

## **EXPERIMENT NO: - 06**

**Name:-** Tejas Gunjal

**Class:-** D15A

**Roll:No: -** 18

**AIM: -** To connect Flutter UI with Firebase database.

---

### **Theory: -**

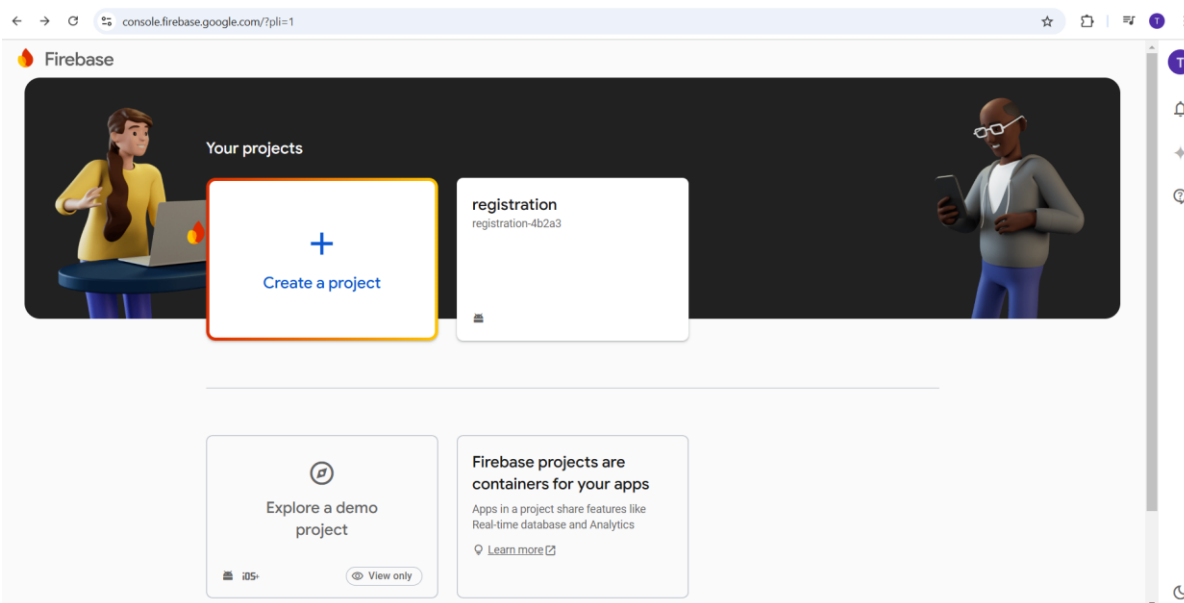
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend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

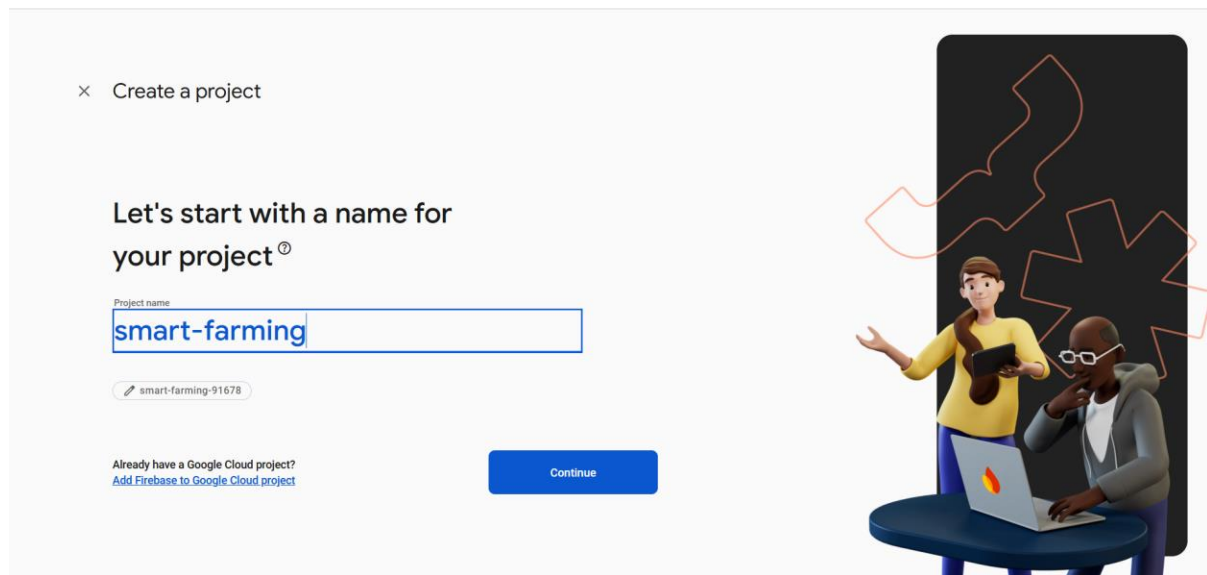
### ➤ **Steps to Connect Flutter UI with Firebase Database**

