Grocery web App

Team ID	LTVIP2025TMID53180
Project Name	ShopSmart: Your Digital Grocery Store Experience

Introduction:

"Welcome to our Grocery Web App – your ultimate destination for hassle-free grocery shopping! We bring you a smooth and easy-to-use platform packed with top-quality products. From fresh fruits and vegetables to everyday essentials, everything you need is just a few clicks away. Browse our digital shelves, fill your cart effortlessly, and get your groceries delivered straight to your home. Say goodbye to long queues and hello to convenient shopping – all from the comfort of your couch!"

Description:

Welcome to our Grocery Web App – your personalized, powerful, and future-ready online grocery solution.

We've created more than just a shopping app – we've built a complete experience that brings the supermarket to your screen. Whether you're a busy professional, a student, a homemaker, or someone who simply values convenience, our platform is designed to serve every lifestyle and every household.

All-in-One Grocery Destination:

From locally-sourced fresh fruits and vegetables to international brands and everyday kitchen staples, our app offers a huge range of categories to explore. Need personal care products? Cleaning supplies? Baby items? It's all here – neatly categorized and easy to find.

Smart & Seamless Navigation:

Our intuitive design ensures a smooth and stress-free shopping experience. Use smart search, filters, and sorting options to find exactly what you need. No need to wander through aisles—your favourite products are now just a tap away.

Personalized Experience:

Get recommendations based on your shopping habits, save your favourite items for quick reordering, and enjoy customized offers and promotions. Our app adapts to your preferences, helping you save both time and money.

Flexible Delivery Options:

We understand your time is valuable. Schedule your delivery for a time that suits you best—even same-day delivery in selected areas! Track your order in real-time, and get updates so you're always in the loop.

Diverse Product Range for Every Need:

We cater to all dietary lifestyles—whether you're looking for gluten-free, vegan, low-sugar, keto-friendly, or organic products, we've got you covered. We believe grocery shopping should be inclusive, thoughtful, and empowering.

Secure Payments & Easy Checkout:

Choose from multiple secure payment methods, including UPI, credit/debit cards, net banking, and cash on delivery. Our one-click checkout process makes ordering fast and reliable.

Stay Updated with Notifications:

Receive alerts about new arrivals, restocked items, ongoing sales, and limited-time deals. Never miss an offer or run out of your everyday essentials again.

Customer Support That Cares:

Our dedicated support team is available via chat, email, and phone to help you with queries, returns, and feedback. We're here to ensure your shopping journey is always smooth.

Environmentally Friendly Packaging:

We care about the planet as much as you do. Our packaging is eco-conscious, and we're constantly working on reducing our carbon footprint.

Why choose our Grocery Web App?

✅ Convenience at your fingertips

✅ High-quality, trusted products

✅ Smart technology for easy browsing

✅ Reliable, on-time doorstep delivery

✅ Personalized for your lifestyle

✅ Secure and fast checkout

✅ Excellent customer service

Step into the future of shopping – download our Grocery Web App today and experience freshness, flexibility, and freedom, all from the comfort of your home. Let's make everyday grocery shopping smarter together!

Scenario Based Case Study:

Meet Ramesh, a caring father and devoted homemaker who manages household responsibilities and takes care of his two young children.

Ramesh's day is packed with tasks—from school runs to meal prep to household chores. He prefers to shop for groceries online to avoid crowded markets and save time. However, he often struggles to find everything he needs in one place and dislikes complicated online shopping experiences.

Ramesh's Solution: The Grocery Web App

When Ramesh discovers the Grocery Web App, his shopping experience transforms completely. The app becomes his go-to platform for convenient, stress-free grocery management.

User Registration and Profile Setup:

Ramesh quickly signs up using his email and mobile number. He sets preferences such as dietary requirements (low sugar, child-friendly products), and favorite brands.

Easy-to-Navigate Categories:

Ramesh finds the user interface simple and clean. He explores categories like Baby Care, Snacks, Cleaning Essentials, and Vegetables—all laid out clearly. The smart search helps him find items like “mild detergent” or “sugar-free biscuits” in seconds.

Shopping List Feature:

With two toddlers to care for, Ramesh often forgets items. The app’s built-in Shopping List feature allows him to create and save weekly lists, which he can reuse or update as needed.

