Aim: To connect Flutter UI with Firebase.

Theory:

Firebase is a comprehensive platform provided by Google for developing mobile and web applications. It offers a wide range of services to help developers build high-quality apps more efficiently, including authentication, cloud storage, real-time databases, analytics, hosting, and more.

Here are some of the key Firebase services and how they are commonly used in Flutter projects:

1. Firebase Authentication: Firebase Authentication provides secure user authentication services, supporting various methods such as email/password, phone number, Google Sign-In, Facebook Login, etc. It allows you to easily manage user authentication in your Flutter app, handling user sign-up, sign-in, and managing user sessions.

2. Cloud Firestore / Realtime Database: Firebase offers two database options: Cloud Firestore and the Realtime Database. Both are NoSQL databases that allow you to store and sync data in real-time between your Flutter app and the cloud. Cloud Firestore is a more scalable and powerful option, offering advanced querying capabilities and a flexible data model. The Realtime Database provides a simpler JSON-based database structure and is well-suited for real-time data synchronization.

3. Firebase Storage: Firebase Storage is a cloud-based storage solution that allows you to store and serve user-generated content such as images, videos, and other files. It provides secure and scalable storage, making it easy to integrate file storage into your Flutter app.

4. Firebase Analytics: Firebase Analytics provides powerful tools to help you understand user behavior and app performance. It allows you to track user engagement, measure app performance metrics, and gain insights into how users interact with your Flutter app. This data can be used to optimize user experiences, improve app performance, and make data-driven decisions.

5. Firebase Hosting: Firebase Hosting is a static web hosting service that allows you to deploy and serve your Flutter web app quickly and securely. It provides a fast and reliable hosting solution with built-in SSL encryption, global CDN, and automatic scaling, making it easy to deploy and manage your Flutter web app.

These are just some of the key Firebase services commonly used in Flutter projects.

Firebase offers many more services and features that can help you build powerful and scalable apps with Flutter. It provides easy-to-use SDKs and comprehensive documentation, making it a popular choice for developers building mobile and web apps with Flutter.

Connecting Firebase to Flutter App:

To connect Firebase to a Flutter app, follow these steps:

1. Create a Firebase Project: Go to the Firebase Console (https://console.firebase.google.com/) and create a new project. Follow the setup instructions to configure your project.

2. Add Firebase to Your App: In your Flutter project, add the Firebase SDK dependencies by including the necessary plugins in your pubspec.yaml file. For example, to use Firebase Authentication, you would add the firebase_auth plugin. Run flutter pub get to install the dependencies.

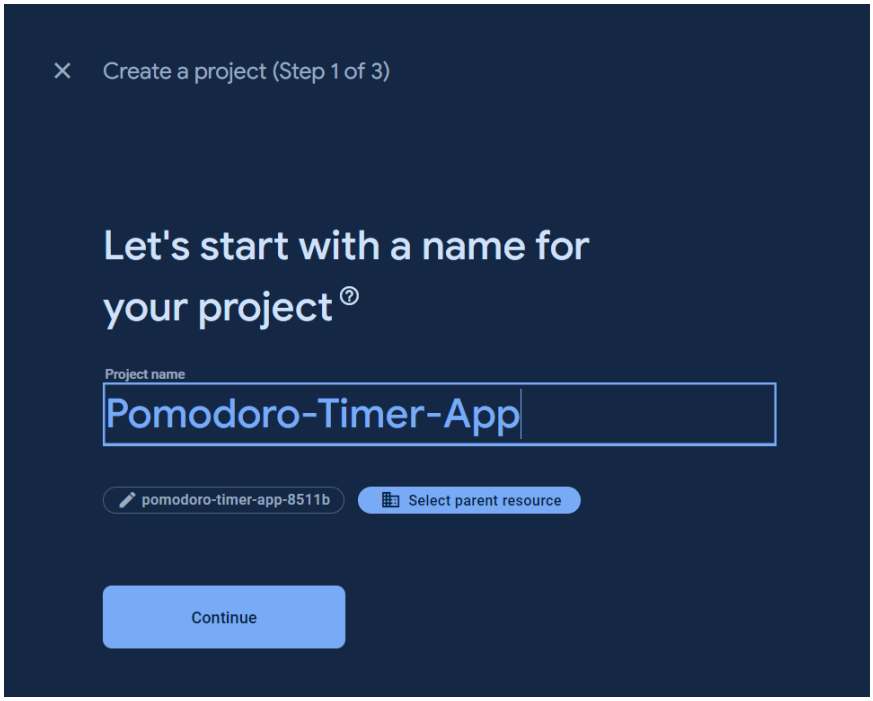
3. Initialize Firebase: In your Flutter app, initialize Firebase by calling Firebase.initializeApp() in the main() function:

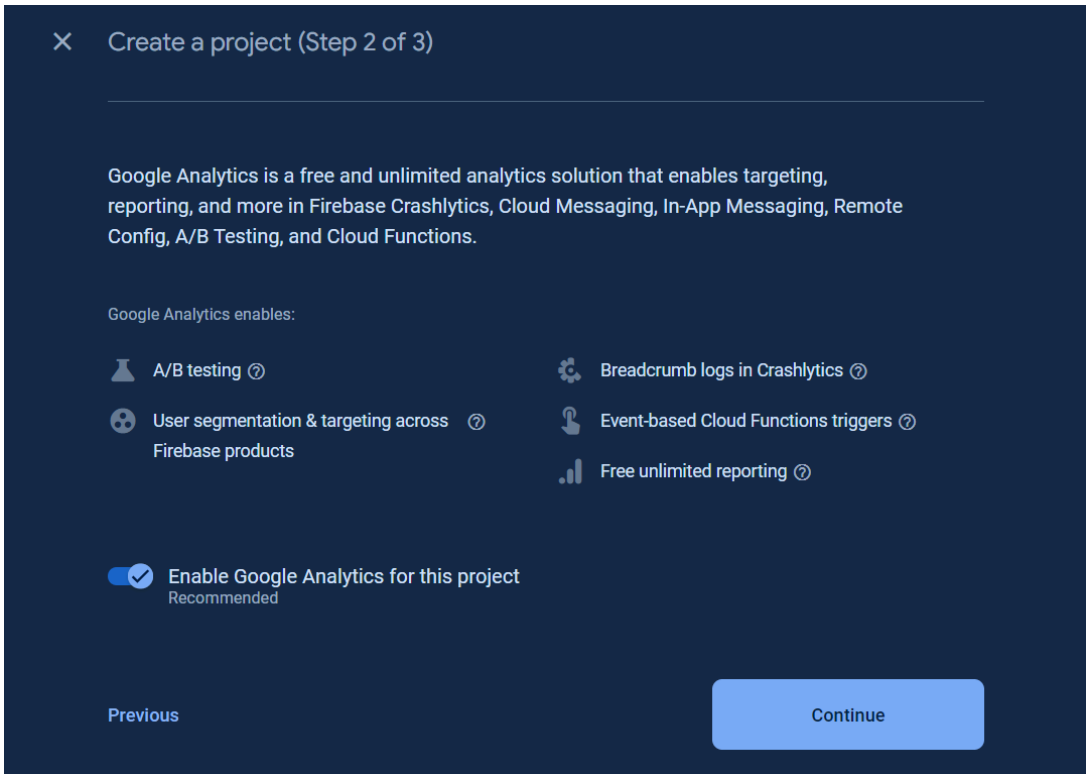
4. Use Firebase Services: Once Firebase is initialized, you can use Firebase services in your Flutter app. For example, to authenticate users with Firebase Authentication, you can use the FirebaseAuth class:

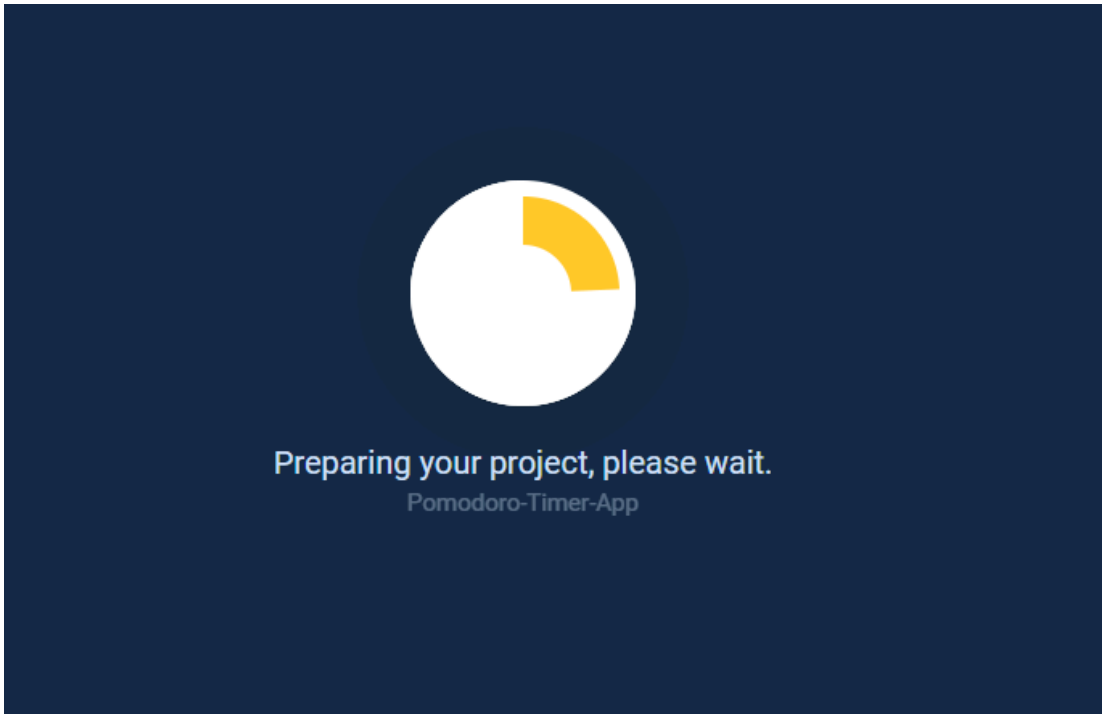
5. Configure Firebase Services: Depending on the Firebase services you're using, you may need to configure them in the Firebase Console. For example, for Firebase Authentication, you need to set up sign-in methods like email/password, Google sign-in, etc.

6. Test Your App: Test your Flutter app to ensure that Firebase integration is working correctly. You can use Firebase Emulators for local development and testing.

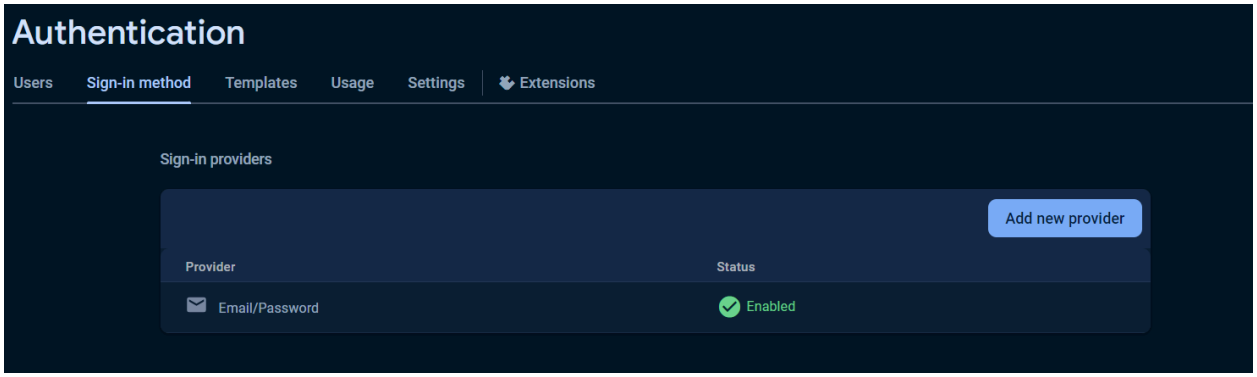
Setup firebase on google console:
Visit firebase.google.com and create a firebase project.







Under authentication enable the sign in methods that you want to add in you project