#### **Step 1:**

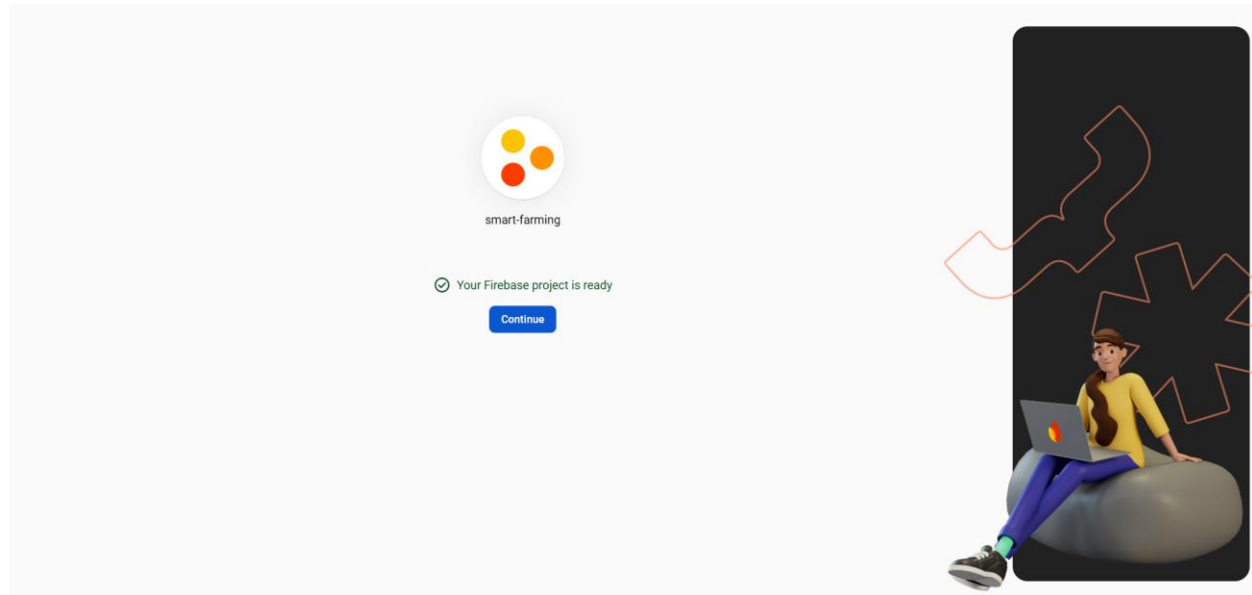
1.1) Go to Firebase Console and Create a Firebase Project



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



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



Add Firebase to your Android app

Go to docs

1 Register app

Android package name ⓘ  
com.example.agriapp

App nickname(optional) ⓘ  
AgriApp

Debug signing certificate SHA-1 (optional) ⓘ  

[0:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00]


ⓘ Required for Dynamic Links and Google Sign-in or phone number support in Auth.  
Edit SHA-1s in settings.

