

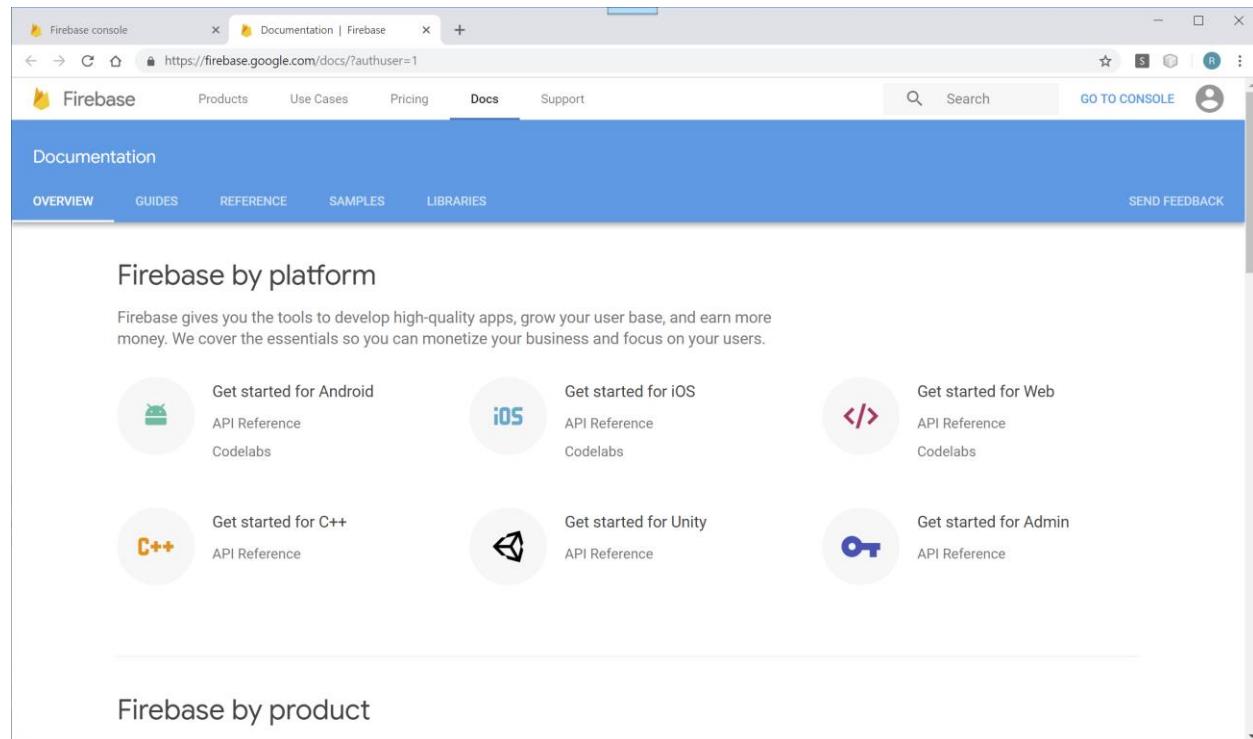
We will be using two core features of firebase. Their real time database and their authentication feature.

Our data will consist of chat message and these will be retrievable on all the devices that run the chat app. The users will have to provide their unique email and password combination on the registration screen, and we use that information to authenticate users in firebase and log them into the app where they can join the chat.

The first thing we need to do is to go to <https://firebase.google.com> and set up a new firebase account.

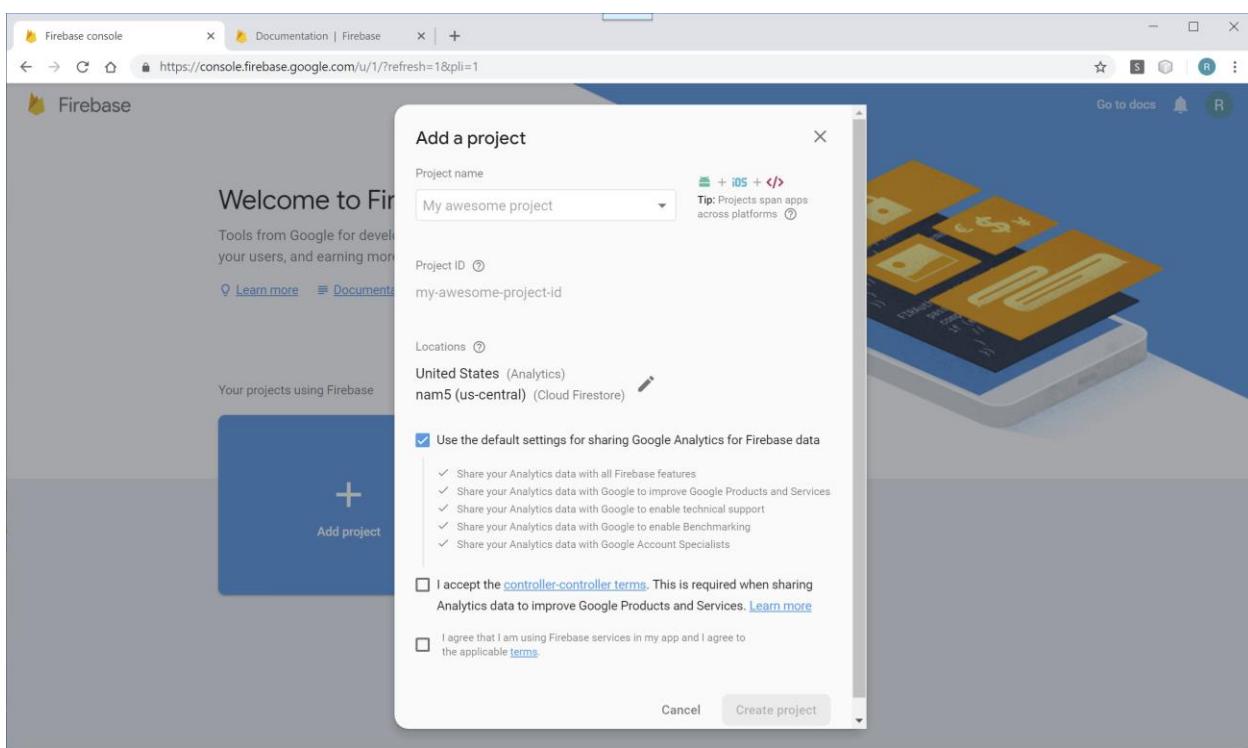
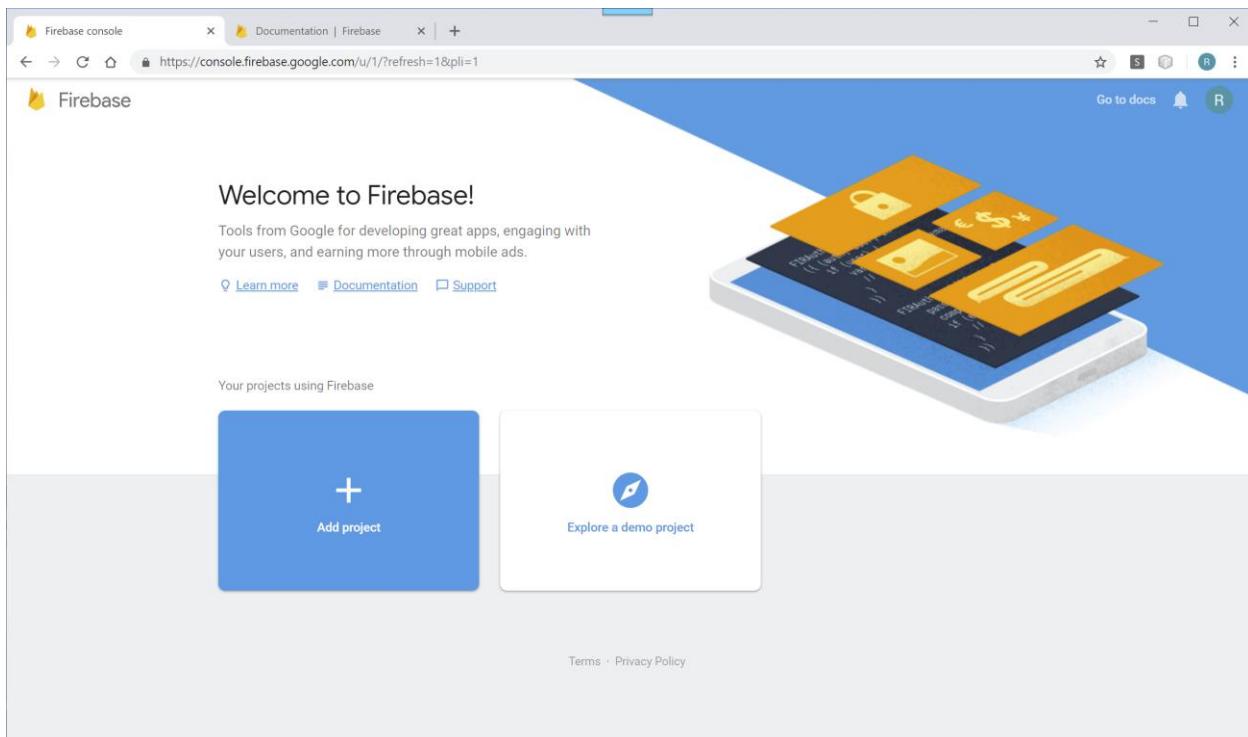
Once you are on the firebase web site, click SIGN IN and you can sign in with your Google account. This is the same account that you use for Gmail or any of the other Google Apps.

In the unlikely case that you don't have a Google account, go ahead and create an account using the link below. Once you are logged in, go to your console and that basically is your dashboard which is located at console.firebaseio.google.com.

