Seshadri Rao Gudlavalleru Engineering College

PROJECT DOCUMENTATION

SmartShop:Your Digital Grocery Store Experience

Team ID: LTVIP2025TMID59025

Team Members:
Shaik Asmin(Team Leader)
Shaik Mahammad Riyaz
Shaik John bee
Shabbir Khan

SMART SHOP WEB USING MERN

INTRODUCTION

SmartShop Web is your ultimate online shopping companion, designed to bring convenience, personalization, and intelligence to your digital retail experience. Whether you're looking for the best deals, comparing product features.

DESCRIPTION

SmartShop is an easy-to-use online shopping platform that helps you find the best products at the best prices. It lets you browse, compare, and buy items from different stores all in one place. With smart features like product recommendations, price tracking, and secure checkout, SmartShop makes your shopping experience faster, easier, and more enjoyable.

1. **All-in-One Shopping Experience**
No more switching between websites or apps. SmartShop lets you browse products from multiple online stores in one place, saving you time and effort.
2. **Smart Search & Filters**
Find exactly what you need with intelligent search options, advanced filters, and category sorting that help you narrow down your choices quickly.
3. **Price Comparison Made Easy**
Instantly compare prices from different sellers so you always get the best deal. Our price tracking feature also notifies you when a product drops in price.
4. **Personalized Recommendations**
Get suggestions tailored to your preferences and browsing history. SmartShop learns what you like and helps you discover great products effortlessly.
5. **Wishlist & Cart Management**
Save items for later with your wishlist or add them to your cart to check out quickly. Organize your shopping in a way that works for you.

6. **Secure Checkout**
Enjoy safe and fast checkout with trusted payment gateways. Your personal and financial information is always protected.

Features Of Smart Shop :

SmartShop lets users search for products across multiple online stores from a single platform. No need to open different tabs—just type in what you're looking for, and SmartShop will bring you the best options available. Be the first to know about new deals, limited-time offers, and seasonal discounts. SmartShop alerts you in real-time so you never miss a chance to save.

1. Unified Product Search

- Search products from multiple online stores in one place.
- Filter by category, price, brand, rating, and more.

2. Price Comparison

- Compare prices for the same product across different sellers.
- Display best deals and alternative buying options.

3. Price Drop Alerts

- Set alerts for specific products.
- Get notified when the price drops or a sale starts.

4. Personalized Recommendations

- AI-powered suggestions based on your browsing and shopping history.
- Smart categories like “Top Picks for You” or “Trending Deals.”

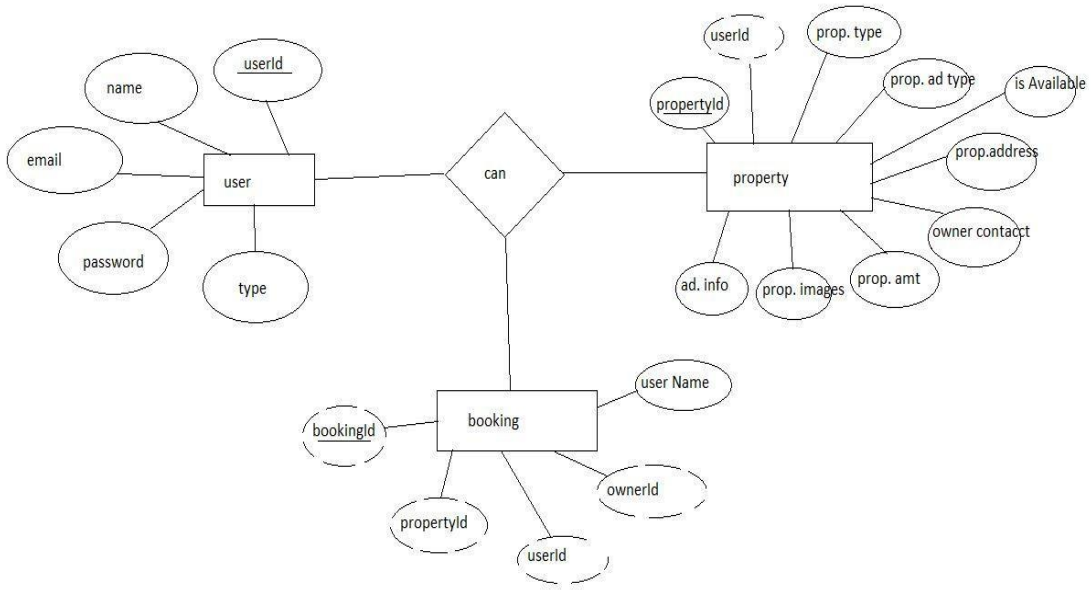
5. Wishlist & Save for Later

- Save products to your wishlist.
- Organize items into categories (e.g., Gifts, Tech, Fashion).