Register app

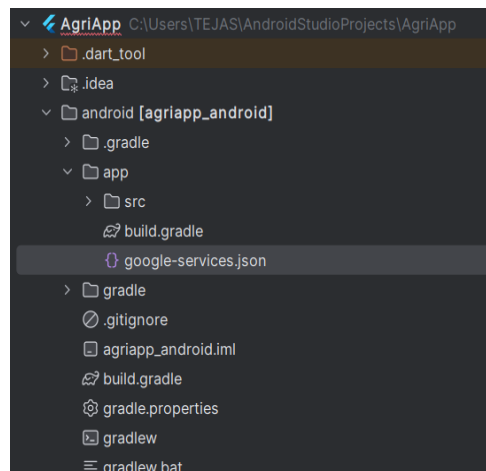
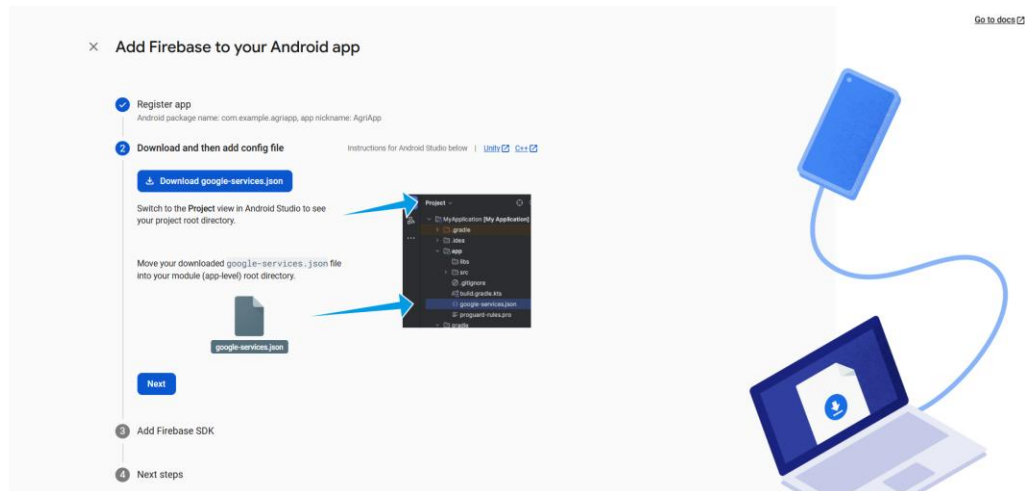
2 Download and then add config file

