

EXPERIMENT NO: - 06

Aim: To connect Flutter UI with Firebase database.

Theory:

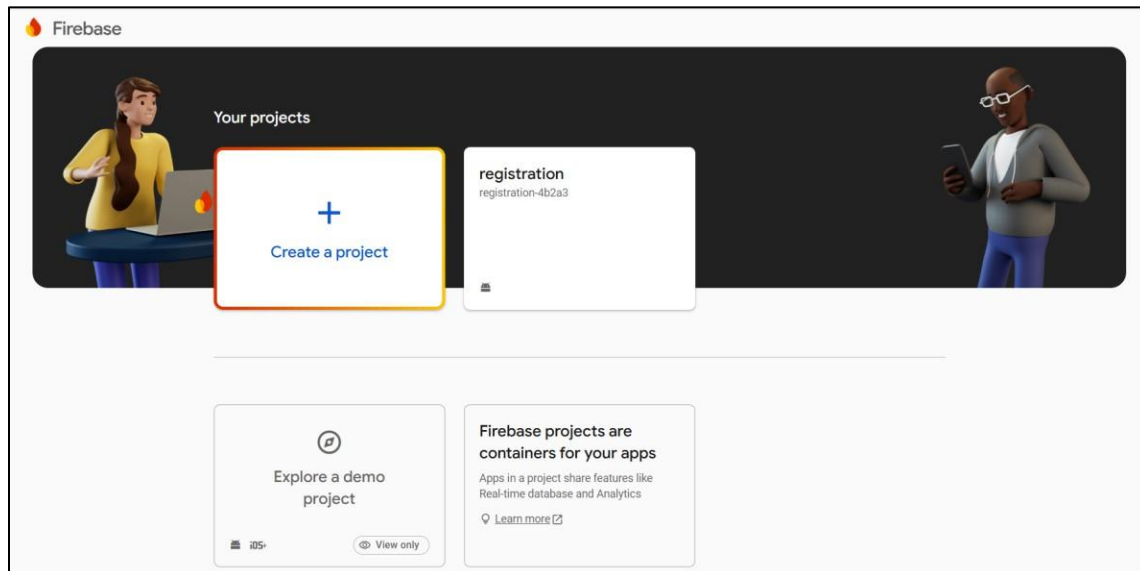
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend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

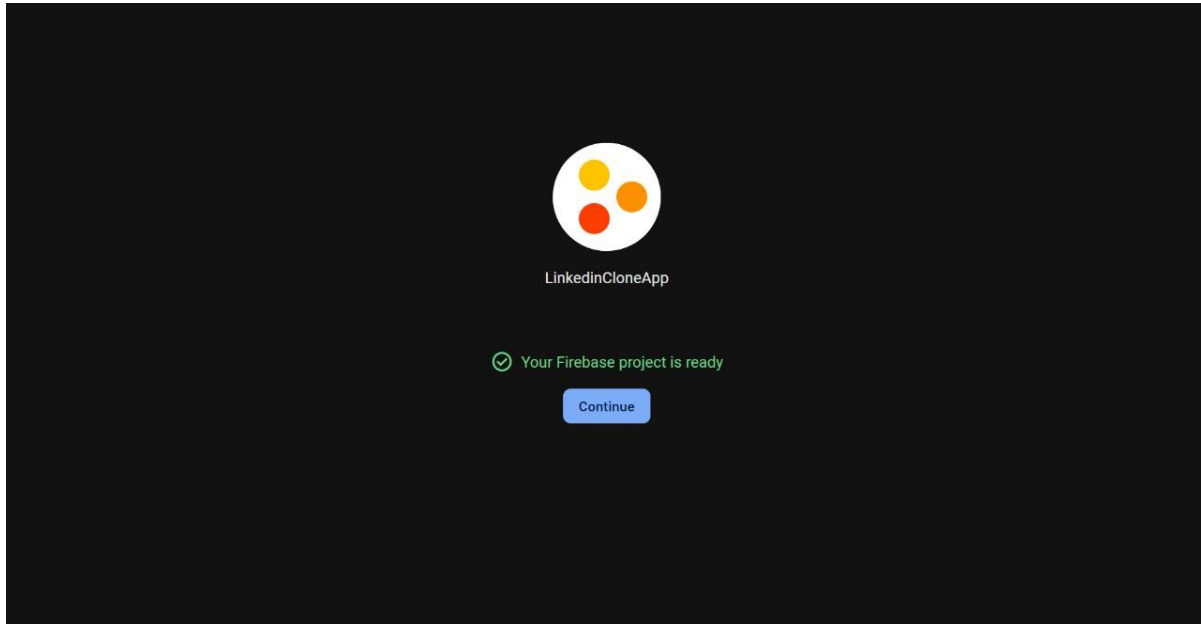
- **Steps to Connect Flutter UI with Firebase Database**

