error/Debug:

Stepper crashes on hot reload, and horizontal change not always working

### Learning:

Make a stepper widget that shows progress through a series of steps.

---



---

**Output:**



---

# Practical 10

Aim: Audio file

Code:

```
import 'package:flutter/material.dart';
//import 'dart:typed_data';
import 'package:audioplayers/audioplayers.dart';
//import 'package:flutter/services.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,
      ),
      home: const Birds_Home());
  }
}

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

  @override
  State<Birds_Home> createState() => _Birds_HomeState();
}

class _Birds_HomeState extends State<Birds_Home> {
  AudioPlayer player = AudioPlayer();
  bool isPlaying = false;
  Duration duration = Duration.zero; //duration for how long to be played
  Duration position = Duration.zero; //current position

  @override
  void initState() {
    super.initState();

    setAudio();
  }
}
```

---

```
//Listen to the states:playing, paused, stopped
player.onPlayerStateChanged.listen((state) {
  setState(() {
    isPlaying = state == PlayerState.playing;
  });
});

//Listen to audio duration
player.onDurationChanged.listen((newDuration) {
  setState(() {
    duration = newDuration;
  });
});

//Listen to audio position changers
player.onPositionChanged.listen((newPosition) {
  setState(() {
    position = newPosition;
  });
});
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("Play the sound of Wild Babbler bird"),
      backgroundColor: Colors.redAccent,
    ),
    body: Padding(
      padding: const EdgeInsets.all(20),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          ClipRRect(
            borderRadius: BorderRadius.circular(20),
            child: Image.network(
              'https://cdn.pixabay.com/photo/2022/10/27/13/26/yellow-billed-babbler-7550869__340.jpg',
              width: double.infinity,
              height: 350,
              fit: BoxFit.cover,
            ),
          ),
          const SizedBox(height: 32),
          const Text(
            'the Wild Babbler sings',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
```

```

    ),
  ),
  const SizedBox(height: 4),
  const Text(
    'free source',
    style: TextStyle(fontSize: 20),
  ),
  Slider(
    min: 0,
    max: duration.inSeconds.toDouble(),
    value: position.inSeconds.toDouble(),
    onChanged: (value) async {
      final position = Duration(seconds: value.toInt());
      await player.seek(position);

      //optional: Play audio if it was paused
      await player.resume();
    },
  ),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 16.0),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text(formatTime(position)),
        Text(formatTime(duration)),
      ],
    ),
  ),
  CircleAvatar(
    radius: 35,
    child: IconButton(
      icon: Icon(
        isPlaying ? Icons.pause : Icons.play_arrow,
      ),
      iconSize: 50,
      onPressed: () async {
        if (isPlaying) {
          await player.pause();
        } else {
          //String url = "assets/audio/wild_babbler_bird.mp3";
          //await player.play(url);
          final myplayer = AudioCache();
          //final url = await myplayer.load('wild_babbler_bird.mp3');
          //player.setSourceUrl(url.path);
          await myplayer.load('assets/audio/wild_babbler_bird.mp3');
        }
      },
    ),
  )
)

```

```

    ],
  ),
),
);
}

String formatTime(Duration duration) {
  String twoDigits(int n) => n.toString().padLeft(2, '0');
  final hours = twoDigits(duration.inHours);
  final minutes = twoDigits(duration.inMinutes.remainder(60));
  final seconds = twoDigits(duration.inSeconds.remainder(60));

  return [
    if (duration.inHours > 0) hours,
    minutes,
    seconds,
  ].join(':');
}

Future setAudio() async {
  //Repeat song when completed
  player.setReleaseMode(ReleaseMode.loop);

  //load audio from url code goes here
  // String url = "assets/audio/wild_babbler_bird.mp3";
  //player.setSourceUrl(url);

  //load audio from file using file picker
  //final file = File(...);
  //player.setSourceUrl(file.path, isLocal:true);

  final myplayer = AudioCache(prefix: 'assets/audio/');
  final url = await myplayer.load('wild_babbler_bird.mp3');
  player.setSourceUrl(url.path);
}
}

