Aditya P Tahiliani **Batch :- C**

D15A_59

Experiment no. :- 06

Aim: **To Connect Firebase Database With Flutter UI**

Theory:

Firebase is a great backend solution for anyone that wants to use authentication, databases, cloud functions, ads, and countless other features within an app.

Prerequisites

To complete this tutorial, you will need:

● A Google account to use Firebase.

● Developing for iOS will require XCode.

● To download and install Flutter.

● To download and install Android Studio and Visual Studio Code.

● It is recommended to install plugins for your code editor:

 ○ Flutter and Dart plugins installed for Android Studio.

 ○ Flutter extension installed for Visual Studio Code

This tutorial was verified with Flutter v2.0.6, Android SDK v31.0.2, and Android Studio v4.1.

Creating a New Flutter Project

This tutorial will require the creation of an example Flutter app.

Once you have your environment set up for Flutter, you can run the following to create a new application:

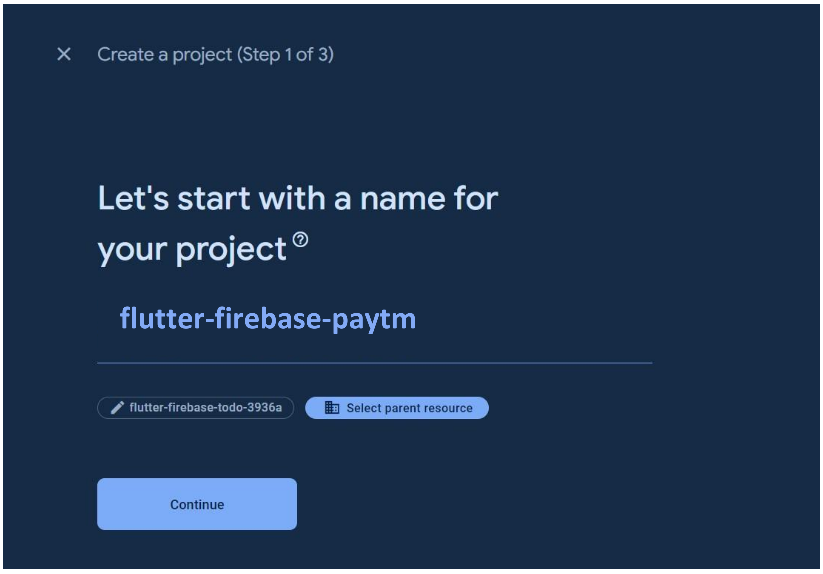
```
flutter create flutterfirebaseexample
```

Using flutter create will produce a demo application that will display the number of times a button is clicked.

Now that we’ve got a Flutter project up and running, we can add Firebase.

Creating a New Firebase Project

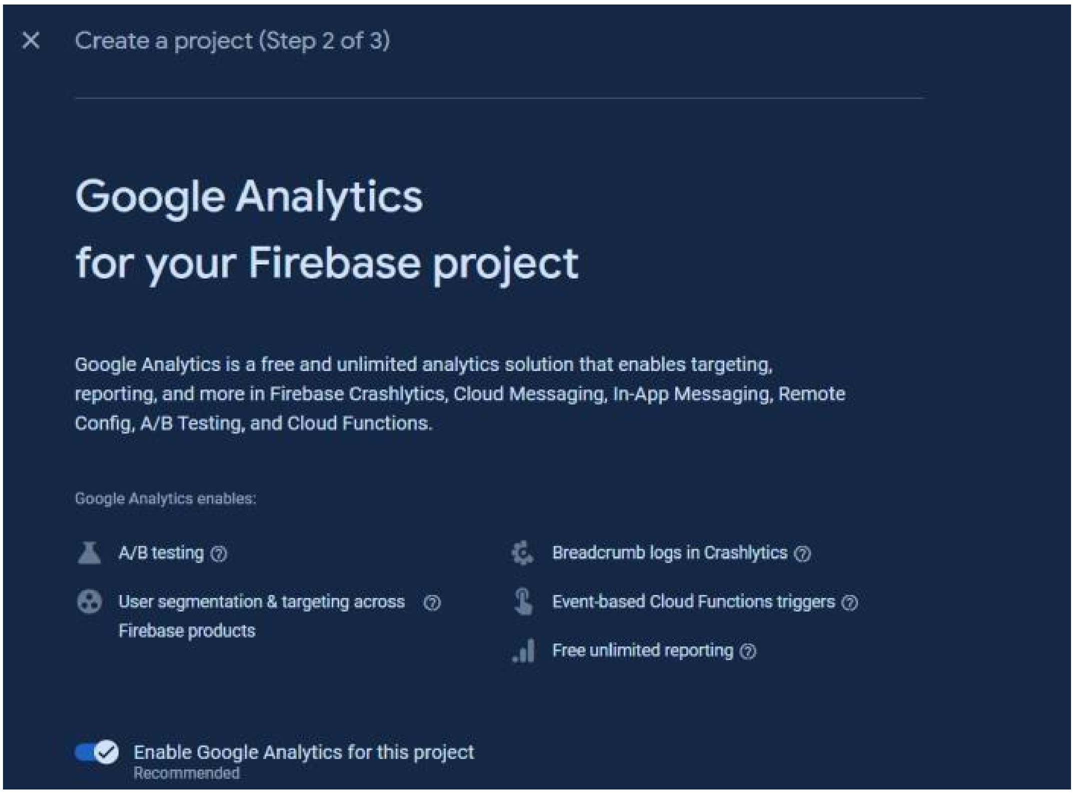
First, log in with your Google account to manage your Firebase projects. From within the Firebase dashboard, select the Create new project button and give it a name:



× Create a project (Step 1 of 3)

Let's start with a name for your project

flutter-firebase-paytm



× Create a project (Step 2 of 3)

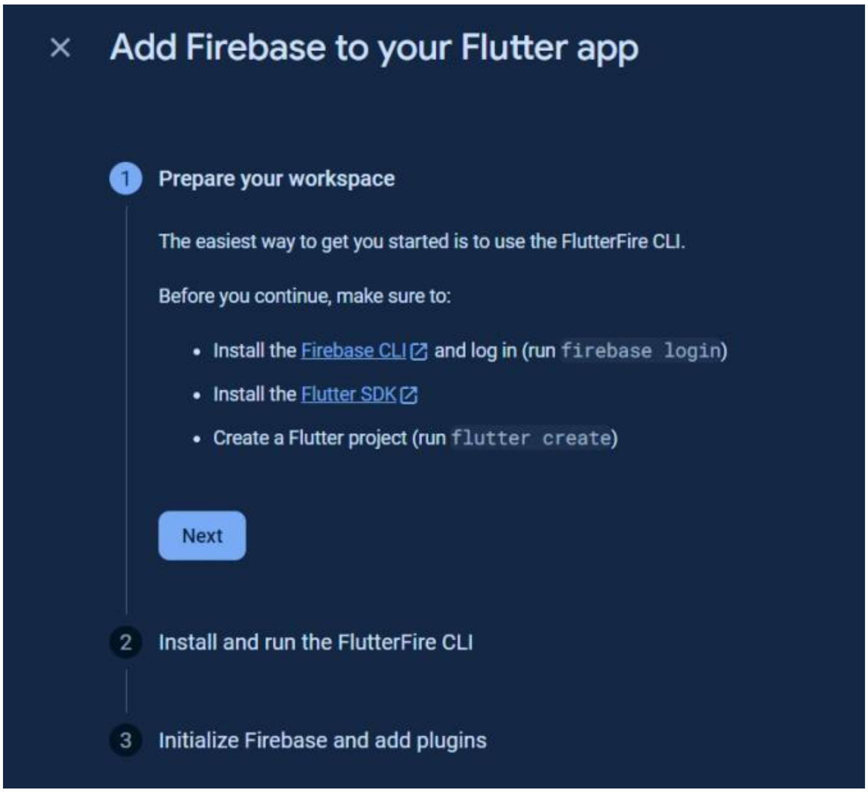
Google Analytics for your Firebase project

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



Add Firebase to your Flutter app

1 Prepare your workspace

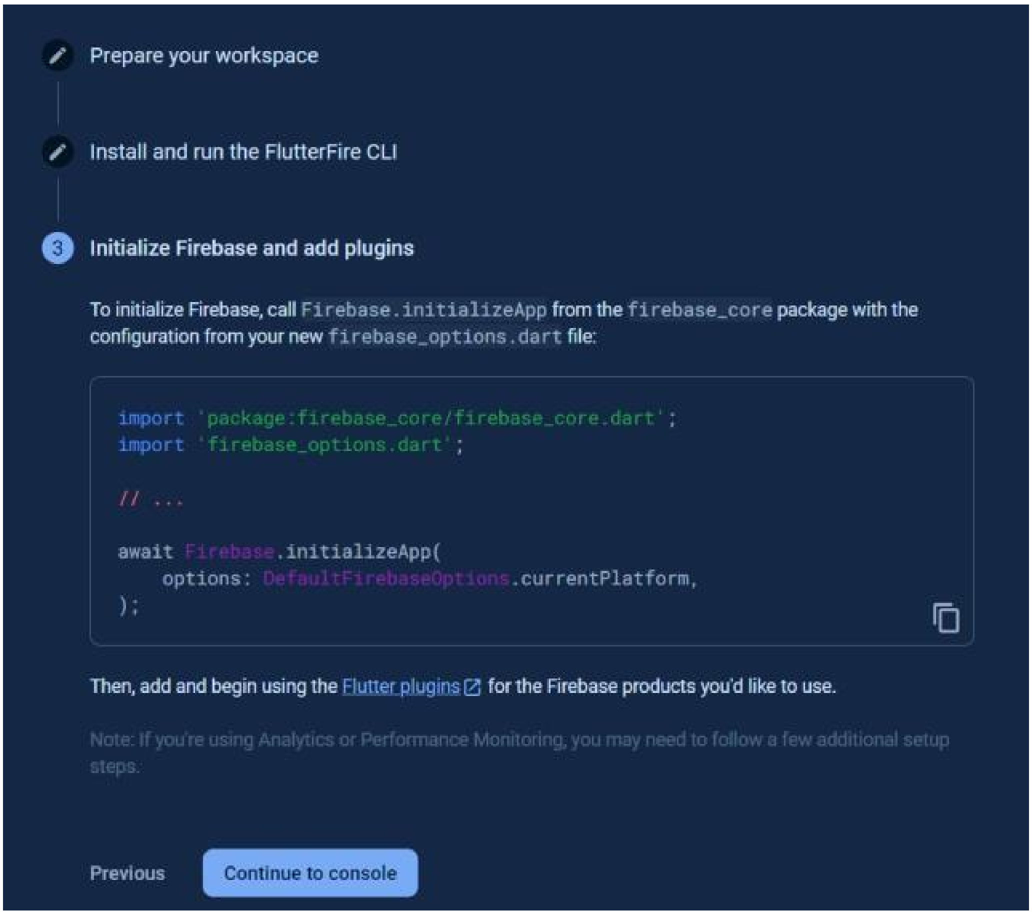
The easiest way to get you started is to use the FlutterFire CLI.

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



Prepare your workspace

Install and run the FlutterFire CLI

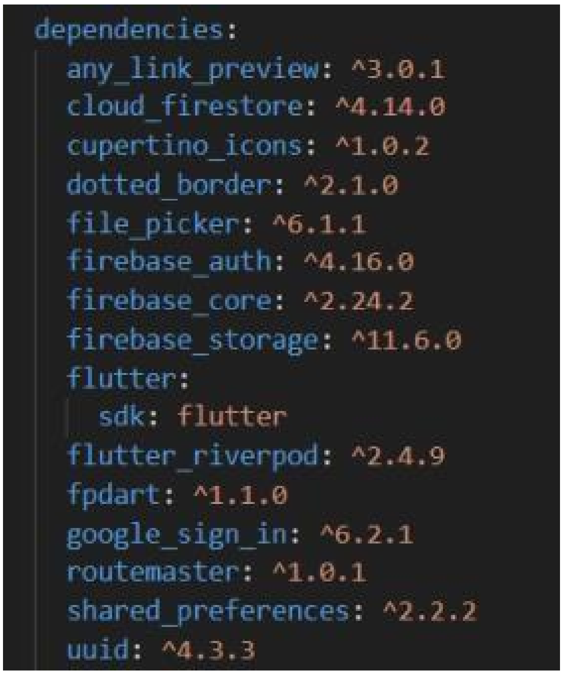
3 Initialize Firebase and add plugins

To initialize Firebase, call `Firebase.initializeApp` from the `firebase_core` package with the configuration from your new `firebase_options.dart` file:

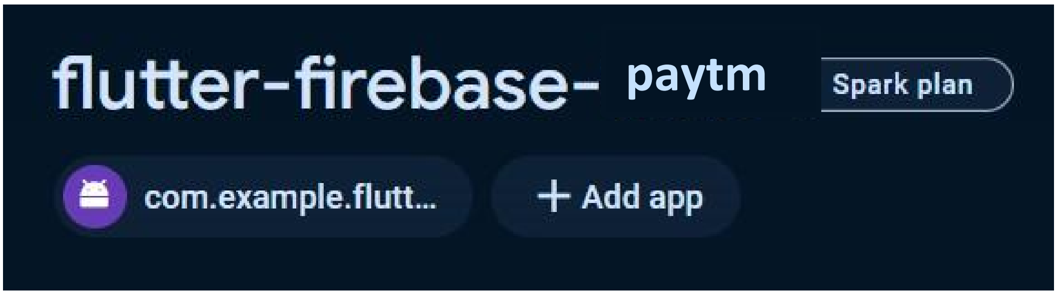
```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// ...

await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
);
```

Then, add and begin using the Flutter plugins for the Firebase products you'd like to use.

Note: If you're using Analytics or Performance Monitoring, you may need to follow a few additional setup steps.



```
dependencies:
  any_link_preview: ^3.0.1
  cloud_firestore: ^4.14.0
  cupertino_icons: ^1.0.2
  dotted_border: ^2.1.0
  file_picker: ^6.1.1
  firebase_auth: ^4.16.0
  firebase_core: ^2.24.2
  firebase_storage: ^11.6.0
  flutter:
    sdk: flutter
  flutter_riverpod: ^2.4.9
  fpdart: ^1.1.0
  google_sign_in: ^6.2.1
  routemaster: ^1.0.1
  shared_preferences: ^2.2.2
  uuid: ^4.3.3
```



Conclusion :- Thus, we successfully Connected Firebase Database With Flutter User Interface.