3 Add Firebase SDK

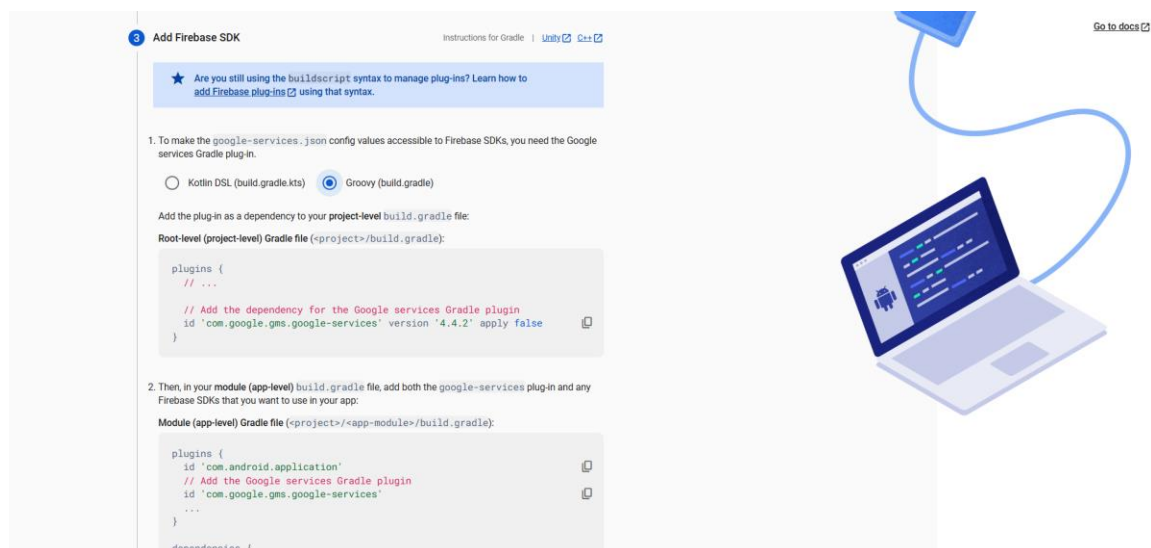
4 Next steps

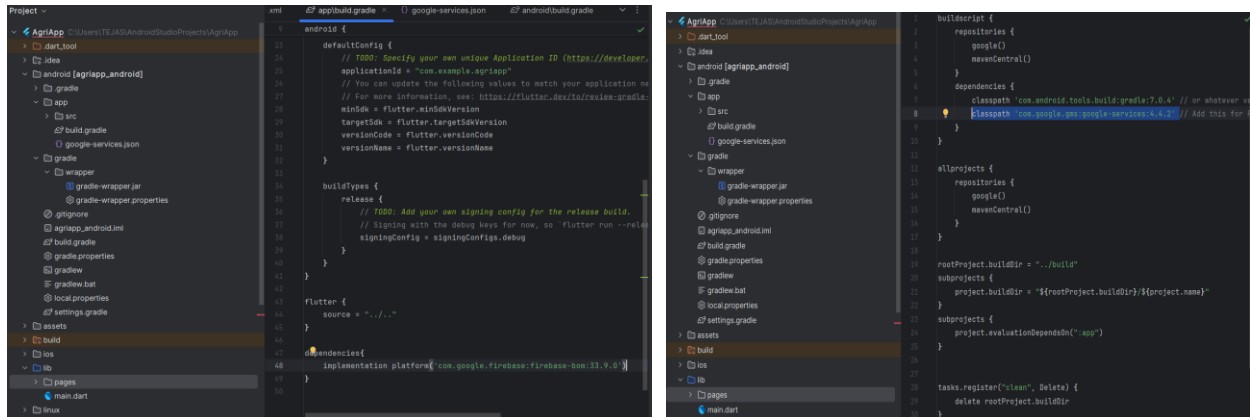


2.3) Download the google-services.json (for Android) & place the JSON file inside android/app/ directory.



