Nidhi Pednekar
D15B / 43
MPL Experiment 6

AIM: To Set Up Firebase with Flutter for iOS and Android Apps

Theory: Firebase is a Backend-as-a-Service (BaaS) platform by Google that provides various services like authentication, real-time databases, cloud storage, and hosting for mobile and web applications. Integrating Firebase with Flutter allows developers to leverage these services seamlessly for both Android and iOS platforms. By configuring Firebase correctly in a Flutter project, developers can enable backend functionalities such as user authentication, data storage, and cloud messaging without needing a separate server-side implementation.

Step-by-Step Guide to Setting Up Firebase with Flutter (iOS & Android)

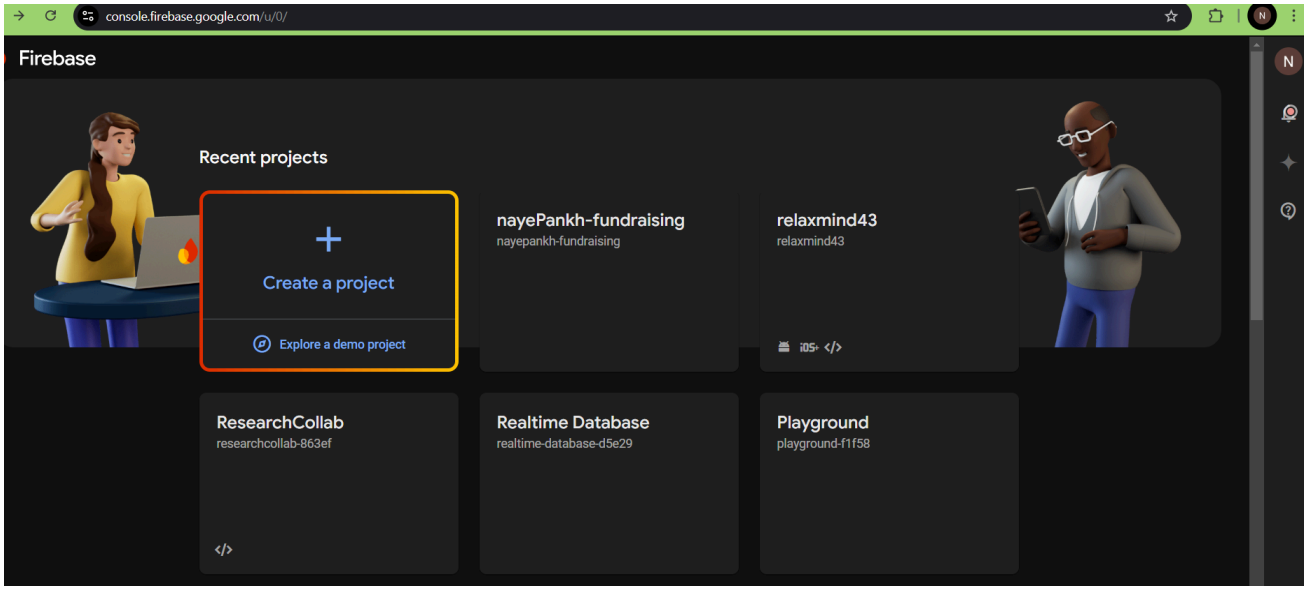
Step 1: Prerequisites

Ensure you have the following:

- A **Google account** to access Firebase.
- **Flutter installed** on your system.
- **Android Studio** and **Visual Studio Code** installed.
- **Xcode installed** (for iOS development).
- **Flutter and Dart plugins** installed in Android Studio.
- **Flutter extension** installed in Visual Studio Code.

