

Experiment No 6

Name : Kunal Pal

Div/Roll no : D15B/50

Lab : MAD & PWA Lab

Aim : To Connect Flutter UI with fireBase database.

Firestore Setup:

- Firestore is a platform provided by Google for developing mobile and web applications. It offers various services like Firestore for databases, Firebase Authentication for user authentication, Firebase Cloud Messaging for notifications, etc.
- To integrate Firestore with your Flutter app, you need to create a Firestore project in the Firestore console and add your Flutter app to it.

Firestore Authentication Plugin:

- Firestore Authentication provides a set of APIs for handling user authentication using email/password, phone number, Google Sign-In, etc., in Flutter apps.
- By adding the firebase_auth plugin to your Flutter project, you can easily integrate Firestore Authentication functionalities into your app.
- Initializing Firestore in Flutter:
- Before using any Firestore services, you need to initialize Firestore in your
- Flutter app. This is typically done in the main() function using Firestore.initializeApp().

Authentication Operations:

- Implement authentication operations such as sign up, sign in, sign out, and getting the current user using the firebase_auth plugin in Flutter.
- These operations involve interacting with Firestore Authentication APIs to create user accounts, authenticate users, and manage user sessions.

AuthService Class:

- Create a separate class like AuthService to encapsulate authentication-related methods. This class can handle interactions with Firebase Authentication and abstract away the implementation details from your UI code.

UI Implementation:

- Develop UI screens for user authentication tasks like login, registration, and logout.
- Use forms and input fields to collect user data such as email and password for authentication purposes.

Connecting UI with Firebase Authentication:

- Call the authentication methods from your AuthService class when users interact with the UI screens.
- Handle the authentication results (success or failure) in your UI code to provide appropriate feedback to users.

Handling User Authentication State:

- Implement logic to check the user's authentication state (logged in or not logged in) when the app starts or when users navigate between screens.
- Redirect users to the appropriate screens based on their authentication status to provide a seamless user experience.

✕ Add Firebase to your Android app

✓ Register app

Android package name: com.example.test_project_2

2 Download and then add config file

Instructions for Android Studio below | [Unity](#) [C++](#)

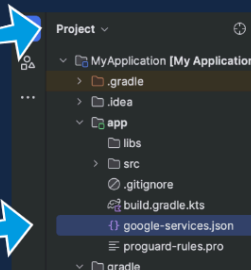
Download google-services.json

Switch to the Project view in Android Studio to see your project root directory.

Move your downloaded google-services.json file into your module (app-level) root directory.



google-services.json



Next

3 Add Firebase SDK

3 Add Firebase SDK

Instructions for Gradle | [Unity](#) [C++](#)

★ Are you still using the buildscript syntax to manage plug-ins? Learn how to [add Firebase plug-ins](#) using that syntax.

1. To make the google-services.json config values accessible to Firebase SDKs, you need the Google services Gradle plug-in.

☒ Kotlin DSL (build.gradle.kts) ☐ Groovy (build.gradle)

Add the plug-in as a dependency to your project-level build.gradle.kts file:

Root-level (project-level) Gradle file (<project>/build.gradle.kts):

```
plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.4.1" apply false
}
```

