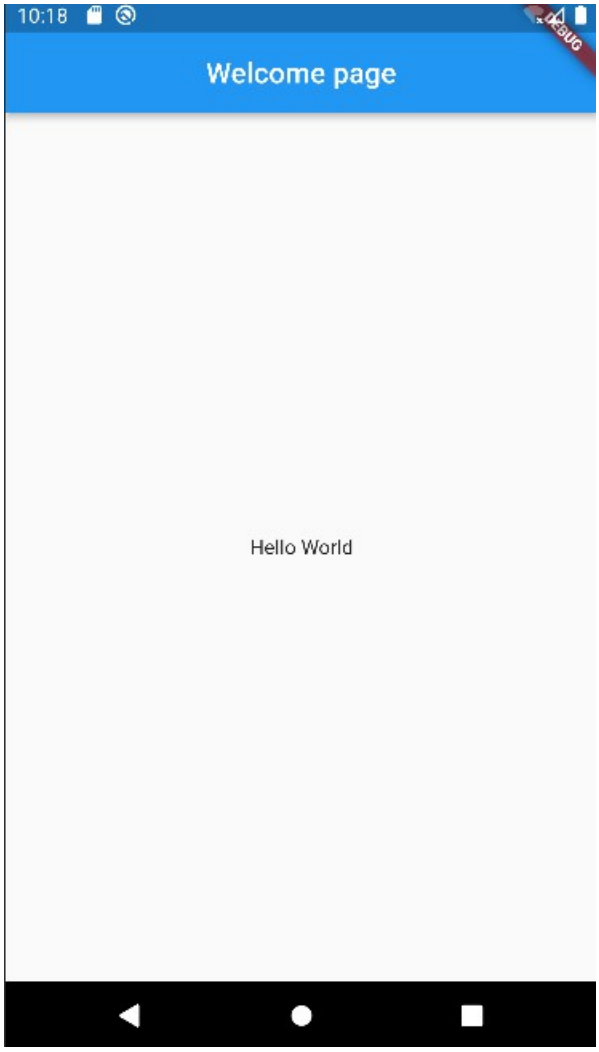
Code:

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

void main()=>runApp(MaterialApp(
  home:Scaffold(
    appBar:AppBar(
      title:const Text('Welcome page'),
      centerTitle:true,
    ),
    body:Center(
        child:Text(
            "Hello World",
        ),
)
  ),
));
```

Output:



Code:

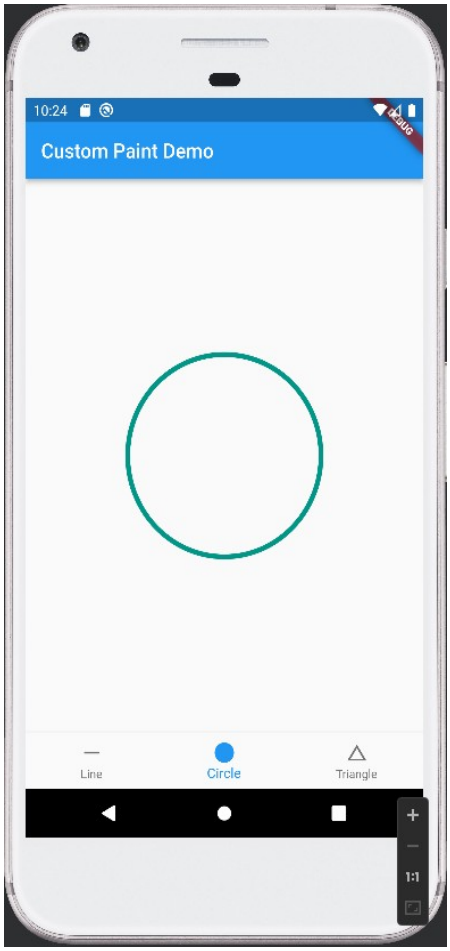
```
import 'package:flutter/material.dart';
import 'dart:math' as math;

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

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

  // This widget is the root of your application.
  static const String _title= 'Flutter Sample';
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(

      title:_title,
      home:StatefulShapes(),
    );
  }
}

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

  @override
  State<StatefulShapes> createState() => _StatefulShapesState();
}

class _StatefulShapesState extends State<StatefulShapes> {
  int _selectedIndex=0;
  static const List<Widget> _widgetOptions=<Widget>[
    CustomPaint(
      size:Size(300,200),
      painter: LinePainter(),
    ),

    CustomPaint(
      size:Size(300,200),
      painter: CirclePainter(),
    ),

    CustomPaint(
      size:Size(300,200),
      painter: TrianglePainter(),
```

```
    )
  ];

  void _onItemTapped(int index){
    setState((){
      _selectedIndex=index;
    });
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar:AppBar(
          title:const Text("Custom Paint Demo"),
        ),
      body:Center(
        child:_widgetOptions.elementAt(_selectedIndex),
      ),
      bottomNavigationBar: BottomNavigationBar(
        items:const<BottomNavigationBarItem>[
          BottomNavigationBarItem(
              icon: Icon(Icons.horizontal_rule),
              label:'Line'
          ),
          BottomNavigationBarItem(
              icon: Icon(Icons.circle),
              label:'Circle'),
          BottomNavigationBarItem(
              icon: Icon(Icons.change_history),
              label:'Triangle'),
        ],
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.blue,
        onTap: _onItemTapped,
      ),
    );
  }
}



class LinePainter extends  CustomPainter {

  const LinePainter() ;
  @override
  void paint(Canvas canvas,Size size){
    var paint = Paint()
      ..color = Colors.teal
```

```
      ..strokeWidth = 15;

      Offset start=Offset(0,size.height/2);
      Offset end=Offset(size.width,size.height/2);
      canvas.drawLine(start,end,paint);
  }

  @override
  bool shouldRepaint(covariant CustomPainter oldDelegate) {
    return false;
  }


}

class CirclePainter extends CustomPainter{
  const CirclePainter() ;
  @override
  void paint(Canvas canvas,Size size){
    var paint=Paint()
        ..color=Colors.teal
        ..strokeWidth=5
        ..style=PaintingStyle.stroke
        ..strokeCap=StrokeCap.round;
    Offset center=Offset(size.width/2,size.height/2);
    canvas.drawCircle(center,100,paint);
  }
  @override
  bool shouldRepaint(covariant CustomPainter oldDelegate){
    return false;

  }
}

class TrianglePainter extends CustomPainter{
  const TrianglePainter() ;
  @override
  void paint(Canvas canvas, Size size) {
    // TODO: implement paint
    var paint=Paint()
        ..color=Colors.teal
        ..strokeWidth=5
        ..style=PaintingStyle.stroke
        ..strokeCap=StrokeCap.round;
    var path=Path();
    var angle=(math.pi *2)/3;
    var radius=100;
    Offset center=Offset(size.width/2,size.height/2);
    Offset startPoint=Offset(radius*math.cos(0.0),radius*math.sin(0.0));
```

```
    path.moveTo(startPoint.dx+center.dx,startPoint.dy+center.dy);
    for(int i=1;i<=3;i++){
      double x=radius*math.cos(angle*i)+center.dx;
      double y=radius*math.sin(angle*i)+center.dy;
      path.lineTo(x,y);
    }
    path.close();
    canvas.drawPath(path,paint);


  }

  @override
  bool shouldRepaint(covariant CustomPainter oldDelegate) {
    // TODO: implement shouldRepaint
    return false;
  }

}
```

Output:


10:24
Custom Paint Demo
Line
Circle
Triangle


10:23
Custom Paint Demo
Line
Circle
Triangle


10:25
DEBUG
Custom Paint Demo
Line
Circle
Triangle
1:1

Code:

main.dart

```
import 'package:flutter/material.dart';
import 'package:mad_ex3/success.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(
        title: 'Forms Data',
        home: PersonalData()
    );
  }
}

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

  @override
  State<PersonalData> createState() => _PersonalDataState();
}

class _PersonalDataState extends State<PersonalData> {
  late String _name;
  late String _email;
  late String _password;
  late String _phoneNumber;

  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();

  Widget _buildName() {
    return TextFormField(
      decoration: const InputDecoration(
        icon: Icon(Icons.person),
        hintText: 'What do people call you?',
        labelText: 'Name *',
      ),
      validator: (String? value) {
        if (value == null){
          return 'Name is required';
```

```
      }
      return null;
    },
    onSaved: (String? value) {


      _name = value!;