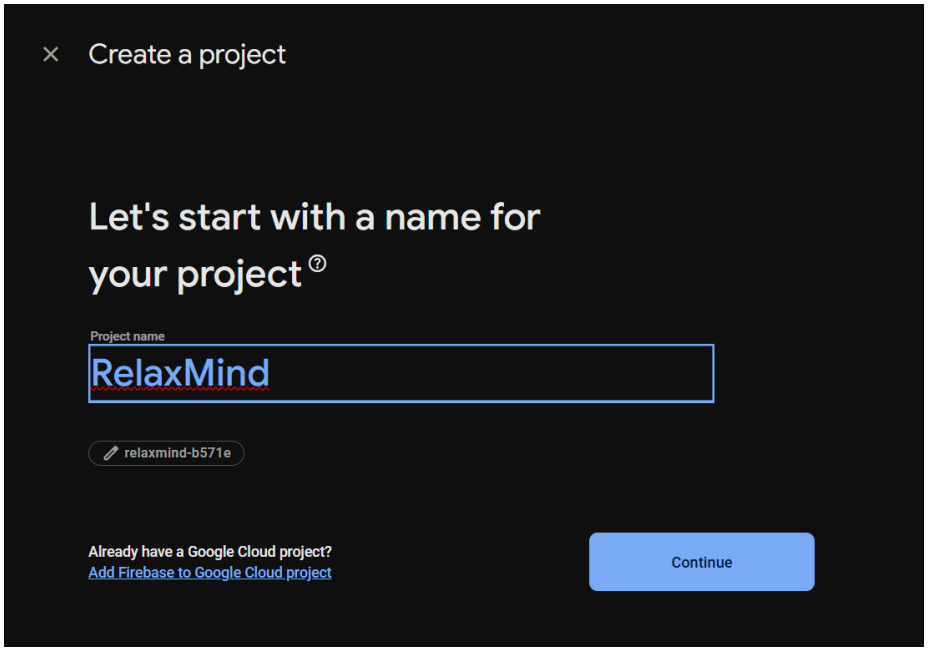
Step 2: Create a New Flutter Project

- Open a terminal and create a new Flutter project.
- Navigate to the project directory.



Step 3: Create a Firebase Project

1. Go to the Firebase Console.
2. Click **"Create a project"** and provide a project name.
3. Choose whether to enable **Google Analytics** (optional).
4. Click **"Continue"** and wait for Firebase to set up the project.



Create a project

AI assistance for your Firebase project

Gemini in Firebase is integrated within the Firebase console to help streamline your development process.

- Chat with Gemini to plan and design your application, troubleshoot issues, and get recommendations based on best practices
- Get AI assistance in Firebase Crashlytics for debugging and troubleshooting issues in your Apple and Android apps

Enable Gemini in Firebase
Recommended

Previous

Continue

Disclaimers:
Gemini in Firebase is still under development and may give inaccurate information. Test any code it generates before using it in

Create a project

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

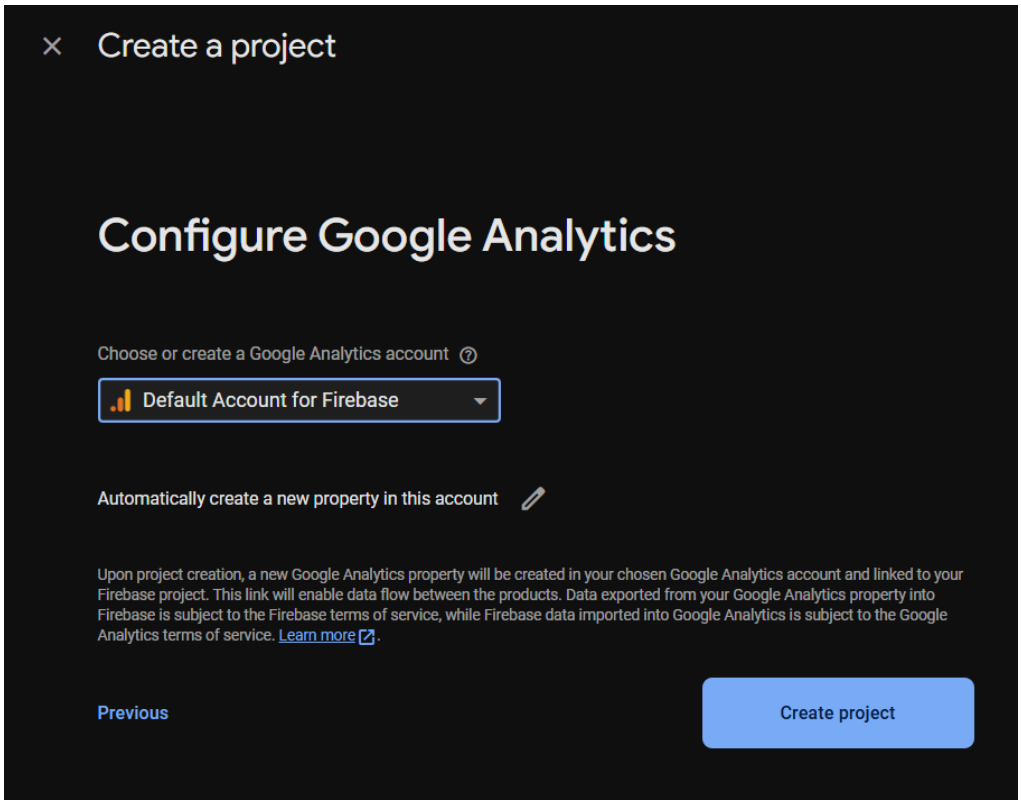
Google Analytics enables:

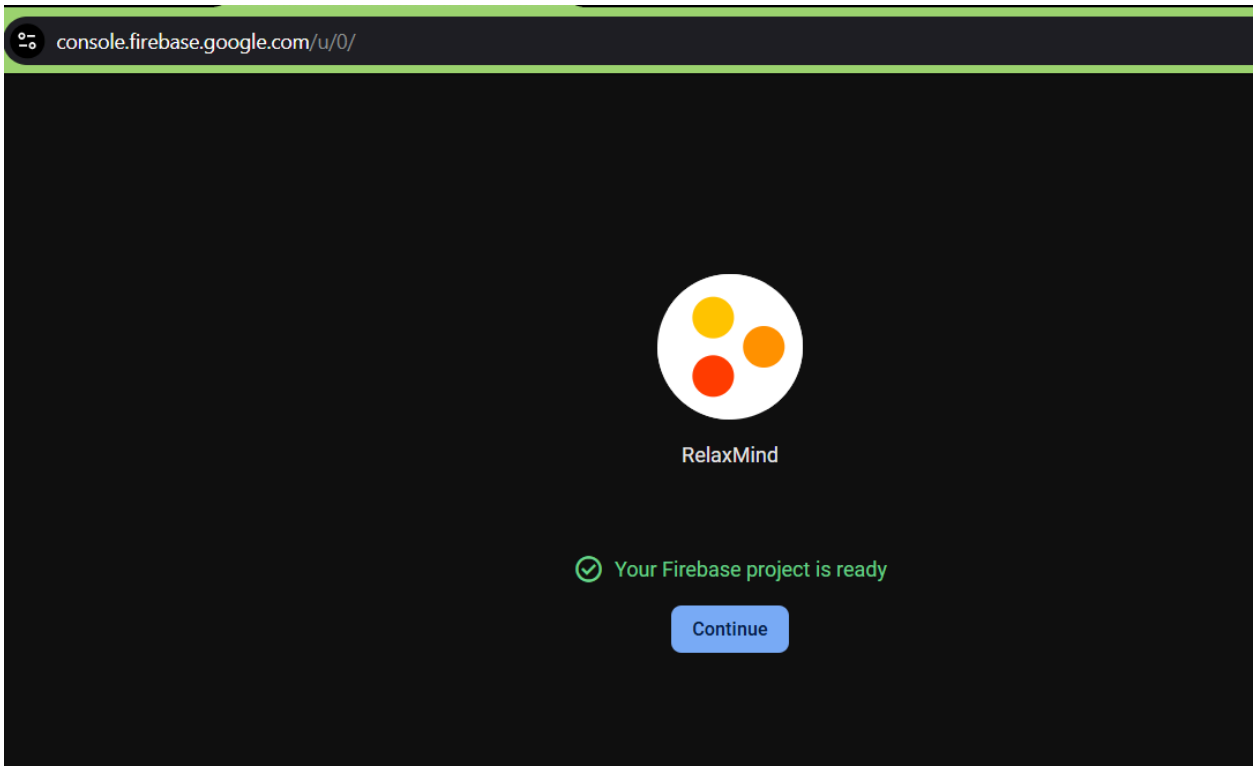
- A/B testing
- User segmentation & targeting across Firebase products
- Breadcrumb logs in Crashlytics
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

