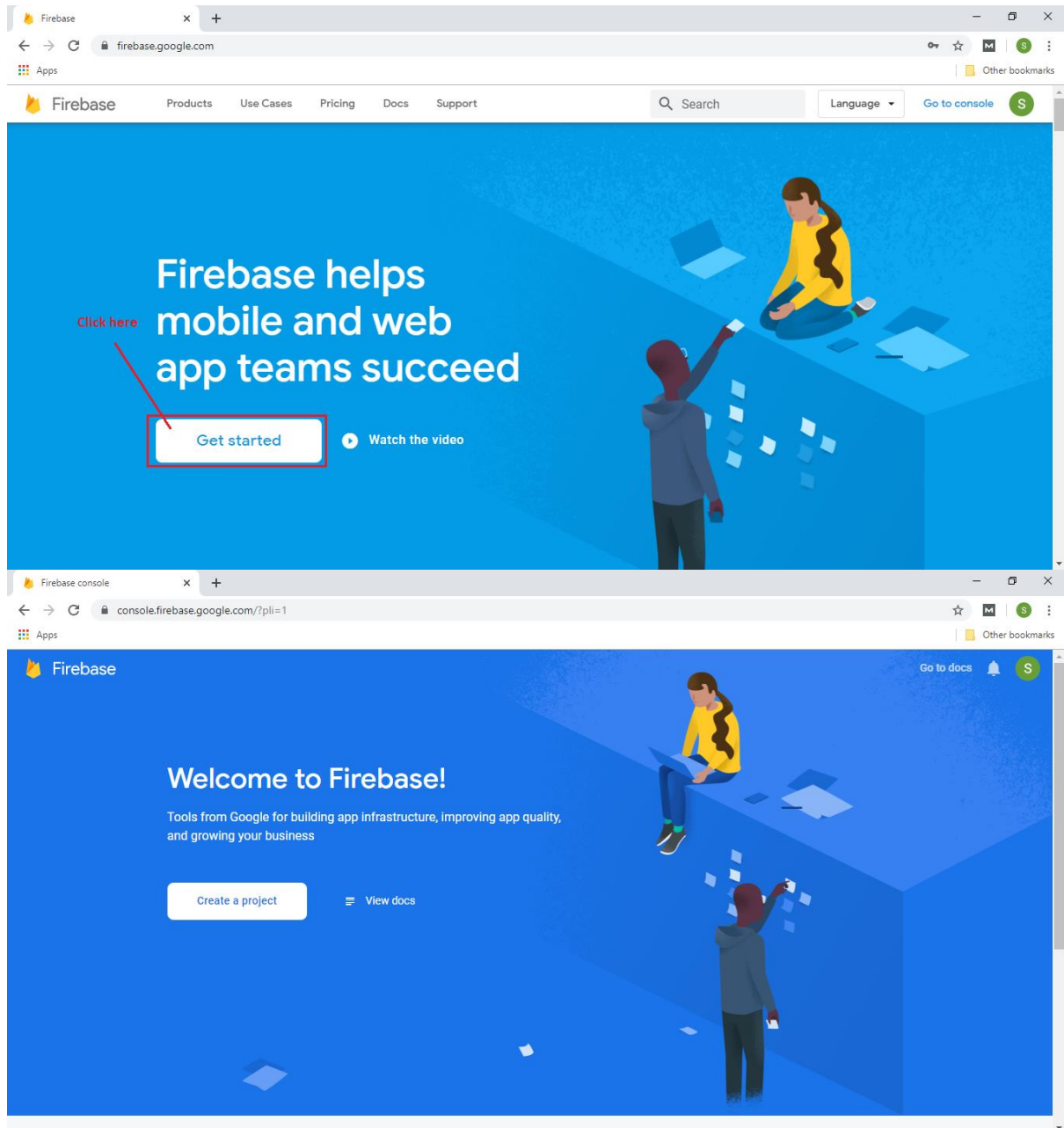


Step By Step Guide for Setting Firebase First Time

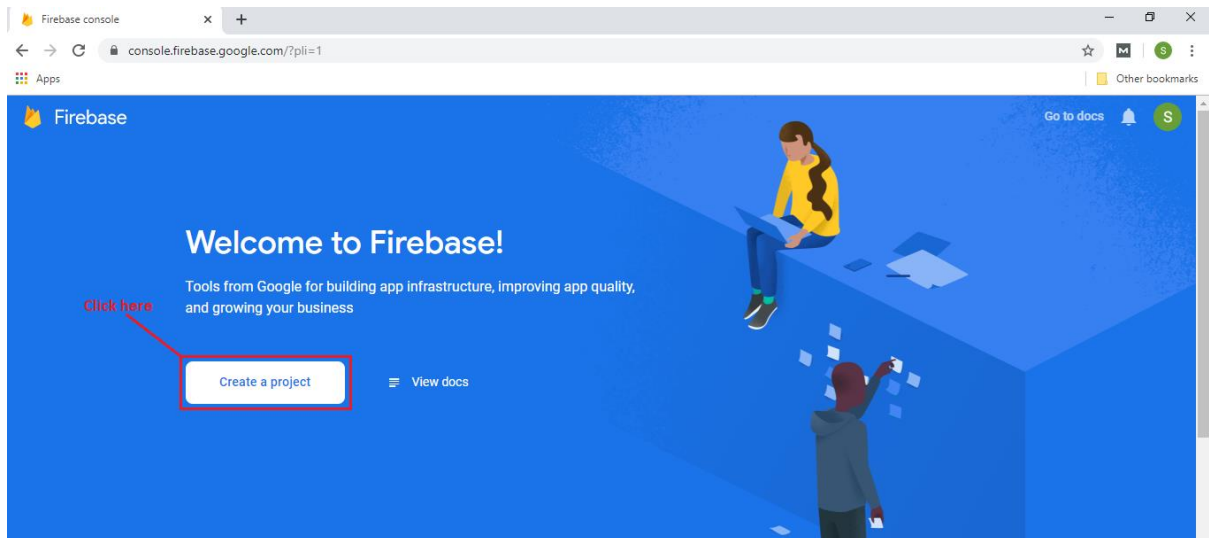
Step 1:

In the first step, we have to go to the firebase console by using the following link <https://firebase.google.com/>



Step 2:

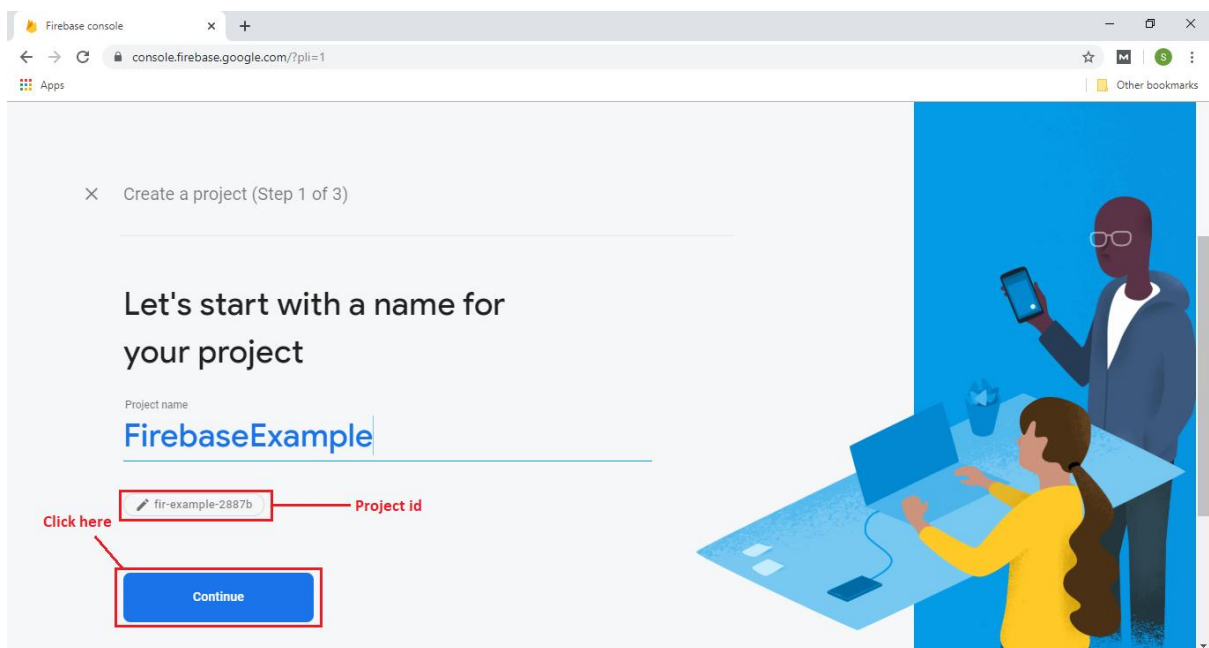
Now, we first have to create a Firebase project. Firebase project will go through and describe what it is. This allows us to add an app to that project can have many projects inside of Firebase.



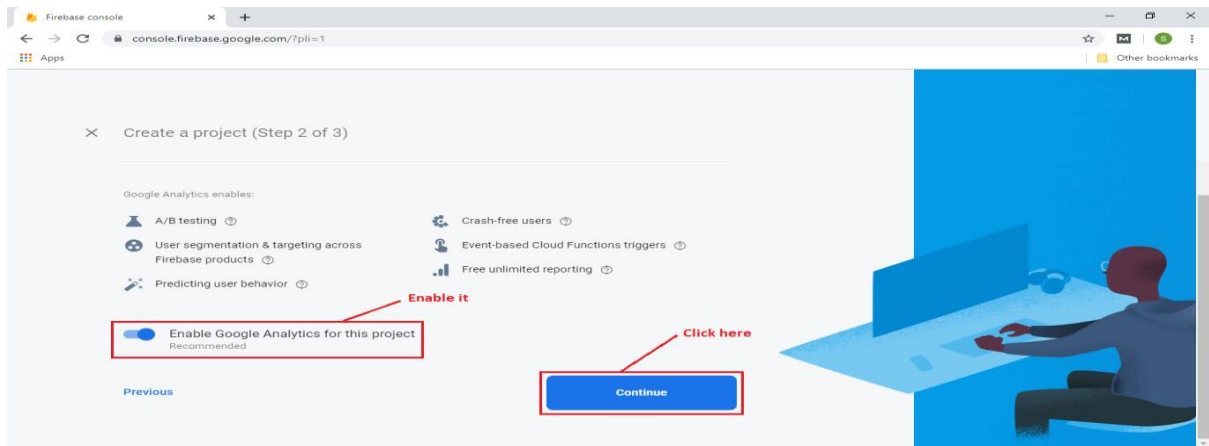
Step 3:

Now, we have to give a name and id to the project, enable Google Analytics for the project, set an analytics location, and accept all the terms and conditions. Firebase console automatically creates a unique id for the project.

