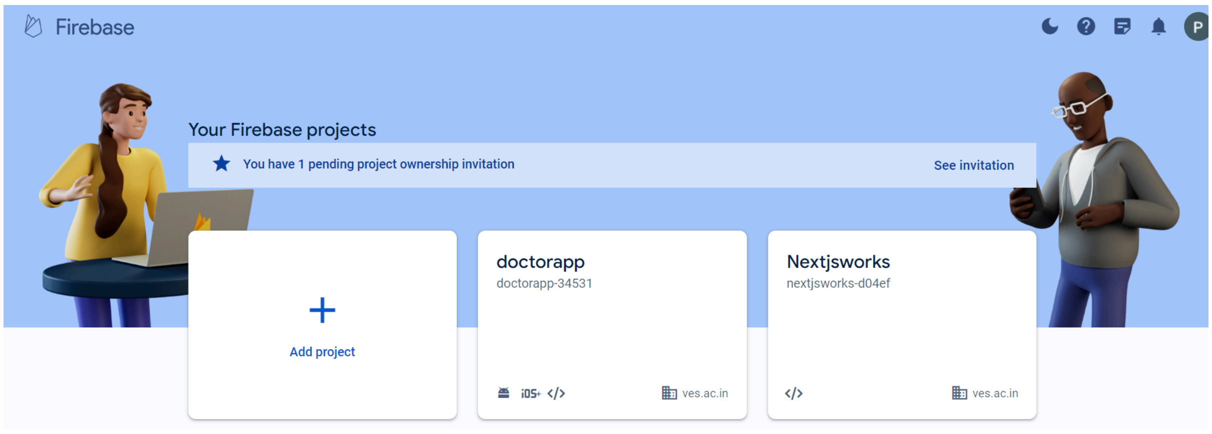
Name : Pooja Panjwani **Exp : 6** D15B_51

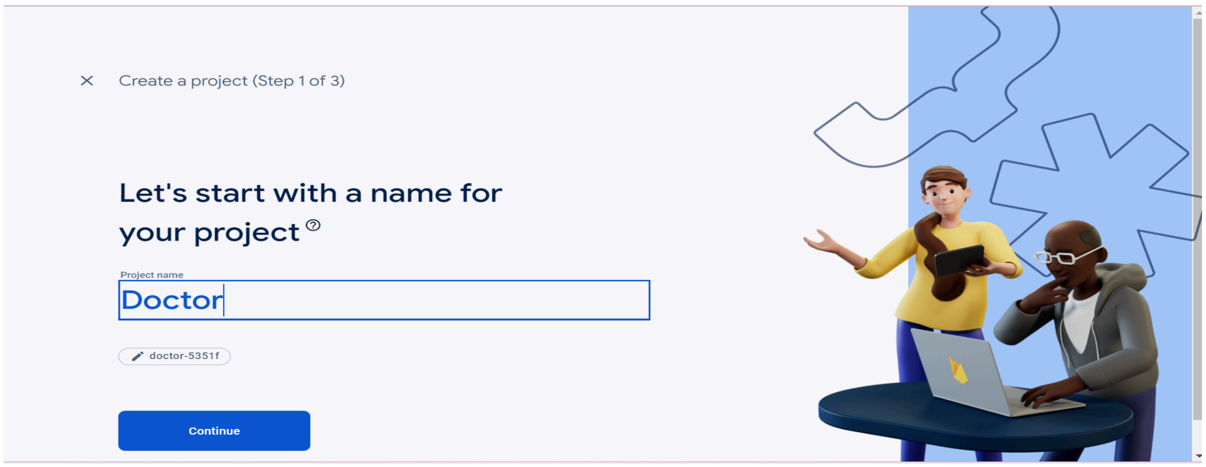
AIM : To connect Flutter app UI with Firebase database

THEORY : To integrate Flutter with Firebase, begin by setting up a Firebase project, linking your Flutter app, and configuring dependencies in the `pubspec.yaml` file. Initialize Firebase in your app's entry point and consider adding the optional `firebase_auth` package for authentication. For database operations, utilise the `cloud_firestore` package to interact with Firestore, performing CRUD operations and enabling real-time data updates. If your app involves file storage, integrate Firebase Storage using the `firebase_storage` package. Implement robust error-handling mechanisms for asynchronous Firebase operations. Thoroughly test and debug your app, ensuring seamless functionality on both iOS and Android platforms. Once satisfied, deploy your Flutter app, now connected to Firebase, to make the most of the platform's features for authentication, real-time databases, and storage.

