MAD and PWA Lab

Experiment - 6

Name: Anuj Chitari **Class: D15A Roll no:04**

Aim: To Connect Flutter UI with FireBase database

Theory:

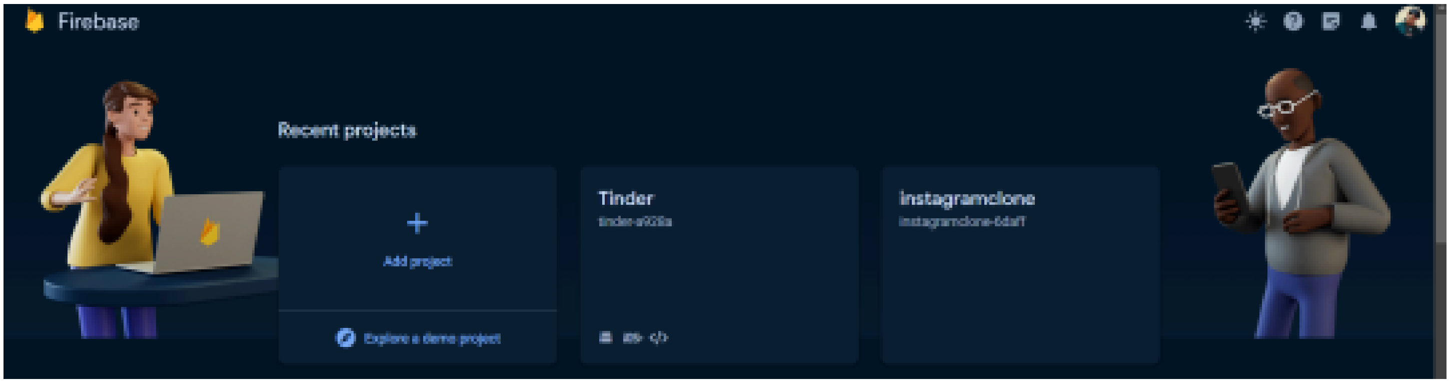
Prerequisites

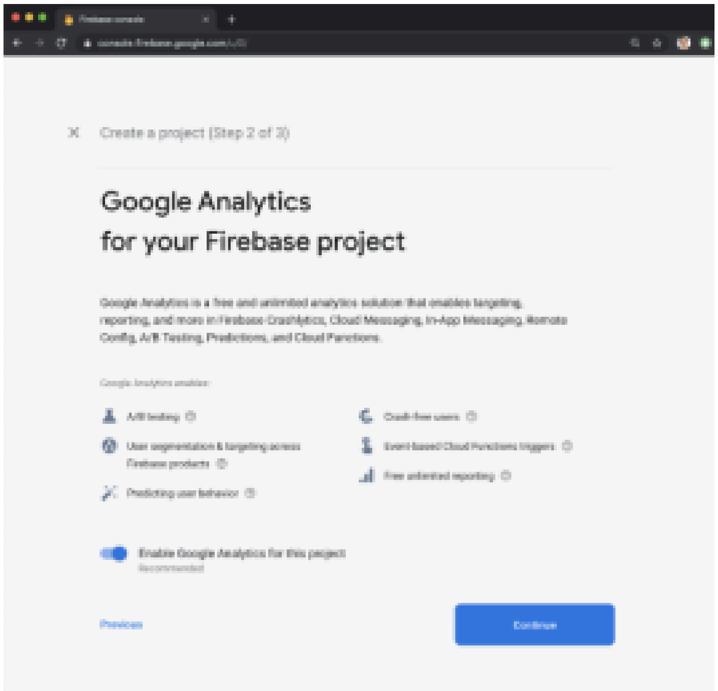
To complete this tutorial, you will need:

- A Google account to use Firebase.
- Developing for iOS will require XCode.
- To download and install Flutter.
- To download and install Android Studio and Visual Studio Code.
- It is recommended to install plugins for your code editor:
  - Flutter and Dart plugins installed for Android Studio.
  - Flutter extension installed for Visual Studio Code.