6. Deal & Discount Notifications

- Real-time alerts on flash sales, coupons, and limited-time offers.
- Personalized deals based on shopping habits.

ER DIAGRAM



PREREQUISITE:

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

✓Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.
Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.
Download: https://nodejs.org/en/download/

Installation instructions: https://nodejs.org/en/download/package-manager/

npm init

✓Express.js:

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.
Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.
Installation: Open your command prompt or terminal and run the following command:

npm install express

✓MongoDB:

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.
Download: https://www.mongodb.com/try/download/community

Installation instructions: https://docs.mongodb.com/manual/installation/

✓Moment.js:

Moment Js is a JavaScript package that makes it simple to parse, validate, manipulate, and display date/time in JavaScript. Moment. js allows you to display dates in a human-readable format based on your location. Install React.js, a JavaScript library for building user interfaces.
Follow the installation guide: https://momentjs.com/

✓React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.
Install React.js, a JavaScript library for building user interfaces.
Follow the installation guide: https://reactjs.org/docs/create-a-new-react-app.html

✓Antd:

Ant Design is a React. js UI library that contains easy-to-use components that are useful for building interactive user interfaces. It is very easy to use as well as integrate. It is one of the smart options to design web applications using react.
Follow the installation guide: https://ant.design/docs/react/introduce

✓HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

✓Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:
https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

✓Front-end Framework: Utilize Reactjs to build the user-facing part of the application, including entering the booking room, status of the booking, and user interfaces for the admin dashboard. For making better UI we have also used some libraries like material UI and bootstrap.

✓Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at: https://gitscm.com/downloads

✓Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from https://code.visualstudio.com/download

To run the existing Video Conference App project downloaded from GitHub:
Follow below steps:
Clone the Repository:

- Open your terminal or command prompt.
- Navigate to the directory where you want to store the e-commerce app.
- Execute the following command to clone the repository: git clone: https://github.com/awdhesh-student/house-rent.git

Install Dependencies:

• Navigate into the cloned repository directory:
cd house-rent
• Install the required dependencies by running the following commands:
cd frontend npm
install cd ../backend
npm install

Start the Development Server:
• To start the development server, execute the following command: npm start
• The house rent app will be accessible at http://localhost:3000

You have successfully installed and set up the online complaint registration and management app on your local machine. You can now proceed with further customization, development, and testing as needed.

Roles and Responsibilities:

The project has 2 types of users – Renter and Owner and the other will be Admin which takes care of all the users. The roles and responsibilities of these two types of users can be inferred from the API endpoints defined in the code. Here is a summary:

Renter/Tenant:

1. Create an account and log in to the system using their email and password.
2. They will be shown automatically all the properties in their dashboard.
3. After clicking on the Get Info, all the information of the property and owner will come and a small form will be generated in which the renter needs to send his\her details.
4. After that they can see their booking in the booking section where the status of booking will be showing “pending”. It will be changed by the owner of the property.

Admin:

1. He/she can approve the user as “owner” for the legit user to add properties in his app 2. He monitors the applicant of all doctors and approves them and then doctors are registered in the app.
3. Implement and enforce platform policies, terms of service, and privacy regulations.

Owner:

1. Gets the approval from the admin for his Owner account.
2. After approval, he/she can do all CRUD operation of the property in his/her account
3. He/she can change the status and availability of the property.