Repeat Orders & Quick Reorder:

Ramesh loves the quick reorder option. With one tap, he can reorder his last grocery list—no need to search every time.

Flexible Delivery Scheduling:

Ramesh chooses delivery times that fit between his kids’ nap times and chores. The app even sends a reminder before delivery time to help him plan.

Multiple Payment Options & Wallets:

He pays securely through UPI or cash on delivery depending on convenience. The app stores his payment preferences safely for faster checkout.

Real-Time Notifications:

Ramesh receives alerts about offers on baby products, seasonal fruits, and cleaning supplies. He never misses a deal!

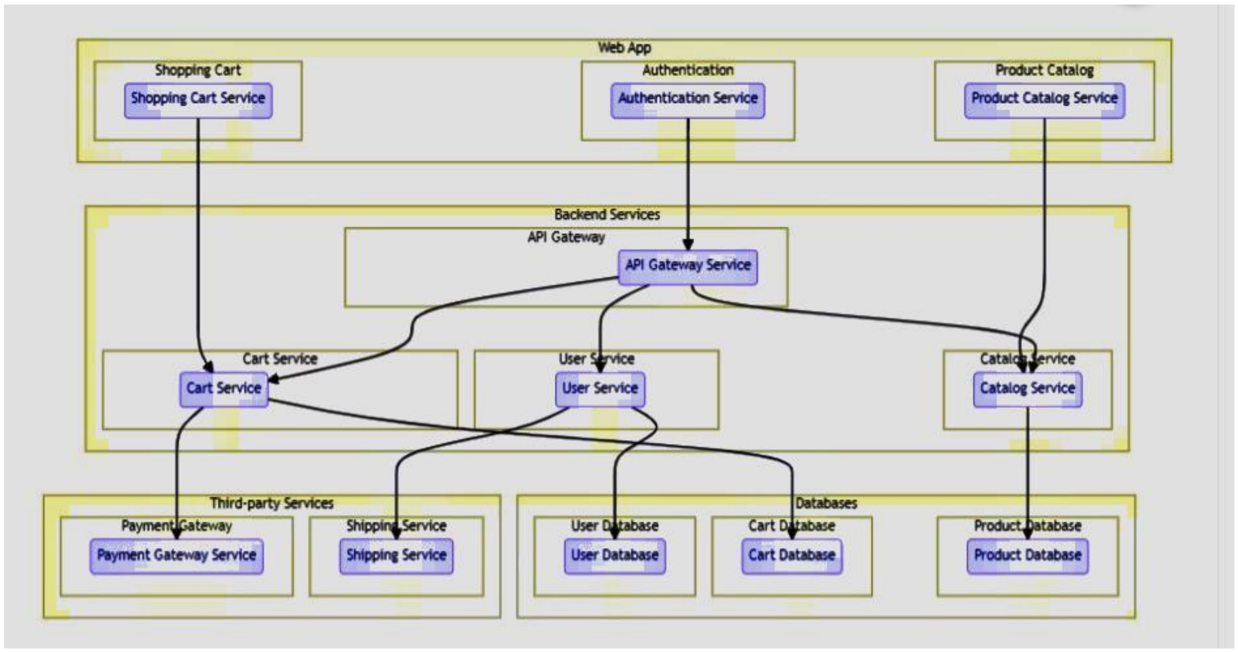
Live Chat & Friendly Customer Service:

If he has any issues—like missing items or payment problems—Ramesh uses the in-app live chat. The support team responds promptly and resolves his concerns quickly.

Ramesh’s Experience:

With the Grocery Web App, Ramesh no longer feels overwhelmed with weekly grocery shopping. He saves time, avoids last-minute store runs, and keeps his home well-stocked—all while taking great care of his family.