Create a Firebase Project:

First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name:





Create a project (Step 2 of 3)

Google Analytics
for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Predicting user behavior
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

Go to the Firebase Console and create a new project.
Add your Flutter app to the Firebase project:

Register your app in the Firebase project, and follow the instructions to download the configuration files (google-services.json for Android, GoogleService-Info.plist for iOS).

The most important thing here is to match up the Android package name that you choose here with the one inside of our application.

The structure consists of at least two segments. A common pattern is to use a domain name, a company name, and the application name:

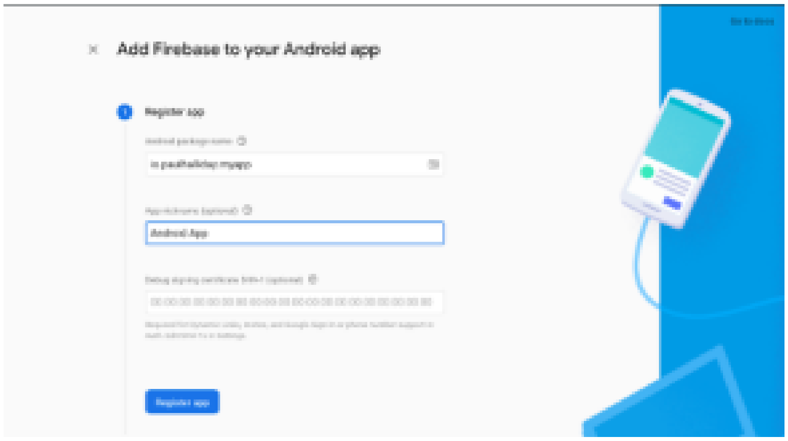
```
com.example.flutterfirebaseexample
```

Once you've decided on a name, open android/app/build.gradle in your code editor and update the applicationId to match the Android package name:

```
android/app/build.gradle
...
defaultConfig {
   // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
   applicationId 'com.example.flutterfirebaseexample'
    ...
}
...
```

Downloading the Config File

The next step is to add the Firebase configuration file into our Flutter project. This is important as it contains the API keys and other critical information for Firebase to use.
Select Download `google-services.json` from this page:



Add Firebase to your Android app

1 Register app

Android package name

io.paulhalliday.myapp

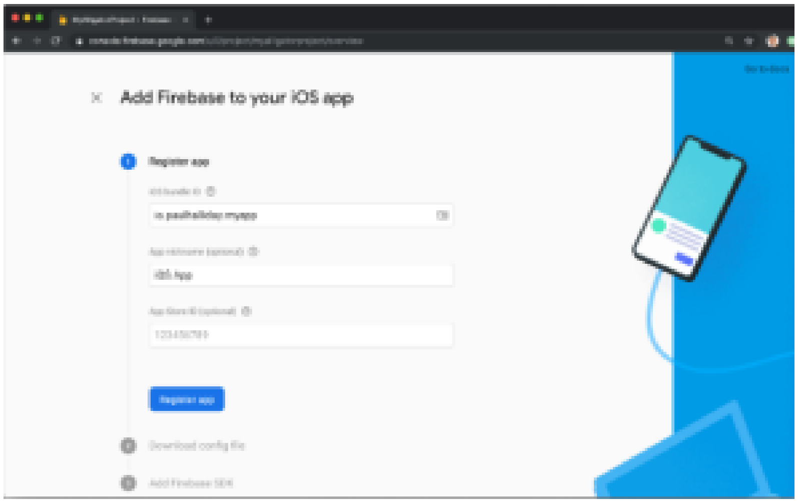
App nickname (optional)

Android App

Debug signing certificate SHA-1 (optional)

