# Lab 24

Instructor: Sidra Khatoon

E-mail: skhatoon[@uit.edu](mailto:ad@uit.edu)

# Objective

# The objective of lab is to Setting Up Firebase in Flutter

**Student Information**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Student ID** |  |
| **Date** |  |

**Assessment**

|  |  |
| --- | --- |
| **Marks Obtained** |  |
| **Remarks** |  |
| **Signature** |  |

# Objective

# The objective of lab is to Setting Up Firebase in Flutter

# Instructions

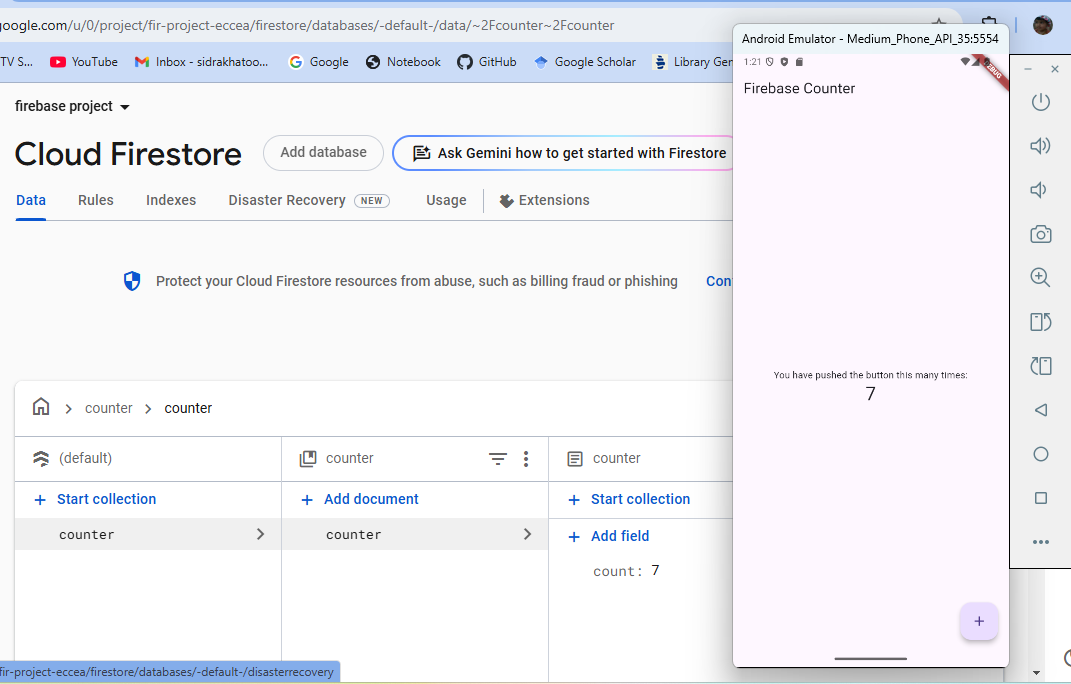
You have to perform the following tasks yourselves. Raise your hand if you face any difficulty in understanding and solving these tasks. **Plagiarism** is an abhorrent practice and you should not engage in it.

# How to Submit?

Submit lab work using Teams.

**Firestore Counter App: Firestore and Flutter**

Develop your Flutter Firestore integration in a risk-free, costless environment with Firebase cloud firestore. Switch over to Production Firebase once your app is ready, with 0 changes to application code



.

# Introduction:

As a developer, there are many things that you need to do before deploying your application. One of them is testing the application to make sure that it works right when integrating with your back-end infrastructure.

Today, we’ll be transforming the much-loved Flutter starter Counter app into a distributed app using Firebase Firestore.

# A Complete Guide to Setting Up Firebase in Flutter

Firebase, a comprehensive development platform provided by Google, offers a wide range of powerful tools and services for building mobile and web applications.

The process of setting up Firebase in Flutter, a popular framework for building cross-platform apps. By the end of this guide, you will have a solid understanding of how to integrate Firebase into your Flutter project and leverage its features for enhanced functionality.

**Prerequisites:** Before we dive into setting up Firebase in Flutter, make sure you have the following:

* Flutter SDK installed on your system
* A Flutter project created and ready for Firebase integration
* A Google account to create a Firebase project

**Creating a Firebase Project:** To get started, visit the Firebase website (firebase.google.com) and sign in with your Google account. Follow these steps to create a new Firebase project:

* Click on “Go to Console” or “Add project” button.
* Click on “Create a project.”
* Provide a project name and select your country/region.
* Enable Google Analytics if desired (optional).
* Click on “Create Project.”

**Adding Firebase to Your Flutter Project:** To integrate Firebase into your Flutter project, you need to add the Firebase packages to your pubspec.yaml file. Open your Flutter project in your preferred code editor, locate the pubspec.yaml file, and add the following dependencies:

dependencies:

firebase\_core: ^3.9.0

  firebase\_database: ^11.2.0

  cloud\_firestore: ^5.6.0

Save the file and run flutter pub get in the terminal to download the packages.

