

**Aim:** To Connect Flutter UI with fireBase database

### **Theory:**

#### **What is Firebase?**

Firebase is a platform developed by Google that offers Backend-as-a-Service (BaaS) to support the development of web and mobile applications. It provides a wide range of tools and services such as Firestore (a cloud-hosted NoSQL database), Realtime Database, Firebase Authentication, Cloud Storage, Cloud Functions, Analytics, Crash Reporting, and Push Notifications. These services allow developers to focus more on frontend development while Firebase handles backend tasks like data storage, user authentication, and serverless computing.

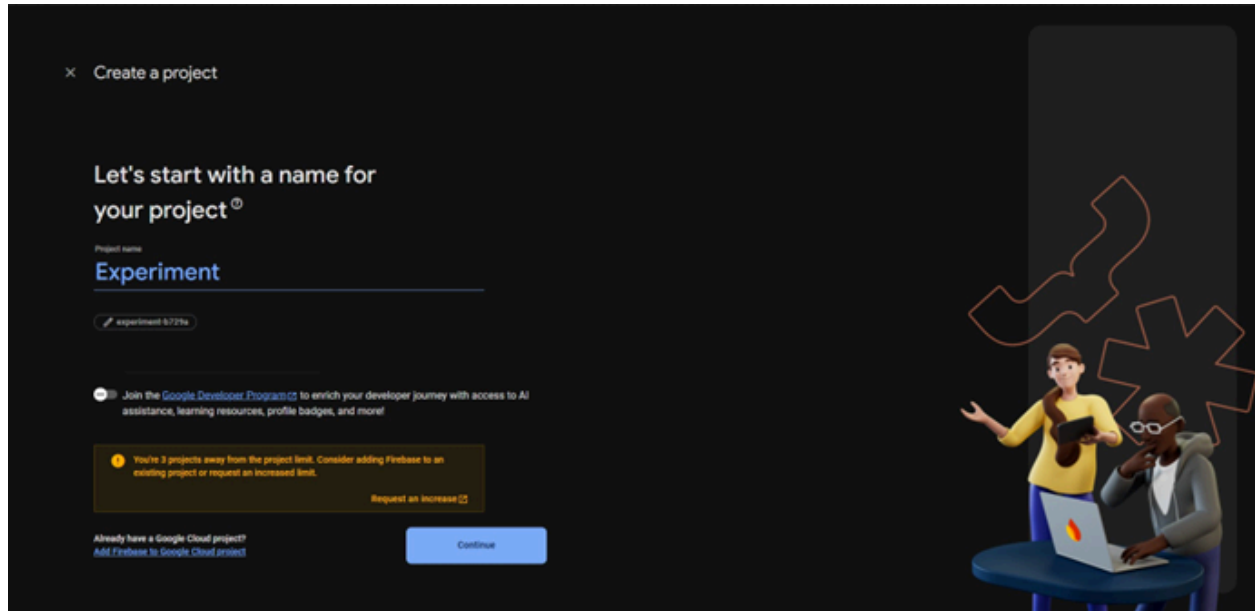
#### **Why is Firebase Used?**

Firebase is widely used in application development due to its real-time capabilities, scalability, and ease of integration. It allows developers to synchronize data across clients in real time without needing to manage their own servers. Firebase provides simple APIs and SDKs that reduce development time and complexity, especially for real-time applications like chat apps, collaborative tools, and live dashboards. Additionally, Firebase supports user authentication, secure data access, and automatic scaling, making it suitable for both small and large-scale applications.

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

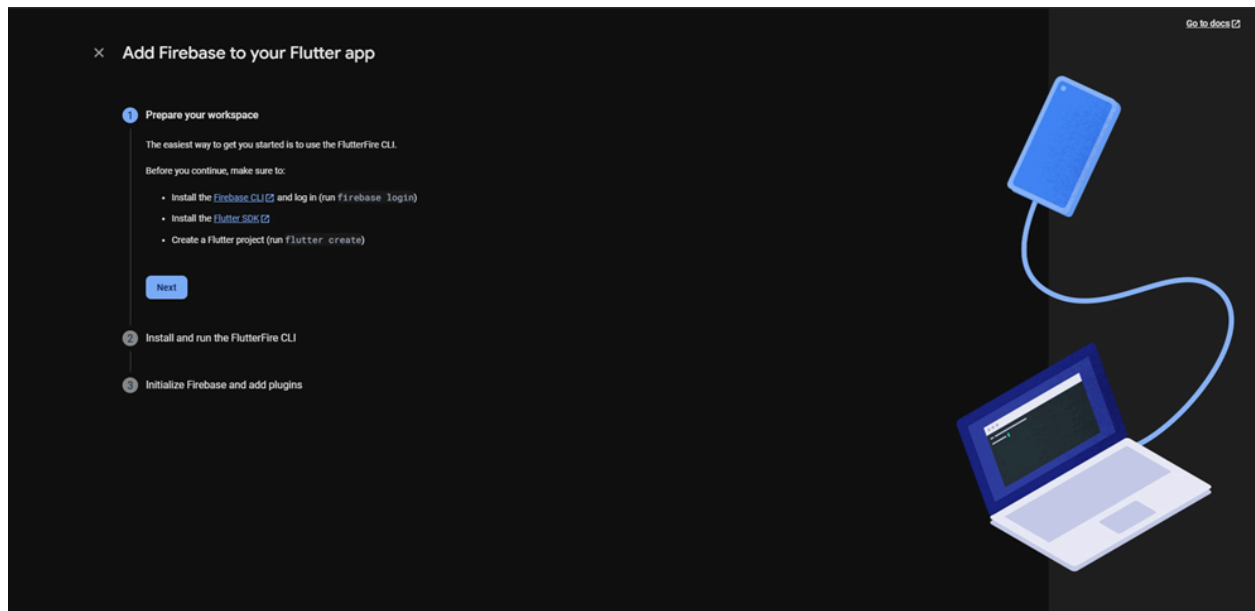
##### **1. Create a Firebase Project**

Go to the Firebase Console (<https://console.firebase.google.com/> ) and create a new project by providing a project name and enabling necessary services like Google Analytics if required.