```

## Conclusion:

### Error/Debug:

Stepper crashes on hot reload, and horizontal change not always working

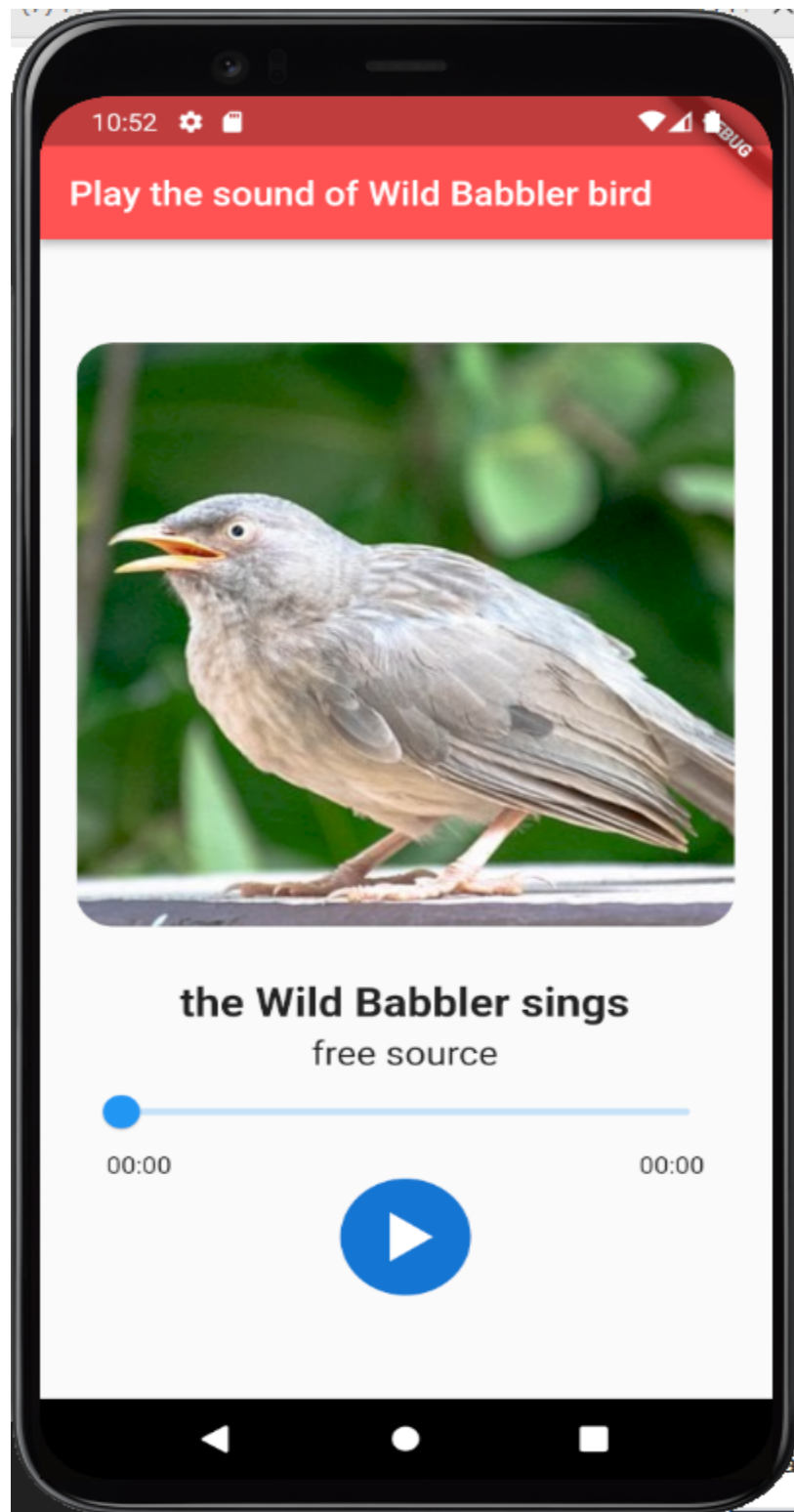
### Learning:

Make a stepper widget that shows progress through a series of steps.

### Dependencies Imported:

Make a stepper widget that shows progress through a series of steps.

Output:



---

# Practical 11

Aim: Text files

Code:

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'dart:async';
import 'dart:io';
import 'package:path_provider/path_provider.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Kindacode.com',
      home: HomePage(),
    );
  }
}

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

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

class _HomePageState extends State<HomePage> {
  // This will be displayed on the screen
  String? _content;

  // Find the Documents path
  Future<String> _getDirPath() async {
    final dir = await getApplicationDocumentsDirectory();
    return dir.path;
  }

  // This function is triggered when the "Read" button is pressed
  Future<void> _readData() async {
    final dirPath = await _getDirPath();
```

```

final myFile = File('$dirPath/data.txt');
final data = await myFile.readAsString(encoding: utf8);
setState() {
  _content = data;
});
}

// TextField controller
final _textController = TextEditingController();
// This function is triggered when the "Write" button is pressed
Future<void> _writeData() async {
  final _dirPath = await _getDirPath();

  final _myFile = File('$_dirPath/data.txt');
  // If data.txt doesn't exist, it will be created automatically

  await _myFile.writeAsString(_textController.text);
  _textController.clear();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Kindacode.com'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(20),
      child: Column(
        children: [
          TextField(
            controller: _textController,
            decoration: const InputDecoration(labelText: 'Enter your name'),
          ),
          ElevatedButton(
            child: const Text('Save to file'),
            onPressed: _writeData,
          ),
          const SizedBox(
            height: 150,
          ),
          Text(_content ?? 'Press the button to load your name',
            style: const TextStyle(fontSize: 24, color: Colors.pink)),
          ElevatedButton(
            child: const Text('Read my name from the file'),
            onPressed: _readData,
          ),
        ],
      ),
    ),
  ),
)

```



---

```
}  
}  
};
```

### **Conclusion:**

#### **Error/Debug:**

Stepper crashes on hot reload, and horizontal change not always working

#### **Learning:**

Make a stepper widget that shows progress through a series of steps.

#### **Dependencies Imported:**

Make a stepper widget that shows progress through a series of steps.

### **Output:**

---