



Simple World Documentation

Simple World is a Social App built in Flutter and Firebase with both android and ios app. This application can be used to post pictures in timeline, chat with other users and can be logged in using google accounts.

Features

- Users can use Email and Google Authentication to login.
- Posts Images and Text
- Upload Videos and PDF
- Users can Chat with each other.
- Users can Add Stories
- Users can see all users using application from user's page
- Users can see all users posts from posts page
- Profile account with edit capability.
- Change profile picture.
- Add and Change cover picture.
- User can Comment and like each other's post.
- Users can Follow and unfollow each other to see each other's posts in their timeline.
- Users Searching setup
- Push Notification via FCM.
- Simple UI

Project Setup

Install Flutter:

Visit flutter official website: flutter.dev for full install guide. if you prefer video tutorials, we recommend this playlist that will guide throw the full installation process:

<https://www.youtube.com/playlist?list=PLSzsOkUDsvdtl3Pw48-R8lcK2oYkk40cm>

Set up your editor Install the Flutter and Dart plugins.

To install and get ready to build IOS application visit this [link](#).

Please make sure you have the latest flutter stable version.

Setup:

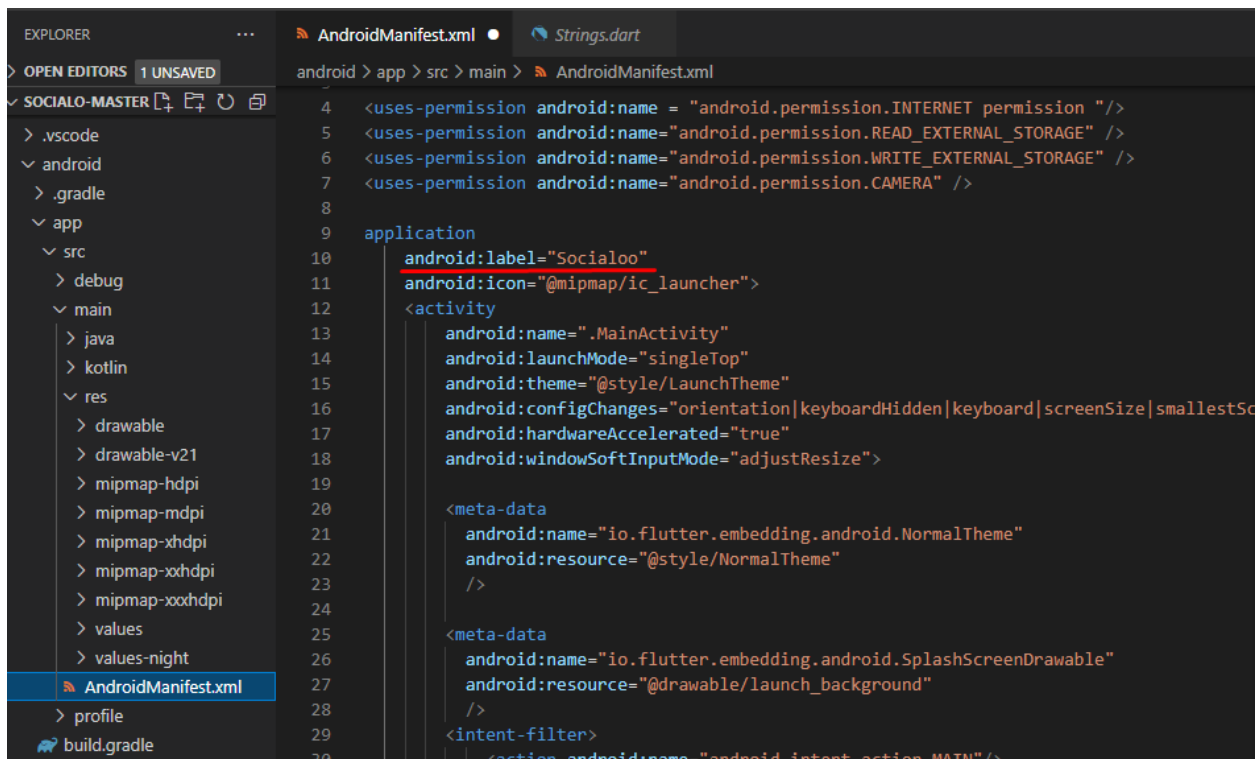
1. Open Code in visual studio.
2. To get the packages run the following command

Command: `flutter pub get`

3. To change the app icon replace icon.png in assets/ folder and run the command below it would change icons for both Android and IOS

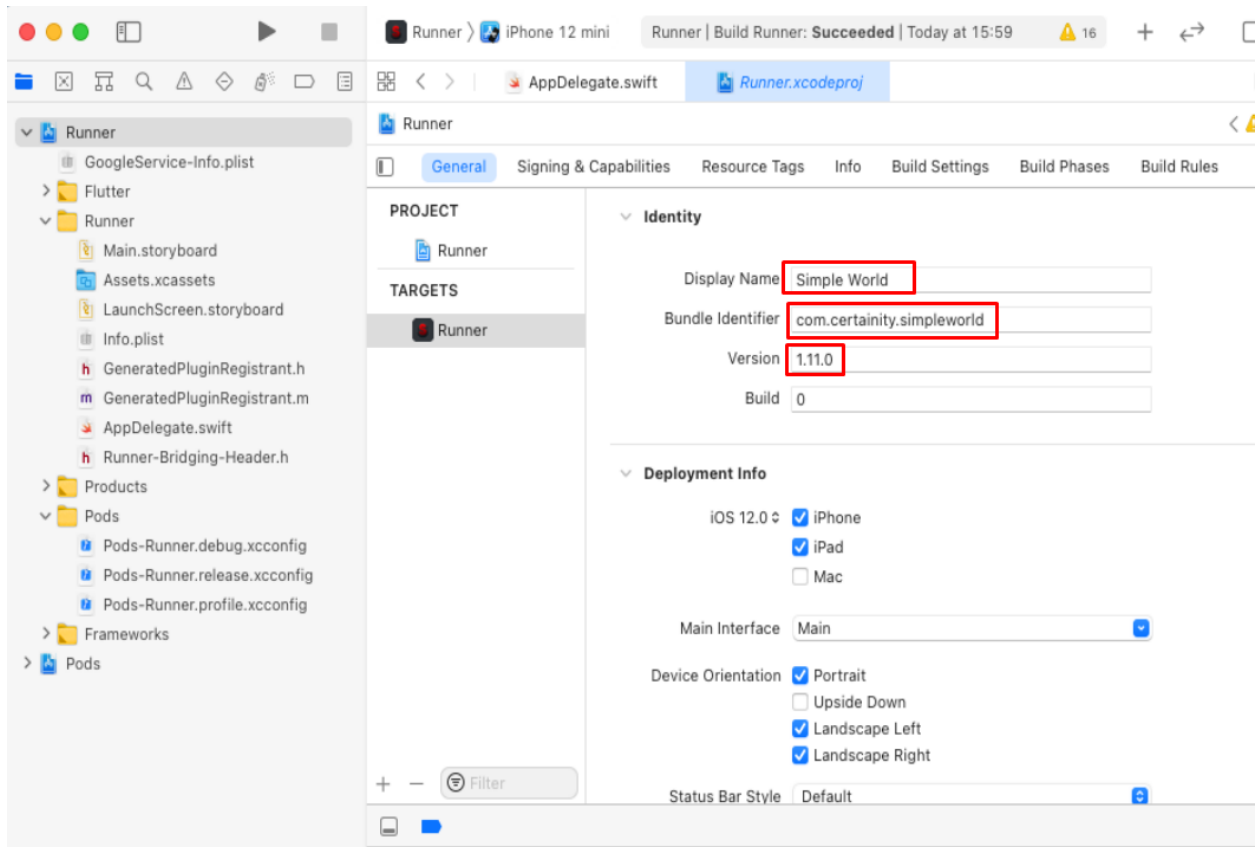
Command: `flutter pub run flutter_launcher_icons:main`

4. To change the android app name
 - Change android label in AndroidManifest.xml file



```
EXPLORER
...
OPEN EDITORS 1 UNSAVED
SOCIALO-MASTER
  .vscode
  android
  .gradle
  app
    src
      debug
      main
        java
        kotlin
        res
          drawable
          drawable-v21
          mipmap-hdpi
          mipmap-mdpi
          mipmap-xhdpi
          mipmap-xxhdpi
          values
          values-night
      AndroidManifest.xml
      profile
      build.gradle
    AndroidManifest.xml
4  <uses-permission android:name = "android.permission.INTERNET" />
5  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
6  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
7  <uses-permission android:name="android.permission.CAMERA" />
8
9  application
10     android:label="Socialoo"
11     android:icon="@mipmap/ic_launcher">
12     <activity
13         android:name=".MainActivity"
14         android:launchMode="singleTop"
15         android:theme="@style/LaunchTheme"
16         android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|uiMode"
17         android:hardwareAccelerated="true"
18         android:windowSoftInputMode="adjustResize">
19
20         <meta-data
21             android:name="io.flutter.embedding.android.NormalTheme"
22             android:resource="@style/NormalTheme"
23             />
24
25         <meta-data
26             android:name="io.flutter.embedding.android.SplashScreenDrawable"
27             android:resource="@drawable/launch_background"
28             />
29         <intent-filter>
30             <action android:name="android.intent.action.MAIN" />
```

5. To change IOS app name
 - Open the default Xcode workspace in your project by running `open ios/Runner.xcworkspace` in a terminal window from your Flutter project directory.
 - Within Xcode change Display name, bundle id and version as need



6. Customize app theme

- To customize your app theme, go to app.dart file in the main lib folder, change the colors as you prefer.
- Color codes related to theme are saved in palette.dart file in lib/config folder

```
lib > config > palette.dart > Palette
2
3  class Palette {
4    // light theme
5    static const Color primaryColor = Color(0xFFC62828);
6    static const Color apptitlecolor = Color(0xFFC62828);
7    static const Color backgroundColor = Colors.white;
8    static const Color appBarbackgroundColor = Colors.white;
9    static const Color appBariconcolor = Color(0xff495057);
10   static const Color tabBarlabelColor = Colors.black;
11   static const Color iconThemeColor = Colors.black54;
12   static const Color scaffoldBackgroundColor = Color(0xffff2f6f9);
13   static const Color cardColor = Colors.white;
14   static const Color shadowColor = Color(0xFFE0E0E0);
15   static const Color inputfillcolor = Color(0xffff8f8f8);
16
17   // light theme
18   static const Color apptitlecolordark = Colors.white;
19   static const Color backgroundColordark = Color(0xff464c52);
20   static const Color appBarbackgroundColordark = Color(0xff242526);
21   static const Color appBariconcolordark = Color(0xffffffff);
22   static const Color tabBarlabelColordark = Colors.white;
23   static const Color iconThemeColordark = Colors.white70;
24   static const Color scaffoldBackgroundColordark = Color(0xFF18191A);
25   static const Color cardColordark = Color(0xFF3A3B3C);
26   static const Color shadowColordark = Color(0xFF424242);
27   static const Color inputfillcolordark = Color(0xff242526);
28
29   static const Color simpleWorldText = Color(0xFF161f3d);
30
```

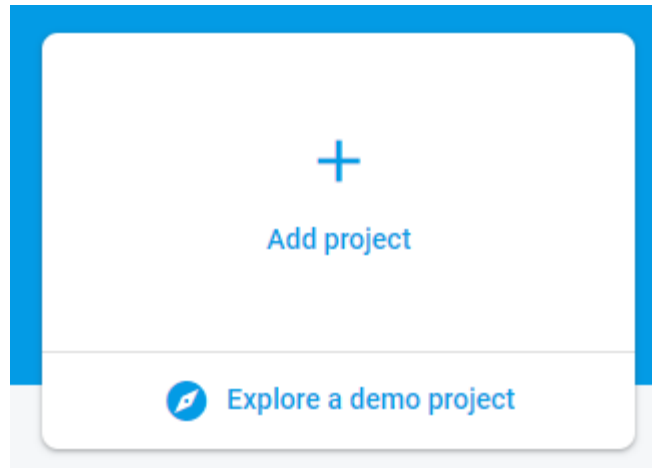
Create a Firebase Project

To know more about how to add Firebase to your Android project visit the link below:

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

Step by step guide to make firebase setting to use this app

1. Go to Firebase dashboard, select the “Create new project” button and give it a name:




2. Give a name for your project and click to continue

× Create a project (Step 1 of 3)

Let's start with a name for
your project?

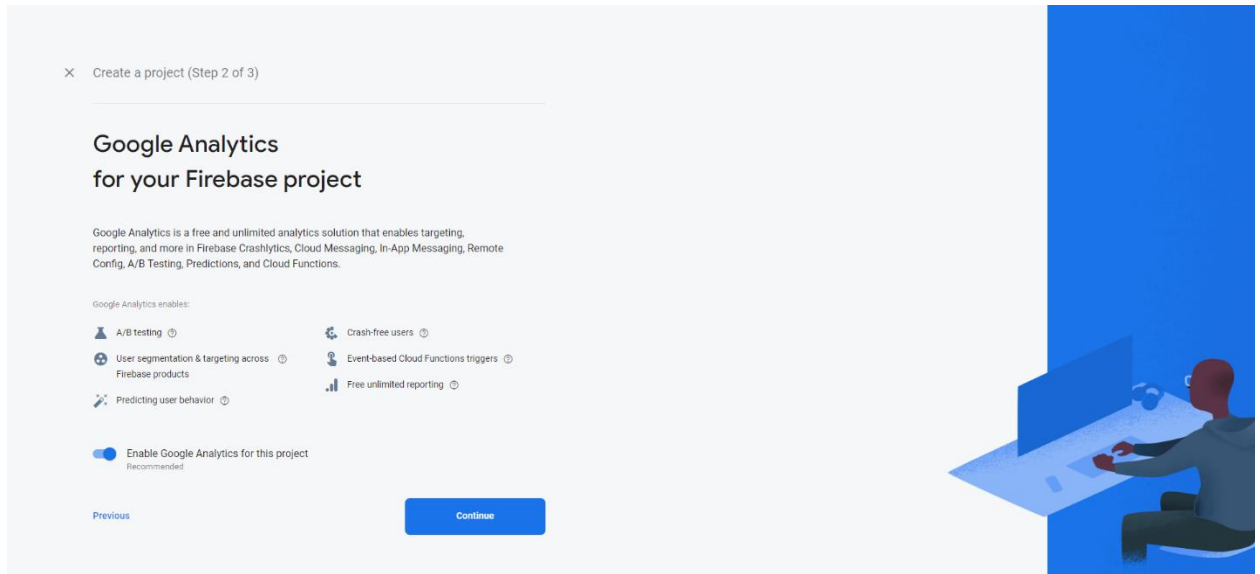
Project name

Socialoo

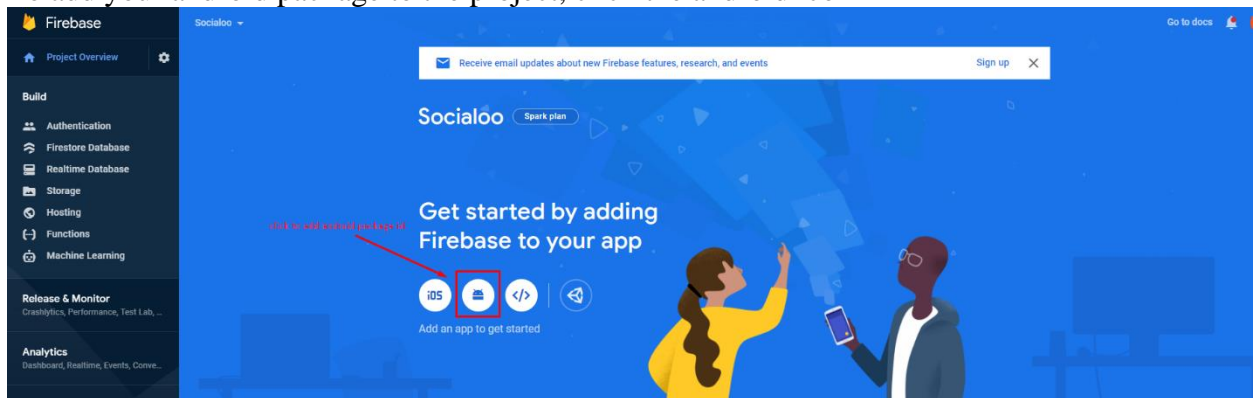
 socialoo-97c50

Continue

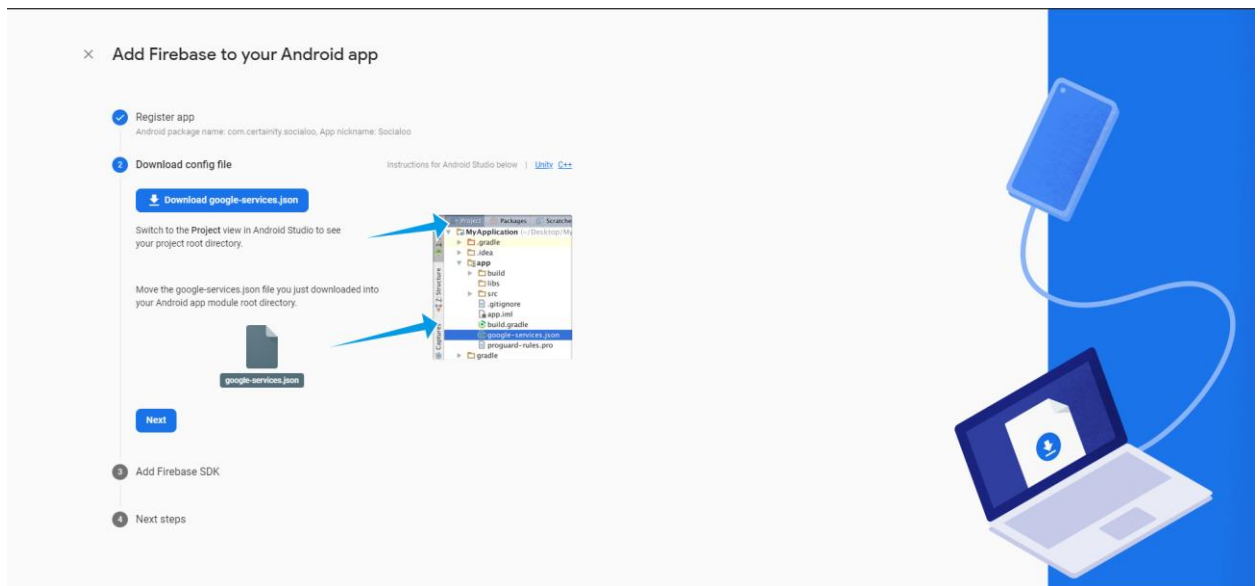
3. After that, we have the choice of enabling Google Analytics. It isn't required for what we're trying to do, so do whatever seem right for your use-case.



4. To add your android package to the project, click the android icon

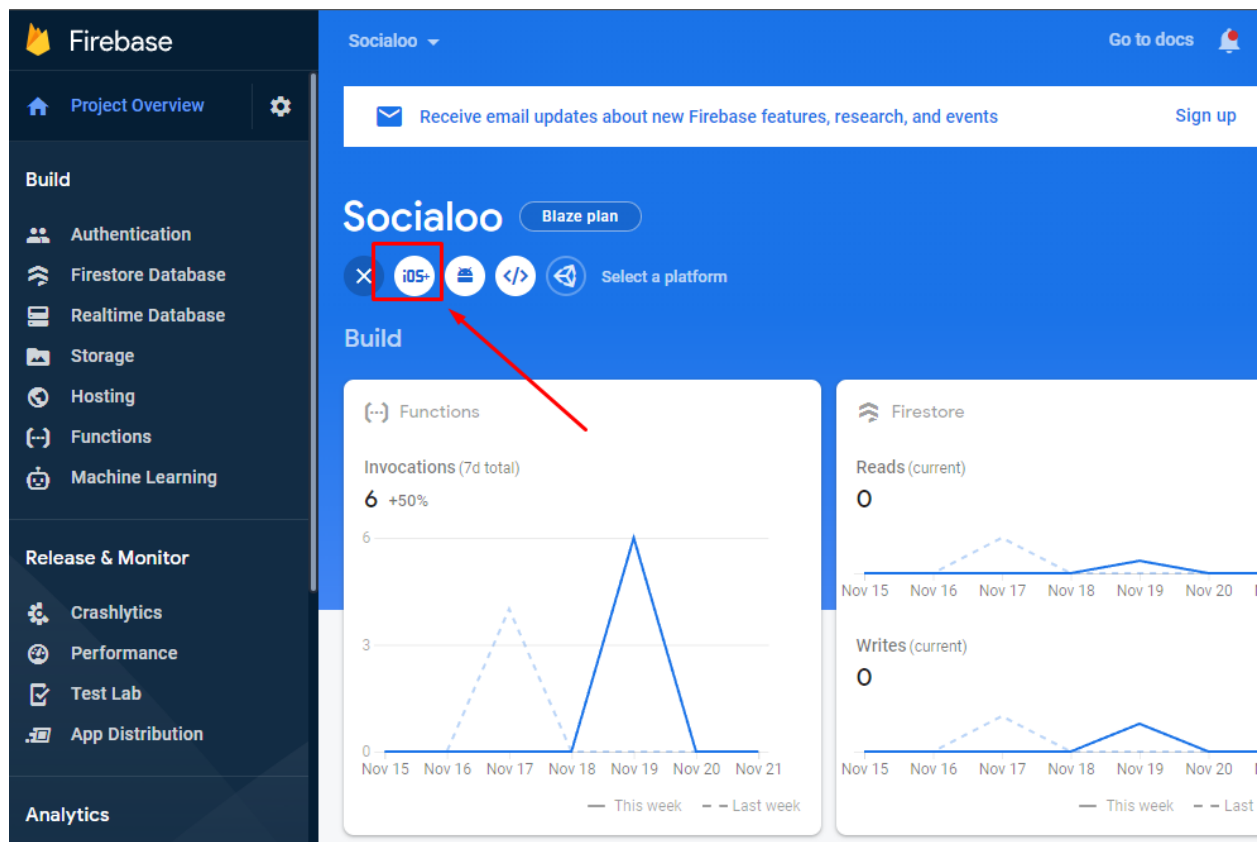


5. Add package name and SHA-1 key for your



Note: Before you run app, make sure to setup [Google Auth](#) and [Cloud Functions](#)

7. To add your IOS package to the project, click on add app button and select the IOS icon



Enter a unique bundle id

× Add Firebase to your Apple app

1 Register app

Apple bundle ID ?

com.company.appname

App nickname (optional) ?

My Apple App

App Store ID (optional) ?

123456789

Register app

2 Download config file

3 Add Firebase SDK

4 Add initialization code

5 Next steps

Place GoogleService-Info.plist file within \ios\Runner folder of your flutter project

× Add Firebase to your Apple app

✓ Register app
Apple bundle ID: com

2 Download config file

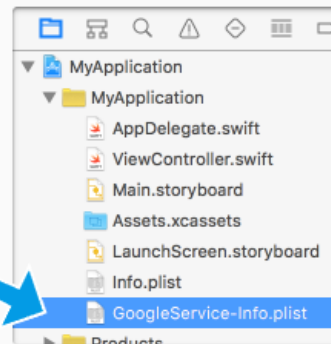
Instructions for Xcode below | [Unity](#) [C++](#)

↓ Download GoogleService-Info.plist

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.



GoogleService-Info.plist

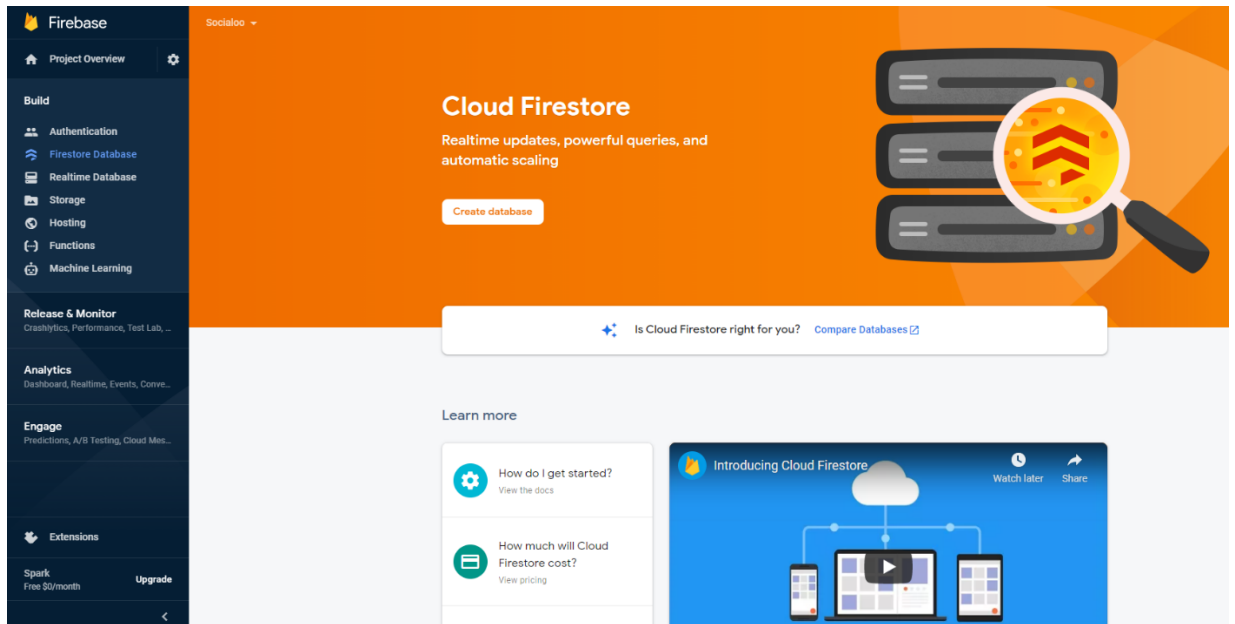


Next

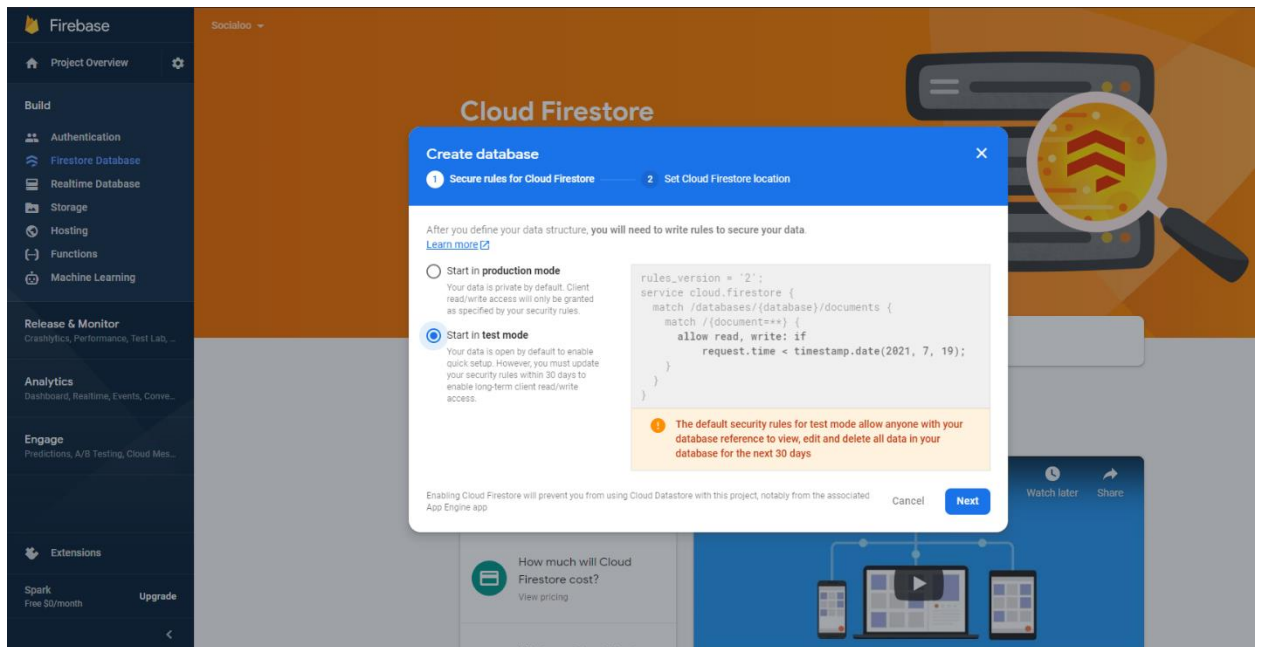
3 Add Firebase SDK

Setup Firestore:

1. Select Cloud Firestore tab and click on create database



2. Select test mode then hit Next



3. Select region that suite you.

Create database

✓ Secure rules for Cloud Firestore

2 Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.

⚠

After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

Learn more

Cloud Firestore location

us-west2

Note for Cloud Functions customers: Deploying a function to this location is not yet available. If your function and Cloud Firestore are in different locations, then you could potentially experience increased latency and billing costs. [Learn more](#)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

CancelDone

4. Add security rules like show below

Flutterapp

Cloud Firestore

DataRulesIndexesUsage

Edit rulesMonitor rules

Develop & Test

Yesterday • 3:30 PM

Jul 31, 2021 • 8:36 PM

Jul 31, 2021 • 8:32 PM

Jul 31, 2021 • 8:23 PM

Jun 24, 2021 • 4:37 PM

Jun 24, 2021 • 4:34 PM

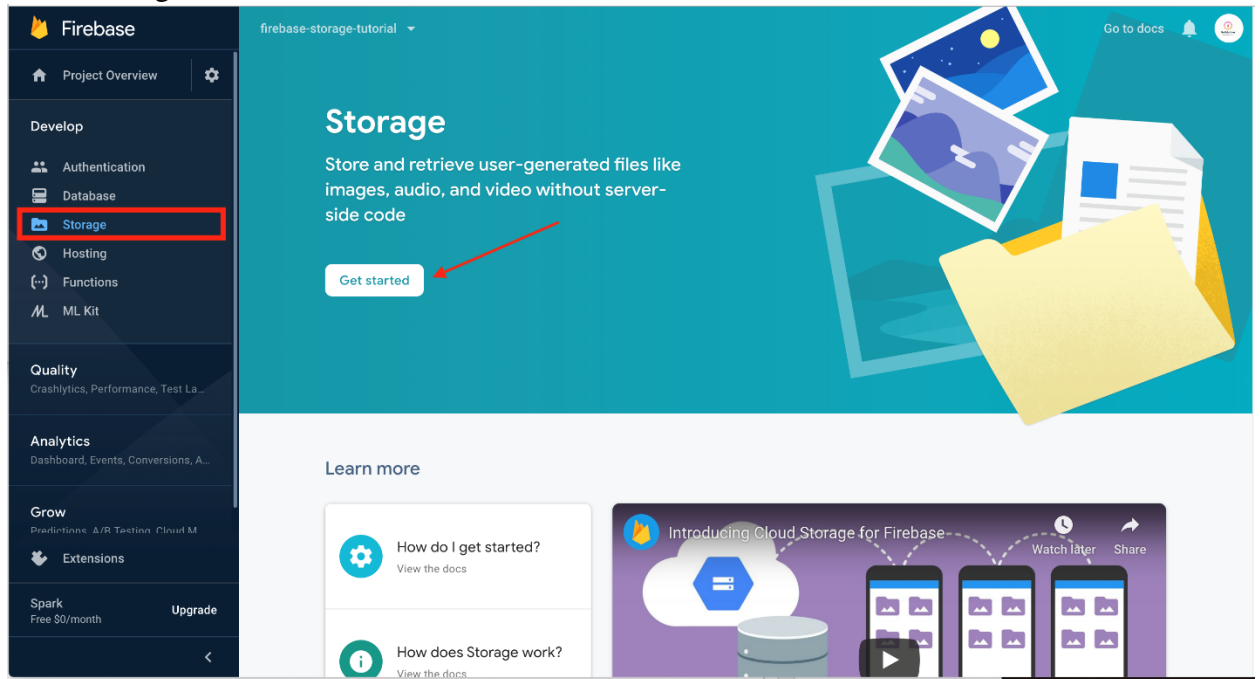
Jun 24, 2021 • 4:30 PM

Rules Playground

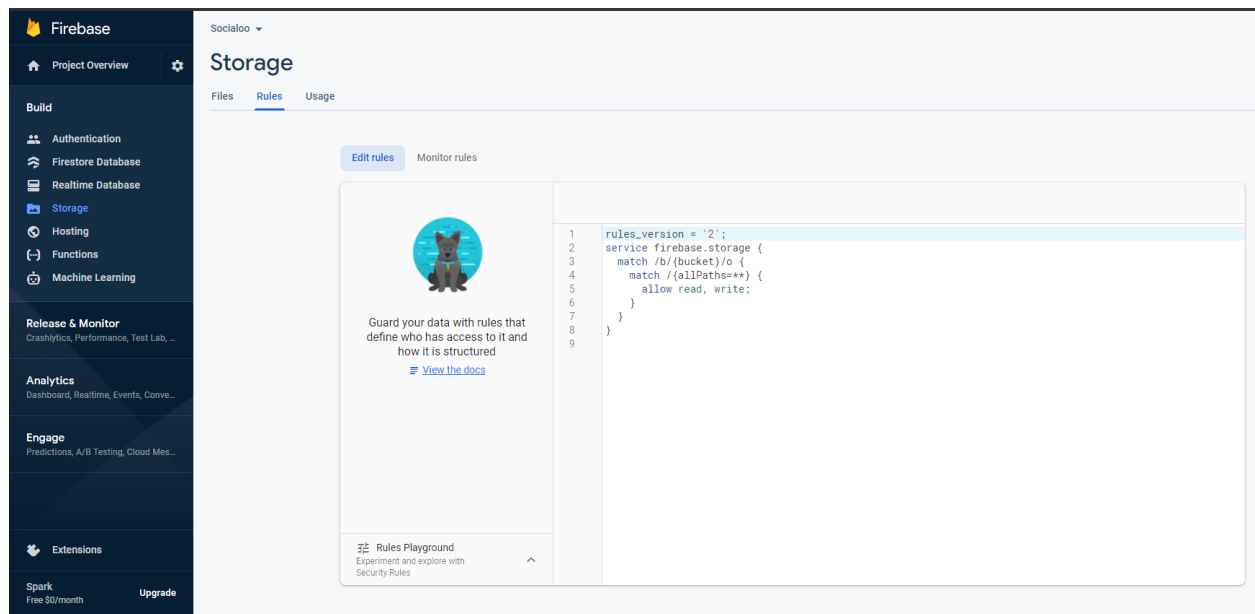
```
1 rules_version = '2';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4     match /{document=**} {
5       allow read, write: if true;
6     }
7   }
8 }
```

Setup Storage:

1. Select storage tab



2. Add security rules like show below



```
rules_version = '2';
```

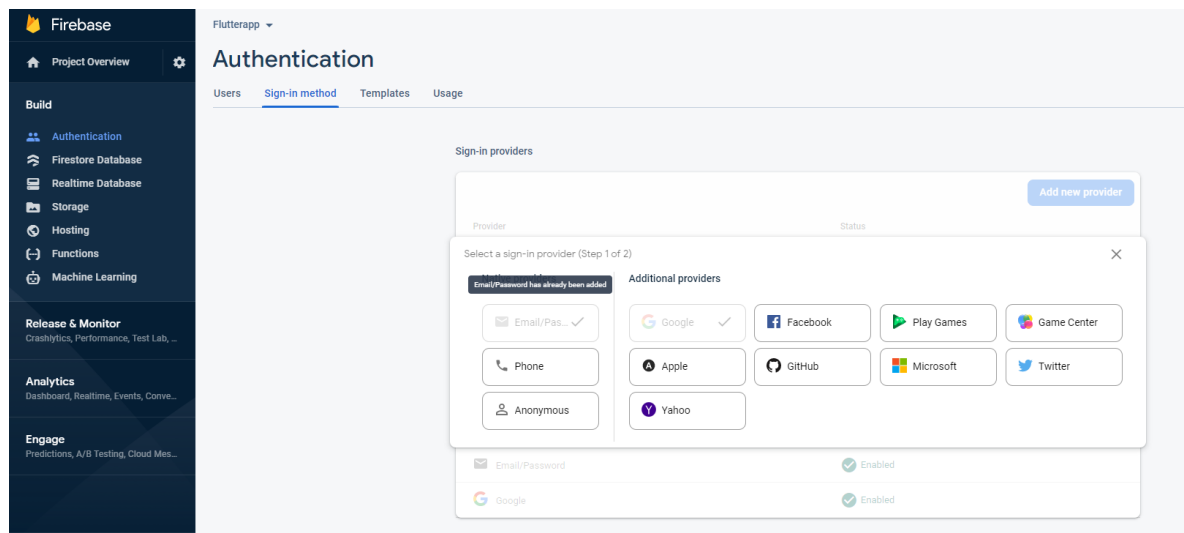
```
service firebase.storage {  
  match /b/{bucket}/o {  
    match /{allPaths=**} {
```

```
    allow read, write;
  }
}
}
```

Setup Email Authentication

Enable “Google Authentication” on the Firebase console.

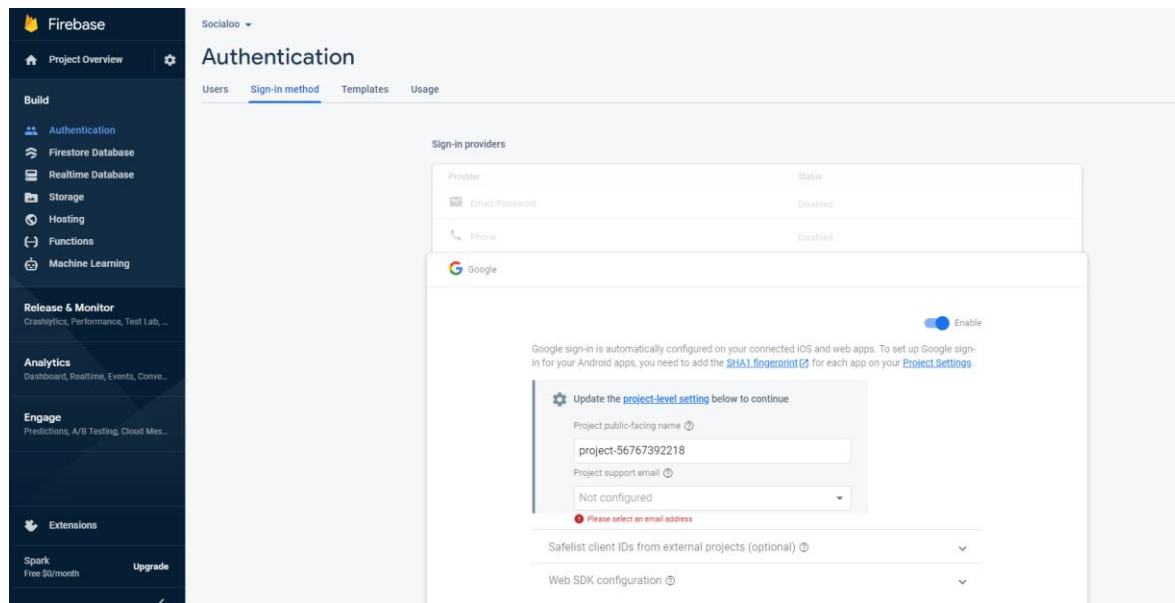
Firebase Console -> Develop (on the left menu) -> Authentication -> Sign-In Method -> Email Password is enabled.



Setup Google Authentication

Enable “Google Authentication” on the Firebase console.

Firebase Console -> Develop (on the left menu) -> Authentication -> Sign-In Method -> Google is enabled.



Add SHA certificate to Firebase console

This step is necessary for Android Google Authentication, without that authentication will not work.

SHA-1 key is required to use google login in your app from the below link you can find a detailed guide on how to create SHA-1 key

<https://developers.google.com/android/guides/client-auth>

To get the debug certificate fingerprint:

Command: `keytool -list -v -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore`

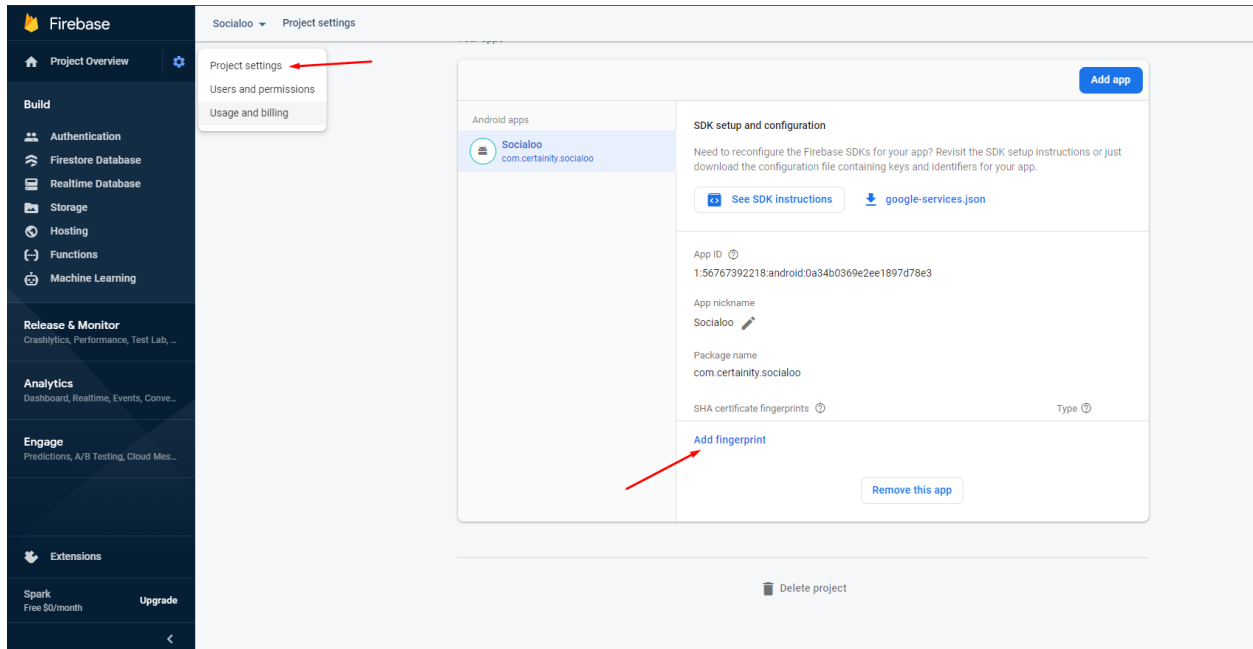
password: **android**

you have to copy the **SHA1**

The same thing is for release, you have to refer to your keystore location, the one below is for testing purposes.

Command: `keytool -list -v -alias key0 -keystore ./android/app/test_keystore`

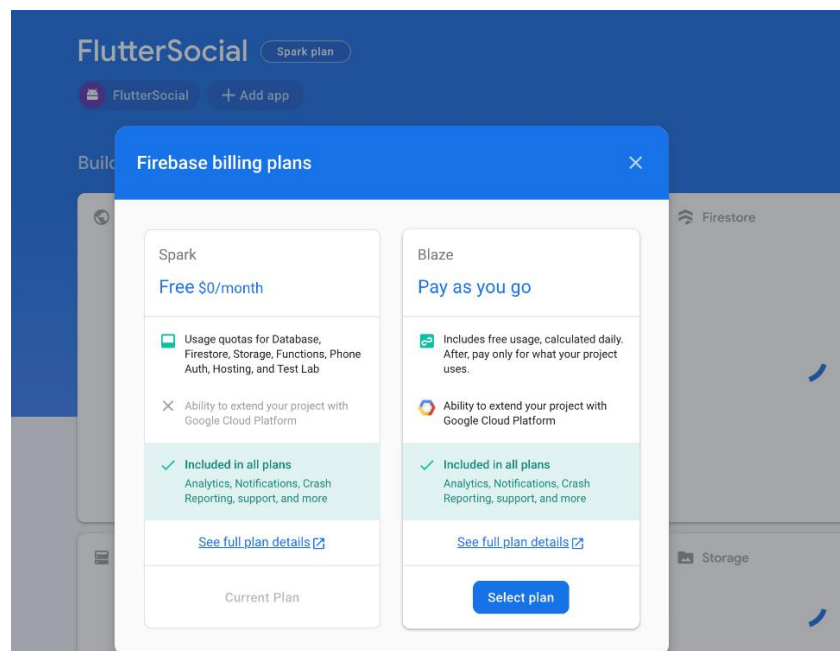
Copy the two SHA1 and SHA-256 keys to Firebase Console



Deploy Firebase config

In this section, we are going to config firebase which will allow Push notifications and generate all Firestore rules and indexes

Note: *You need to enable billing in your firebase account (Pay as you go Plan)*



Environment setup

First of all, you need have installed Node.js, at least the version 6.9.x or higher. You can download and install [from the Node.js project website](#). When you have done it, you can install the Firebase CLI in order to init a new project and setup functions feature. You can install it globally using npm:

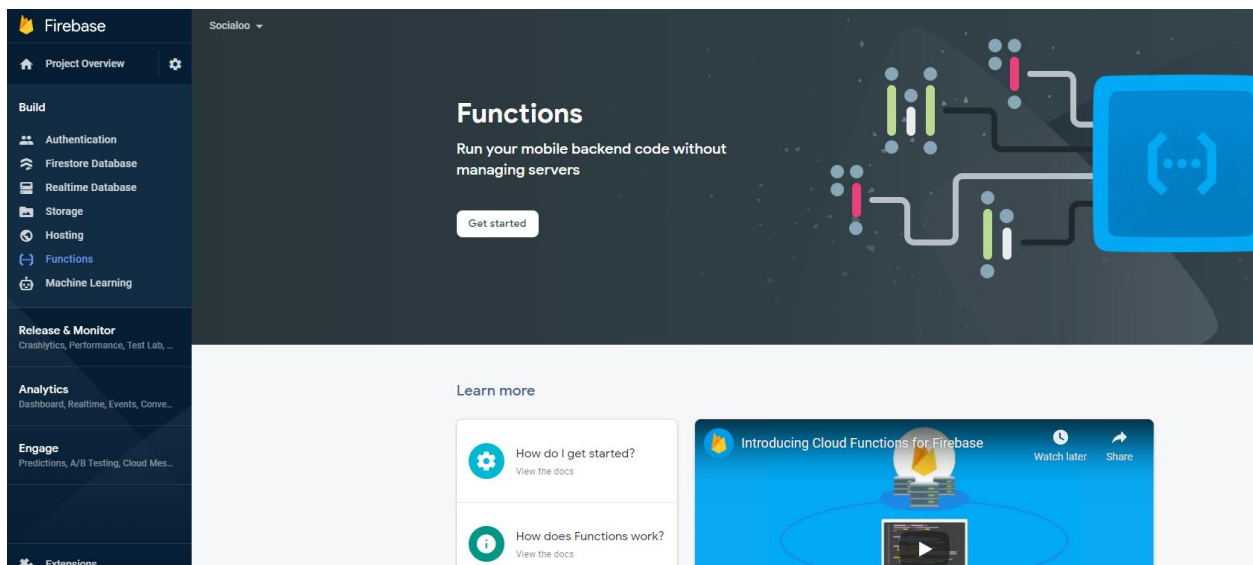
Command: `npm install -g firebase-tools`

You can verify if it is correctly installed whit the command:

Command: `firebase version`

`> 3.9.1`

Config your firebase project



1. Run `firebase login` to log in via the browser and authenticate the firebase tool.

Command: `firebase login`

2. **Command:** `firebase init functions`

- Are you ready to proceed? Yes
- What language would you like to use to write Cloud Functions? JavaScript
- Do you want to use ESLint to catch probable bugs and enforce style? No
- File functions/package.json already exists. Overwrite? No
- File functions/index.js already exists. Overwrite? No

- File functions/.gitignore already exists. Overwrite? No
- Do you want to install dependencies with npm now? Yes
-

3. Do firebase use --add with:

Command: firebase use --add

4. Then select your project and hit Enter

```
? Which project do you want to add?  
flutter-app-969bd  
flutterapp-f75ac  
simple-world-43bda  
> socialoo-97c50
```

5. make project alias as default

```
? What alias do you want to use for this project? (e.g. staging) default
```

6. Install dependencies inside functions folder

Command: cd functions

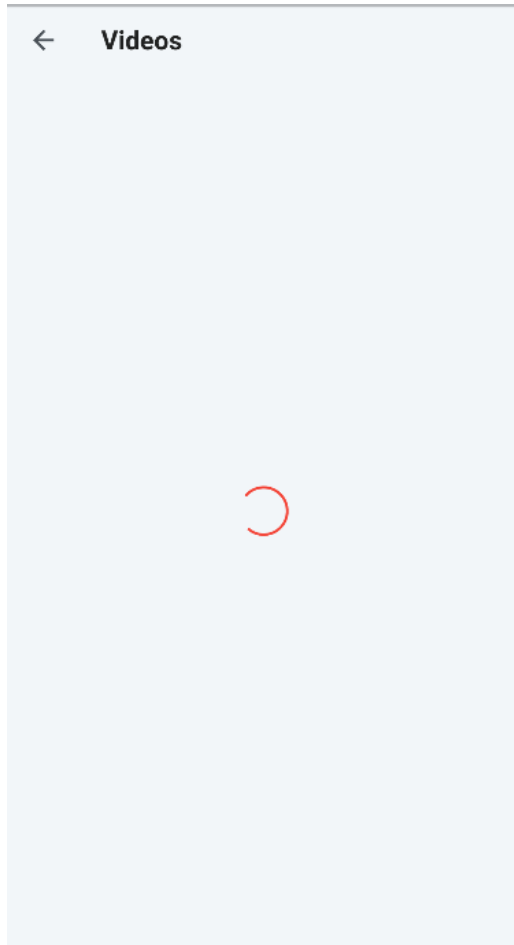
Command: npm install

7. Deploy

Command: firebase deploy --only functions

To Get Videos and pdf

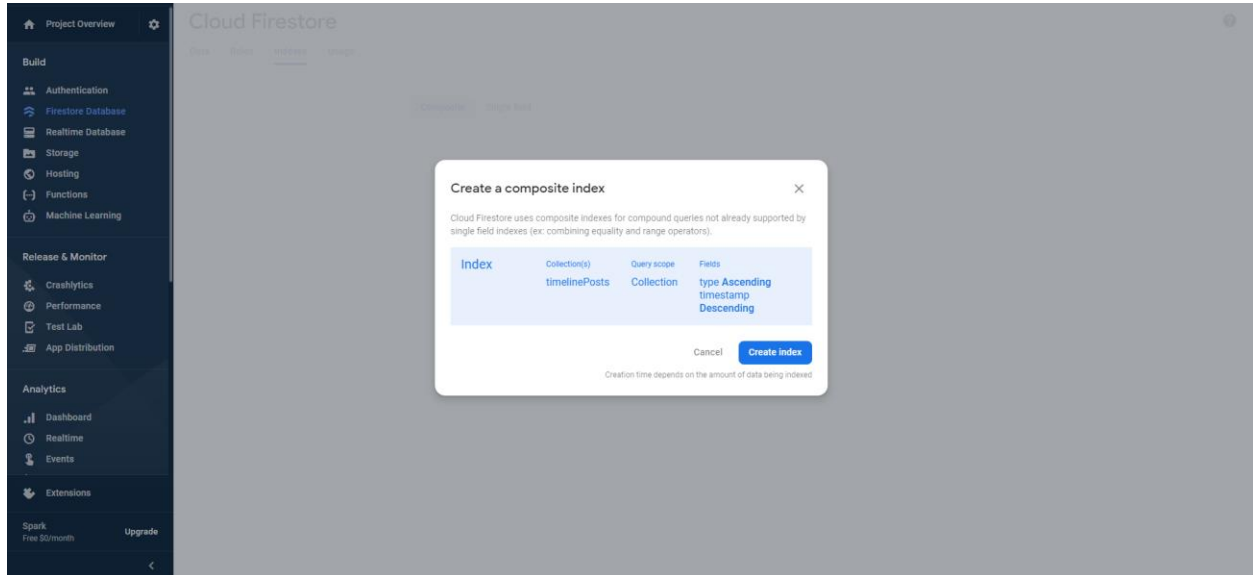
Run your app from VS code go to videos page or documents page in your application it will keep showing a loading sign as shown below



Now go back to VS code and check debug console, it will show a firebase link as shown below,

```
PROBLEMS (102) OUTPUT DEBUG CONSOLE TERMINAL COMMENTS Filter (e.g. text, exclude)
W/Firestore(23815): (24.0.0) [Firestore]: Listen for Query(target=Query(timeline/OskbUu3aI0cqjXUVF804xXMF2/timelinePosts where type == video order by -timestamp, -__name__);limitType=LIMIT_TO_FIRST) failed: Status[code=FAILED_PRECONDITION, description=The query requires an index. You can create it here: https://console.firebase.google.com/v1/r/project/simpleworldnew/firestore/indexes?create_composite=ClRucm9qZWNOcy92aW1ubGV3b3JsZG5ldy9KYXRhYmFzZXNvKGRlZmF1bGpl2Nvb6x1Y3Rpb25Hcm91cHMvdGltZmxpbWVQb3N0cy9pbmRleGVzL18QARoICgR0eXB1EAEaQo3dGltZXN0YV1wEAIAdAoIX19uYV11X18QAg, cause=null]
E/Flutter (23815): [ERROR:flutter/lib/ui/ui_dart_state.cc(289)] Unhandled Exception: [cloud_firestore/failed-precondition] The query requires an index. You can create it here: https://console.firebase.google.com/v1/r/project/simpleworldnew/firestore/indexes?create_composite=ClRucm9qZWNOcy92aW1ubGV3b3JsZG5ldy9KYXRhYmFzZXNvKGRlZmF1bGpl2Nvb6x1Y3Rpb25Hcm91cHMvdGltZmxpbWVQb3N0cy9pbmRleGVzL18QARoICgR0eXB1EAEaQo3dGltZXN0YV1wEAIAdAoIX19uYV11X18QAg
E/Flutter (23815):
```

Click on it to create index for your query

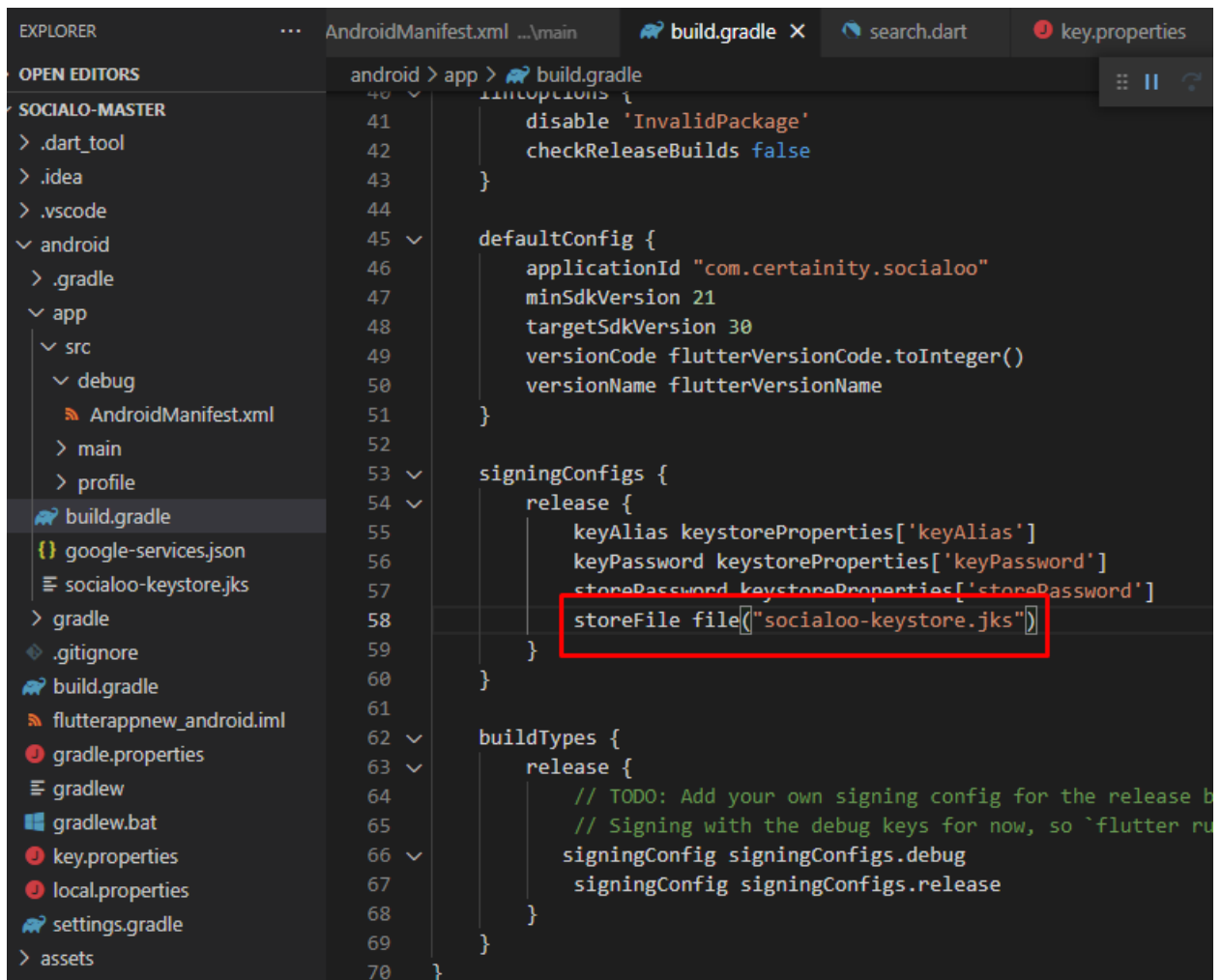


Build & Publish Flutter App

To build and publish Android App, follow the official Documentation [Android](#)

Or follow steps below

1. Place your keystore file in android\app folder
2. Open android > app> build.gradle and change the keystore name with your keystore



3. Open key.properties file and add your alias and password for your keystore
4. Then run the command below

Command: `flutter build appbundle`

1. (Running `flutter build` defaults to a release build.)

The release bundle for your app is created at `[project]/build/app/outputs/bundle/release/app.aab`.

By default, the app bundle contains your Dart code and the Flutter runtime compiled for [armeabi-v7a](#) (ARM 32-bit), [arm64-v8a](#) (ARM 64-bit), and [x86-64](#) (x86 64-bit).

For more information visit docs.flutter.com

To build and publish IOS App follow the link below
<https://docs.flutter.dev/deployment/ios>

```
open ios/Runner.xcworkspace
```

How to update your app

V1.10

- If you are using a previous version below v 1.9 we recommend you to build the app from start, follow above instruction to make firebase setups.
- If you are using version 1.9 you don't need to make any changes in firebase setup, just download the script and connect to your own firebase and customize you app by following project setups

V1.11

- To add your IOS package to the project, click om add app button and select the IOS icon
- Add a unique bundle id to create GoogleService-Info.plist file then place it within \ios\Runner folder of your flutter project

V1.12

- To update you code from V1.11 to V1.12 you have to make many changes in Firestore Database hence we recommend to clean your current database post collection and add new posts
- However if you have users and don't want to remove the current posts follow the steps below to make changes in posts you already have for your users
- Go to profile.dart file in your old script (code form version 1.11) add the given codes within the widget shown below and enter each of your users profile and load all their posts

```
profile.dart X
lib > pages > profile.dart > _ProfileState > buildPostHeader

679 |   ), // PaginateFirestore
680 |   ); // Scaffold
681 | }
682 |
683 | Widget buildPostHeader(post) {
684 |   /* TO CREATE HAPPY AND SAD FIELDS IN EACH POST FOR THE USER */
685 |
686 |   // postsRef
687 |   //   .doc(post['ownerId'])
688 |   //   .collection("userPosts")
689 |   //   .doc(post['postId'])
690 |   //   .set(
691 |   //     {'Happy': [], 'Sad': []},
692 |   //     SetOptions(
693 |   //       merge: true,
694 |   //     ),
695 |   //   ).then((value) {
696 |   //     print('New fields Added');
697 |   //   });
698 |
699 |   /* IF HAVE A IMAGE IN CURRENT POST CREATE A SUB COLLECTION AND TO MAKE NEW POST FOR THE IMAGE */
700 |   // bool hasMedia = post['mediaUrl']?.isNotEmpty == true;
701 |   // var photoId = postsRef
702 |   //   .doc(post['ownerId'])
703 |   //   .collection('userPosts')
704 |   //   .doc(post['postId'])
705 |   //   .collection("albumposts")
706 |   //   .doc()
707 |   //   .id
708 |   //   .toString();
709 |   // postsRef
710 |   //   .doc(post['ownerId'])
711 |   //   .collection("userPosts")
```

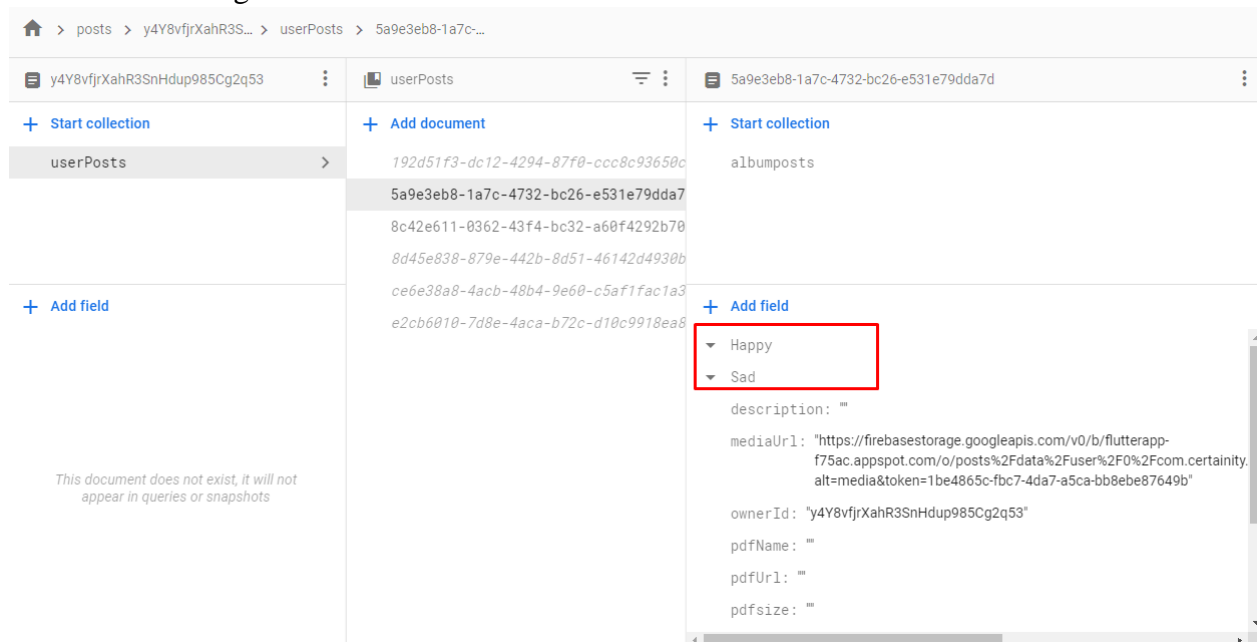
Step 1

Code 1

TO CREATE HAPPY AND SAD FIELDS IN EACH POST FOR THE USER

```
postsRef
  .doc(post['ownerId'])
  .collection("userPosts")
  .doc(post['postId'])
  .set(
    {'Happy': [], 'Sad': []},
    SetOptions(
      merge: true,
    ),
  ).then((value) {
    print('New fields Added');
  });
```

- To check if new fields are added go to your Firestore Database and check posts collection
- You should have got 2 new fields as shown below



Step 2

Code 2

IF HAVE A IMAGE IN CURRENT POST CREATE A SUB COLLECTION AND TO MAKE NEW POST FOR THE IMAGE

```
bool hasMedia = post['mediaUrl']?.isNotEmpty == true;

if (hasMedia) {
  var photoId = postsRef
    .doc(post['ownerId'])
    .collection('userPosts')
    .doc(post['postId'])
    .collection("albumposts")
    .doc()
    .id
    .toString();
  postsRef
    .doc(post['ownerId'])
    .collection("userPosts")
    .doc(post['postId'])
    .collection("albumposts")
    .doc(photoId)
    .set({
```

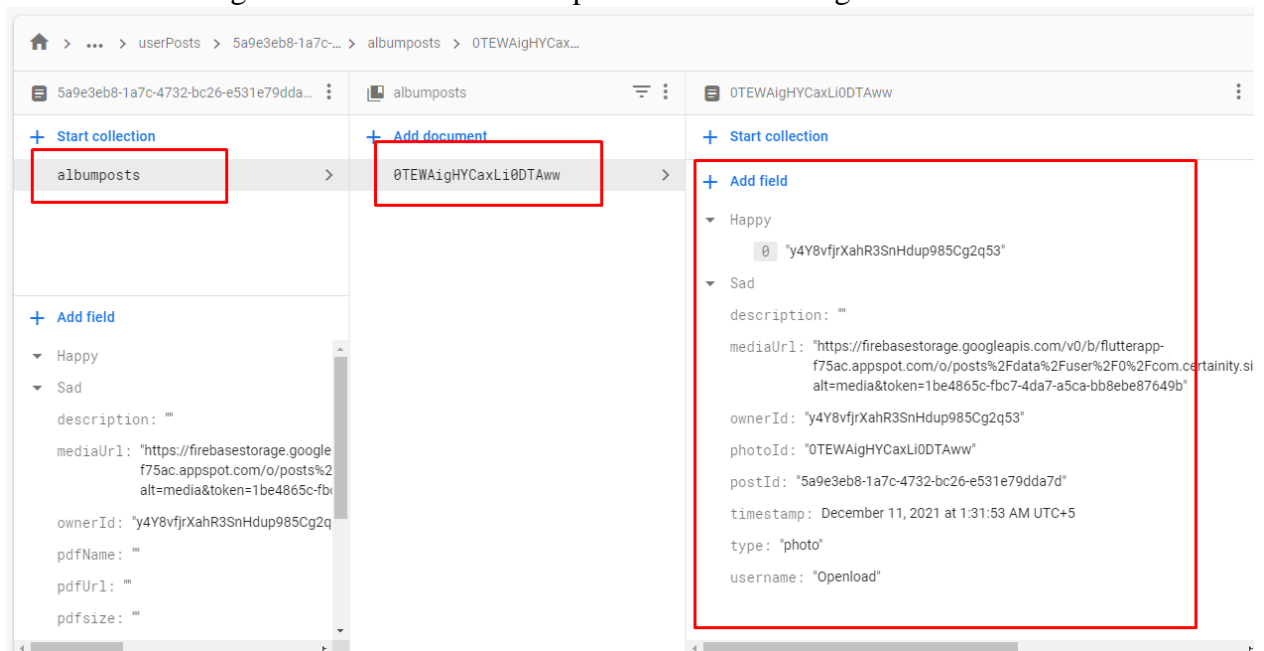


```

"mediaUrl": post['mediaUrl'],
"postId": post['postId'],
"ownerId": post['ownerId'],
"username": globalName,
"description": '',
"timestamp": timestamp,
"type": 'photo',
"Happy": [],
"Sad": [],
"photoId": photoId,
});
}

```

- While using this code be careful not to load each post 2 time, if it loads 2 times, 2 sub posts will be created in your subcollection
- To check if new collection is added go to your Firestore Database and check posts collection
- You should have got a subcollection for the posts contain an image as shown below



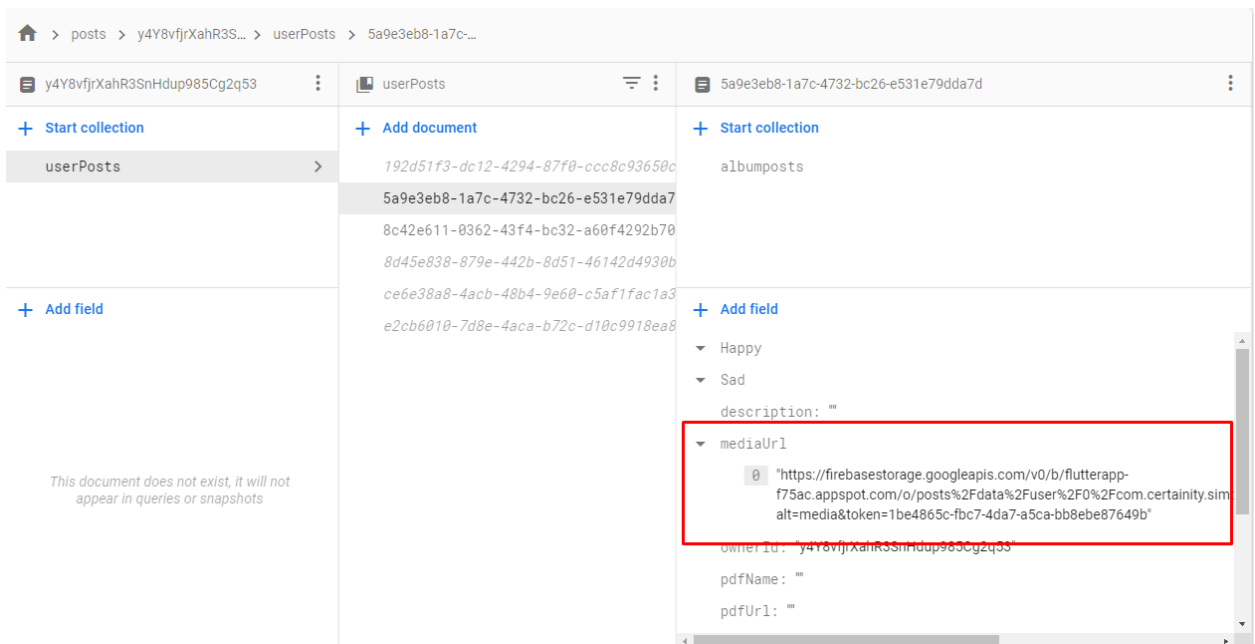
Step 3

Code 3

CONVERT MEDIALURL FIELD FROM STRING TO A LIST

```
bool hasMedia = post['mediaUrl']?.isEmpty == true;
postsRef
  .doc(post['ownerId'])
  .collection("userPosts")
  .doc(post['postId'])
  .update(
    hasMedia
      ? {
        'mediaUrl': [post['mediaUrl']],
      }
      : {
        'mediaUrl': [],
      },
  )
  .then((value) {
    print('New fields Added');
  });
```

- To check if mediaUrl string is changed to an array check a post in your posts collection it should have changed as shown below,



If the post is other than an image an empty mediaUrl array should have been created

+ Add field

▼ Happy

▼ Sad

description: "hi"

▼ mediaUrl

- The above steps will help you to change your current posts to the new version of code
- After completing each step remove the code used to change the posts for that step

V1.13

- To update you code from V1.12 to V1.13 we don't need much changes in firebase, however if you have stories delete the stories collection so that new format of stories would appear
- Easiest way to update it is download code from codecanyon and connect it to your firebase
- However, if you want to make manual changes, we have added changed files below
 - New Files
 - All files in lib/widgets/count folder
 - All files in lib/story folder
 - Changed files
 - new_timeline.dart
 - profile.dart
 - all_pdfs.dart
 - all_videos.dart
 - login_page.dart
 - reaction_data.dart
 - upload.dart
 - video_upload.dart
 - pdf_upload.dart
 - add_post.dart
 - post_box.dart
 - login_page.dart
 - home.dart
 - app.dart
 - palette.dart

V1.14

- To update you code from V1.13 to V1.14 many changes were made in firebase.
- Easiest way to update it is download code from codecanyon and connect it to your firebase
- However, if you want to make manual changes, we have added changed files below

- New Files
 - post_layout.dart
 - simpleworld_chat_main.dart
 - suggested_users_tile.dart
 - messages_count.dart
- Changed files
 - new_timeline.dart
 - profile.dart
 - all_pdfs.dart
 - all_videos.dart
 - upload.dart
 - video_upload.dart
 - pdf_upload.dart
 - add_post.dart
 - post_box.dart
 - home.dart
 - palette.dart
 - all_stories.dart
 - suggested_users.dart
 - all_simpleworld_stories.dart
 - simpleworld_stories.dart

Note: As we have brought major changes in post time and how its being displayed, the posts in your post collection wont'work, you'll need to change t

For messages count to work in your code you'll need to add a new field in all existing documents under messenger collection as badge: '0' as shown below

[+ Add document](#)

0UvdSl8DbCXe08wofbQfKN5ADWD3 >

nxsm9nXzr3QbS7XwxKEJYZNKFzc2

[+ Start collection](#)

[+ Add field](#)

badge: "0"

content: "Hi"

id: "0UvdSl8DbCXe08wofbQfKN5ADWD3"

name: "Demo"

profileImage: "https://firebasestorage.googleapis.com/v0/b/flutterapp-f75ac.appspot.com/o/avatar_0UvdSl8DbCXe08wofbQfKN5ADWD3alt=media&token=423887fa-6441-4750-ac06-39d125de252c"

timestamp: "1640120500727"

type: 0

V1.14.1

- To update you code from V1.14 to V1.14.1 many no changes were made in firebase.
- Easiest way to update it is download code from codecanyon and connect it to your firebase
- However, if you want to make manual changes, we have added changed files below
 - Changed files
 - new_timeline.dart
 - activity_feed.dart
 - story folder