News Hour Admin Doc (v3.0.4)

05-03-21 Last updated

“It's an offline document. There is an online documentation here. We are highly recommending the online version. Because it will always be updated.”

Introduction

Welcome to News hour documentation & thank you for buying our product. So there are two setup steps here. Admin panel & App. Both are developed on Flutter. So, You have to install flutter on your computer. You can use Both Visual Studio Code & Android Studio for Flutter. We have used Visual Studio Code. So, Our Setup will be based on this IDE.

To install flutter on your computer, follow the official documentation from Google.

Flutter Official Site : https://flutter.dev/.

Admin Panel Setup

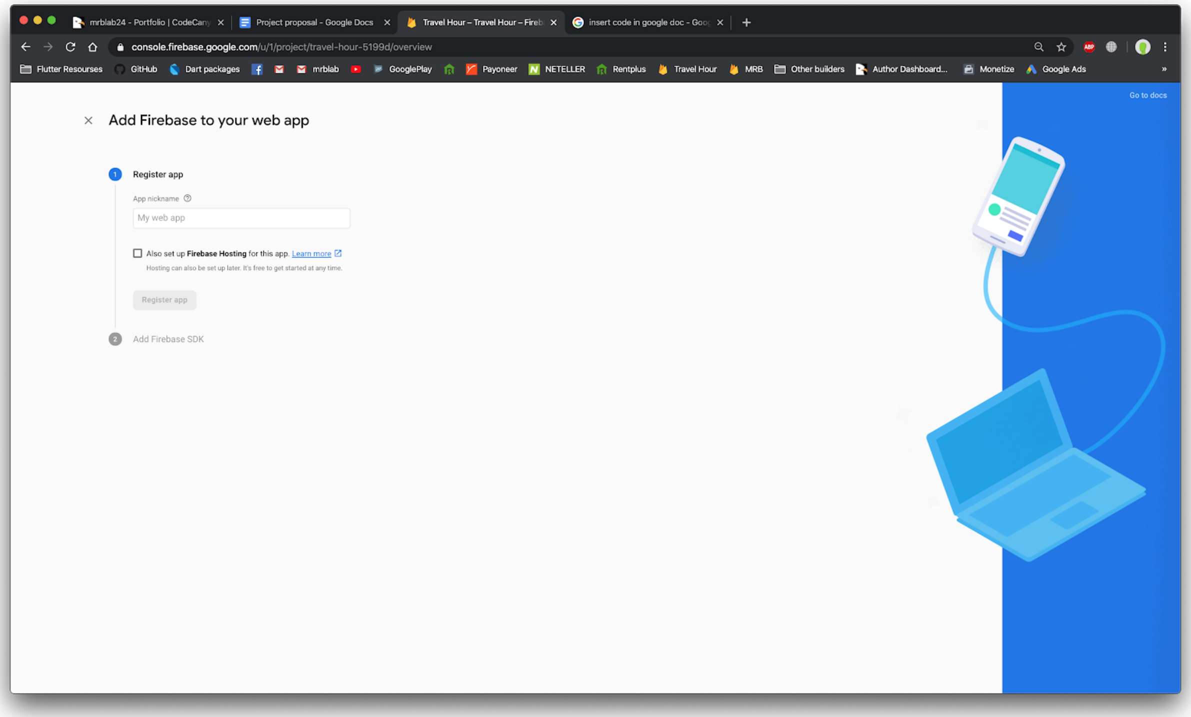
After successfully installing Flutter on your computer, You have to enable flutter web support. Make sure you have installed the stable channel. From Flutter stable version 2.0.0, it will support the web. So, If you are already on the latest stable channel of flutter, you don’t have to do anything for the flutter web. Just run the following code to enable flutter web support.

```
flutter config --enable-web
```

And then it will enable the web feature for your project.

Firebase(Database) Setup for Admin Panel :

1. First, you need to set up the database. We have used the firestore **database** as backend. So, go to the firebase console and create a project by your app name. And go to the project overview and click on the **plus icon** and then click on the **web**. You will see a popup like this :