Step 1:

1.1) Go to Firebase Console and Create a Firebase Project

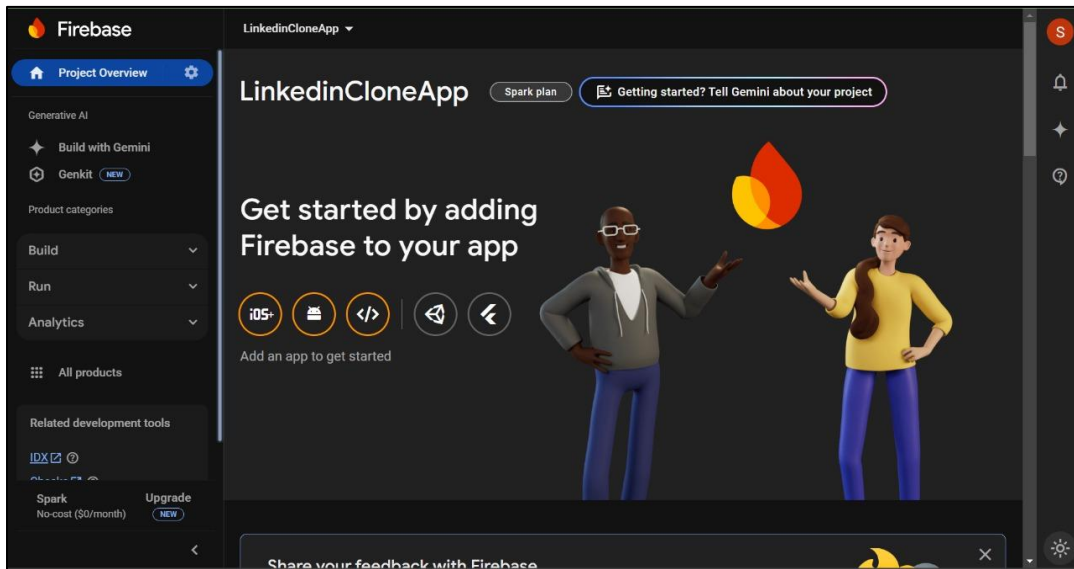


- 1.2) Click on Create a Project and give it a suitable name.
- 1.3) Enable Google Analytics (optional) & Click continue and complete the setup



