Name : Anushka Karhadkar   Roll No : 28   Div: D15B   Batch: B

Experiment No: 06

Aim: To Setup Firebase with Flutter for iOS and Android app

Theory:

Firebase

Google Firebase is a set of cloud-based development tools that helps mobile app developers build, deploy and scale their apps.

Firebase is a mobile and web application development platform that provides a suite of cloud-based services to help developers build and scale their applications more easily. It was initially developed by Firebase, Inc., which was later acquired by Google in 2014.

Prerequisites

- 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:
    1. Flutter and Dart plugins installed for Android Studio.
    2. Flutter extension installed for Visual Studio Code.

Implementation
Creating a New Flutter Project

Creating a New Flutter Project:

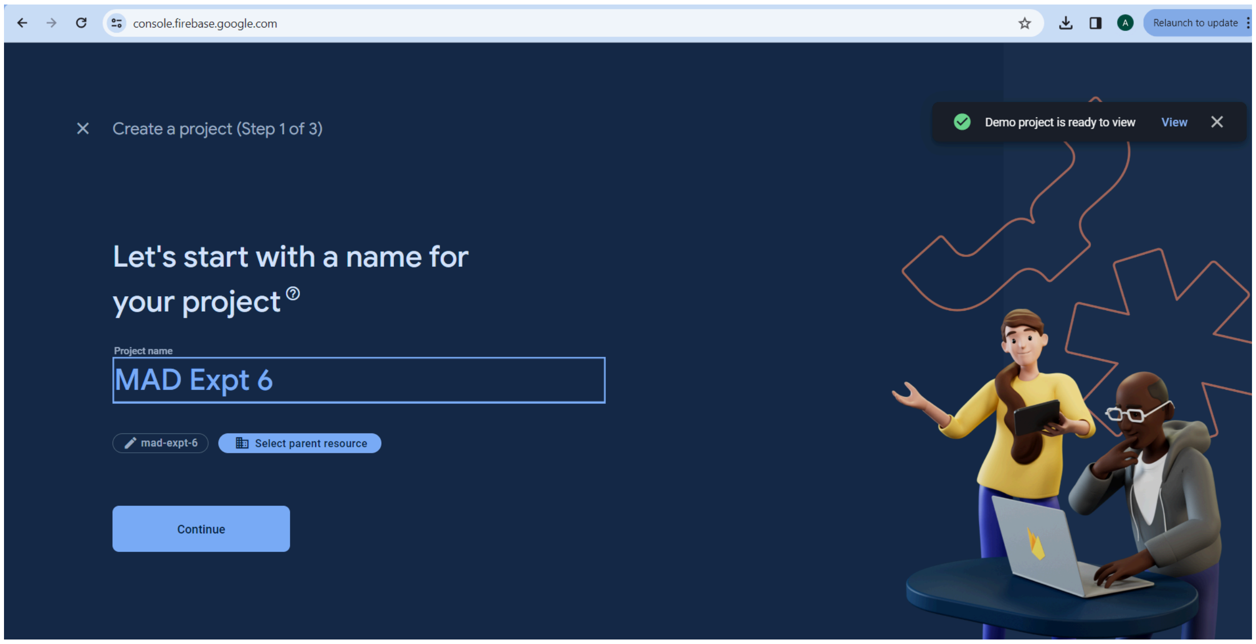
Once you have your environment set up for Flutter, you can run the following to create a new
application:

Flutter create MAD Expt 6
Copy
Navigate to the new project directory:
cd MADExp4
Using flutter create will produce a demo application that will display the number of times a
button is clicked.

