MERN stack powered by MongoDB

Naan Mudhalvan – Project Documentation

Introduction

Project Title: Grocery Webapp (online shopping platform)

College: Tagore Engineering College

Department: B.E. CSE IV year 7th Sem

Team ID: NM2024TMID02077

Team Members:

S.no	Student Name	Register No.	Nan Mudhalvan ID	Role
1	J. Pooja	412721104034	211193A8CD7FC38BD9947D5F4AA5519D	Frontend
2	N. Priya Dharshini	412721104037	29D88228BAD4FF8B804468F156BFC660	Backend
3	S. Bashira	412721104008	26E577250645635A85544D0E225860C0	Frontend
4	J.S. Shiny Angel	412721104044	72FB0499FCBBD49B804474DF5F6DD521	Frontend
5	N. Dharshini Priya	412721104011	EC0DE4A8986D412818B49C6E7FE4947E	Backend

Project Overview

Purpose:

G-Mart is an online grocery shopping platform designed to connect customers with local vendors in a seamless, user-friendly environment. The platform enables product listings, customer registration, and order management while fostering a smooth shopping experience. G-Mart also features a robust order tracking system and a review forum to enhance customer satisfaction

Features :

Customer/User:

- Browse products, add to cart/wishlist, and track orders.
- Access order history and participate in a discussion forum.

Vendor/Store Owner:

- Manage products: create, edit, delete listings with images and details.
- Upload product sections (e.g., fresh produce, pantry items).
- View customer orders and manage deliveries.

Admin:

- Manage users, vendors, and products.
- Delete listings and view vendor-specific products.
- Admin access via direct route (no login page).
- Placeholder page for payment gateway.

Architecture :

Frontend:

- Developed using **React** styled with **Bootstrap**.
- Responsive design ensures a seamless shopping experience across all devices.

Backend:

- Built using **Node.js** and **Express.js** for API handling and business logic.

Database:

- **MongoDB** as the database, using **Mongoose** for schema design and data interaction.

Installation:

1. **Clone the repository:**

https://github.com/Pooji2410/Grocery-Webapp

cd grocery Webapp

2. **Install dependencies:**

- **Frontend:**

  cd client

  npm install

- **Backend:**

  cd server

  npm install

```
MONGODB_URI=
'mongodb+srv://manjudharshu54:Dharshini123@dharshistores.hde5r.mongodb.net/stores?retryWrites=true&w=majority&appName=dharshistores'
```

EMAIL_USER='poojajayachandran2410@gmail.com'

