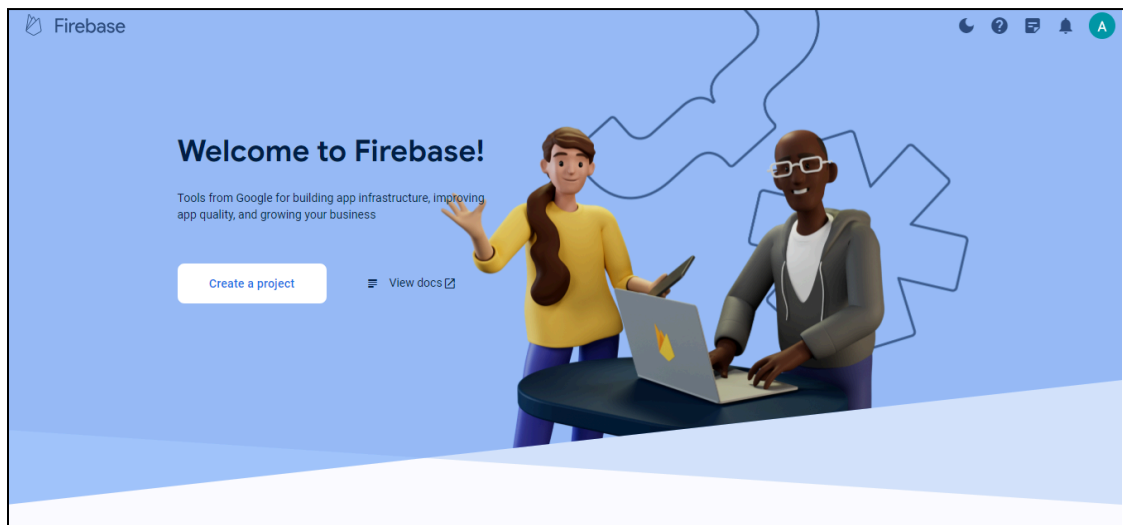


Experiment No 6

Aim: Connect Flutter UI with Firebase Database

Theory:

- Step 1: Set up Firebase Project
Create a Firebase Project and add your Flutter app to Firebase.
Enable Firebase services like Firestore or Realtime Database.



- Step 2: Set up Flutter Project

Add Firebase dependencies in pubspec.yaml.
Initialize Firebase in your Flutter app's main file.

3 Add Firebase SDKInstructions 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` file:

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

```
plugins {  
    // ...  
  
    // Add the dependency for the Google services Gradle plugin  
    id 'com.google.gms.google-services' version '4.4.1' apply false  
}
```

- Step 3: Connect Flutter UI with Firebase
 - For Firestore Database:
 - Use Firestore to read and write data.
 - For Realtime Database:
 - Use Realtime Database to interact with data in real-time.

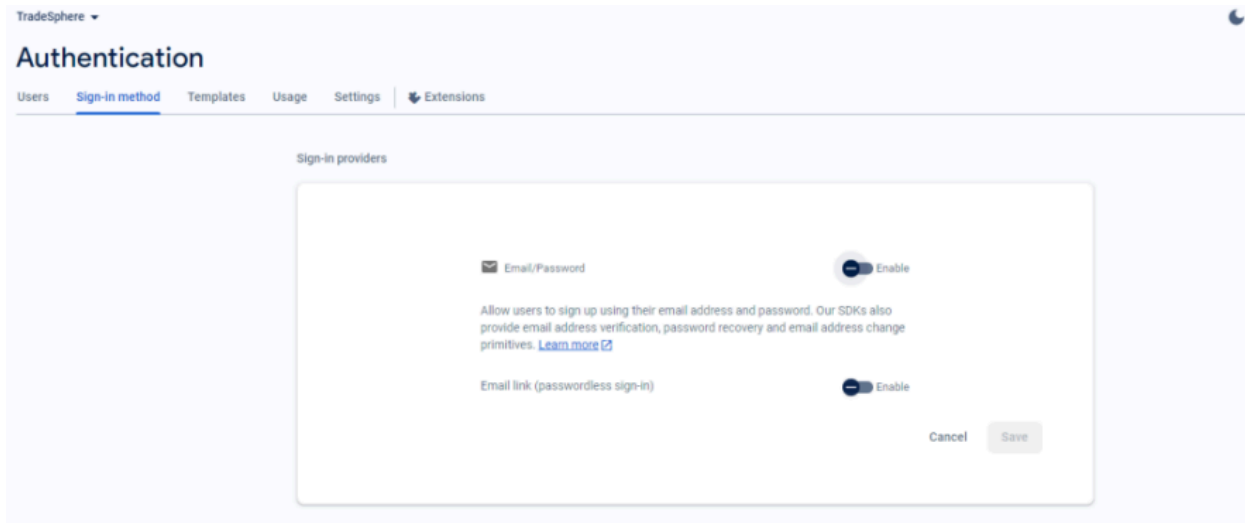
2. Then, in your **module (app-level) build.gradle** file, add both the **google-services** plug-in and any Firebase SDKs that you want to use in your app:

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

```
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:32.7.3')  
  
    // TODO: Add the dependencies for Firebase products you want to use  
    // When using the BoM, don't specify versions in Firebase dependencies  
    implementation 'com.google.firebase:firebase-analytics'  
  
    // Add the dependencies for any other desired Firebase products  
    // https://firebase.google.com/docs/android/setup#available-libraries  
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

- Step 4: Authentication
 - Integrate Firebase Authentication for user authentication.



Conclusion:

Connecting Flutter UI with a Firebase database is crucial for creating dynamic and interactive mobile applications. By understanding this connection, developers can build apps that store and retrieve data seamlessly.