Name - Gaurang A Raorane Div - D15A Roll no - 49 Batch - C

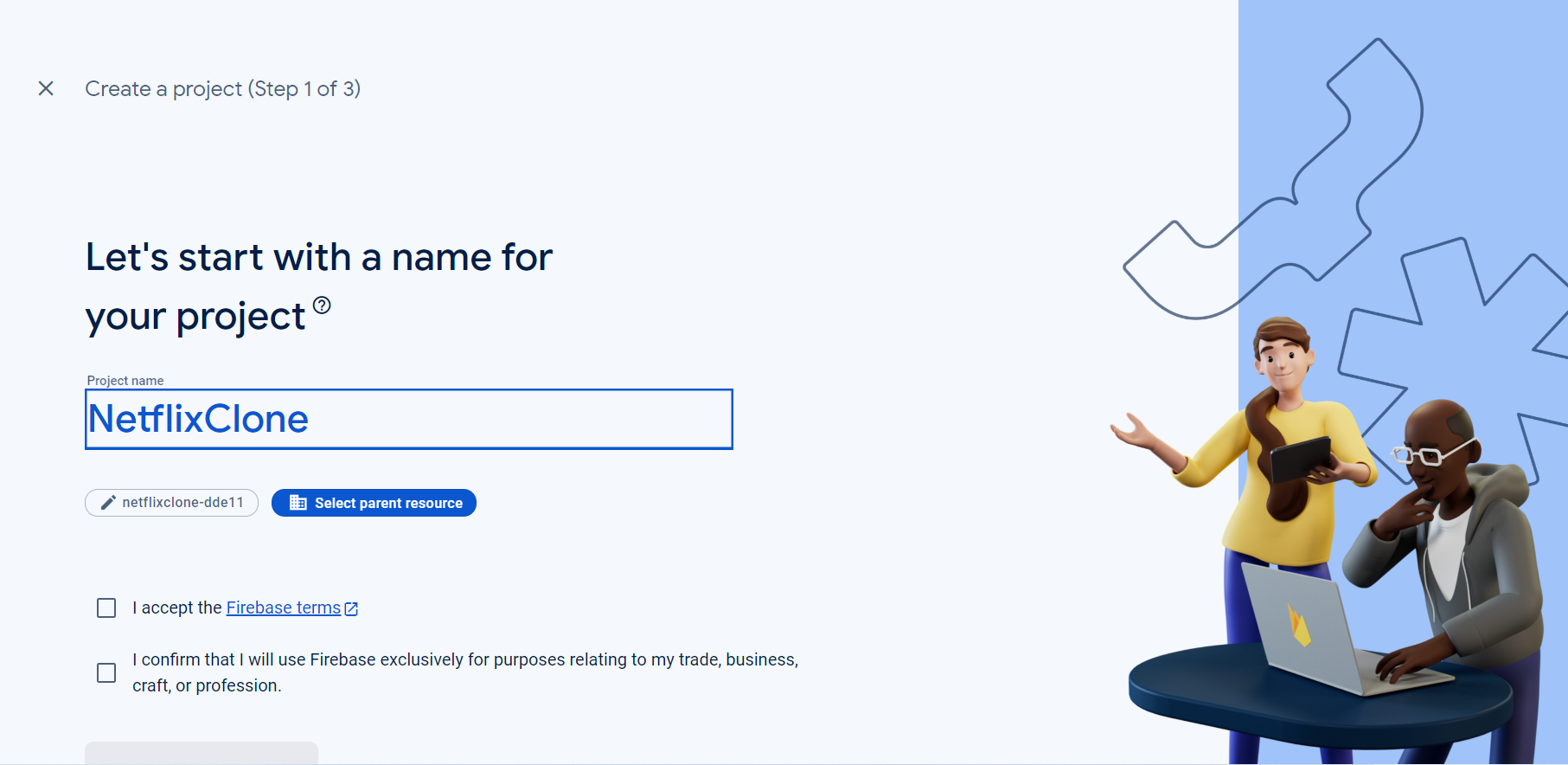
**Experiment - 6**

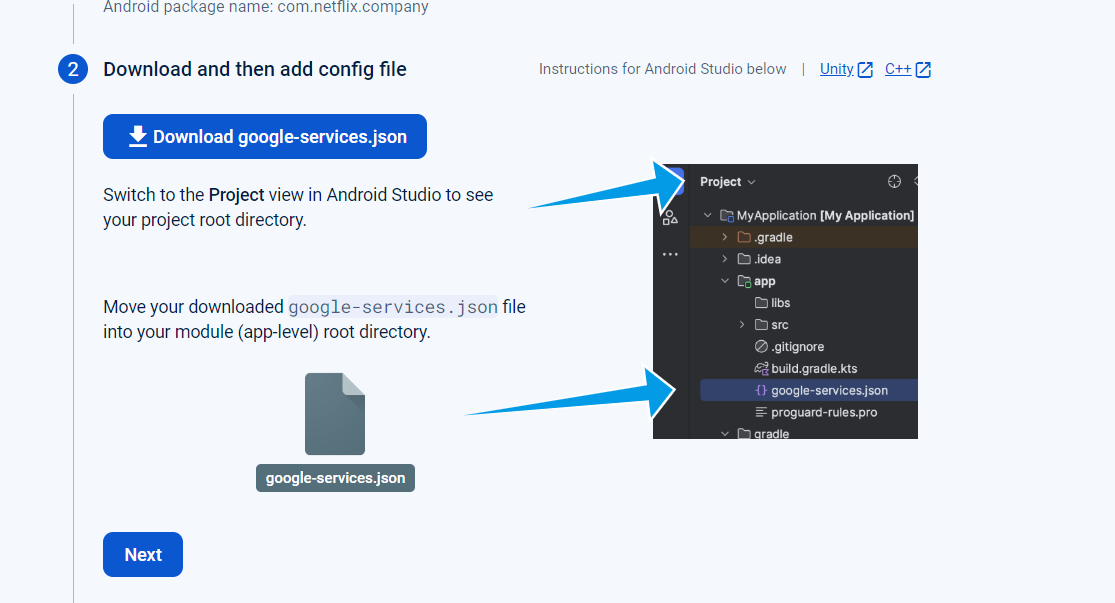
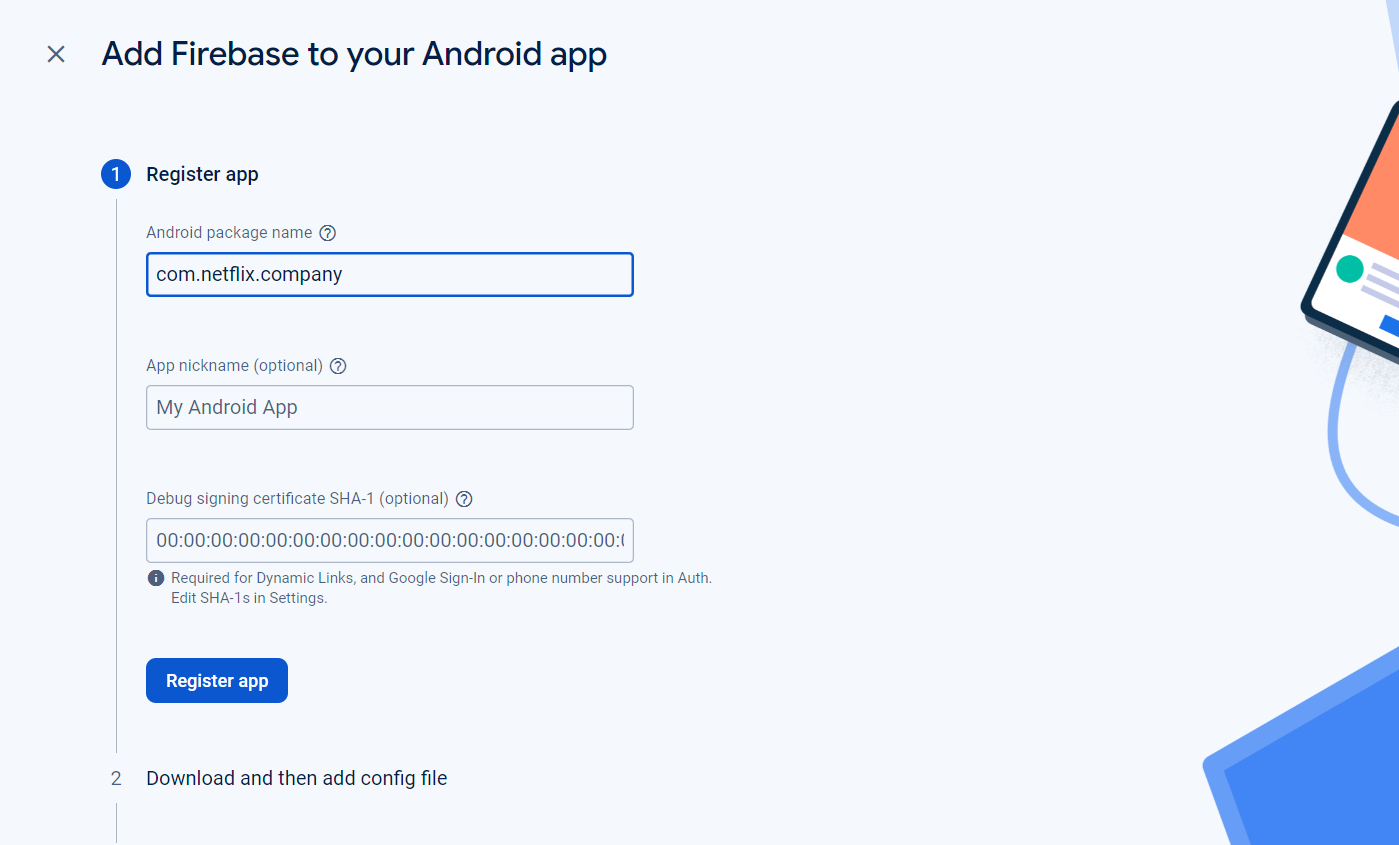
**Aim:-** To Connect Flutter UI with fireBase database

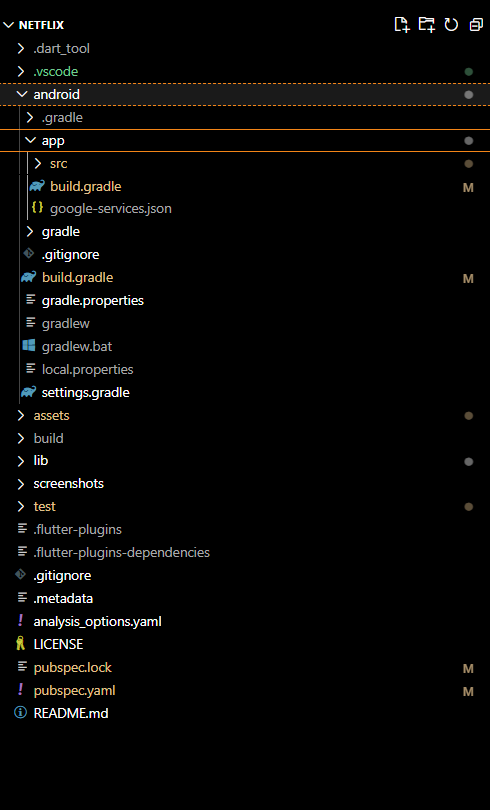
**Theory:-**

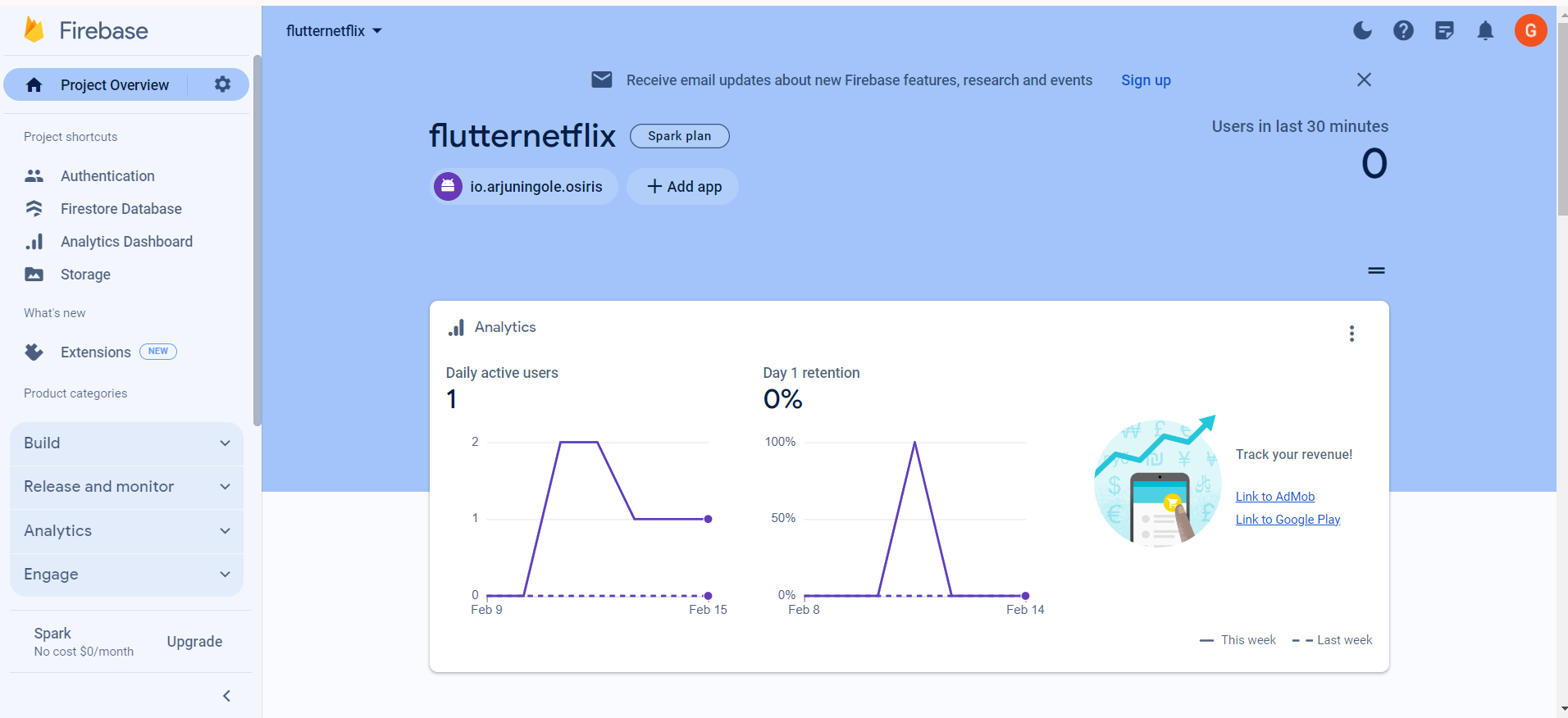
Firebase is a comprehensive platform provided by Google for building mobile and web applications. It offers various services, including a real-time database, authentication, cloud storage, and more. Connecting