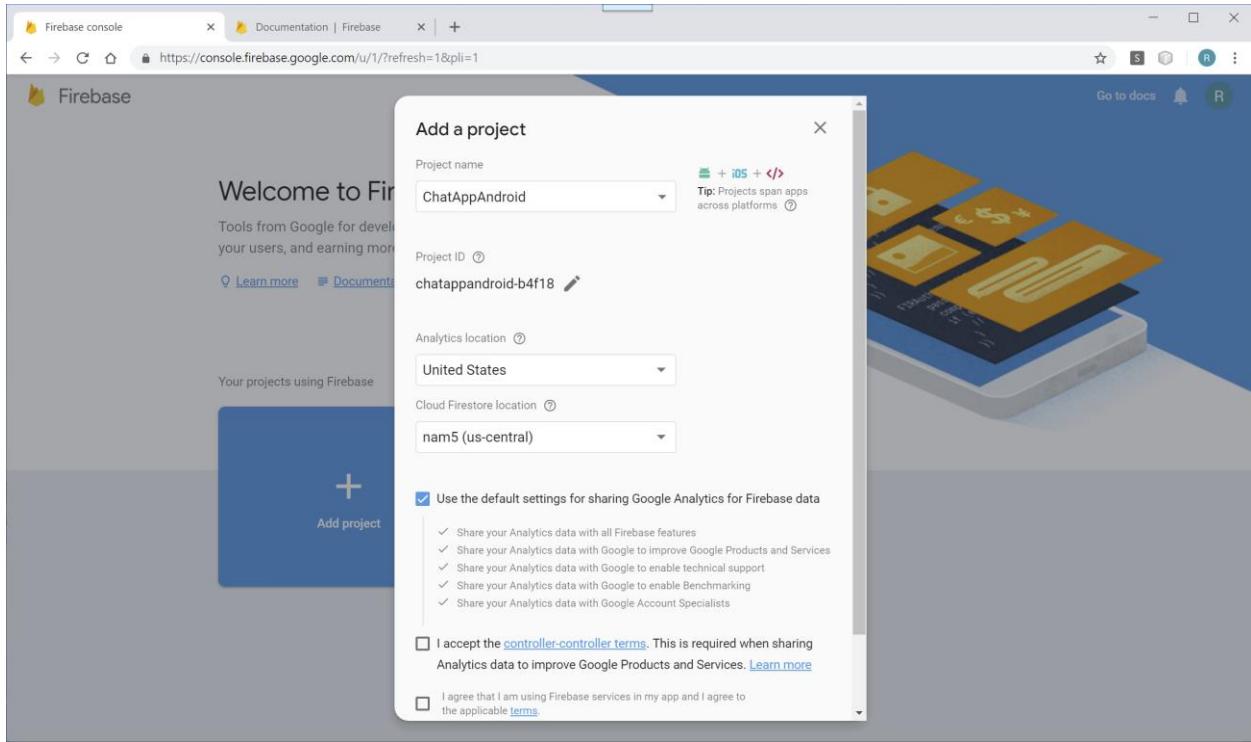


The screenshot shows the Firebase Documentation website (<https://firebase.google.com/docs/?authuser=1>). The top navigation bar includes links for Products, Use Cases, Pricing, Docs (which is selected), and Support. Below the navigation is a search bar and a 'GO TO CONSOLE' button. The main content area is titled 'Documentation' and features a blue header with 'OVERVIEW', 'GUIDES', 'REFERENCE', 'SAMPLES', and 'LIBRARIES' tabs, along with a 'SEND FEEDBACK' button. The main section is titled 'Firebase by platform' and lists six categories: 'Get started for Android' (with icons for Android, API Reference, and Codelabs), 'Get started for iOS' (with icons for iOS, API Reference, and Codelabs), 'Get started for Web' (with icons for a browser, API Reference, and Codelabs), 'Get started for C++' (with icons for C++, API Reference, and Codelabs), 'Get started for Unity' (with icons for Unity, API Reference, and Codelabs), and 'Get started for Admin' (with icons for a key, API Reference, and Codelabs). At the bottom of this section is a horizontal line followed by the heading 'Firebase by product'.

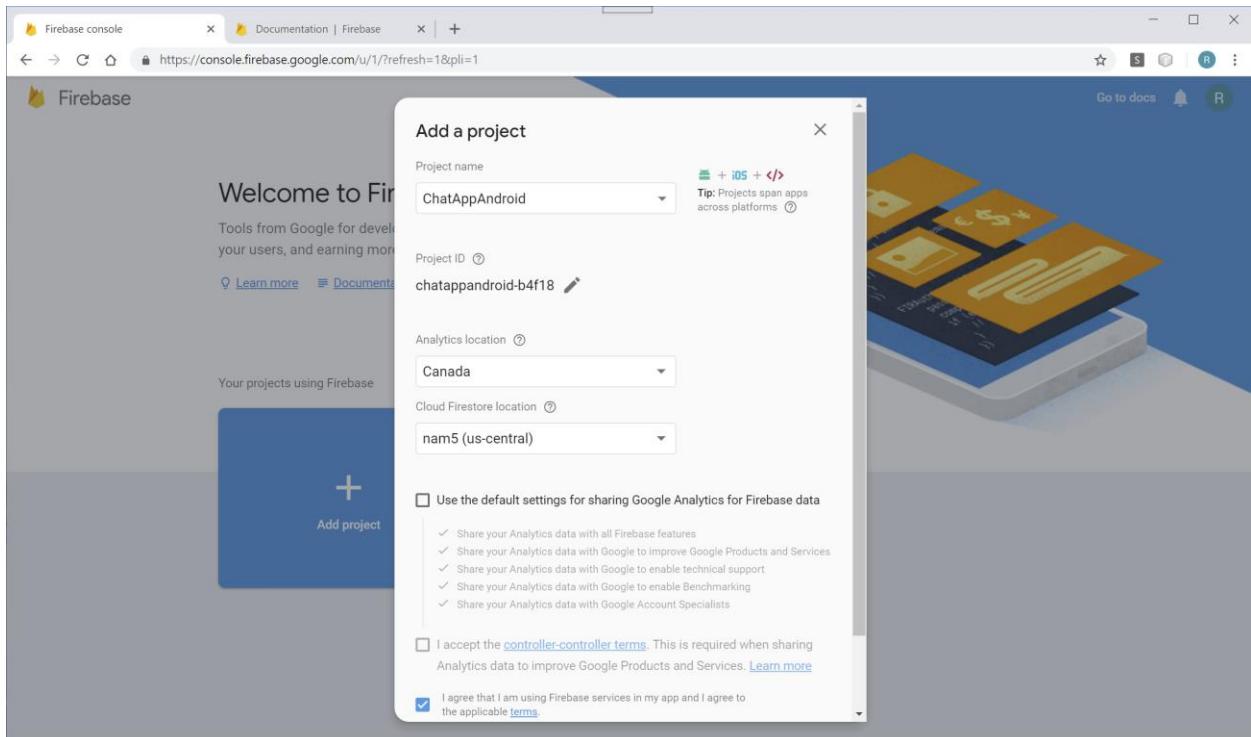
On your console you can see all your projects listed. Add a new project for your chat app by clicking Add Project and you will be presented with a wizard as shown:



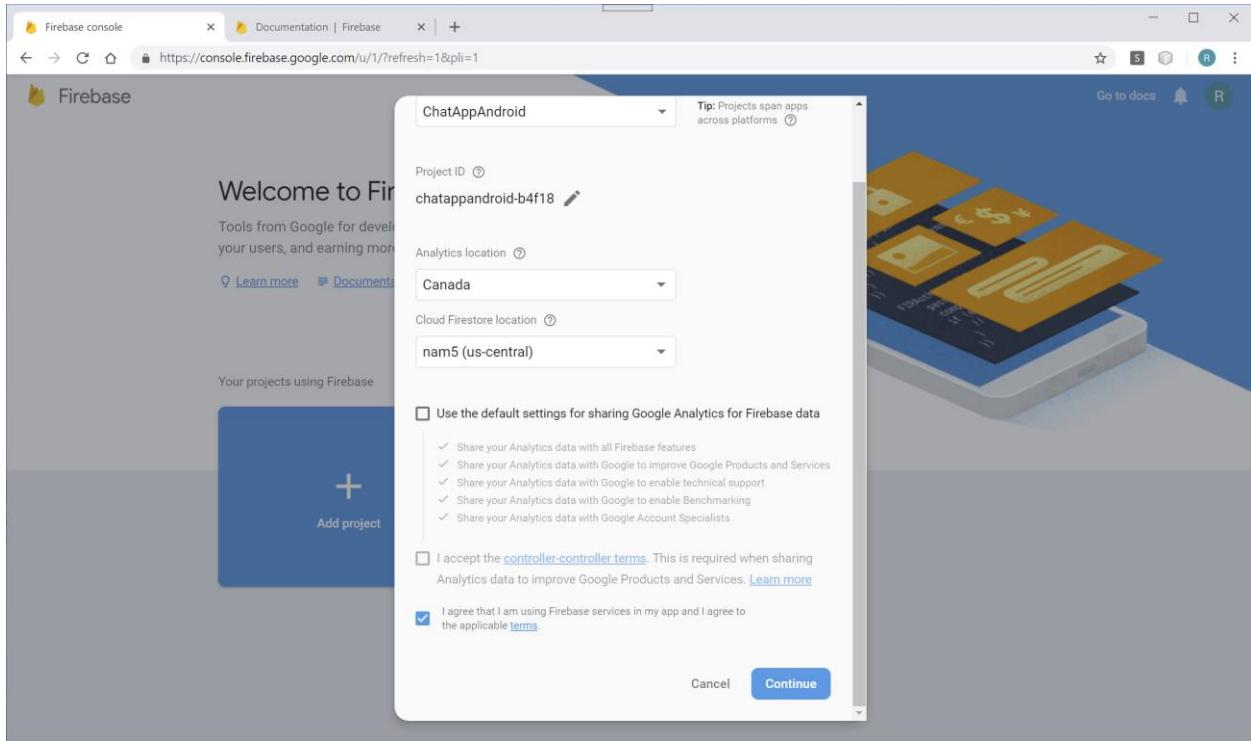
Enter a name for your project and click on the pen icon at the locations section as shown below:



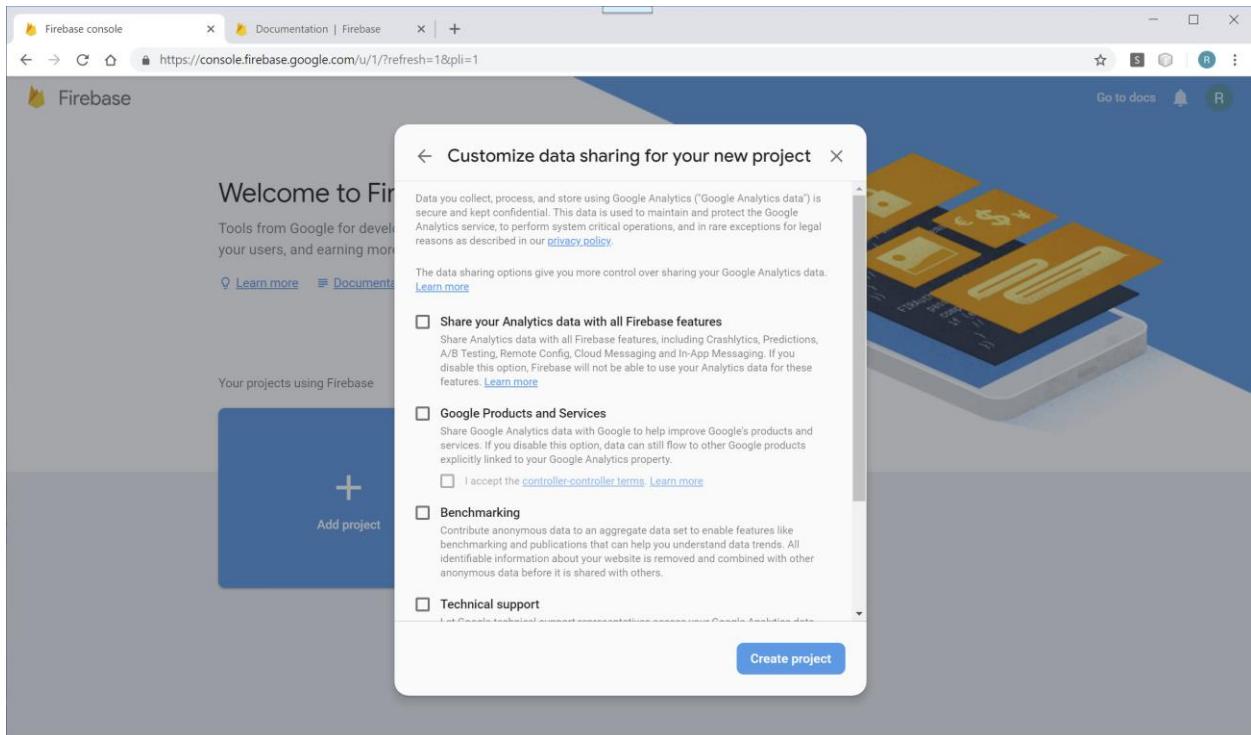
Then change the country to where you are located. For us that will be Canada and then click the checkbox to agree with the Firebase terms as shown:

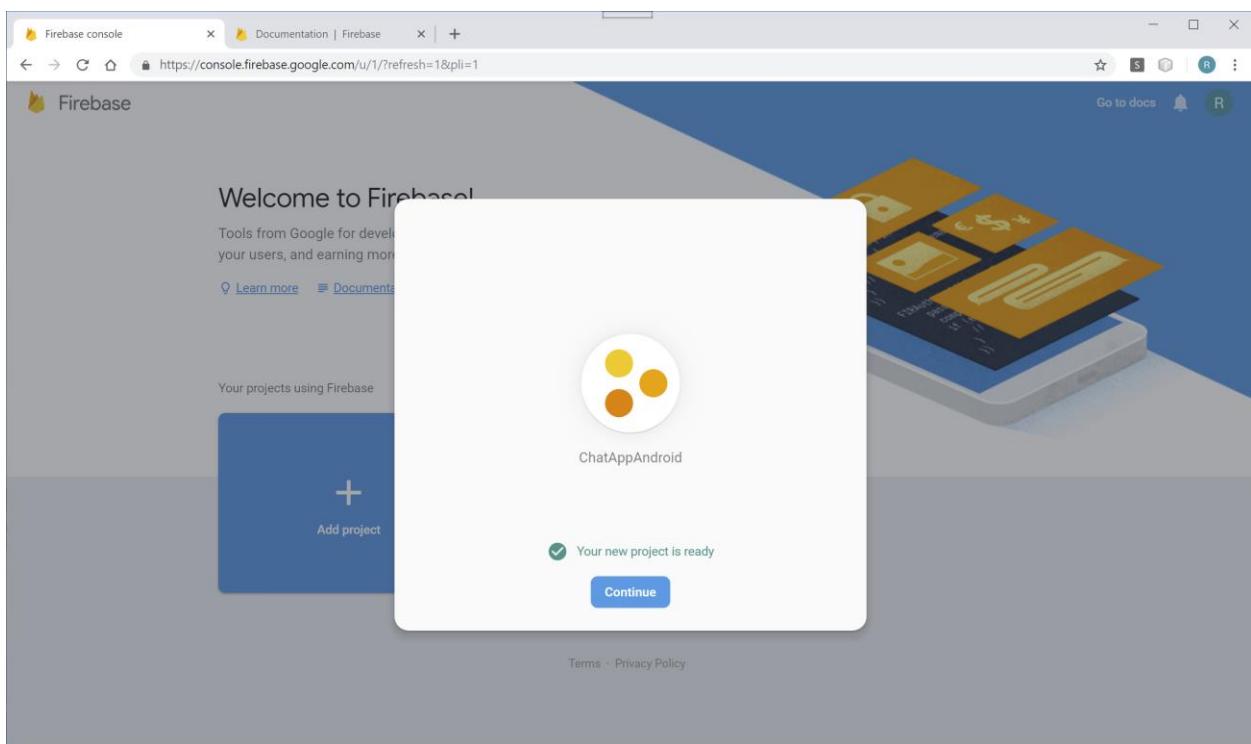
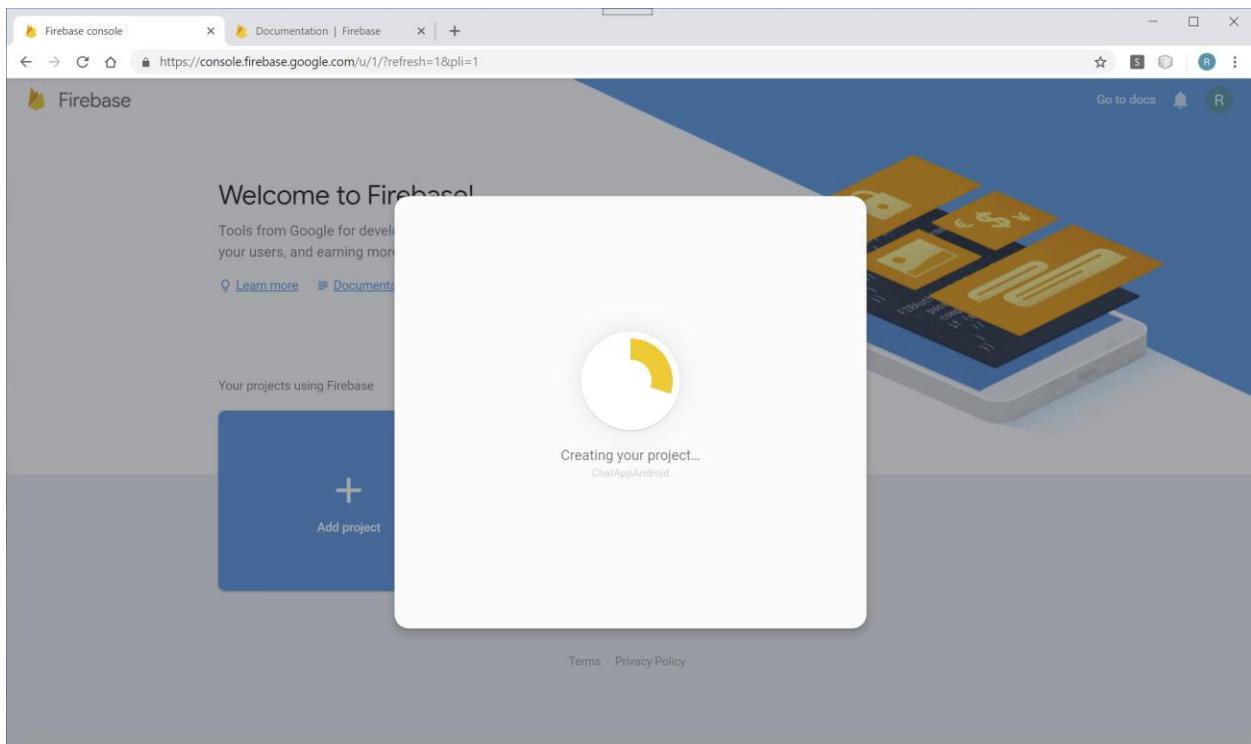


And then click Continue as shown:

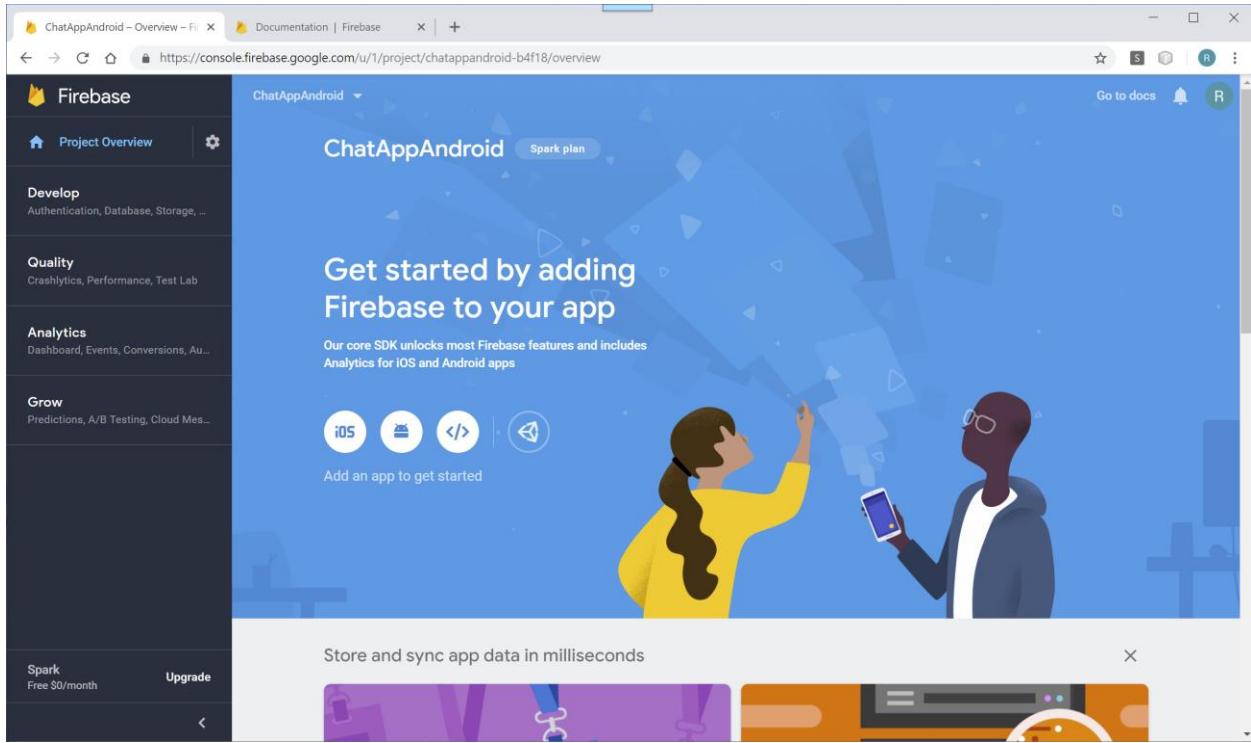


Click Create Project in the next window as shown:

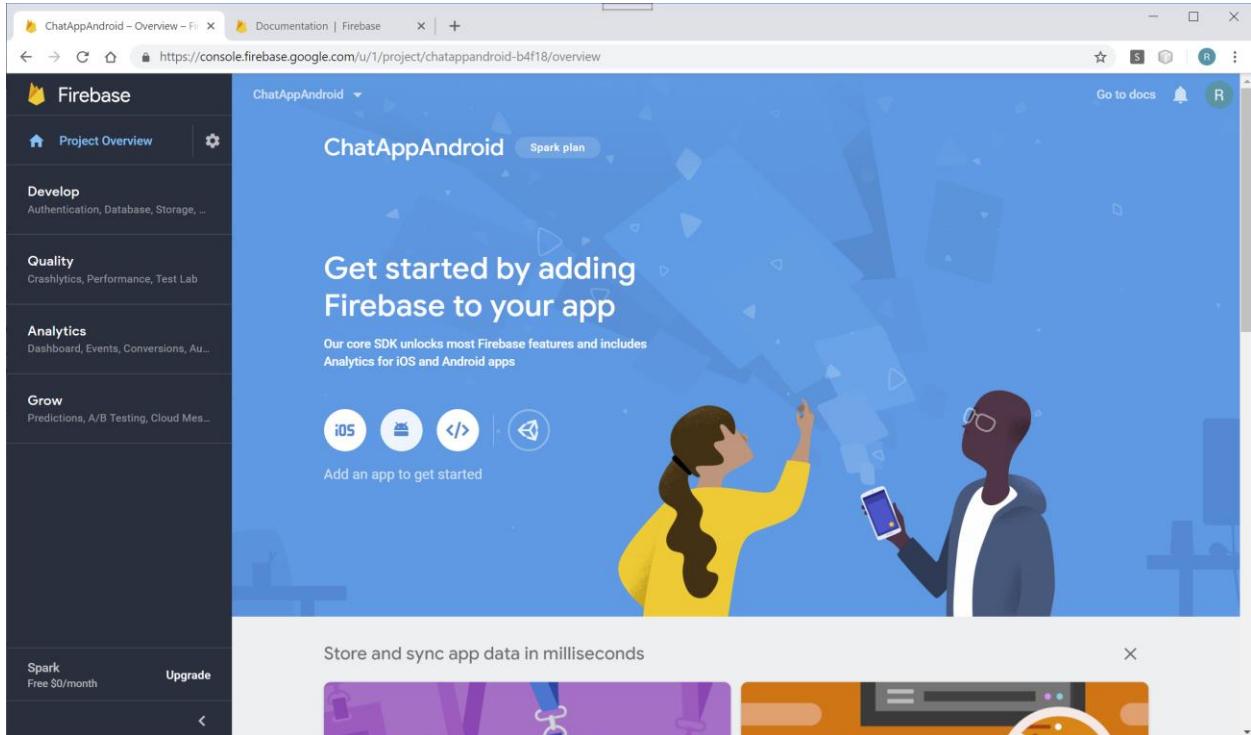


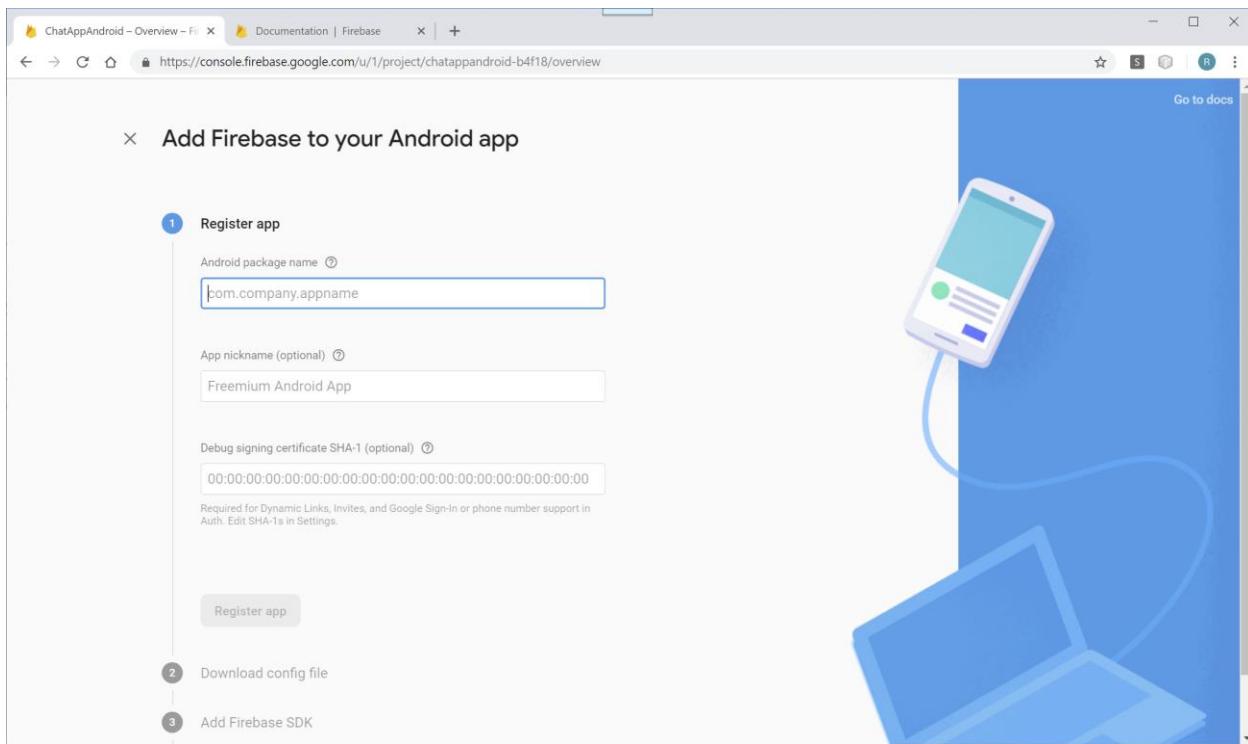


Click Continue and you are at the following screen:



At this point we have successfully created a firebase project. All we have to do is to add Firebase to our app. Since we are making an Android app, click on the Android icon above the text “Add an app to get started” as shown (that’s the icon next to iOS icon):





In “Add Firebase to your Android app” window, we need to enter our package name. You can find the package name for your app in Android Studio manifest file listed at the top as shown:

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.chatapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Chat App"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

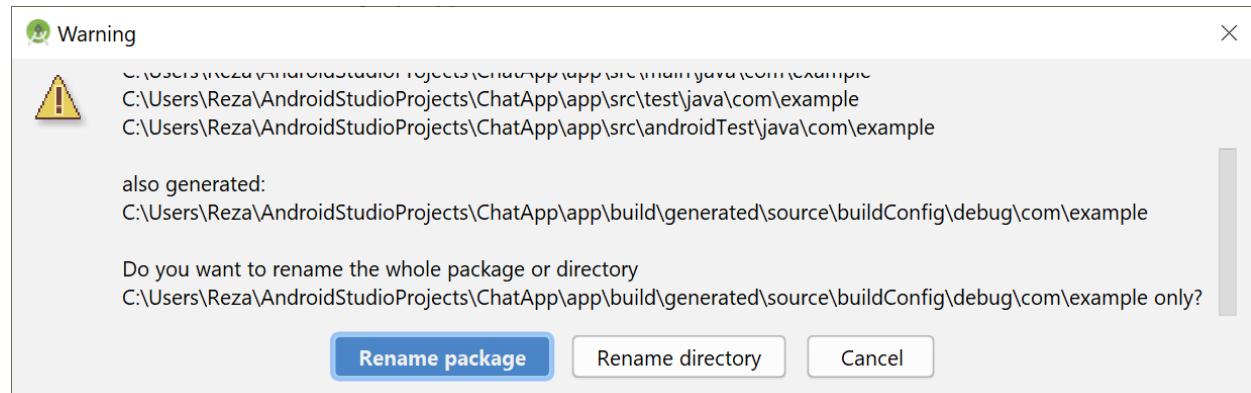
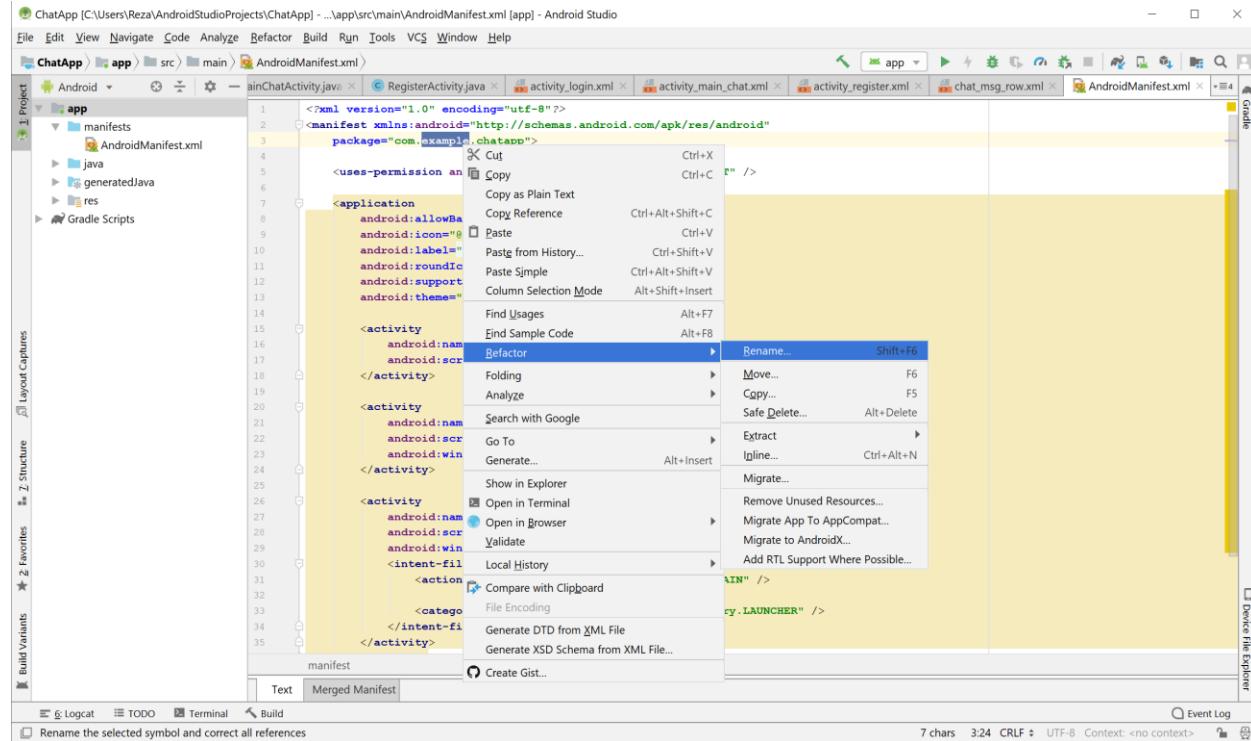
        <activity
            android:name=".RegisterActivity"
            android:screenOrientation="portrait">
        </activity>

        <activity
            android:name=".MainChatActivity"
            android:screenOrientation="portrait"
            android:windowSoftInputMode="stateHidden|adjustResize">
        </activity>

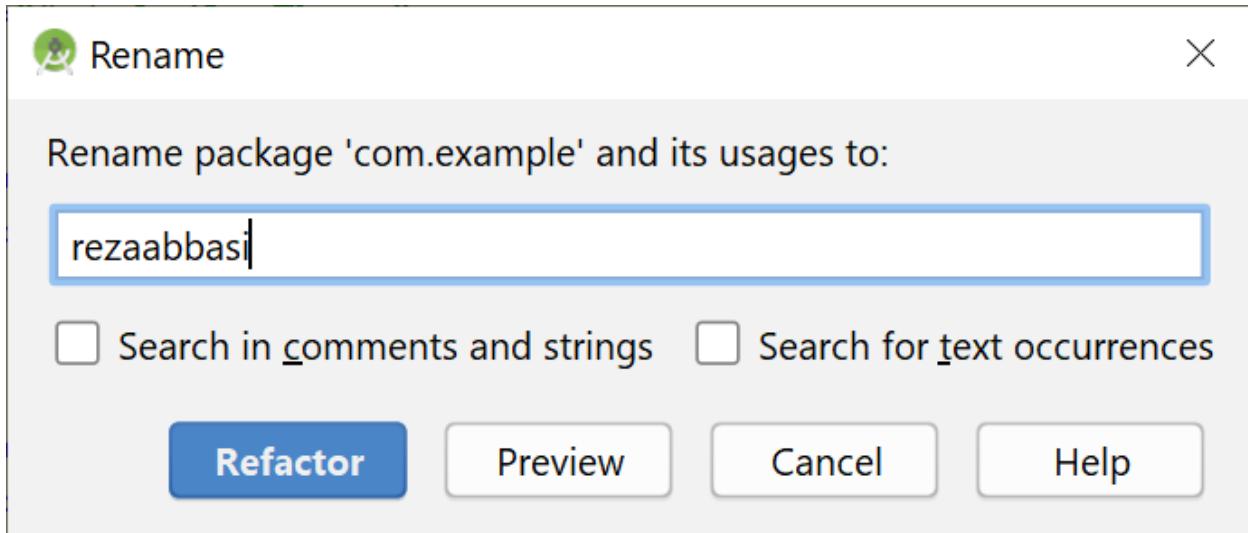
        <activity
            android:name=".LoginActivity"
            android:screenOrientation="portrait"
            android:windowSoftInputMode="stateHidden|adjustPan">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

