

EXPERIMENT NO: - 06

Name:- Ronak Katariya

Class:- D15A

Roll:No:- 23

AIM:- To connect Flutter UI with Firebase database.

Theory: -

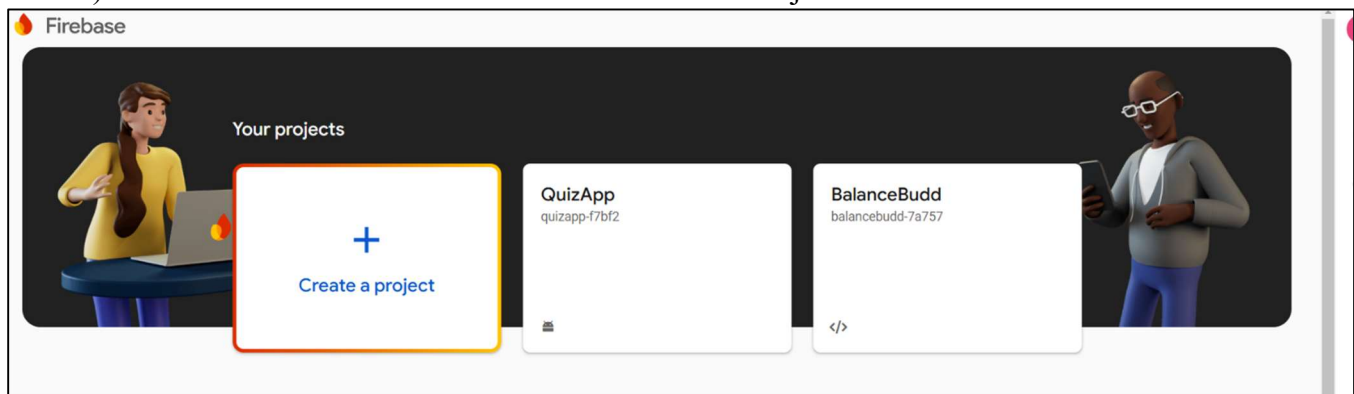
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

➤ **Steps to Connect Flutter UI with Firebase Database**

Step 1:

1.1) Go to Firebase Console and Create a Firebase Project



1.2) Click on Create a Project and give it a suitable name.

×

Create a project

Let's start with a name for your project[?]

Project name

Quiz

quiz-f4a32

ves.ac.in

☒

Join the [Google Developer Program](#) to enrich your developer journey with access to AI assistance, learning resources, profile badges, and more!

Already have a Google Cloud project?
[Add Firebase to Google Cloud project](#)

Continue

1.3) Enable Google Analytics (optional) & Click continue and complete the setup

×

Create a project

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, and Cloud Functions.

Google Analytics enables:

A/B testing[?]

Breadcrumb logs in Crashlytics[?]

User segmentation & targeting across Firebase products[?]

Event-based Cloud Functions triggers[?]

Free unlimited reporting[?]

☒

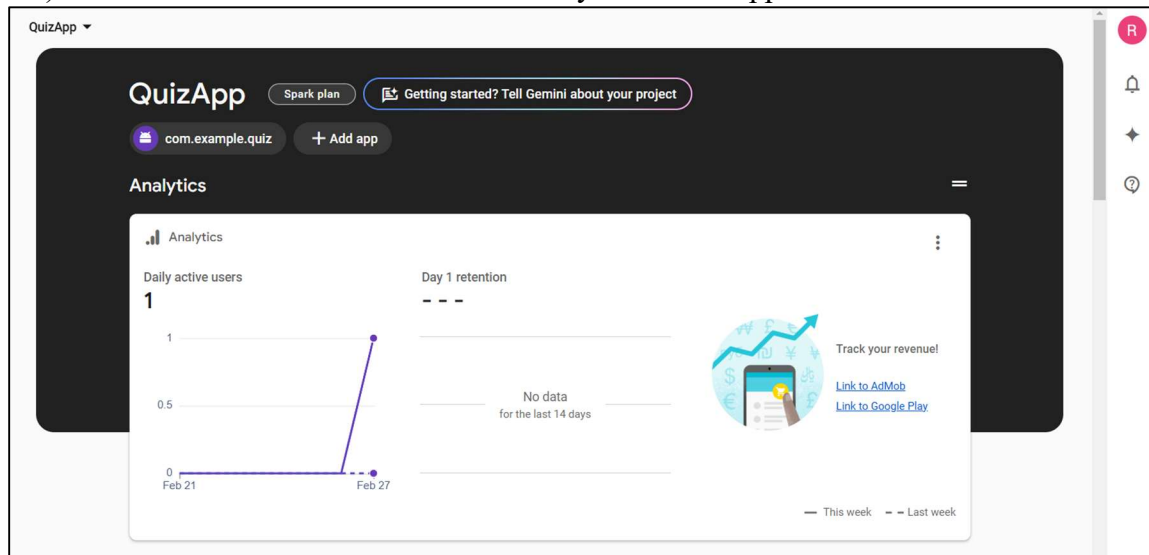
Enable Google Analytics for this project
Recommended