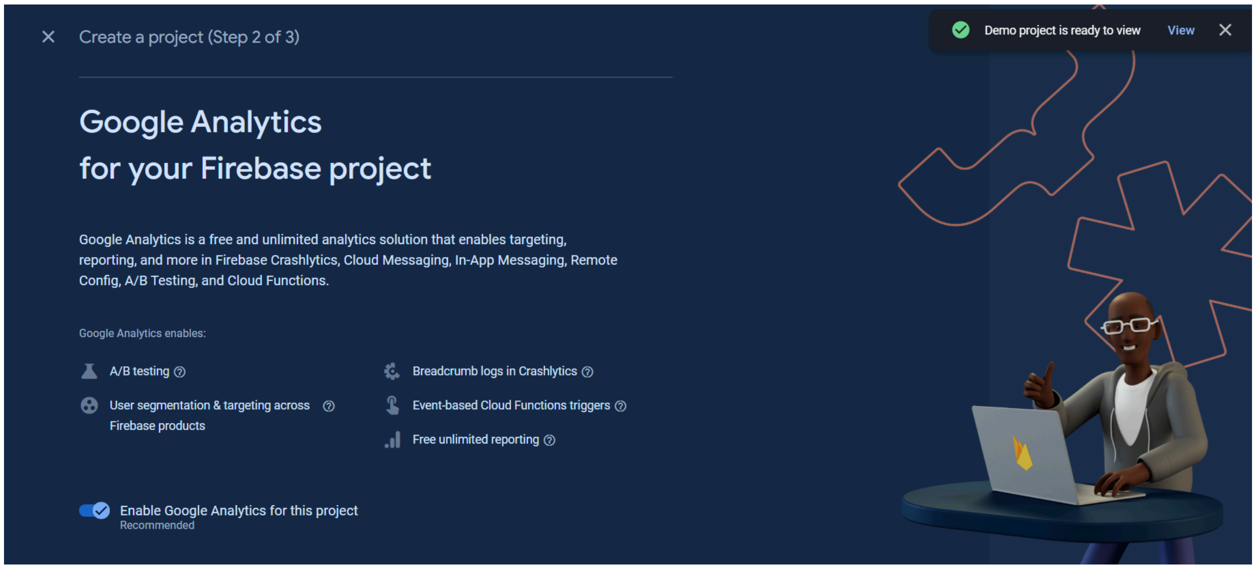
Now that we've got a Flutter project up and running, we can add Firebase.

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


console.firebase.google.com
Relaunch to update
Create a project (Step 1 of 3)
Demo project is ready to view
View
Let's start with a name for your project
Project name
MAD Expt 6
mad-expt-6
Select parent resource
Continue

Next, we're given the option to enable Google Analytics. This tutorial will not require Google Analytics, but you can also choose to add it to your project.


Create a project (Step 2 of 3)

Demo project is ready to view View

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

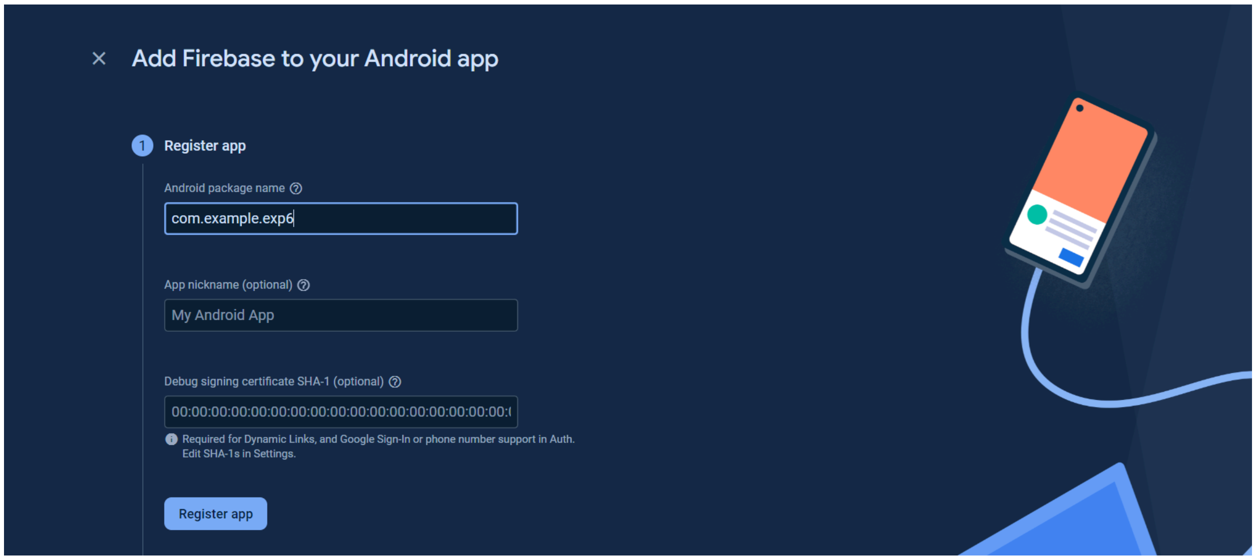
Enable Google Analytics for this project ✓
Recommended

if you choose to use Google Analytics, you will need to review and accept the terms and conditions prior to project creation.
After pressing Continue, your project will be created and resources will be provisioned. You will then be directed to the dashboard for the new project.

