

Name: Lisa Sharma

Roll No: 53

MAD and PWA Lab Experiment - 6

Aim: To Connect Flutter UI with FireBase database

Theory:

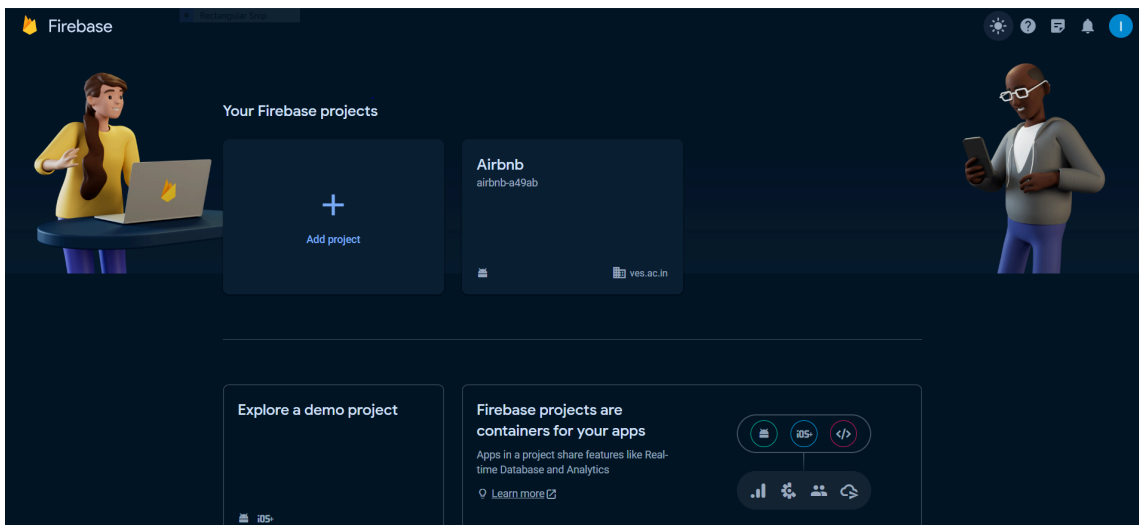
Prerequisites

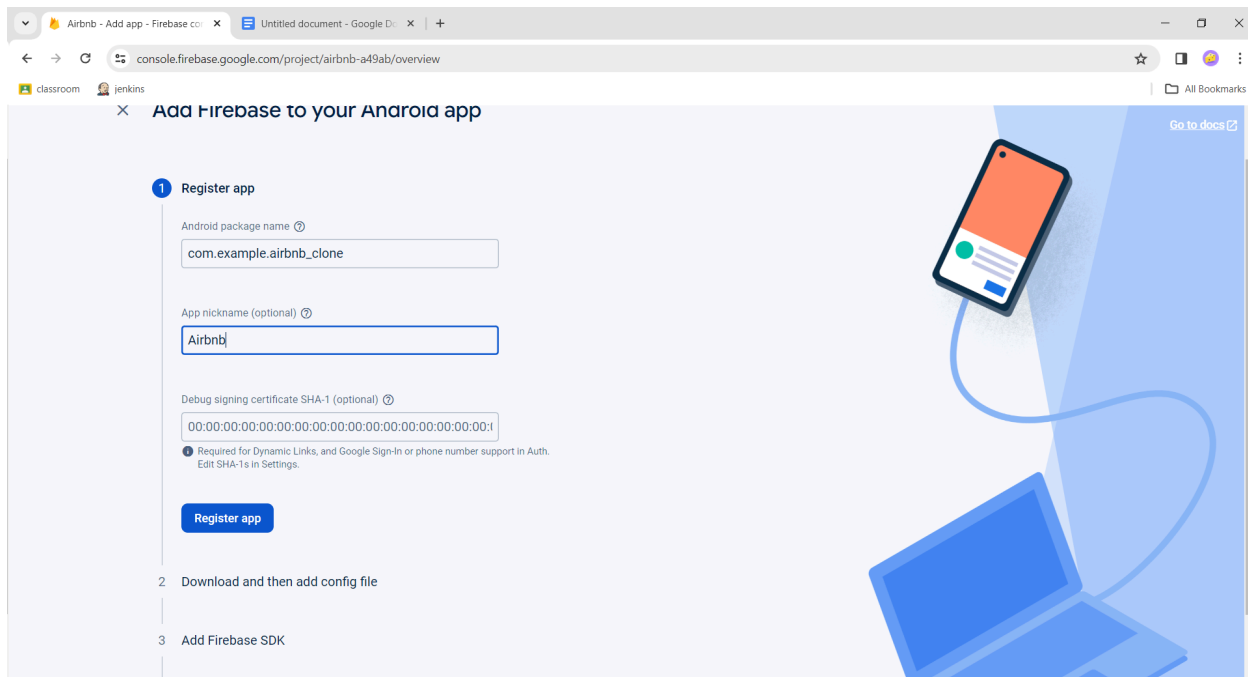
To complete this tutorial, you will need:

- 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:





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:

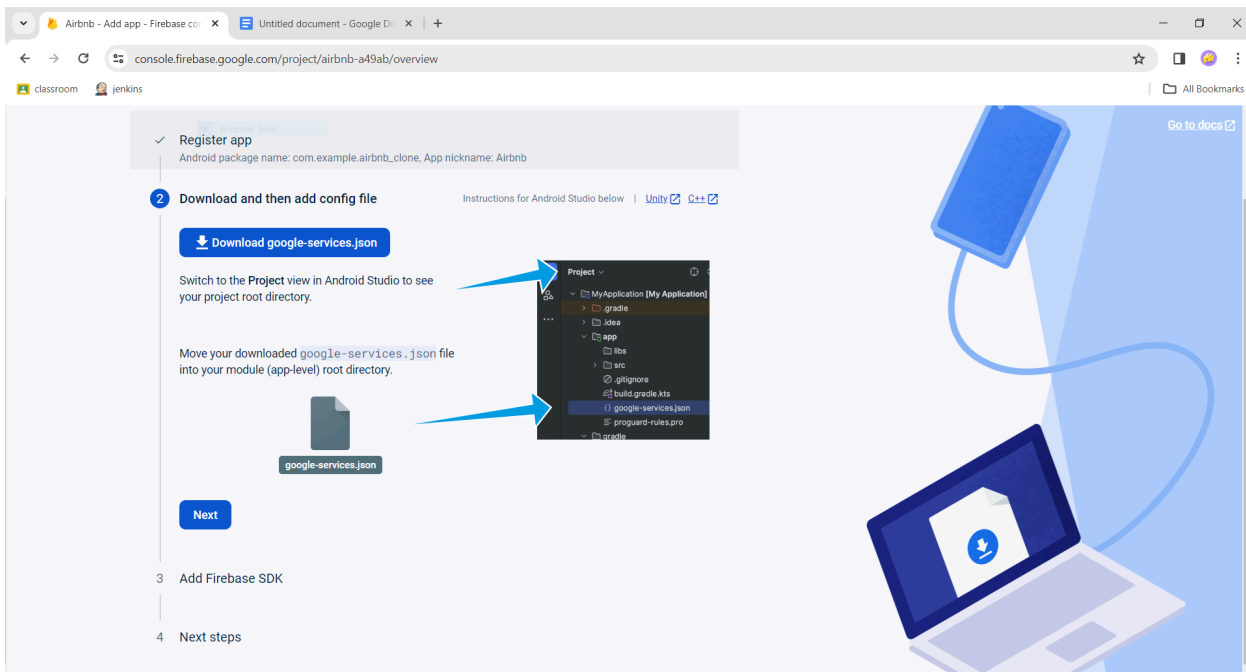
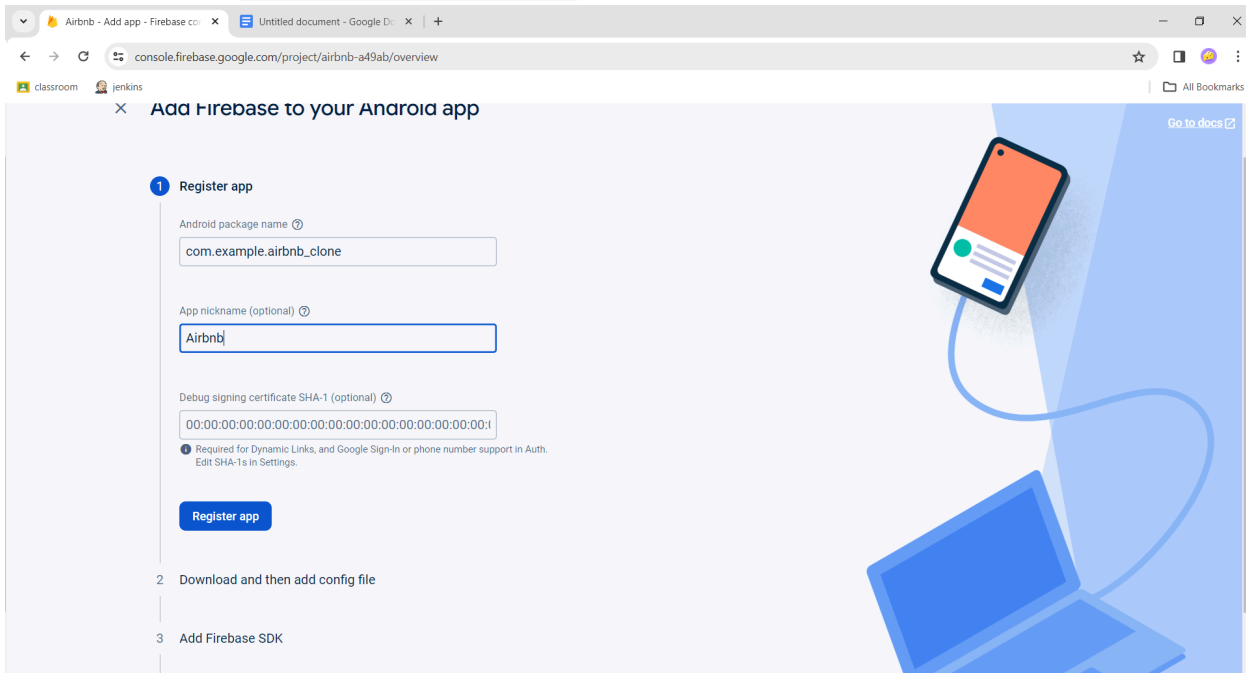
android/app/build.gradle

```
...
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:



2. Add Firebase to your Flutter project:
Add Dependencies:

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

yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  cupertino_icons: ^1.0.2  
  flutter_animate: ^4.2.0+1  
  go_router: ^10.1.2  
  syncfusion_flutter_datepicker: ^22.2.12  
  intl: ^0.18.1  
  modal_bottom_sheet: ^2.1.2  
  equatable: ^2.0.5  
  uuid: ^4.1.0  
  dots_indicator: ^3.0.0  
  firebase_core: ^2.25.3  
  firebase_auth: ^4.17.3  
  cloud_firestore: ^4.15.4
```

Code:

main.dart

```
import 'package:firebase_core/firebase_core.dart';  
import 'package:flutter/material.dart';  
import 'shared/navigation/app_router.dart';  
import 'shared/theme/colors.dart';  
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: FirebaseOptions(  
      apiKey: "AlzaSyCDqrgiDjoGFckVQkxb2proOWI_rDBLR3s",  
      appId: "1:95463089972:android:e8d8294c8f9d651f136761",  
      messagingSenderId: "95463089972",  
      projectId: "airbnb-a49ab"),  
    );  
  runApp(const MyApp());  
}
```