Previous

Continue



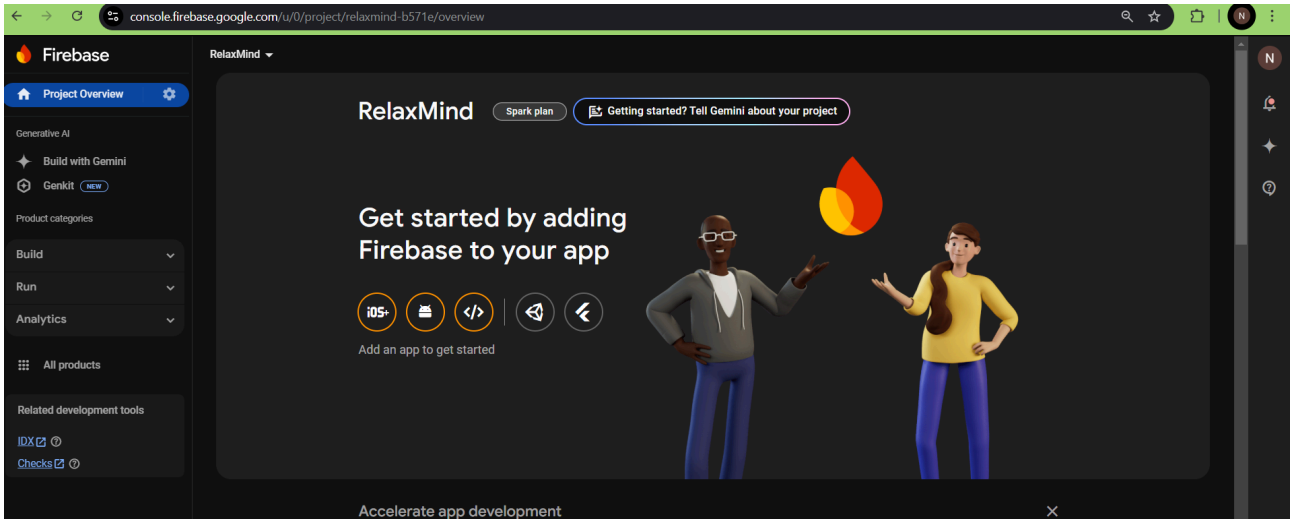


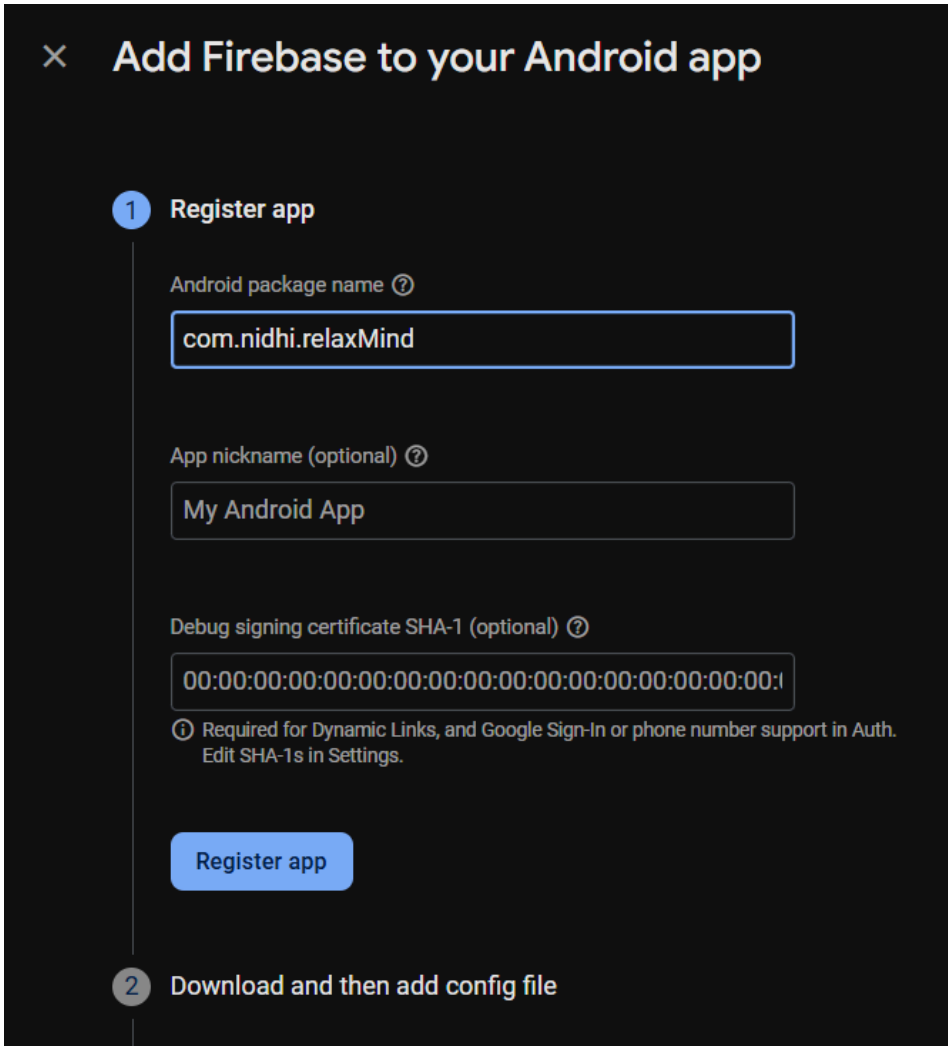
Step 4: Add Firebase to Android

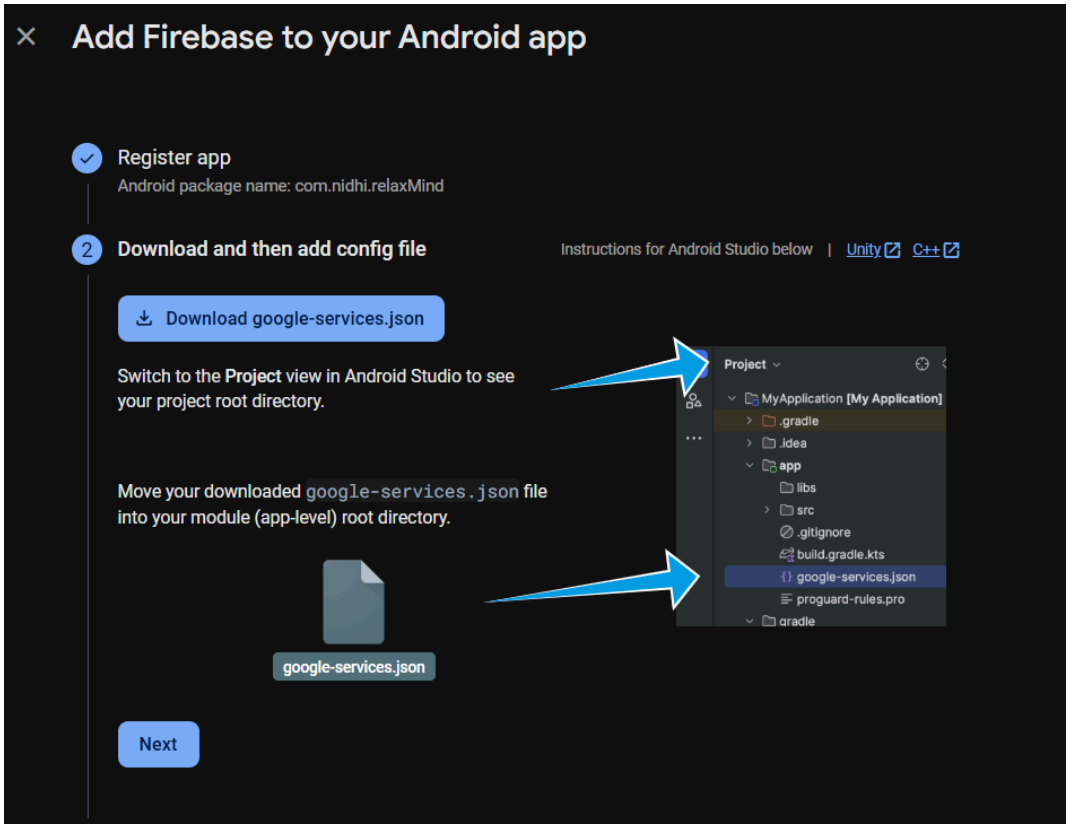
1. In Firebase, select **Android** and register your app.
2. Enter the **Android package name** (must match the one in your app).
3. Download the **google-services.json** file from Firebase.
4. Move the file to the appropriate location in your Flutter project.
5. Update the project's build configuration files to include Firebase dependencies.
6. Run the Flutter app on an Android device or emulator to verify the setup.

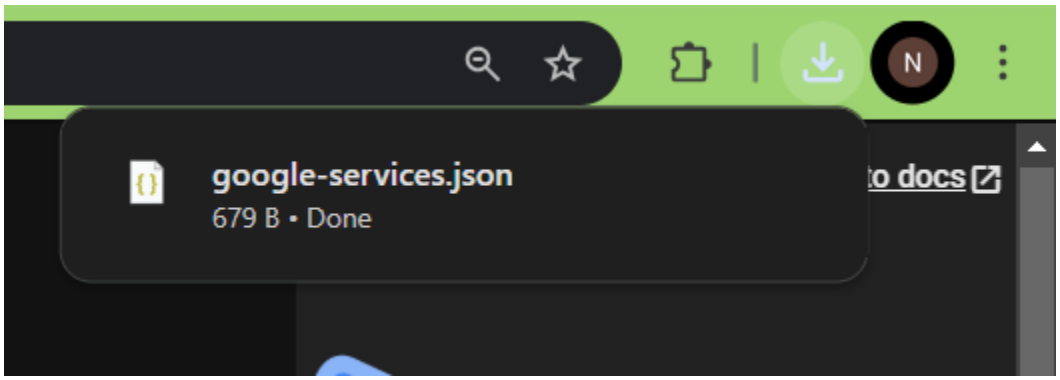
Step 5: Add Firebase to iOS

1. In Firebase, select **iOS** and register your app.
2. Enter the **iOS Bundle ID** (should match the one in your project).
3. Open the iOS project in Xcode and update the **Bundle Identifier**.
4. Download the **GoogleService-Info.plist** file from Firebase.
5. Move the file into the correct directory inside your Xcode project.
6. Ensure Firebase is initialized properly for iOS.




Add Firebase to your Android app
1
Register app
Android package name
com.nidhi.relaxMind
App nickname (optional)
My Android App
Debug signing certificate SHA-1 (optional)
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0
Required for Dynamic Links, and Google Sign-In or phone number support in Auth.
Edit SHA-1s in Settings.
Register app
2
Download and then add config file


Add Firebase to your Android app
Register app
Android package name: com.nidhi.relaxMind
2
Download and then add config file
Instructions for Android Studio below | Unity C++
Download google-services.json
Switch to the Project view in Android Studio to see your project root directory.
Move your downloaded google-services.json file into your module (app-level) root directory.
google-services.json
Project
MyApplication [My Application]
.gradle
.idea
app
libs
src
.gitignore
build.gradle.kts
google-services.json
proguard-rules.pro
gradle
Next


N
google-services.json
679 B • Done
to docs

3 Add Firebase SDK

Instructions for Gradle | Unity C++

> Are you still using the `buildscript` syntax to manage plugins? Learn how to add Firebase plugins using that syntax.

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.kts` file:

   Root-level (project-level) Gradle file (`<project>/build.gradle.kts`):

```
plugins {
  // ...

  // Add the dependency for the Google services Gradle plugin
  id("com.google.gms.google-services") version "4.4.2" apply false
}
```

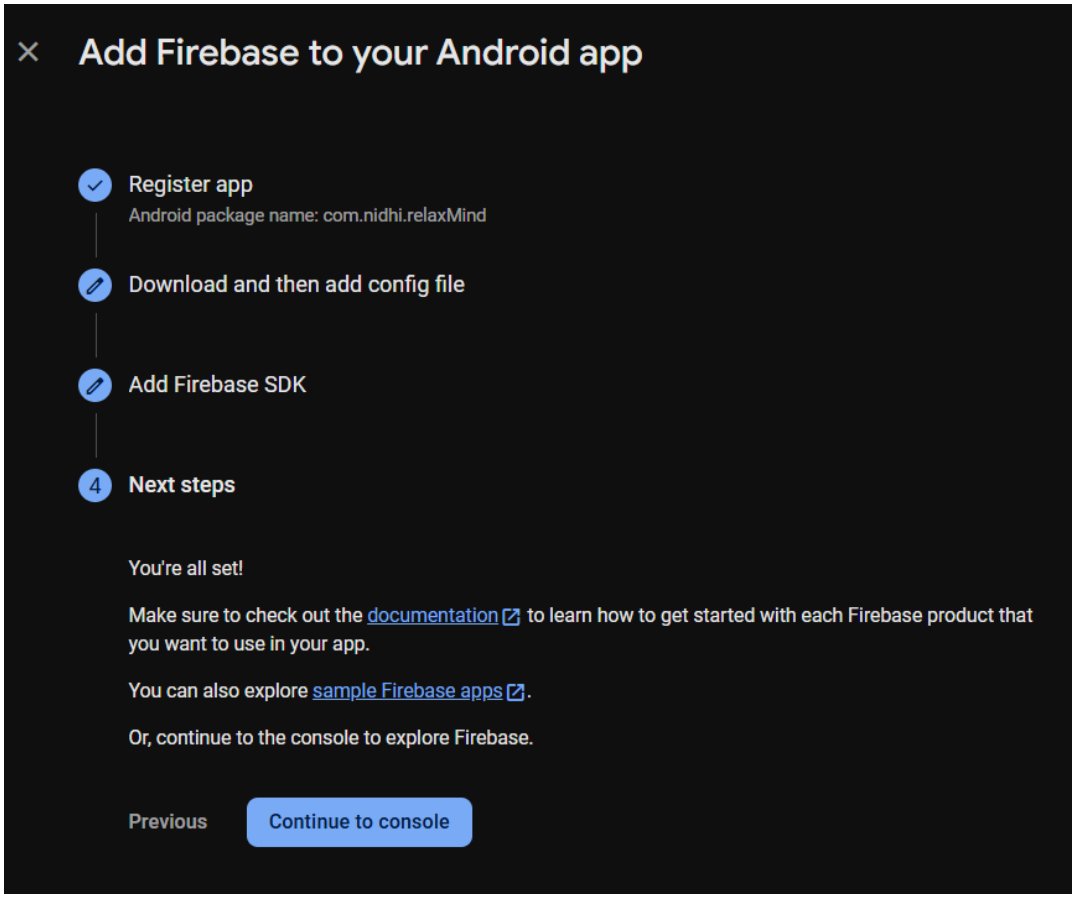
2. Then, in your **module (app-level)** `build.gradle.kts` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

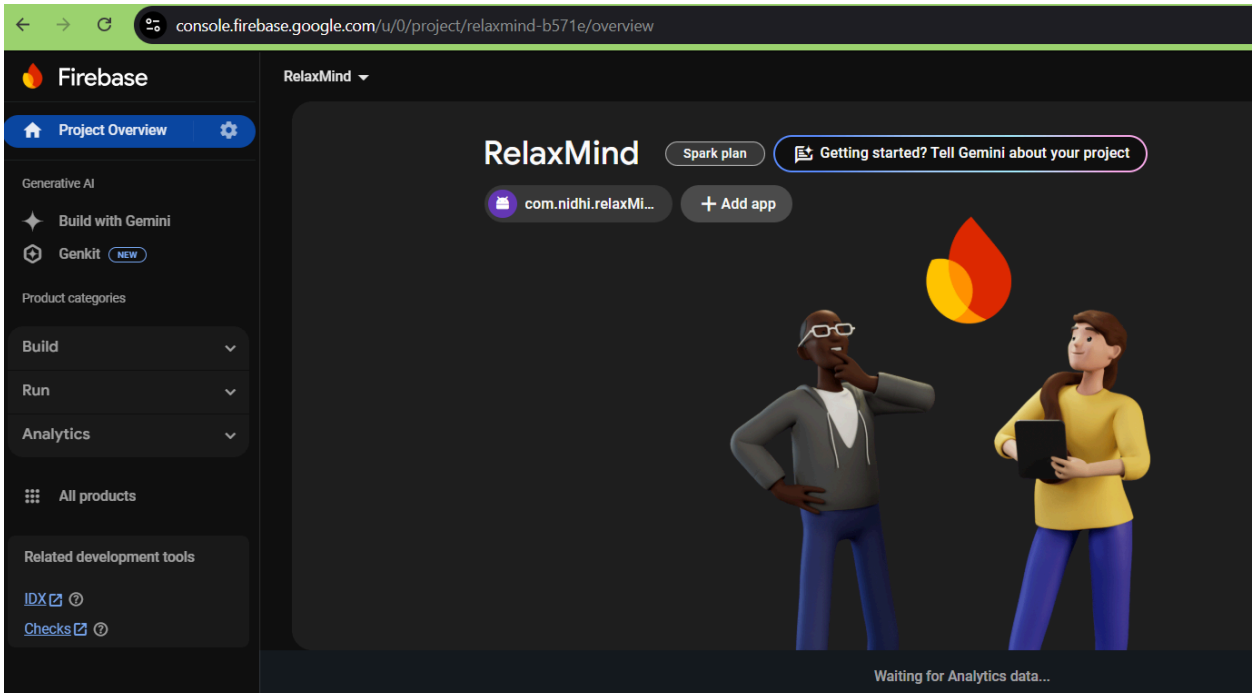
   Module (app-level) Gradle file (`<project>/<app-module>/build.gradle.kts`):

```
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:33.10.0"))

  // TODO: Add the dependencies for Firebase products you want to use
  // When using the BoM, don't specify versions in Firebase dependencies
  implementation("com.google.firebase:firebase-analytics")

  // Add the dependencies for any other desired Firebase products
  // https://firebase.google.com/docs/android/setup#available-libraries
}
```

   By using the Firebase Android BoM, your app will always use compatible Firebase library versions. Learn more

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.


Add Firebase to your Android app
Register app
Android package name: com.nidhi.relaxMind
Download and then add config file
Add Firebase SDK
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


console.firebase.google.com/u/0/project/relaxmind-b571e/overview
Firebase
RelaxMind
Project Overview
Generative AI
Build with Gemini
Genkit
NEW
Product categories
Build
Run
Analytics
All products
Related development tools
IDX
Checks
RelaxMind
Spark plan
Getting started? Tell Gemini about your project
com.nidhi.relaxMi...
Add app
Waiting for Analytics data...

Step 6: Run the Flutter App

1. Run the Flutter app on a real device or simulator.
2. Check the Firebase dashboard to confirm that the app is successfully connected.

Conclusion: Successfully setting up Firebase with Flutter enables seamless backend integration for mobile applications. By following the step-by-step process of creating a Firebase project, registering Android and iOS apps, and configuring dependencies, developers can establish a reliable connection between their app and Firebase services. Running the Flutter app and verifying Firebase integration ensures that authentication, database, and other services function correctly, enhancing app development efficiency.