Experiment – 6: Connecting Flutter UI with firebase database.

Aim: To connect Flutter UI with firebase database.
Objectives: After study of this experiment, the student will be able to

- Create a production ready Flutter App by including files and firebase backend service.

Outcomes: After study of this experiment, the student will be able to

- Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android/iOS

Prerequisite: Dart Programming Language.

Requirements: Android Studio, Flutter framework, Internet Connection.

Pre-Experiment Exercise:
Brief Theory:

Firebase is a platform developed by Google for creating mobile and web applications. Firebase is a set of server-side services and tools. If you use Firebase, you don’t need to buy or rent your own server. Firebase is made up of over a dozen tools. Let’s glance at these three:

- Cloud Firestore – A database with an API to read and write data
- Cloud Functions – Logic that is kicked off by an API call
- Authentication – Single sign-on to allow users to securely log in to your app using their social accounts or a username/password combination

Steps to connect UI with firebase database:

1. Visit http://firebase.google.com to register an app with Firebase.
2. Create a Firebase Project.
3. Create the database by using the option given on your project’s dashboard.
4. Create an Android app.
   a. Going through the next step while creating an android app, the wizard will create a google-services.json file and will tell you where to save it.
5. Add FlutterFire plugins.

Laboratory Exercise
A. Program

1. Connect your UI with the firebase database and show the values entered in UI being reflected in the database.

B. Result/Observation

Print out of program code and output.

Post-Experiments Exercise

A. Questions:

1. What are the other different databases that you can use to connect your UI?
2. Demonstrate the use of firebase Authentication using UI.

B. Conclusion:

1. Write what you have learnt in the experiment.

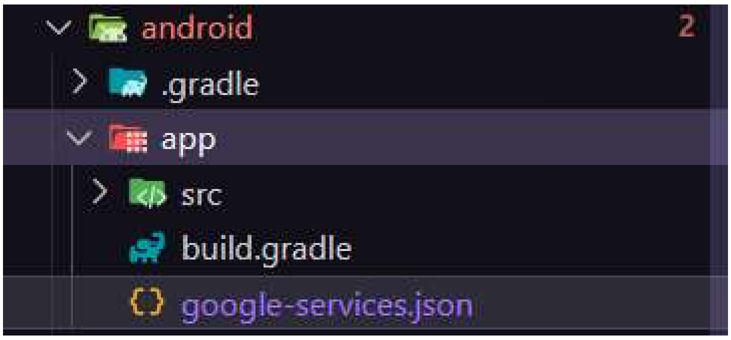
C. References:

2. Beginning App Development with Flutter: Create Cross-Platform Mobile Apps, By Rap Payne, 2019.
3. Google Flutter Mobile Development Quick Start Guide, Packt Publishing, 2019.