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Register app



Add Firebase to your iOS app

1 Register app

iOS bundle ID

io.paulhalliday.myapp

App nickname (optional)

iOS App

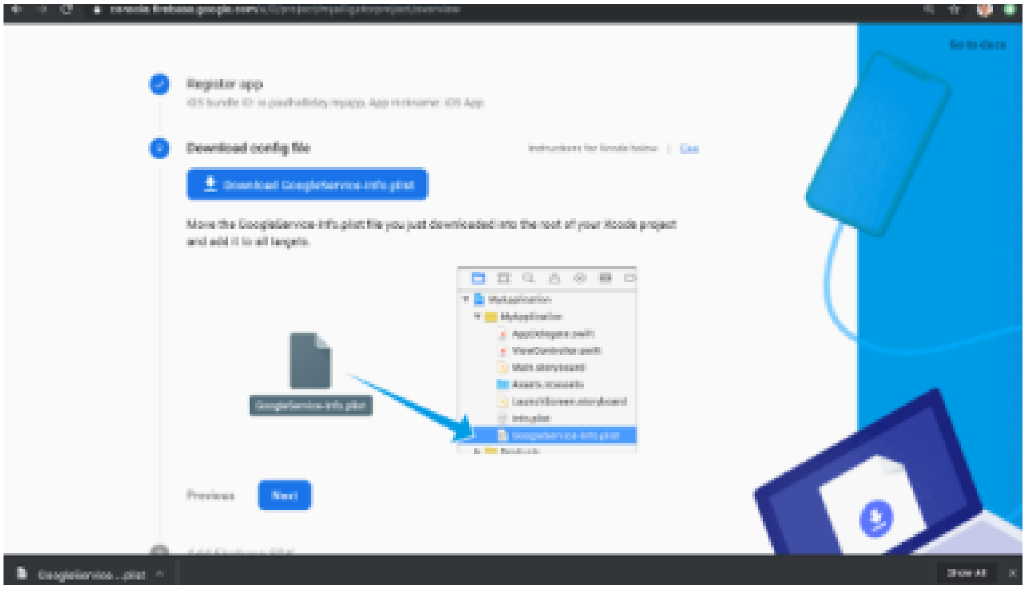
App Store ID (optional)

