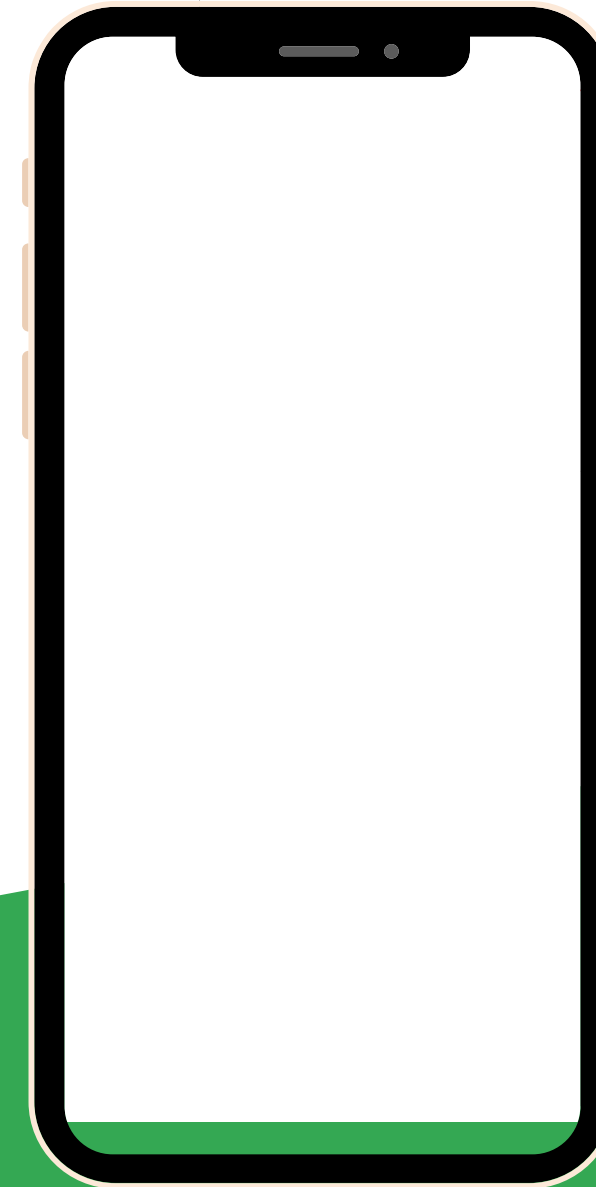


Google Login Setup In Flutter

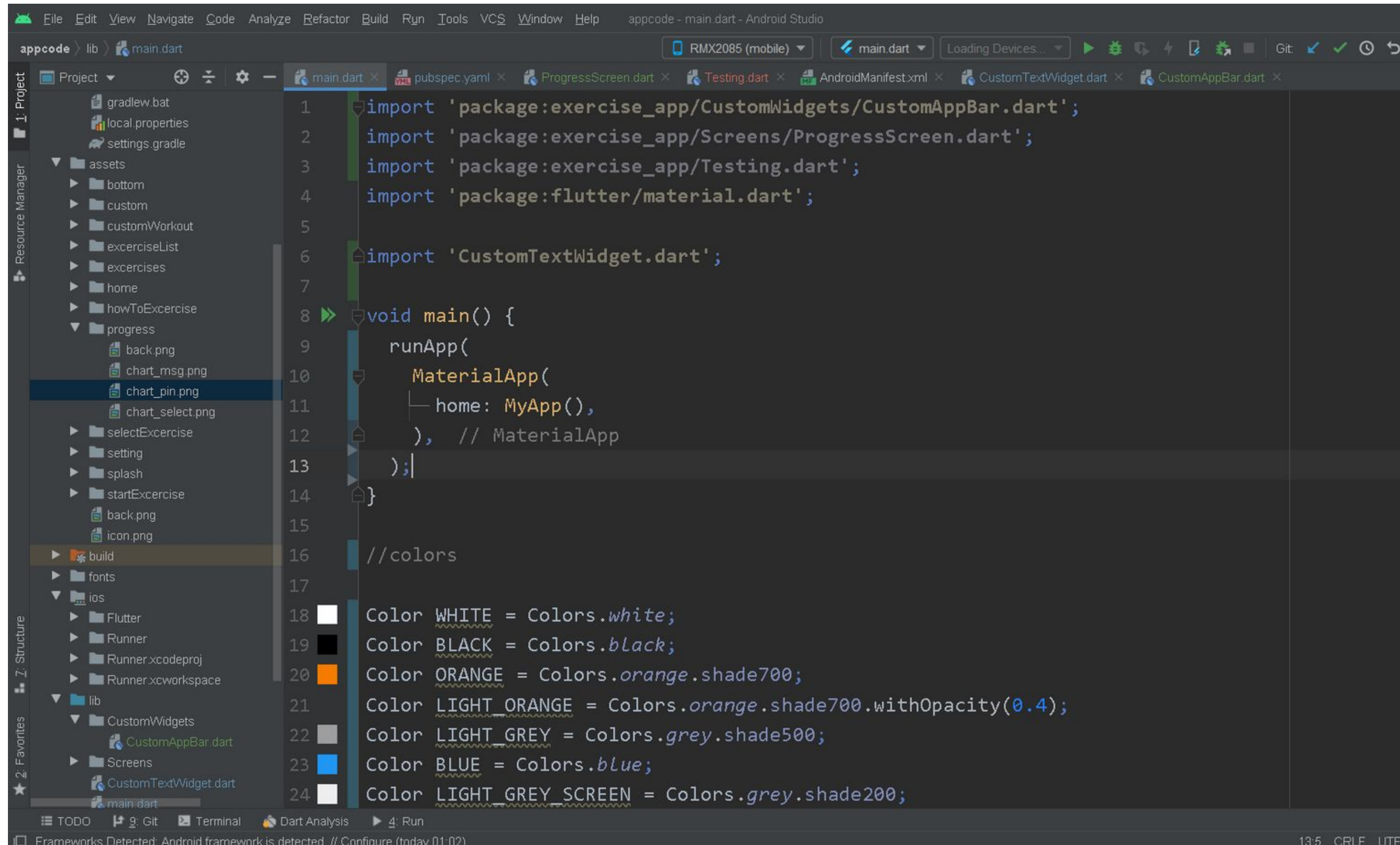
Requirements:-

- 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



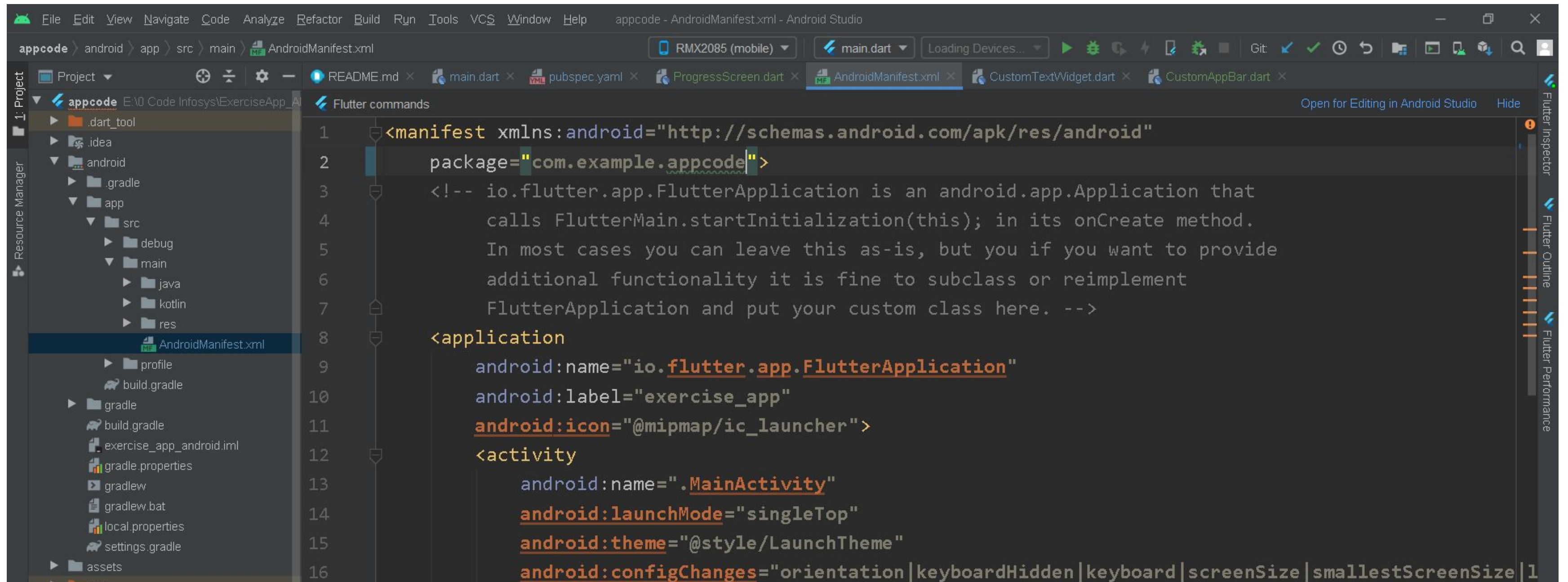
```
1 import 'package:exercise_app/CustomWidgets/CustomAppBar.dart';
2 import 'package:exercise_app/Screens/ProgressScreen.dart';
3 import 'package:exercise_app/Testing.dart';
4 import 'package:flutter/material.dart';
5
6 import 'CustomTextWidget.dart';
7
8 void main() {
9   runApp(
10     MaterialApp(
11       home: MyApp(),
12     ), // MaterialApp
13   );
14 }
15
16 //colors
17
18 Color WHITE = Colors.white;
19 Color BLACK = Colors.black;
20 Color ORANGE = Colors.orange.shade700;
21 Color LIGHT_ORANGE = Colors.orange.shade700.withOpacity(0.4);
22 Color LIGHT_GREY = Colors.grey.shade500;
23 Color BLUE = Colors.blue;
24 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

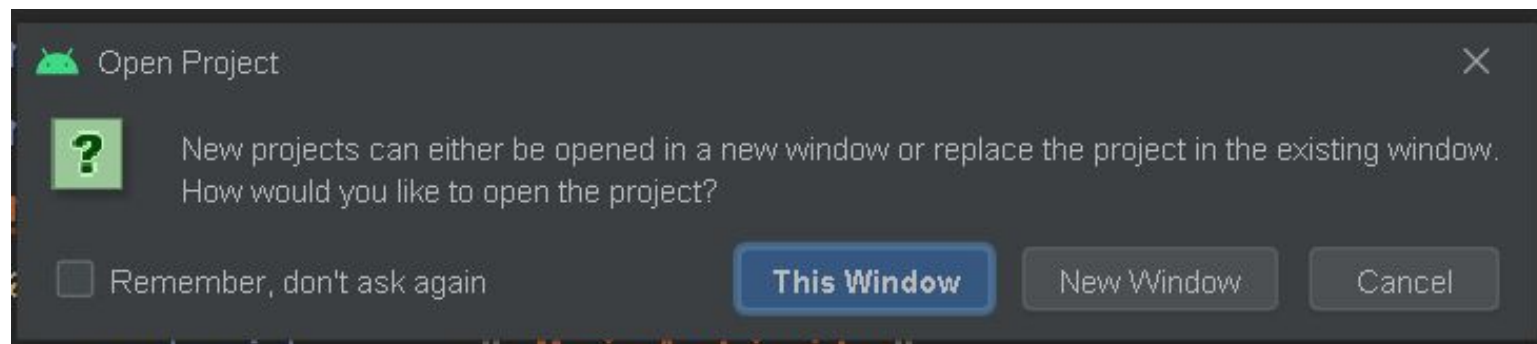
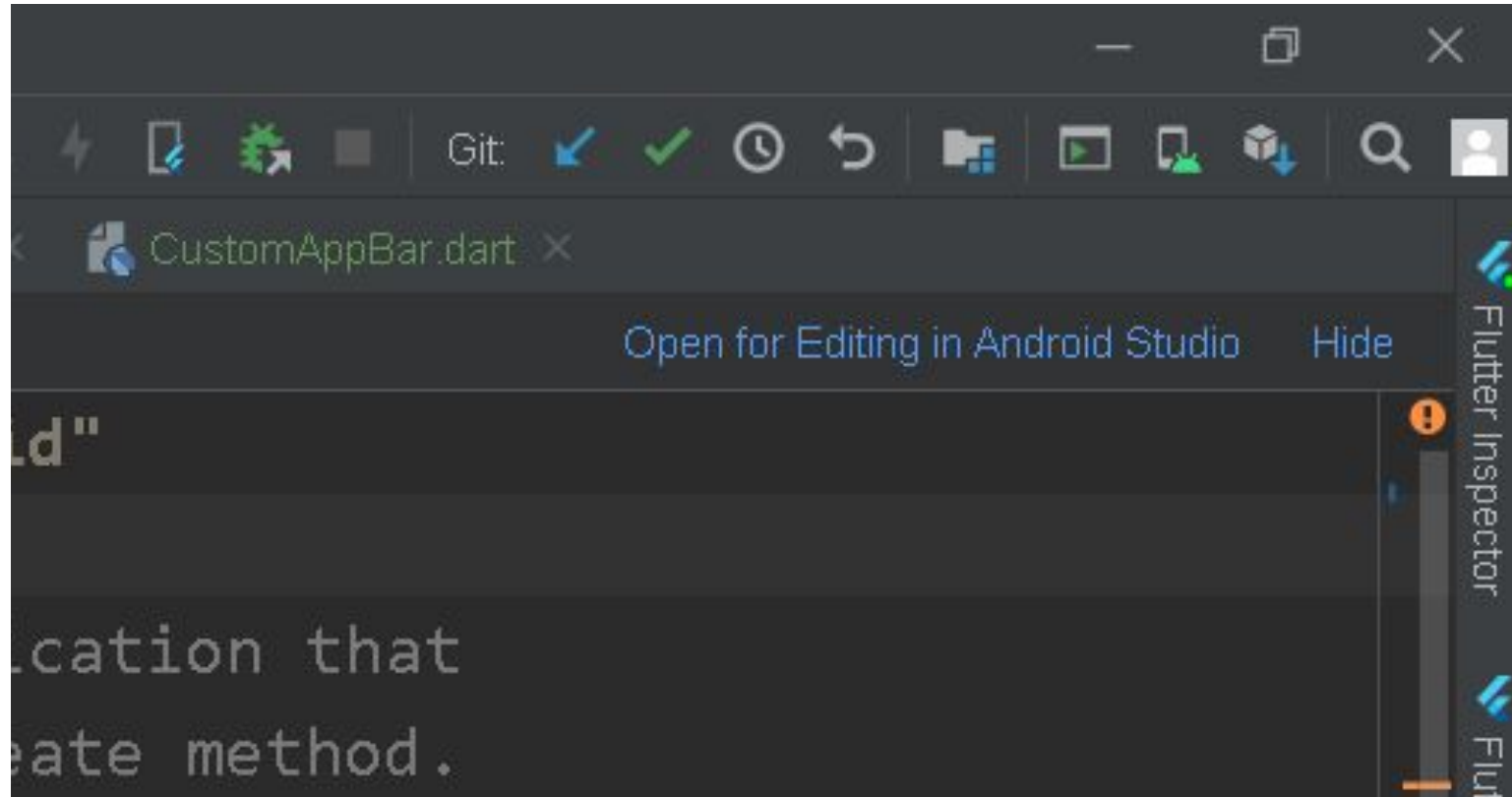
Navigate to AndroidManifest.xml file in your project

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

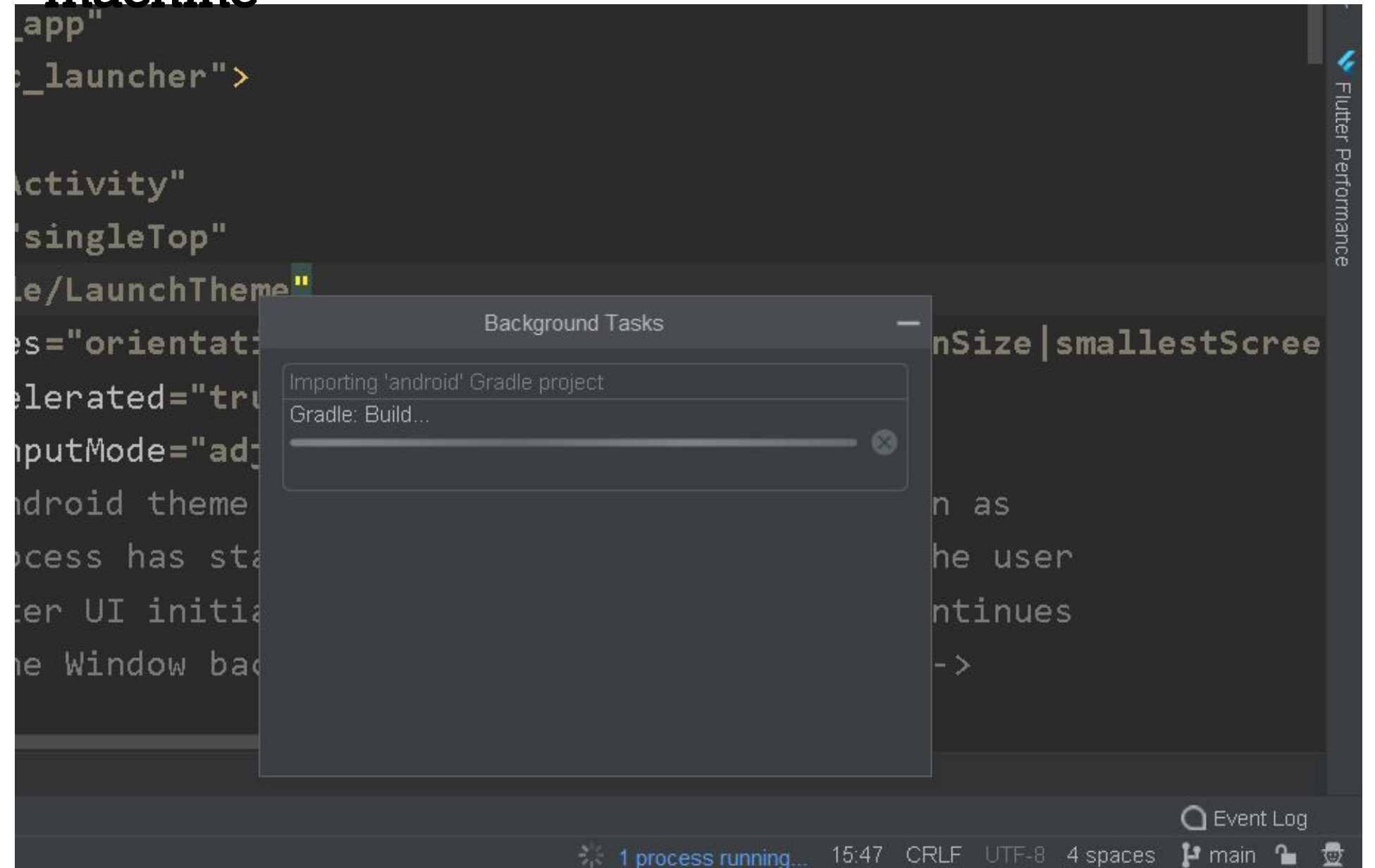


Step - 3

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

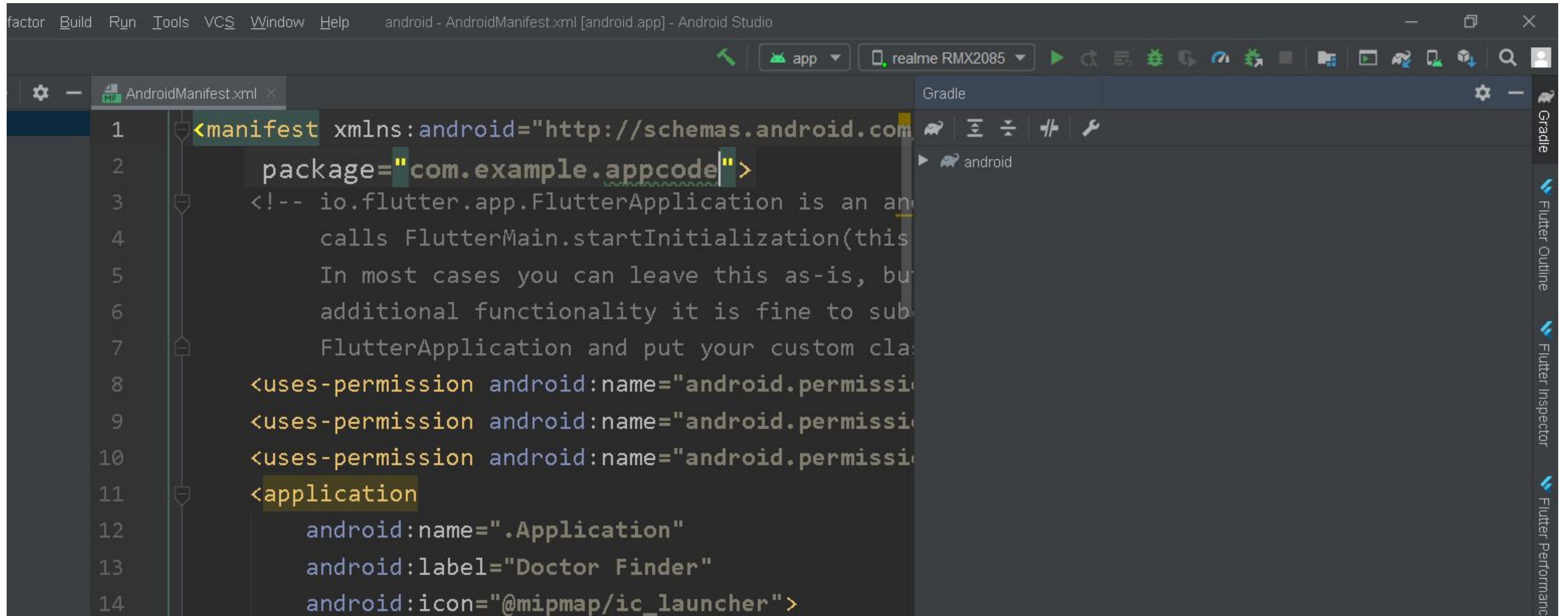


Wait while gradle is building this process is going to take upto 5-10 minutes depending on the performance of your machine



Step - 4

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

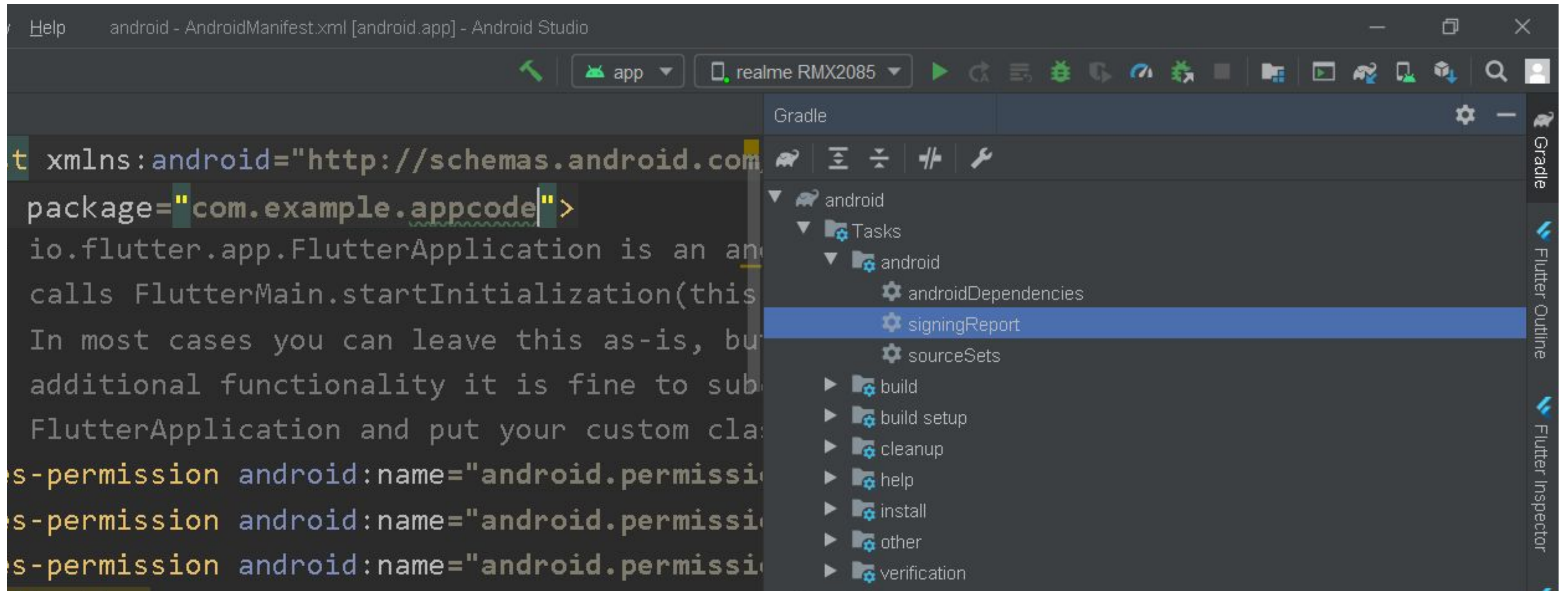


Step - 5

Navigate to "signingReport"

android -> Tasks -> android -> signingReport

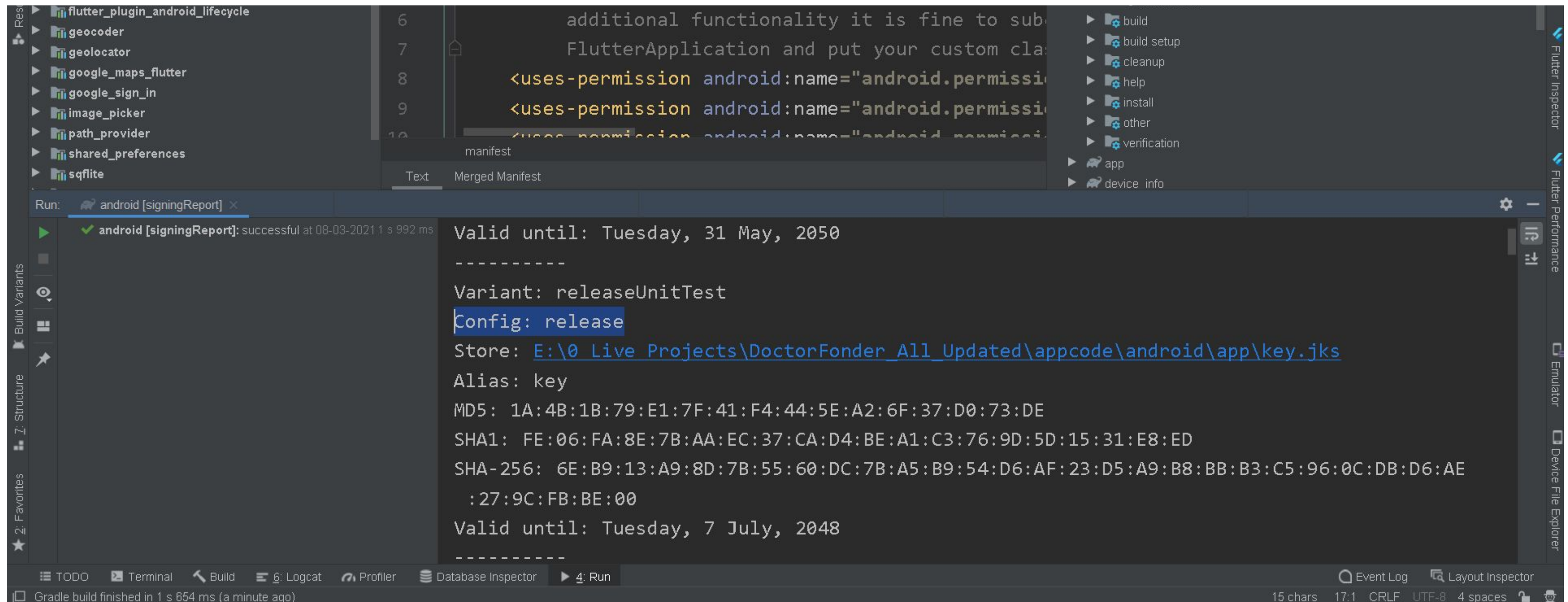
double click on it and keep an eye on the console



Step - 6

if your project contains .jks 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).

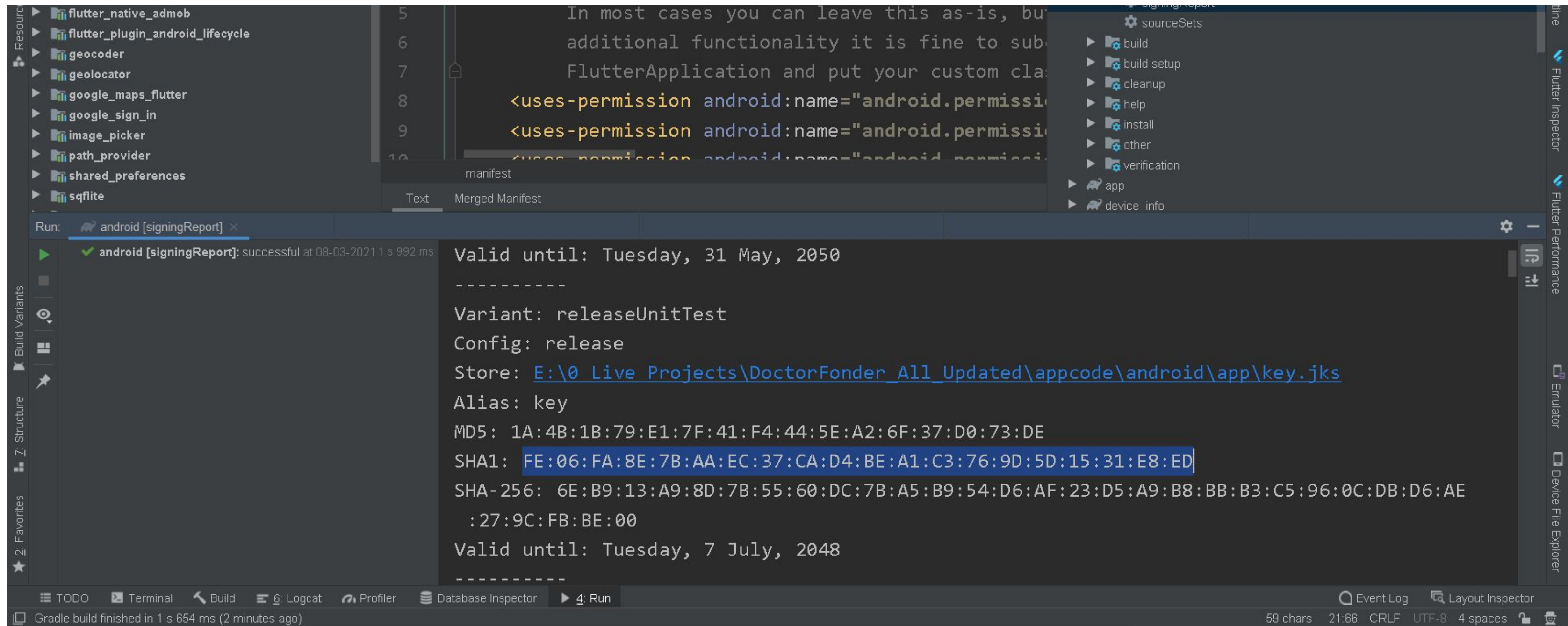


The screenshot shows an IDE interface with a project structure on the left, a code editor in the center, and a console at the bottom. The console displays the output of an Android signing report for a release build. The report includes the variant name, configuration, keystore path, alias, and both MD5 and SHA1 hashes for the release key. The SHA1 hash is highlighted in blue in the original image.

```
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
-----
```

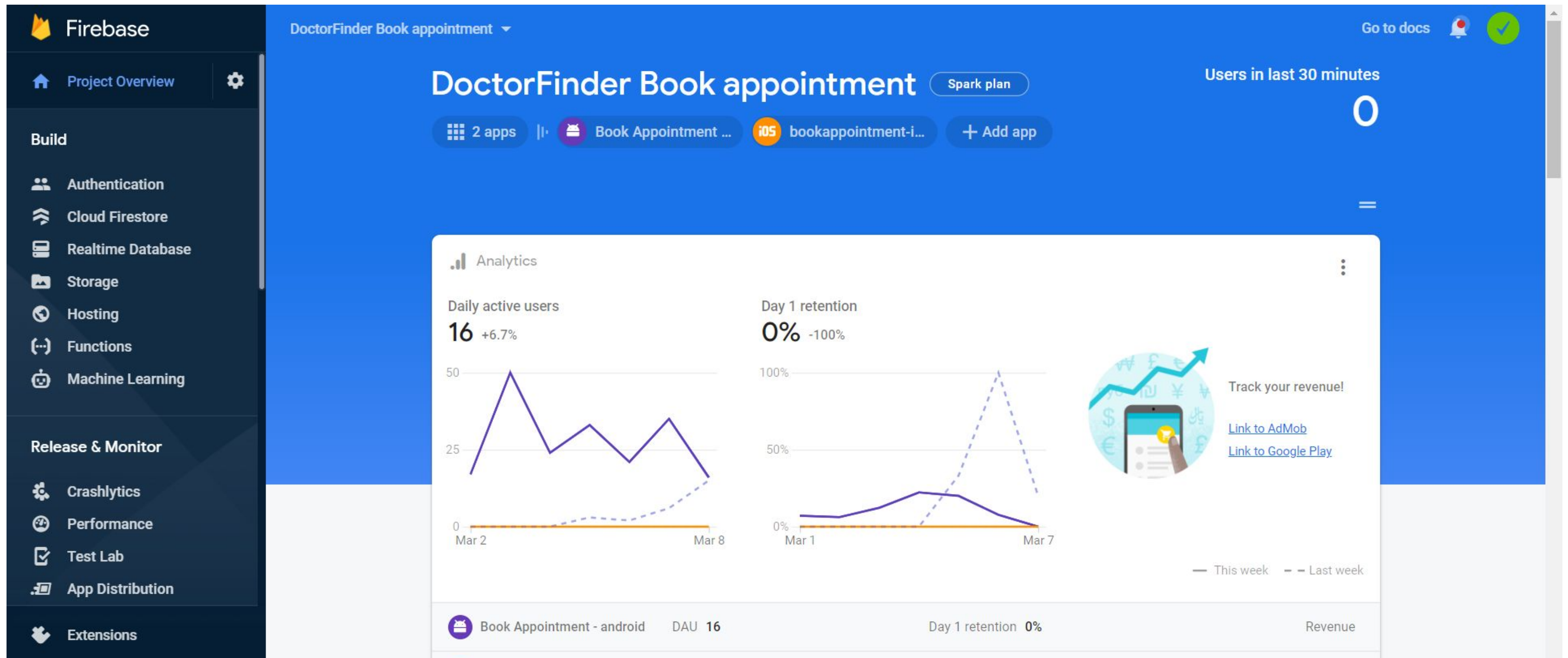

Step - 7

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



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.

The screenshot shows the Firebase console interface for the project "DoctorFinder Book appointment". The left sidebar contains the "Build" menu with options: Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning. The main content area displays the project name and a list of two apps: "Book Appointment - android" (com.template.doctorfinder) and "bookappointment-ios" (com.template.bookAppointment). A modal window is open showing the app list. Below the app list, there are two line charts: one for "Users in last 30 minutes" and another for "Day 1 retention". A "Track your revenue!" section with links to AdMob and Google Play is also visible.

2 apps visible (max 2) 2 apps in project

Book Appointment - android
com.template.doctorfinder

bookappointment-ios
com.template.bookAppointment

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

Users in last 30 minutes
0

Track your revenue!
[Link to AdMob](#)
[Link to Google Play](#)

— This week — Last week

Step -

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

The screenshot shows the Firebase Project settings page. On the left is a dark blue sidebar with the 'Firebase' logo and navigation links: 'Project Overview' (active), 'Build' (containing Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, and Machine Learning), 'Release & Monitor' (containing Crashlytics, Performance, Test Lab, and App Distribution), and 'Extensions'. The main content area is titled 'Project settings' for the project 'DoctorFinder Book appointment'. It has tabs for 'General' (selected), 'Cloud Messaging', 'Integrations', 'Service accounts', 'Data privacy', and 'Users and permissions'. Under the 'General' tab, there are two sections: 'Your project' and 'Public settings'. The 'Your project' section contains a table of project details:

| | |
|-------------------------------|---|
| Project name | DoctorFinder Book appointment |
| Project ID | doctorfinder-book-appointment |
| Project number | 509575966389 |
| Default GCP resource location | nam5 (us-central) |
| Web API Key | AlzaSyAhIX00ANu8t_0MA0FHVnxfxApgw8n4Nz4 |

The 'Public settings' section has a subtitle 'These settings control instances of your project shown to the public' and contains:

| | |
|--------------------|--|
| Public-facing name | project-509575966389 |
| Support email | <input type="text" value="chirag.redixbit@gmail.com"/> |

At the top right of the settings page, there is a 'Go to docs' link, a notification bell icon, a green checkmark icon, and a help icon.

Step -

Click on add fingerprint button.

The screenshot shows the Firebase Project settings page for a project named 'DoctorFinder Book appointment'. The left sidebar contains navigation links for Project Overview, Build (Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor (Crashlytics, Performance, Test Lab, App Distribution), and Extensions. The main content area is divided into two columns. The left column lists apps: 'Book Appointment - android' (com.template.doctorfinder) and 'bookappointment-ios' (com.template.bookAppointment). The right column shows the 'SDK setup and configuration' for the selected Android app. It includes instructions to reconfigure SDKs, a link to 'See SDK instructions', and a download link for 'google-services.json'. Below this, the 'App ID' is '1:509575966389:android:c433648edbb93ebdb458fc', the 'App nickname' is 'Book Appointment - android', and the 'Package name' is 'com.template.doctorfinder'. A table of 'SHA certificate fingerprints' is displayed, with five rows of fingerprints and their corresponding 'SHA-1' types. A red box highlights the 'Add fingerprint' button at the bottom of the table.

| 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 |


[Add fingerprint](#)

Paste your SHA1 key in here and click on save button

Go to


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

Add fingerprint

Certificate fingerprint

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

SHA1

SHA256

Cancel

Save

Remove this app

Step -

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

13

The screenshot shows the Android Studio interface. On the left, the 'Resource Manager' pane lists various Flutter plugins. The main editor displays the 'manifest' file with XML code. The 'Run' tab at the bottom shows a successful signing report for the 'profileUnitTest' variant in 'debug' configuration. The report includes the keystore path, alias, and various hashes (MD5, SHA1, SHA-256) along with the validity date.

```
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.permission
<uses-permission android:name="android.permission
<uses-permission android:name="android.permission
<application
```

Run: android [signingReport] x

✓ 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

13 chars 26:1 CRLF UTF-8 4 spaces

Step -

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

Build -> Authentication -> Sign In Method

The screenshot displays the Firebase Authentication console for the project 'DoctorFinder Book appointment'. The 'Sign-in method' tab is active, showing a table of configured providers. The 'Facebook' provider is the only one that is 'Enabled', while all other providers are 'Disabled'.

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

Step -

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

15

The screenshot shows the Firebase Authentication console for a project named "Doctor-Finder Book appointment". The left sidebar contains navigation links for Project Overview, Build (Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor (Crashlytics, Performance, Test Lab, App Distribution), and Extensions. The main content area is titled "Authentication" and has tabs for Users, Sign-in method (selected), Templates, and Usage. Under the "Sign-in providers" section, a table lists providers: Email/Password (Disabled), Phone (Disabled), Google (selected), and Play Games (Disabled). The Google provider's configuration modal is open, showing an "Enable" toggle that is switched on. Below the toggle, there is explanatory text about Google sign-in configuration for iOS, web, and Android apps, followed by expandable sections for "Safelist client IDs from external projects (optional)" and "Web SDK configuration". At the bottom right of the modal are "Cancel" and "Save" buttons.

| Provider | Status |
|----------------|----------|
| Email/Password | Disabled |
| Phone | Disabled |
| Google | Enabled |
| Play Games | Disabled |

Google

☒ 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 [?](#)

Cancel Save

Step -

15 Add "Support Email" and click on save button.

The screenshot shows the Firebase console interface for the 'DoctorFinder Book appointment' project, specifically the 'Authentication' settings page. A modal dialog is open for configuring Google sign-in. At the top right of the modal, there is a toggle switch labeled 'Enable' which is currently turned on. Below this, a text block states: '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](#).' The main section of the modal is titled 'Update the [project-level setting](#) below to continue'. It contains two input fields: 'Project public-facing name' with the value 'project-509575966389' and 'Project support email' with the value 'chirag.redixbit@gmail.com'. Below these are two expandable sections: 'Safelist client IDs from external projects (optional)' and 'Web SDK configuration', both currently collapsed. At the bottom right of the modal are 'Cancel' and 'Save' buttons. The background shows the Firebase sidebar with 'Authentication' selected, and the top navigation bar with 'Go to docs', a notification bell, and a green checkmark icon.

Your App is ready for
Google Login