Step 2:- Add Firebase to Your Flutter App

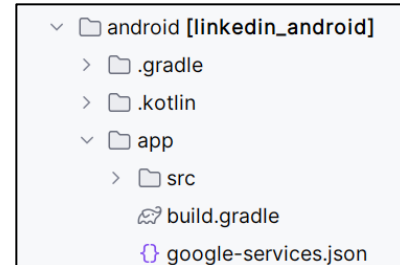
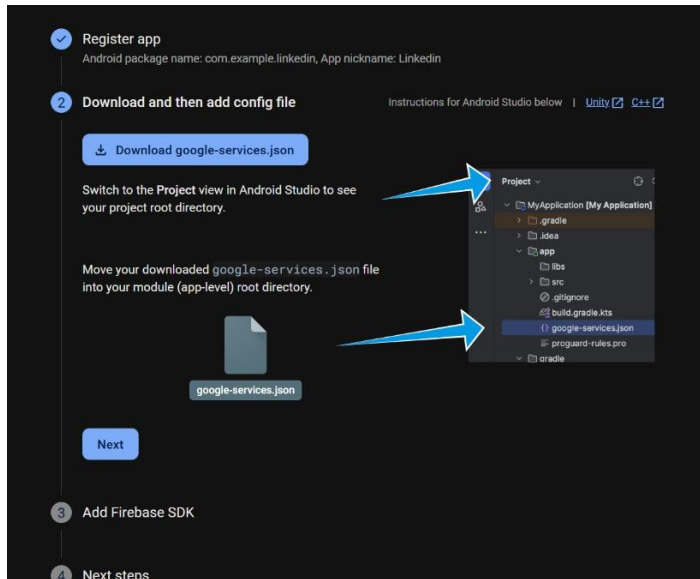
2.1) Click on Android/iOS/Web based on your Flutter application



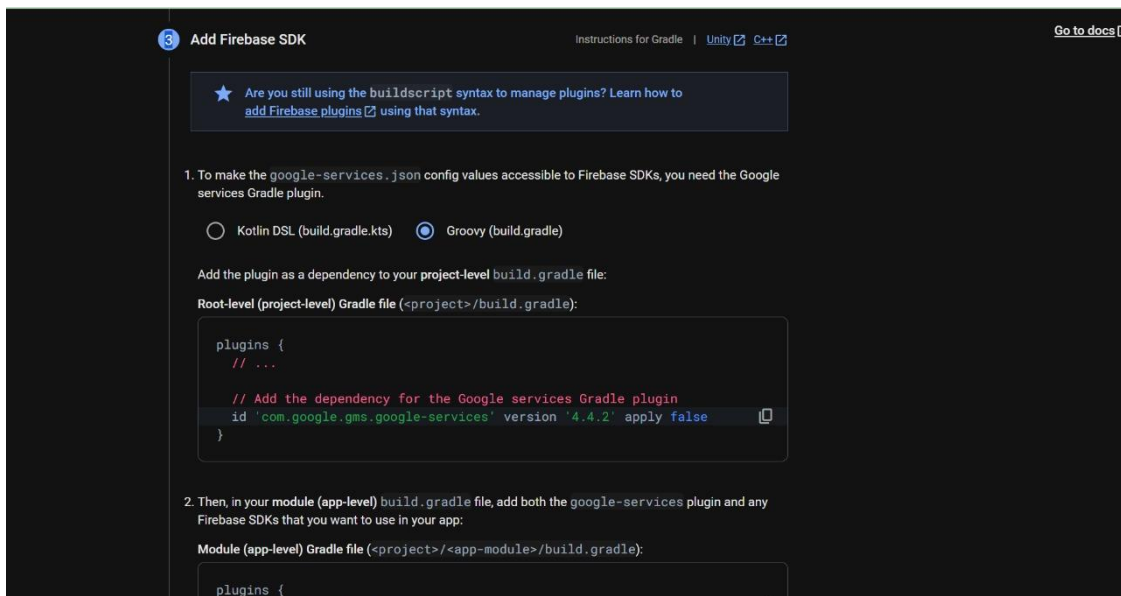
2.2) Register your app with a unique package name (found in android/app/build.gradle for Android).

The screenshot shows the 'Add Firebase to your Android app' registration form. It has a title bar with a close button and a 'Go to docs' link. The form is divided into two steps. Step 1, 'Register app', includes three input fields: 'Android package name' (with a help icon) containing 'com.example.linkedin', 'App nickname (optional)' (with a help icon) containing 'LinkedIn', and 'Debug signing certificate SHA-1 (optional)' (with a help icon) containing a long hexadecimal string. Below these fields is a note: 'Required for Dynamic Links, and Google Sign-in or phone number support in Auth. Edit SHA-1s in Settings.' and a 'Register app' button. Step 2, 'Download and then add config file', is partially visible at the bottom.