123456789

Register app

Download config file

Add Firebase SDK



Register app

iOS bundle ID: io.paulhalliday.myapp, App nickname: iOS App

Download config file

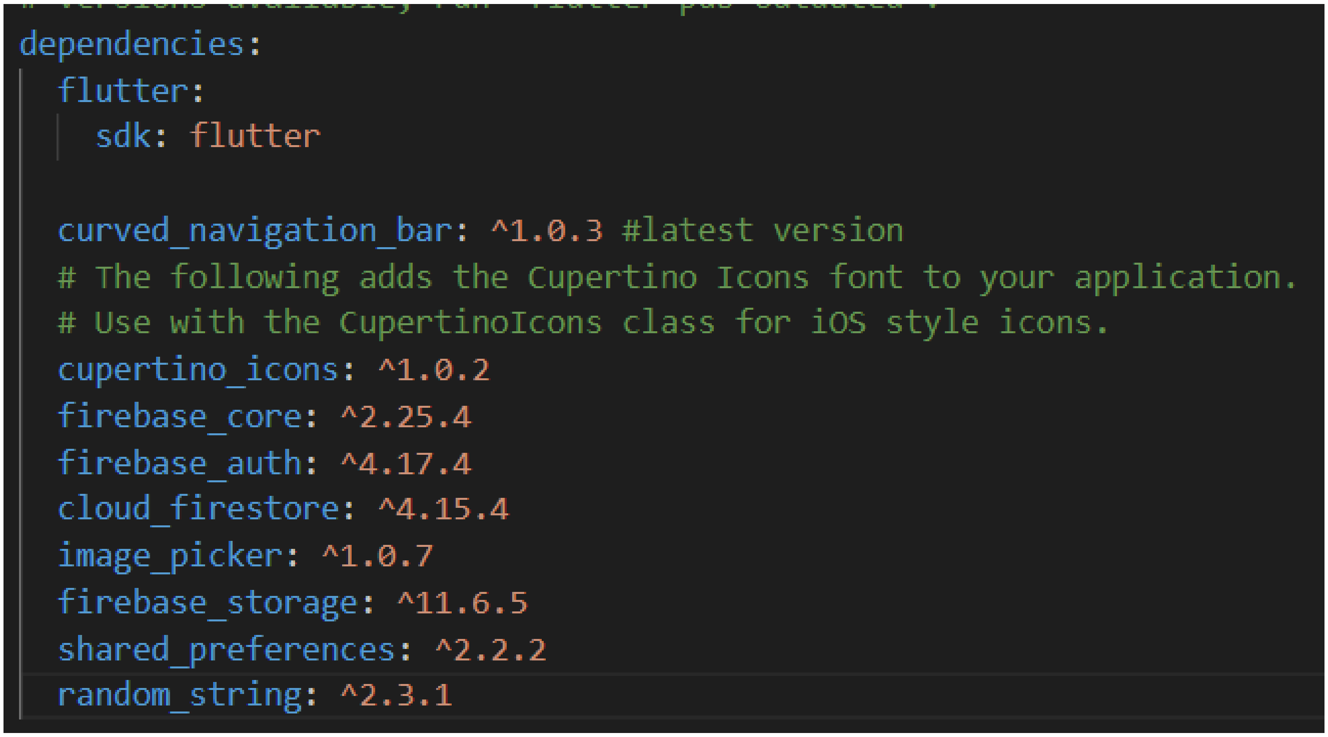
Download GoogleService-Info.plist

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.

2. Add Firebase to your Flutter project:

Add Dependencies:

Open your pubspec.yaml file and add the necessary dependencies:
yaml



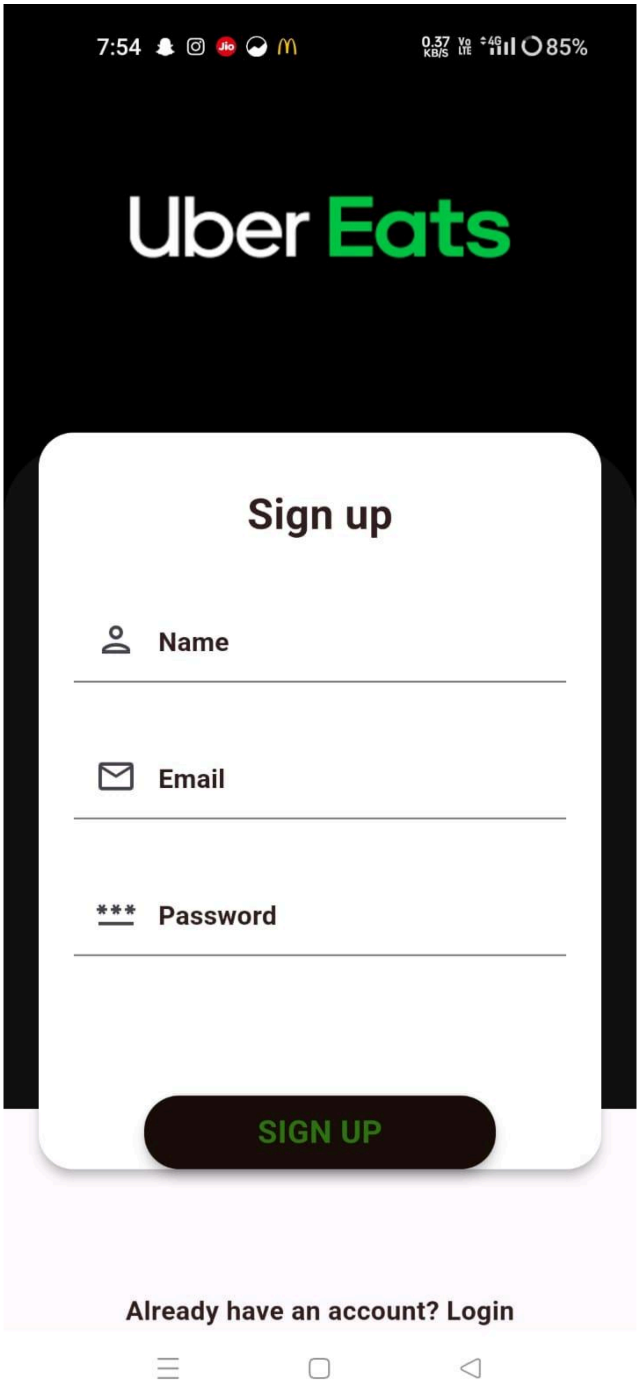
```yaml
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

  curved_navigation_bar: ^1.0.3 #latest version
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  firebase_core: ^2.25.4
  firebase_auth: ^4.17.4
  cloud_firestore: ^4.15.4
  image_picker: ^1.0.7
  firebase_storage: ^11.6.5
  shared_preferences: ^2.2.2
  random_string: ^2.3.1
```