Previous

Continue

Step 2:- Add Firebase to Your Flutter App

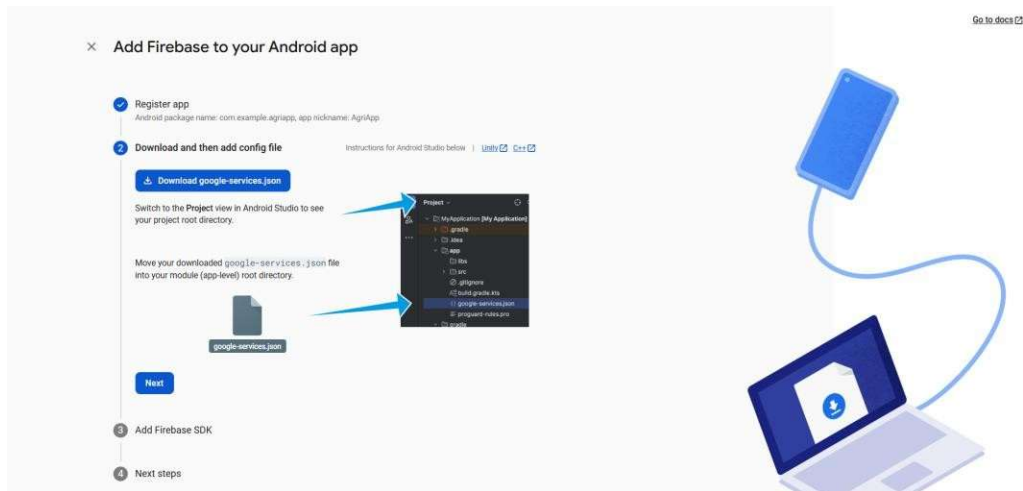
2.1) Click on Android/iOS/Web based on your Flutter application



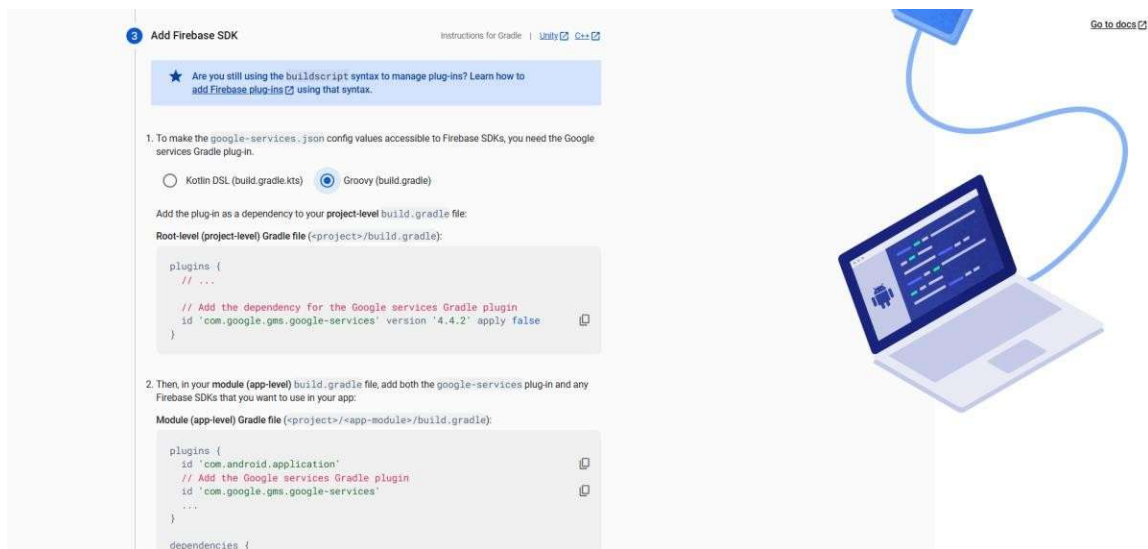
2.2) Register your app with a unique package name (found in android/app/build.gradle for Android).