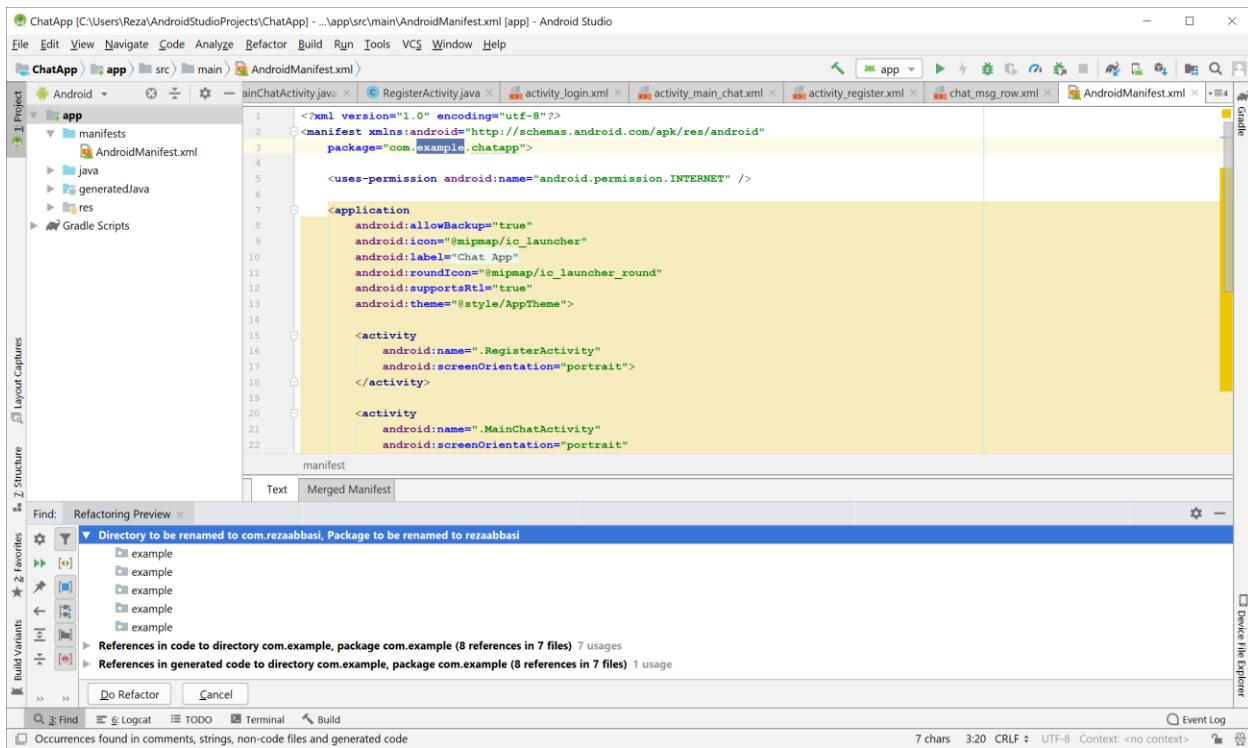
Here my package name is “com.example.chatapp” which doesn’t seem unique. You want to make this package name unique to something like your full name or a domain name that you own. Highlight the word example between “com” and “chatapp” and right click on it and choose Refactor and then Rename it as shown:



In the prompt, click the Rename package button and to make this package name unique enter your full name or a domain name you own.



Click Refactor and Android will show us a preview of all the refactoring changes. Click “Do Refactor” as shown:



The new package name in my case reads “com.rezaabbasi.chatapp” as shown:

The screenshot shows the Android Studio interface with the project 'ChatApp' open. The manifest file is being edited in the main code editor window. The code is as follows:

```

<manifest version="1.0" encoding="utf-8">
    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.rezaabbasi.chatapp">

        <uses-permission android:name="android.permission.INTERNET" />

        <application
            android:allowBackup="true"
            android:icon="@mipmap/ic_launcher"
            android:label="Chat App"
            android:roundIcon="@mipmap/ic_launcher_round"
            android:supportsRtl="true"
            android:theme="@style/AppTheme">

            <activity
                android:name=".RegisterActivity"
                android:screenOrientation="portrait">
            </activity>

            <activity
                android:name=".MainActivity"
                android:screenOrientation="portrait"
                android:windowSoftInputMode="stateHidden|adjustResize">
            </activity>

            <activity
                android:name=".LoginActivity"
                android:screenOrientation="portrait"
                android:windowSoftInputMode="stateHidden|adjustPan">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
        </application>
    </manifest>

```

The code editor has tabs for 'Text' and 'Merged Manifest'. The bottom status bar shows '22 chars 3:36 CRLF: UTF-8 Context: <no context>'.

Now remember the package name is what uniquely identifies your app on the Android device as well as in the Google Play store. In this case it will also be used to identify our app on Firebase.

Now the package name is also used in one of the Gradle files. Under Gradle Scripts to build.gradle (Module: app) and you see under android the old package name as shown:

The screenshot shows the Android Studio interface with the build.gradle file for the 'app' module open. The code is as follows:

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.chatapp"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:design:28.0.0'
}

```

The code editor has tabs for 'Text' and 'Gradle'. The bottom status bar shows '19 chars 6:43 CRLF: UTF-8 Context: <no context>'.

Change it so it matches with package name in Android manifest file and click Sync Now at the top right corner:

```

1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.rezaabbasi.chatapp"
7         minSdkVersion 19
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17        }
18    }
19}
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28     implementation 'com.android.support:design:28.0.0'
29}
30