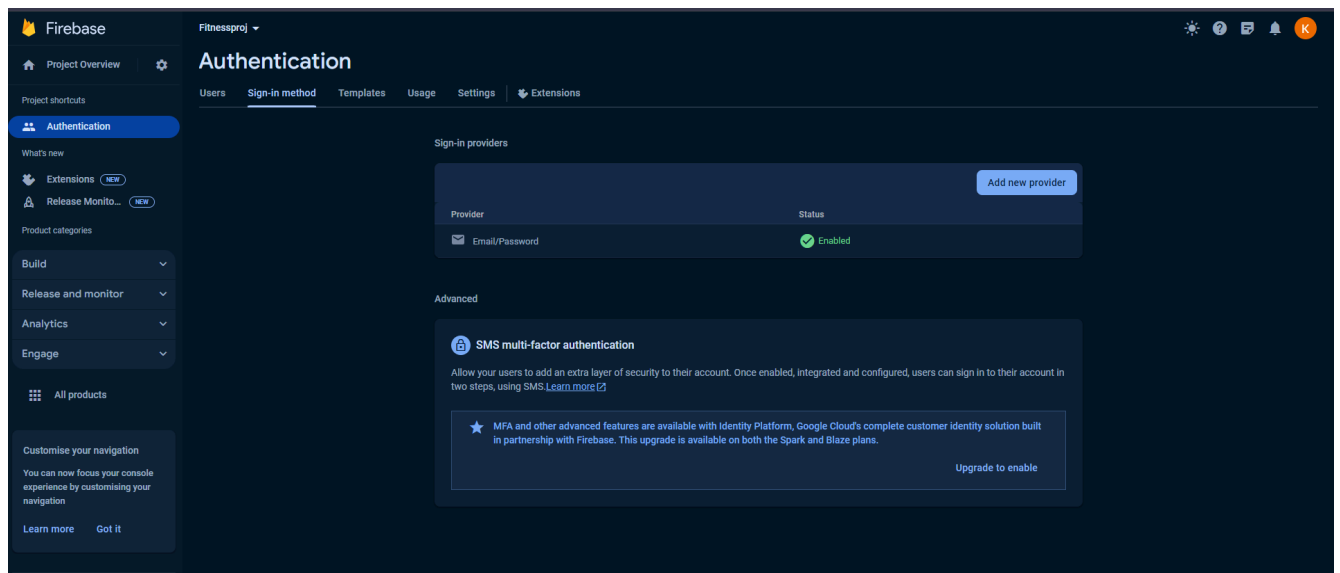
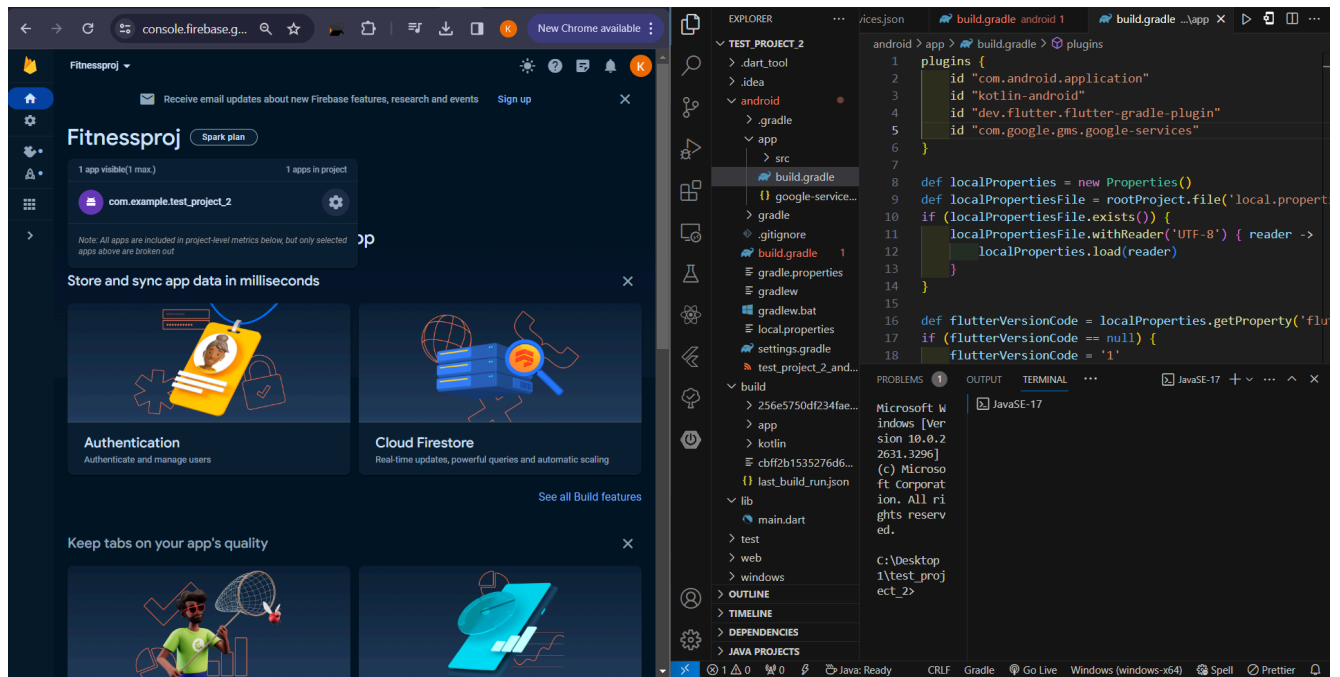
2. Then, in your module (app-level) build.gradle.kts file, add both the google-services plug-in and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (<project>/<app-module>/build.gradle.kts):

```
plugins {
    id("com.android.application")
    // Add the Google services Gradle plugin
    id("com.google.gms.google-services")
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:32.8.0"))

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    // https://firebase.google.com/docs/android/setup#available-libraries
```



Login Code :

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

class LoginPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Login'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            TextFormField(
              decoration: InputDecoration(
                labelText: 'Email',
              ),
            ),
            TextFormField(
              decoration: InputDecoration(
                labelText: 'Password',
              ),
              obscureText: true,
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                // Add your login logic here
                // For example, authenticate user and
                navigate to home screen
              },
            ),
          ],
        ),
      ),
    );
  }
}
```