The screenshot shows the 'Add Firebase to your Android app' wizard. The first step is 'Register app', which includes fields for 'Android package name' (com.example.agriapp), 'App nickname(optional)' (AgriApp), and 'Debug signing certificate SHA-1 (optional)' (a long hexadecimal string). A 'Register app' button is at the bottom of the first step. The subsequent steps are 'Download and then add config file', 'Add Firebase SDK', and 'Next steps'. On the right side of the wizard, there's an illustration of a smartphone connected to a laptop by a blue cable.

2.3) Download the google-services.json (for Android) & place the JSON file inside android/app/ directory.



2.4) Add Firebase SDK dependencies to android/build.gradle



Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies

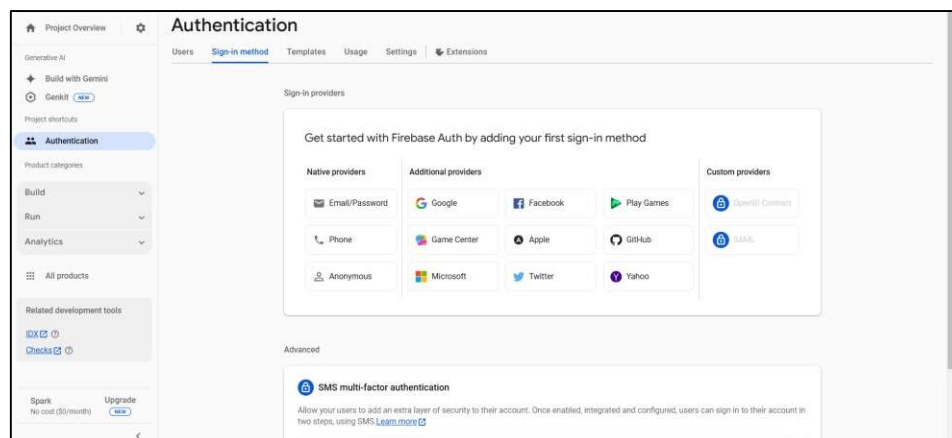
```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core: ^3.11.0  
  firebase_auth: ^5.4.2 # For authentication  
  cloud_firestore: ^5.6.3 # For Firestore, if you need it  
  firebase_messaging: ^15.2.2  
  http: ^0.13.3  
  image_picker: ^1.0.4  
  tflite_flutter: ^0.11.0  
  image: ^3.2.0  
  url_launcher: ^6.1.14
```

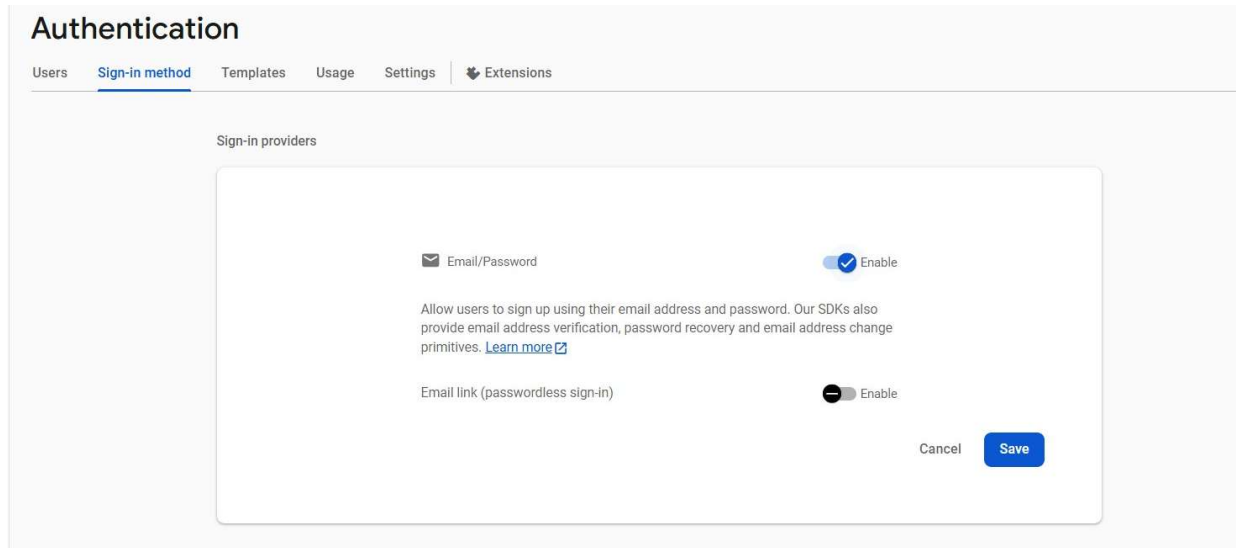
3.2) Enable Authentication in Firebase Console