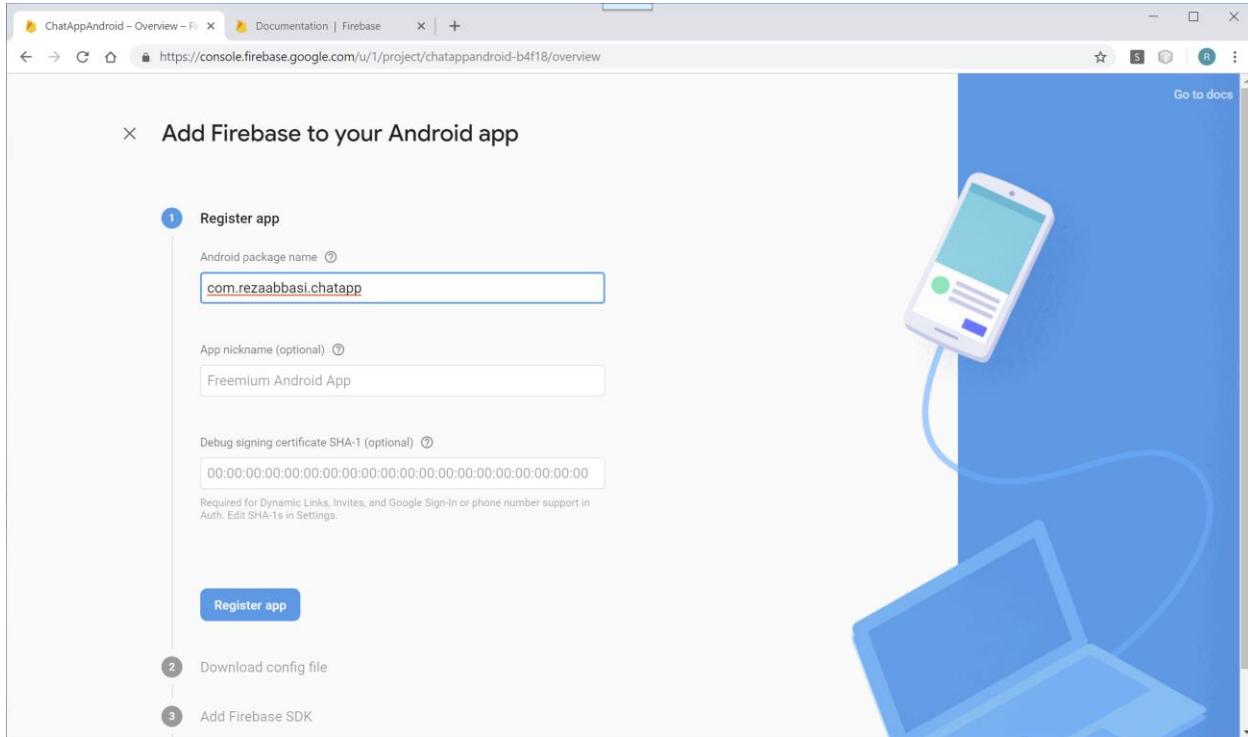
```

```

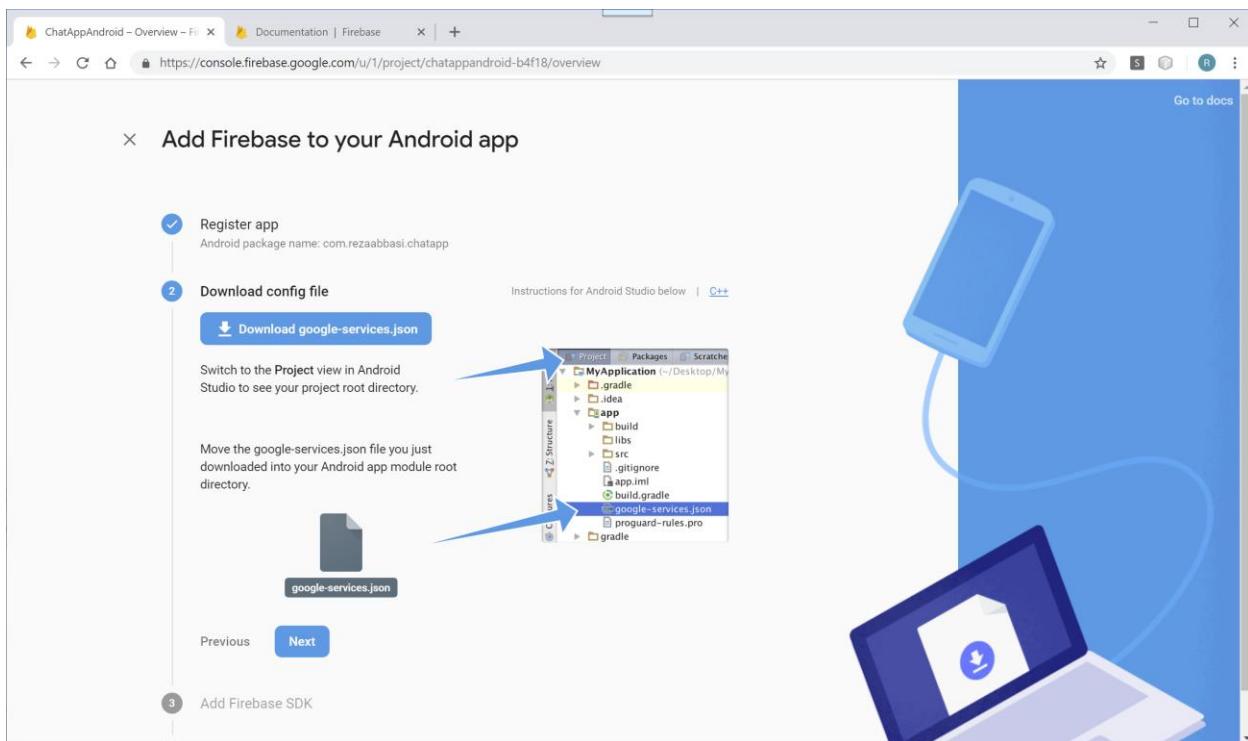
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.rezaabbasi.chatapp"
7         minSdkVersion 19
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17        }
18    }
19}
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28     implementation 'com.android.support:design:28.0.0'
29}
30

```

Now go back to Firebase site and paste your unique package name into the dialog box and leave other fields blank and click Register app as shown:

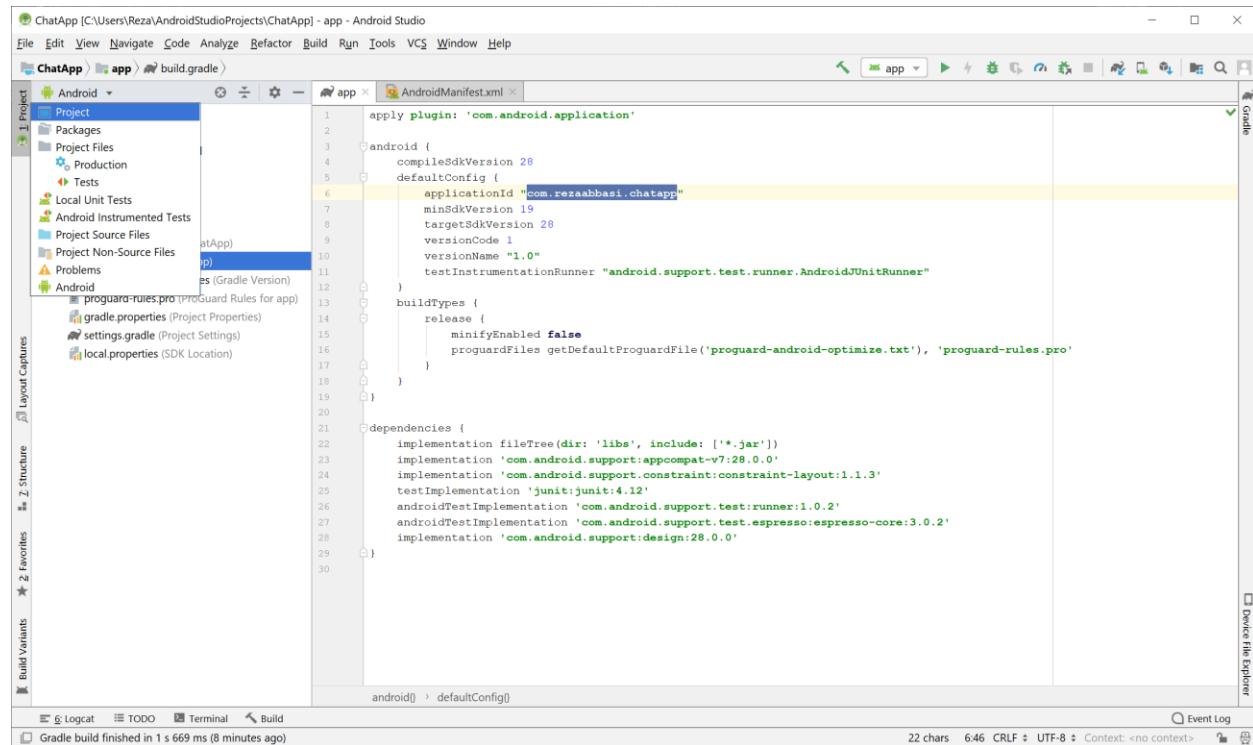


And here what you get:



Now we need to download a small file as instructed by Firebase and we need to place this file in our app folder as shown. So, download the file.

Now in Android Studio switch the view from Android to Project as shown:

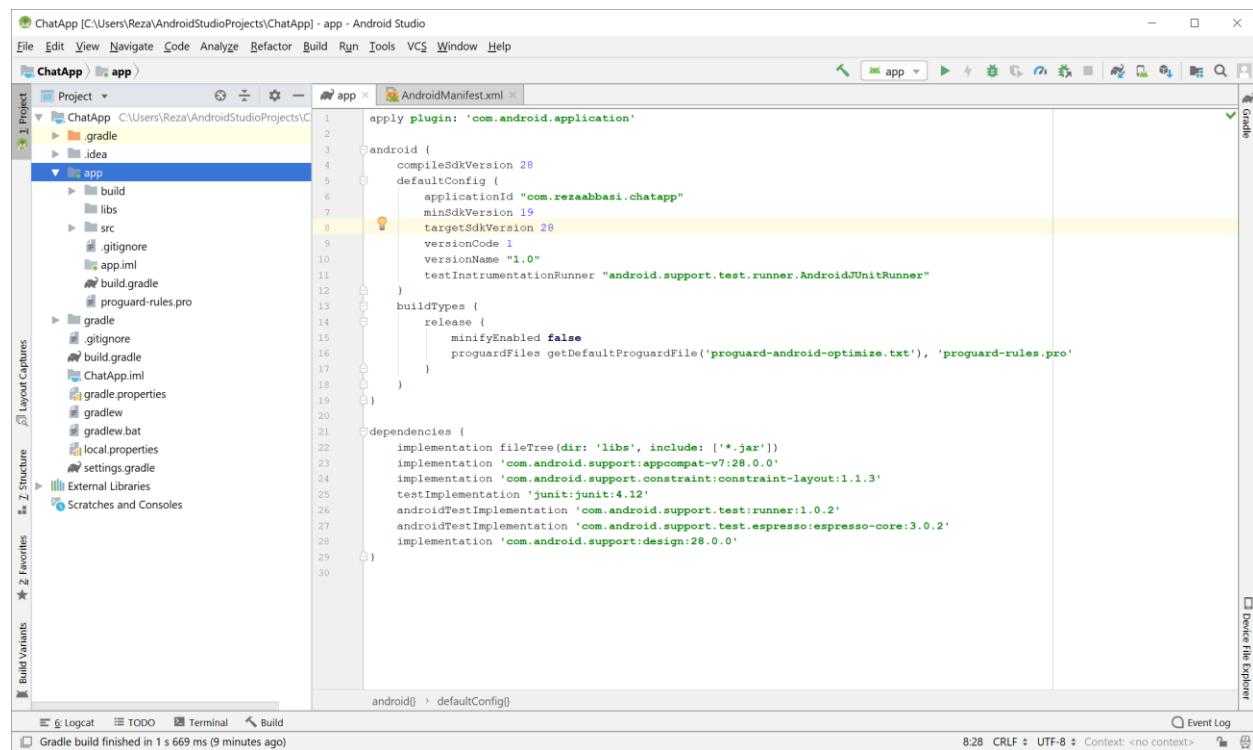


The screenshot shows the Android Studio interface with the 'Project' view selected in the left sidebar. The central code editor displays the `build.gradle` file for the 'app' module. The code defines the application's configuration, including its ID, version, and dependencies. A specific line of code, `minifyEnabled false`, is highlighted in yellow, indicating it is being modified or has been modified. The bottom status bar shows a successful Gradle build completed in 1 second.

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

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.rezaabbasi.chatapp"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:design:28.0.0'
}
```



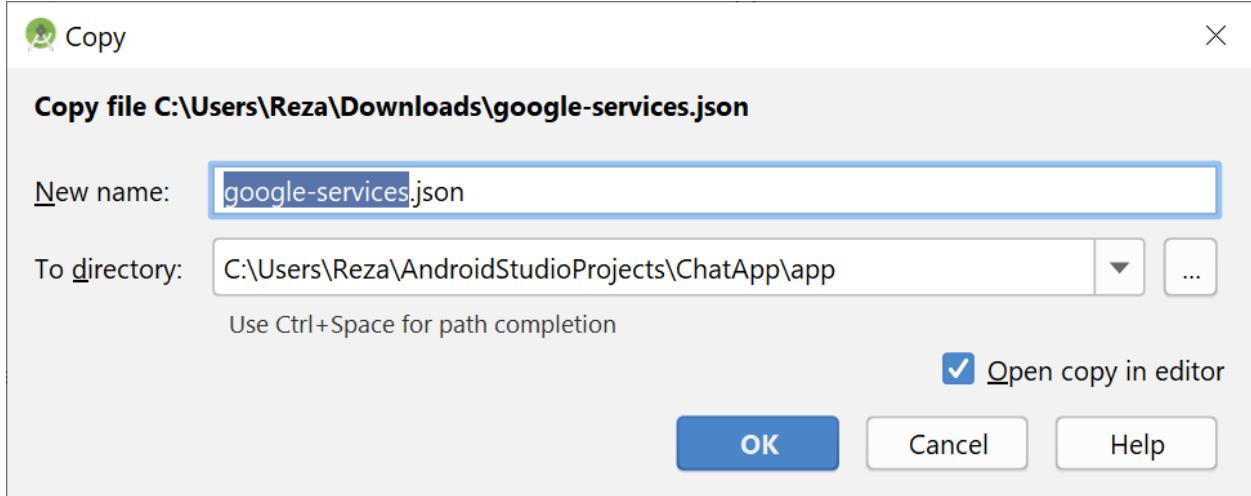
This screenshot is nearly identical to the one above, showing the 'Project' view in Android Studio. The `build.gradle` file is open in the code editor, and the same line of code, `minifyEnabled false`, is highlighted in yellow. The bottom status bar indicates a successful Gradle build completed in 1 second.