Name: Vishal Rajesh Mahajan MAD EXP 06

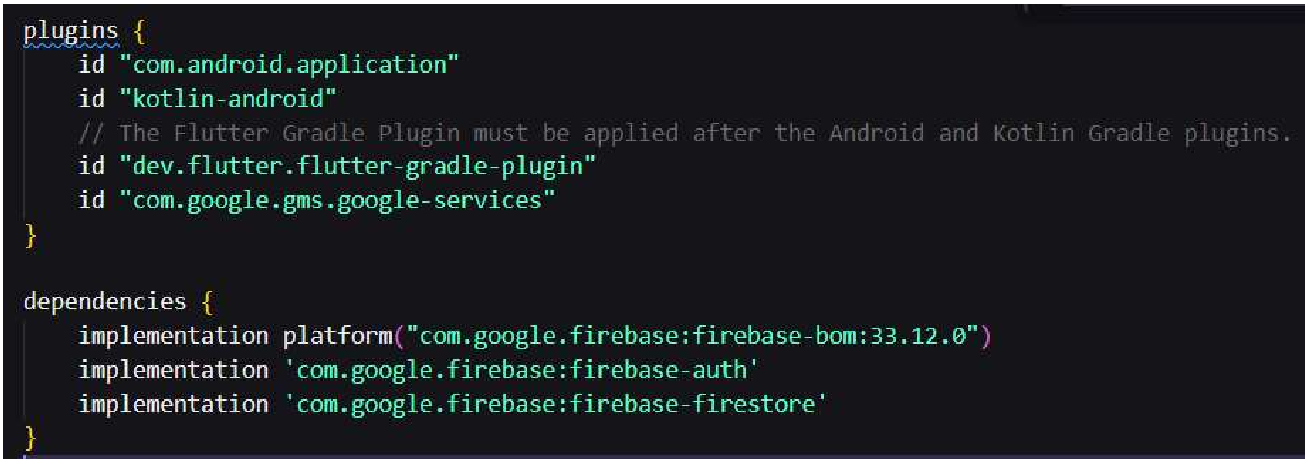
Class: TE IT A Roll No: 56

Create a Firebase app, download the google-services.json file, and place it in the app directory.



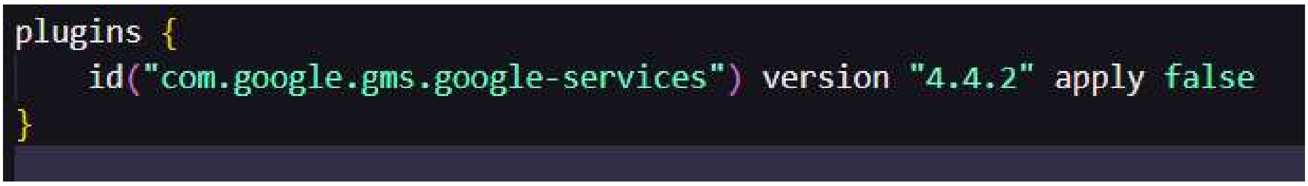
```
android 2
  .gradle
  app
    src
    build.gradle
    google-services.json
```

Modify the build.gradle in the app directory.



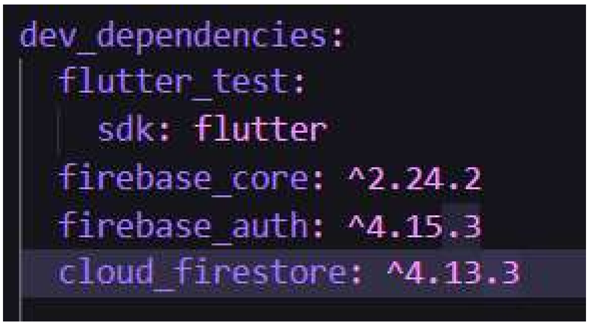
```
plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
    id "dev.flutter.flutter-gradle-plugin"
    id "com.google.gms.google-services"
}

dependencies {
    implementation platform("com.google.firebase:firebase-bom:33.12.0")
    implementation 'com.google.firebase:firebase-auth'
    implementation 'com.google.firebase:firebase-firestore'
}
```

Modify the build.gradle in the ANdroid directory.