Go to **Firebase Console** → **Authentication**.

Click on **Sign-in method** and enable **Email/Password** (or any other method like Google).

Click Save





3.3) Implement Authentication in Flutter

Modify main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

Step 4: -Configure Firebase Realtime Database

- 4.1) Go to Firebase Console → Realtime Database.
- 4.2) Click **Create Database** → Choose location → Set rules (for development, set read/write to true).
- 4.3) Click **Publish**.

Code:

Signup screen.dart

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

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

  @override
  State<SignUpScreen> createState() =>
  _SignUpScreenState();
}

class _SignUpScreenState extends
State<SignUpScreen> {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController
_passwordController = TextEditingController();
  bool _isLoading = false;

  Future<void> _signUp(BuildContext context)
  async {
    setState(() => _isLoading = true);

    try {
      await
FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: _emailController.text.trim(),
      password: _passwordController.text.trim(),
    );
    Navigator.pushReplacement(context,
MaterialPageRoute(builder: (_) => const
LoginScreen()));
  } on FirebaseAuthException catch (e) {

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(e.message ??
"Signup failed")),
    );
  }

  setState(() => _isLoading = false);
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(colors:
[Color(0xFF4A00E0), Color(0xFF8E2DE2)],
        begin: Alignment.topCenter, end:
Alignment.bottomCenter),
    ),
    child: Center(
      child: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            _buildTextField(Icons.email, "Email",
controller: _emailController),
            const SizedBox(height: 15),
            _buildTextField(Icons.lock, "Password",
isPassword: true, controller: _passwordController),
            const SizedBox(height: 20),
            ElevatedButton(
              onPressed: _isLoading ? null : () =>
_signUp(context),
              style: _buttonStyle(),
              child: _isLoading ? const
CircularProgressIndicator(color:
Colors.deepPurple) : const Text("Sign Up", style:
TextStyle(fontSize: 18)),
            ),
          ],
        ),
      ),
    ),
  );
}

Widget _buildTextField(IconData icon, String
hint, {bool isPassword = false, required
TextEditingController controller}) {
```

```

return TextField(controller: controller,
obscureText: isPassword, decoration:
InputDecoration(prefixIcon: Icon(icon)));
}

```

```

ButtonStyle _buttonStyle() =>
ElevatedButton.styleFrom(minimumSize: const
Size(double.infinity, 50));
}

```

Login_screen.dart:

```

import 'package:flutter/material.dart';
import
'package:firebase_auth/firebase_auth.dart';
import
'package:google_sign_in/google_sign_in.dart';
import 'signup_screen.dart';
import 'landing_page.dart';

```

```

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

```

```

  @override
  State<LoginScreen> createState() =>
  _LoginScreenState();
}

```

```

class _LoginScreenState extends
State<LoginScreen> {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController
_passwordController = TextEditingController();
  bool _isLoading = false;

```

```

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

```

```

  Future<void> _login(BuildContext context)
  async {
    setState(() => _isLoading = true);

```

```

    try {
      await
      FirebaseAuth.instance.signInWithEmailAndPasswo
      rd(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const
        LandingPage()),
      );
    } on FirebaseAuthException catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text(e.message ?? "Login
        failed")),
      );
    }
  }

```