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

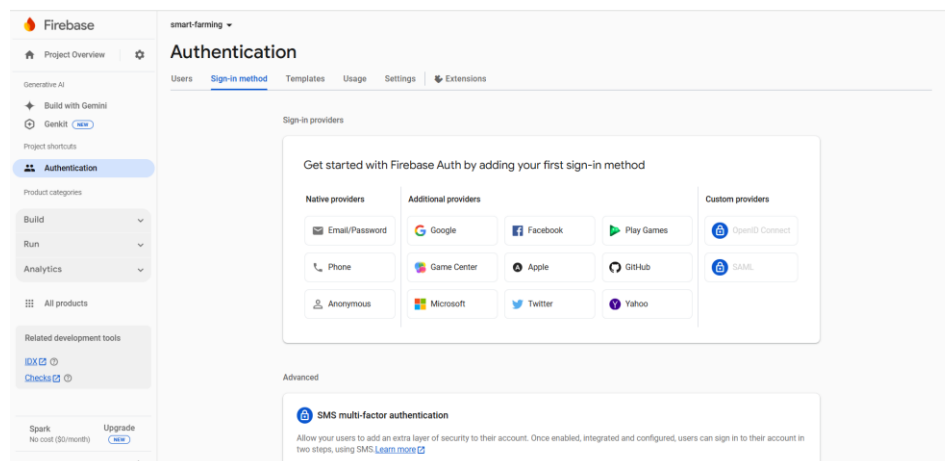
  firebase_core: ^3.11.0
  firebase_auth: ^5.4.2 # For authentication
  cloud_firestore: ^5.6.3 # For Firestore, if you need it
  firebase_messaging: ^15.2.2
  http: ^0.13.3
  image_picker: ^1.0.4
  tflite_flutter: ^0.11.0
  image: ^3.2.0
  url_launcher: ^6.1.14
```

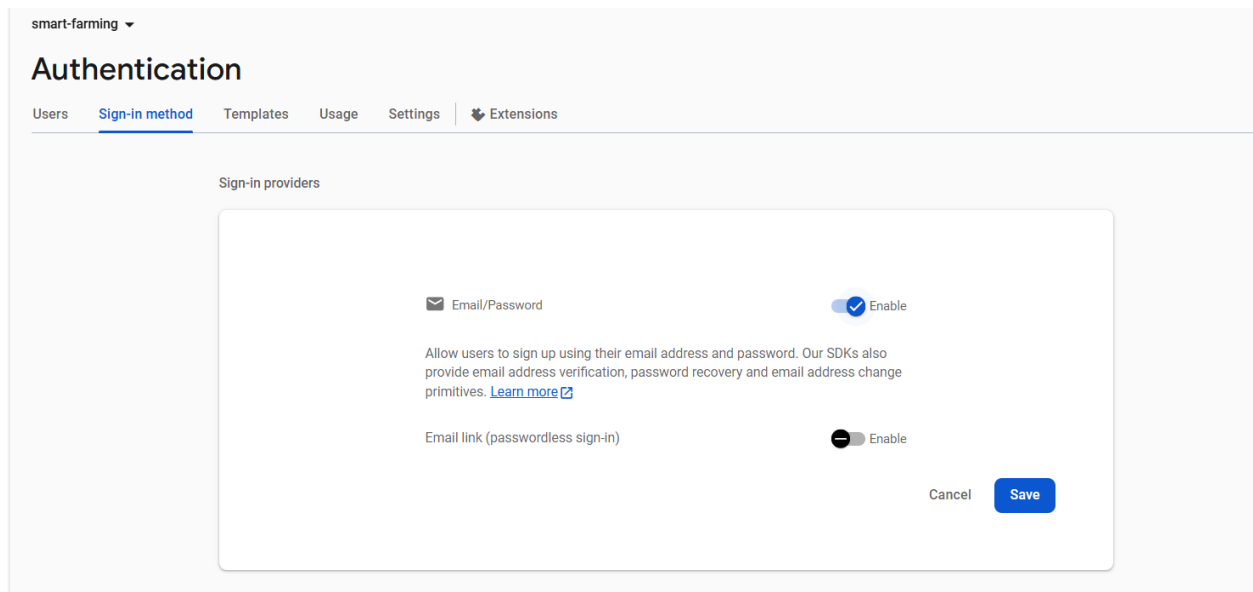
### 3.2) Enable Authentication in Firebase Console

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:-

### Register\_page.dart

```
class _RegistrationPageState extends
State<RegistrationPage> {
  final _formKey =
GlobalKey<FormState>();
  bool _isLoading = false;
  String _name = "";
  String _email = "";
  String _state = "";
  String _district = "";
  String _password = "";
  String _phone = "";

  final FirebaseAuth _auth =
FirebaseAuth.instance;

  // Firebase Registration
  Future<void> _handleRegistration() async
{
  if (_formKey.currentState!.validate()) {
    setState(() {
      _isLoading = true;
    });

    try {
      // Register user with Firebase
      Authentication (Email and Password)
      UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(
        email: _email,
        password: _password,
      );

      // Get the user ID from Firebase
      Authentication user
      String userId =
userCredential.user!.uid;

      // Store extra user information in
      Firestore
```

```
      await
FirebaseFirestore.instance.collection('users')
.doc(userId).set( {
      'name': _name,
      'email': _email,
      'phone': _phone,
      'state': _state,
      'district': _district,
      // You can add other fields here
    });

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

    // Navigate to OTP verification page
    after successful registration
    Navigator.pushReplacementNamed(
      context,
      '/otp',
      arguments: _phone, // Pass the phone
      number here
    );
  } catch (e) {
    setState(() {
      _isLoading = false;
    });