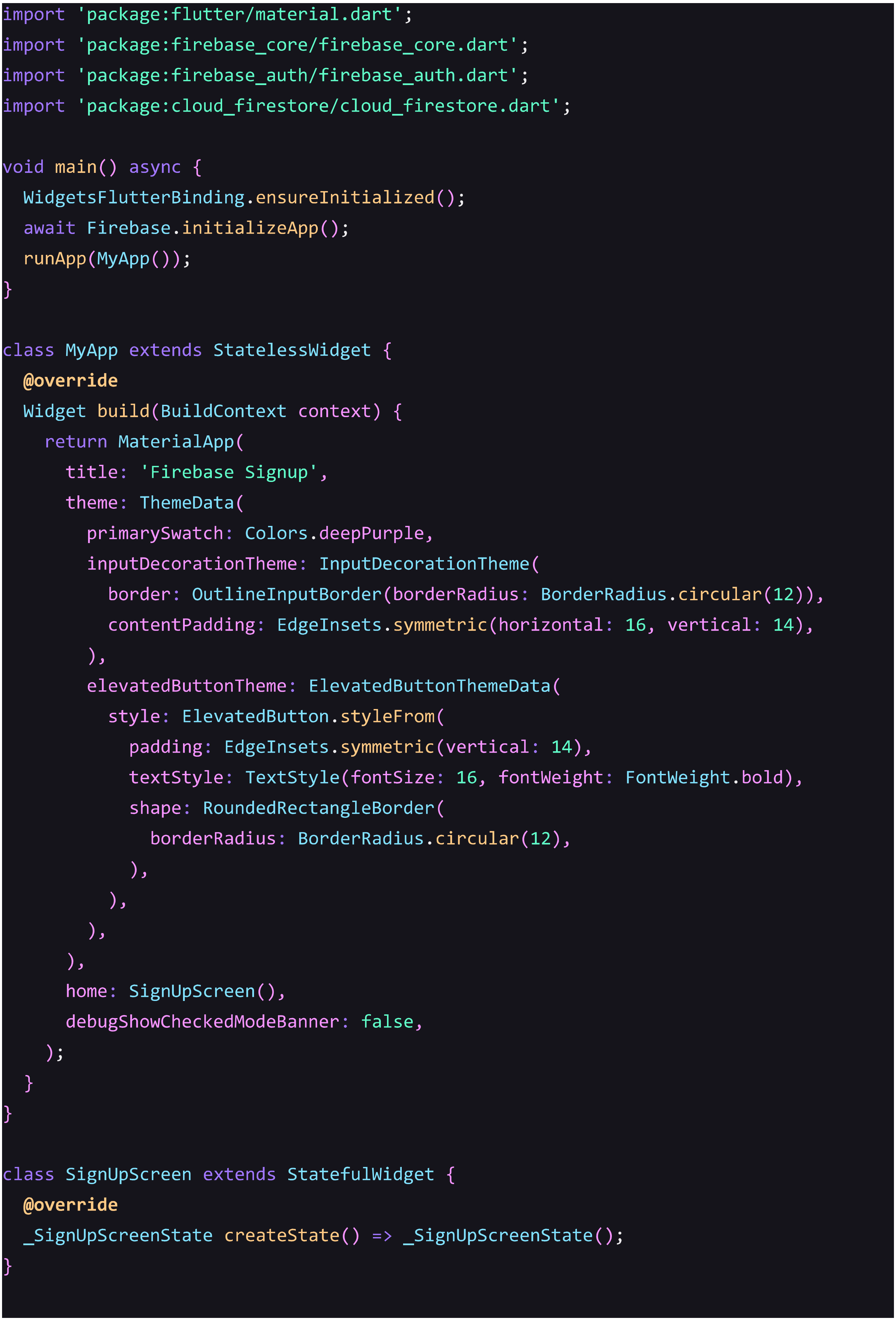
```
plugins {
    id("com.google.gms.google-services") version "4.4.2" apply false
}
```

Add Firebase dependencies in pubspec.yaml and run flutter pub get.