```

    setState(() => _isLoading = false);
  }

```

```

  Future<void> _signInWithGoogle(BuildContext
  context) async {
    try {
      final GoogleSignInAccount? googleUser =
      await GoogleSignIn().signIn();
      if (googleUser == null) return;

      final GoogleSignInAuthentication googleAuth =
      await googleUser.authentication;
      final credential =
      GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );

```



```

    await
    FirebaseAuth.instance.signInWithCredential(credential);
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) =>
const LoginPage()),
    );
  } catch (e) {

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text("Google Sign-In
failed")),
);
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topCenter,
          end: Alignment.bottomCenter,
          colors: [Color(0xFF4A00E0),
Color(0xFF8E2DE2)],
        ),
      ),
      child: Center(
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              const Text(
                "Welcome Back",
                style: TextStyle(fontSize: 28,
fontWeight: FontWeight.bold, color:
Colors.white),
              ),
              const SizedBox(height: 10),
              const Text(
                "Enter your credentials to login",
                style: TextStyle(fontSize: 16, color:
Colors.white70),
              ),
              const SizedBox(height: 30),
              _buildTextField(Icons.email, "Email",

```

```

controller: _emailController),
              const SizedBox(height: 15),
              _buildTextField(Icons.lock, "Password",
isPassword: true, controller: _passwordController),
              const SizedBox(height: 20),
              ElevatedButton(
                onPressed: _isLoading ? null : () =>
_login(context),
                style: _buttonStyle(),
                child: _isLoading
                  ? const
CircularProgressIndicator(color:
Colors.deepPurple)
                  : const Text("Login", style:
TextStyle(fontSize: 18)),
              ),
              const SizedBox(height: 10),
              _googleSignInButton(),
              const SizedBox(height: 10),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const Text("Don't have an account?",
style: TextStyle(color: Colors.white)),
                  TextButton(
                    onPressed: () =>
Navigator.push(context,
MaterialPageRoute(builder: (_) => const
SignUpScreen())),
                    child: const Text("Sign Up", style:
TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
                ],
              ),
            ],
          ),
        ),
      );
    }

Widget _buildTextField(IconData icon, String
hint, {bool isPassword = false, required
TextEditingController controller}) {
  return TextField(
    controller: controller,
    obscureText: isPassword,

```

```

style: const TextStyle(color: Colors.white),
decoration: InputDecoration(
  prefixIcon: Icon(icon, color: Colors.white),
  hintText: hint,
  hintStyle: const TextStyle(color:
Colors.white70),
  filled: true,
  fillColor: Colors.white.withOpacity(0.2),
  border: OutlineInputBorder(borderRadius:
BorderRadius.circular(30), borderSide:
BorderSide.none),
),
);
}

```

```

Widget _googleSignInButton() {
  return ElevatedButton.icon(
    onPressed: () =>
_signInWithGoogle(context),
    style: _buttonStyle(),
    icon:
Image.asset("assets/images/google.png", height:
24),
    label: const Text("Sign In with Google",
style: TextStyle(fontSize: 16)),
  );
}

```

```

ButtonStyle _buttonStyle() {
  return ElevatedButton.styleFrom(
    backgroundColor: Colors.white,
    foregroundColor: Colors.deepPurple,
    minimumSize: const Size(double.infinity,
50),
    shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(30)),
  );
}
}

```

Auth service.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import
'package:cloud_firestore/cloud_firestore.dart';
import
'package:google_sign_in/google_sign_in.dart';
```

```
class AuthService {
  final FirebaseAuth _auth =
  FirebaseAuth.instance;
  final FirebaseFirestore _firestore =
  FirebaseFirestore.instance;

  // Sign Up
  Future<User?> signUp(String email, String
  password) async {
    try {
      UserCredential userCredential = await
      _auth.createUserWithEmailAndPassword(
        email: email,
        password: password,
      );
      User? user = userCredential.user;

      // Save user details in Firestore
      await
      _firestore.collection('users').doc(user!.uid).set({
        'email': email,
        'createdAt': DateTime.now(),
      });

      return user;
    } catch (e) {
      return null;
    }
  }
}
```

```
// Sign In
Future<User?> signIn(String email, String
password) async {
  try {
    UserCredential userCredential = await
    _auth.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
    return userCredential.user;
  } catch (e) {
    return null;
  }
}
```

```
// Google Sign-In
Future<User?> googleSignIn() async {
  final GoogleSignInAccount? googleUser = await
  GoogleSignIn().signIn();
  final GoogleSignInAuthentication? googleAuth =
  await googleUser?.authentication;
```

```
  if (googleAuth == null) return null;
```

```
  final credential = GoogleAuthProvider.credential(
    accessToken: googleAuth.accessToken,
    idToken: googleAuth.idToken,
  );
```

```
  UserCredential userCredential = await
  _auth.signInWithCredential(credential);
  return userCredential.user;
}
```

```
// Sign Out
Future<void> signOut() async {
  await _auth.signOut();
}
}
```