    },

  );
}

Widget _buildEmail() {
  return TextFormField(
      decoration: const InputDecoration(
        icon: Icon(Icons.email),
        hintText: '',
        labelText: 'Email *',
      ),
      validator: (String? value) {
        if (value == null || value.isEmpty){
          return 'Email is required';
        }
        return null;
      },
      onSaved: (value) {

        _email = value!;

      }

  );
}

//
Widget _buildPassword() {
  return TextFormField(
    decoration: const InputDecoration(
      icon: Icon(Icons.person),
      hintText: 'The secret Identity',
      labelText: 'Password *',
    ),
    validator: (String? value) {
      if (value == null || value.isEmpty){
        return 'password is required';
      }
      return null;
    },
```

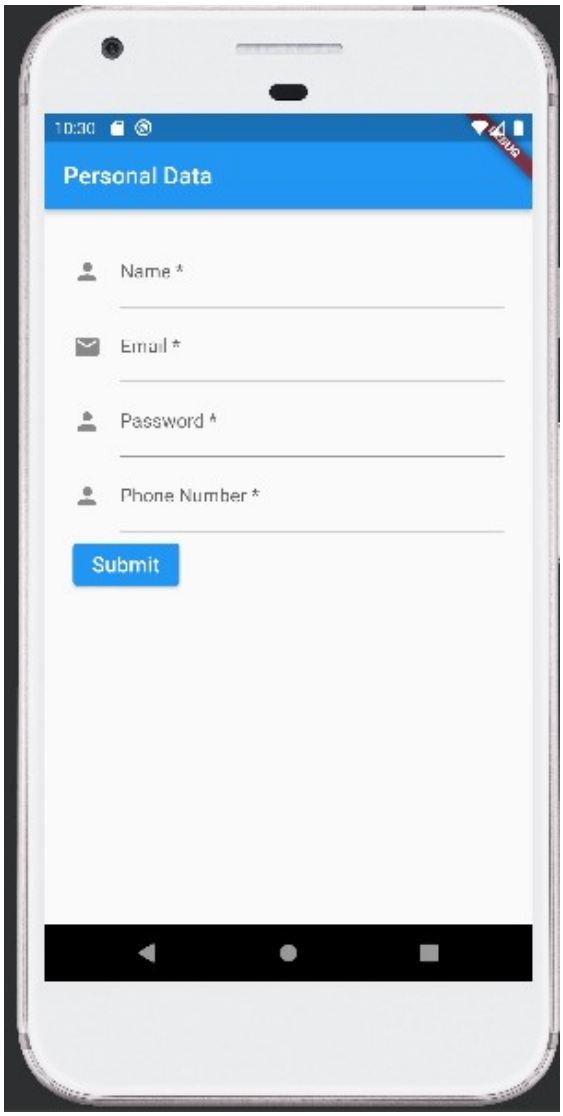
```
    onSaved: ( value) {
       _password = value!;

    },

  );
}

Widget _buildPhoneNumber() {
  return TextFormField(
    decoration: const InputDecoration(
      icon: Icon(Icons.person),
      hintText: 'The one call away',
      labelText: 'Phone Number *',
    ),
    validator: (String? value) {
      if (value == null || value.isEmpty){
        return 'phone number is required';
      }
      return null;
    },
    onSaved: (value) {
       _phoneNumber=value!;
    },

  );
}



@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Personal Data')),
    body: Container(
       margin: const EdgeInsets.all(24),
       child: Form(
         key:_formKey,
         child: Wrap(
            direction: Axis.horizontal,
            spacing: 8.0, // gap between adjacent chips
            runSpacing: 4.0,
            children:<Widget>[
            _buildName(),
         _buildEmail(),
         _buildPassword(),
         _buildPhoneNumber(),
         ElevatedButton(

              onPressed: (){
```

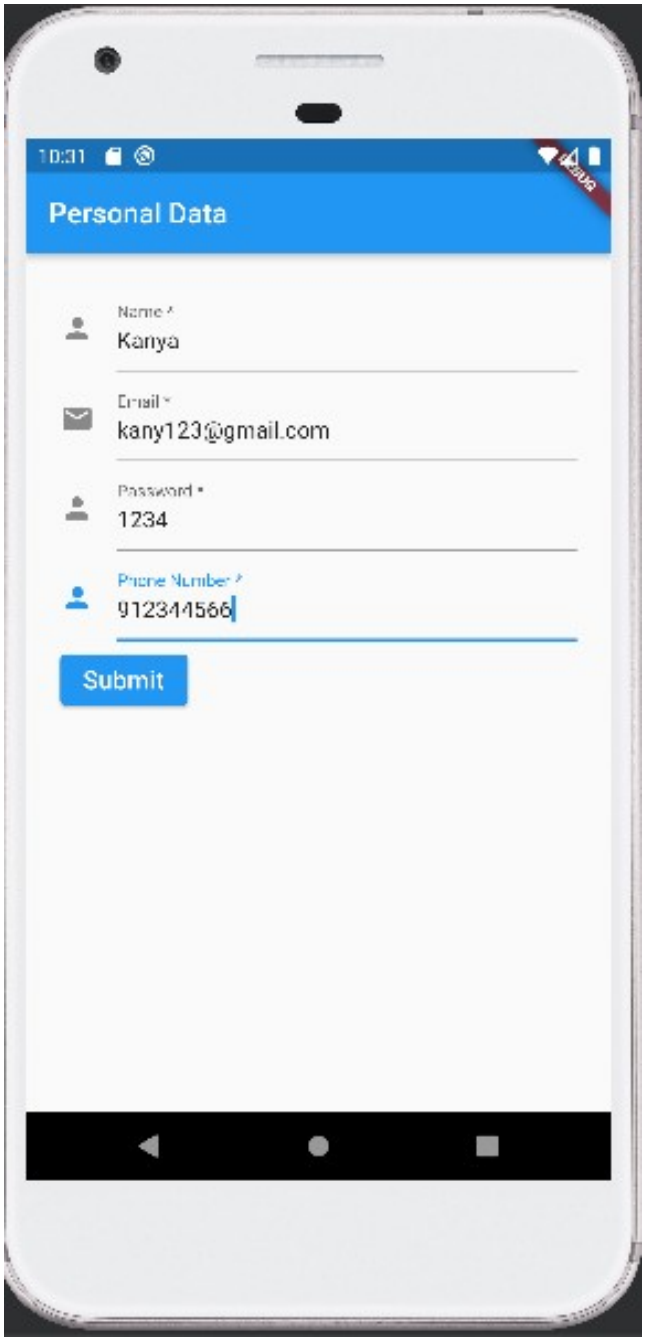
```
                if (_formKey.currentState!=null && _formKey.currentState!.validate()) {
                  _formKey.currentState!.save();
                  _formKey.currentState!.reset();
                  ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Data stored')),
                  );
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context)=>Success(

name:_name,email:_email,password:_password,phoneNumber:_phoneNumber),

                    )

                  );

                }
              },


            child: const Text(
                'Submit',
                style: TextStyle(
                    fontSize: 18,
                    color: Colors.white)
            ),
          ),


          ]
      ),
    ),)
    ,

    );