Code:
Signup.dart

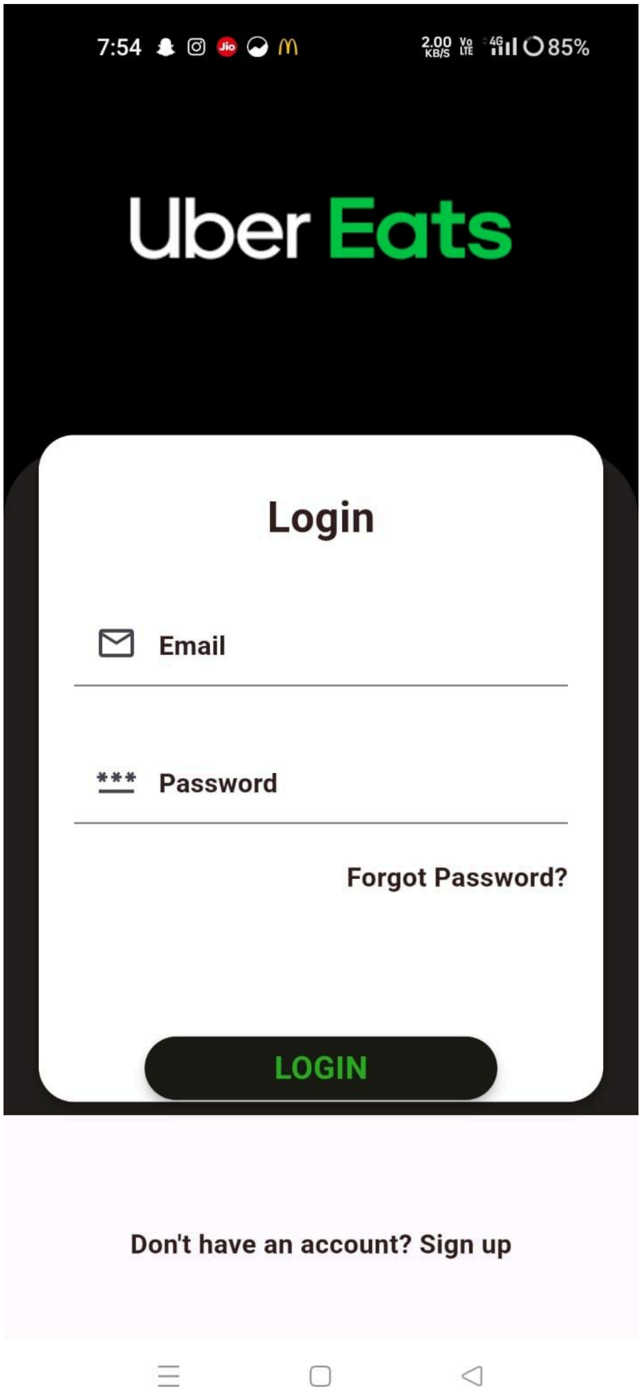
```dart
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/pages/bottomnav.dart';
import 'package:food_panda/pages/login.dart';
import 'package:food_panda/widget/widget_support.dart';


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

  @override
  State<SignUp> createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
  String email = "", password = "", name = "";

  TextEditingController namecontroller = new TextEditingController();
TextEditingController passwordcontroller = new TextEditingController();

TextEditingController mailcontroller = new TextEditingController();

final _formkey = GlobalKey<FormState>();

registration() async {
  if (password != null) {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(email: email, password: password);

      ScaffoldMessenger.of(context).showSnackBar((SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text(
            "Registered Successfully",
            style: TextStyle(fontSize: 20.0),
          ))));

      // ignore: use_build_context_synchronously
      Navigator.pushReplacement(
          context, MaterialPageRoute(builder: (context) => BottomNav()));
    } on FirebaseException catch (e) {
      if (e.code == 'weak-password') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Password Provided is too Weak",
              style: TextStyle(fontSize: 18.0),
            )));
      } else if (e.code == "email-already-in-use") {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Account Already exsists",
              style: TextStyle(fontSize: 18.0),
            )));
      }
    }
  }
}


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

