

# 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 7<sup>th</sup> 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@dhmarshistores.hde5r.mongodb.net/sto  
res?retryWrites=true&w=majority&appName=dharshistores'
```

EMAIL\_USER='poojayachandran2410@gmail.com'

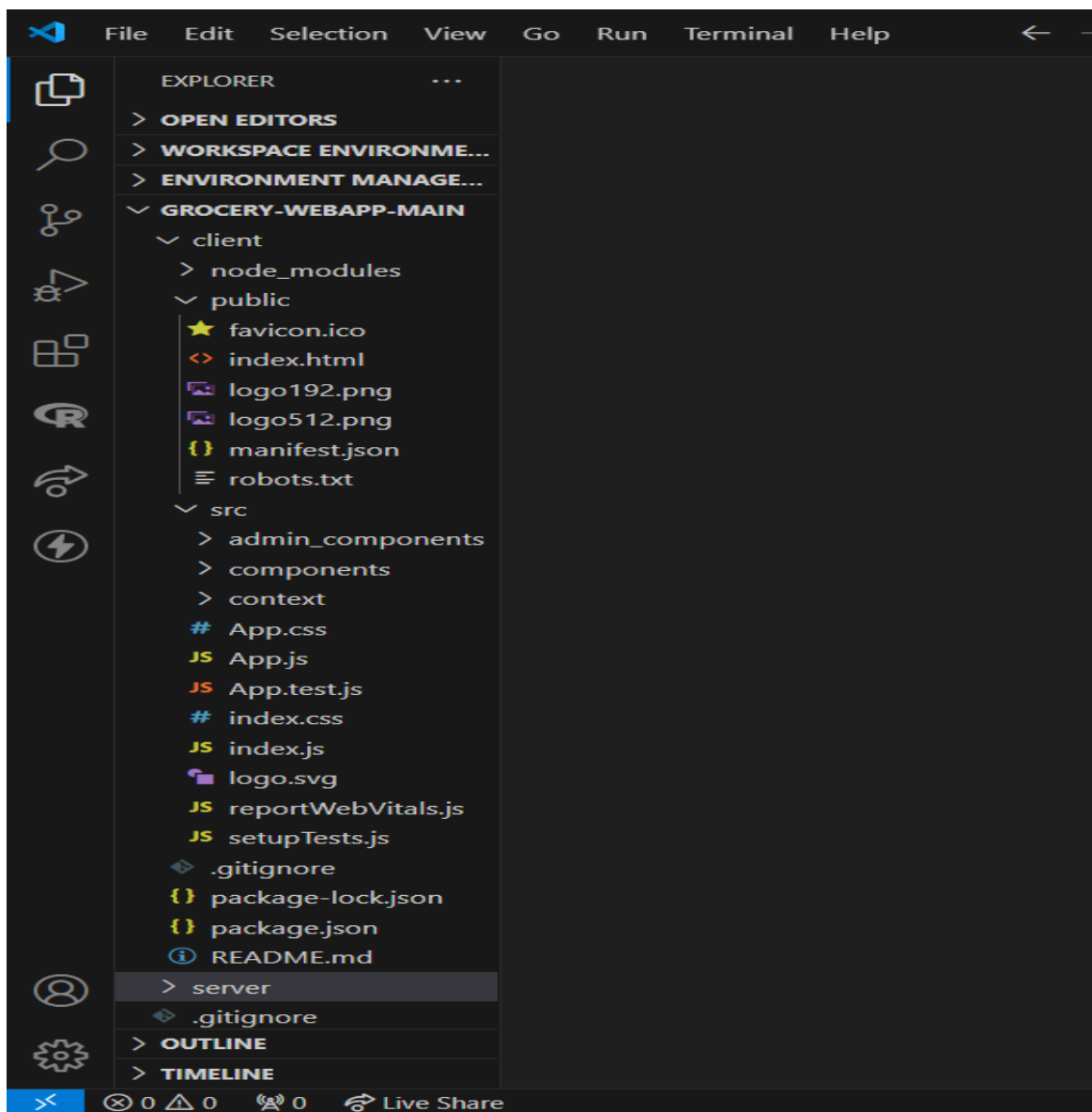
EMAIL\_PASS='poojayachandran\_2410'

**VITE\_REACT\_APP\_BACKEND\_BASEURL**=http://localhost:3000

## Folder Structure

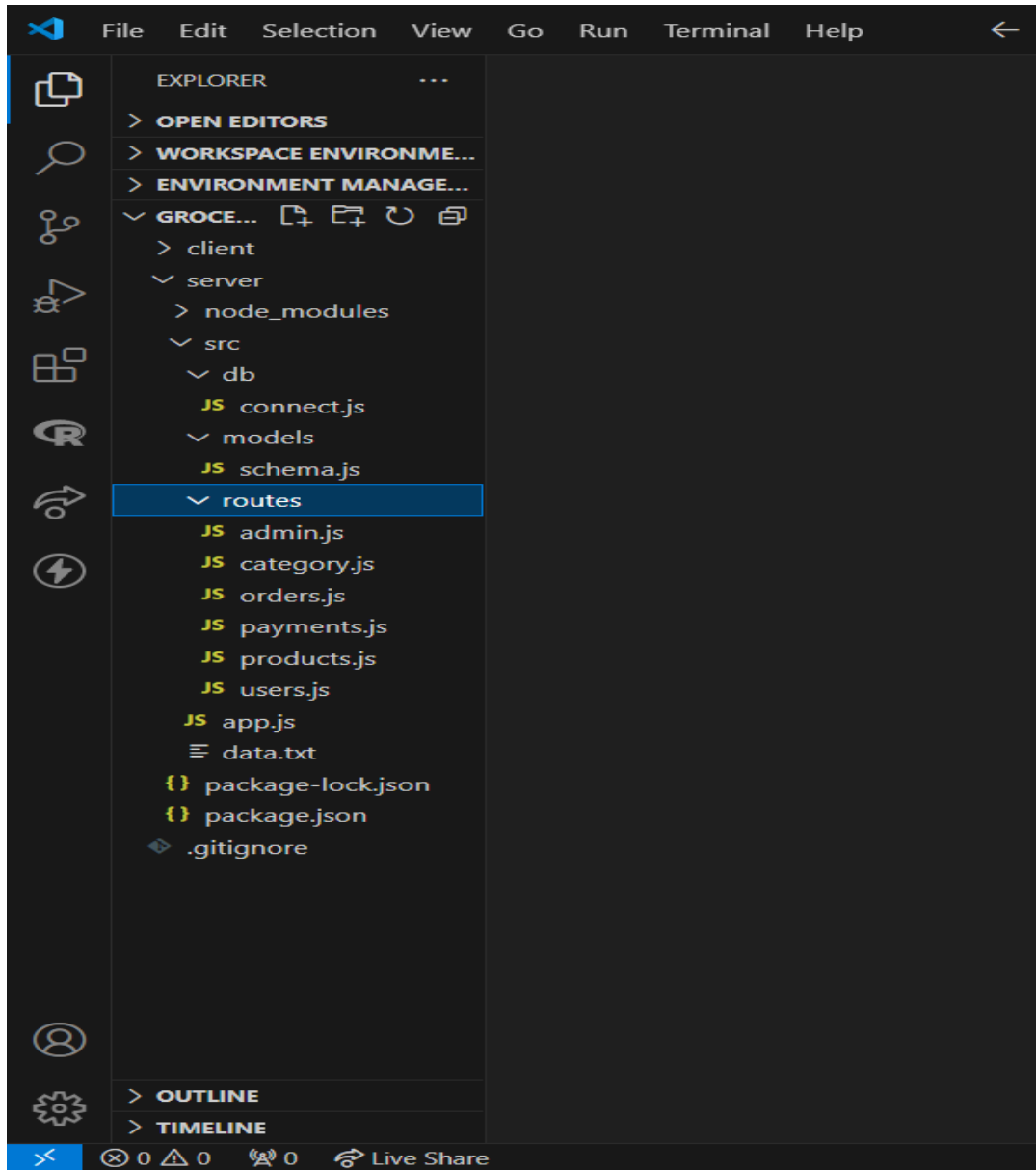
### Client (Frontend):

- **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 Groceries.