  }
}
```

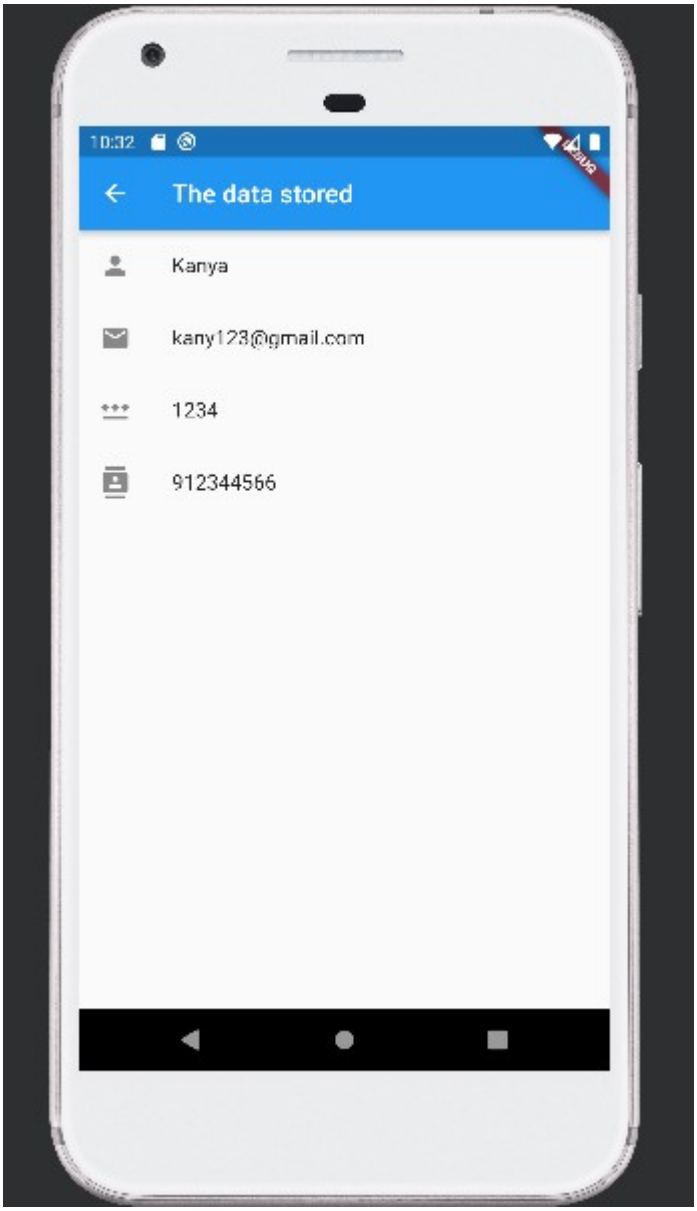
Success.dart

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

class Success extends StatelessWidget {
  final String name;
  final String email;
  final String password;
  final String phoneNumber;


  const Success(
      {Key? key, required this.name,
       required this.email,
       required this.password,
       required this.phoneNumber
      }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title:Text('The data stored')),
      body: ListView(
        children: <Widget>[
          ListTile(
            leading: Icon(Icons.person),
            title: Text(name),
          ),
          ListTile(
            leading: Icon(Icons.email),
            title: Text(email),
          ),
          ListTile(
            leading: Icon(Icons.password),
            title: Text(password),
          ),
          ListTile(
            leading: Icon(Icons.contacts),
            title: Text(phoneNumber),
          ),
        ],
      ),
    );
  }
}
```

Output:


10:30
Personal Data
Name *
Email *
Password *
Phone Number *
Submit


10:31
Personal Data
Name *
Kanya
Email *
kany123@gmail.com
Password *
1234
Phone Number *
9123441566
Submit


10:32
The data stored
Kanya
kany123@gmail.com
1234
912344566

Code:

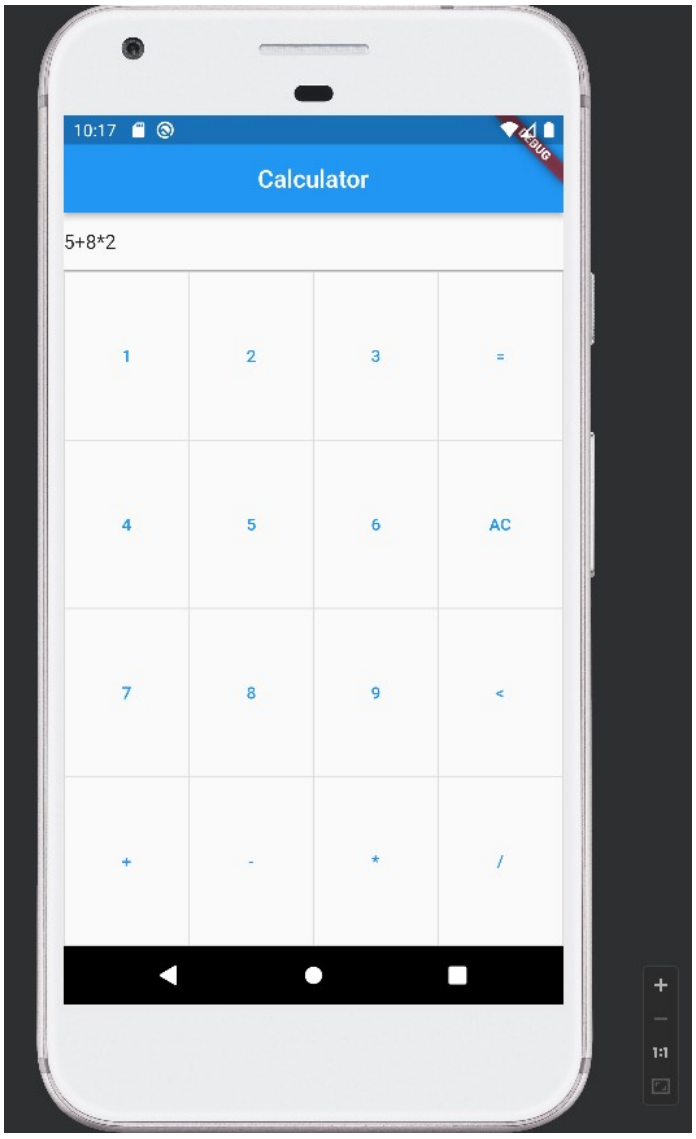
```
import 'package:flutter/material.dart';
import 'package:flutter_grid_button/flutter_grid_button.dart';
import 'package:math_expressions/math_expressions.dart';

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: "A Simple Calculator",
        theme: ThemeData(
          primarySwatch: Colors.blue,
        ),
      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(
          "Calculator"
        ),
        centerTitle: true,
      ),
      body: const Calculator(),
    );
  }
}

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

  @override
  State<Calculator> createState() => _CalculatorState();
```

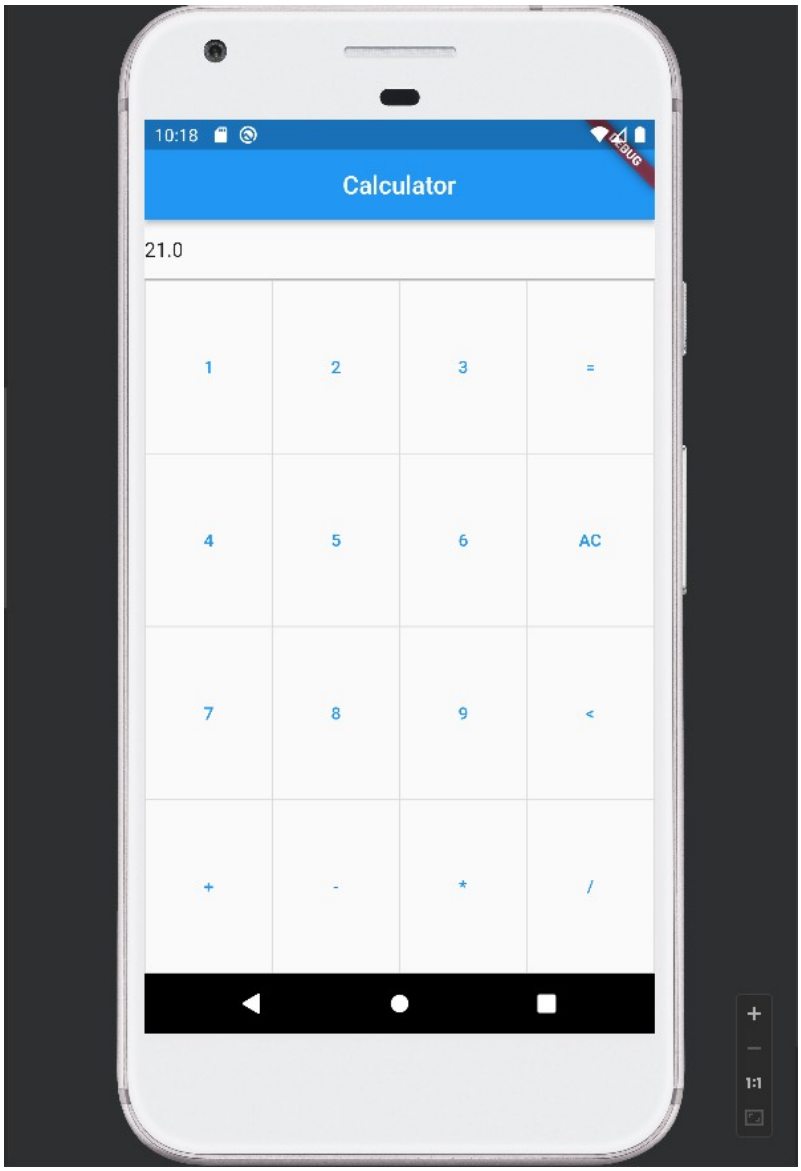
```
}

class _CalculatorState extends State<Calculator> {
  final inputFieldController = TextEditingController();
  final specialFunctionObj = _SpecialFunctions();

  @override
  void dispose() {
    // Clean up the controller when the widget is removed from the
    // widget tree.
    inputFieldController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        TextField(
          controller: inputFieldController,
          readOnly: true,
        ),
        Expanded(
            child: GridButton(
              onPressed: (dynamic value) {
                value = value.toString();
                // all clear
                if (value == "AC") {
                  inputFieldController.text = "";
                }
                // backspace
                else if (value == "<") {
                  inputFieldController.text =
specialFunctionObj.backspace(inputFieldController.text);
                }
                else if (value == "=") {
                  inputFieldController.text = specialFunctionObj.evaluate(inputFieldController.text);
                }
                else {
                  inputFieldController.text += value;
                }
              },
              items: const [
                [
                  GridButtonItem(
                    title: "1",
                  ),
                  GridButtonItem(
                    title: "2",
```

```
  ),
  GridButtonItem(
    title: "3",
  ),
  GridButtonItem(
    title: "="
  ),
],
[
  GridButtonItem(
    title: "4",
  ),
  GridButtonItem(
    title: "5",
  ),
  GridButtonItem(
    title: "6",
  ),
  GridButtonItem(
    title: "AC"
  ),
],
[
  GridButtonItem(
    title: "7",
  ),
  GridButtonItem(
    title: "8",
  ),
  GridButtonItem(
    title: "9",
  ),
  GridButtonItem(
     title: "<"
  ),
],
[
  GridButtonItem(
    title: "+",
  ),
  GridButtonItem(
    title: "-",
  ),
  GridButtonItem(
    title: "*",
  ),
  GridButtonItem(
     title: "/"
  ),
```