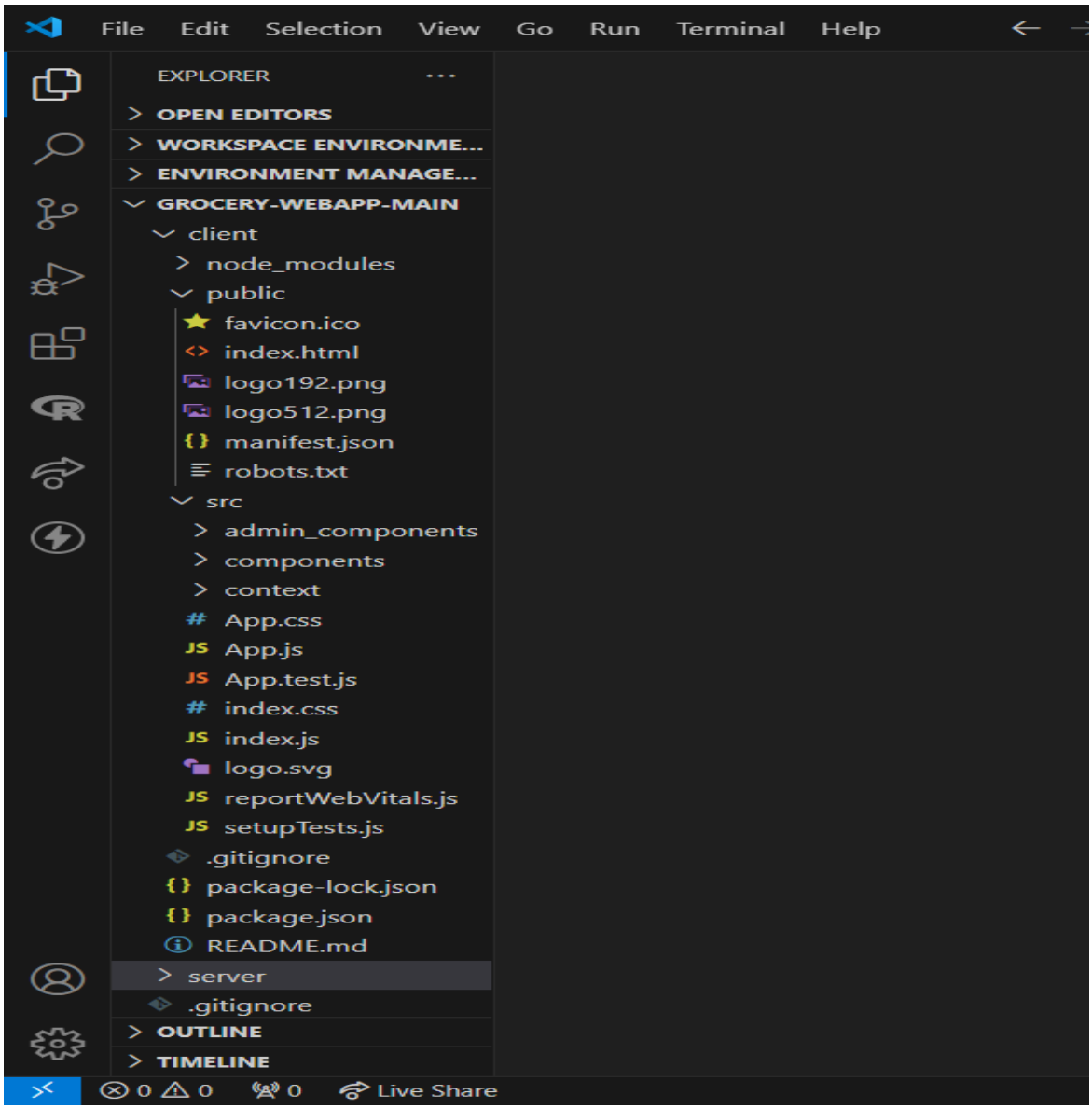
EMAIL_PASS='poojajayachandran_2410'

