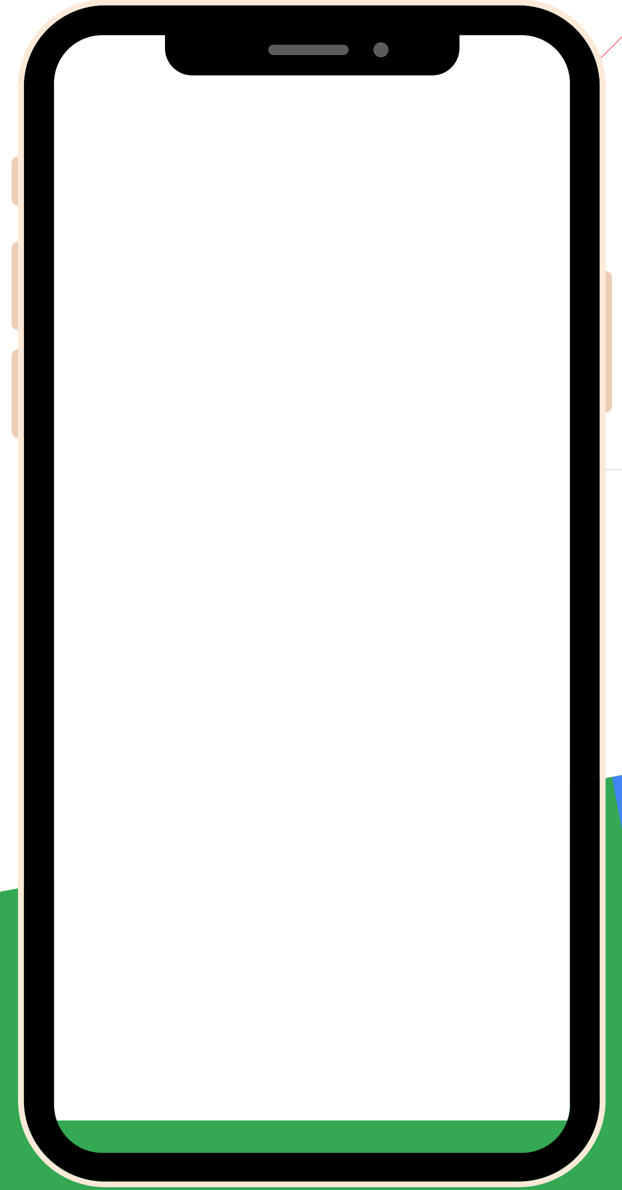
Google Login Setup In Flutter

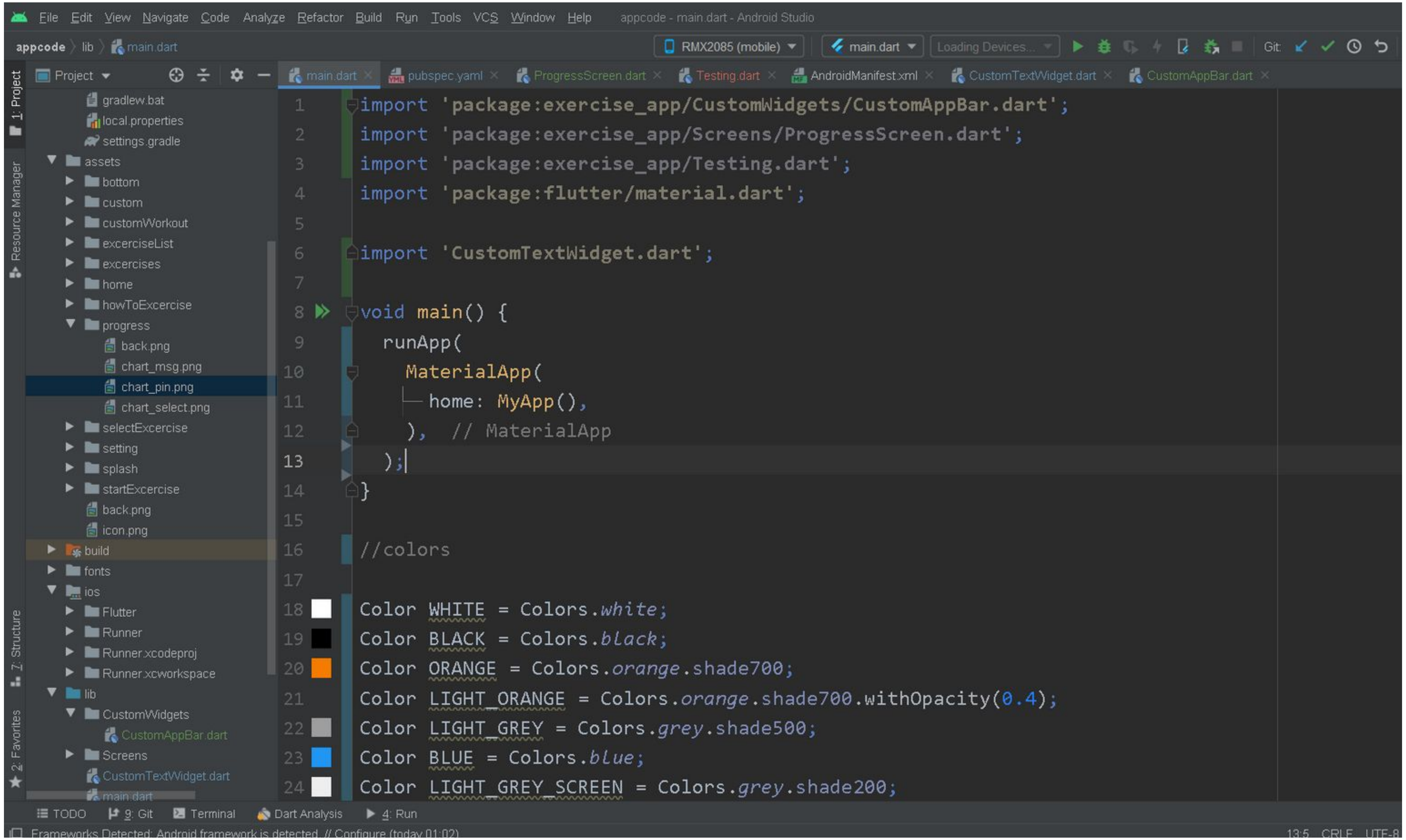
Requirements:-

- Flutter should be installed on your system (latest version)
- Android Studio should be installed on your system.
- Must have a Google Account
- Must have a firebase account.
- Emulator or Android/iOS device for testing.



Before moving further it is assumed that all the requirements are fulfilled. So, without wasting any time lets move to our first step.

Step - 1



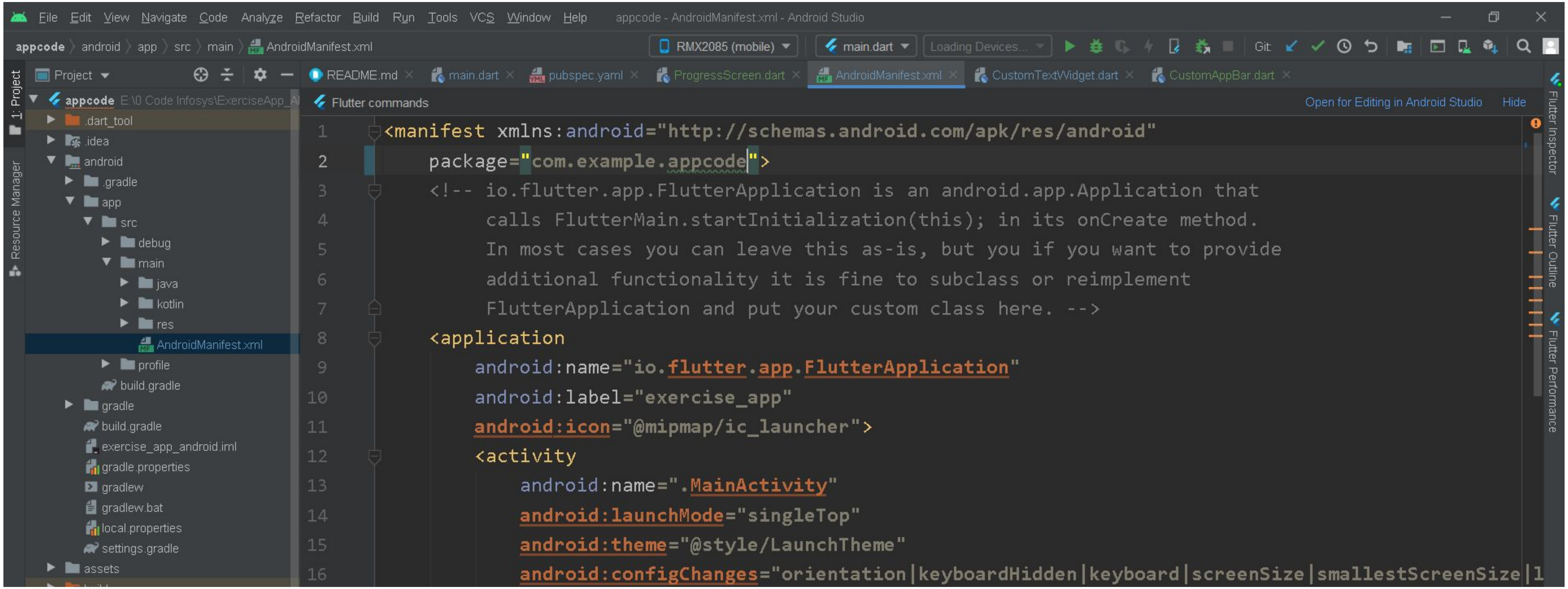
```dart
import 'package:exercise_app/CustomWidgets/CustomAppBar.dart';
import 'package:exercise_app/Screens/ProgressScreen.dart';
import 'package:exercise_app/Testing.dart';
import 'package:flutter/material.dart';

import 'CustomTextWidget.dart';

void main() {
  runApp(
    MaterialApp(
      home: MyApp(),
    ), // MaterialApp
  );
}

//colors

Color WHITE = Colors.white;
Color BLACK = Colors.black;
Color ORANGE = Colors.orange.shade700;
Color LIGHT_ORANGE = Colors.orange.shade700.withOpacity(0.4);
Color LIGHT_GREY = Colors.grey.shade500;
Color BLUE = Colors.blue;
Color LIGHT_GREY_SCREEN = Colors.grey.shade200;
```