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

class _SpecialFunctions {
  String evaluate(String expression) {
    Parser p = Parser();
    Expression exp = p.parse(expression);
    ContextModel cm = ContextModel();
    double eval = exp.evaluate(EvaluationType.REAL, cm);
    String answer = '$eval';
    return answer;
  }

  String backspace(String expression) {
    int n = expression.length;
    if (n >= 1) {
      return expression.substring(
          0,
          n - 1
      );
    }
    else {
      return "";
    }
  }
}
```

Output:





Printed till this

Code:

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

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return const MaterialApp(title: 'Forms Data', home: PersonalData());
  }
}

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

class _PersonalDataState extends State<PersonalData> {
  final _database = FirebaseDatabase.instance.ref();
  late String _name;
  late String _email;
  late String _password;
  late String _phoneNumber;
  late  DataSnapshot _data;
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  var newPostKey;
  @override
  void initState() {
    // TODO: implement initState
    super.initState();

  }
  Widget _buildName() {
    return TextFormField(
      decoration: const InputDecoration(
```

```
      icon: Icon(Icons.person),
      hintText: 'What do people call you?',
      labelText: 'Name *',
    ),
    validator: (String? value) {
      if (value == null) {
        return 'Name is required';
      }
      return null;
    },
    onSaved: (String? value) {
      _name = value!;
    },
  );
}

Widget _buildEmail() {
  return TextFormField(
      decoration: const InputDecoration(
        icon: Icon(Icons.email),
        hintText: '',
        labelText: 'Email *',
      ),
      validator: (String? value) {
        if (value == null || value.isEmpty) {
          return 'Email is required';
        }
        return null;
      },
      onSaved: (value) {
        _email = value!;
      });
}

//
Widget _buildPassword() {
  return TextFormField(
    decoration: const InputDecoration(
      icon: Icon(Icons.person),
      hintText: 'The secret Identity',
      labelText: 'Password *',
    ),
    validator: (String? value) {
      if (value == null || value.isEmpty) {
        return 'password is required';
      }
      return null;
    },
    onSaved: (value) {
```

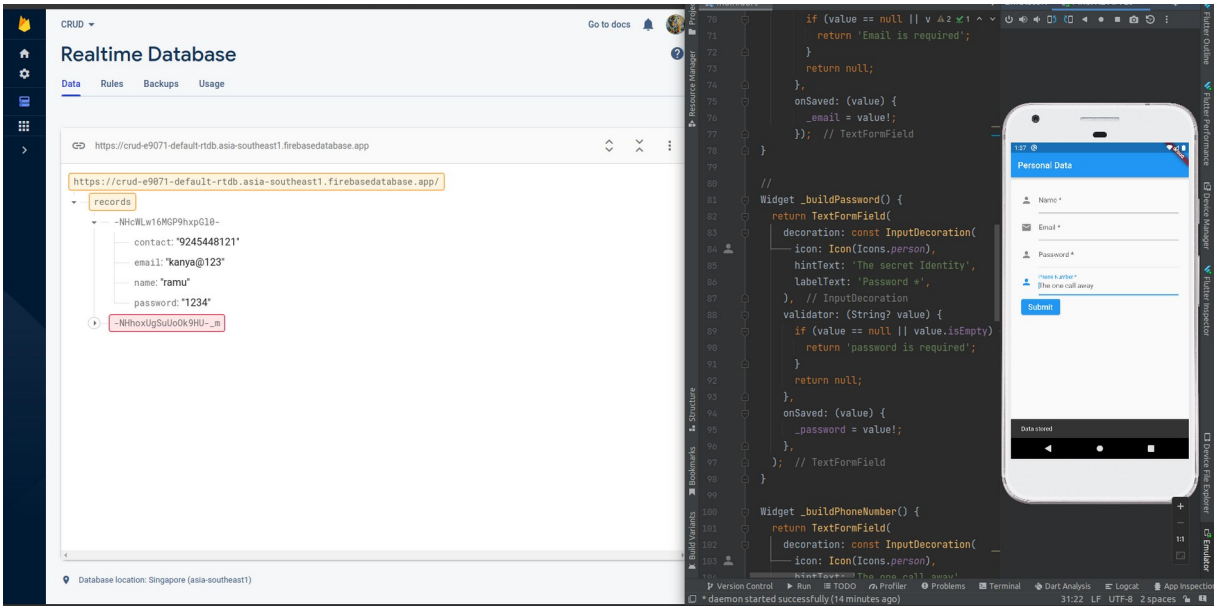
```
      _password = value!;
    },
  );
}

Widget _buildPhoneNumber() {
  return TextFormField(
    decoration: const InputDecoration(
      icon: Icon(Icons.person),
      hintText: 'The one call away',
      labelText: 'Phone Number *',
    ),
    validator: (String? value) {
      if (value == null || value.isEmpty) {
        return 'phone number is required';
      }
      return null;
    },
    onSaved: (value) {
      _phoneNumber = value!;
    },
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Personal Data')),
    body: Container(
      margin: const EdgeInsets.all(24),
      child: Form(
        key: _formKey,
        child: Wrap(
            direction: Axis.horizontal,
            spacing: 8.0, // gap between adjacent chips
            runSpacing: 4.0,
            children: <Widget>[
              _buildName(),
              _buildEmail(),
              _buildPassword(),
              _buildPhoneNumber(),
              ElevatedButton(
                onPressed: () async {
                  if (_formKey.currentState != null &&
                      _formKey.currentState!.validate()) {
                    _formKey.currentState!.save();
                    _formKey.currentState!.reset();
                    ScaffoldMessenger.of(context).showSnackBar(
                      const SnackBar(content: Text('Data stored')),
```

```
                );
                Map<String,dynamic> data={
                  "name":_name,
                  "email":_email,
                  "password":_password,
                  "contact":_phoneNumber

                };
                //Insert
                DatabaseReference ref= _database.child("records").push();
                newPostKey=ref.key;
                await ref.set(data);
               }
               //View
               displayData();

               //update
               DatabaseReference ref=FirebaseDatabase.instance.ref("records/$newPostKey");
               await ref.update({
                 "name":"ramu",
               });
               //view
               displayData();
               //delete
                ref=FirebaseDatabase.instance.ref("records/$newPostKey");
               await ref.remove();
               displayData();
              },
              child: const Text('Submit',
                 style: TextStyle(fontSize: 18, color: Colors.white)),
            ),
          ]
        )
        ,
      ),
    ),
  );
 }

 Future<void> displayData() async {
   _data=await _database.child("records").get();
   print(_data.value);
 }
}
```