```
return Scaffold(
  body: Container(
    child: Stack(
      children: [
        Container(
          width: MediaQuery.of(context).size.width,
          height: MediaQuery.of(context).size.height / 2.5,
          decoration: BoxDecoration(
              gradient: LinearGradient(
                  begin: Alignment.topLeft,
                  end: Alignment.bottomRight,
                  colors: [
                  Color(0xFF000000),
                Color(0xFF000000),
              ])),
        ),
        Container(
          margin:
              EdgeInsets.only(top: MediaQuery.of(context).size.height / 3),
          height: MediaQuery.of(context).size.height / 2,
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
              color: Color.fromARGB(255, 18, 17, 17),
              borderRadius: BorderRadius.only(
                  topLeft: Radius.circular(40),
                  topRight: Radius.circular(40))),
          child: Text(""),
        ),
        Container(
          margin: EdgeInsets.only(top: 60.0, left: 20.0, right: 20.0),
          child: Column(
            children: [
              Center(
                  child: Image.asset(
                "images/logo copy.png",
                width: MediaQuery.of(context).size.width / 1.5,
                fit: BoxFit.cover,
              )),
              SizedBox(
                height: 50.0,
              ),
              Material(
                elevation: 5.0,
                borderRadius: BorderRadius.circular(20),
                child: Container(
padding: EdgeInsets.only(left: 20.0, right: 20.0),
width: MediaQuery.of(context).size.width,
height: MediaQuery.of(context).size.height / 1.8,
decoration: BoxDecoration(
    color: const Color.fromARGB(255, 255, 255, 255),
    borderRadius: BorderRadius.circular(20)),
child: Form(
  key: _formkey,
  child: Column(