a Flutter UI with a Firebase database enables developers to store, retrieve, and synchronize data in real-time, providing a seamless experience for users. Let's explore the theory behind connecting Flutter UI with Firebase Database:

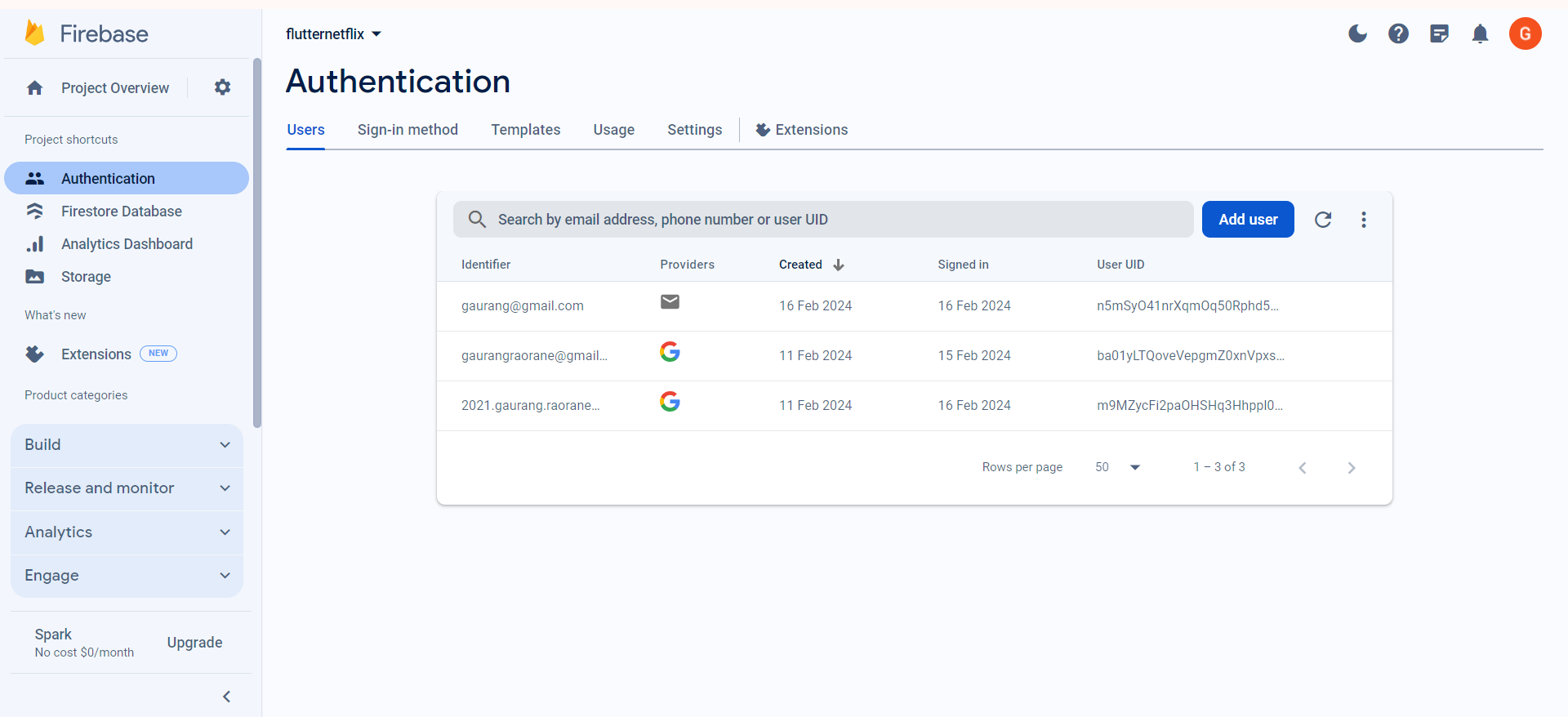
* **Firebase Realtime Database:**
  + Firebase Realtime Database is a cloud-hosted NoSQL database that allows developers to store and sync data between users in real-time.
  + It provides a JSON-based data model, making it easy to organize and structure data.