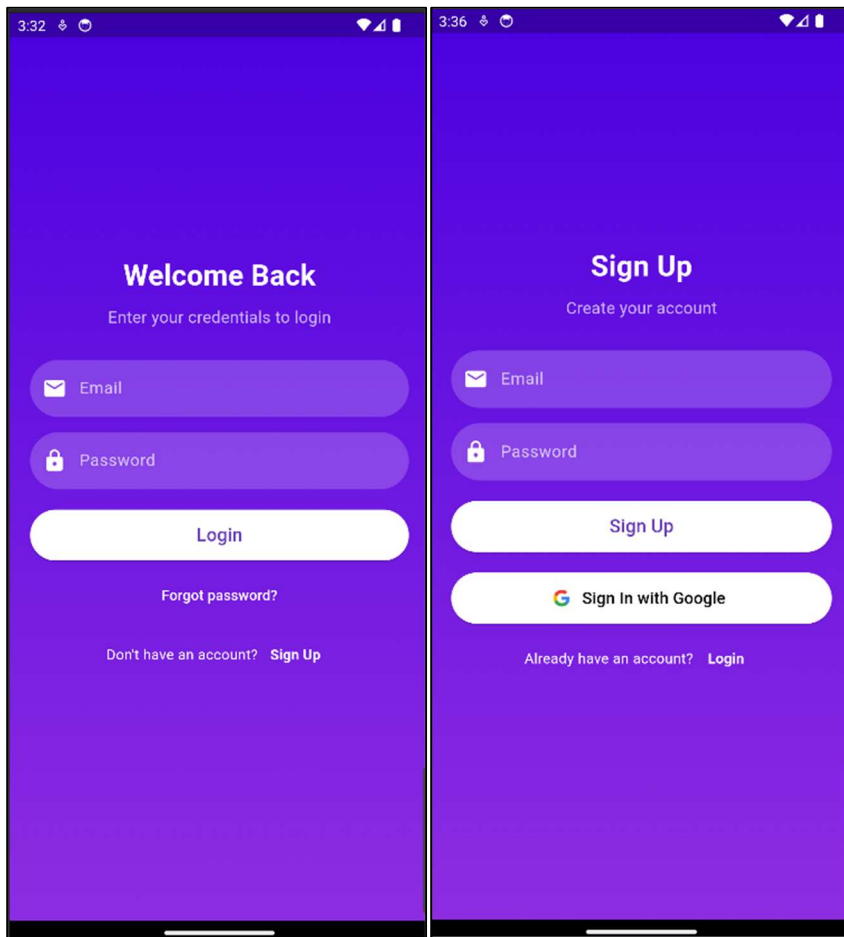
Main.dart

```
import
'package:flutter/material.dart';
import
'package:firebase_core/firebase_core.dart';
import
'package:quiz/screens/landing_page.dart';
import
'package:quiz/screens/category_selection_scre
n.dart';
import 'package:quiz/screens/leaderboard.dart';
import 'package:quiz/screens/profile.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Quiz App',
      theme: ThemeData.dark(),
      home: const LandingPage(),
    );
  }
}
```



QuizApp ▾

Authentication

[Users](#) | [Sign-in method](#) | [Templates](#) | [Usage](#) | [Settings](#) | [Extensions](#)

The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

Search by email address, phone number, or user UID

Add user

Identifier	Providers	Created ▾	Signed In	User UID
2022.ronak.katariya@v...		Feb 27, 2025	Feb 27, 2025	jlxCXmUEVCRvHlflaFJovamAx...

Rows per page: 50 ▾ 1 - 1 of 1 < >

After Registering, the user details is saved in the database.

User details get fetched from the database in the profile page.