- **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 :

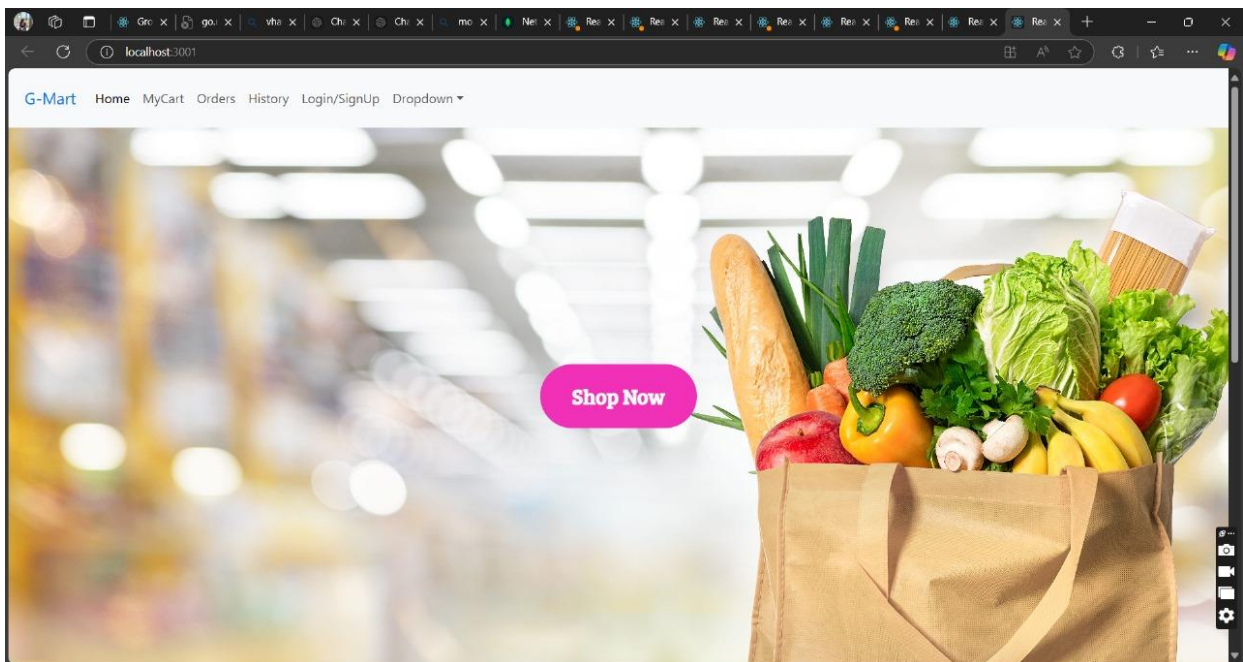
- **Student Dashboard:** Displays enrolled Product, profile management, and discussion forum.