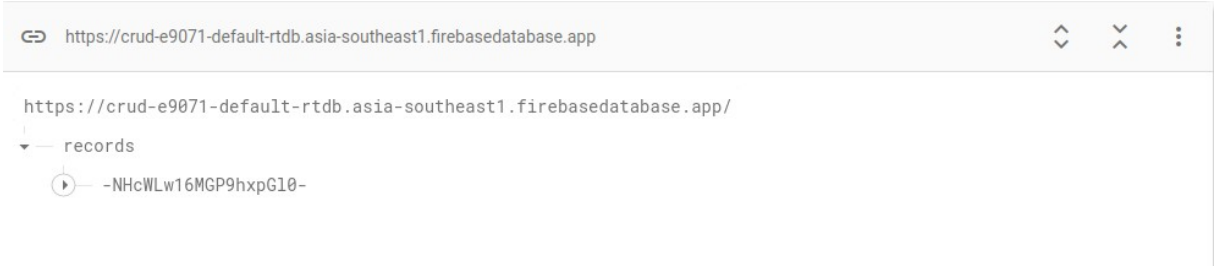
Output:

https://crud-e9071-default-rtdb.asia-southeast1.firebasedatabase.app/

- records
  - -NHcWLw16MGP9hxpGl0-
    - contact: "9245448121"
    - email: "kanya@123"
    - name: "ramu"
    - password: "1234"




https://crud-e9071-default-rtdb.asia-southeast1.firebasedatabase.app/
records
-NHcWLw16MGP9hxpGl0-
contact: "9245448121"
email: "kanya@123"
name: "ramu"
password: "1234"

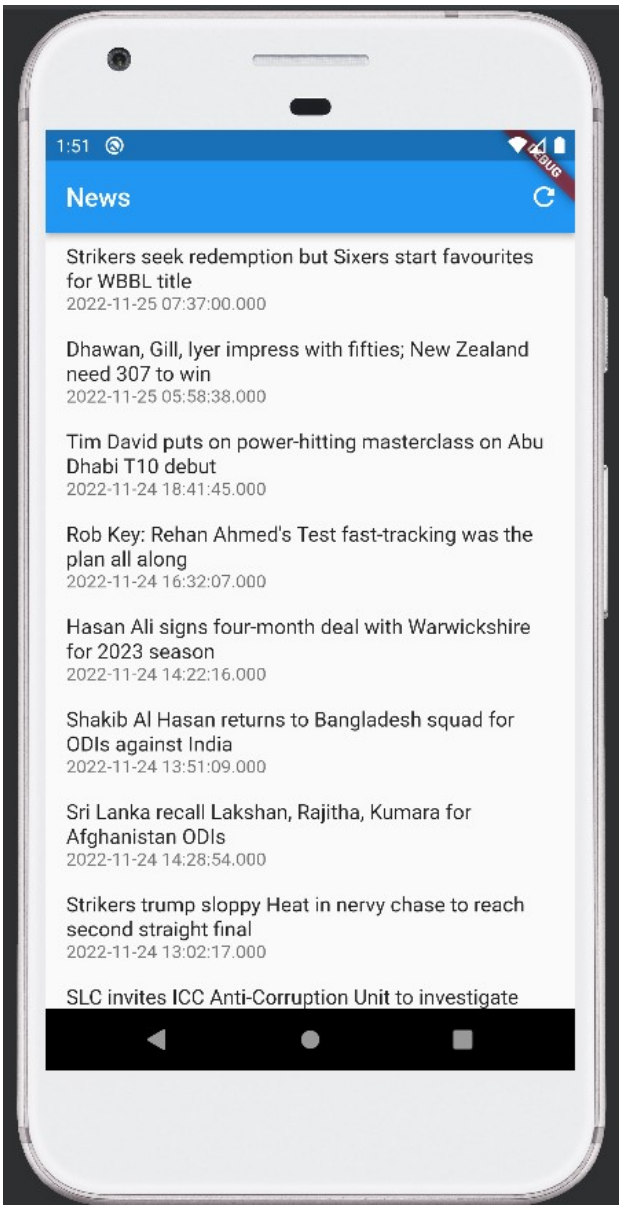

https://crud-e9071-default-rtdb.asia-southeast1.firebasedatabase.app
https://crud-e9071-default-rtdb.asia-southeast1.firebasedatabase.app/
records
-NHcWLw16MGP9hxpGl0-

Code:

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

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

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

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

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

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("News"),
```

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

Output:


1:51
DEBUG
News
Strikers seek redemption but Sixers start favourites for WBBL title
2022-11-25 07:37:00.000
Dhawan, Gill, Iyer impress with fifties; New Zealand need 307 to win
2022-11-25 05:58:38.000
Tim David puts on power-hitting masterclass on Abu Dhabi T10 debut
2022-11-24 18:41:45.000
Rob Key: Rehan Ahmed's Test fast-tracking was the plan all along
2022-11-24 16:32:07.000
Hasan Ali signs four-month deal with Warwickshire for 2023 season
2022-11-24 14:22:16.000
Shakib Al Hasan returns to Bangladesh squad for ODIs against India
2022-11-24 13:51:09.000
Sri Lanka recall Lakshan, Rajitha, Kumara for Afghanistan ODIs
2022-11-24 14:28:54.000
Strikers trump sloppy Heat in nervy chase to reach second straight final
2022-11-24 13:02:17.000
SLC invites ICC Anti-Corruption Unit to investigate

Code:

```
import 'package:flutter/material.dart';
import 'package:geolocator/geolocator.dart';
import 'dart:async';
void main() {
  runApp(const MyApp());}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const Home(),    );  }}
class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);
  @override
  State<Home> createState() => _HomeState();
}
class _HomeState extends State<Home> {
  bool servicesstatus=false;
  bool haspermission=false;
  late LocationPermission permission;
  late Position position;
  String long="",lat="";

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

  checkGps() async{
    servicesstatus=await Geolocator.isLocationServiceEnabled();
    if(servicesstatus){
      permission=await Geolocator.checkPermission();
      if(permission==LocationPermission.denied){
          permission=await Geolocator.requestPermission();
          if(permission==LocationPermission.denied){
            print("location permission denied");
          }else if(permission==LocationPermission.deniedForever){
            print("Permission denied permentently");}
          }else{
            haspermission=true;          }
          if(haspermission){
```

```
          setState(() {
            //refresh the UI;
          });
          geoLocation();}
        }else{
          print("GPS location not enabled,turn on gps location");
        }
        setState(() {    });   }
  geoLocation() async {
  position=await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high);
  long=position.longitude.toString();
  lat=position.latitude.toString();
  setState(() {
    //refresh UI
  });
  LocationSettings locationSettings = const LocationSettings(
    accuracy: LocationAccuracy.high, //accuracy of the location data
    distanceFilter: 100,
  );
  StreamSubscription<Position> positionStream = Geolocator.getPositionStream(
      locationSettings: locationSettings).listen((Position position) {
    long = position.longitude.toString();
    lat = position.latitude.toString();

    setState(() {});});   }
 @override
 Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title:Text('GPS location'),
      backgroundColor: Colors.redAccent,
    ),   body:Container(
      alignment:Alignment.center,
      padding: EdgeInsets.all(50),
      child:Column(
        children:[
          Text(servicesstatus?"GPS enabled":"GPS disabled"),
          Text(haspermission?"GPS is Enabled":"GPS is disabled"),
          Text("Longitute:$long",style:TextStyle(fontSize: 20)),
          Text("latitude:$lat",style:TextStyle(fontSize:20),)
        ]
      )
    )
   );

  }
}
```

Output:

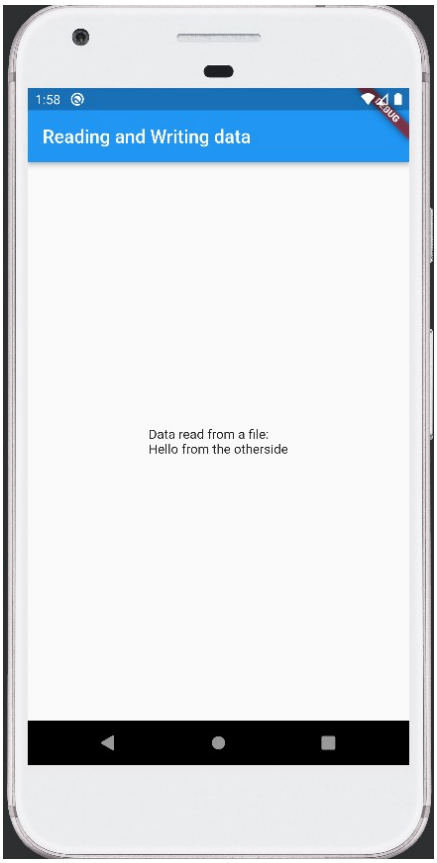

1:56
DEBUG
GPS location
GPS enabled
GSP permission granted
Longitute:-122.084
latitude:37.4219983

Code:

```
import 'dart:async';
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:path_provider/path_provider.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 Home(),
    );
  }
}

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

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

class _HomeState extends State<Home> {

   late String _data;
   @override
   void initState() {
     super.initState();
     writeContent();
     readContent().then((String value) {
       setState(() {
         _data = value;
       });
     });
```

```
  }

  Future<String?> get _localPath async {
    final directory = await getExternalStorageDirectory();
    // print(directory?.path);
    return directory?.path;
  }
  Future<File> get _localFile async {
    final path = await _localPath;
    return File('$path/file.txt');
  }

  Future<String> readContent() async {
    try {
      final file = await _localFile;
      // Read the file
      String contents = await file.readAsString();
      // Returning the contents of the file
      return contents;
    } catch (e) {
      // If encountering an error, return
      return 'Error!';
    }
  }

  Future<File> writeContent() async {
    final file = await _localFile;
    // Write the file
    return file.writeAsString('Hello from the otherside');   }
  @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(title:Text('Reading and Writing data')),
     body:Center(
       child:Text(
         'Data read from a file:\n$_data,$_localFile',
       )
     )
   );
  }
}
```

Output:

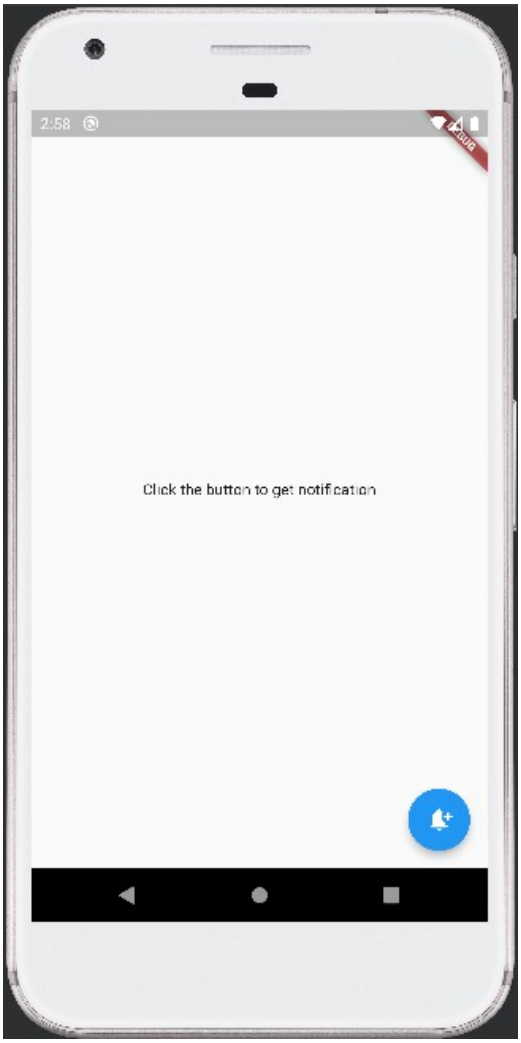

1:58
DEBUG
Reading and Writing data
Data read from a file:
Hello from the otherside

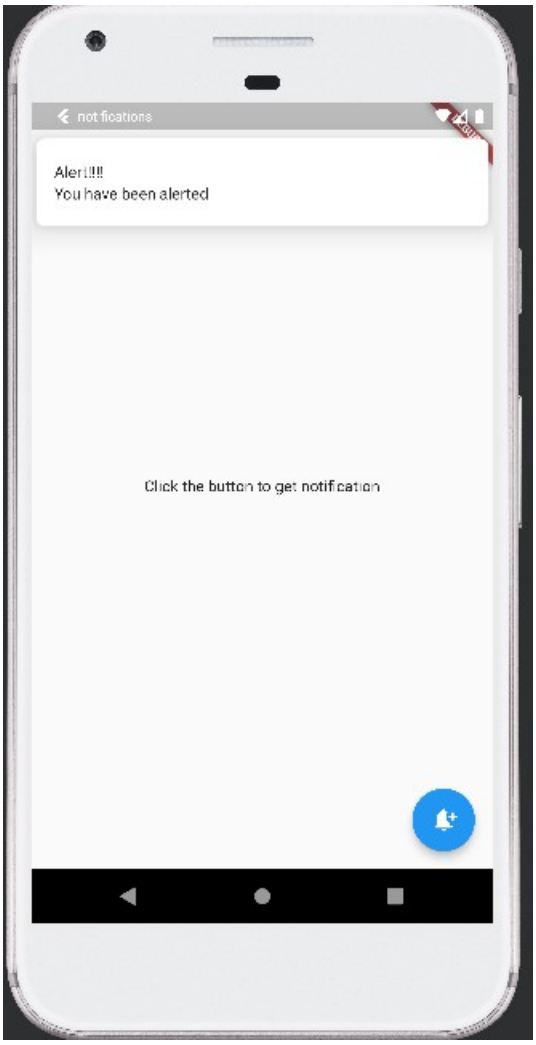
Code:

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

void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(

        primarySwatch: Colors.blue,
      ),
      home: const NotificationApp(),
    );
  }
}

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

  @override
  State<NotificationApp> createState() => _NotificationAppState();
}
class _NotificationAppState extends State<NotificationApp> {
//local notification object

late FlutterLocalNotificationsPlugin localNotifaction;

@override
  void initState() {
    // TODO: implement initState
    super.initState();
    var andriodInitialize =new AndroidInitializationSettings('ic_launcher');
    var initialzationSettings=new InitializationSettings(android: andriodInitialize);
    localNotifaction=FlutterLocalNotificationsPlugin();
    localNotifaction.initialize(initialzationSettings);
  }

  Future _showNotification() async{
    AndroidNotificationDetails _androidNotificationDetails =
    const AndroidNotificationDetails(
```

```
      'channel ID',
      'channel name',
      playSound: true,
      priority: Priority.high,
      importance: Importance.high,
    );
   var generalNotificationDetails=new
NotificationDetails(android:_androidNotificationDetails);
   await localNotifaction.show(0,'Alert!!!!',"You have been
alerted",generalNotificationDetails);
  }




  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body:const Center(
        child:Text("Click the button to get notification"),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _showNotification,
        child:const Icon(Icons.notification_add_rounded)
      ),
    );
  }
}
```

Output:


2:58
Click the button to get notification

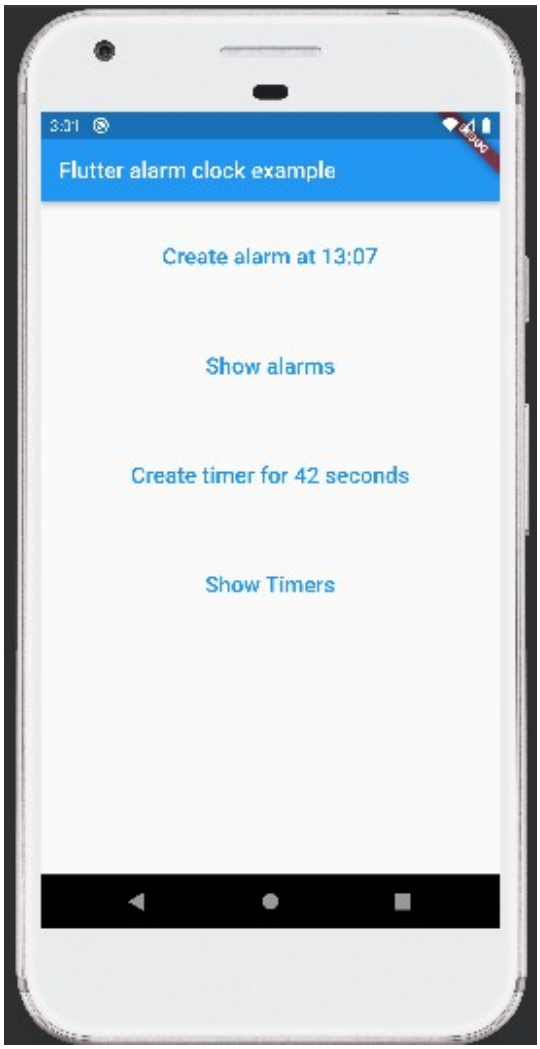

notifications
Alert!!!
You have been alerted
Click the button to get notification