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

@override

```
Widget build(BuildContext context) {  
  return MaterialApp.router(  
    debugShowCheckedModeBanner: false,  
    theme: ThemeData(  
      useMaterial3: true,  
      colorScheme: ColorScheme.fromSeed(seedColor: appRed),  
    ),  
    routerConfig: AppRouter().router,  
  );  
}
```

login.dart

```
import 'package:flutter/material.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:airbnb_clone/screens/home_screen.dart';
```

```
class LoginPage extends StatelessWidget {  
  final FirebaseAuth _auth = FirebaseAuth.instance;  
  final TextEditingController _emailController = TextEditingController();  
  final TextEditingController _passwordController = TextEditingController();  
  
  Future<void> _signInWithEmailAndPassword(BuildContext context) async {  
    try {  
      final UserCredential userCredential =  
        await _auth.signInWithEmailAndPassword(  
          email: _emailController.text,  
          password: _passwordController.text,  
        );  
      if (userCredential.user != null) {  
        Navigator.pushReplacement(  
          context,  
          MaterialPageRoute(builder: (context) => HomeScreen()),  
        );  
      }  
    } catch (e) {  
      print("Failed to sign in with email and password: $e");  
      // Handle error messages here  
    }  
  }  
}
```

@override

```

Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [Colors.transparent, Colors.redAccent],
        ),
      ),
    ),
    child: Center(
      child: SingleChildScrollView(
        child: Container(
          padding: EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset(
                'assets/airbnb_logo.png',
                height: 100,
              ),
              SizedBox(height: 30.0),
              Text(
                'Login to continue',
                style: TextStyle(
                  fontSize: 28.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ),
              ),
              ),
              ),
              SizedBox(height: 50.0),
              TextField(
                controller: _emailController,
                decoration: InputDecoration(
                  labelText: 'Email',
                  border: OutlineInputBorder(),
                  fillColor: Colors.white,
                  filled: true,
                ),
              ),
              ),
              ),
              SizedBox(height: 20.0),
              TextField(
                controller: _passwordController,
                obscureText: true,
                decoration: InputDecoration(
                  labelText: 'Password',
                  border: OutlineInputBorder(),
                  fillColor: Colors.white,
                  filled: true,
                ),
              ),
              ),
            ),
          ),
        ),
      ),
    ),
  );
}