    children: [
      SizedBox(
        height: 30.0,
      ),
      Text(
        "Sign up",
        style: AppWidget.HeadlineTextFieldStyle(),
      ),
      SizedBox(
        height: 30.0,
      ),
      TextFormField(
        controller: namecontroller,
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please Enter Name';
          }
          return null;
        },
        decoration: InputDecoration(
            hintText: 'Name',
            hintStyle: AppWidget.semiBoldTextFieldStyle(),
            prefixIcon: Icon(Icons.person_outlined)),
      ),
      SizedBox(
        height: 30.0,
      ),
      TextFormField(
        controller: mailcontroller,
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please Enter E-mail';
          }
          return null;
        },
        decoration: InputDecoration(
      hintText: 'Email',
      hintStyle: AppWidget.semiBoldTextFieldStyle(),
      prefixIcon: Icon(Icons.email_outlined)),
),
SizedBox(
  height: 30.0,
),
TextFormField(
  controller: passwordcontroller,
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please Enter Password';
    }
    return null;
  },
  obscureText: true,
  decoration: InputDecoration(
      hintText: 'Password',
      hintStyle: AppWidget.semiBoldTextFieldStyle(),
      prefixIcon: Icon(Icons.password_outlined)),
),
SizedBox(
  height: 80.0,
),
GestureDetector(
  onTap: () async {
    if (_formkey.currentState!.validate()) {
      setState(() {
        email = mailcontroller.text;
        name = namecontroller.text;
        password = passwordcontroller.text;
      });
    }
    registration();
  },
  child: Material(
    elevation: 5.0,
    borderRadius: BorderRadius.circular(20),
    child: Container(
      padding: EdgeInsets.symmetric(vertical: 8.0),
      width: 200,
      decoration: BoxDecoration(
          color: Color.fromARGB(255, 26, 14, 10),