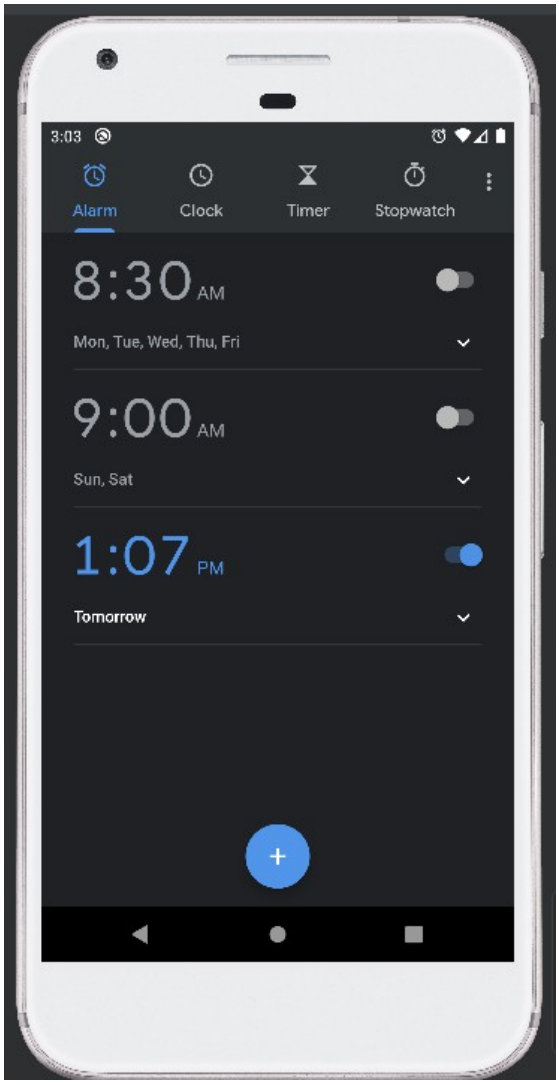
Code:

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

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

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

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

class _MyAppState extends State<MyApp> {
  @override
  void initState() {
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Flutter alarm clock example'),
        ),
        body: Center(
            child: Column(children: <Widget>[
              Container(
                margin: const EdgeInsets.all(25),
                child: TextButton(
                  child: const Text(
                    'Create alarm at 13:07',
                    style: TextStyle(fontSize: 20.0),
                  ),
                  onPressed: () {
                    FlutterAlarmClock.createAlarm(13, 07);
                  },
                ),
              ),
              Container(
                margin: const EdgeInsets.all(25),
                child: TextButton(
                  child: const Text(
                    'Show alarms',
```

```
                style: TextStyle(fontSize: 20.0),
              ),
              onPressed: () {
                FlutterAlarmClock.showAlarms();
              },
            ),
          ),
          Container(
            margin: const EdgeInsets.all(25),
            child: TextButton(
              child: const Text(
                'Create timer for 42 seconds',
                style: TextStyle(fontSize: 20.0),
              ),
              onPressed: () {
                FlutterAlarmClock.createTimer(42);
              },
            ),
          ),
          Container(
            margin: const EdgeInsets.all(25),
            child: TextButton(
              child: const Text(
                'Show Timers',
                style: TextStyle(fontSize: 20.0),
              ),
              onPressed: () {
                FlutterAlarmClock.showTimers();
              },
            ),
          ),
        ])),
    ),
  );
 }
}
```

Output:


3:01
Flutter alarm clock example
Create alarm at 13:07
Show alarms
Create timer for 42 seconds
Show Timers


3:03
Alarm
Clock
Timer
Stopwatch
8:30 AM
Mon, Tue, Wed, Thu, Fri
9:00 AM
Sun, Sat
1:07 PM
Tomorrow

Code:

main.dart

```
import 'package:flutter/material.dart';
import './pong.dart';
void main() => runApp(MyApp());
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Pong Demo',
theme: ThemeData(
  primarySwatch: Colors.blue,
),
home: Scaffold(
  appBar: AppBar(
    title: Text('Simple Pong'),
  ),
  body: SafeArea(
    child: Pong(),
  )));  }}
```

bat.dart

```
import 'package:flutter/material.dart';
class Bat extends StatelessWidget {
  final double width;
  final double height;

  Bat(this.width, this.height);

  @override
  Widget build(BuildContext context) {
    return Container(
      width: width,
      height: height,
      decoration: new BoxDecoration(
        color: Colors.blue[900],
      ));
  }
}
```

Ball.dart

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

class Ball extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final double diam = 50;
    return Container(
      width: diam,
      height: diam,
      decoration:
      new BoxDecoration(color: Colors.amber[400], shape: BoxShape.circle),
    );
  }
}
```

pong.dart

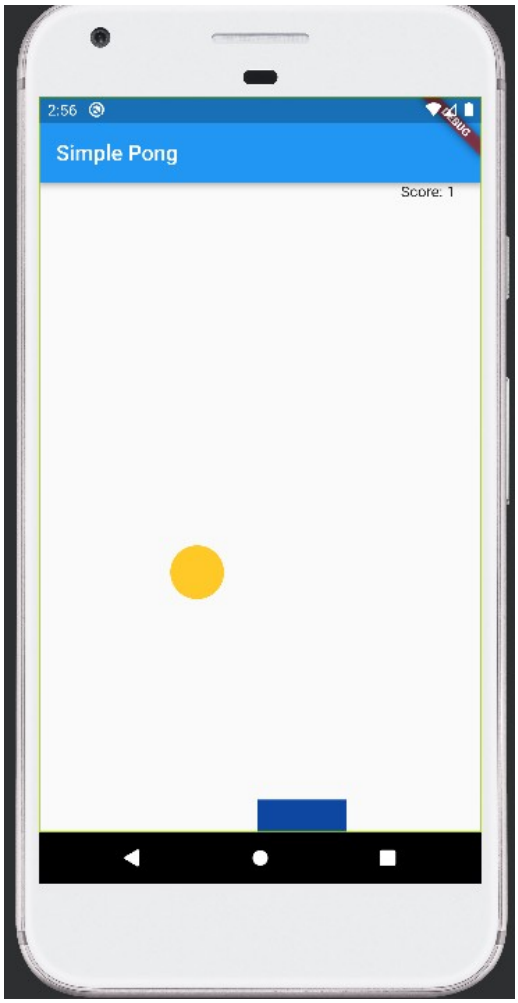
```
import 'package:flutter/material.dart';
import './ball.dart';
import './bat.dart';
import 'dart:math';

enum Direction { up, down, left, right }

class Pong extends StatefulWidget {
  @override
  _PongState createState() => _PongState();
}

class _PongState extends State<Pong> with SingleTickerProviderStateMixin {
  double increment = 5;

  Direction vDir = Direction.down;
  Direction hDir = Direction.right;

  late Animation<double> animation;
  late AnimationController controller;

   double width=5.0;
   double height=5.0;
  double posX = 0;
  double posY = 0;

  double batWidth = 0;
```