VITE_REACT_APP_BACKEND_BASEURL=http://localhost:3000

Folder Structure

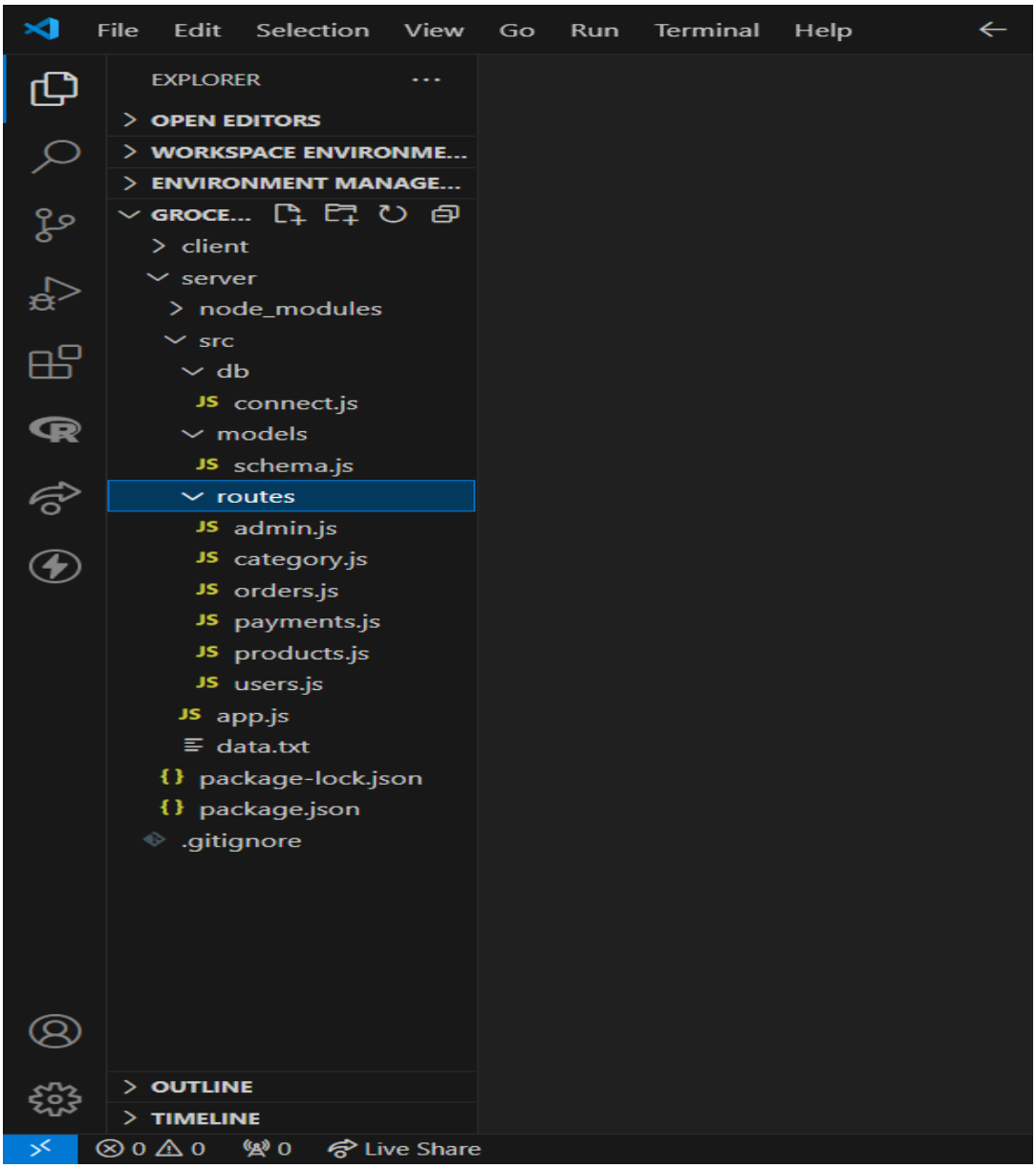
Client (Frontend):

- **src/**: Contains all React components, pages, and assets.
- **Components/**: Reusable UI elements like Navbar, Footer, ProductCard, and SideBar.
- **Pages/**: Organized folders for different user roles and functionalities



Server (Backend):

- **routes/**: Defines API routes for users, vendors, products, orders, and admin.
- **models/**: MongoDB schemas for User, Vendor, Product, Order, and Review.



Running the Application

- **Frontend:**

  cd client

  npm start

- **Backend:**

  cd server

  node app.js

API Documentation

User Routes:

- **POST /signup:** Register a new user.
- **POST /login:** Login user.
- **PUT /add Product**: For adding the Product to the purchased Grocerioes.
- **GET /User -dashboard/product:** Gets all the Product from the platform.
- **GET /user-dashboard/my-Product:** Fetch the Product enrolled by the use.