PROJECT STRUCTURE

```
ShopSmart/
├── Backend/
|   ├── db/
|   |   ├── connect.js
|   |   ├── schema.js
|   |   ├── product.js
|   ├── index.js
|   └── package.json
|
├── Frontend/
|   ├── src/
|   |   ├── components/
|   |   ├── pages/
|   |   ├── App.js
|   |   └── index.js
|   └── package.json
```

The first image is of frontend part which is showing all the files and folders that have been used in UI development

The second image is of Backend part which is showing all the files and folders that have been used in backend development

Project Flow:

Before starting to work on this project, let's see the demo.
Project demo

Milestone 1: Project setup and configuration.

- ✓ Folder setup:
  1. Create frontend and
  2. Backend folders

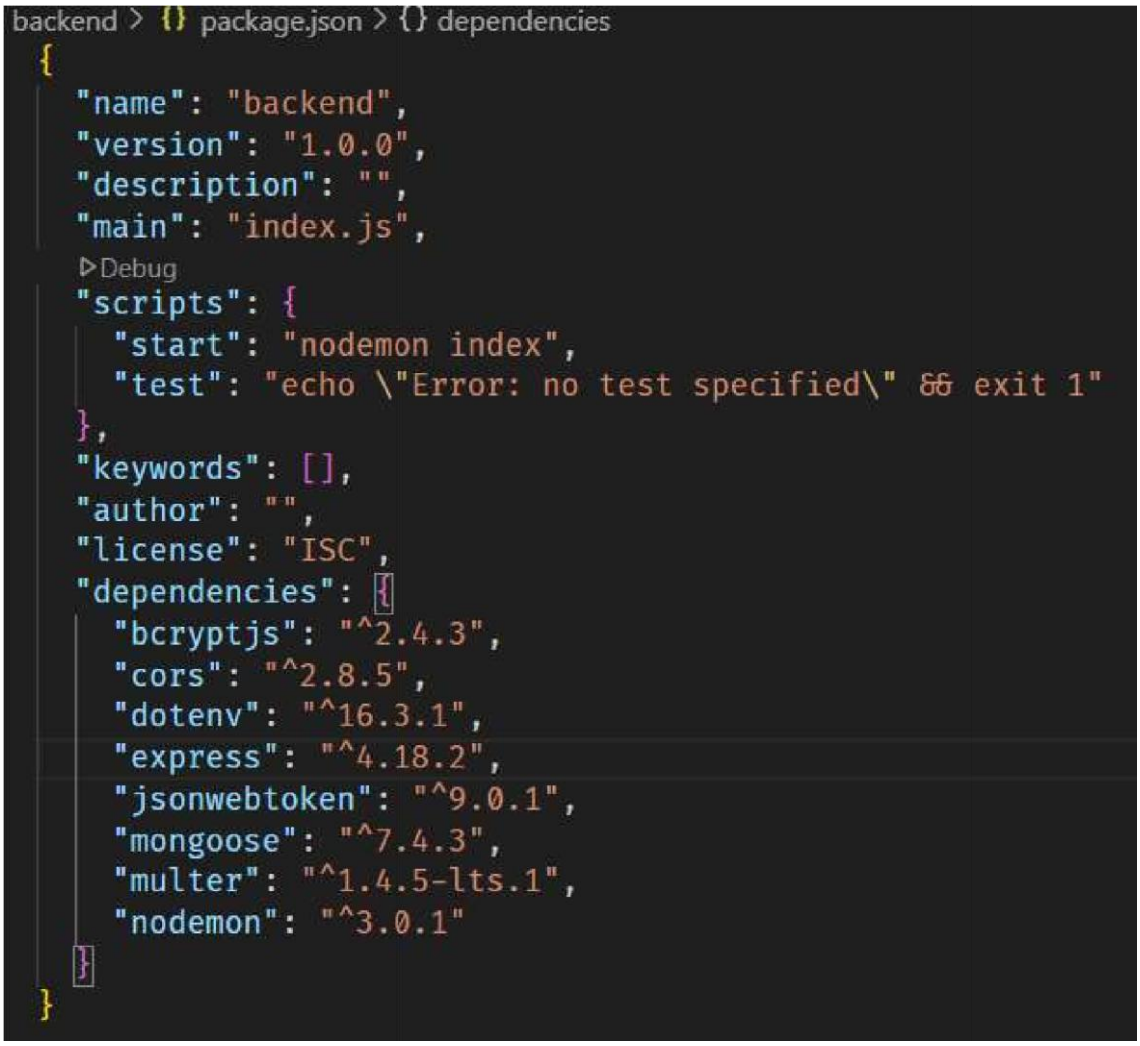
- ✓ Installation of required tools:
  1. Open the frontend folder to install necessary tools