**Configuring Firebase for Android:** To configure Firebase for your Flutter project on Android, follow these steps:

* Go back to the Firebase console and select your project.
* Click on the “Android” icon to add an Android app.
* Provide the package name of your Flutter app (e.g., com.example.myapp).
* Download the google-services.json file.
* Move the downloaded google-services.json file to the android/app directory of your Flutter project.
* Open the android/build.gradle file and add the following code in the build.gradle

buildscript {

ext.kotlin\_version = '1.7.10'

repositories {

google()

mavenCentral()

}

dependencies {

classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin\_version"

classpath 'com.google.gms:google-services:4.3.15'

}

}

* Open the android/app/build.gradle file and add the following plugins at the bottom of the file:

plugins {

id "com.android.application"

id "kotlin-android"

// The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.

id "dev.flutter.flutter-gradle-plugin"

id 'com.google.gms.google-services'(add this line in plugin)

}

* Change main() code into this code

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

runApp(MyApp());

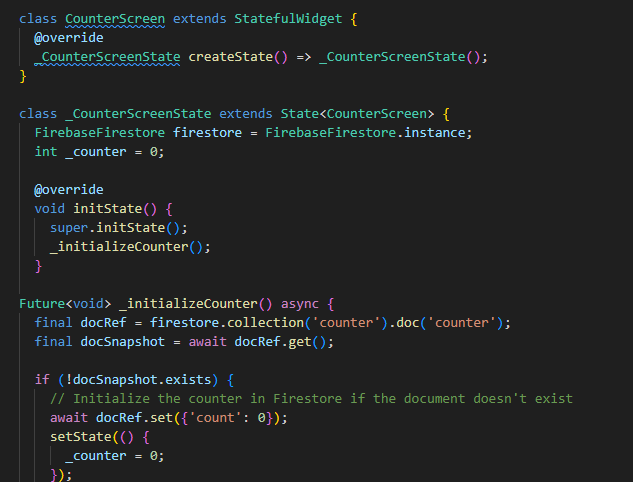
}

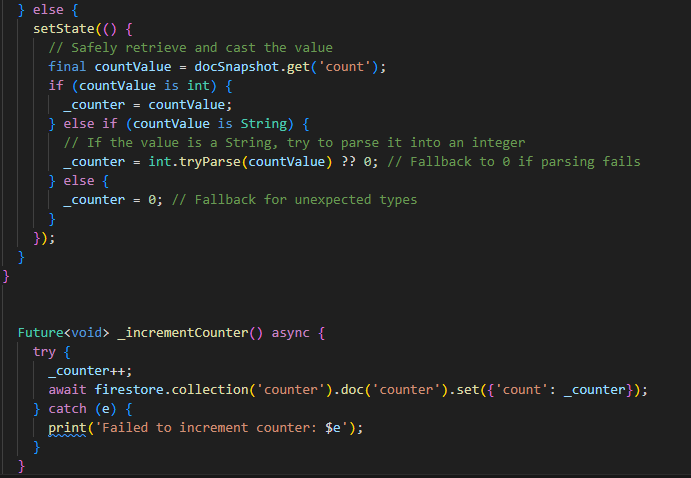
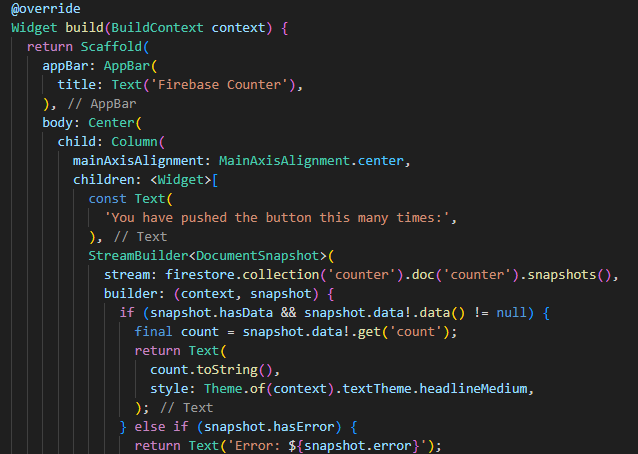
* Save the files, and Android configuration is complete.

Now run the code with android Emulator.

**Firestore Implementation of Starter Counter App**

We need to build some actual calls to Firestore. To do that, simply replace the entire \_CounterScreen class, so that it stores and retrieves our counter value from Firestore, rather than in-memory.





**Assessment:**

1. What is Firebase? Explain its core purpose.
2. List at least three services provided by Firebase.
3. How do you initialize Firebase in a Flutter application?
4. How do you set up and use Firebase Cloud Firestore?