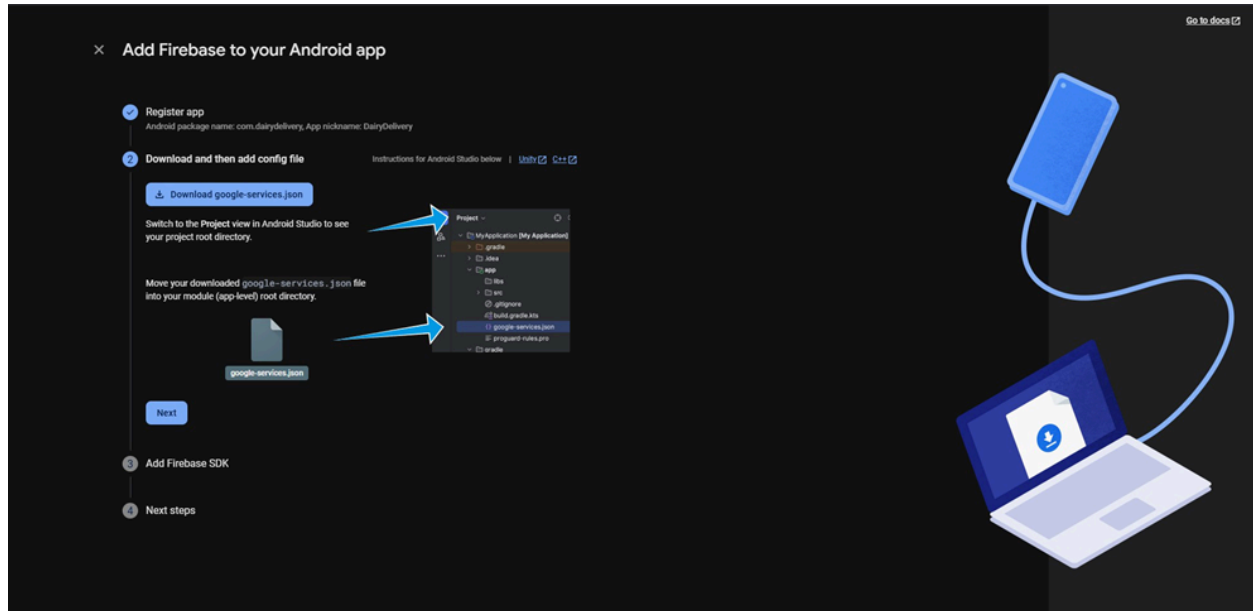
## 2. Register Your App with Firebase

In the Firebase console, add a new application by selecting the Android or iOS platform. For Android, provide the package name, nickname, and SHA-1 key (if needed).



## 3. Download and Add Configuration File

For Android, download the google-services.json file and place it in the android/app directory of your Flutter project. For iOS, download the GoogleService-Info.plist file and add it to the iOS runner project using Xcode.



#### 4. Add Firebase Dependencies

Open your pubspec.yaml file and add necessary Firebase dependencies, such as:  
dependencies:

firebase\_core: latest\_version

cloud\_firestore: latest\_version

firebase\_auth: latest\_version

Then run the flutter pub get to fetch the packages.

#### 5. Configure Firebase in Your App

Import and initialize Firebase in your Flutter app. In the main.dart file, ensure Firebase is initialized before the app runs:

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

```
}
```

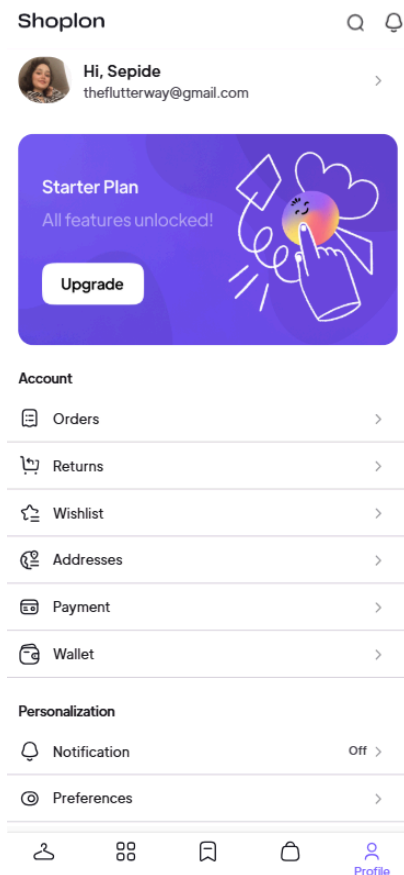
## 6. Use Firebase Services in the UI

After initializing Firebase, you can use its services within your UI. For example, you can use FirebaseAuth for authentication and Firestore to fetch or write data in the app interface.

## 7. Update Android and iOS Configuration

For Android, make sure to apply the Google Services plugin in android/build.gradle and android/app/build.gradle. For iOS, ensure the platform version is at least 10.0 and required permissions are configured in Info.plist.

### Output:



**Conclusion:** By successfully connecting the Flutter UI with the Firebase database, you have learned how to create dynamic and data-driven applications. This integration allows real-time data synchronization between the app and the backend, enabling

functionalities such as data storage, retrieval, and updates. Understanding how to connect Flutter with Firebase is essential for developing modern applications that require persistent data, user management, and seamless cloud connectivity.