**FLUTTER E-COMMERCE ADMIN PANEL**

Designed by Coding with T

**Quick Setup Guide for Your Flutter E-Commerce Admin Panel**

**Introduction:** Congratulations on your purchase of our Flutter Admin Panel! This guide will walk you through the essential steps to set up and get started quickly.

**Requirements:**

Before we begin, ensure you have the following prerequisites in place:

1. **Flutter Installed:** Make sure you have Flutter installed on your development machine. If not, you can download it from the official website: [Flutter Install](https://flutter.dev/docs/get-started/install), & watch a YouTube tutorial about [how to install and setup](https://codingwitht.com/how-to-install-flutter-on-android-studio-2022-flutter-crash-course/).
2. **Code Files:** Ensure you have downloaded and unzipped the code files from your purchase.

**Installation Steps:**

Follow these simple steps to set up your Flutter E-Commerce admin panel:

**Step 1: Navigate to the App Directory**

Using your terminal or command prompt, navigate to the directory where you've saved the Flutter E-Commerce Admin Panel code files.

bash

*cd /path/to/your/app/directory*

**Step 2: Install Dependencies**

Run the following command to install the required dependencies:

bash

*flutter pub get*

**Step 3: Configure Firebase (if applicable)**

If your app uses Firebase, configure it by following these steps:

* Create a Firebase project on the Firebase Console.
* Follow [this tutorial](https://codingwitht.com/how-to-setup-firebase-in-flutter-firebase-setup-2022/) to setup and run Firebase in your project.

**Step 4: Init the Firebase:**

Now, it's time to initialize the firebase. Use the following command:

bash

*firebase init*

You will have to do the following steps:

* Are you ready to proceed? Yes
* Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (firestore, hosting(configure file), Storage)
* === Project Setup
* Please select an option: Use an existing project
* Select a default Firebase project for this directory: Enter
* === Firestore Setup
* What file should be used for Firestore Rules? firestore.rules (Enter)
* File firestore.rules already exists. Do you want to overwrite it with the Firestore Rules from the Firebase Console? Yes
* What file should be used for Firestore indexes? firestore.indexes.json (Enter)
* File firestore.indexes.json already exists. Do you want to overwrite it with the Firestore Indexes from the Firebase Console? Yes
* === Hosting Setup
* What do you want to use as your public directory? build/web
* Configure as a single-page app (rewrite all urls to /index.html)? Yes
* Set up automatic builds and deploys with GitHub? No
* === Storage Setup
* What file should be used for Storage Rules? storage.rules
* File storage.rules already exists. Overwrite? Yes

**Step 5: Release the Build**

Now, it's time to release the build. Use the following command:

bash

*flutter build web --release --no-tree-shake-icons*

**Step 6: Deploy the Build**

Now we have to deploy the build. Use the following command:

bash

*firebase deploy --only hosting*

**Step 7: Register Admin**

* Go to lib => features => authentication => controller => login\_controller.dart
* Copy function name registerAdmin()
* Go to lib => features => authentication => screens => login => widgets => login\_form.dart
* Find ElevatedButton() in the end
* Replace controller.emailAndPasswordSignIn() with controller.registerAdmin()
* Modify Admin Credentials
* Go to lib => utils => constants => text\_strings.dart
* Change adminEmail and adminPassword
* Sign In
* Run the code and press Sign In button
* Modify SignIn buttton function
* After Logged in
* Go to lib => features => authentication => screens => login => widgets => login\_form.dart
* Find ElevatedButton() in the end
* Replace controller.registerAdmin() with controller.emailAndPasswordSignIn()

**Step 8: Media Unable to load Images [Create Index in Firestore Firebase]**

Add index in the Firestore index by going into Firebase console of your project. Select Firestore database and then select indexes from top menu.

Click on Add index button in the composite tab and fill the form as below.

Collection Id = Images

Fields to index:

mediaCategory = Ascending

createdAt = Descending

\_\_name\_\_ = Descending

Query Scopes = Collection

Note: In the name there are two underscores at the start and end. Wait for the Index to be Enabled.

**Step 9: Enable CORS to view Images**

Login to your google cloud console:

https://console.cloud.google.com/home.

Select the same Firebase Project from the top left menu. Click on "Activate Google Cloud Shell" in the upper right corner icon.

At the bottom of your window, a shell terminal will be shown, where gcloud and gsutil are already available. Execute the command shown below. It creates a json-file which is needed to setup the cors-configuration for your bucket. This configuration will allow every domain to access your bucket using XHR-Requests in the browser.

echo '[{"origin": ["\*"],"responseHeader": ["Content-Type"],"method": ["GET", "HEAD"],"maxAgeSeconds": 3600}]' > cors-config.json

Replace YOUR\_BUCKET\_NAME with your actual bucket which is in the Firebase Console -> Storage -> Copy the gs://,,, bucket name and replace with the below command.

gsutil cors set cors-config.json gs://YOUR\_BUCKET\_NAME

Run this command in the Google shell terminal and you are done.

To check if everything worked as expected, you can get the cors-settings of a bucket with the following command:

gsutil cors get gs://YOUR\_BUCKET\_NAME

**Step 10: Testing and Debugging**

Test your app thoroughly to ensure it functions as expected. Use Flutter's debugging tools to identify and fix any issues.

**Step 11: Deployment**

When you're satisfied with your app, follow the deployment guidelines to release it to your audience.

**Conclusion:**

You're now ready to start using and customizing your Flutter E-Commerce Admin Panel. If you encounter any issues or have specific questions, refer to the documentation included with your purchase for detailed guidance.

Thank you for choosing our Flutter E-Commerce Admin Panel. We wish you great success with your project!