Connecting a Flutter app UI with a Firebase database involves several steps. Firebase is a mobile and web application development platform that provides various services, including a real-time NoSQL database.

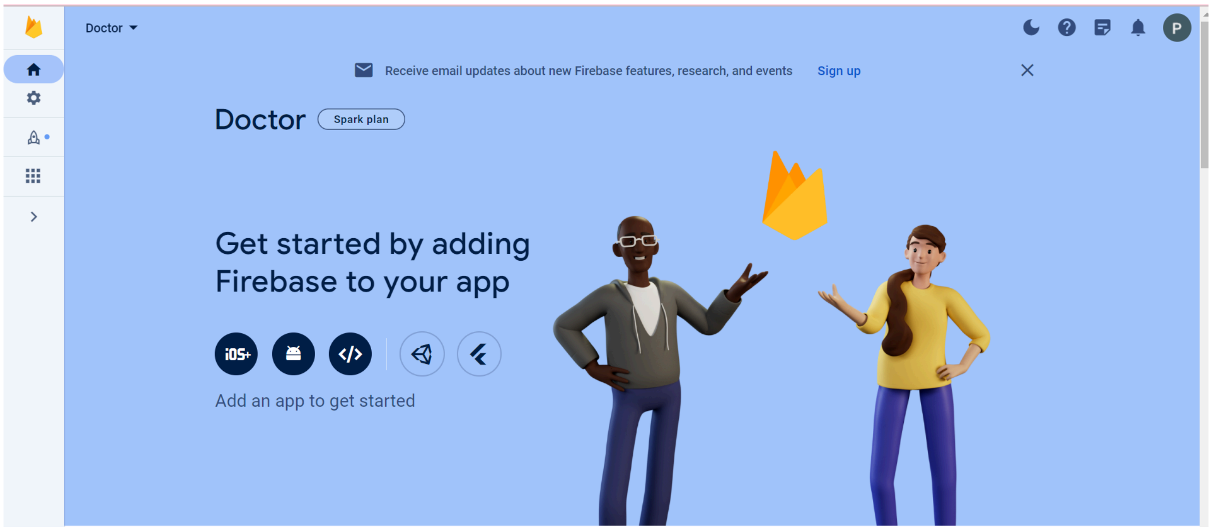
1. **Create a Firebase Project:**
-Go to the [Firebase Console](https://console.firebase.google.com/).
- Click on "Add Project" and follow the setup instructions.