  + The database automatically synchronizes data across all connected clients, ensuring that changes made by one client are immediately reflected on other clients.
* **Firebase SDK for Flutter:**
  + Google provides the Firebase Flutter plugin, which allows Flutter apps to integrate seamlessly with Firebase services.
  + The Firebase plugin for Flutter provides APIs to interact with Firebase services, including authentication, database, cloud storage, etc.
  + It simplifies the process of connecting Flutter apps to Firebase services, enabling developers to focus on building the app's functionality.
* **Integration Process:**
  + Step 1: Set up Firebase Project:
    - Create a Firebase project from the Firebase Console (https://console.firebase.google.com).
    - Add your Flutter app to the Firebase project by registering its package name.
    - Download the google-services.json configuration file and place it in the android/app directory of your Flutter project.
  + Step 2: Configure Flutter Project:
    - Add the Firebase Flutter plugin to your Flutter project by adding the necessary dependencies in the pubspec.yaml file.
    - Initialize Firebase in your Flutter app by calling Firebase.initializeApp() in the main() function.
  + Step 3: Access Firebase Database:
    - Use the Firebase Database SDK for Flutter to interact with the Firebase Realtime Database.
    - Create references to database locations using the FirebaseDatabase class.
    - Perform CRUD (Create, Read, Update, Delete) operations on the database using methods like set(), push(), update(), remove(), etc.
  + Step 4: Display Data in Flutter UI:
    - Retrieve data from the Firebase database and display it in the Flutter UI.