Admin Routes:

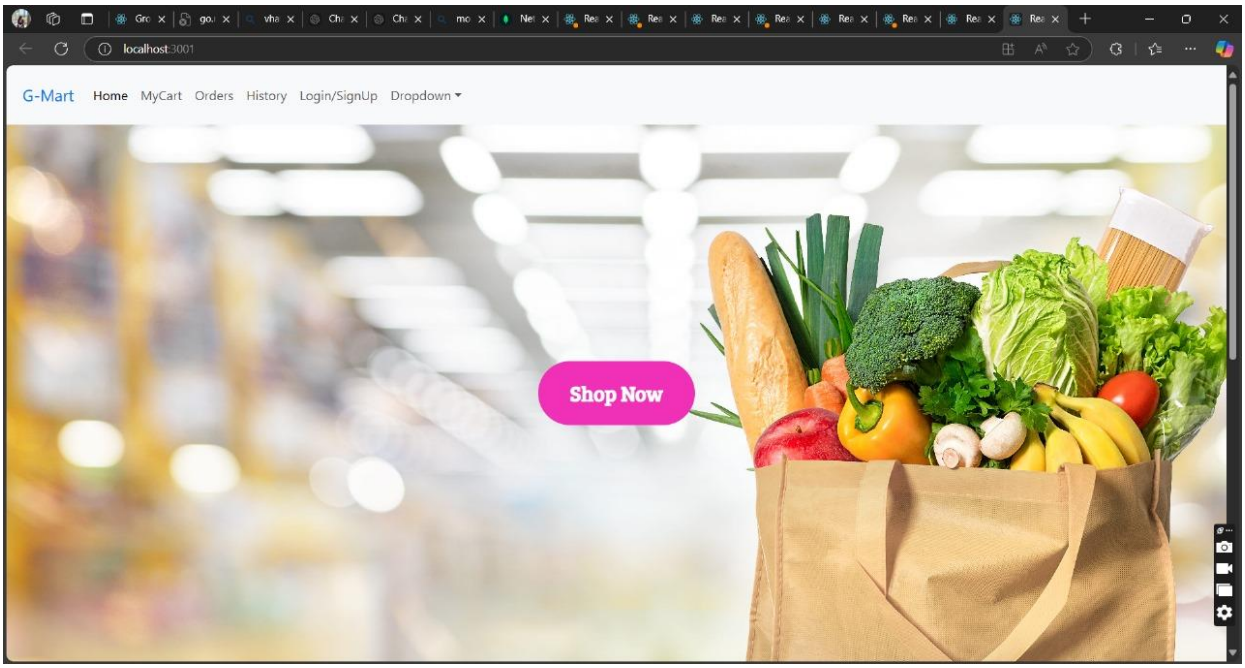
- **GET /users:** View all registered users.
- **GET /Product:** Get all the courses in the platform
- **DELETE /product/:id**: Delete a specific course.

User Interface :