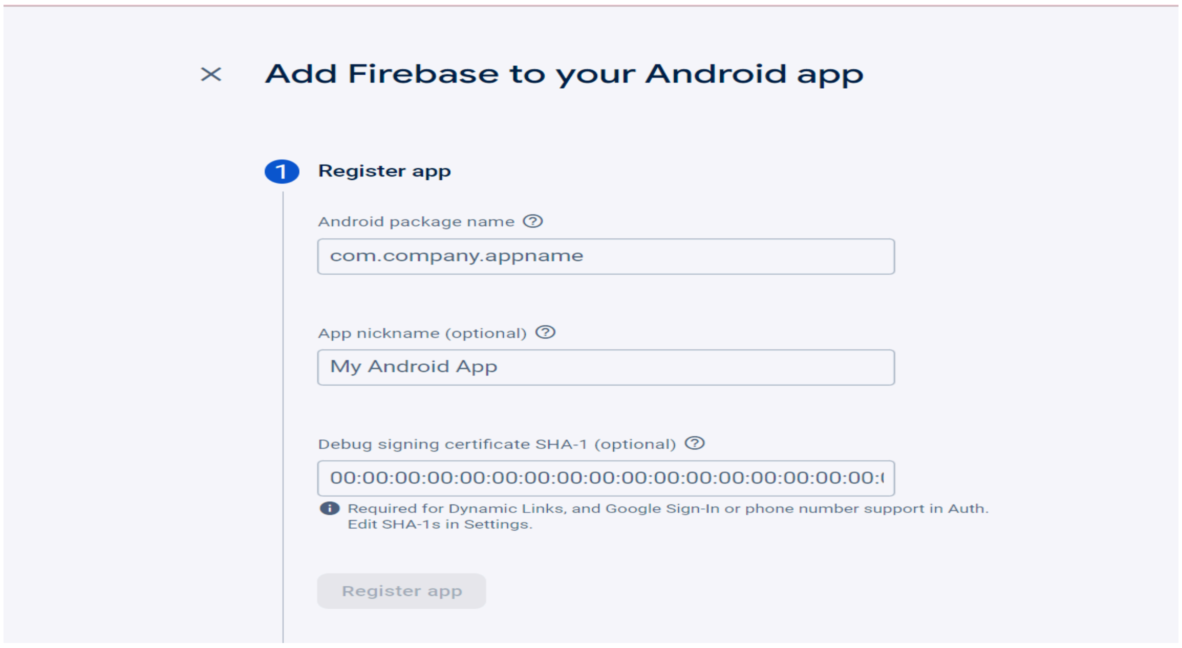

Create a project (Step 1 of 3)
Let's start with a name for your project
Project name
doctor-5351f
Continue

Project name: Doctor

2. **Add a Flutter App to Firebase Project:**
- In the Firebase Console, select your project.
- Click on "Add app" and choose the Flutter icon. - Follow the setup instructions to register your app


Doctor
Receive email updates about new Firebase features, research, and events
Sign up
Doctor
Spark plan
Get started by adding Firebase to your app
iOS+
Add an app to get started

- Can go with android app:


Add Firebase to your Android app

1 Register app

Android package name
com.company.appname

App nickname (optional)
My Android App

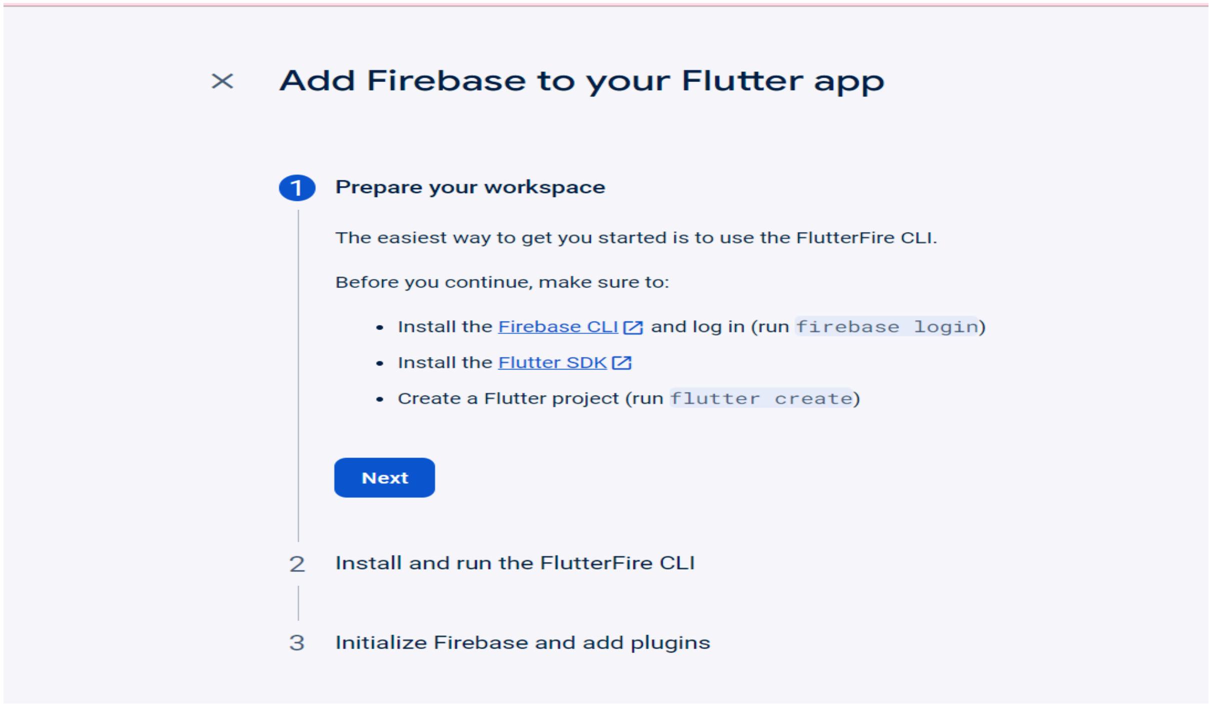
Debug signing certificate SHA-1 (optional)
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:

Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

OR

- Can go with Flutter CLI



× Add Firebase to your Flutter app

1 Prepare your workspace

The easiest way to get you started is to use the FlutterFire CLI.

Before you continue, make sure to:

- Install the Firebase CLI and log in (run `firebase login`)
- Install the Flutter SDK
- Create a Flutter project (run `flutter create`)

2 Install and run the FlutterFire CLI

3 Initialize Firebase and add plugins

Dependencies:
Flutter:
sdk: flutter
firebase_core: ^latest_version
firebase_database: ^latest_version
- Run flutter pub get to install the dependencies.

```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2
  velocity_x: ^3.6.0
  get: ^4.6.5
  firebase_core: ^2.19.0
  firebase_auth: ^4.11.1
  cloud_firestore: ^4.15.6
```

4. **Initialize Firebase in Flutter:**
- import the Firebase packages in your main.dart file:dart
  import 'package:firebase_core/firebase_core.dart';
- Initialize Firebase in the main function: dart

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

5. **Set Up Firebase Realtime Database:**
- In the Firebase Console, go to "Database" and click on "Create Database."
- Choose "Start in test mode" and click "Next."
 - Set up your database rules.

6. **Create Firebase Database Reference:**
- Import the necessary package in your Dart file:dart
   import 'package:firebase_database/firebase_database.dart';
- Create a reference to your Firebase database:dart final databaseReference = FirebaseDatabase.instance.reference();

 7. **Read Data from Firebase:**
- To read data from Firebase, you can use methods like once() or onValue() :
Dart
DatabaseReference reference = FirebaseDatabase.instance.reference();
// Read data once
DataSnapshot snapshot = await reference.once();
 // Access the data
print('Data: ${snapshot.value}');

8. **Write Data to Firebase:**

- To write data to Firebase, you can use methods like set() or push() :
Dart
DatabaseReference reference = FirebaseDatabase.instance.reference();
 // Set data
reference.child('users').set({'username': 'JohnDoe', 'email': 'john@example.com'});