    - Use Flutter widgets like ListView, GridView, Text, Image, etc., to render database content in the app's UI.
    - Implement real-time data synchronization to automatically update the UI when changes occur in the database.

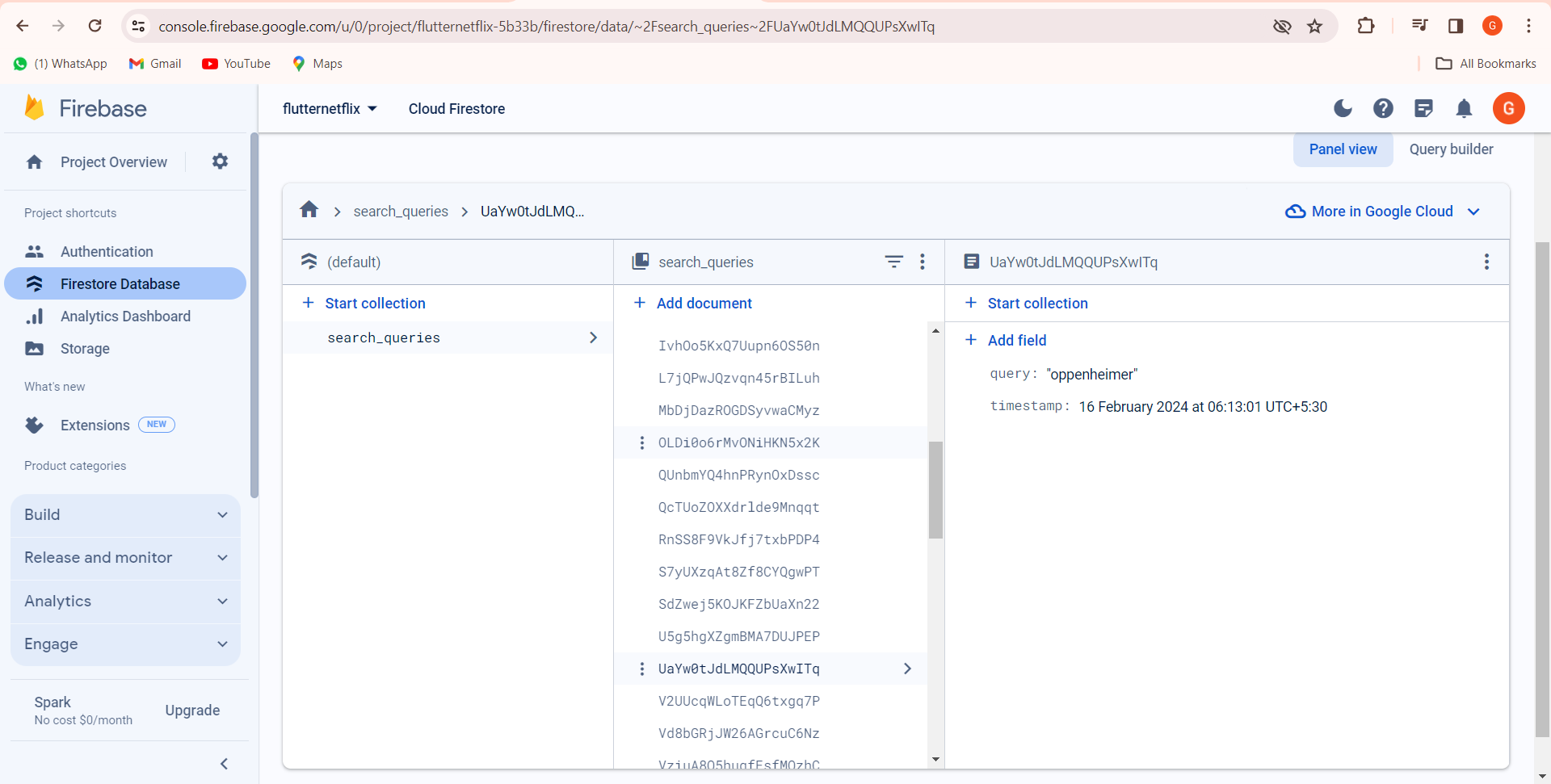












Hence Firebase is Successfully Connected.