Adding Android support

Registering the App

In order to add Android support to our Flutter application, select the Android logo from the dashboard. This brings us to the following screen:


Add Firebase to your Android app

1 Register app

Android package name: com.example.exp6

App nickname (optional): My Android App

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0

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

Register app

The most important thing here is to match up the Android package name that you choose here

with the one inside of our application.
The structure consists of at least two segments. A common pattern is to use a domain name, a
company name, and the application name:
com.example.exp6
Once you've decided on a name, open android/app/build.gradle in your code editor and update the applicationId to match the Android package name:
android/app/build.gradle

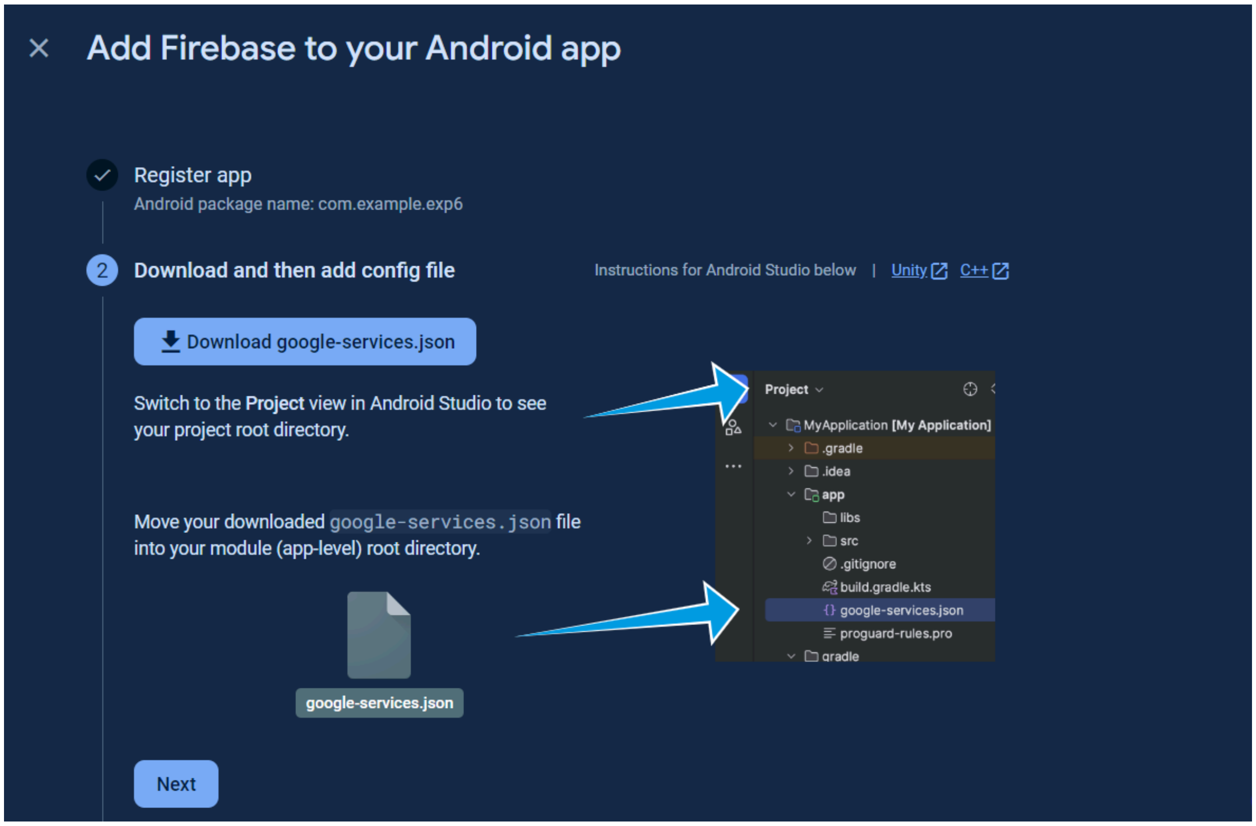
```
...
defaultConfig {
// TODO: Specify your own unique Application ID
(https://developer.android.com/studio/build/application-id.html).
applicationId 'com.example.exp6'
… }
```

You can skip the app nickname and debug signing keys at this stage. Select Register app to
continue.