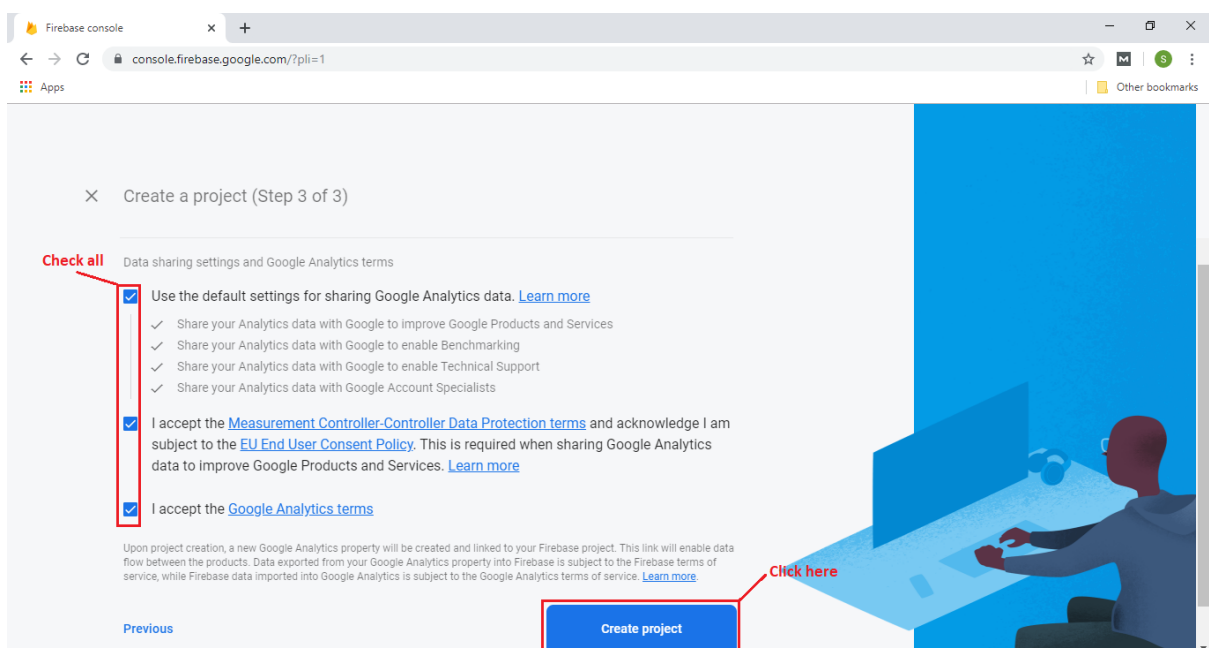
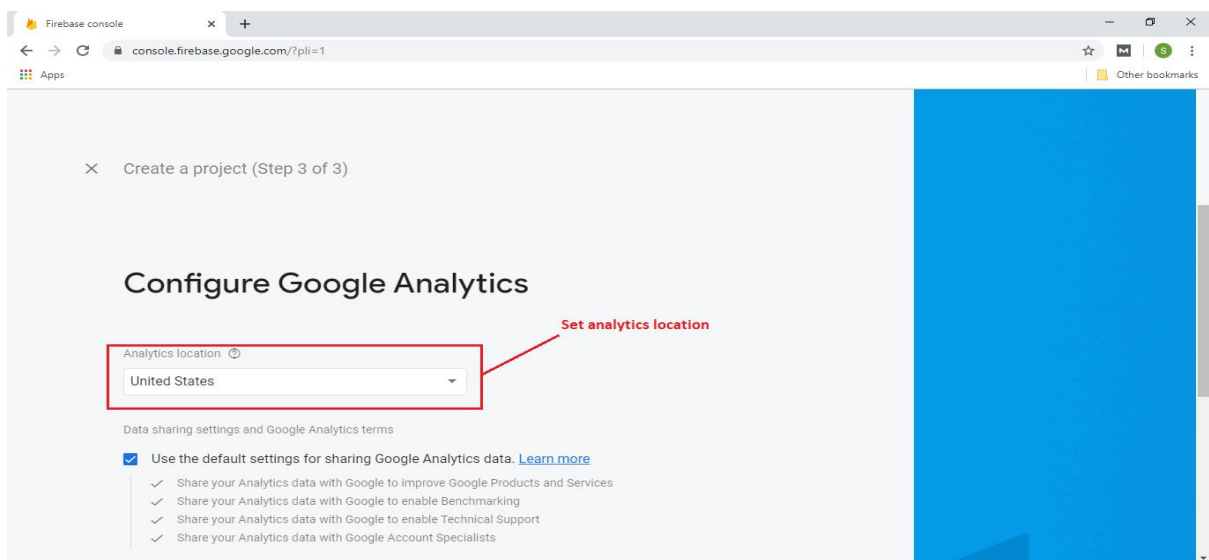
1) Giving a name to the project