It is assumed that you have downloaded and extracted appcode on your system along with your app is also connected to firebase. First of all we need to find SHA1 for our app. In order to proceed, open your project in

Step - 2

Navigate to AndroidManifest.xml file in your project

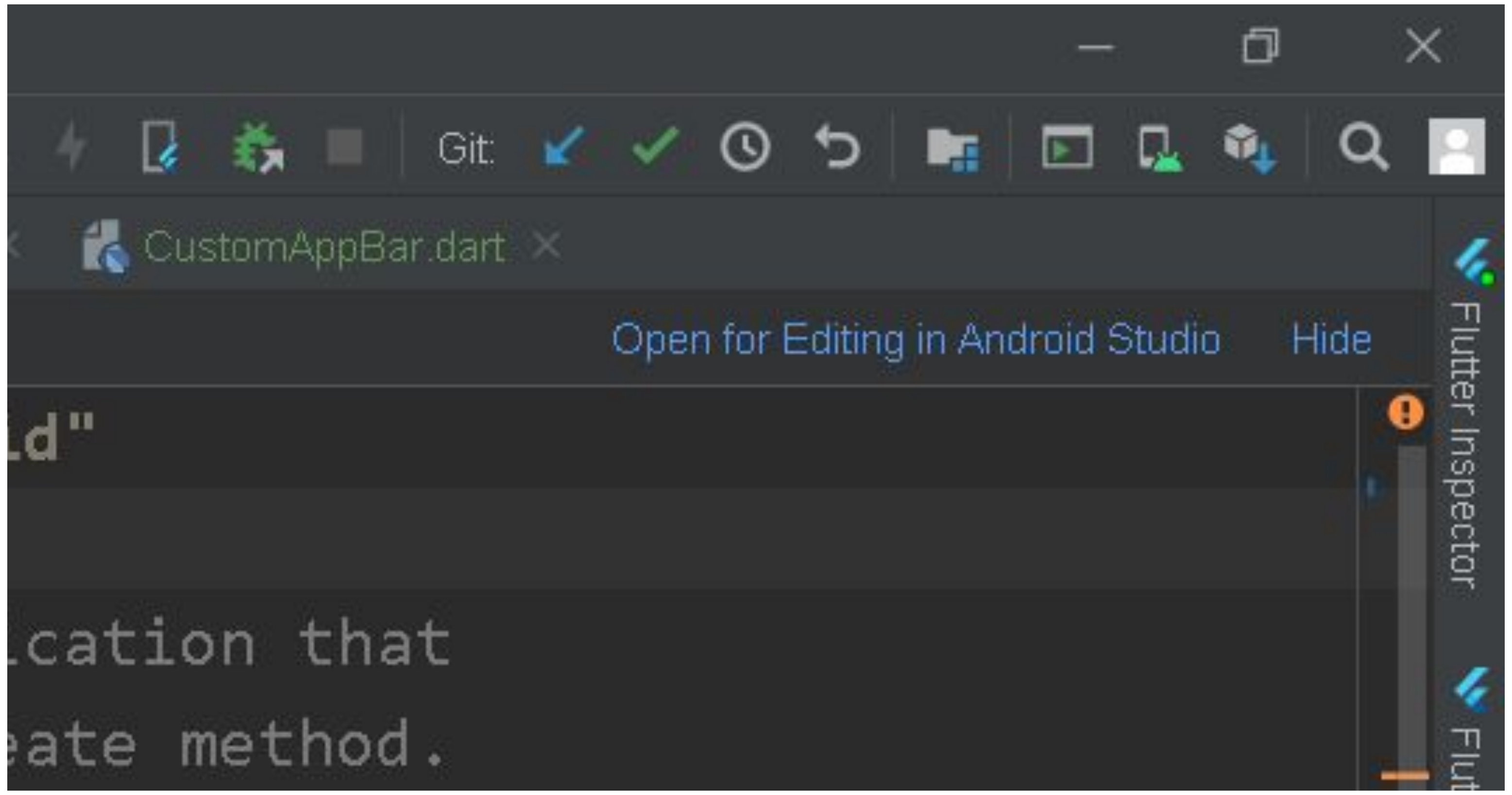
Root -> android -> app -> src -> main -> AndroidManifest.xml



```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.appcode">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="exercise_app"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|l
```

Step - 3

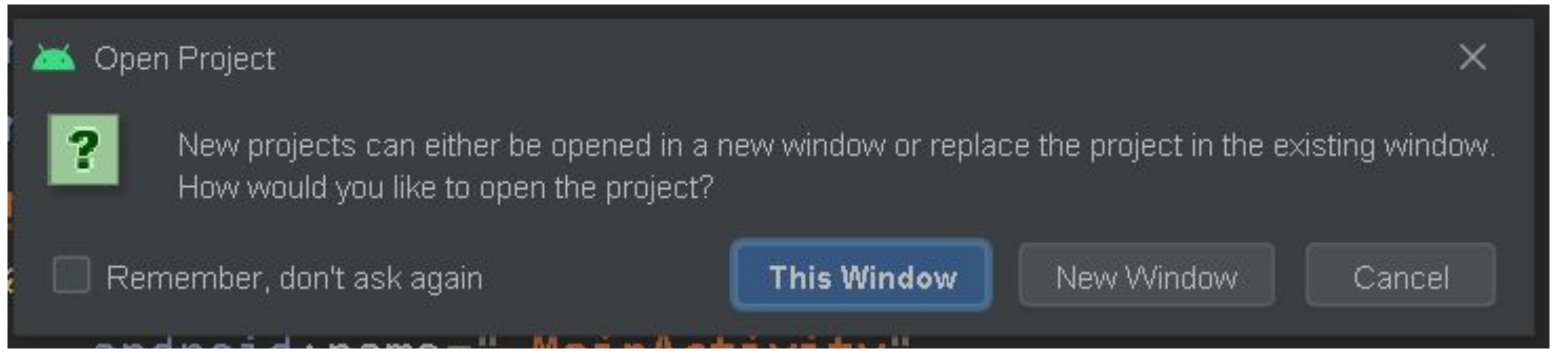
In top right corner you can see text witten in blue color "open for editing in android studio". click on this text. and open project in new window



CustomAppBar.dart

Open for Editing in Android Studio   Hide

```
d"

cation that
ate method.
```

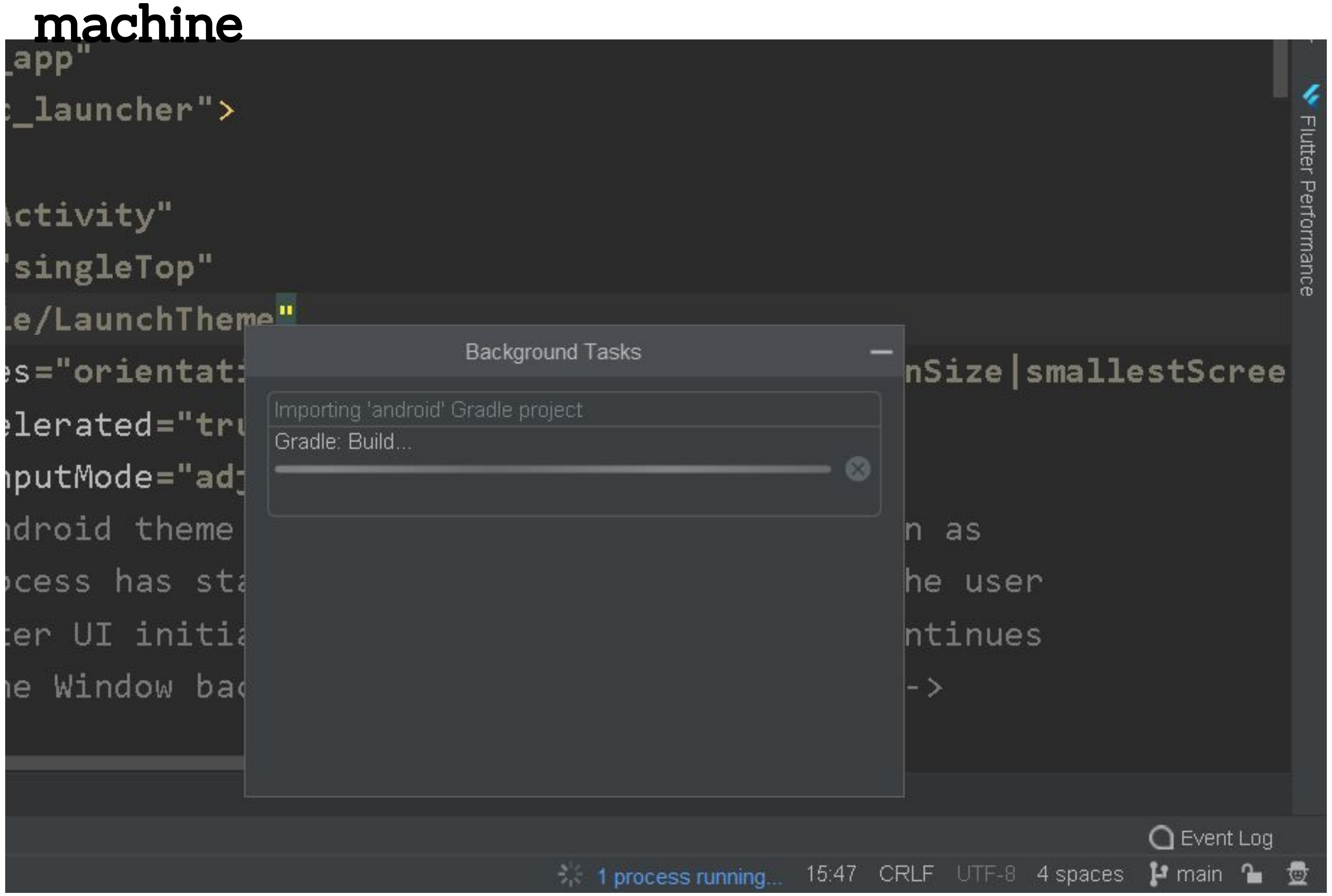


Open Project

New projects can either be opened in a new window or replace the project in the existing window.
How would you like to open the project?