- **Admin View:** User, instructor, and product management in a streamlined

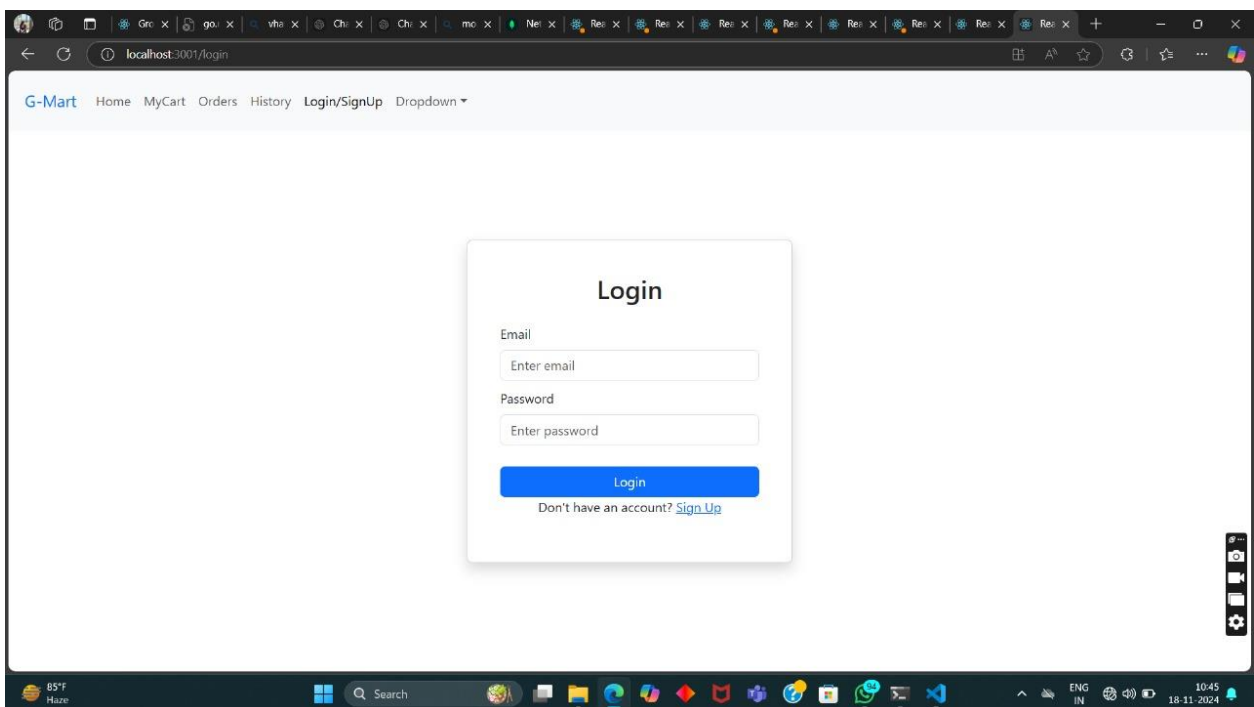
UI

## SCREENSHOT:

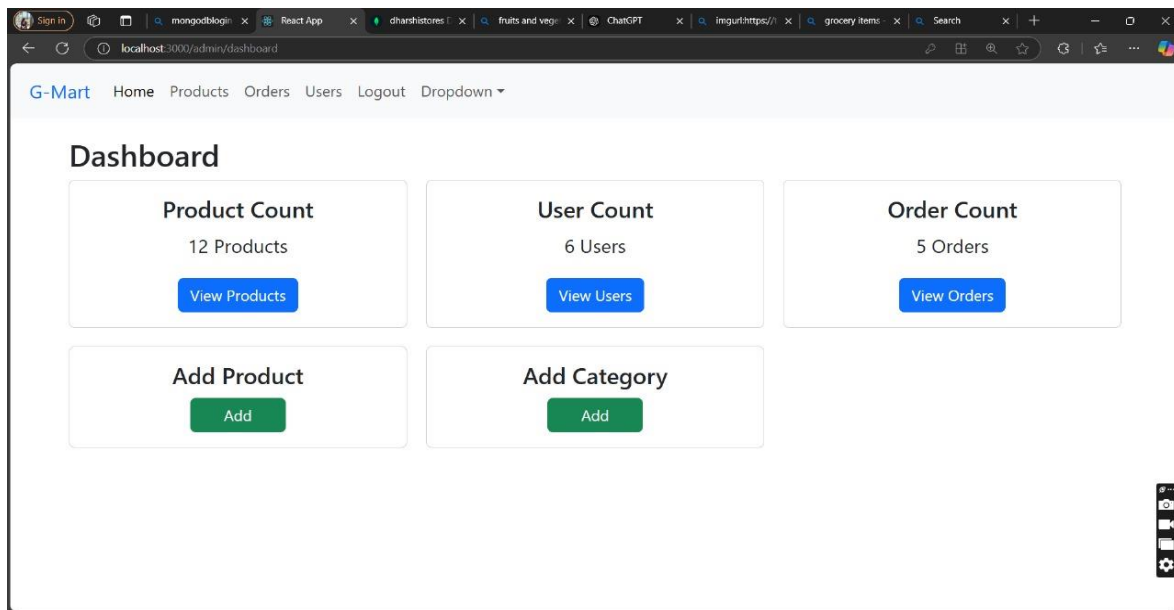
### 1.Landing page



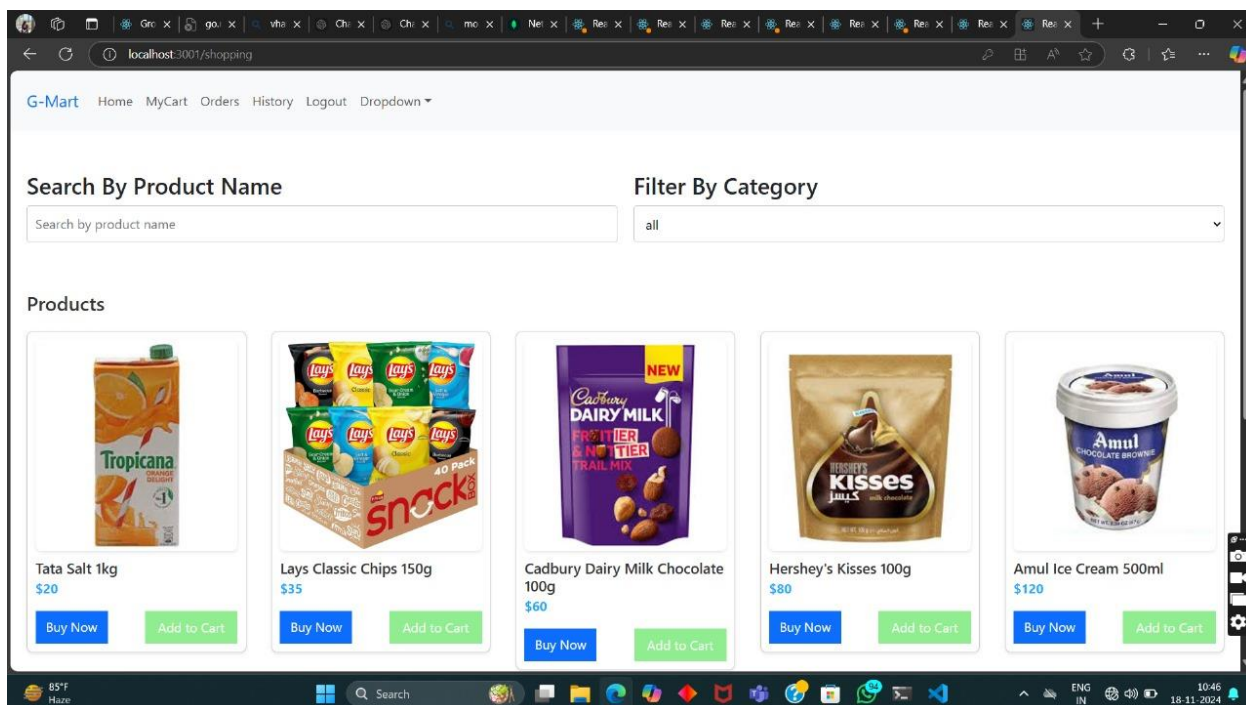
### 2.User Login:



### 3.Admin Dashboard

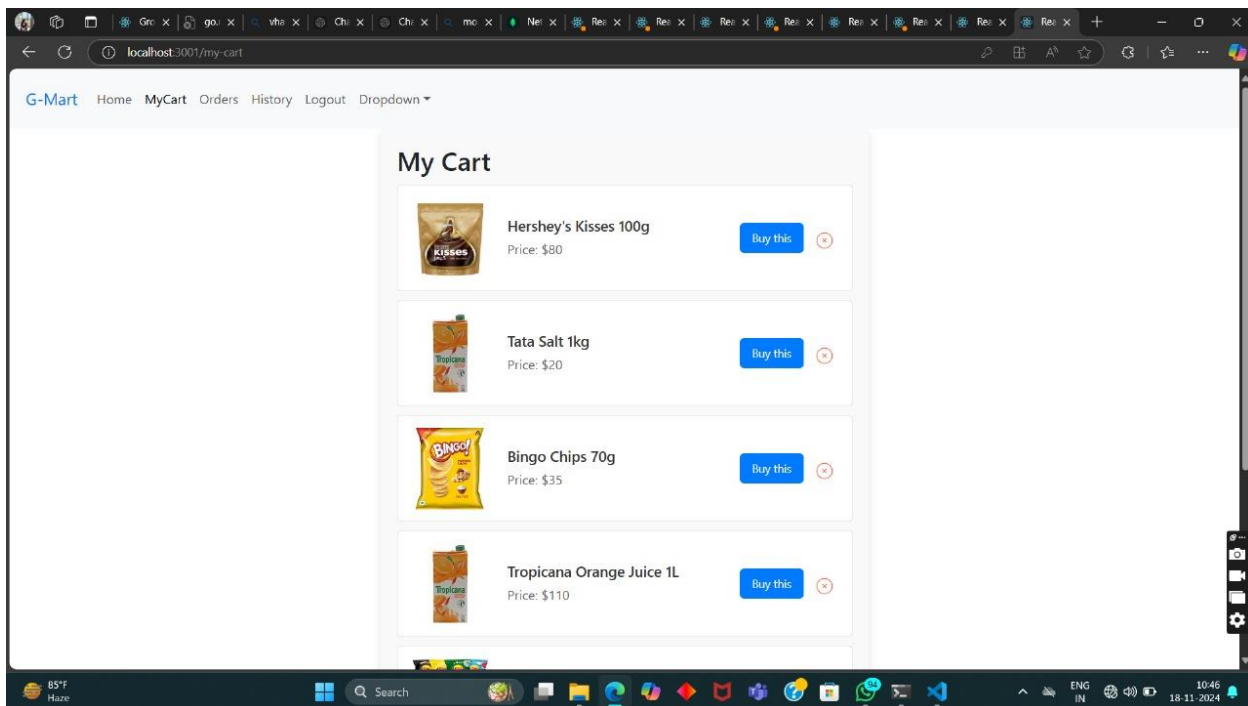


### 4. User Dashboard:

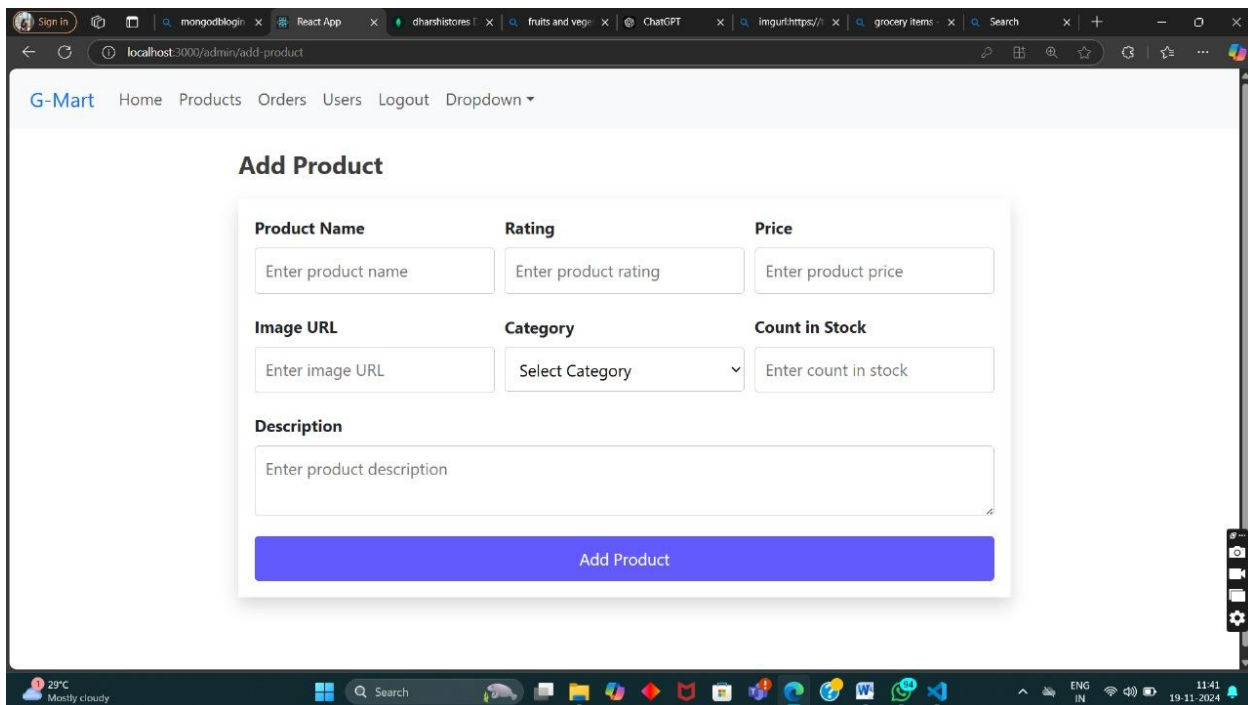




## 5.Cart Details:



## 6.Add Product Dashboard:



## 7.Add Category:

The screenshot shows a web browser window with the URL `localhost:3000/admin/add-category`. The page has a navigation bar with links: [G-Mart](#), [Home](#), [Products](#), [Orders](#), [Users](#), [Logout](#), and a [Dropdown](#) menu. The main content area is titled "Add Category" and contains a form with two input fields: "Category" (with placeholder text "Enter category") and "Description" (with placeholder text "Enter description"). Below these fields is a blue button labeled "Add Category". The browser's taskbar at the bottom shows the system clock as 11:41 on 19-11-2024.

## 8.Order Details:

The screenshot shows a web browser window with the URL `localhost:3000/my-orders`. The page has a navigation bar with links: [G-Mart](#), [Home](#), [MyCart](#), [Orders](#), [History](#), [Logout](#), and a [Dropdown](#) menu. The main content area is titled "My Orders" and displays a list of three orders. Each order card contains the following details:

- Order ID:** 673acb7f18e44b834d1578a9
- Name:** pooja J
- Phone:** 06382622547
- Date:** 2024-11-18T05:07:11.727Z
- Price:** 35
- Status:** Pending
- Payment Method:** cod

The second order card shows:

- Order ID:** 673acd2b2655f3ca930bd3f
- Name:** pooja J
- Phone:** 06382622547
- Date:** 2024-11-18T05:17:38.548Z
- Price:** 175
- Status:** Pending
- Payment Method:** debit

The third order card shows:

- Order ID:** 673ace15b2655f3ca930bd48
- Name:** dharshini priya N
- Phone:** 1234567890
- Date:** 2024-11-18T05:18:13.764Z
- Price:** 240
- Status:** Pending
- Payment Method:** cod

The browser's taskbar at the bottom shows the system clock as 10:48 on 18-11-2024.