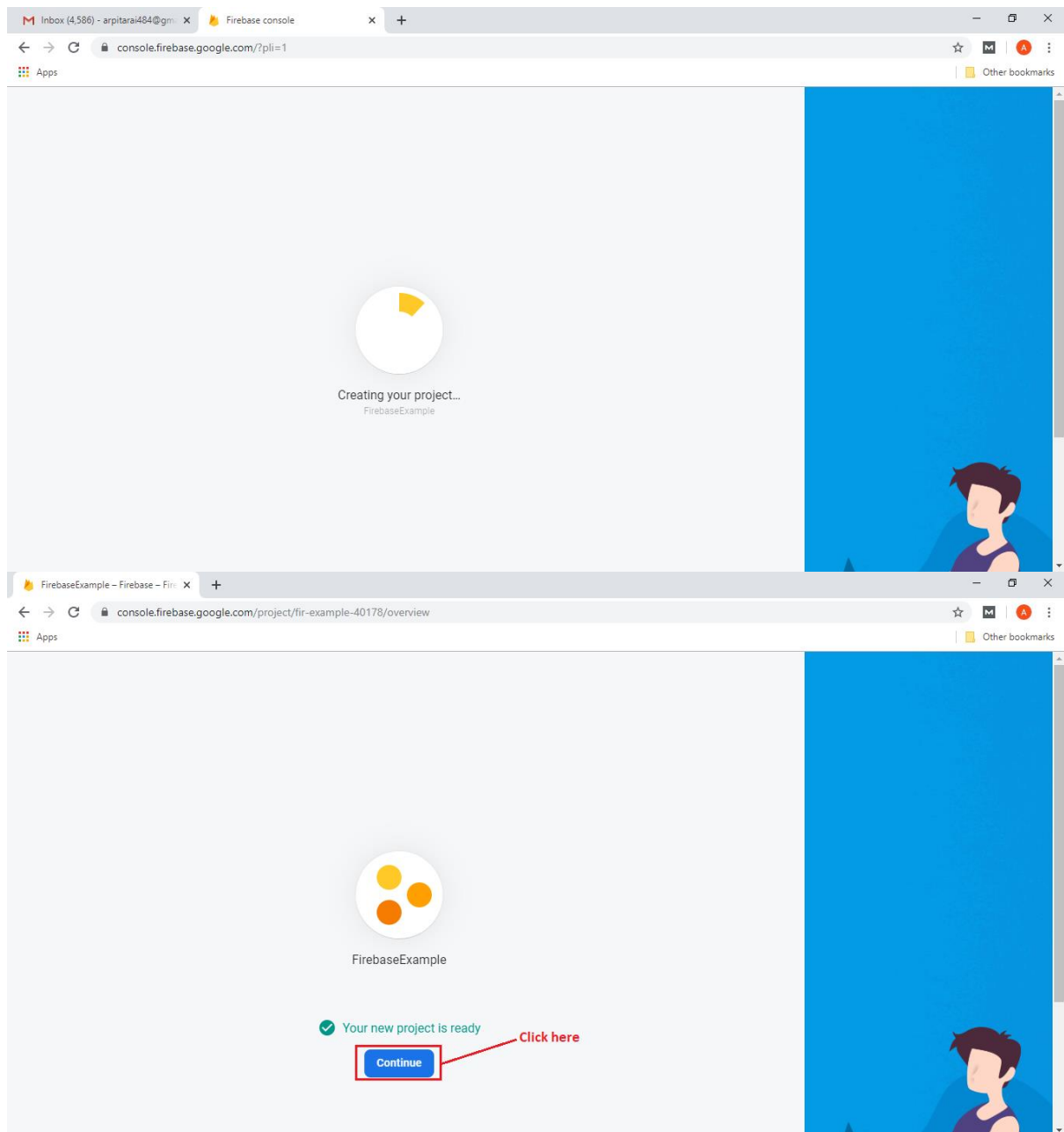
2) Enable Google analytics for the project