- 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



```

1  buildscript {
2      ext.kotlin_version = '1.8.0'
3
4      repositories {
5          google()
6          mavenCentral()
7      }
8      dependencies {
9          // Updated Android Gradle Plugin version
10         classpath 'com.android.tools.build:gradle:8.1.1' // Use a stable version compatible
11
12         // Updated Google Services Plugin
13         classpath 'com.google.gms:google-services:4.4.0'
14
15         // Kotlin Gradle Plugin
16         classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
17     }
18 }
19
20 allprojects {
21     repositories {
22         google()
23         mavenCentral()
24     }
25 }
26

```

```

10 android {
11     buildTypes {
12         release {
13             signingConfig signingConfigs.debug
14             minifyEnabled false
15             shrinkResources false
16         }
17     }
18 }
19
20 Flutter {
21     source = "../.."
22 }
23
24 dependencies {
25     implementation 'androidx.appcompat:appcompat:1.6.1'
26     implementation 'com.google.android.material:material:1.11.0'
27     implementation platform('com.google.firebase:firebase-bom:33.10.0')
28     implementation 'com.google.firebase:firebase-analytics'
29 }
30

```

Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies

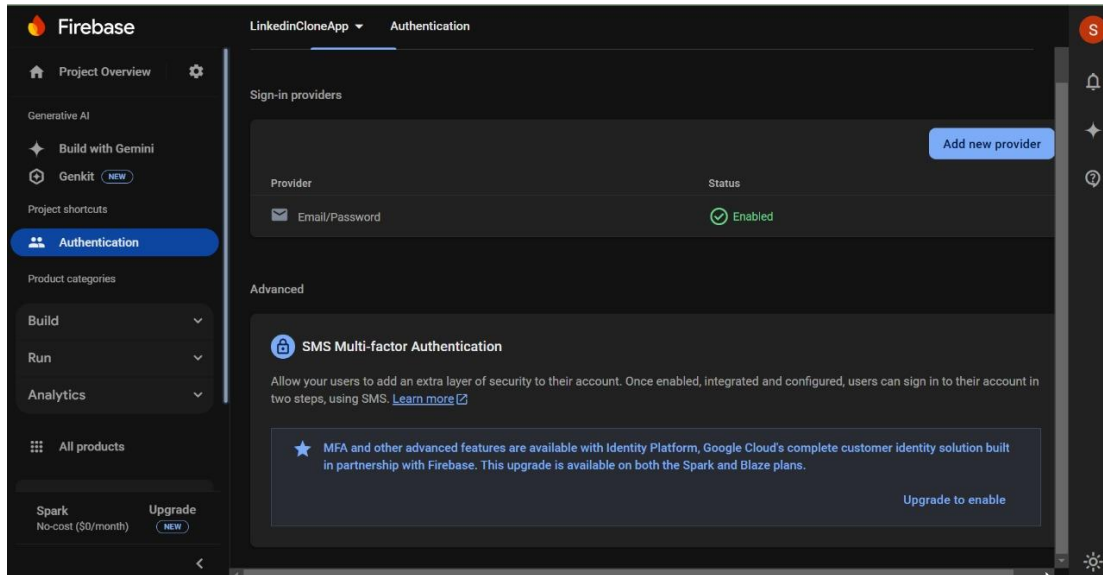
```

dependencies:
  flutter:
    sdk: flutter
  image_picker: ^1.0.7

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  google_fonts: ^6.2.1
  firebase_core: ^3.12.1
  firebase_auth: ^5.5.1
  cloud_firestore: ^5.6.5
  google_sign_in: ^6.2.2
  firebase_storage: any
  cached_network_image: any
  intl: ^0.18.0