Downloading the Config File

The next step is to add the Firebase configuration file into our Flutter project. This is important as it contains the API keys and other critical information for Firebase to use.

Select Download google-services.json from this page:


Add Firebase to your Android app

Register app
Android package name: com.example.exp6

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.

Next

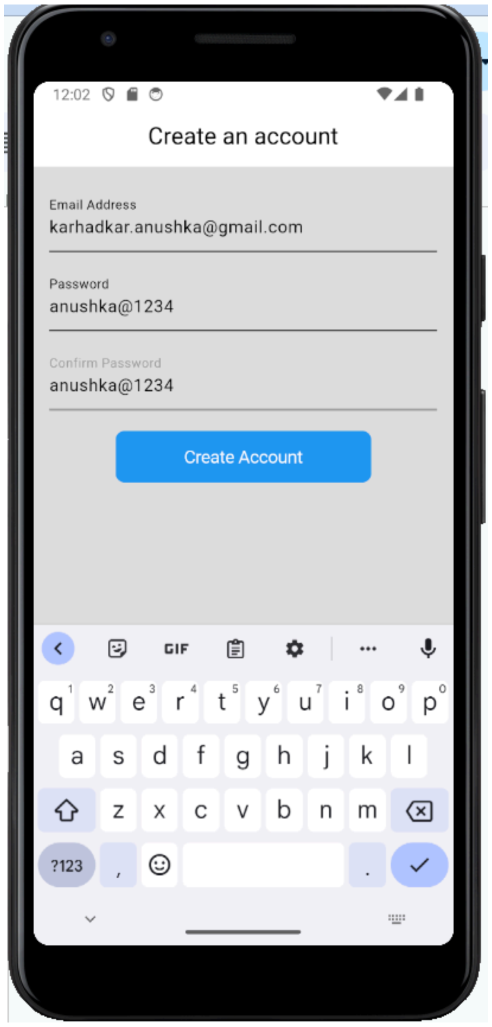
Next, move the google-services.json file to the android/app directory within the Flutter project.
Adding the Firebase SDK
We'll now need to update our Gradle configuration to include the Google Services plugin.
Open android/build.gradle in your code editor and modify it to include the following:
android/buiild.gradle

```
buildscript {
repositories {
// Check that you have the following line (if not, add it):
google() // Google's Maven repository
}
dependencies {
...
// Add this line
classpath 'com.google.gms:google-services:4.3.6'
}
}
allprojects {
...
repositories {
// Check that you have the following line (if not, add it):
google() // Google's Maven repository
...
}
}
```

Finally, update the app level file at android/app/build.gradle to include the following:

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'
dependencies {
// Import the Firebase BoM
implementation platform('com.google.firebase:firebase-bom:28.0.0')
// Add the dependencies for any other desired Firebase products
// https://firebase.google.com/docs/android/setup#available-libraries }
```

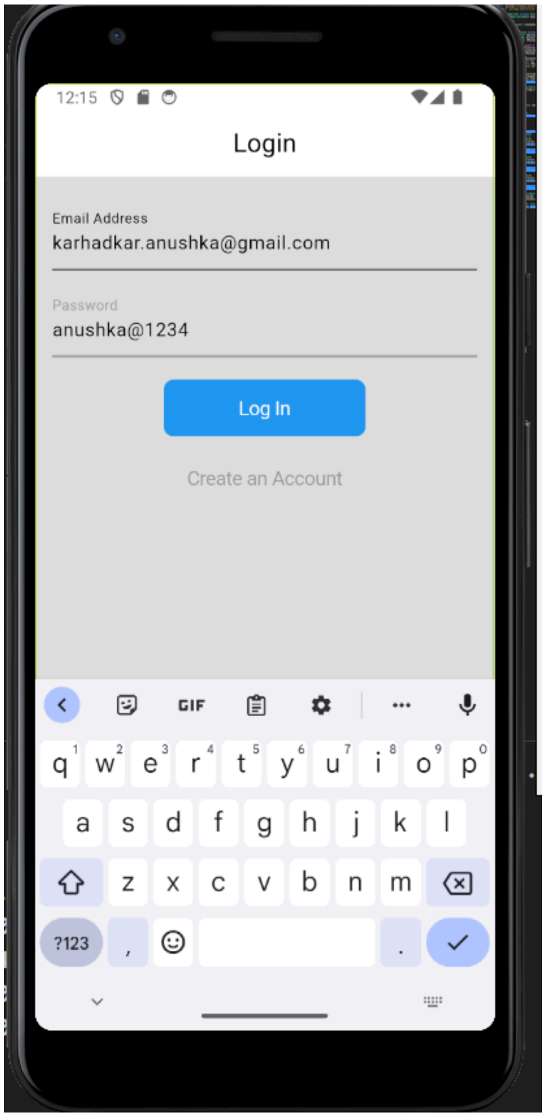
With this update, we’re essentially applying the Google Services plugin as well as looking at
how other Flutter Firebase plugins can be activated such as Analytics.
From here, run your application on an Android device or simulator. If everything has worked
correctly, you should get the following message in the dashboard:


Create an account

Email Address
karhadkar.anushka@gmail.com

Password
anushka@1234

Confirm Password
anushka@1234


12:15

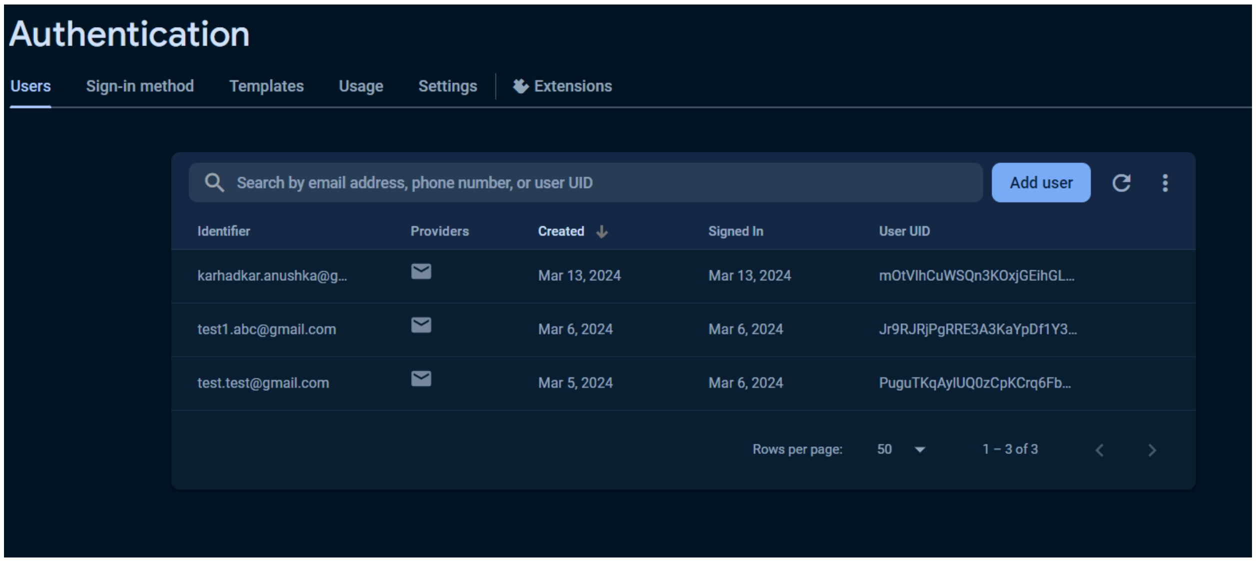
Login

Email Address
karhadkar.anushka@gmail.com

Password
anushka@1234

Log In

Create an Account


Authentication

Users | Sign-in method | Templates | Usage | Settings | Extensions

Search by email address, phone number, or user UID

Add user

Identifier	Providers	Created	Signed In	User UID
karhadkar.anushka@g...	✉	Mar 13, 2024	Mar 13, 2024	mOtVlhCuWSQn3KOxjGEihGL...
test1.abc@gmail.com	✉	Mar 6, 2024	Mar 6, 2024	Jr9RJRjPgRRE3A3KaYpDf1Y3...
test.test@gmail.com	✉	Mar 5, 2024	Mar 6, 2024	PuguTKqAylUQ0zCpKCrq6Fb...

Rows per page: 50 1 – 3 of 3

Conclusion:

We have learned how to set up and ready our Flutter applications to be used with Firebase. Flutter has official support for Firebase with the FlutterFire set of libraries.