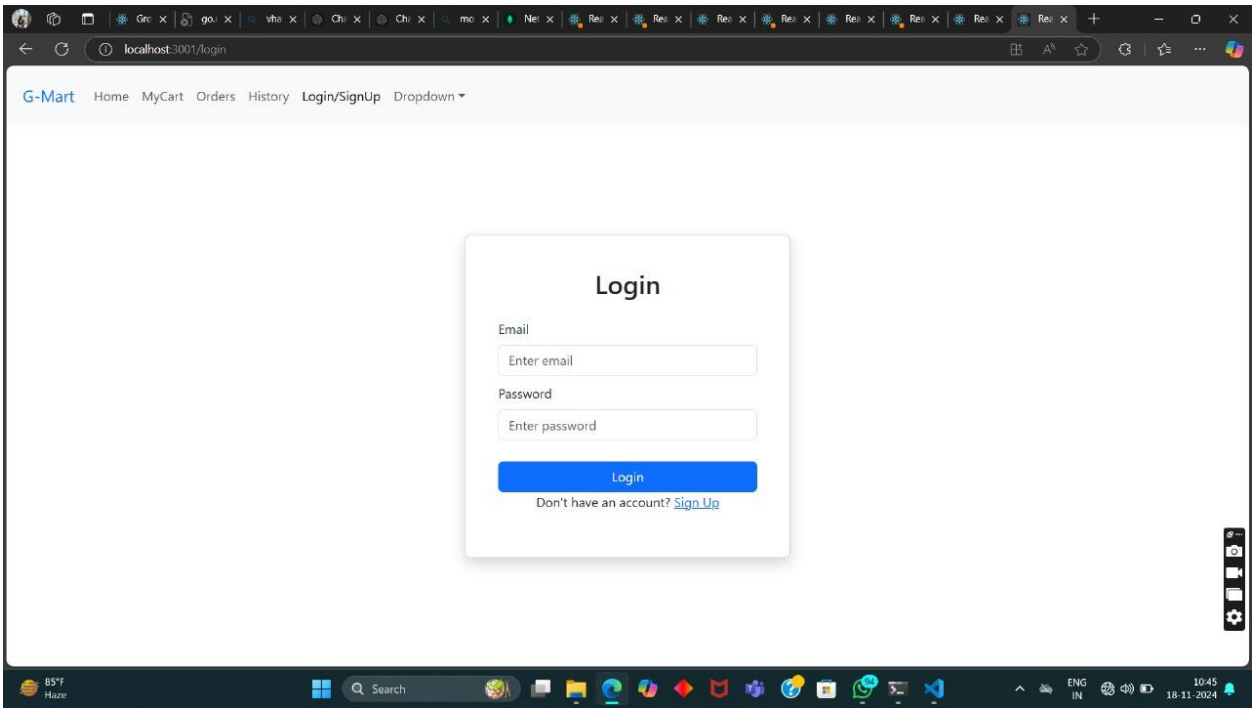
- **Student Dashboard:** Displays enrolled Product, profile management, and discussion forum.
- **Admin View:** User, instructor, and product management in a streamlined UI