☐ Remember, don't ask again

Wait while gradle is building this process is going to take upto 5-10 minutes depending on the performance of your machine



```
app"
_launcher">

ctivity"
'singleTop"
e/LaunchTheme"
s="orientat:                                   nSize|smallestScree
lerated="tru
putMode="adj
droid theme                                    n as
cess has sta                                   he user
er UI initia                                   ntinues
e Window bac                                   ->
```

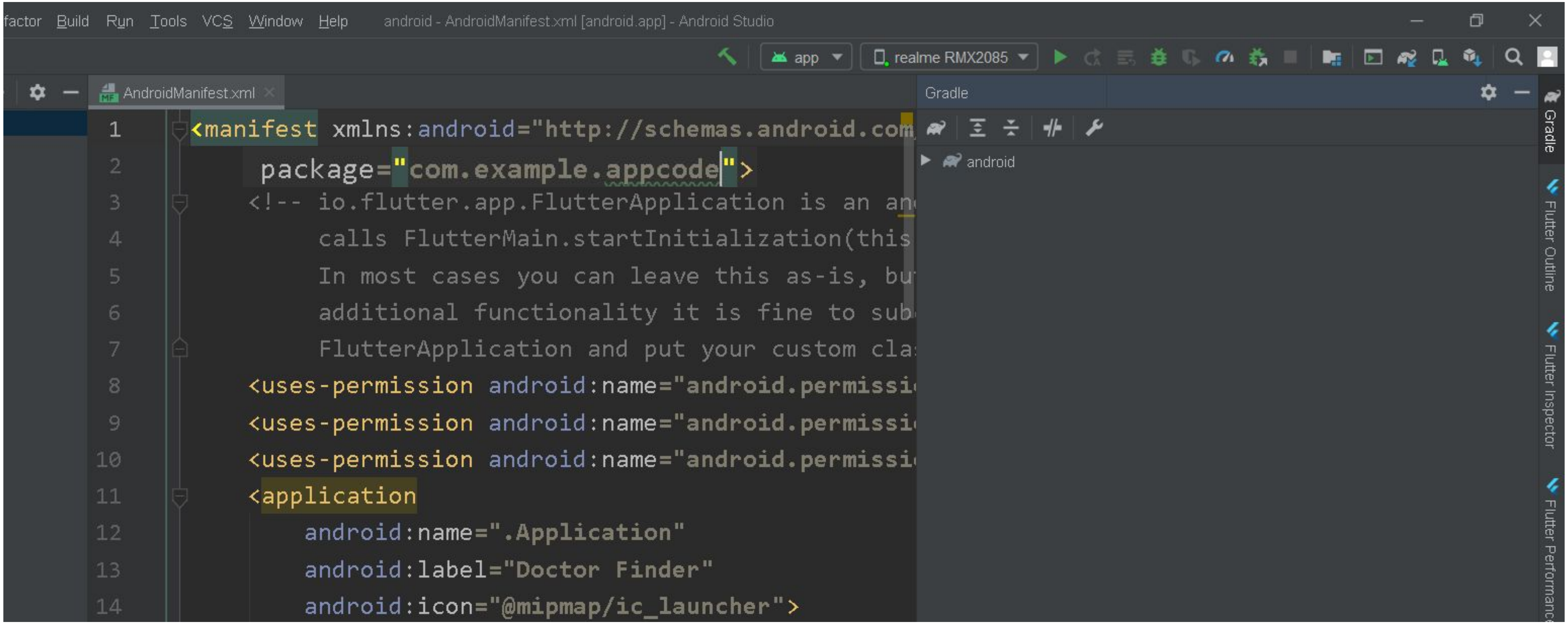
Background Tasks

Importing 'android' Gradle project
Gradle: Build...

1 process running...   15:47   CRLF   UTF-8   4 spaces   main

Step - 4

After gradle build finishes click on "gradle" in the right vertical bar



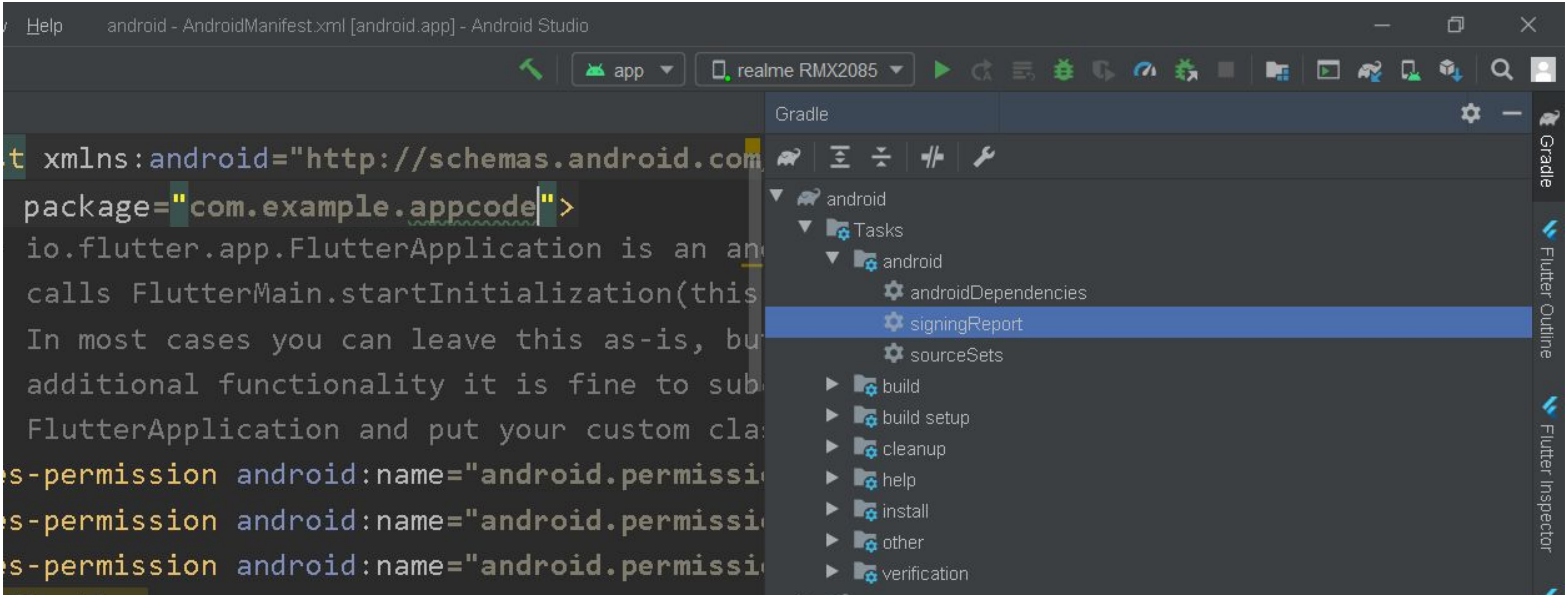
```
<manifest xmlns:android="http://schemas.android.com
    package="com.example.appcode">
    <!-- io.flutter.app.FlutterApplication is an an
         calls FlutterMain.startInitialization(this
         In most cases you can leave this as-is, bu
         additional functionality it is fine to sub
         FlutterApplication and put your custom cla
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
    <application
        android:name=".Application"
        android:label="Doctor Finder"
        android:icon="@mipmap/ic_launcher">
```