Add Firebase to your web app

1 Register app

App nickname

My web app

□ Also set up **Firebase Hosting** for this app. Learn more

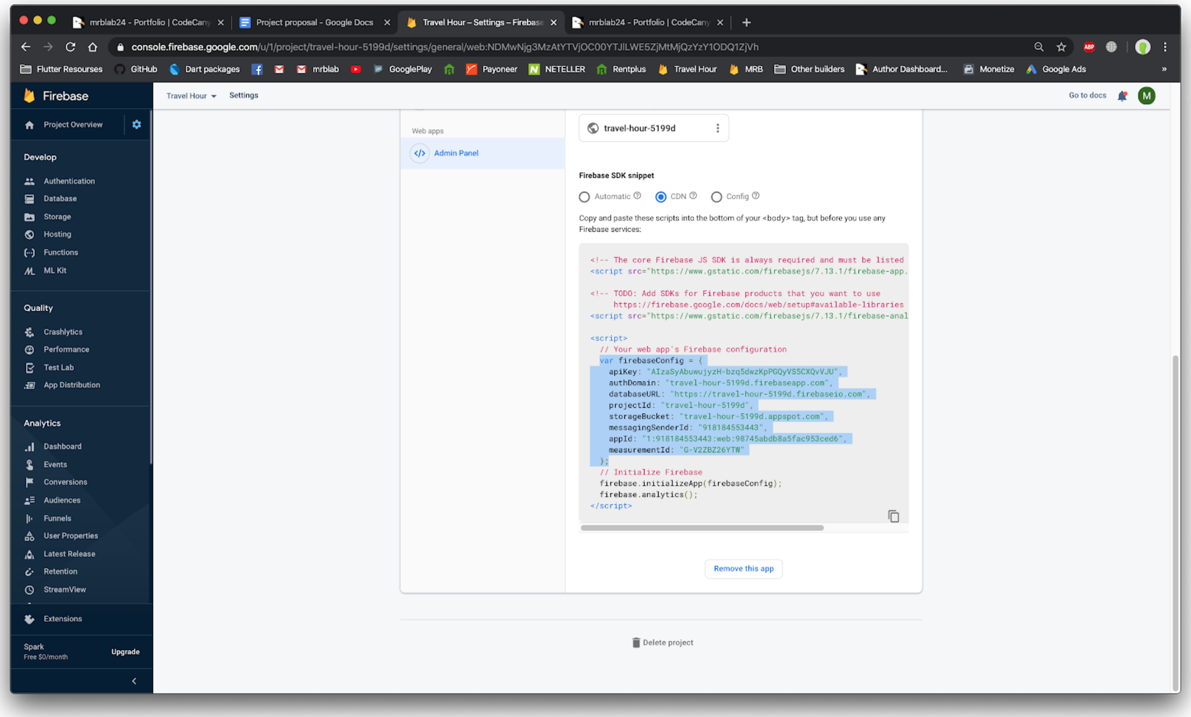
Hosting can also be set up later. It's free to get started at any time.

mrblab24 - Portfolio | CodeCany
Project proposal - Google Docs
Travel Hour – Travel Hour – Fireb
insert code in google doc - Goo
console.firebase.google.com/u/1/project/travel-hour-5199d/overview
Flutter Resourses
GitHub
Dart packages
mrblab
GooglePlay
Payoneer
NETELLER
Rentplus
Travel Hour
MRB
Other builders
Author Dashboard...
Monetize
Google Ads
Go to docs
Register app
2 Add Firebase SDK

2. Input the name like as Admin Panel and fill the checkbox because you need hosting too and then click on **register app**. After that, Skip the other options by clicking next. (Don't input custom name for hosting else you need extra setup for this)

3. Now go to the Project settings and in the bottom section click on the **Admin panel** that you have just created.

4. After that click on **CDN** and copy the selected code which is shown in the picture below.



Firebase SDK snippet

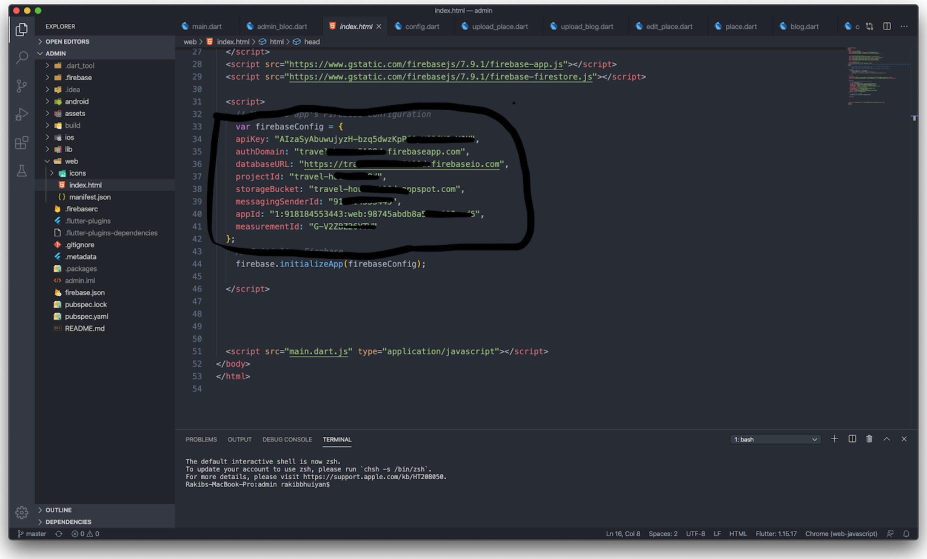
○ Automatic ● CDN ○ Config

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed
<script src="https://www.gstatic.com/firebasejs/7.13.1/firebase-app.

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries
<script src="https://www.gstatic.com/firebasejs/7.13.1/firebase-anal

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyAbuwujyzH-bzq5dwzKpPGQyVS5CXQvVJU",
    authDomain: "travel-hour-5199d.firebaseapp.com",
    databaseURL: "https://travel-hour-5199d.firebaseio.com",
    projectId: "travel-hour-5199d",
    storageBucket: "travel-hour-5199d.appspot.com",
    messagingSenderId: "918184553443",
    appId: "1:918184553443:web:98745abdb8a5fac953ced6",
    measurementId: "G-V2ZBZ26YTW"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

5. Now, go to the IDE VSCode or Android Studio and open the **admin** folder.
6. Go to this file and replace the selected code with your config code.

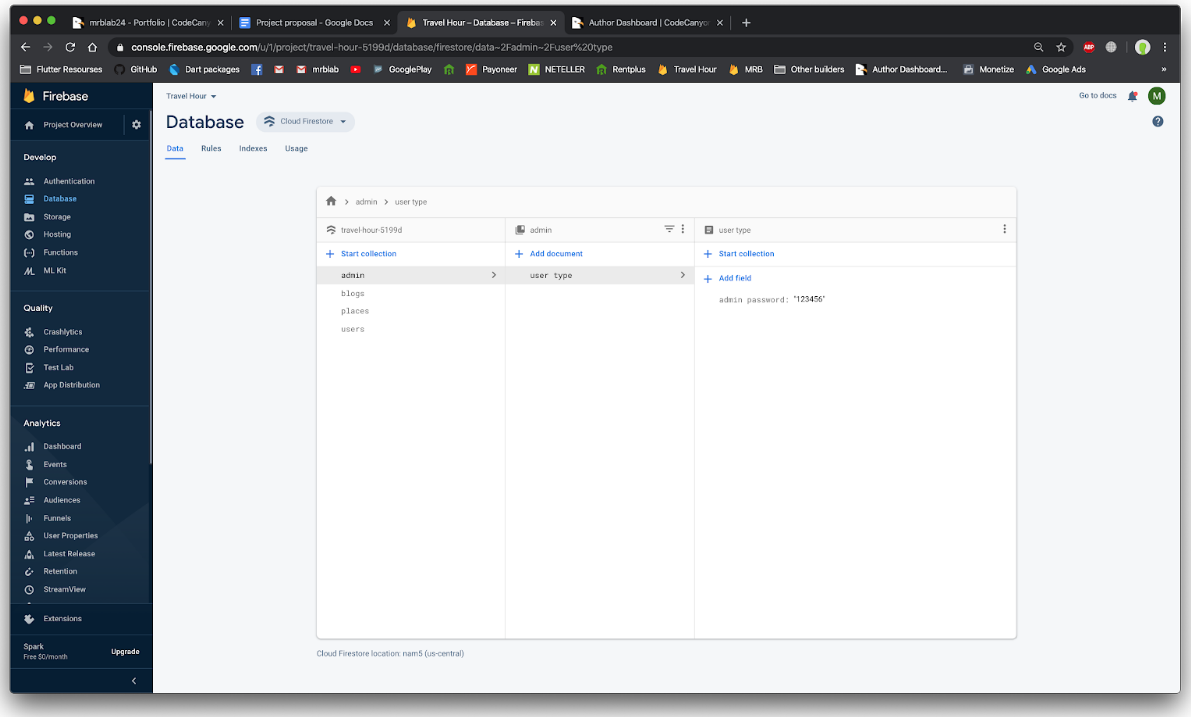
admin/web/index.html



```
  </script>
  <script src="https://www.gstatic.com/firebasejs/7.9.1/firebase-app.js"></script>
  <script src="https://www.gstatic.com/firebasejs/7.9.1/firebase-firestore.js"></script>

  <script>
    // Your web app's Firebase configuration
    var firebaseConfig = {
    apiKey: "AIzaSyAbuwujyzH-bzq5dwzKpP[redacted]",
    authDomain: "travel[redacted].firebaseapp.com",
    databaseURL: "https://tra[redacted].firebaseio.com",
    projectId: "travel-h[redacted]",
    storageBucket: "travel-hou[redacted]ppspot.com",
    messagingSenderId: "91[redacted]",
    appId: "1:918184553443:web:98745abdb8a5[redacted]5",
    measurementId: "G-V2Z[redacted]"
  };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);

  </script>




  <script src="main.dart.js" type="application/javascript"></script>
</body>
</html>
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Rakibs-MacBook-Pro:admin rakibbhuiyan$
```

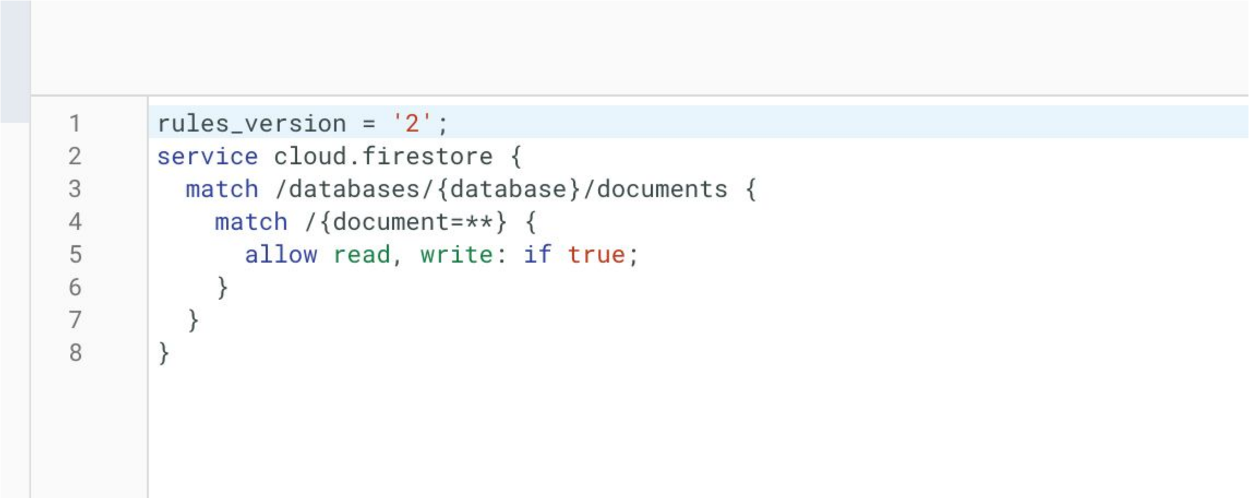
7. Now go to the Project Overview page and click on database and create a **firestore database** as test mode.
8. Now, Create a collection by **admin** and create a document by **user type** and create a textfield by **admin password**. Just like the picture below. Only the admin section. And then Change the password. Don't use the testing password here.



admin password: "123456"

Database Rules Setup :

Go to your firebase console > project overview > database > cloud firestore > rules and then edit the rules like this and click publish. That's it.



```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

Database Index Setup :

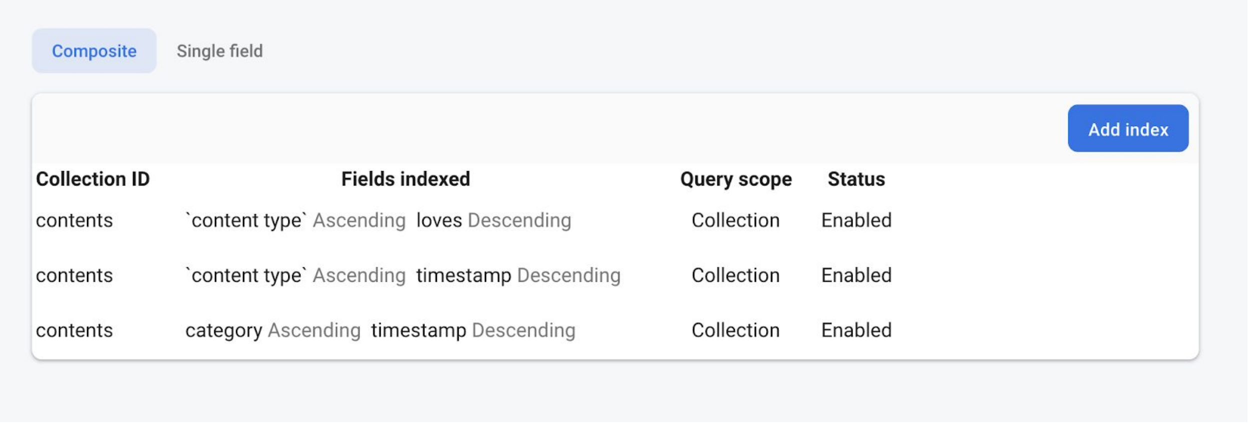
Now go to the right tab (Indexes) of the same page. You have to create 3 indexes one by one. Click on **Add Index** and then create an index by following the list.

1. Enter **contents** as Collection ID, **`content type`** into the first field and set **Ascending,** Enter **loves** in the second field and set **Descending** and then select **Collection** as query scope**.**

2. Enter **contents** as Collection ID, **`content type`** into the first field and set **Ascending,** Enter **timestamp** in the second field and set **Descending** and then select **Collection** as query scope**.**

3. Enter **contents** as Collection ID, **category** into the first field and set **Ascending,** Enter **timestamp** in the second field and set **Descending** and then select **Collection** as query scope**.**

After completing this, your index page should look like this :

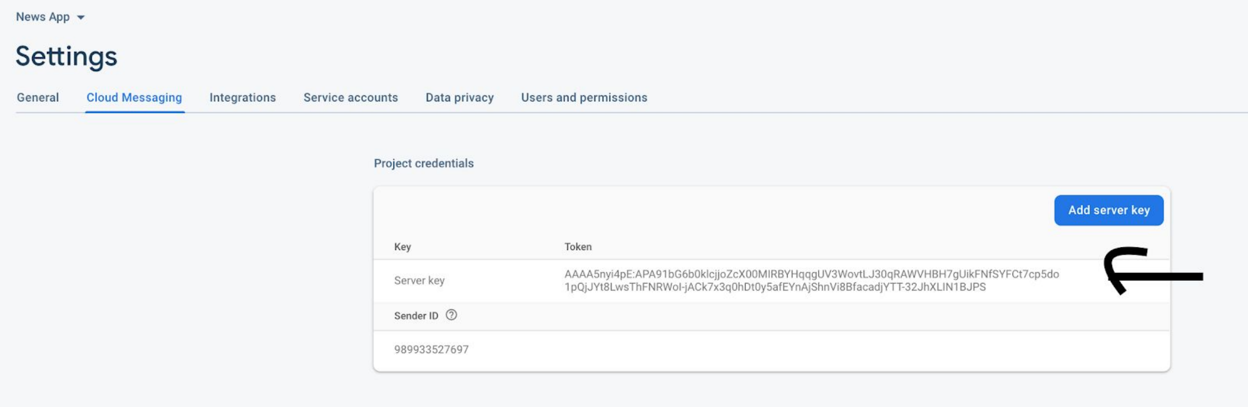


Collection ID	Fields indexed	Query scope	Status
contents	\`content type\` Ascending loves Descending	Collection	Enabled
contents	\`content type\` Ascending timestamp Descending	Collection	Enabled
contents	category Ascending timestamp Descending	Collection	Enabled

That's it. Your database setup is complete.

Push Notification & Other Setup :

1. Now, You have to set up push notifications to notify your users. To do that, go to project settings > cloud messaging tab and copy the **Server Key**.



Project credentials

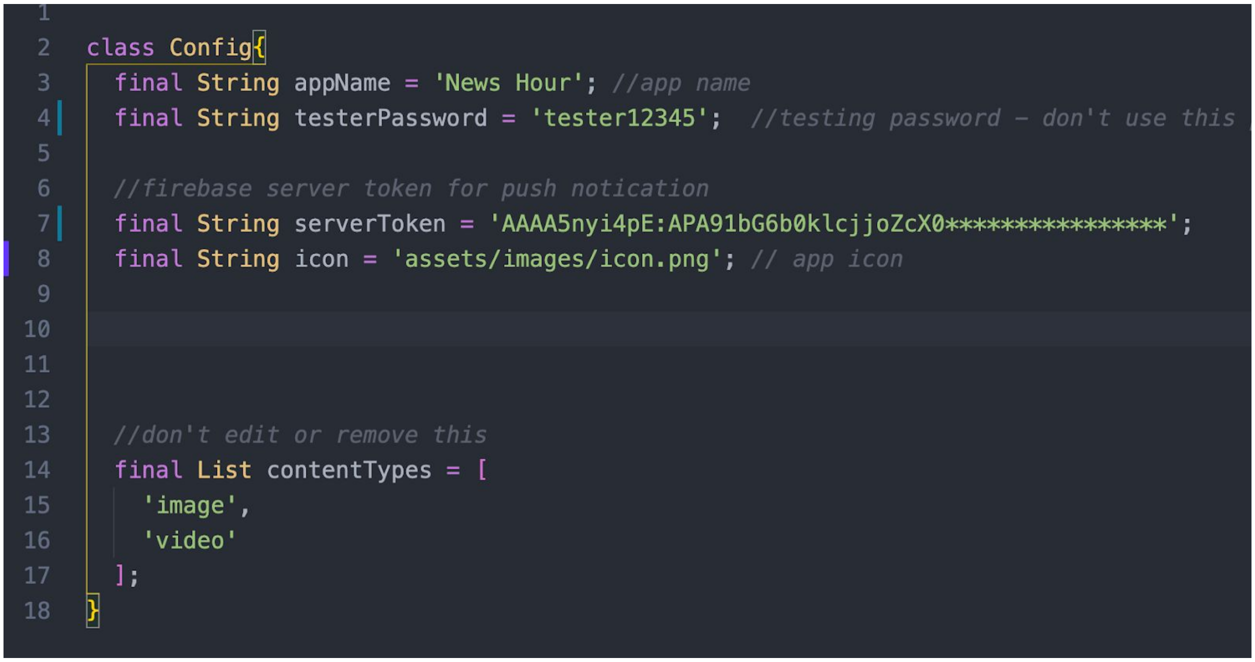
Key	Token
Server key	AAAA5nyi4pE:APA91bG6b0klcjjoZcX00MIRBYHqqgUV3WovtLJ30qRAWVHBH7gUikFNfSYFCt7cp5do1pQjJYt8LwsThFNRWol-jACk7x3q0hDt0y5afEYnAjShnVi8BfacadjYTT-32JhXLIN1BJPS

Sender ID

989933527697

2. Now go to this file and replace the **app name**, **server token** that you have just copied.

admin/lib/config/config.dart



```dart

class Config{
  final String appName = 'News Hour'; //app name
  final String testerPassword = 'tester12345';  //testing password - don't use this

  //firebase server token for push notication
  final String serverToken = 'AAAA5nyi4pE:APA91bG6b0klcjjoZcX0****************';
  final String icon = 'assets/images/icon.png'; // app icon




  //don't edit or remove this
  final List contentTypes = [
    'image',
    'video'
  ];
}
```

3. No you have to change the app icon. First rename your icon to **icon** and make sure the icon is in **.png** format. Now drag & drop your icon to the **assets/images** folder and replace the older icon.

4. That’s it. Your Admin Setup is complete. You can run the admin app now.

Run The Admin App :

1. You can run the admin app now running the following command in the terminal.

```
flutter run -d chrome
```

2. After running the admin app, Use the password that you have saved in the database before and check if everything is okay or not.

Upload the Admin App to the Firebase Hosting :

1. Now you need to upload your site to your firebase hosting. For this, you need to install firebase tools on your IDE. Run this command on the terminal:

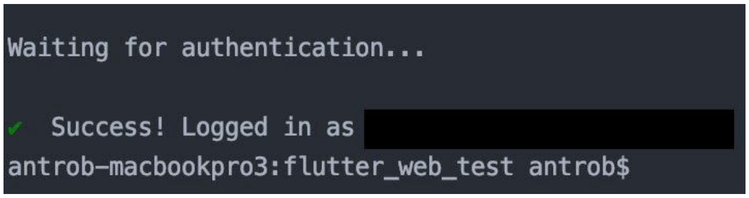
   It will download firebase tools for you. Or you directly download from [here](#) and install this manually.

```
npm install -g firebase-tools
```

2. After that, You need to login to firebase console with your google account. To do that, Run:

```
firebase login
```

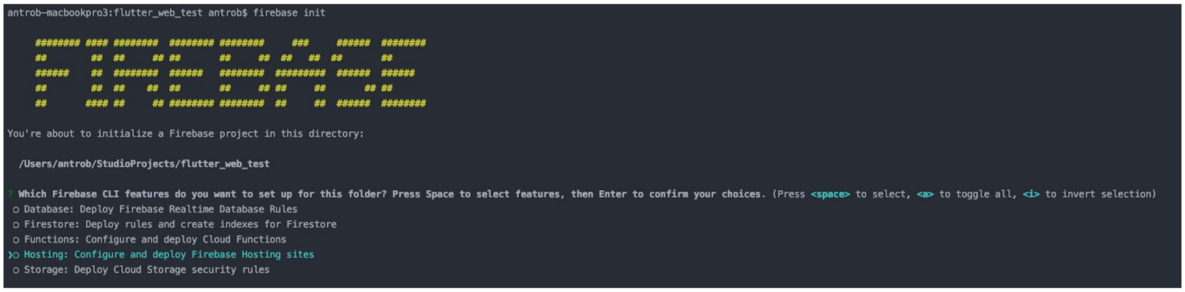
   Follow the command prompt and the link in your browser. Accept the permissions by clicking Ok. When you return to your terminal you should see that you are now logged in:



```
Waiting for authentication...

✔  Success! Logged in as [redacted]
antrob-macbookpro3:flutter_web_test antrob$
```

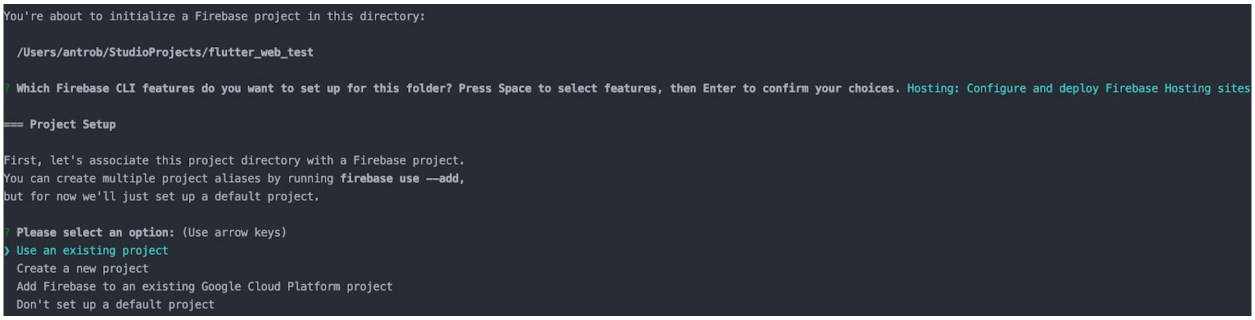
3. Now you need to initialize firebase. Run this command:

```
firebase init
```



```
antrob-macbookpro3:flutter_web_test antrob$ firebase init

     ######## #### ########  ######## ########     ###     ######  ########
     ##        ##  ##     ## ##       ##     ##  ##   ##  ##       ##
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########

You're about to initialize a Firebase project in this directory:

  /Users/antrob/StudioProjects/flutter_web_test

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection)
 ○ Database: Deploy Firebase Realtime Database Rules
 ○ Firestore: Deploy rules and create indexes for Firestore
 ○ Functions: Configure and deploy Cloud Functions
❯○ Hosting: Configure and deploy Firebase Hosting sites
 ○ Storage: Deploy Cloud Storage security rules
```

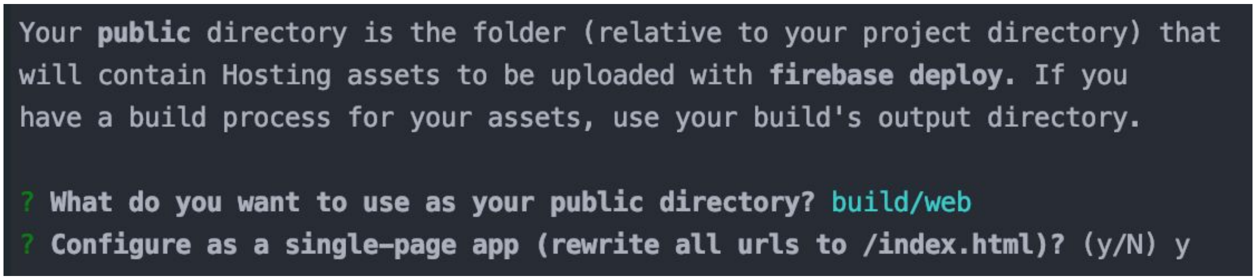
4. Use the arrow keys to navigate the cursor to Hosting and hit the spacebar to select it, then press enter. Now you will see this screen:



```
You're about to initialize a Firebase project in this directory:

  /Users/antrob/StudioProjects/flutter_web_test

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. Hosting: Configure and deploy Firebase Hosting sites

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)
❯ Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

Select Use an existing project by pressing Enter. Use the arrow keys to select the project you have made in the firebase console.

5. Next, enter **build/web** as the public directory and press enter, then enter y (for yes) to select the single page app option:



```
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build/web
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) y
```

6. Now Run this command to build the web app.

```
Flutter build web
```

7. After that run this command to upload the app to your hosting.

```
firebase deploy
```

That’s it. Your Admin Panel is 100% complete. You will get a url from the terminal and it will be the domain of your admin panel and you can use this domain for further use of the admin panel. You can also get will domain from your **firebase console > hosting page.**

Now, Go for the app setup!

Before the App Setup (Impotant!!!!!!!!!):

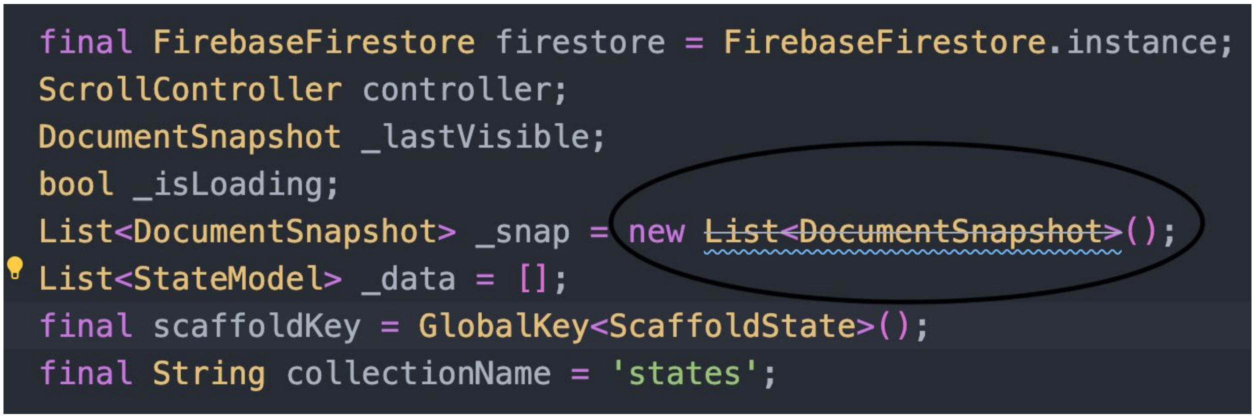
You have to upload some categories & articles. Otherwise the app will show nothing at all and can cause errors. In order to upload articles, you have to upload at least **4 categories which is mandatory!! Because these 4 categories will be static in the app for UI purposes.**

Make sure you have uploaded at least 4 categories, 1 video article & 1 image article.

And also add at least 1 item to the featured list.

Bugs & Fixes :

1. **List deprecation error :** You could see an error like this because of the flutter recent update.



```
final FirebaseFirestore firestore = FirebaseFirestore.instance;
ScrollController controller;
DocumentSnapshot _lastVisible;
bool _isLoading;
List<DocumentSnapshot> _snap = new ~~List<DocumentSnapshot>~~();
List<StateModel> _data = [];
final scaffoldKey = GlobalKey<ScaffoldState>();
final String collectionName = 'states';
```

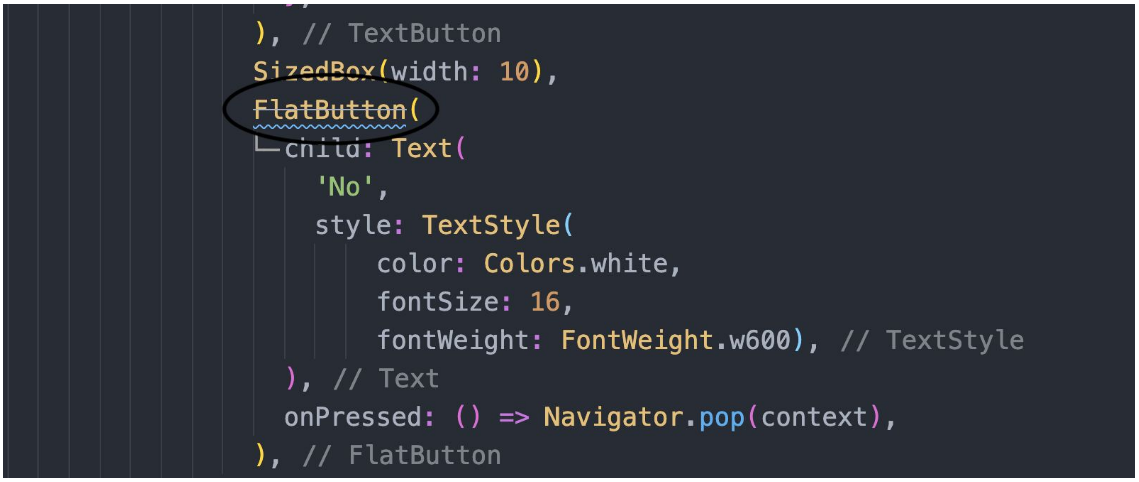
To fix this, just remove this :

`new List<DocumentSnapshot>()`

and add this :

`[]`

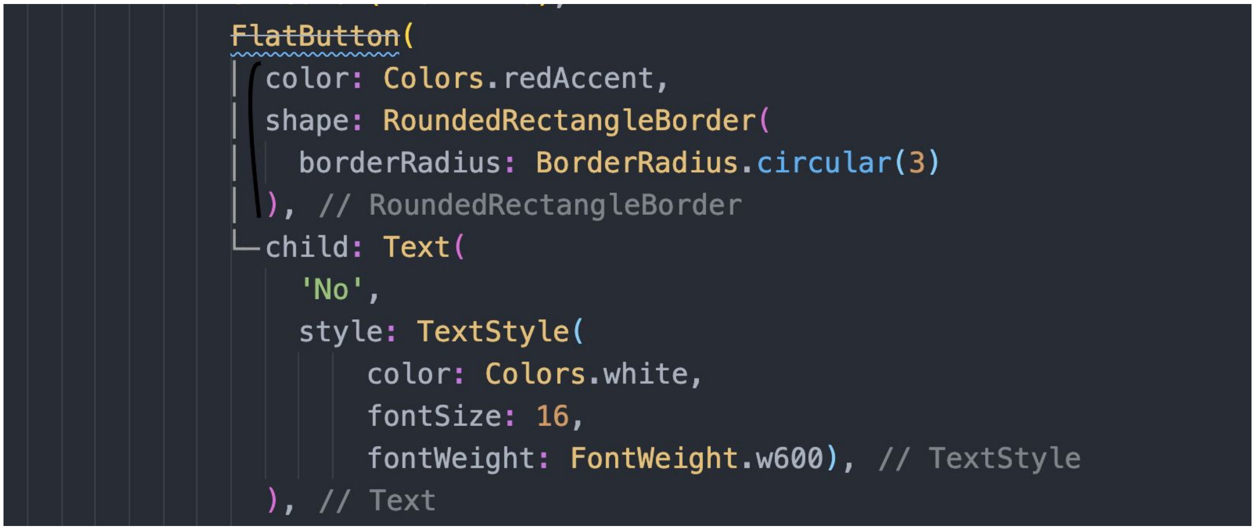
2. **FlatButton Deprecation Error (Without color & shape) :**



```
  ),
), // TextButton
SizedBox(width: 10),
~~FlatButton~~(
└child: Text(
    'No',
    style: TextStyle(
      color: Colors.white,
      fontSize: 16,
      fontWeight: FontWeight.w600), // TextStyle
  ), // Text
  onPressed: () => Navigator.pop(context),
), // FlatButton
```

To fix this error, just rename **FlatButton** by **TextButton**.

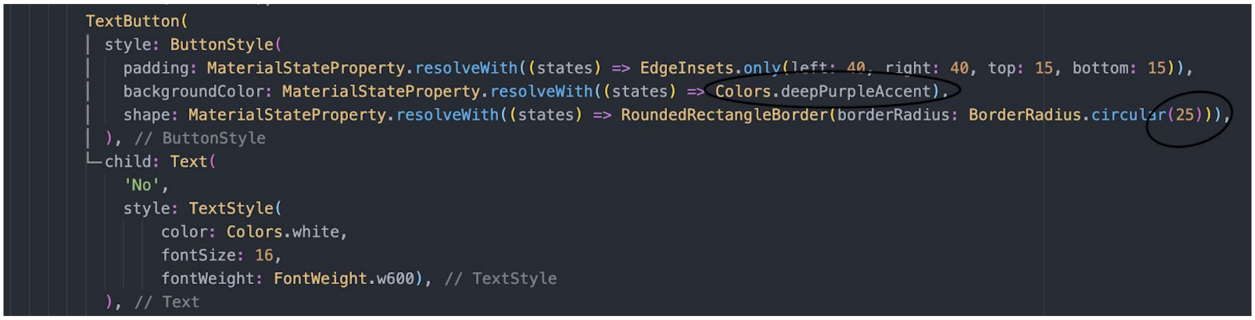
3. **FlatButton with color & shape Error** : If the deprecated FlatButton has color & shape like the picture below :



```
FlatButton(
  color: Colors.redAccent,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(3)
  ), // RoundedRectangleBorder
  child: Text(
    'No',
    style: TextStyle(
        color: Colors.white,
        fontSize: 16,
        fontWeight: FontWeight.w600), // TextStyle
  ), // Text
```

Then first rename **FlatButton** to **TextButton** and then remove the **color & shape** parameter with value and also **remember the color & radius.**

Now add some code like this :



```
TextButton(
  style: ButtonStyle(
    padding: MaterialStateProperty.resolveWith((states) => EdgeInsets.only(left: 40, right: 40, top: 15, bottom: 15)),
    backgroundColor: MaterialStateProperty.resolveWith((states) => Colors.deepPurpleAccent),
    shape: MaterialStateProperty.resolveWith((states) => RoundedRectangleBorder(borderRadius: BorderRadius.circular(25))),
  ), // ButtonStyle
  child: Text(
    'No',
    style: TextStyle(
        color: Colors.white,
        fontSize: 16,
        fontWeight: FontWeight.w600), // TextStyle
  ), // Text
```

Make sure you have replaced the old color & shape value. That’s it.