```

        child: Text('Login'),
      ),
    ],
  ),
),
);
}
}

```

Registration Code:

```

import 'package:flutter/material.dart';

class RegistrationPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Registration'),
      ),
      body: Center(
        child: Column(

```

```


mainAxisAlignment: MainAxisAlignment.center,
children: <Widget>[
  TextFormField(
    decoration: InputDecoration(
      labelText: 'First Name',
    ),
  ),
  TextFormField(
    decoration: InputDecoration(
      labelText: 'Last Name',
    ),
  ),
  TextFormField(
    decoration: InputDecoration(
      labelText: 'Email',
    ),
  ),
  TextFormField(
    decoration: InputDecoration(
      labelText: 'Password',
    ),
    obscureText: true,
  ),
  SizedBox(height: 20),
  ElevatedButton(
    onPressed: () {
      // Add your registration logic here
      // For example, create user account and
navigate to login screen
    },
    child: Text('Register'),
  ),
],
),

```



```
    },  
    );  
}  
}
```

Hay there,
Create an account

 Kunal

 pal

 kunal79@gmail.com





By continuing you accept our privacy policy and
term & conditions


Register

Or



Already have an account? [Login](#)

Hay there,
Welcome back,

 kunal79@gmail



.....



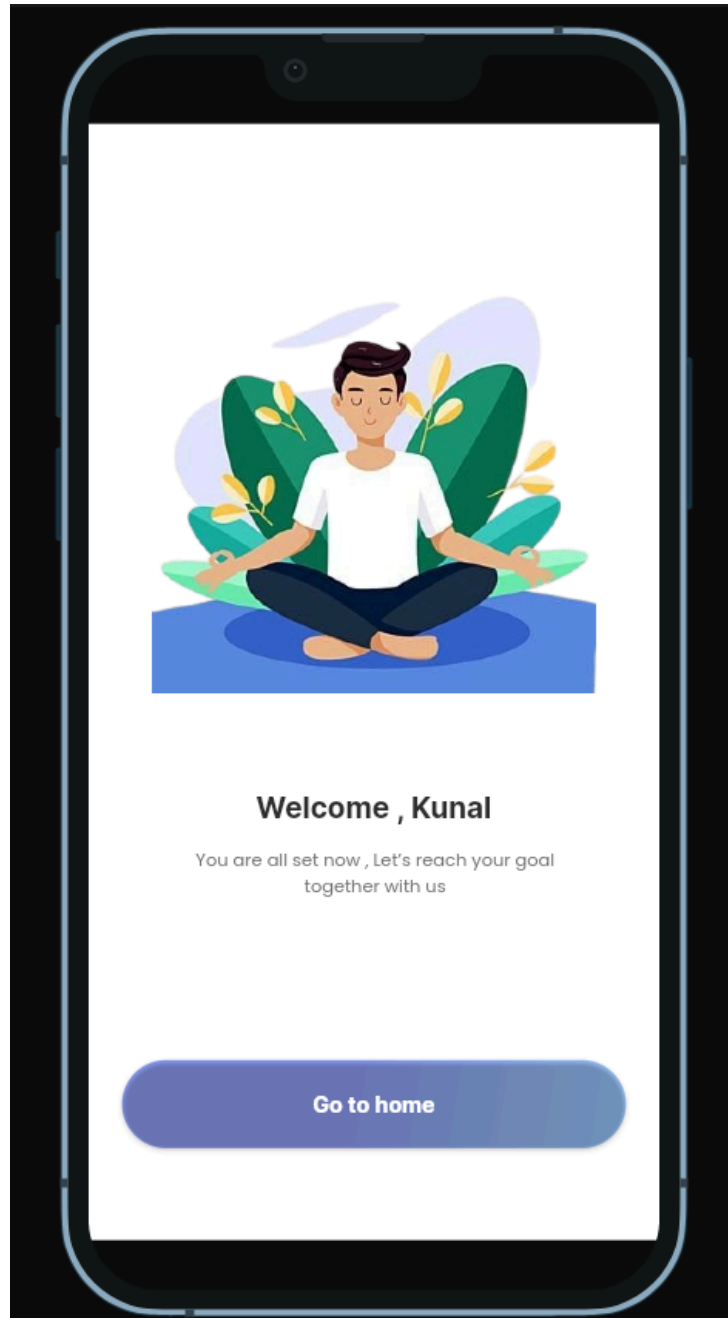
[Forgot your password ?](#)

 **Login**

Or



Don't have an account yet ? [register](#)



Conclusion : After Implementation of this I learn how to connect the firebase with our flutter using CLI method & While doing this we have Learned How the dependencies work in flutter .