Step - 5

Navigate to "signingReport"

android -> Tasks -> android -> signingReport

double click on it and keep an eye on the console

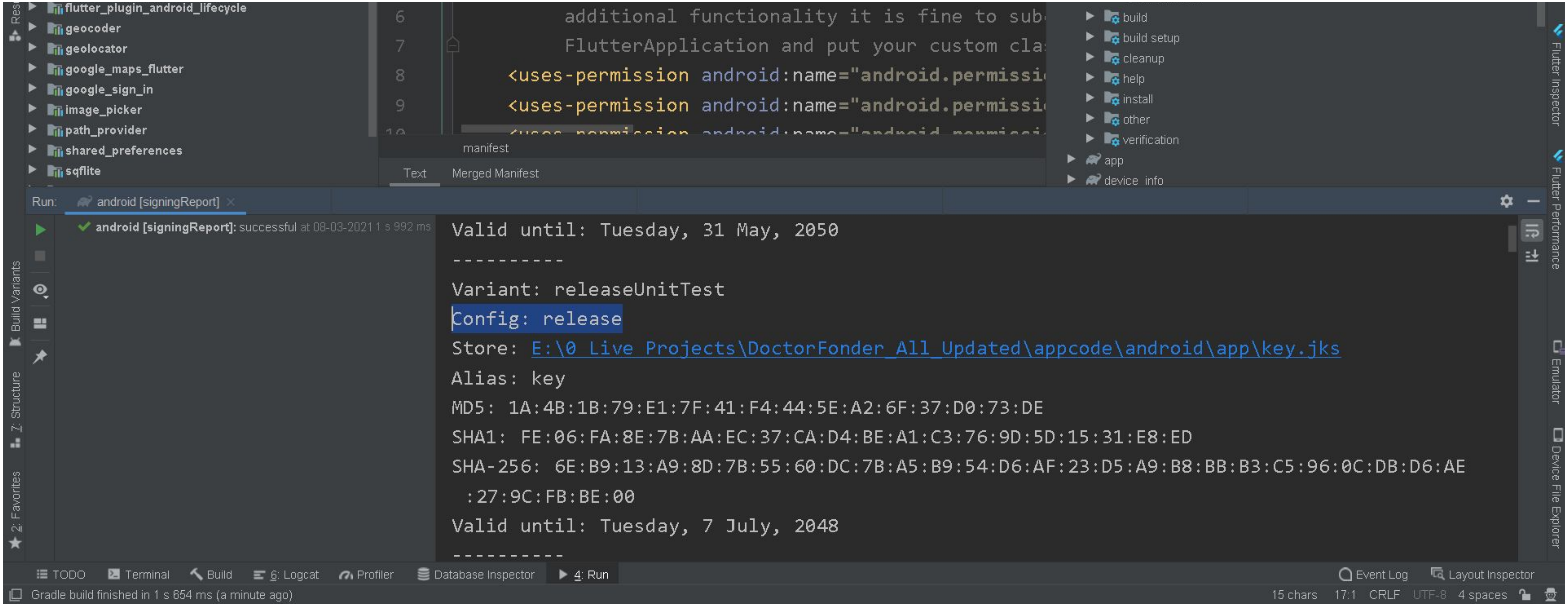


```
t xmlns:android="http://schemas.android.com
 package="com.example.appcode">
  io.flutter.app.FlutterApplication is an an
  calls FlutterMain.startInitialization(this
  In most cases you can leave this as-is, bu
  additional functionality it is fine to sub
  FlutterApplication and put your custom cla
s-permission android:name="android.permissi
s-permission android:name="android.permissi
s-permission android:name="android.permissi
```

Step - 6

if your project contains .jsk file you will be able to see two SHA1 keys in console, one for released apk and other for debug apk. Otherwise only debug apk's SHA1 will be available.

First select SHA1 with release configurations (if available).



```
        additional functionality it is fine to sub
        FlutterApplication and put your custom cla
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
```

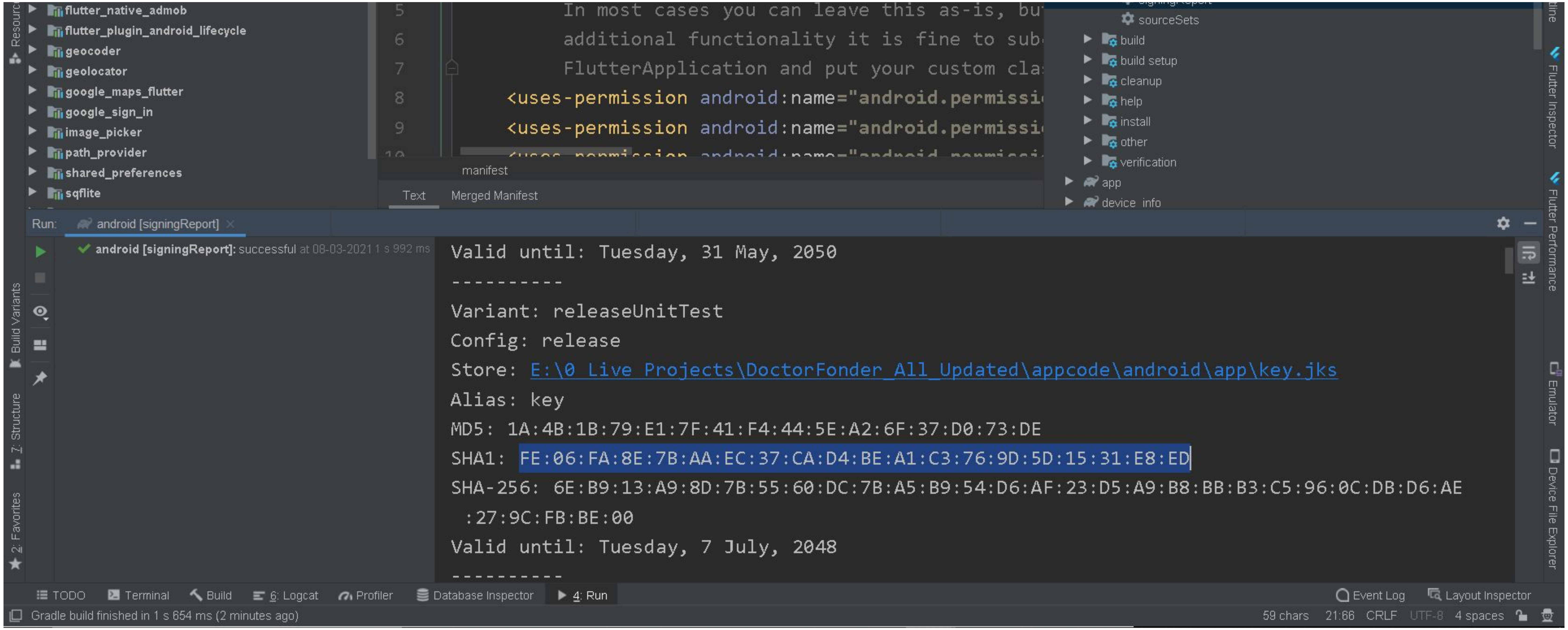
android [signingReport]: successful at 08-03-2021 1 s 992 ms

```
Valid until: Tuesday, 31 May, 2050
----------
Variant: releaseUnitTest
Config: release
Store: E:\0 Live Projects\DoctorFonder_All_Updated\appcode\android\app\key.jks
Alias: key
MD5: 1A:4B:1B:79:E1:7F:41:F4:44:5E:A2:6F:37:D0:73:DE
SHA1: FE:06:FA:8E:7B:AA:EC:37:CA:D4:BE:A1:C3:76:9D:5D:15:31:E8:ED
SHA-256: 6E:B9:13:A9:8D:7B:55:60:DC:7B:A5:B9:54:D6:AF:23:D5:A9:B8:BB:B3:C5:96:0C:DB:D6:AE
 :27:9C:FB:BE:00
Valid until: Tuesday, 7 July, 2048
----------
```

Gradle build finished in 1 s 654 ms (a minute ago)

Step - 7

Select and copy SHA1 with release configuration. and open up your firebase console where you have created project.

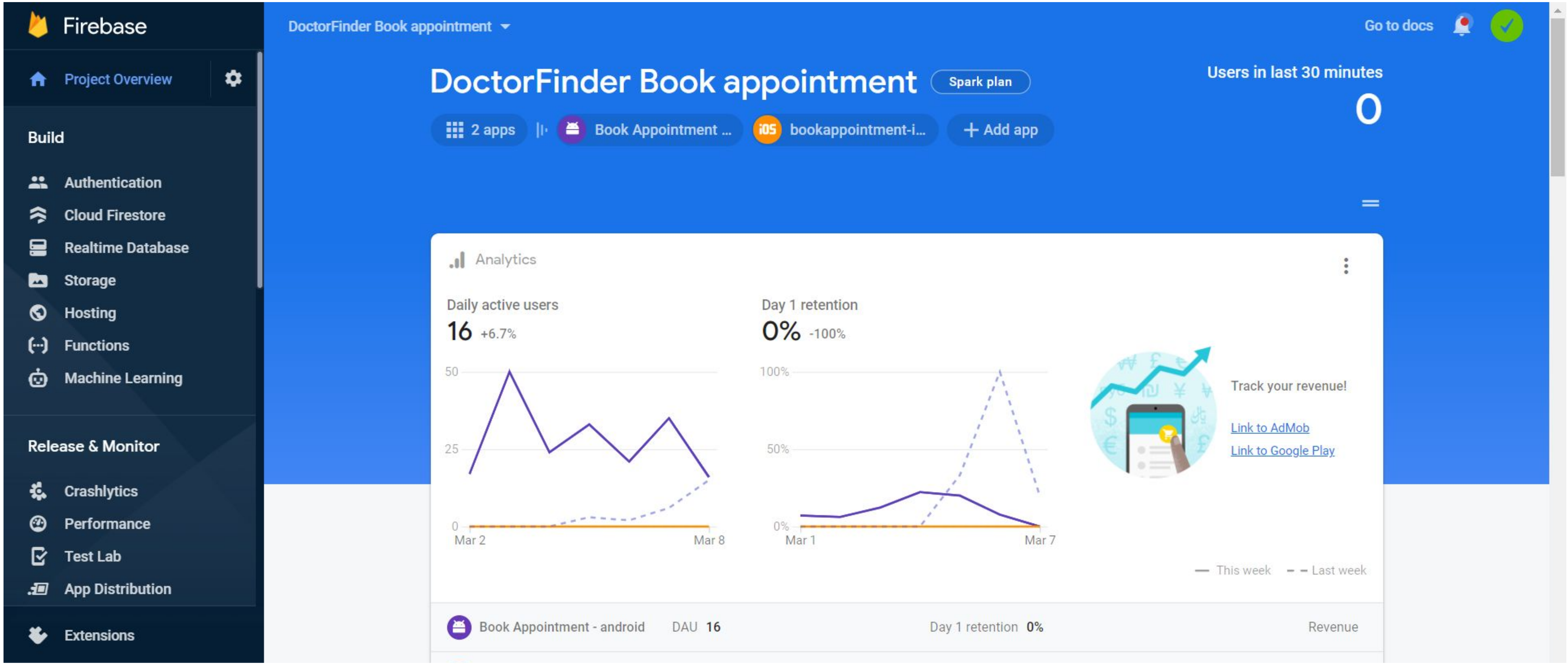


```
        In most cases you can leave this as-is, bu
        additional functionality it is fine to sub
        FlutterApplication and put your custom cla
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
```

```
Valid until: Tuesday, 31 May, 2050
----------
Variant: releaseUnitTest
Config: release
Store: E:\0 Live Projects\DoctorFonder_All_Updated\appcode\android\app\key.jks
Alias: key
MD5: 1A:4B:1B:79:E1:7F:41:F4:44:5E:A2:6F:37:D0:73:DE
SHA1: FE:06:FA:8E:7B:AA:EC:37:CA:D4:BE:A1:C3:76:9D:5D:15:31:E8:ED
SHA-256: 6E:B9:13:A9:8D:7B:55:60:DC:7B:A5:B9:54:D6:AF:23:D5:A9:B8:BB:B3:C5:96:0C:DB:D6:AE
 :27:9C:FB:BE:00
Valid until: Tuesday, 7 July, 2048
----------
```

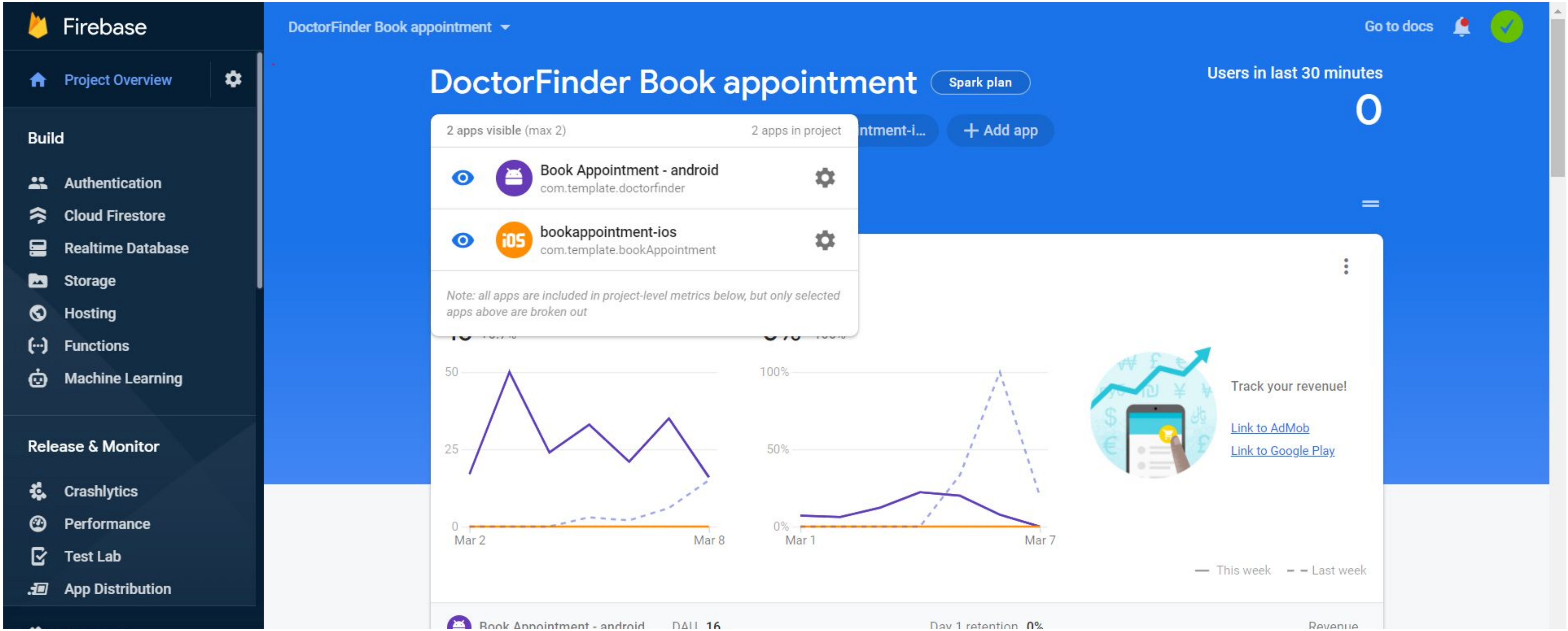
Step - 8

Open up your firebase project.



Step - 9

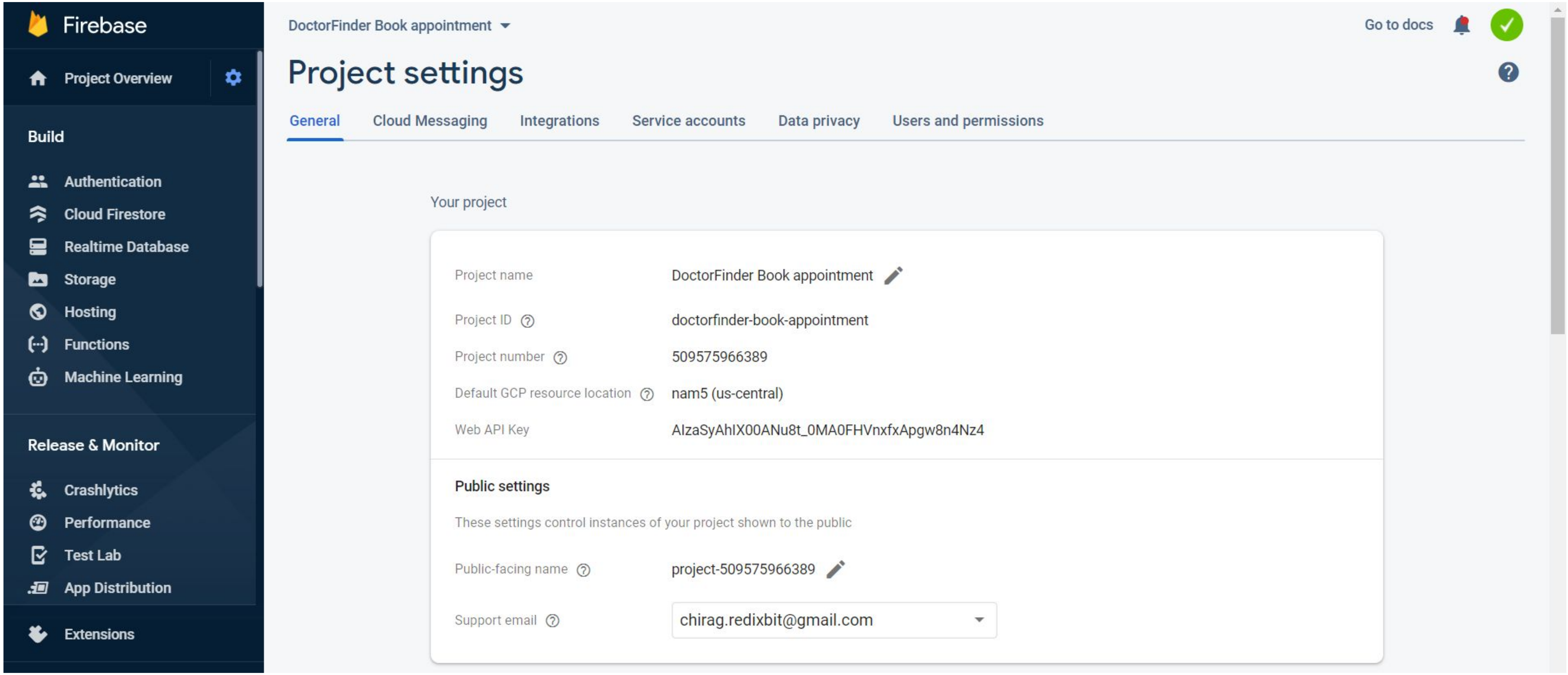
Select the app in which you want to add google login. click on "settings" button to open project.



Note: all apps are included in project-level metrics below, but only selected apps above are broken out

Step - 10

Project setting will appear in front of you. scroll down to the bottom of page



Project settings

Your project

Project name	DoctorFinder Book appointment
Project ID	doctorfinder-book-appointment
Project number	509575966389
Default GCP resource location	nam5 (us-central)
Web API Key	AIzaSyAhIX00ANu8t_0MA0FHVnxfxApgw8n4Nz4

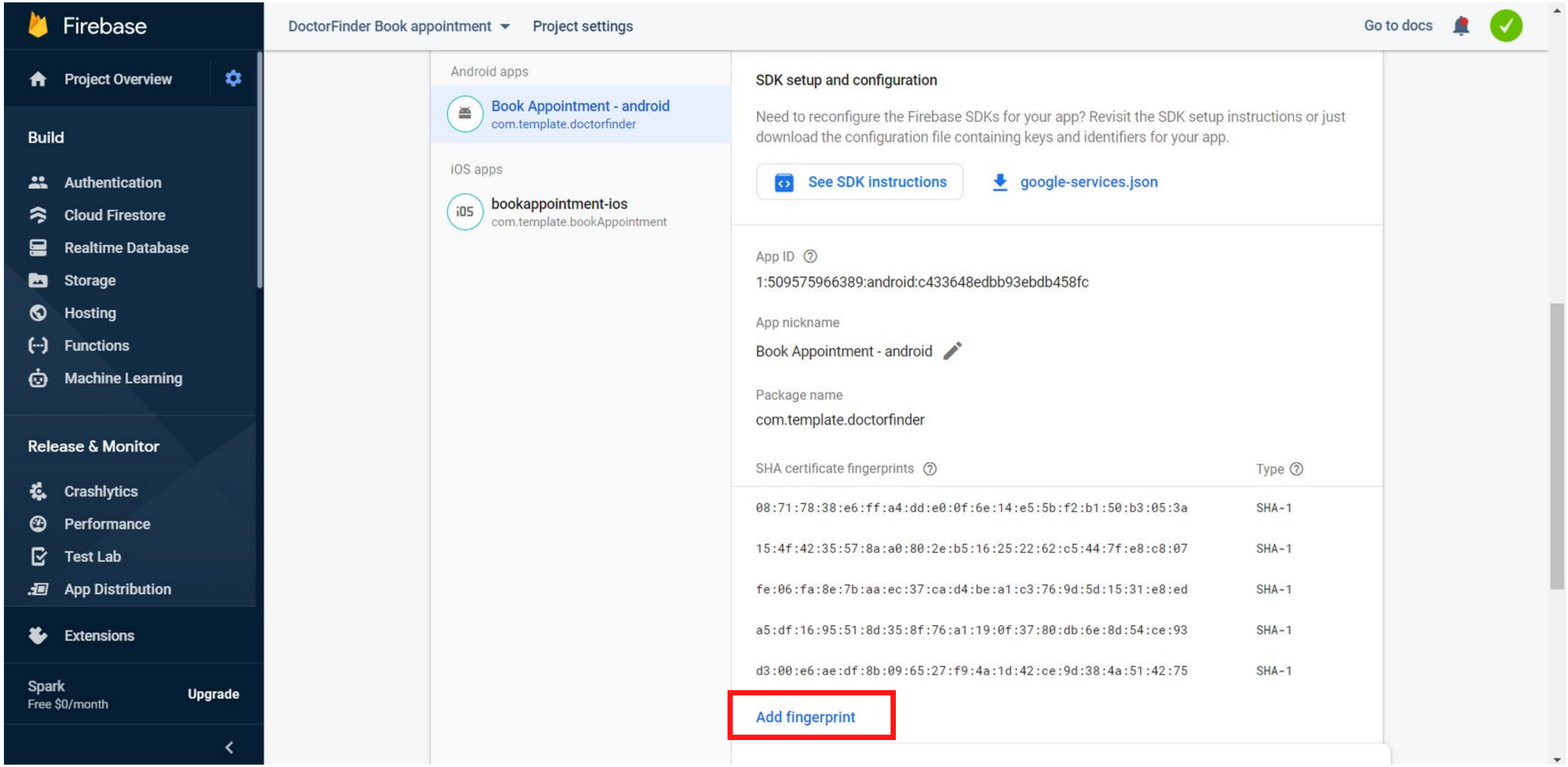
Public settings

These settings control instances of your project shown to the public

Public-facing name	project-509575966389
Support email	chirag.redixbit@gmail.com

Step - 11

Click on add fingerprint button.



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

App ID
1:509575966389:android:c433648edbb93ebdb458fc

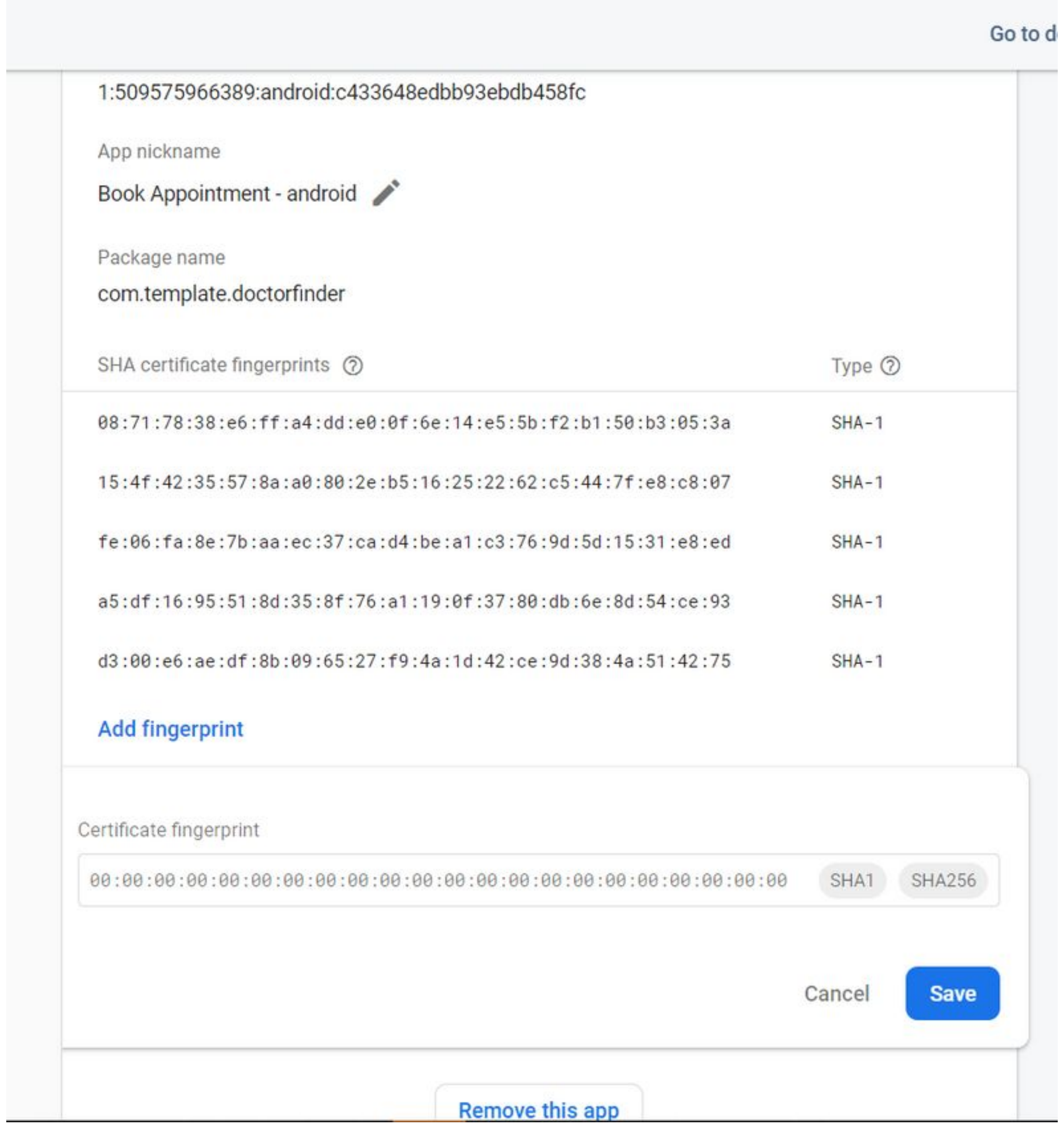
App nickname
Book Appointment - android

Package name
com.template.doctorfinder

SHA certificate fingerprints	Type
08:71:78:38:e6:ff:a4:dd:e0:0f:6e:14:e5:5b:f2:b1:50:b3:05:3a	SHA-1
15:4f:42:35:57:8a:a0:80:2e:b5:16:25:22:62:c5:44:7f:e8:c8:07	SHA-1
fe:06:fa:8e:7b:aa:ec:37:ca:d4:be:a1:c3:76:9d:5d:15:31:e8:ed	SHA-1
a5:df:16:95:51:8d:35:8f:76:a1:19:0f:37:80:db:6e:8d:54:ce:93	SHA-1
d3:00:e6:ae:df:8b:09:65:27:f9:4a:1d:42:ce:9d:38:4a:51:42:75	SHA-1

Step - 12

Paste your SHA1 key in here and click on save button



1:509575966389:android:c433648edbb93ebdb458fc

App nickname

Book Appointment - android

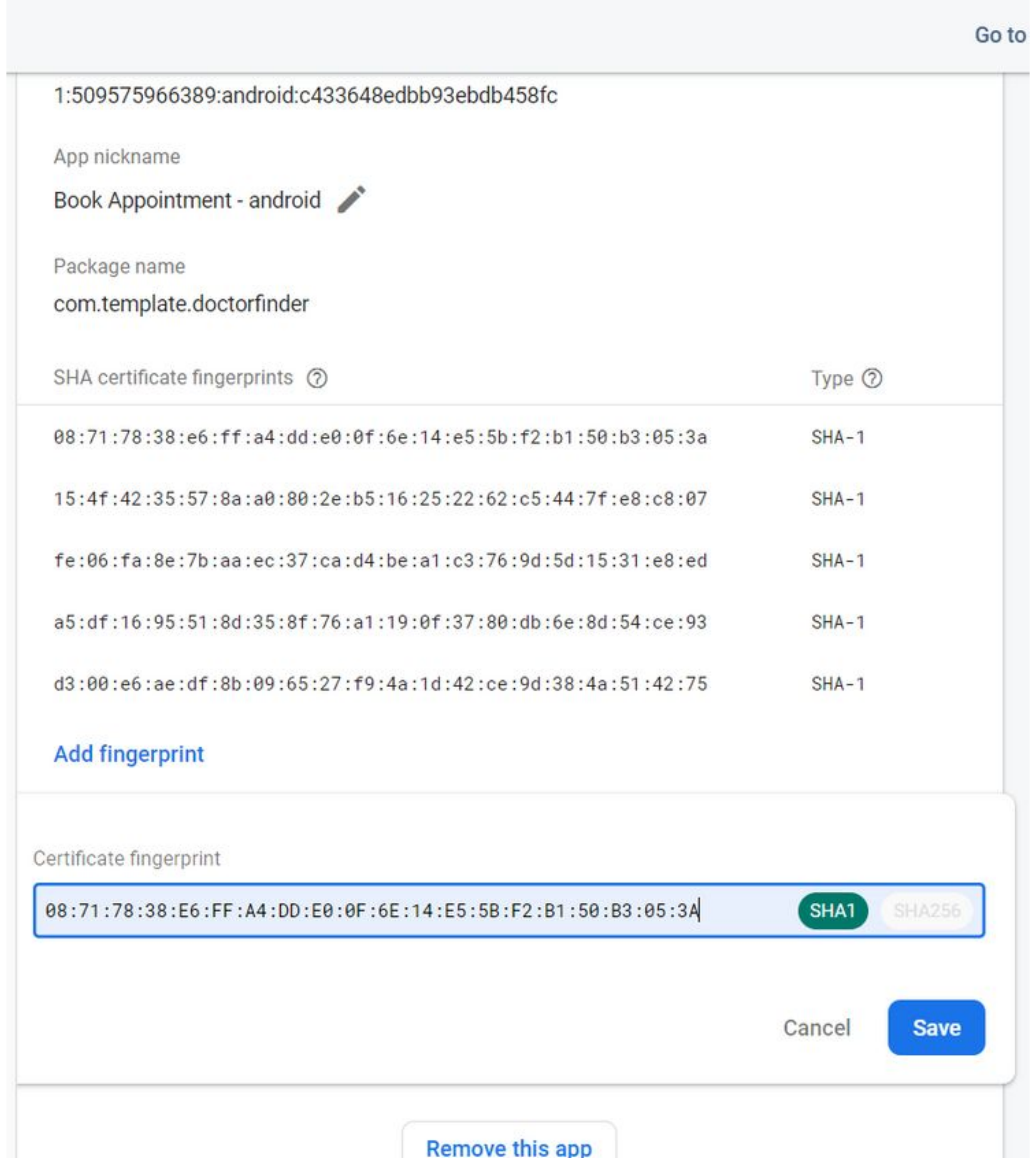
Package name

com.template.doctorfinder

SHA certificate fingerprints	Type
08:71:78:38:e6:ff:a4:dd:e0:0f:6e:14:e5:5b:f2:b1:50:b3:05:3a	SHA-1
15:4f:42:35:57:8a:a0:80:2e:b5:16:25:22:62:c5:44:7f:e8:c8:07	SHA-1
fe:06:fa:8e:7b:aa:ec:37:ca:d4:be:a1:c3:76:9d:5d:15:31:e8:ed	SHA-1
a5:df:16:95:51:8d:35:8f:76:a1:19:0f:37:80:db:6e:8d:54:ce:93	SHA-1
d3:00:e6:ae:df:8b:09:65:27:f9:4a:1d:42:ce:9d:38:4a:51:42:75	SHA-1

Certificate fingerprint

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00



1:509575966389:android:c433648edbb93ebdb458fc

App nickname

Book Appointment - android

Package name

com.template.doctorfinder

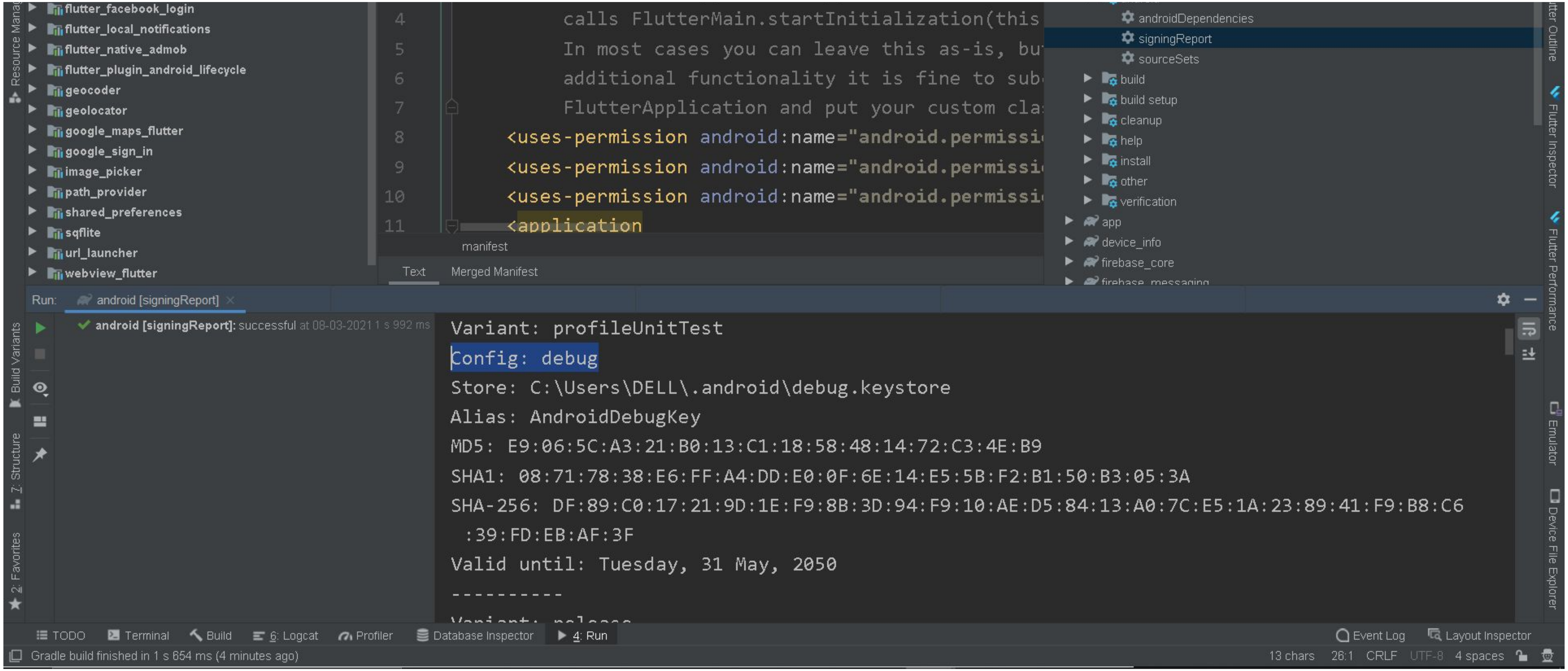
SHA certificate fingerprints	Type
08:71:78:38:e6:ff:a4:dd:e0:0f:6e:14:e5:5b:f2:b1:50:b3:05:3a	SHA-1
15:4f:42:35:57:8a:a0:80:2e:b5:16:25:22:62:c5:44:7f:e8:c8:07	SHA-1
fe:06:fa:8e:7b:aa:ec:37:ca:d4:be:a1:c3:76:9d:5d:15:31:e8:ed	SHA-1
a5:df:16:95:51:8d:35:8f:76:a1:19:0f:37:80:db:6e:8d:54:ce:93	SHA-1
d3:00:e6:ae:df:8b:09:65:27:f9:4a:1d:42:ce:9d:38:4a:51:42:75	SHA-1

Certificate fingerprint

08:71:78:38:E6:FF:A4:DD:E0:0F:6E:14:E5:5B:F2:B1:50:B3:05:3A

Step - 13

This time pick SHA1 with debug configuration and repeat step 8-12.



```
            calls FlutterMain.startInitialization(this
            In most cases you can leave this as-is, bu
            additional functionality it is fine to sub
            FlutterApplication and put your custom cla
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
    <uses-permission android:name="android.permissi
    <application
```

android [signingReport]: successful at 08-03-2021 1 s 992 ms

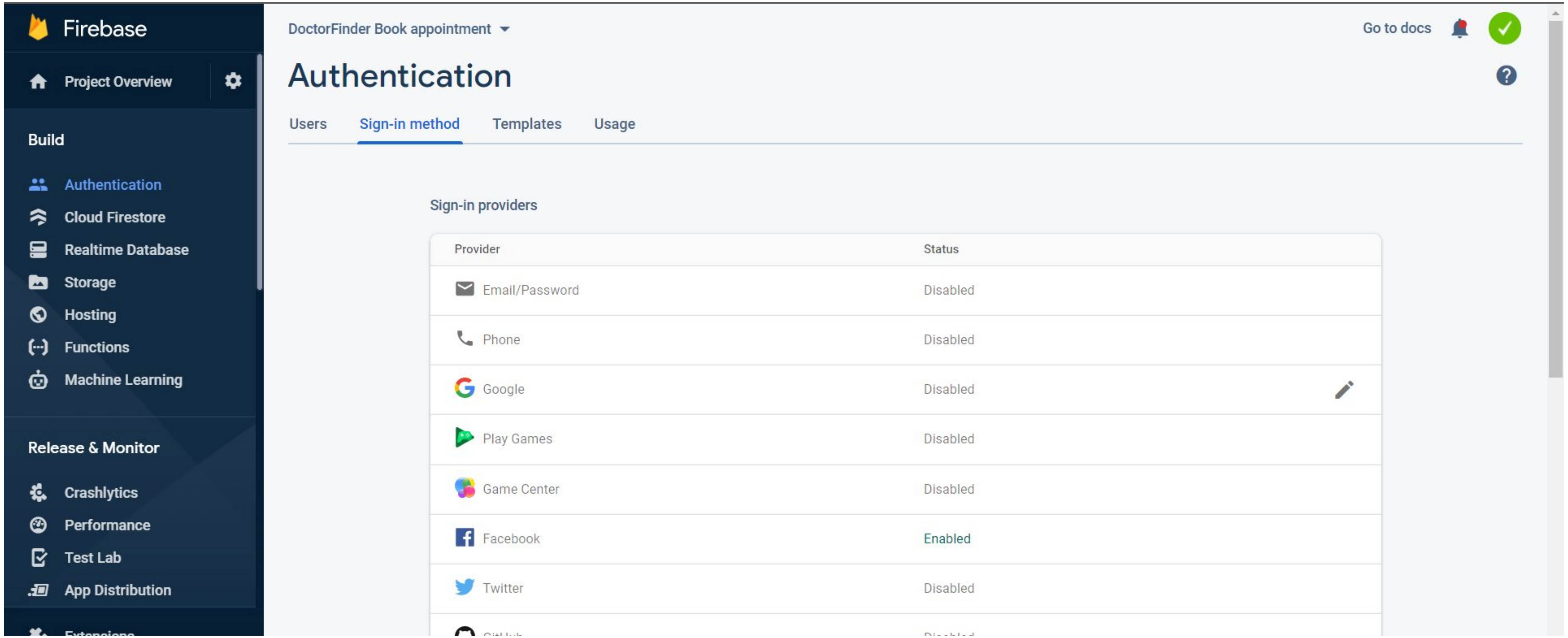
```
Variant: profileUnitTest
Config: debug
Store: C:\Users\DELL\.android\debug.keystore
Alias: AndroidDebugKey
MD5: E9:06:5C:A3:21:B0:13:C1:18:58:48:14:72:C3:4E:B9
SHA1: 08:71:78:38:E6:FF:A4:DD:E0:0F:6E:14:E5:5B:F2:B1:50:B3:05:3A
SHA-256: DF:89:C0:17:21:9D:1E:F9:8B:3D:94:F9:10:AE:D5:84:13:A0:7C:E5:1A:23:89:41:F9:B8:C6
 :39:FD:EB:AF:3F
Valid until: Tuesday, 31 May, 2050
----------
Variant: release
```

Gradle build finished in 1 s 654 ms (4 minutes ago)

Step - 14

In your firebase project console navigate to "Sign In Method".

Build -> Authentication -> Sign In Method

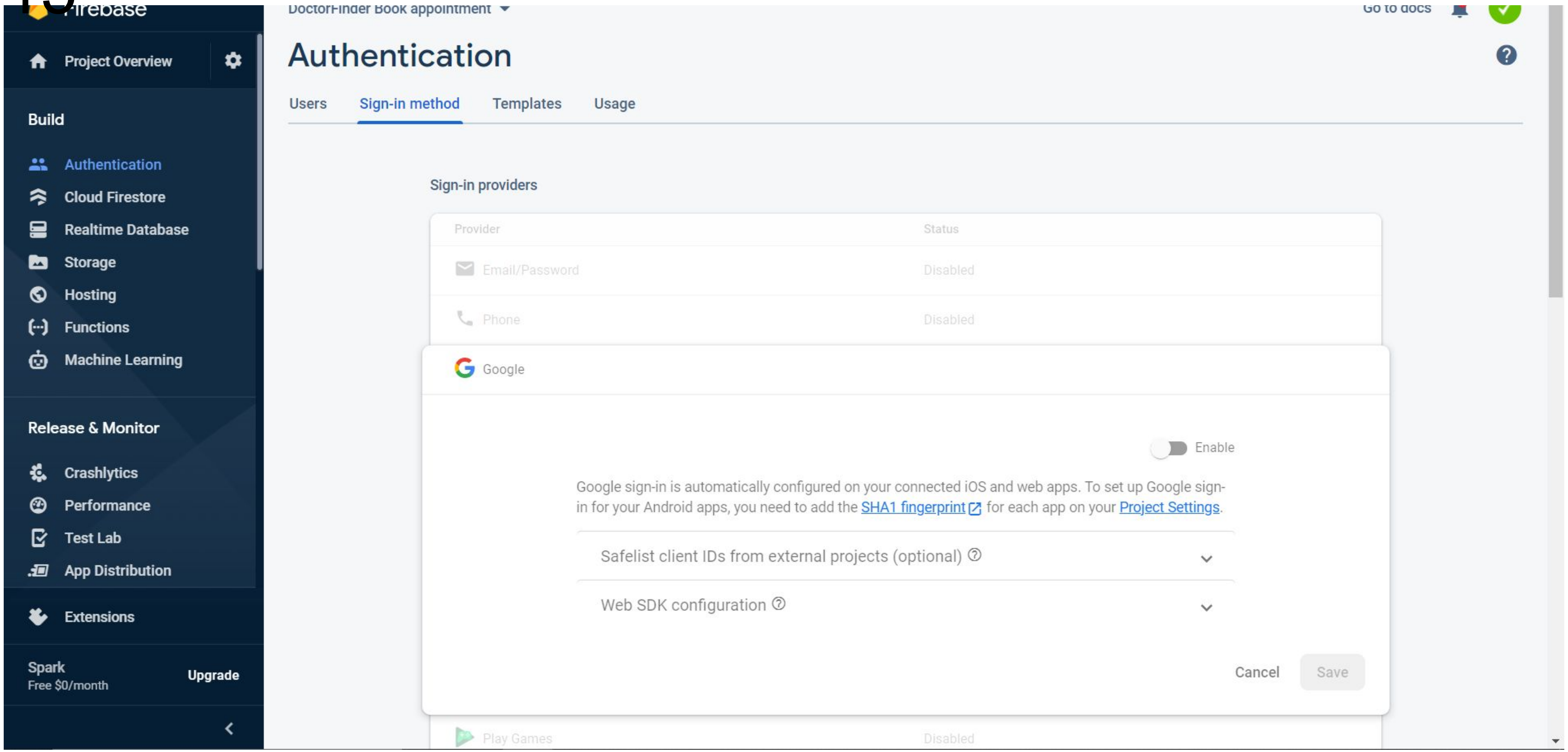


Sign-in providers

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	Disabled
Play Games	Disabled
Game Center	Disabled
Facebook	Enabled
Twitter	Disabled
GitHub	Disabled

Step - 15

Select "Google" in here and click on Enable Button.



DoctorFinder Book appointment

Authentication

Sign-in providers

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	
Play Games	Disabled

Enable

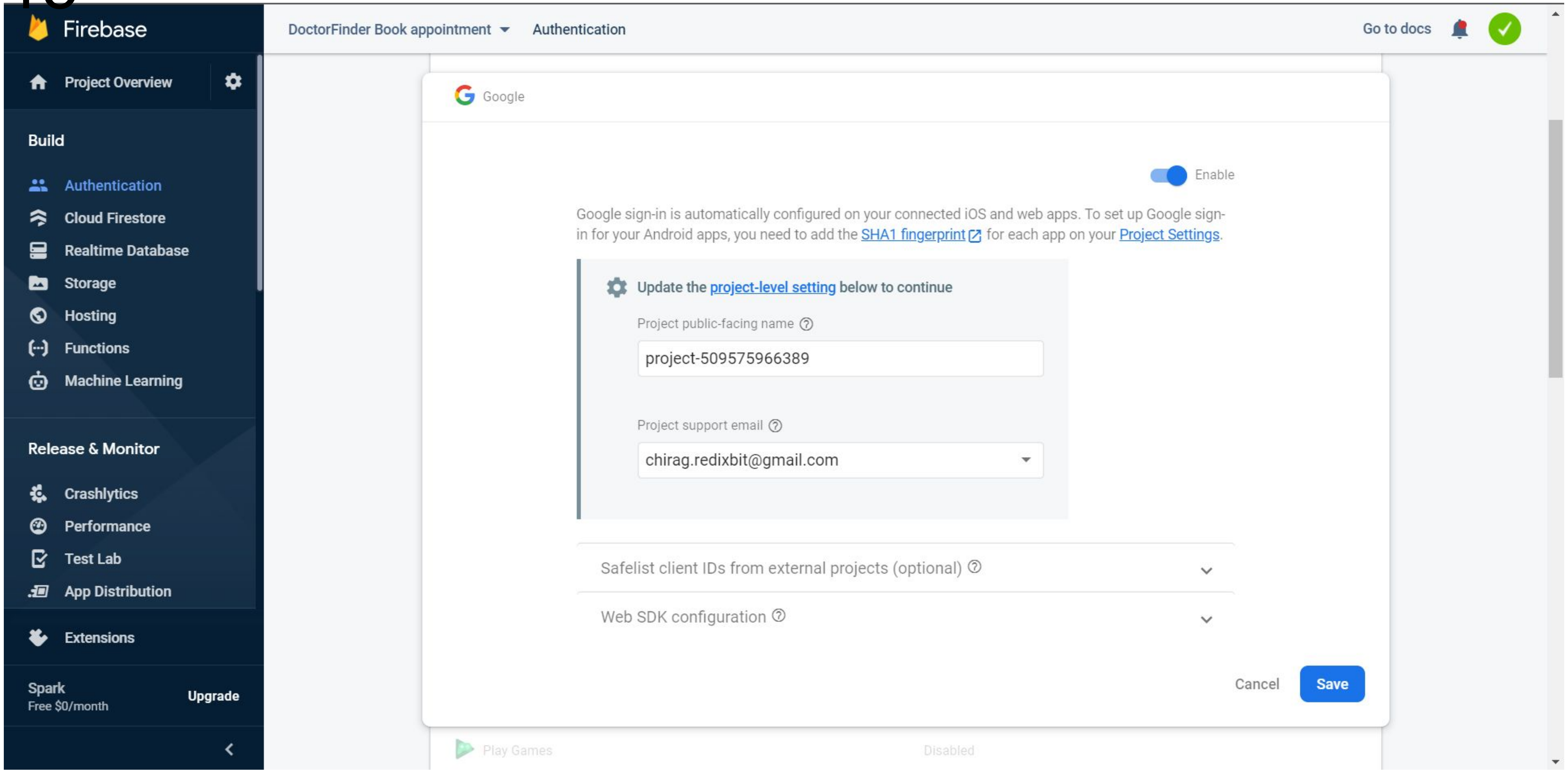
Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the SHA1 fingerprint for each app on your Project Settings.

Safelist client IDs from external projects (optional)

Web SDK configuration

Step - 15

Add "Support Email" and click on save button.



Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the SHA1 fingerprint for each app on your Project Settings.

Update the project-level setting below to continue

Project public-facing name: project-509575966389

Project support email: chirag.redixbit@gmail.com

Your App is ready for Google Login