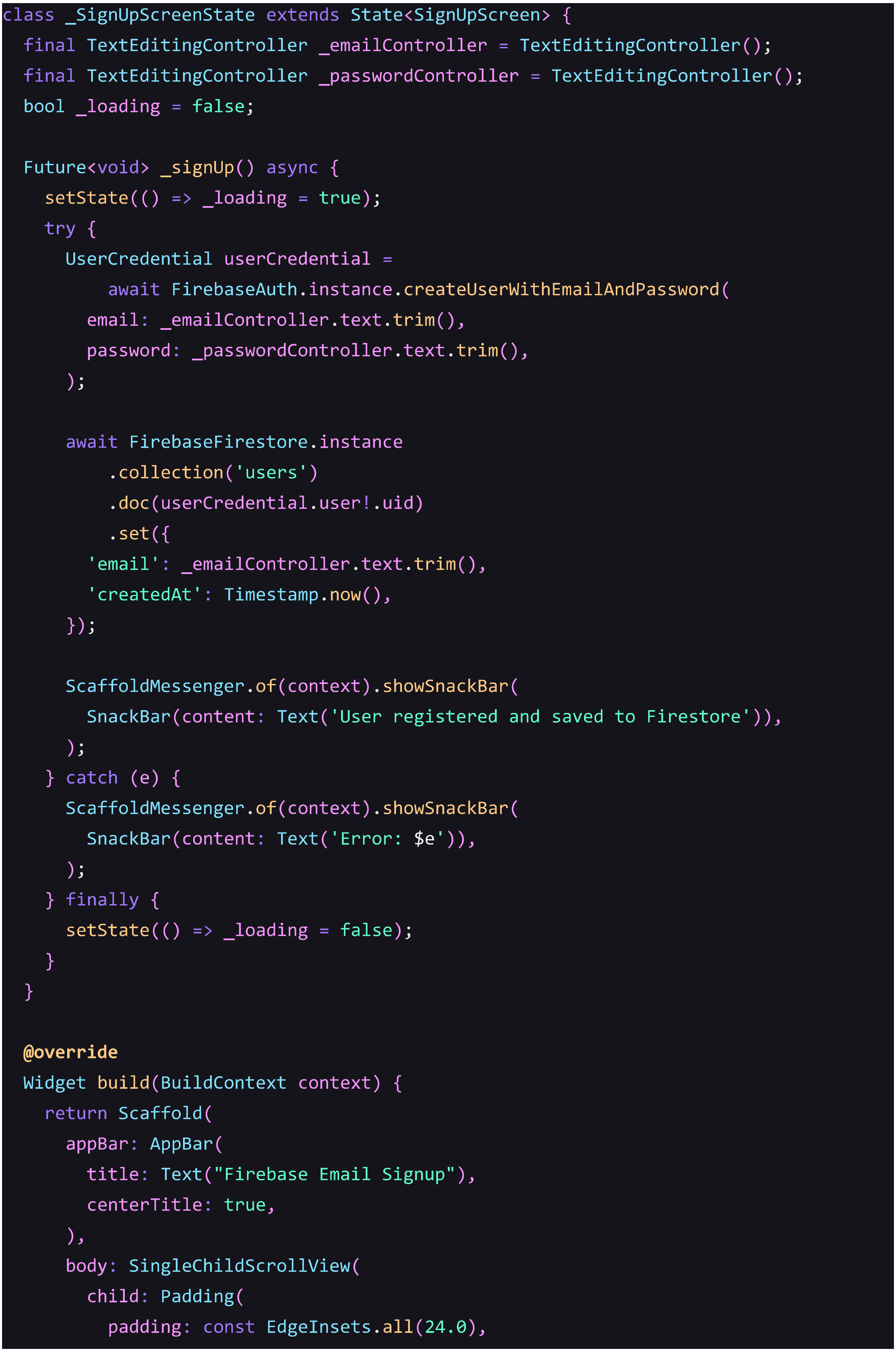


```
dev_dependencies:
  flutter_test:
    sdk: flutter
  firebase_core: ^2.24.2
  firebase_auth: ^4.15.3
  cloud_firestore: ^4.13.3
```