SCREENSHOT:

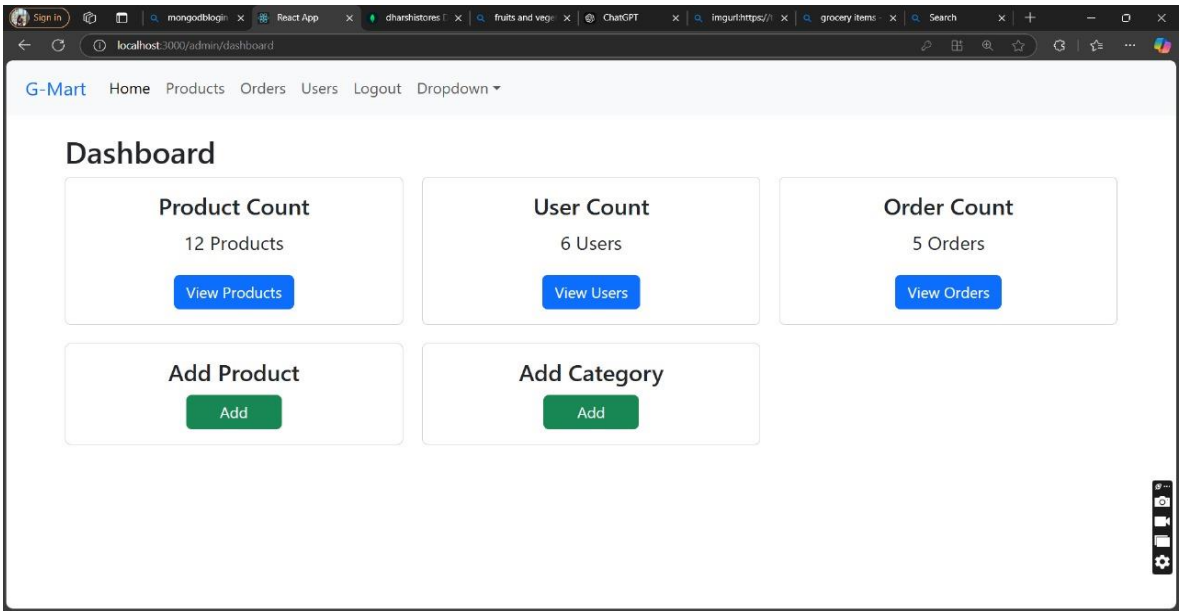
1.Landing page



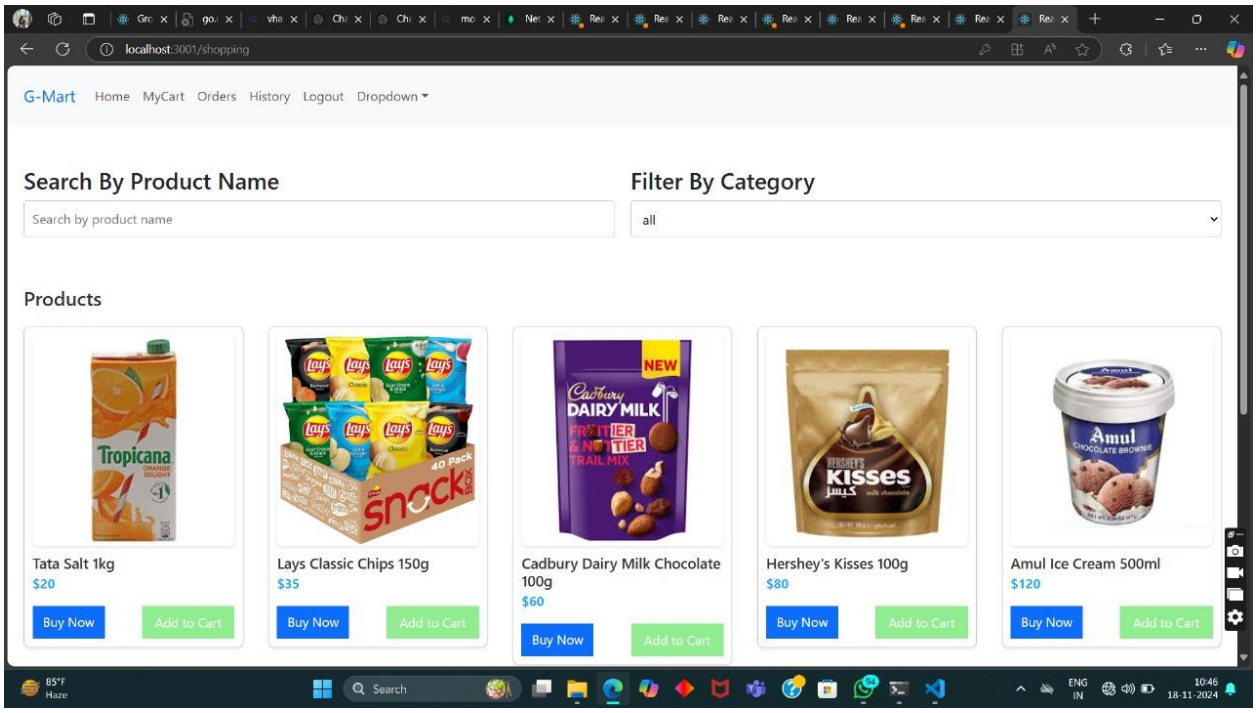
2.User Login:



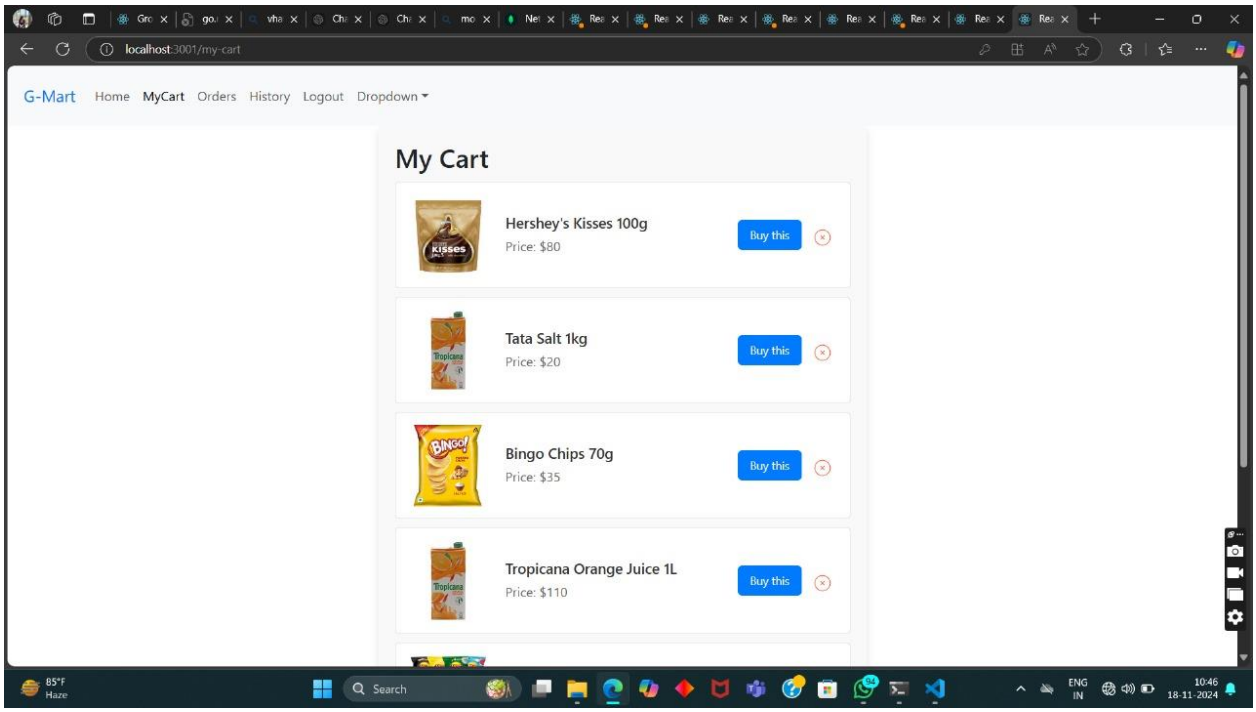
3.Admin Dashboard



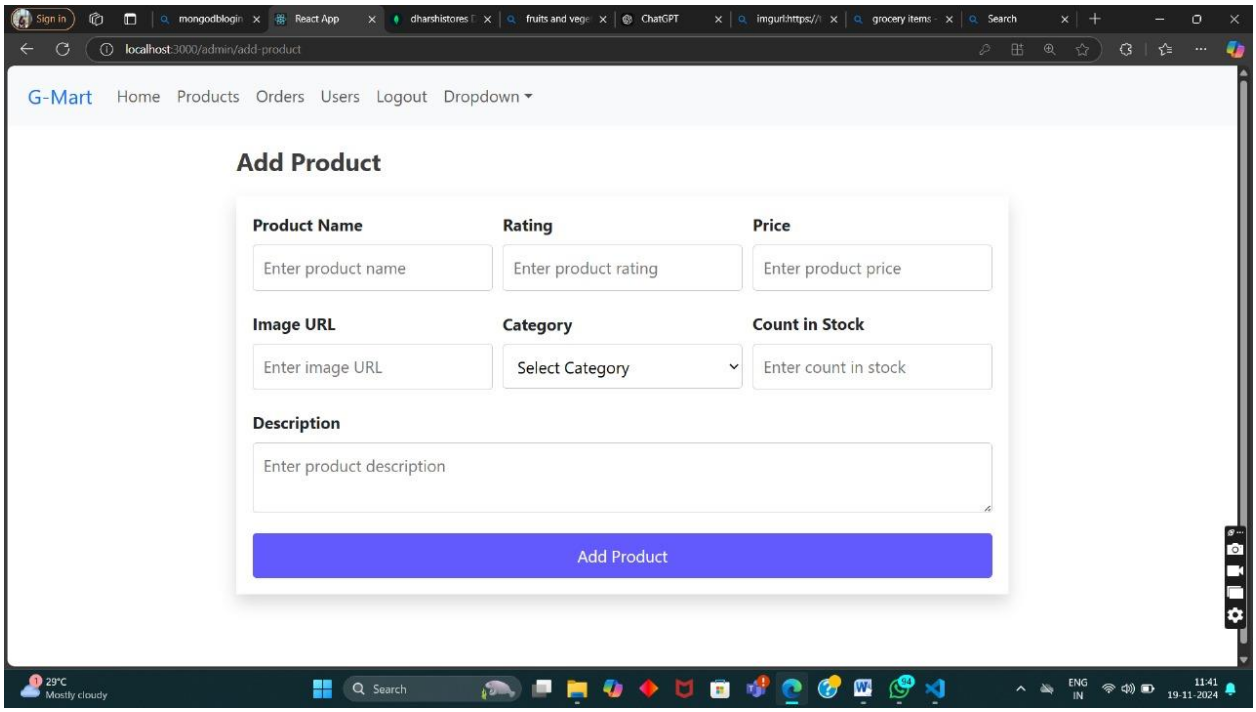
4. User Dashboard:



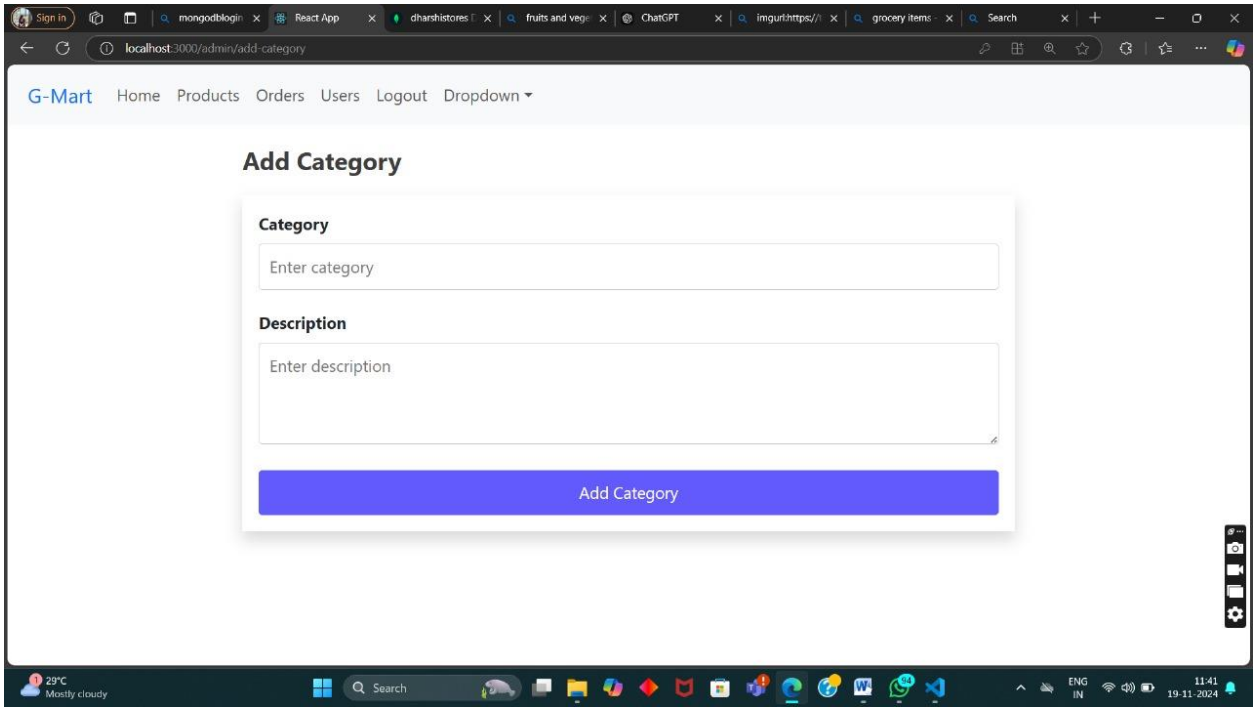
5.Cart Detaials:



6.Add Product Dashboard:



7.Add Category:



8.Order Details: