

Maintenance Guide

1.About:

2HandProduct is a mern stack application with multiple different layers, utilizing different technologies:

- 1.A front end application/website created using React, which can be deployed to a hosted website like render or a mobile application.
- 2.A back end server created using node.js with express.js framework, hosted on render, along with a MongoDB database on Atlas.
- 3.A Firebase project to handle login Authentication, Cloud Storage, Firestore Database and Hosting
- 4.On the home page, scrolling down reveals the latest post and any posts with offers.
- 5.Our website is designed to be responsive and can adapt to various screen sizes, including phones, laptops, and computers. This adaptability is achieved using CSS, particularly with the Tailwind CSS framework.

2.Front End:

The front end is developed using React, using CSS Tailwind and MaterialUI for designing, React Hook Form for form validation, firebase, Redux Toolkit for state management, and more than 50 react Hooks

3.Back End:

The back end was developed using the express.js framework in node.js, and a MongoDB (on Atlas) database.

Running the project

In order to run and update the front- back – end sides project:

1. Visual Studio Code (VSC):

Install Visual Studio Code from the official website (<https://code.visualstudio.com/>).

2. Node.js and npm:

Install Node.js and npm from the official website (<https://nodejs.org/>).

Ensure that Node.js and npm are added to your system's PATH.

3. Clone the Repository:

Download or clone the project repository containing the client and server code from version control system like Git.

4. Install Dependencies:

4.1. Client: Navigate to the client directory in the terminal (`cd client`) and run `npm install` to install the client-side dependencies.

4.2. Server: Similarly, navigate to the server directory (`cd server`) and run `npm install` to install the server-side dependencies.

5. Start the Server:

- In the terminal, navigate to the server directory.
- Run the server using the command `npm run dev` or `node server.js`.

6. Start the Client:

- Open a new terminal window or tab.
- Navigate to the client directory.
- Run the client using the command `npm start`.

7. Access the Application:

Open a web browser and navigate to `http://localhost:3000` to view the React client.

The server will be running on a different port (5000) and is accessed the client's code.

Project Structure:

Front - End Structure The project starts in the ``client/src /App.js`` component, which wraps the entire application. The theme gets initialized here, and a Redux Provider wraps the rest of the app so that we can access the global state from everywhere in the App.

The global style definitions, including theme colors and default component stylings, can be found in the ``client/src/index.css`` file and another 3 files: `postcss.config.js`, `tailwind.config.js`, `vite.config.js`.

Folders in the ``client/src`` directory:

All the base components that get reused throughout the project can be found in the ``Components`` folder, where they are divided into three categories:

Pages: The "page" file likely contains the code for each individual page of our website. Each page represents a different view or section of your site that users can navigate to. Pages often include components and content specific to that page's purpose.

Components: The "components" file contains reusable parts of our website that are used across multiple pages. Components can include things like navigation bars, buttons, forms, or any other elements that appear in different parts of your site.

Models: The "models" file contains the data structures that represent the products available for sale on our website. These models define the attributes of each product, such as its name, description, price, and any other relevant information.

Also has Firebase and Redux configuration files and services, as well as the API.

Back - End Structure follows the MVC (Model-View-Controller) pattern, where models define the data structure, controllers handle business logic, and routes manage the flow of requests between clients and controllers. This organization helps keep your server-side code modular and maintainable.

Folders in the ``server/src`` directory:

index.js (Server Entry Point): This file is the main file for your server-side code. It uses Express.js to handle HTTP requests and connect to MongoDB. It listens on port 5000 for incoming requests from clients.

Controllers: The "controllers" directory contains files with JavaScript functions that handle requests from clients and return responses. Each file in this directory likely corresponds to a different type of resource or functionality in your application.

Routes: The "routes" directory contains files that define the routes for your server. These files receive incoming requests from clients and direct them to the appropriate controller function based on the request's path and method (e.g., GET, POST, PUT, DELETE).

Models: The "models" directory contains files that define the schema and structure of our MongoDB database. Each file likely corresponds to a different collection or type of data in your database, such as cars in your example.

Also there another file called **utils** that's contain to code the first one checks the authentication and authorizations of the request user and the second one catch the errors.

Deploying

To deploy the application:

1. The both client and server is hosted on Render hosting.
2. Link how to deploy on Render our mern stack project via GitHub step by step:
<https://www.youtube.com/watch?v=ZsFwpjFmpFQ&t=774s>