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text("Registration
failed: $e")),
    );
  }
}
```

### **login\_page.dart**

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

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() =>
  _LoginPageState();
}

class _LoginPageState extends
State<LoginPage> {
  final _formKey =
GlobalKey<FormState>();
  final FirebaseAuth _auth =
FirebaseAuth.instance;

  TextEditingController _emailController =
TextEditingController();
  TextEditingController _passwordController
= TextEditingController();

  bool _isLoading = false;
  String _errorMessage = "";

  Future<void> _loginUser() async {
    if (!_formKey.currentState!.validate())
return;

    setState() {
      _isLoading = true;
      _errorMessage = "";
    });

    try {
      UserCredential userCredential = await
_auth.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password:
_passwordController.text.trim(),
      );
```

```
User? user = userCredential.user;

    if (user != null) {
      Navigator.pushReplacementNamed(context,
'/home');
    } else {
      setState() {
        _errorMessage = "Something went
wrong. Please try again.";
      });
    }
  } on FirebaseAuthException catch (e) {
    setState() {
      _errorMessage = e.message ?? "An
error occurred. Please try again.";
    });
  }

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



## Myaccountpage.dart

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

class MyAccountPage extends
StatefulWidget {
  @override
  _MyAccountPageState createState() =>
  _MyAccountPageState();
}

class _MyAccountPageState extends
State<MyAccountPage> {
  final _formKey =
GlobalKey<FormState>();
  String name = "";
  String phone = "";
  String email = "";
  String state = "";
  String language = 'English';
  List<String> myCrops = [];
  bool isLoading = true;

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

  // Fetch data from Firestore
  _fetchUserData() async {
    User? user =
FirebaseAuth.instance.currentUser;

    if (user != null) {
      try {
        DocumentSnapshot doc = await
FirebaseFirestore.instance
```

```
.collection('users')
.doc(user.uid)
.get();

if (doc.exists) {
  setState(() {
    name = doc['name'];
    phone = doc['phone'];
    email = doc['email'];
    state = doc['state'];
    language = doc['language'];
    myCrops =
List<String>.from(doc['myCrops']);
    isLoading = false;
  });
}
} catch (e) {
  print("Error fetching user data: $e");
  setState(() {
    isLoading = false;
  });
}
}

// Save updated data to Firestore
_saveUserData() async {
  User? user =
FirebaseAuth.instance.currentUser;

  if (user != null) {
    try {
      await
FirebaseFirestore.instance.collection('users')
.doc(user.uid).update({
        'name': name,
        'phone': phone,
        'email': email,
        'state': state,
        'language': language,
        'myCrops': myCrops,
      });
    }
```

```

        // Show success message

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('Changes saved!')));
    } catch (e) {
      print("Error saving user data: $e");

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('Error saving changes')));
    }
  }
}

```

### **Main.dart**

```

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'AgriApp',
      theme: ThemeData(
        primarySwatch: Colors.green,
        colorScheme:
ColorScheme.fromSeed(seedColor:
Color(0xFF6A9A5B)),
      ),
      home:
FirebaseAuth.instance.currentUser == null ?
LoginPage() : MainScreen(),
      routes: {
        '/login': (context) => LoginPage(),
        '/register': (context) =>

```

```

RegistrationPage(),
        '/otp': (context) =>
OtpVerificationPage(
  phoneNumber:
ModalRoute.of(context)!.settings.arguments
as String,
    ),
        '/home': (context) => MainScreen(),
      },
    );
  }
}