     For frontend, we use: ▪ React
     - Bootstrap
     - Material UI
     - Axios
     - Moment
     - Antd
     - mdb-react-ui-kit
     - react-bootstrap

  2. Open the backend folder to install necessary tools

     For backend, we use:
     - cors
     - bcryptjs
     - express
     - dotenv
     - mongoose
     - Moment

After the installation of all the libraries, the package.json files for the backend looks like the one mentioned below.



Milestone 2: Backend Development

- ✓ Setup express server
  1. Create index.js file in the server (backend folder).
  2. define port number, mongodb connection string and JWT key in env file to access it.
  3. Configure the server by adding cors, body-parser.

- ✓ Configure MongoDB 1. Import mongoose.
  2. Add database connection from config.js file present in config folder
  3. Create a model folder to store all the DB schemas like renter, owner and booking, properties schemas.

✓ Add authentication: for this, You need to make a middleware folder and in that make

Milestone 3: Database

-

,

.

```
const mongoose = require('mongoose');

const connectionOfDb = () => {
  mongoose
    .connect(process.env.MONGO_DB, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    })
    .then(() => {
      console.log('Connected to MongoDB');
    })
    .catch((err) => {
      throw new Error(`Could not connect to MongoDB: ${err}`);
    });
};

module.exports = connectionOfDb;
```

Milestone 4: Frontend Devel opment:

1. Setup React Application:

   Bringing SB Stocks to li fe involves a three -step development process. First, a solid foundation is built using R eact.js. This includes creating the initial applica tion structure, installing necessary librari es, and organizing the project files for efficien t development. Next, the user interface (UI) comes to life. To start the development p rocess for the frontend, follow the below steps.

   - Install required lib raries.
   - Create the structur e directories.

2. Design UI components :

   Reusable components will be created for all the interactive elements you'll see on

authMiddleware.js file for the authentication of the projects and can use it.

sections of SB Stocks, like viewing specific stocks or managing your virtual portfolio.

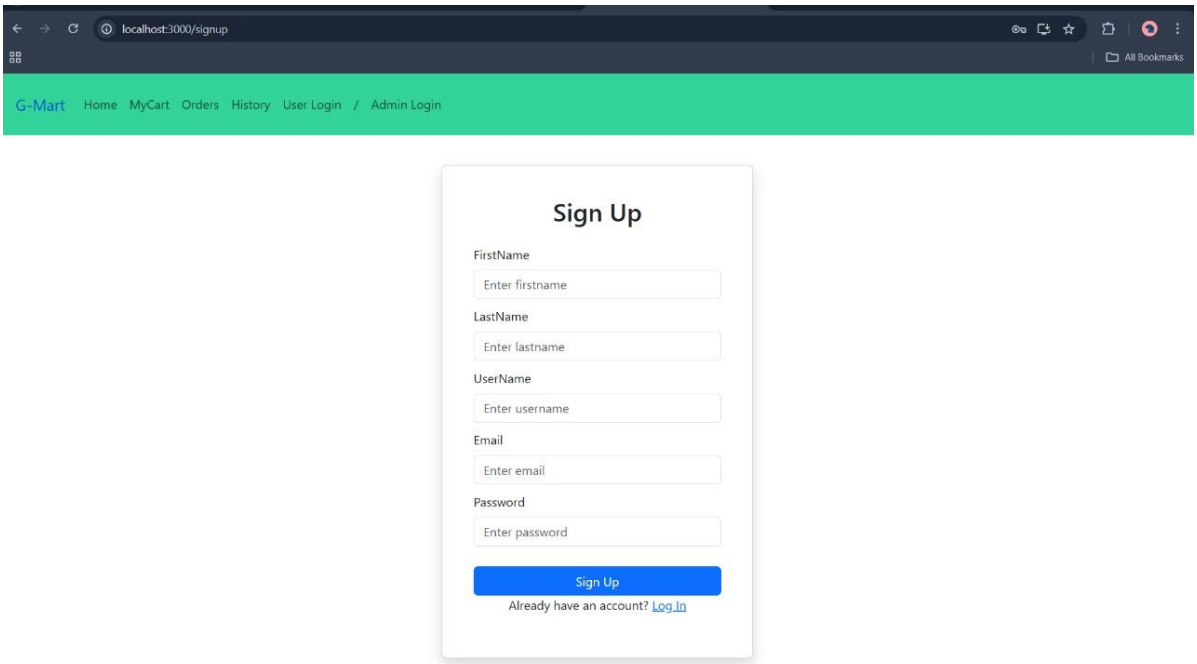
3. Implement frontend logic:

In the final leg of the frontend development, we'll bridge the gap between the visual interface and the underlying data. It involves the below stages.

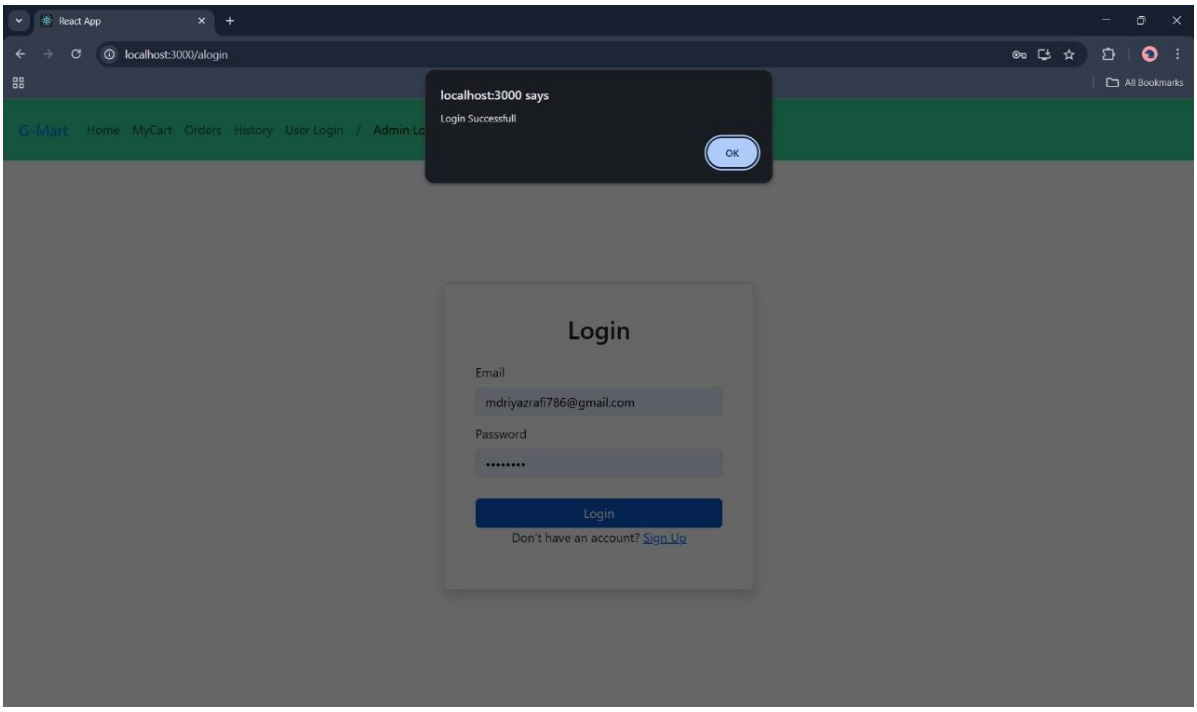
- Integration with API endpoints.
- Implement data binding.

5 . Project Implementation :

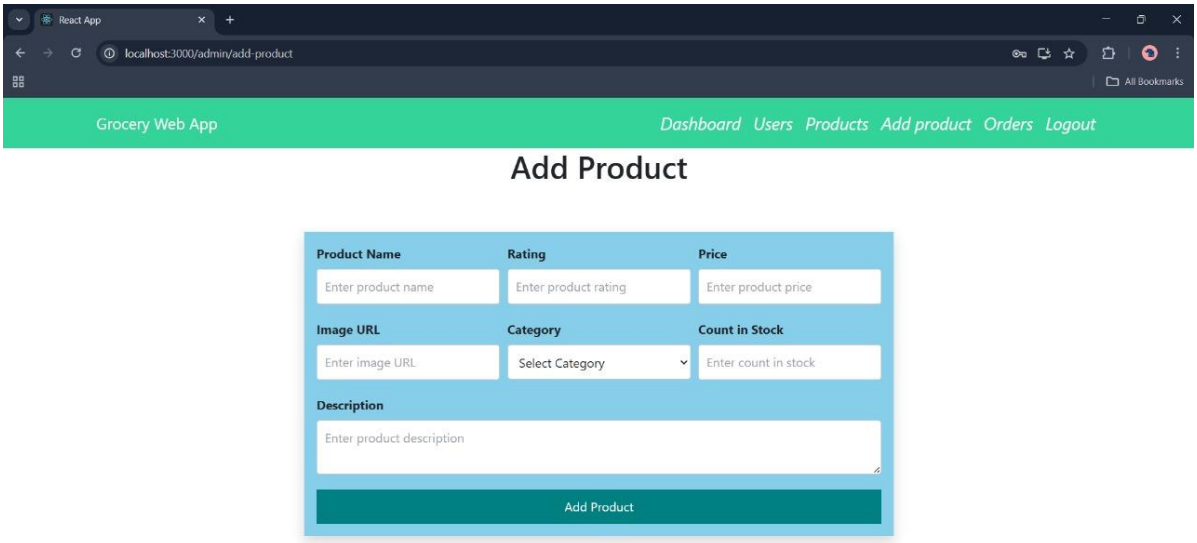
Register or Sign Up:



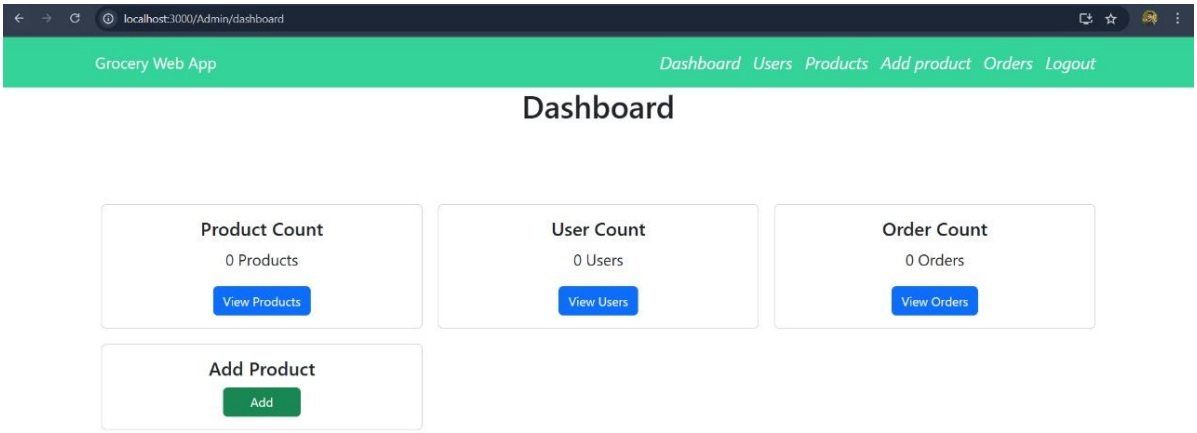
Login:


React App
localhost:3000/alogin
All Bookmarks
localhost:3000 says
Login Successfull
OK
G-Mart
Home
MyCart
Orders
History
User Login
Login
Email
mdriyazrafi786@gmail.com
Password
Login
Don't have an account? Sign Up

Properties:


React App
localhost:3000/admin/add-product
All Bookmarks
Grocery Web App
Dashboard
Users
Products
Add product
Orders
Logout
Add Product
Product Name
Enter product name
Rating
Enter product rating
Price
Enter product price
Image URL
Enter image URL
Category
Select Category
Count in Stock
Enter count in stock
Description
Enter product description
Add Product

Booking History:



Code:
https://github.com/skmdriyaz/SmartShop