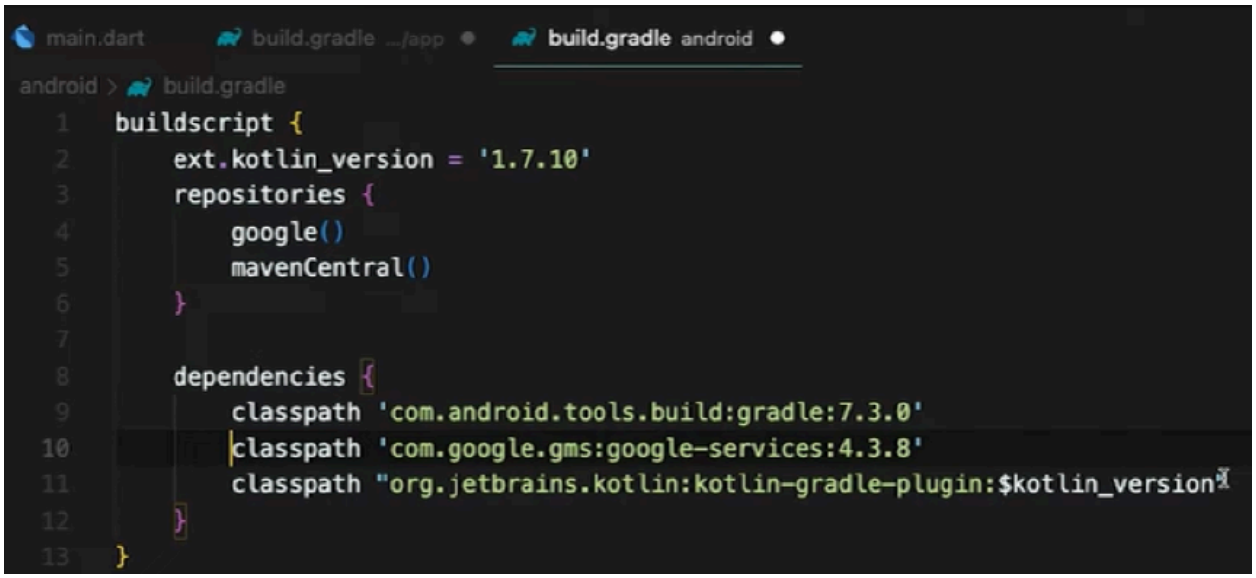


Add the google-services.json file into your src folder of the app folder

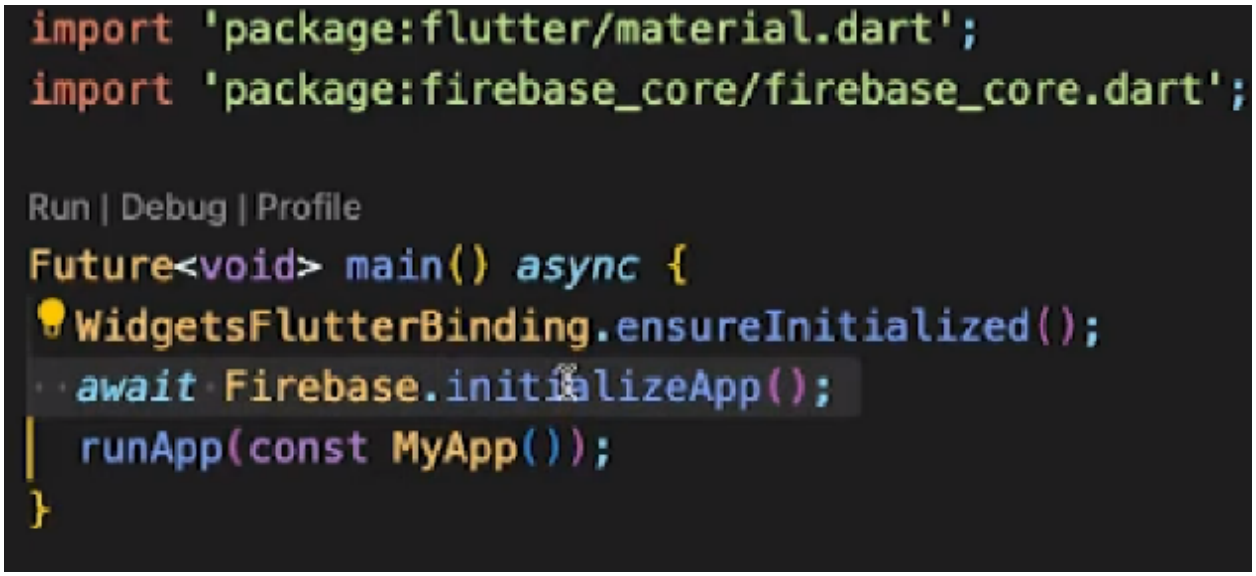
Add plugins to build.gradle file

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'
apply plugin: 'kotlin-android'
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
```

Add classpath in dependencies in build.gradle file



Add following code in main.dart file



After running the app firebase will get connected to our flutter application.

Conclusion: Thereby we understood what firebase is and how we connect it to our flutter application seamlessly. Also we learnt how to enable different authentication methods using firebase as well as how to use firestore to store user data on the cloud.