```

```

class MainScreen extends StatefulWidget {
  @override
  _MainScreenState createState() =>
  _MainScreenState();
}

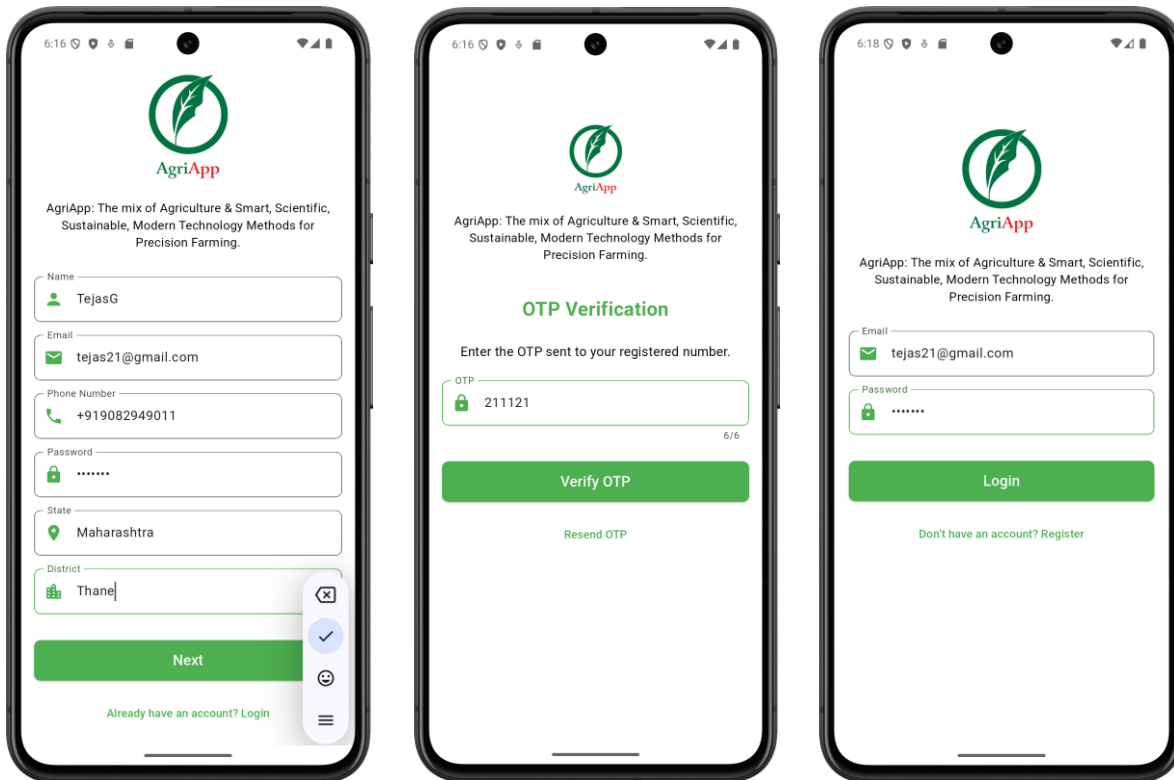
```

```

class _MainScreenState extends
State<MainScreen> {
  int _currentIndex = 0;

  final List<Widget> _pages = [
    HomePage(),
    WeatherPage(),
    DiseaseDetectionPage(),
    CropListPage(),
    MyAccountPage(),
  ];
}

```



**Authentication**

The following authentication features will stop working when Firebase Dynamic Links shuts down on 25 August 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
tejas21@gmail.com	✉	16 Feb 2025	16 Feb 2025	4hBNjovSSZ5GHAohtIU4o5VT...
vedansh21@gmail.com	✉	15 Feb 2025	15 Feb 2025	Es0JJE07aUPJFwe0pdA3Jlins...
mayank23@gmail.com	✉	15 Feb 2025	15 Feb 2025	NKU25rqEWUgNlUwbh1vRis...
+919867624148	☎	15 Feb 2025	15 Feb 2025	oBJUkwxt3iV4Y34bUAAnkeTDu...
tejas021@gmail.com	✉	15 Feb 2025	15 Feb 2025	W2FmV5AWgPtS8HJTv8Qd6...
arjun19@gmail.com	✉	14 Feb 2025	15 Feb 2025	7eXa678RarMJJccg3QZbSPg...
+919082949011	☎	14 Feb 2025	16 Feb 2025	4lJMUClyC9fQWtyC6CDJWa3...
tejasgunjal21@gmail.c...	✉	14 Feb 2025	14 Feb 2025	al3qeqKVZgZfV5qrV8GgU3tpn...

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

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

