

Experiment : 06

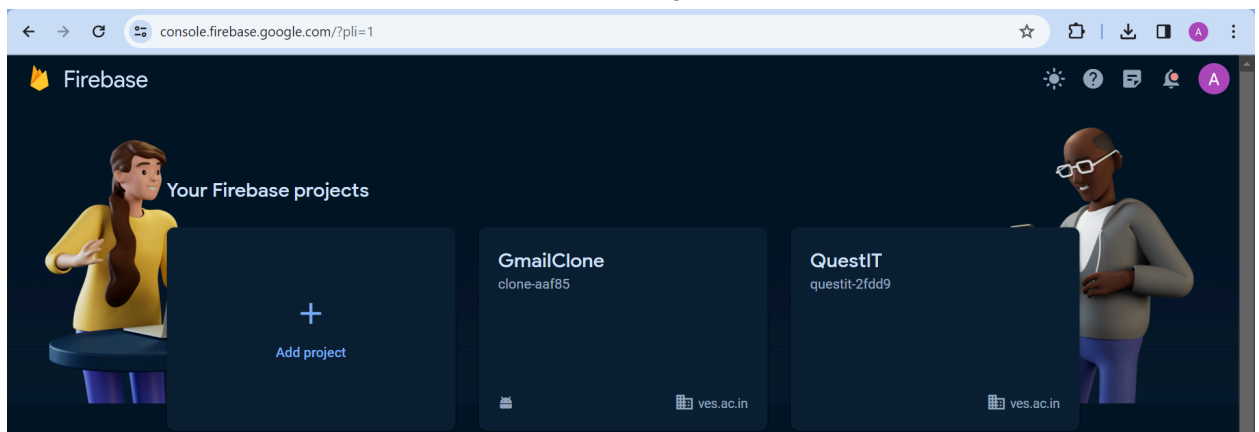
AIM : To Connect Flutter UI with fireBase database

THEORY:

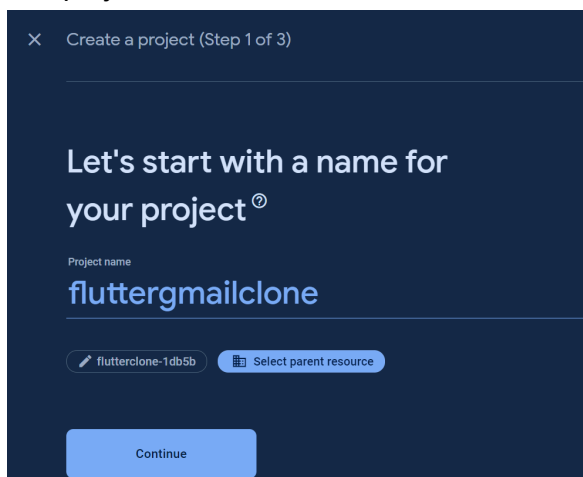
Integrating Firebase with a Flutter application involves several steps, and Firebase offers a variety of backend services that can enhance the functionality of your app. Here's a general process for integrating Firebase with a Flutter application:

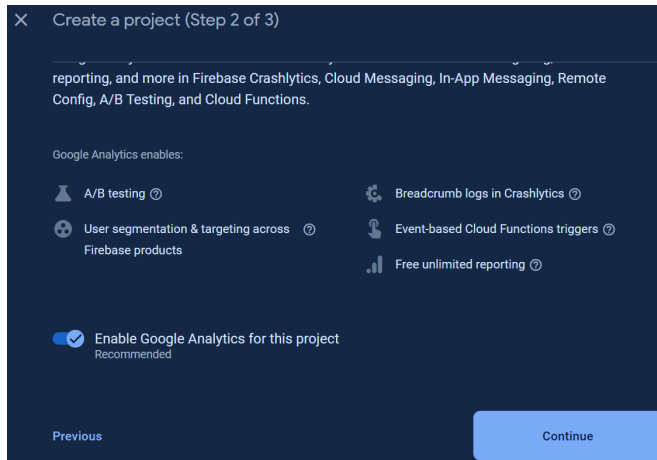
Steps for Firebase Integration in Flutter:

- Go to the **Firebase Console** and click on **"Add Project."**

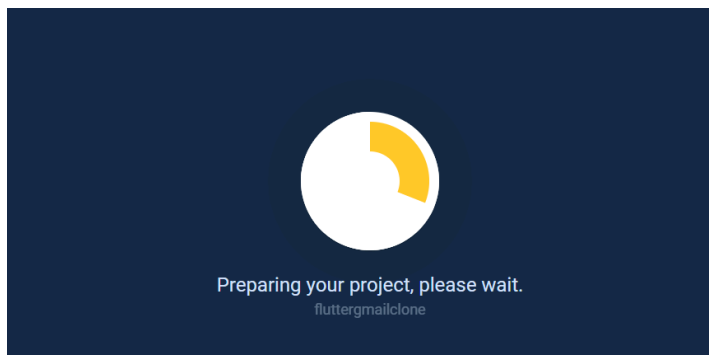


- Add project name:

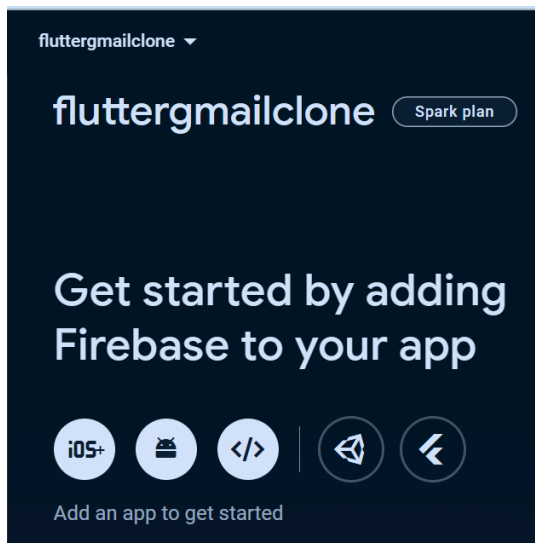




- Create project



- Create app ,click on Android icon



- Register app

The screenshot shows the 'Add Firebase to your Android app' dialog with the title '1 Register app'. It contains three input fields: 'Android package name' with the value 'com.company.appname', 'App nickname (optional)' with the value 'myapp', and 'Debug signing certificate SHA-1 (optional)' with a long hexadecimal string. A 'Register app' button is at the bottom. A note at the bottom states: 'Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.'

- Download the file and place them in the respective directories
app>src>google-services.json

The screenshot shows the '2 Download and then add config file' step. It features a 'Download google-services.json' button, instructions to switch to the Project view in Android Studio, and a diagram showing the file being moved to the 'app' directory. A 'Next' button is at the bottom. On the right, a snippet of the Android Studio Project view shows the file tree with 'google-services.json' highlighted under the 'app' directory. Blue arrows point from the text instructions to the corresponding elements in the Project view.

- Make necessary changes in following files:

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

☐ Kotlin DSL (`build.gradle.kts`) ☒ Groovy (`build.gradle`)

Add the plugin 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
}
```

2. Then, in your **module (app-level)** `build.gradle` file, add both the `google-services` plugin 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:32.7.2')

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

- Continue to console:

4 Next steps

You're all set!

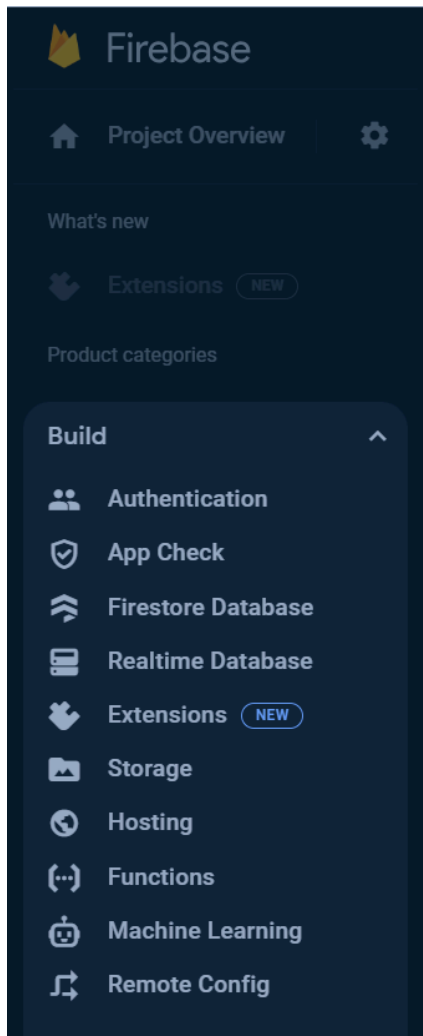
Make sure to check out the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

[Previous](#) [Continue to console](#)

- Firebase Services like Authentication,Firestore Database,Realtime Database can be used,



- Add Firebase dependencies to your pubspec.yaml file:

pubspec.yaml

dependencies:

firebase_core: ^latest_version

firebase_auth: ^latest_version

cloud_firestore: ^latest_version

Run **flutter pub get** to install the dependencies.

- **Initialize Firebase in your Flutter app**

main.dart file:

```
import 'package:firebase_core/firebase_core.dart';
```

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

Conclusion: Connected Flutter UI with fireBase database.