3) Set analytics location and accept terms and condition

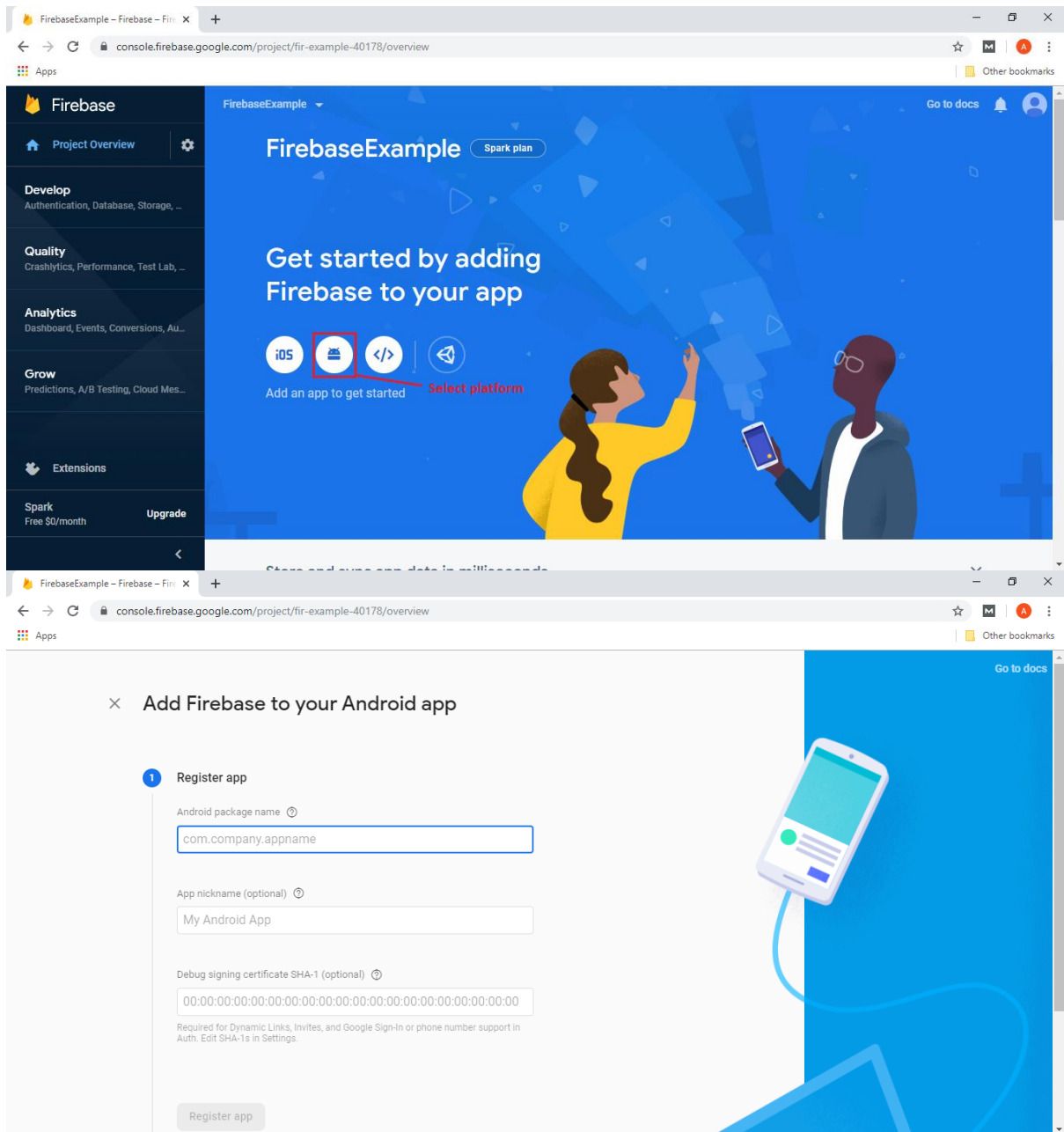


When we click on create the project, the creation of the Firebase project is started, and once our project will be created, we have to click on continue to enter in our project.



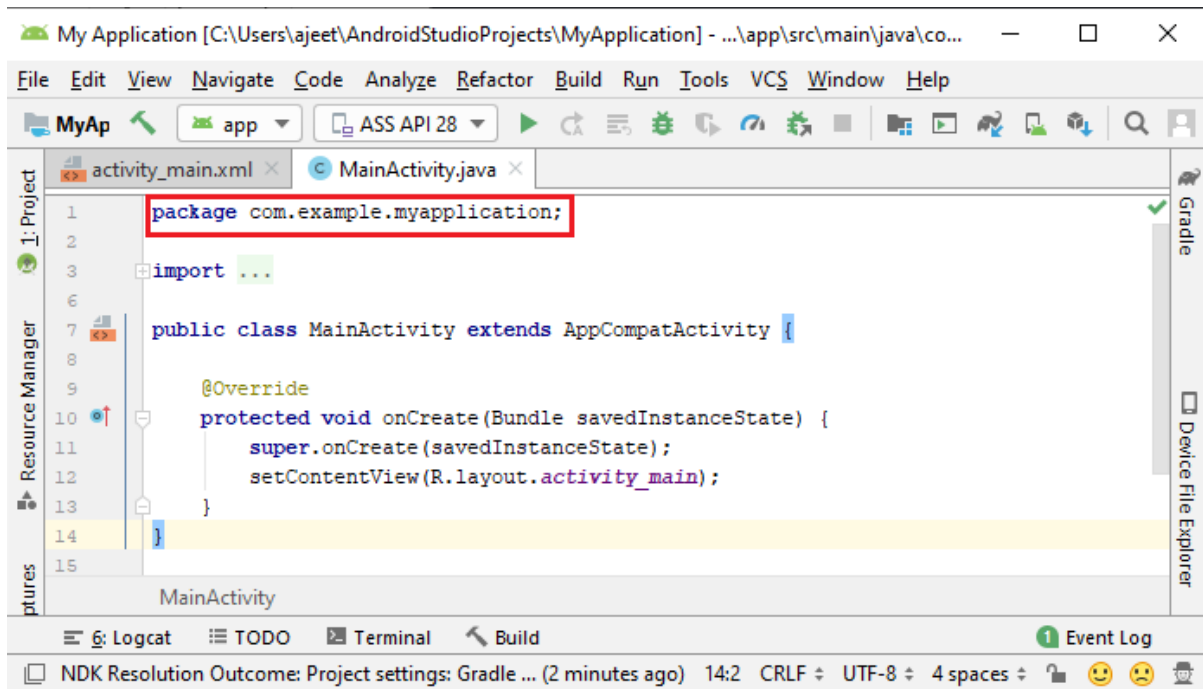
Step 4:

In the next step, we have to choose the platform to add Firebase to our application. The platform can be Android, iOS, Web, and Unity.



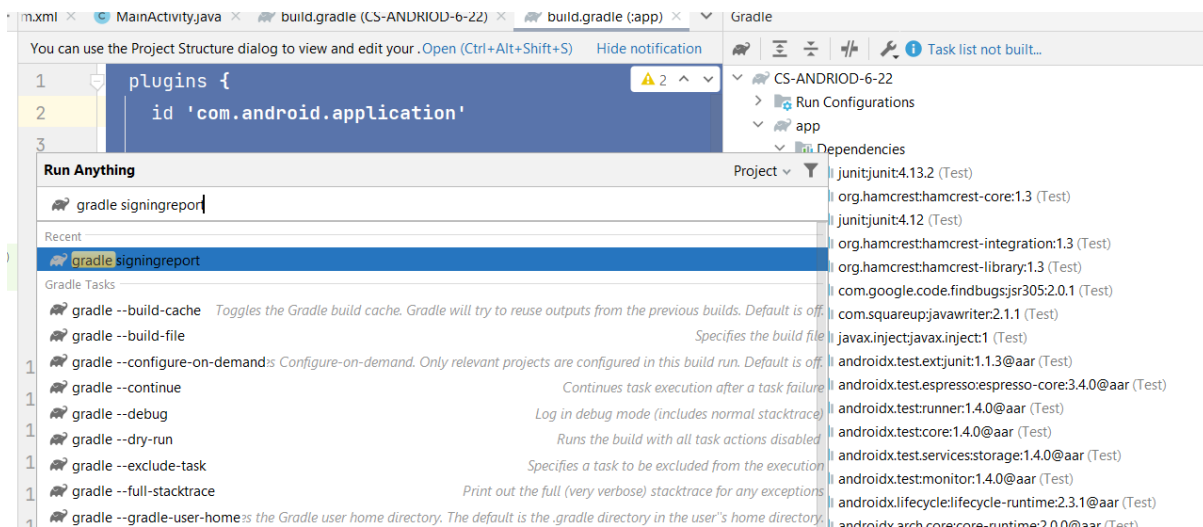
Step 5:

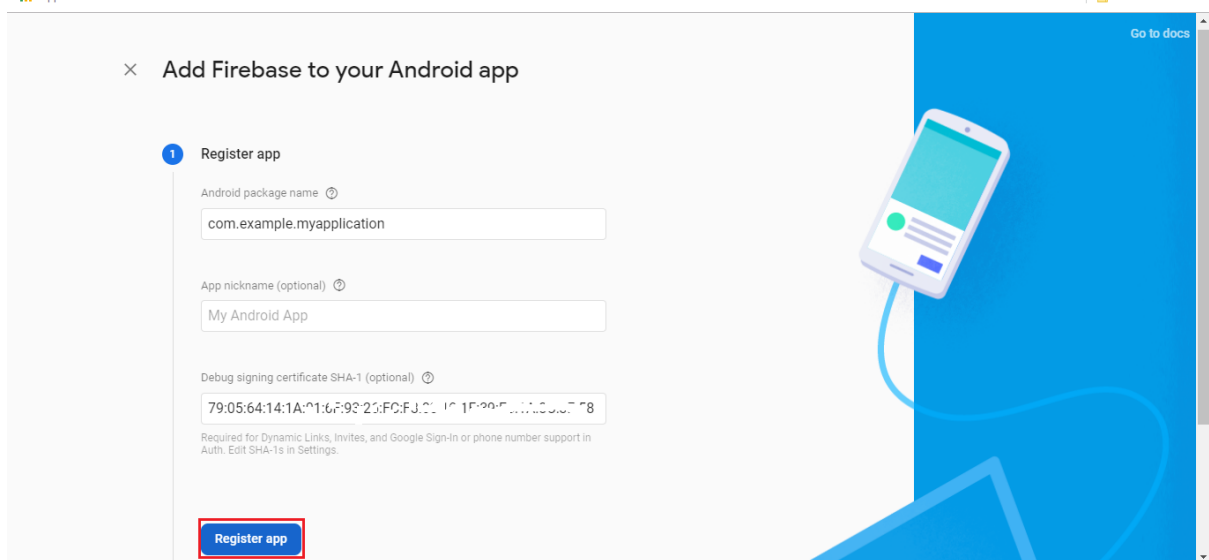
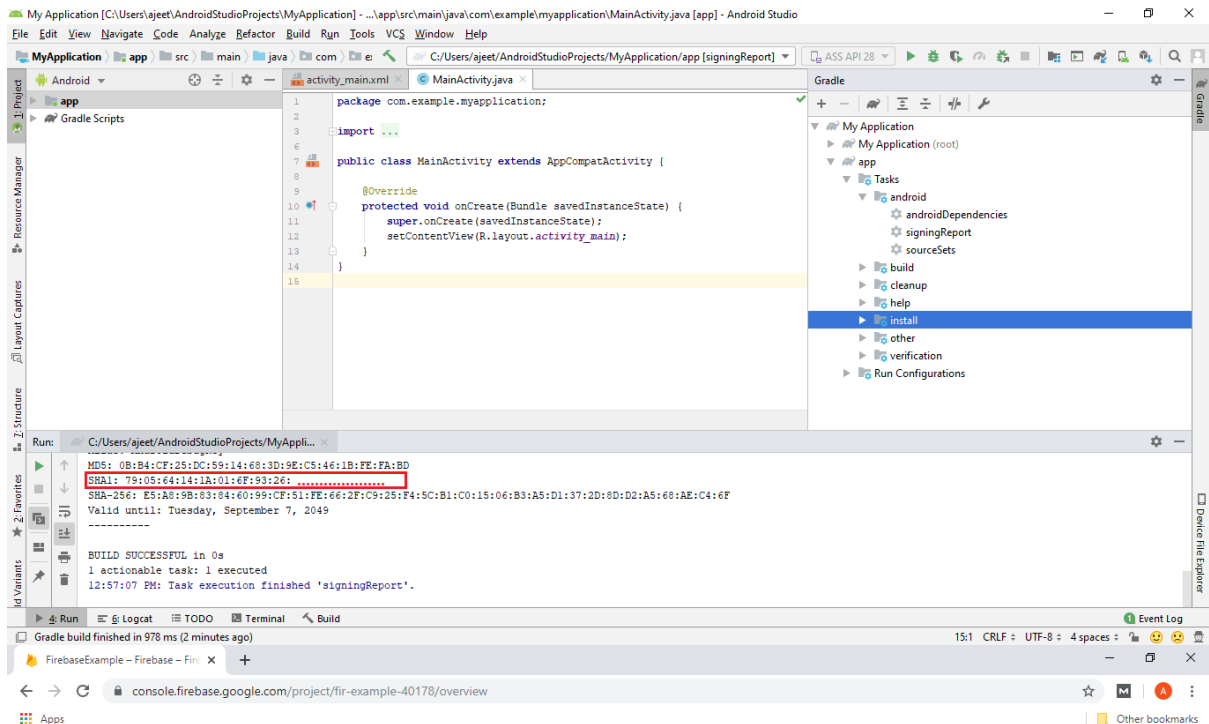
In the next step, we have to mention the package name, and the package name is the mandatory and important thing. The package name should be matched with our Android applications package name. We can find the package name of our application in any of the Java class or Manifest file.



Step 6:

Type **signingreport** in Gridle

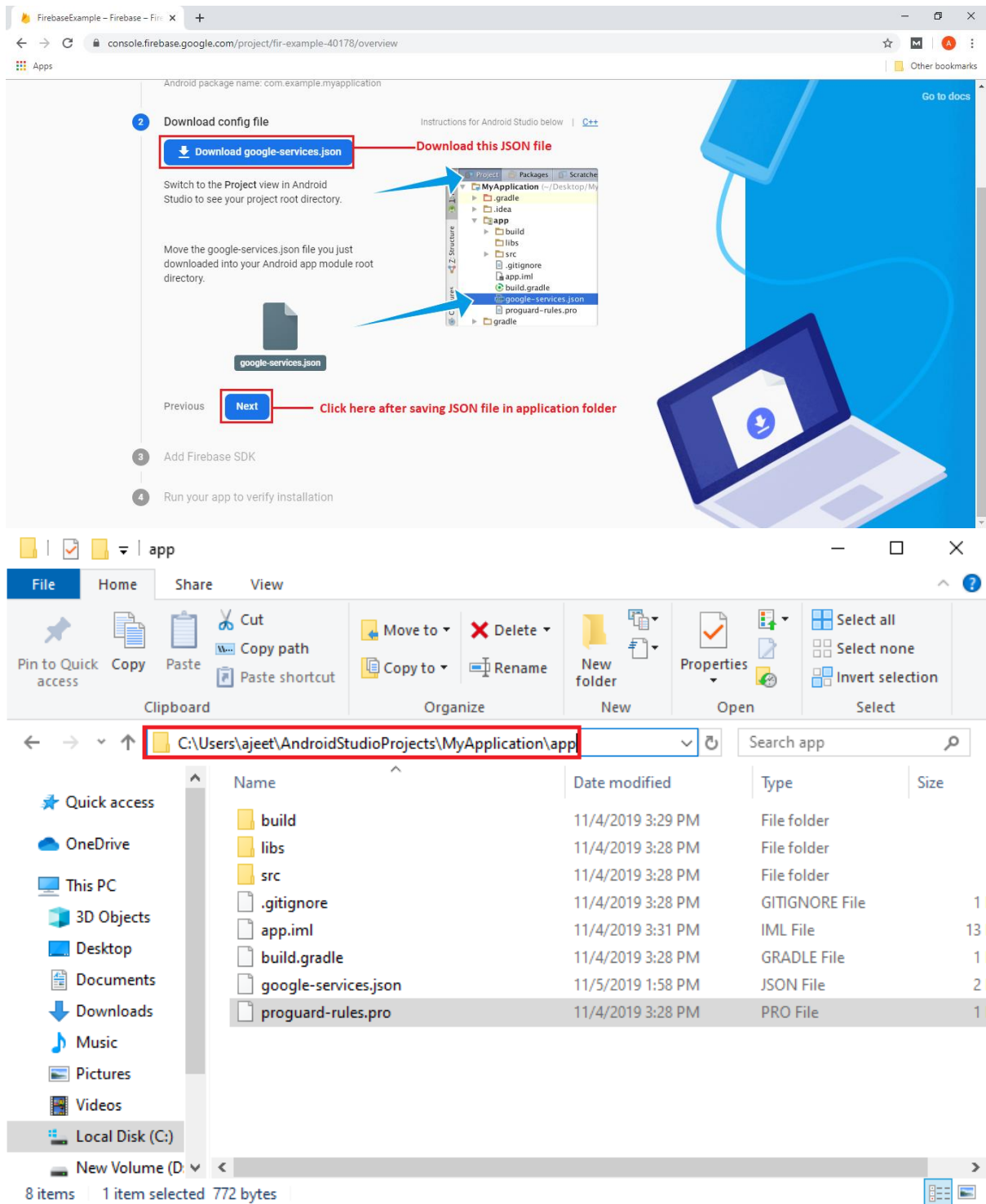




When we click on the **Register app**, we will go to the next field i.e., **Download config file**.

Step 7:

It is one of the most important steps. In this step, we have to download the `google-services.json` file. This file is provided by Firebase for us to put it inside our application folder in our application. This is a necessary file which is needed by our application to connect with Firebase.



When we click on next, we will switch to the next field i.e., Add **Firestore** SDK.

Step 8:

In the next step, we will add the Firebase SDK. This has been done inside the Android Studio. We have to modify the build. Gradle files once in your project with the following lines of code.

The first screenshot shows the 'Project-level build.gradle' configuration. It includes the Google services plugin and the Google's Maven repository. The code is as follows:

```
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:4.3.2'
    }
}

allprojects {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
}
```

The second screenshot shows the 'App-level build.gradle' configuration. It includes the Firebase SDK for Google Analytics and the Google services plugin. The code is as follows:

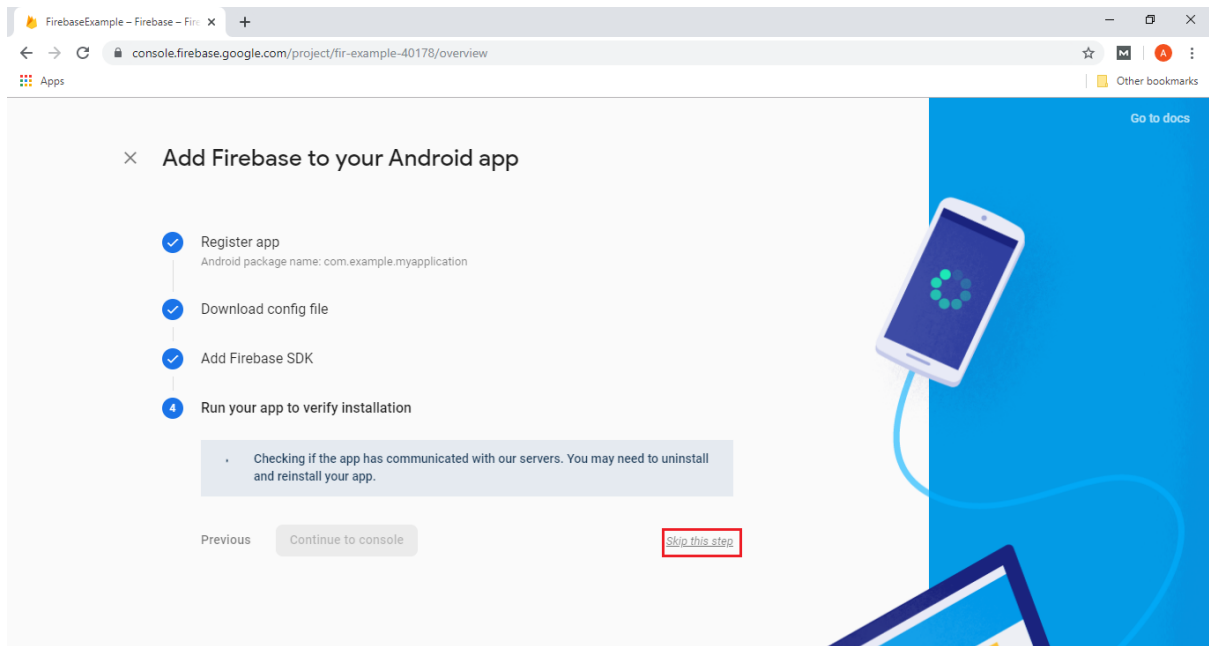
```
apply plugin: 'com.android.application'

dependencies {
    // add the Firebase SDK for Google Analytics
    implementation 'com.google.firebase:firebase-analytics:17.2.0'
    // add SDKs for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}

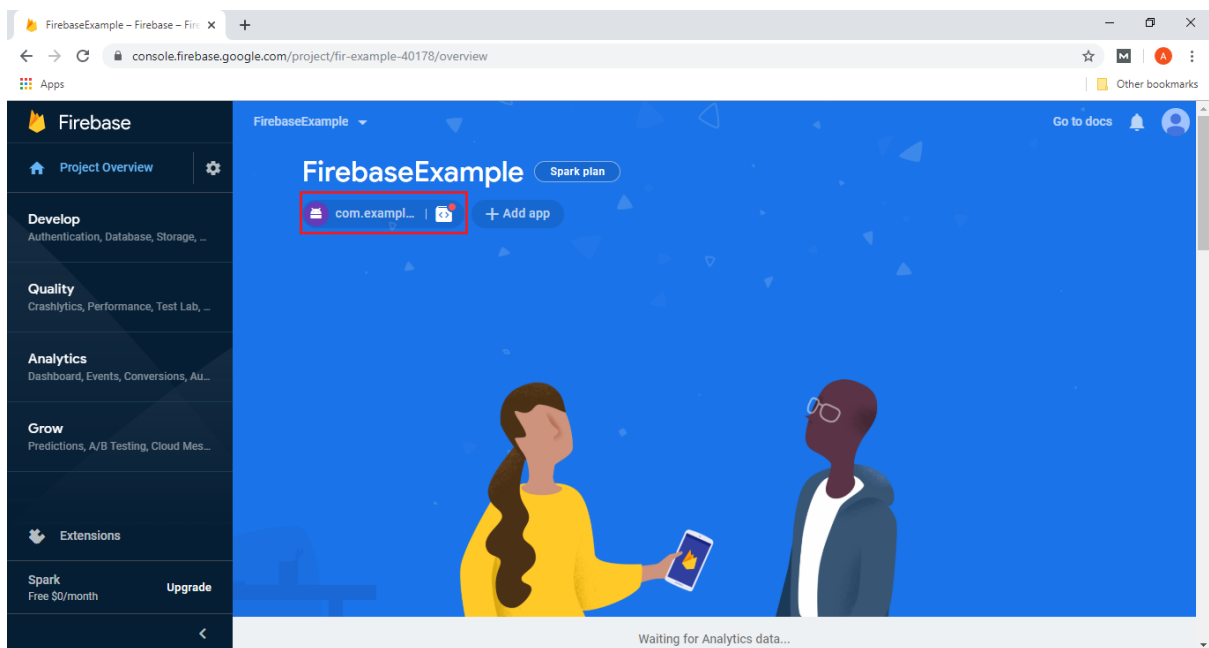
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Below the code, there is a 'Sync now' button and a 'Next' button. A red arrow points to the 'Next' button with the text 'Click here'.

We will skip the four-field, i.e., **Run your app to verify the installation.**



After clicking on the skip, we will switch to the overview page where we can see our application.



When our application runs on the device, we will check the Firebase console to ensure that our application is communicating with Firebase or not.