```

- 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';
```

```
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:**1. SignIn Page**

```

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

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

  @override
  _SignInPageState createState() =>
  _SignInPageState();
}

class _SignInPageState extends
State<SignInPage> {
  final FirebaseAuth _auth =
FirebaseAuth.instance;
  final TextEditingController _emailController
= TextEditingController();
  final TextEditingController
_passwordController =
TextEditingController();
  bool _obscurePassword = true;

  void _signIn() async {
    if (_emailController.text.isEmpty ||
_passwordController.text.isEmpty) {
      _showDialog("Error", "Please fill in all
fields");
      return;
    }

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

      if (userCredential.user != null) {
        _showDialog("Success", "Sign In
Successful!");

        Navigator.of(context).pushReplacementNam
ed('/home');
      } catch (e) {
        _showDialog("Sign In Failed",
e.toString());
      }
    }
  }
}

```

```

void _showDialog(String title, String content)
{
  showDialog(
    context: context,
    builder: (context) {
      return AlertDialog(
        title: Text(title),
        content: Text(content),
        actions: [
          TextButton(
            onPressed: () =>
Navigator.pop(context),
            child: Text("OK"),
          ),
        ],
      );
    },
  );
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: Padding(
        padding: const
EdgeInsets.symmetric(horizontal: 24.0),
        child: Column(
          mainAxisAlignment:
MainAxisAlignment.center,
          children: [
            // LinkedIn Logo
            RichText(
              text: TextSpan(
                children: [
                  TextSpan(
                    text: "Linked",
                    style: TextStyle(fontSize: 32,
fontWeight: FontWeight.bold, color:
Colors.blue),
                  ),
                  WidgetSpan(
                    child: Container(
                      margin: EdgeInsets.only(left:
4),

                      padding:
EdgeInsets.symmetric(horizontal: 4, vertical:
1),

                      decoration: BoxDecoration(
                        color: Colors.blue,
                        borderRadius:
BorderRadius.circular(4),
                      ),

```



```
child: RichText(
    text: TextSpan(
        children: [
            TextSpan(
                text: "Not yet Signed in? ",
                style: TextStyle(color:
Colors.grey),
            ),
            TextSpan(
                text: "Join LinkedIn",
                style: TextStyle(fontWeight:
FontWeight.bold, color: Colors.blue),
            )
        ],
    ),
)
```

2. SignUp page:

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

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

```
@override
_SignUpPageState createState() =>
_SignUpPageState();
}
```

```
class _SignUpPageState extends
State<SignUpPage> {
  final FirebaseAuth _auth =
FirebaseAuth.instance;
  final TextEditingController _nameController
= TextEditingController();
  final TextEditingController _emailController
= TextEditingController();
  final TextEditingController
_passwordController =
TextEditingController();
  bool obscurePassword = true;
```

```
// Function to display messages
void _showMessage(String message) {
```

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text(message)),
);
}
```

```
// Sign-Up Function
```

```
void _signUp() async {
  String name = _nameController.text.trim();
  String email = _emailController.text.trim();
  String password =
    _passwordController.text.trim();
```

```
if (name.isEmpty || email.isEmpty ||
password.isEmpty) {
    _showMessage("All fields are required!");
    return;
}
```

```
if (!RegExp('^[a-zA-Z0-9._%~]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}$').hasMatch(email)) {
    _showMessage("Invalid email format!");
    return;
}
```

```
if (password.length < 6) {
    _showMessage("Password must be at least
6 characters long!");
    return;
}
```

```
try {
    UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(
    email: email,
    password: password,
);
```

```
if (userCredential.user != null) {
  _showMessage("Sign Up Successful!");
}
```

```

Navigator.of(context).pushReplacementNamed('/home');
    }
  } catch (e) {
    _showMessage("Sign Up Failed:
    ${e.toString().split(' ')[1]}");
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Center(
      child: Padding(
        padding: const
        EdgeInsets.symmetric(horizontal: 24.0),
        child: Column(
          mainAxisAlignment:
          MainAxisAlignment.center,
          crossAxisAlignment:
          CrossAxisAlignment.center,
          children: [
            // LinkedIn Logo
            Row(
              mainAxisAlignment:
              MainAxisAlignment.center,
              children: [
                Text("Linked", style:
                TextStyle(fontSize: 32, fontWeight:
                FontWeight.bold, color: Colors.blue)),
                Container(
                  color: Colors.blue,
                  padding:
                  EdgeInsets.symmetric(horizontal: 4, vertical:
                  2),
                  child: Text("in", style:
                  TextStyle(fontSize: 32, fontWeight:
                  FontWeight.bold, color: Colors.white)),
                ),
              ],
            ),
            SizedBox(height: 30),
            Text("Sign Up", style:
            TextStyle(fontSize: 24, fontWeight:
            FontWeight.bold)),
            SizedBox(height: 30),

```