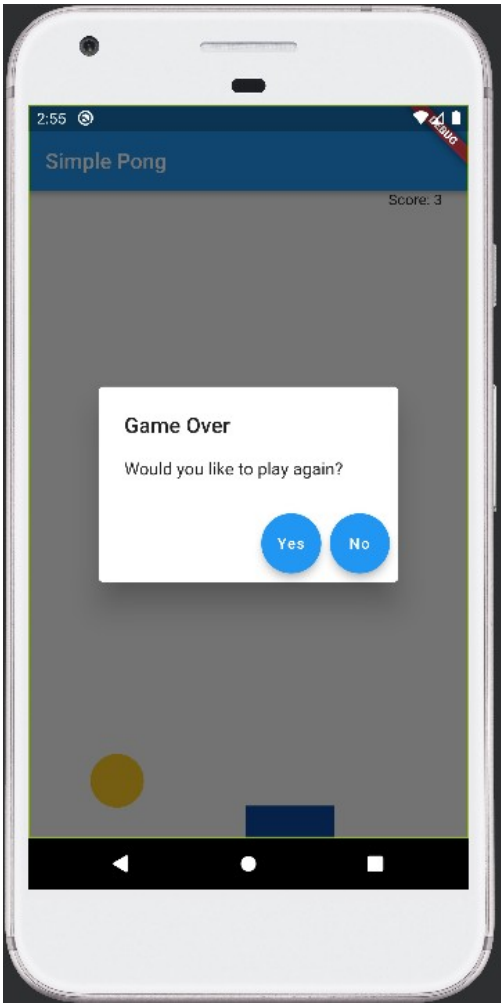
```
double batHeight = 0;
double batPosition = 0;

double randX = 1;
double randY = 1;

int score = 0;

// bool showDialog = false;

double randomNumber() {
  //this is a number between 0.5 and 1.5;
  var ran = new Random();
  int myNum = ran.nextInt(101);
  return (50 + myNum) / 100;
}

void showMessage(BuildContext context) {
  showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('Game Over'),
          content: Text('Would you like to play again?'),
          actions: <Widget>[
            FloatingActionButton(
              child: Text('Yes'),
              onPressed: () {
                setState(() {
                  posX = 0;
                  posY = 0;
                  score = 0;
                });
                Navigator.of(context).pop();
                controller.repeat();
              },
            ),
            FloatingActionButton(
              child: Text('No'),
              onPressed: () {
                Navigator.of(context).pop();
                dispose();
              },
            )
          ],
        );
      });
}
```

```
@override
void initState() {
  posX = 0;
  posY = 0;
  controller = AnimationController(
    duration: const Duration(minutes: 10000),
    vsync: this,
  );
  animation = Tween<double>(begin: 0, end: 100).animate(controller);
  animation.addListener(() {
    safeSetState(() {
      (hDir == Direction.right)
          ? posX += ((increment * randX).round())
          : posX -= ((increment * randX).round());
      (vDir == Direction.down)
          ? posY += ((increment * randY).round())
          : posY -= ((increment * randY).round());
    });
    checkBorders();
  });
  controller.forward();
  super.initState();
}

void checkBorders() {
  double diameter = 50;
  if (posX <= 0 && hDir == Direction.left) {
    hDir = Direction.right;
    randX = randomNumber();
  }
  if (posX >= width - diameter && hDir == Direction.right) {
    hDir = Direction.left;
    randX = randomNumber();
  }
  //check the bat position as well
  if (posY >= height - diameter - batHeight && vDir == Direction.down) {
    //check if the bat is here, otherwise loose
    if (posX >= (batPosition - diameter) &&
        posX <= (batPosition + batWidth + diameter)) {
      vDir = Direction.up;
      randY = randomNumber();
      safeSetState(() {
        score++;
      });
    } else {
      controller.stop();
      showMessage(context);
    }
  }
```

```
  if (posY <= 0 && vDir == Direction.up) {
    vDir = Direction.down;
    randY = randomNumber();
  }
 }

 @override
 Widget build(BuildContext context) {
  return LayoutBuilder(
      builder: (BuildContext context, BoxConstraints constraints) {
        height = constraints.maxHeight;
        width = constraints.maxWidth;
        batWidth = width / 5;
        batHeight = height / 20;

        return Stack(
          children: <Widget>[
            Positioned(
                top: 0, right: 24, child: Text('Score: ' + score.toString())),
            Positioned(child: Ball(), top: posY, left: posX),
            Positioned(
                bottom: 0,
                left: batPosition,
                child: GestureDetector(
                    onHorizontalDragUpdate: (DragUpdateDetails update) =>
                        moveBat(update),
                    child: Bat(batWidth, batHeight))),
          ],
        );
      });
 }
 void moveBat(DragUpdateDetails update) {
  safeSetState(() {
    batPosition += update.delta.dx;
  });
 }

 @override
 void dispose() {
  controller.dispose();
  super.dispose();
 }

 void safeSetState(Function function) {
    if (mounted && controller.isAnimating) {
    setState(() {
      function();
    });
  }  }}
```

Output:


2:56
Simple Pong
Score: 1


2:55
Simple Pong
Score: 3
Game Over
Would you like to play again?
Yes
No

Code:

user_list.dart

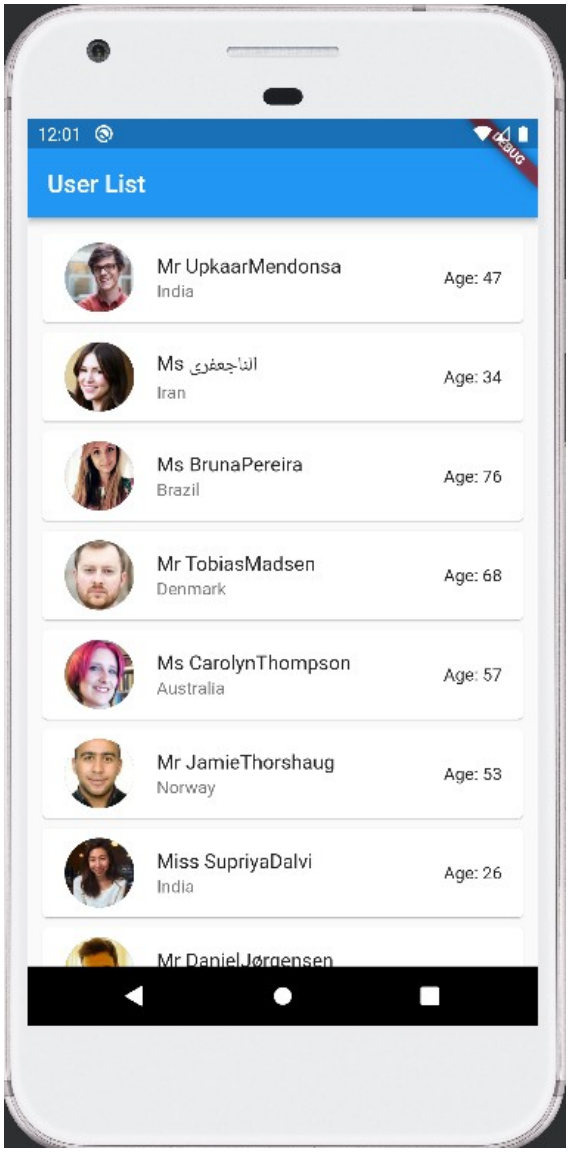
```
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

class UserList extends StatelessWidget {
  final Uri apiUrl=Uri.parse("https://randomuser.me/api/?results=10") as Uri;
  Future<List<dynamic>>fetchUsers()async{
    var result=await http.get(apiUrl);
    return json.decode(result.body)['results'];

  }
  String _name(dynamic user){
    return user['name']['title'] + " "+user['name']['first']+""+user['name']['last'];

  }

  String _location(dynamic user){
    return user['location']['country'];
  }

  String _age(Map<dynamic,dynamic> user){
    return "Age: "+user['dob']['age'].toString();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title:Text('User List'),
      ),
      body:Container(
        child:FutureBuilder<List<dynamic>>(
          future:fetchUsers(),
          builder:(BuildContext context,AsyncSnapshot snapshot){
            if(snapshot.hasData){
              return ListView.builder(
                  padding: EdgeInsets.all(8),
                  itemCount: snapshot.data.length,
                  itemBuilder: (BuildContext context,int index){
                    return
                      Card(
                        child:Column(
                          children:<Widget>[
                            ListTile(
```

```
                        leading:CircleAvatar(
                          radius:30,
                          backgroundImage: NetworkImage(snapshot.data[index]['picture']
['large'])),
                        title:Text(_name(snapshot.data[index])),
                        subtitle:Text(_location(snapshot.data[index])),
                        trailing: Text(_age(snapshot.data[index])),
                      )
                    ]
                  )
                );

              });
          }else{
            return Center(child:CircularProgressIndicator());
          }
        }
      )
    )
  );
 }
}
```

main.dart

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


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

class MyApp extends StatelessWidget {
 // This widget is the root of your application.
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Rest API Demo',
     theme: ThemeData(

       primarySwatch: Colors.blue,
     ),
     home: UserList(),
   );
 }
}
```

Output:


12:01
DEBUG
User List
Mr UpkaarMendonsa
India
Age: 47
Ms الناجعفری
Iran
Age: 34
Ms BrunaPereira
Brazil
Age: 76
Mr TobiasMadsen
Denmark
Age: 68
Ms CarolynThompson
Australia
Age: 57
Mr JamieThorshaug
Norway
Age: 53
Miss SupriyaDalvi
India
Age: 26
Mr DanielJørgensen