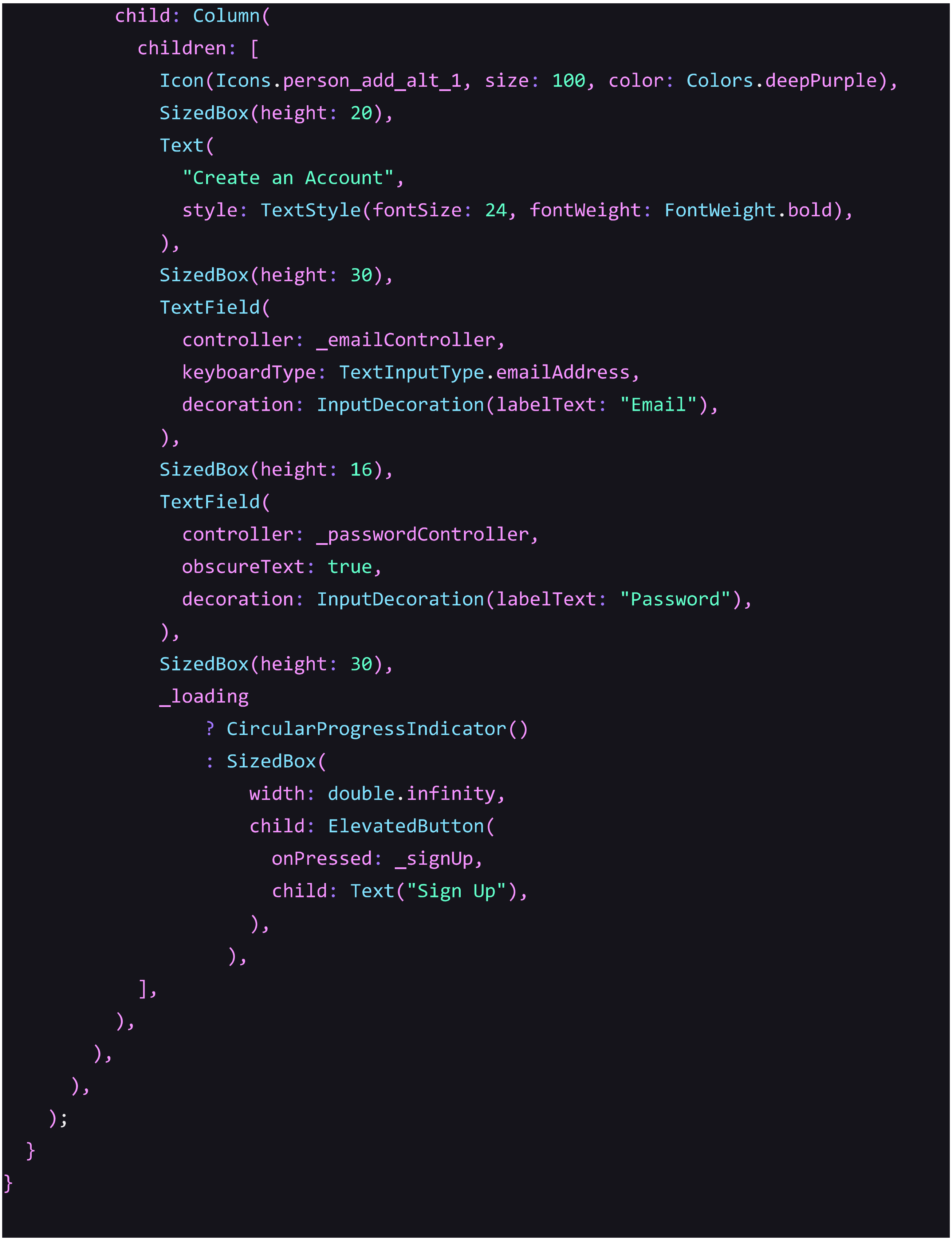
Code:



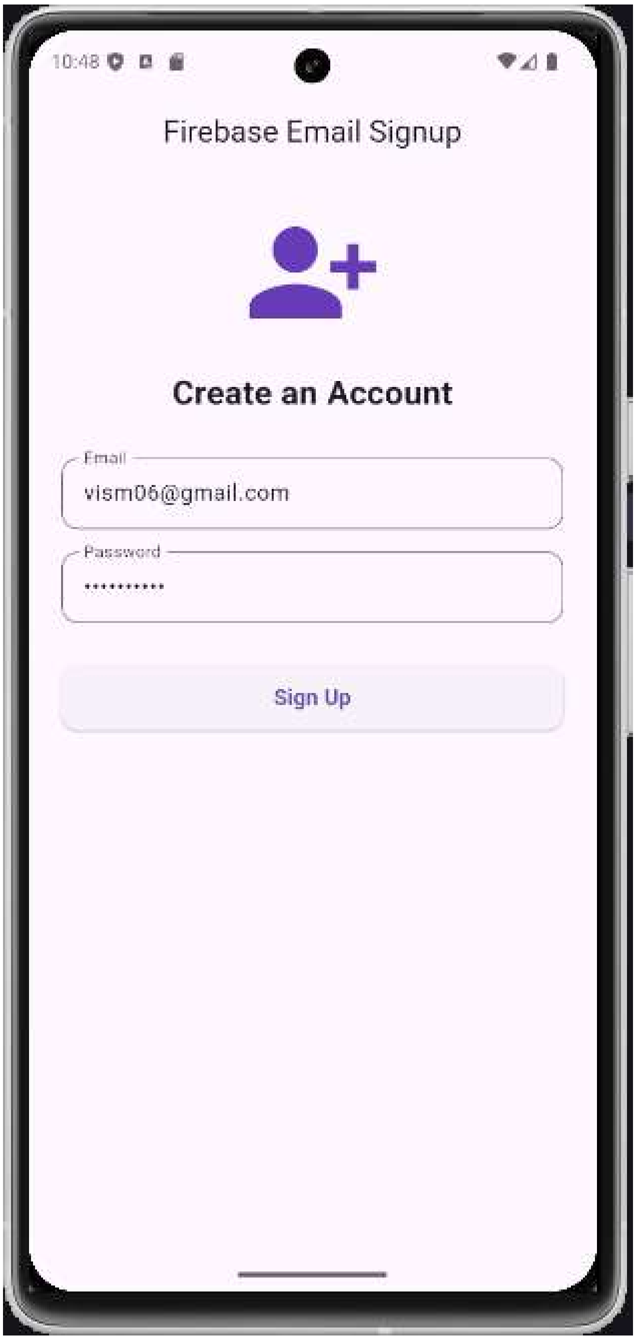
```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Firebase Signup',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
        inputDecorationTheme: InputDecorationTheme(
          border: OutlineInputBorder(borderRadius: BorderRadius.circular(12)),
          contentPadding: EdgeInsets.symmetric(horizontal: 16, vertical: 14),
        ),
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            padding: EdgeInsets.symmetric(vertical: 14),
            textStyle: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(12),
            ),
          ),
        ),
      ),
      home: SignUpScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class SignUpScreen extends StatefulWidget {
  @override
  _SignUpScreenState createState() => _SignUpScreenState();
}
```