          borderRadius: BorderRadius.circular(20)),
      child: Center(
                   child: Text(
                  "SIGN UP",
                  style: TextStyle(
                    color: Color.fromARGB(153, 62, 185, 21),
                    fontSize: 18.0,
                    fontFamily: 'Poppins1',
                    fontWeight: FontWeight.bold),
                 )),
                ),
               ),
              ),
             ],
            ),
           ),
          ),
         ),
         SizedBox(
          height: 70.0,
         ),
         GestureDetector(
           onTap: () {
            Navigator.push(context,
              MaterialPageRoute(builder: (context) => LogIn()));
           },
           child: Text(
            "Already have an account? Login",
            style: AppWidget.semiBoldTextFieldStyle(),
           ))
        ],
       ),
      )
     ],
    ),
   ),
  );
 }
}
```

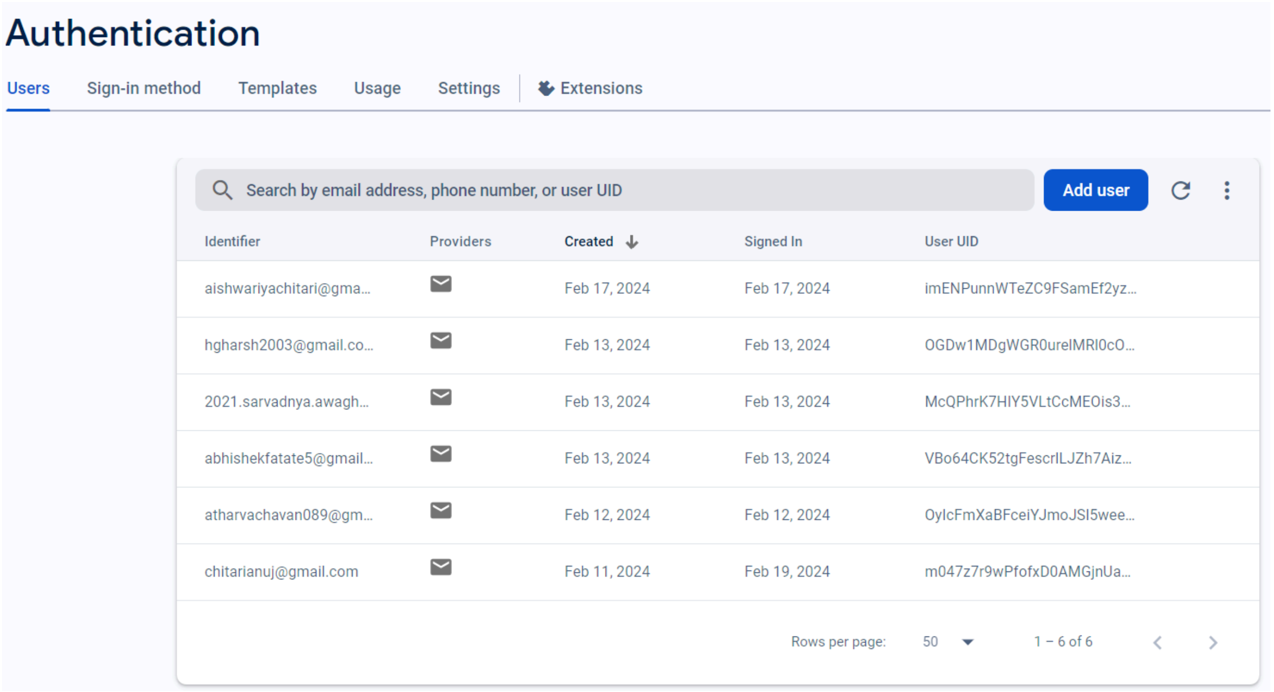
Output:



Already have an account? Login



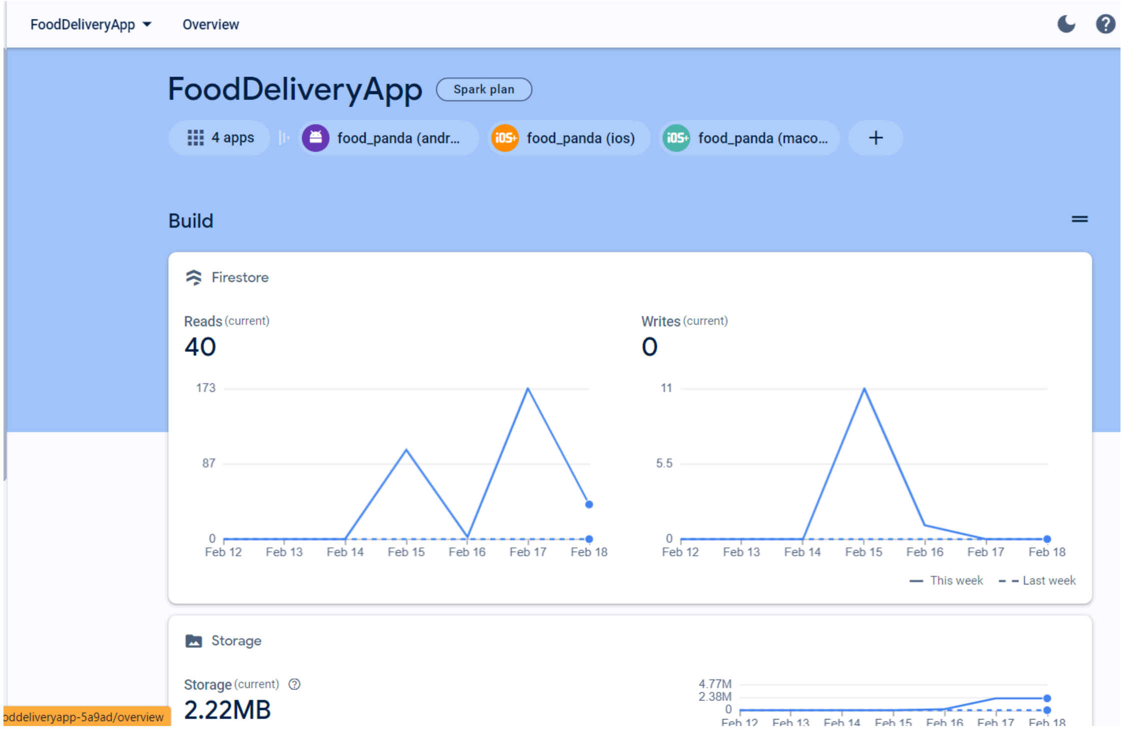
Don't have an account? Sign up



Authentication

Identifier	Providers	Created	Signed In	User UID
aishwariyachitari@gma...	✉	Feb 17, 2024	Feb 17, 2024	imENPunnWTeZC9FSamEf2yz...
hgharsh2003@gmail.co...	✉	Feb 13, 2024	Feb 13, 2024	OGDw1MDgWGR0urelMRl0cO...
2021.sarvadnya.awagh...	✉	Feb 13, 2024	Feb 13, 2024	McQPhrK7HIY5VLtCcMEOis3...
abhishekfatate5@gmail...	✉	Feb 13, 2024	Feb 13, 2024	VBo64CK52tgFescrlLJZh7Aiz...
atharvachavan089@gm...	✉	Feb 12, 2024	Feb 12, 2024	OylcFmXaBFceiYJmoJSI5wee...
chitarianuj@gmail.com	✉	Feb 11, 2024	Feb 19, 2024	m047z7r9wPfofxD0AMGjnUa...

Rows per page: 50 1 – 6 of 6



Conclusion:

In this experiment, we have successfully connected firebase database and authenticated using google signin and email and password without flutter application successfully.