```

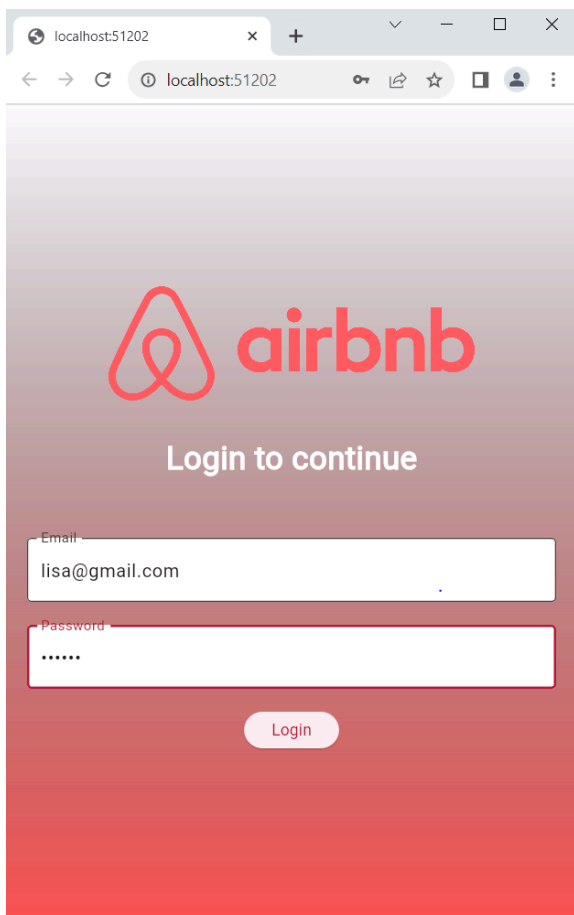
```

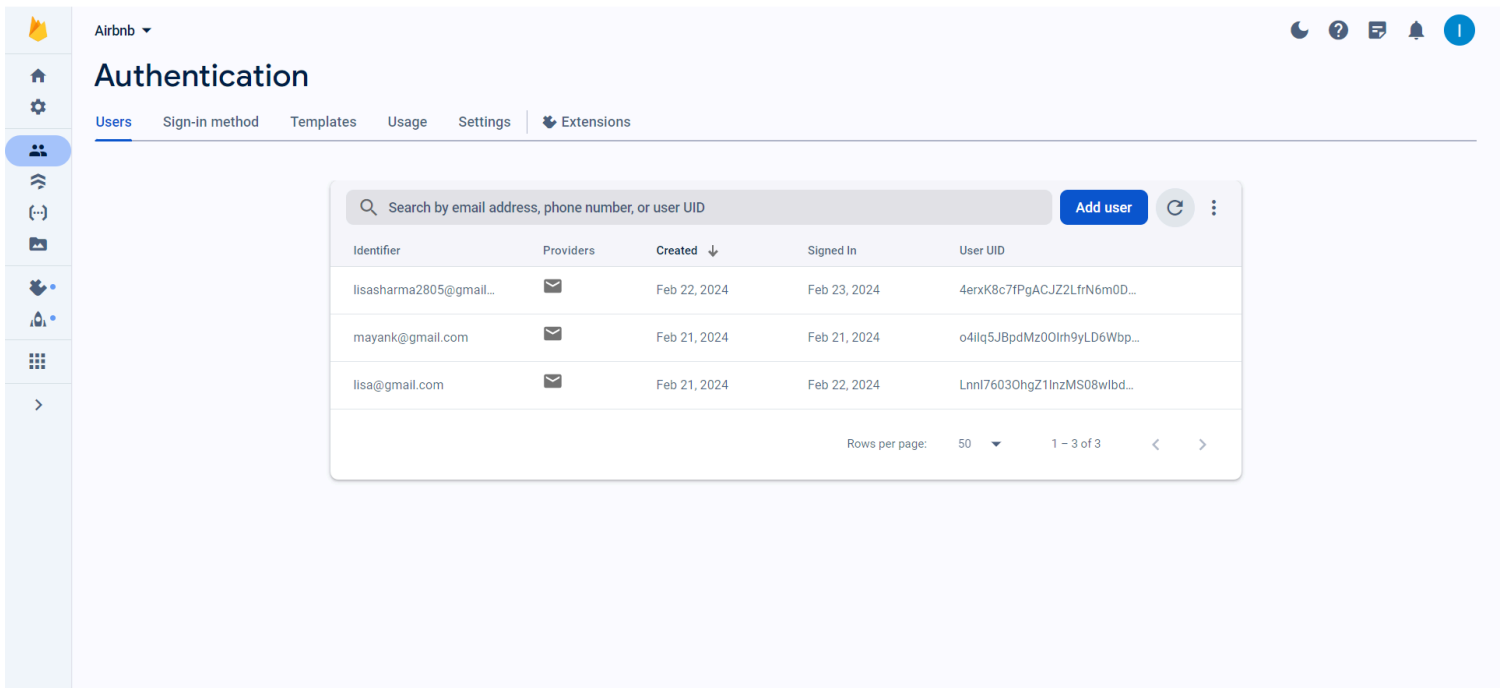
        SizedBox(height: 20.0),
        ElevatedButton(
          onPressed: () => _signInWithEmailAndPassword(context),
          child: Text('Login'),
        ),
      ],
    ),
  ),
),
),
),
),
);
}
}

void main() {
  runApp(MaterialApp(
    home: LoginPage(),
  ));
}

```

Output:





Conclusion:

In this experiment, we have successfully connected firebase database and authenticated using google signin and email and password with out flutter application successfully.