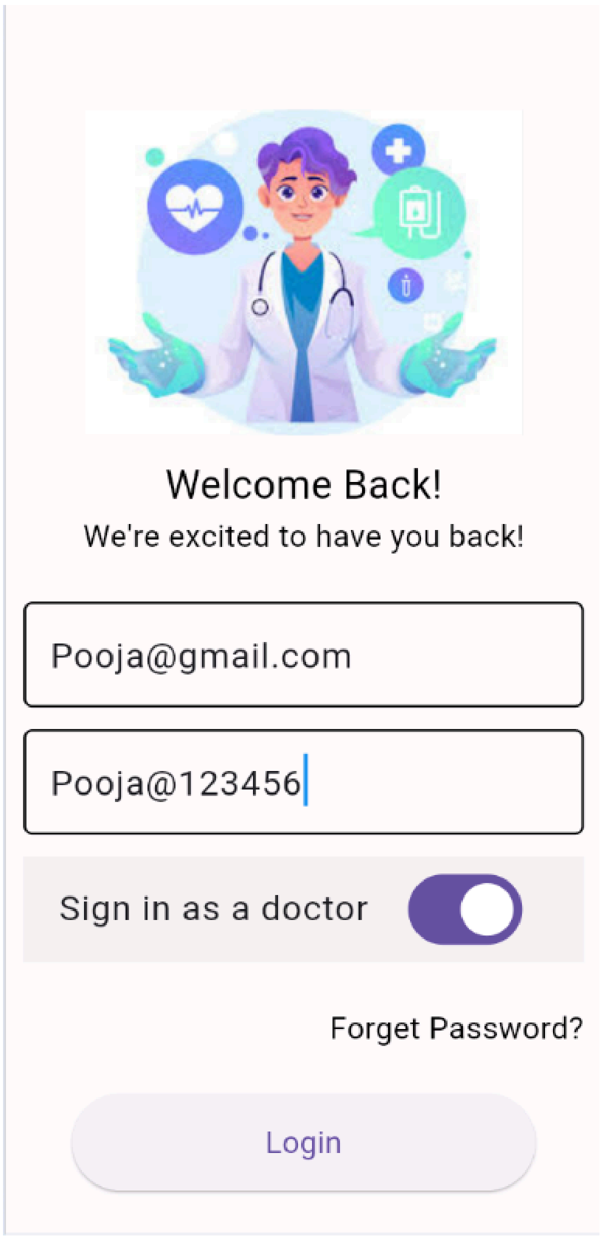
9. **Update or Delete Data:**

- Use the update() or remove() methods to update or delete data:
Dart
database Reference reference = FirebaseDatabase.instance.reference();
// Update data
 reference.child('users').child('userId').update({'username': 'NewUsername'});
// Delete data
reference.child('users').child('userId').remove();

10. **Handle Asynchronous Operations:**
 - Since Firebase operations are asynchronous, make sure to handle them using async/await or .then() .

Name : Pooja Panjwani **Exp : 6** D15B_51

11. **Display Data in Flutter UI:**
- Use Flutter widgets to display the data retrieved from Firebase in your app's UI.



Welcome Back!

We're excited to have you back!

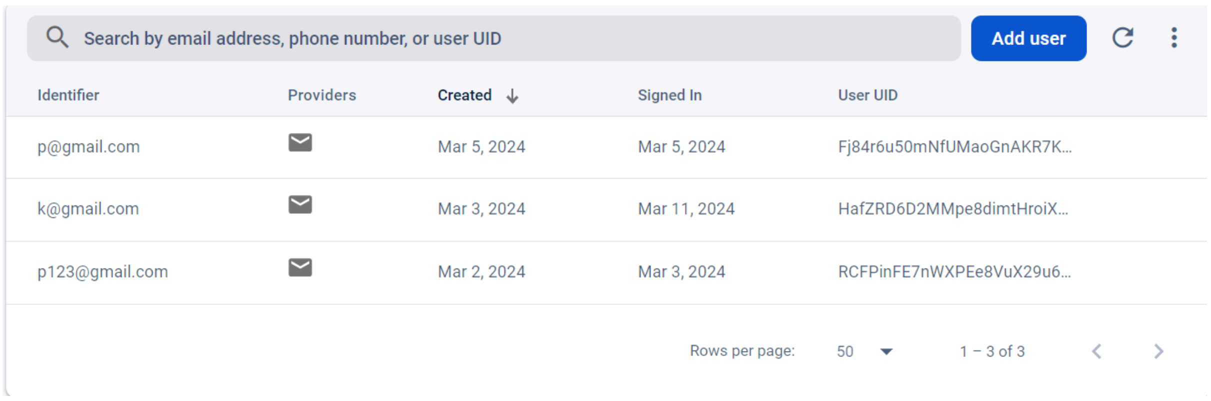
Pooja@gmail.com

Pooja@123456

Sign in as a doctor

Forget Password?

Login



Search by email address, phone number, or user UID

Add user

Identifier	Providers	Created	Signed In	User UID
p@gmail.com	✉	Mar 5, 2024	Mar 5, 2024	Fj84r6u50mNfUMaoGnAKR7K...
k@gmail.com	✉	Mar 3, 2024	Mar 11, 2024	HafZRD6D2MMpe8dimtHroiX...
p123@gmail.com	✉	Mar 2, 2024	Mar 3, 2024	RCFPinFE7nWXPEe8VuX29u6...

Rows per page: 50 1 – 3 of 3

Conclusion : We've connected Flutter, a tool for making app interfaces, with Firebase, a powerful platform for app development. This combination lets developers create apps for both iOS and Android that are secure, interactive, and can store data in real-time. By following a step-by-step process, developers can set up their projects, adjust settings, and test thoroughly before launching their apps to users. It's a smooth way to build and deploy feature-packed apps for multiple platforms.