```

// Name Input
TextField(
  controller: _nameController,
  decoration: InputDecoration(
    labelText: "Full Name",
    prefixIcon: Icon(Icons.person, color:
    Colors.blue),
    border:
    OutlineInputBorder(borderRadius:
    BorderRadius.circular(10)),
  ),
  ),
  SizedBox(height: 15),

// Email Input
TextField(
  controller: _emailController,
  decoration: InputDecoration(
    labelText: "Email",
    prefixIcon: Icon(Icons.email, color:
    Colors.blue),
    border:
    OutlineInputBorder(borderRadius:
    BorderRadius.circular(10)),
  ),
  ),
  SizedBox(height: 15),

// Password Input
TextField(
  controller: _passwordController,
  obscureText: _obscurePassword,
  decoration: InputDecoration(
    labelText: "Password",
    prefixIcon: Icon(Icons.lock, color:
    Colors.blue),
    suffixIcon: IconButton(
      icon: Icon(_obscurePassword ?
      Icons.visibility_off : Icons.visibility, color:
      Colors.grey),
      onPressed: () {
        setState(() {
          _obscurePassword =
          !_obscurePassword;
        });
      },
    ),

```

```

        border: OutlineInputBorder(borderRadius:
BorderRadius.circular(10)),
      ),
    ),
    SizedBox(height: 20),

    // Sign-Up Button
    SizedBox(
      width: double.infinity,
      height: 50,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.blue,
          shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(8)),
        ),
        onPressed: _signUp,
        child: Text("Sign Up", style:
TextStyle(fontSize: 18, color: Colors.white)),
      ),
    ),
    SizedBox(height: 20),

```

```

    // Sign In Link
    Row(
      mainAxisAlignment:
MainAxisAlignment.center,
      children: [
        Text("Already have an account?",
style: TextStyle(color: Colors.grey)),
        TextButton(
          onPressed: () {

Navigator.of(context).pushReplacementNamed('/signin');

        },
        child: Text("Sign In", style:
TextStyle(color: Colors.blue)),
      ],
    ),
  ],
),
),
),
);
}
}


```

Output:


Linkedin


Sign in

Email

 06sannidhi@gmail.com

Password





Forgot password?

Sign in


or

Not yet Signed in? [Join LinkedIn](#)


Linkedin

Sign Up


Full Name


 janvi

Email

 janvi29@gmail.com

Password





Sign Up

Already have an account? [Sign In](#)

LinkedinCloneApp

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 |
|----------------------|---|-------------|-------------|-----------------------------|
| abcd@gmail.com |  | Mar 4, 2025 | Mar 4, 2025 | 1hlZJyh45Zg7kDz1FsxyFok7... |
| akd@gmail.com |  | Mar 3, 2025 | Mar 3, 2025 | DjkWTW0EBfMHfNa8glUBOs... |
| 20vvk@gmail.com |  | Mar 3, 2025 | Mar 3, 2025 | AUc5E1RtE4VkdquoHij7HLDq... |
| 10roh@gmail.com |  | Mar 3, 2025 | Mar 3, 2025 | FmzLDNpcQBMGTL7o6zu3w8... |
| 06sannidhi@gmail.com |  | Mar 3, 2025 | Mar 5, 2025 | EQK9qQDePIT8xAvZJl4whQey... |

Rows per page:

50

1 – 5 of 5

<

>

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

User details get fetched from the database in the Homepage.