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

  Future<void> _signUp() async {
    setState(() => _loading = true);
    try {
      UserCredential userCredential =
          await FirebaseAuth.instance.createUserWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
      );

      await FirebaseFirestore.instance
          .collection('users')
          .doc(userCredential.user!.uid)
          .set({
        'email': _emailController.text.trim(),
        'createdAt': Timestamp.now(),
      });

      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('User registered and saved to Firestore')),
      );
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Error: $e')),
      );
    } finally {
      setState(() => _loading = false);
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Firebase Email Signup"),
        centerTitle: true,
      ),
      body: SingleChildScrollView(
        child: Padding(
          padding: const EdgeInsets.all(24.0),
```



```
          child: Column(
            children: [
              Icon(Icons.person_add_alt_1, size: 100, color: Colors.deepPurple),
              SizedBox(height: 20),
              Text(
                "Create an Account",
                style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
              ),
              SizedBox(height: 30),
              TextField(
                controller: _emailController,
                keyboardType: TextInputType.emailAddress,
                decoration: InputDecoration(labelText: "Email"),
              ),
              SizedBox(height: 16),
              TextField(
                controller: _passwordController,
                obscureText: true,
                decoration: InputDecoration(labelText: "Password"),
              ),
              SizedBox(height: 30),
              _loading
                  ? CircularProgressIndicator()
                  : SizedBox(
                      width: double.infinity,
                      child: ElevatedButton(
                        onPressed: _signUp,
                        child: Text("Sign Up"),
                      ),
                    ),
            ],
          ),
        ),
      ),
    );
  }
}
```

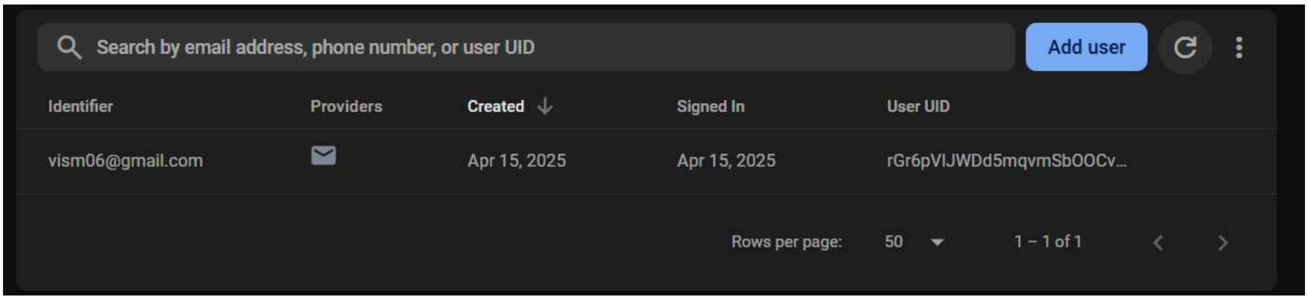


Firebase Email Signup

Create an Account

Email: vism06@gmail.com

Password: ***********



Identifier	Providers	Created ↓	Signed In	User UID
vism06@gmail.com	✉	Apr 15, 2025	Apr 15, 2025	rGr6pVIJWDd5mqvmSbOOCv...

Rows per page: 50 1 – 1 of 1