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

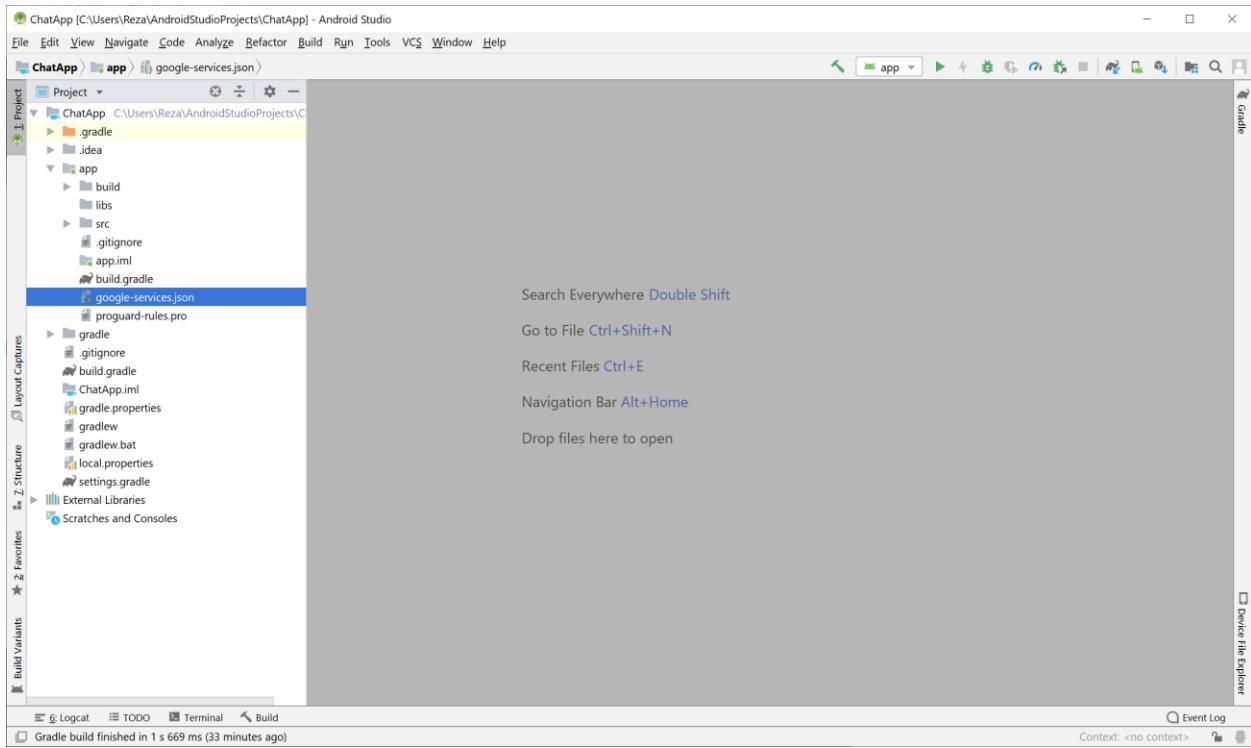
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.rezaabbasi.chatapp"
        minSdkVersion 19
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.support:design:28.0.0'
}
```

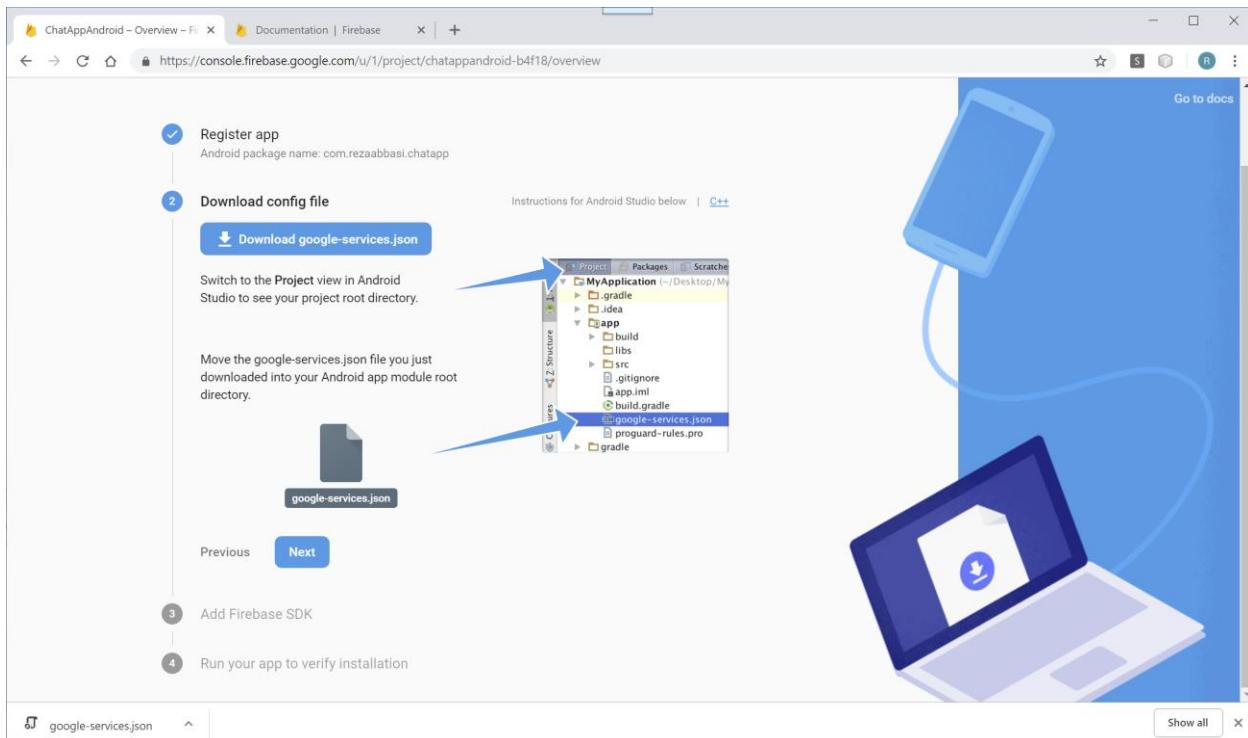
Copy the downloaded file (google-services.json) and right click on the app folder and select paste or you can drag the file to the app folder as shown:



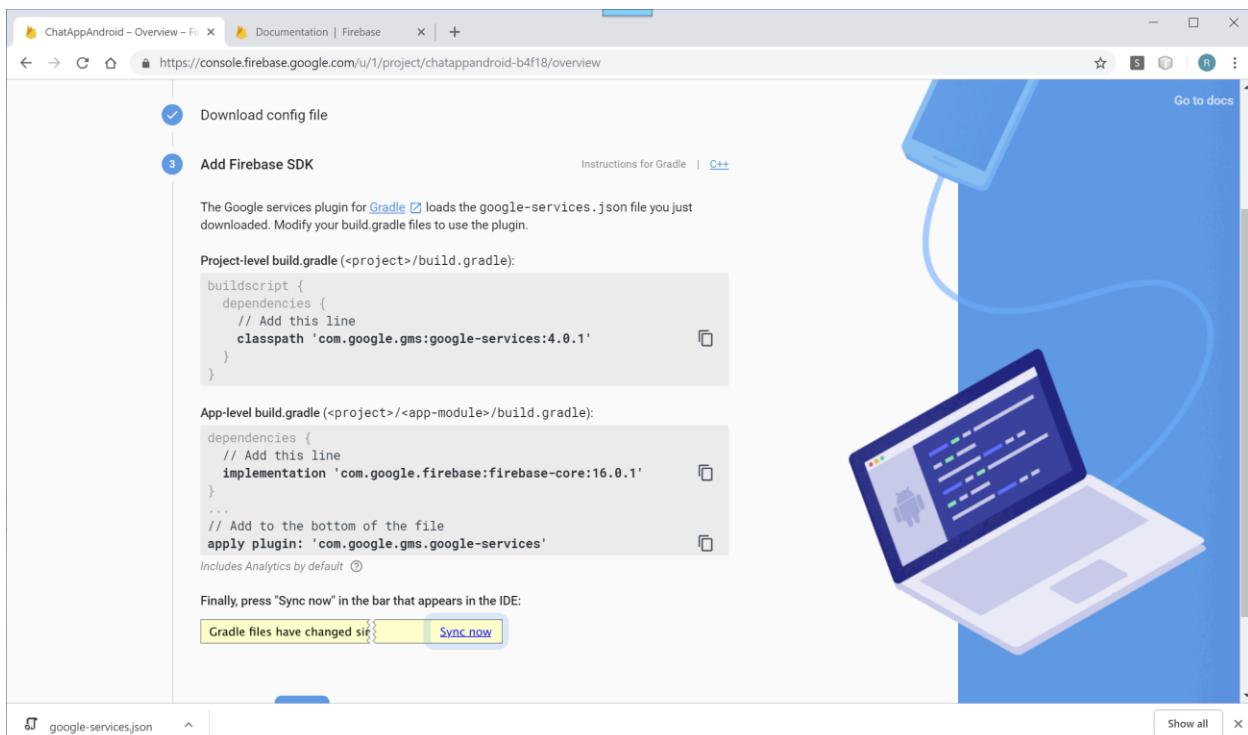
Click OK and the file will be copied to the app directory as shown:



Now go back to Firebase website and click Next:



The screenshot shows the 'Download config file' step in the Firebase console. It includes instructions to move the downloaded `google-services.json` file into the `app` module's root directory of an Android Studio project. A blue arrow points from the `google-services.json` file icon to its location in the project structure.



The screenshot shows the 'Add Firebase SDK' step in the Firebase console. It provides instructions for adding the Google services plugin to the `build.gradle` files. It shows the code snippets for the `Project-level build.gradle` and `App-level build.gradle`. A blue arrow points from the 'Sync now' button in the Firebase console to the 'Sync now' button in the Android Studio status bar.

Copy the lines and add them to your gradle files as instructed and as shown. Click on the little icon to the right of each line to copy it and then paste it in your gradle files:

ChatAppAndroid - Overview - File Documentation | Firebase x +

<https://console.firebaseio.google.com/u/1/project/chatappandroid-b4f18/overview>

Download config file

Add Firebase SDK

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:4.0.1'
    }
}
```

App-level build.gradle (<project>/<app-module>/build.gradle):

```
dependencies {
    // Add this line
    implementation 'com.google.firebase:firebase-core:16.0.1'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Includes Analytics by default ⓘ

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed sir Sync now

Copied to clipboard

ChatApp [C:\Users\Reza\AndroidStudioProjects\ChatApp] - ChatApp - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ChatApp build.gradle

Project ChatApp C:\Users\Reza\AndroidStudioProjects\ChatApp

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

Sync Now

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.3.1'
        classpath 'com.google.gms:google-services:4.0.1'
        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Build Variants Favorites Z Structure External Libraries Scratches and Consoles

Event Log Gradle build finished in 1 s 669 ms (43 minutes ago) 11:57 CRLF UTF-8 Context: <no context>

ChatApp [C:\Users\Reza\AndroidStudioProjects\ChatApp] - app - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ChatApp > app > build.gradle

Project Layout Captures Favorites Z-Structure External Libraries Scratches and Consoles

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.rezaabbasi.chatapp"
7         minSdkVersion 19
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17        }
18    }
19}
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28     implementation 'com.android.support:design:28.0.0'
29     implementation 'com.google.firebaseio:firebase-core:16.0.1'
30 }
31
32
```

dependencies[]

Logcat TODO Terminal Build Event Log

Gradle build finished in 1 s 669 ms (45 minutes ago) 29.62 CRLF UTF-8 Context: <no context>

ChatApp [C:\Users\Reza\AndroidStudioProjects\ChatApp] - app - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ChatApp > app > build.gradle

Project Layout Captures Favorites Z-Structure External Libraries Scratches and Consoles

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 28
5     defaultConfig {
6         applicationId "com.rezaabbasi.chatapp"
7         minSdkVersion 19
8         targetSdkVersion 28
9         versionCode 1
10        versionName "1.0"
11        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12    }
13    buildTypes {
14        release {
15            minifyEnabled false
16            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17        }
18    }
19}
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:28.0.0'
24     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25     testImplementation 'junit:junit:4.12'
26     androidTestImplementation 'com.android.support.test:runner:1.0.2'
27     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28     implementation 'com.android.support:design:28.0.0'
29     implementation 'com.google.firebaseio:firebase-core:16.0.1'
30 }
31
32 apply plugin: 'com.google.gms.google-services'
```

Event Log

Gradle build finished in 1 s 669 ms (45 minutes ago) 32.47 CRLF UTF-8 Context: <no context>

You can find a list of Firebase libraries here:

<https://firebase.google.com/docs/android/setup>

The screenshot shows the Firebase documentation website at <https://firebase.google.com/docs/android/setup>. The page is titled 'LIBRARIES'. It features a sidebar with 'Guides' sections for setting up projects, adding Firebase to an app (iOS, Android, Flutter, Web), adding Firebase to a game, a server, or PWA, managing IAM, and exporting project data to BigQuery. Other sections include 'Analytics', 'DEVELOP', and 'Authentication'. The main content area displays a 'Gradle Dependency Line' table mapping library names to their corresponding services. The 'Authentication' row is highlighted in blue. The right sidebar contains links for 'Contents', 'Prerequisites', 'Add Firebase to your app', 'Use the Firebase Assistant', 'Manually add Firebase', 'Available libraries', and 'Next steps'. A search bar and navigation buttons are also present.

Gradle Dependency Line	Service
com.google.firebaseio.firebaseio-core:16.0.7	Analytics
com.google.firebaseio.firebaseio-database:16.0.6	Realtime Database
com.google.firebaseio.firebaseio-firestore:18.0.1	Cloud Firestore
com.google.firebaseio.firebaseio-storage:16.0.5	Storage
com.crashlytics.sdk.android:crashlytics:2.9.9	Crashlytics
com.google.firebaseio.firebaseio-auth:16.1.0	Authentication
com.google.firebaseio.firebaseio-messaging:17.3.4	Cloud Messaging
com.google.firebaseio.firebaseio-config:16.3.0	Remote Config
com.google.firebaseio.firebaseio-invites:16.1.0	Invites and Dynamic Links
com.google.firebaseio.firebaseio-ads:17.1.3	AdMob
com.google.firebaseio.firebaseio-appindexing:17.1.0	App Indexing
com.google.firebaseio.firebaseio-perf:16.2.3	Performance Monitoring
com.google.firebaseio.firebaseio-functions:16.1.3	Cloud Functions for Firebase Client SDK
com.google.firebaseio.firebaseio-ml-vision:19.0.2	ML Kit (Vision)
com.google.firebaseio.firebaseio-ml-model-interpreter:17.0.3	ML Kit (Custom Model)

Add the highlighted library for Firebase authentication and click “Sync Now” at the top right corner in Android Studio.

The screenshot shows the Android Studio interface with the project 'ChatApp' open. The 'build.gradle' file is displayed in the code editor. The 'dependencies' block is highlighted with a yellow selection bar. Inside the 'dependencies' block, the line 'implementation 'com.google.firebaseio.firebaseio-auth:16.1.0'' is specifically highlighted in orange. The bottom status bar shows the message 'Gradle build finished in 1 s 378 ms (a minute ago)'. The bottom right corner shows the 'Event Log' tab.

```
16    }
17    }
18    }
19    }
20}
21dependencies {
22    implementation fileTree(dir: 'libs', include: ['*.jar'])
23    implementation 'com.android.support:appcompat-v7:28.0.0'
24    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
25    testImplementation 'junit:junit:4.12'
26    androidTestImplementation 'com.android.support.test:runner:1.0.2'
27    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
28    implementation 'com.android.support:design:28.0.0'
29    implementation 'com.google.firebaseio.firebaseio-core:16.0.7'
30    implementation 'com.android.support:support-v4:28.0.0'
31    implementation 'com.google.firebaseio.firebaseio-auth:16.1.0'
32}
33}
34}
35}
36}

apply plugin: 'com.google.gms.google-services'
```

Click Next in Firebase console as shown:

The Google services plugin for [Gradle](#) loads the google-services.json file you just downloaded. Modify your build.gradle files to use the plugin.

Project-level build.gradle (<project>/build.gradle):

```
buildscript {  
    dependencies {  
        // Add this line  
        classpath 'com.google.gms:google-services:4.0.1'  
    }  
}
```

App-level build.gradle (<project>/app-module/build.gradle):

```
dependencies {  
    // Add this line  
    implementation 'com.google.firebase:firebase-core:16.0.1'  
}  
...  
// Add to the bottom of the file  
apply plugin: 'com.google.gms.google-services'
```

Includes Analytics by default [?](#)

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed [Sync now](#)

Previous [Next](#)

4 Run your app to verify installation

X Add Firebase to your Android app

1 Register app
Android package name: com.rezaabbasi.chatapp

2 Download config file

3 Add Firebase SDK

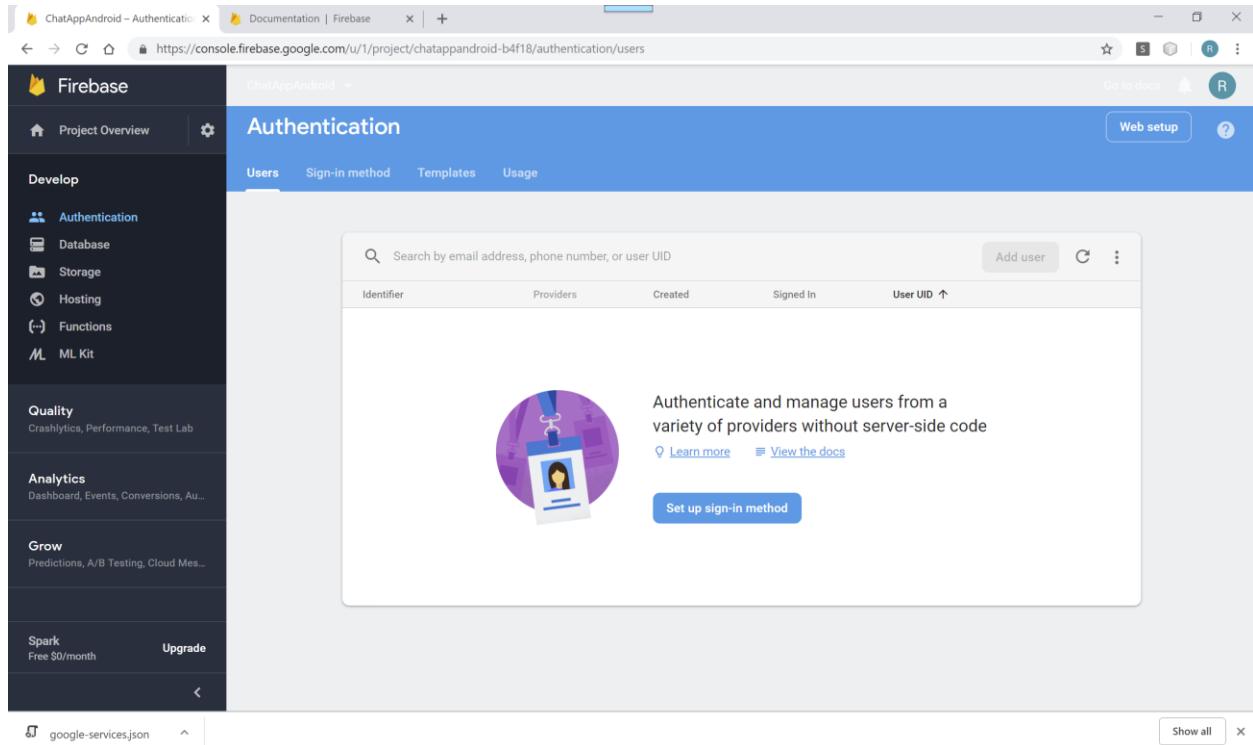
4 Run your app to verify installation

Checking if the app has communicated with our servers. You may need to uninstall and reinstall your app.

Previous [Continue to console](#) [Skip this step](#)

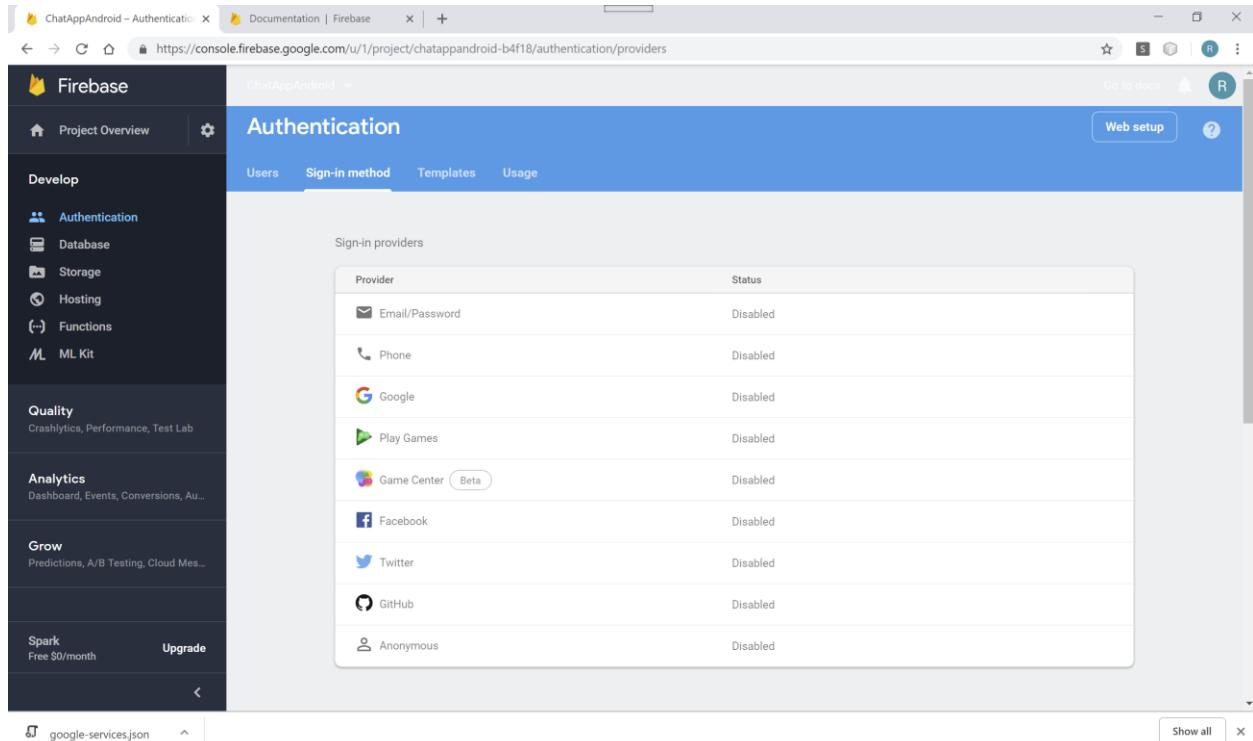
5 google-services.json

Click Skip this step to go back to the console window. The next thing we need to do is to tell Firebase how to handle user authentication. Under the Develop section select Authentication as shown:



The screenshot shows the Firebase console for a project named "ChatAppAndroid - Authentication". The left sidebar has sections for Project Overview, Develop (with Authentication selected), Database, Storage, Hosting, Functions, ML Kit, Quality, Analytics, and Grow. The main content area is titled "Authentication" and has tabs for Users, Sign-in method, Templates, and Usage. The "Sign-in method" tab is active, showing a search bar and a table with columns: Identifier, Providers, Created, Signed In, and User UID. Below the table is a purple circular icon with a person's face and a blue badge, followed by the text: "Authenticate and manage users from a variety of providers without server-side code". It includes links to "Learn more" and "View the docs", and a blue button labeled "Set up sign-in method".

Click on Set up sign-in method:



The screenshot shows the same Firebase console interface as the previous one, but the "Sign-in method" tab is now inactive, and the "Sign-in providers" tab is active. The main content area is titled "Authentication" and has tabs for Users, Sign-in method, Templates, and Usage. The "Sign-in providers" tab is active, showing a table titled "Sign-in providers". The table has two columns: "Provider" and "Status". The providers listed are: Email/Password (Disabled), Phone (Disabled), Google (Disabled), Play Games (Disabled), Game Center (Beta), Facebook (Disabled), Twitter (Disabled), GitHub (Disabled), and Anonymous (Disabled).

Select Email/Password:

The screenshot shows the Firebase Authentication settings page. The 'Sign-in method' tab is selected. Under 'Sign-in providers', the 'Email/Password' provider is listed with its status set to 'Enable'. Below it, other providers like Phone, Google, Play Games, and Custom OAuth are listed as disabled. A 'Save' button is visible at the bottom right of the modal.

Enable the Email/Password sign in:

This screenshot is identical to the one above, but the 'Email/Password' provider's status has been changed to 'Enabled' (indicated by a blue toggle switch). The 'Save' button is now highlighted in blue, indicating it has been clicked.

Click Save.

The screenshot shows the Firebase Authentication screen for a project named "ChatAppAndroid". The left sidebar includes sections for Project Overview, Develop (Authentication, Database, Storage, Hosting, Functions, ML Kit), Quality (Crashlytics, Performance, Test Lab), Analytics (Dashboard, Events, Conversions, etc.), and Grow (Predictions, A/B Testing, Cloud Mes...). The main content area is titled "Authentication" and shows the "Sign-in method" tab selected. It lists various sign-in providers: Email/Password (Enabled), Phone (Disabled), Google (Disabled), Play Games (Disabled), Game Center (Beta) (Disabled), Facebook (Disabled), Twitter (Disabled), GitHub (Disabled), and Anonymous (Disabled). A "Sign-in providers" section header is present above the provider list.

Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Disabled
Play Games	Disabled
Game Center (Beta)	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled
Anonymous	Disabled

Typically, you allow the users to sign in with their Google, Facebook, Twitter, etc. accounts but for the purpose this app we choose Email/Password.