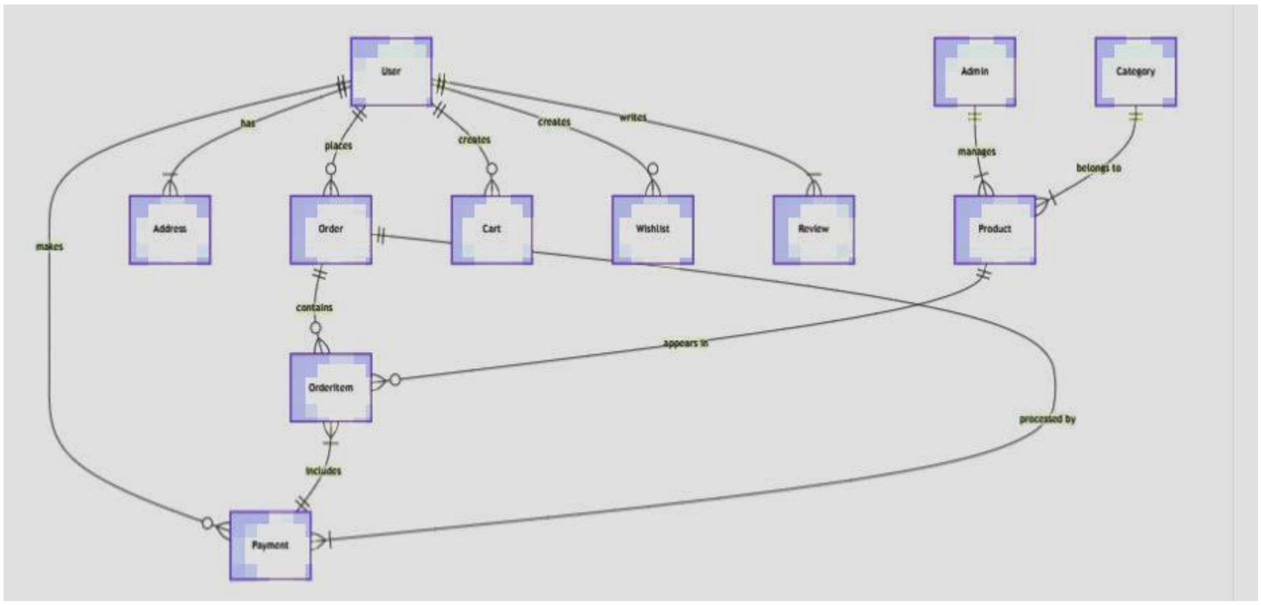
Technical Architecture:-



The technical structure of a grocery web application usually follows a client-server architecture. In this setup, the frontend acts as the client, handling everything the user sees and interacts with—such as the layout, buttons, product listings, and shopping cart. On the other hand, the backend works as the server, managing tasks like storing data, running business logic, processing payments, and communicating with databases.

The frontend and backend connect and communicate using APIs (Application Programming Interfaces). These APIs allow the two parts to share data smoothly, making the app work efficiently as users browse, add items to their cart, make payments, or track orders.

ER-Diagram:



The technical architecture of a grocery web application usually follows a client-server model. In this model:

The frontend (client-side) is what the user interacts with. It handles the user interface (UI), user actions like clicking buttons, and how the content is displayed on the screen.

The backend (server-side) works behind the scenes. It manages data storage, processes business logic (like calculating total bills, handling user logins, etc.), and connects with external services such as payment gateways and databases.

These two parts—frontend and backend—communicate with each other using APIs (Application Programming Interfaces). APIs help send and receive data between them smoothly, so the app runs efficiently and responds to user actions in real time.

Key Features:

Product Catalog

Our Grocery Web App offers a comprehensive product catalog organized into various categories and subcategories. Users can effortlessly browse, search, and filter products based on preferences like price, brand, or ratings—making it easy to find exactly what they need.

Shopping Cart & Checkout

<u>The app features a smart shopping cart that allows users to:</u>

- Add and remove products
- Review selected items
- Proceed to checkout smoothly

The checkout process supports multiple secure payment methods, including credit/debit cards, UPI, wallets, and cash on delivery, ensuring a safe and convenient transaction experience.

Product Reviews & Ratings

- Users can rate and review products based on their experience. <u>This feature:</u>
- Helps other shoppers make informed choices
- Builds trust and transparency
- Fosters a community-driven shopping environment

Order Tracking

<u>After placing an order, users can:</u>

- Track their orders in real-time
- Receive regular updates on order status (processing, shipped, delivered) This ensures transparency, reliability, and peace of mind throughout the delivery process.

Order Management

- The app efficiently handles the entire order lifecycle, including:
- Placing new orders
- Viewing order history
- Tracking shipments
- Managing returns and cancellations
- This gives users full control over their past and current purchases.

Search & Filtering

Our app includes powerful search and filtering tools, enabling users to:

- Search products by name or keyword
- Apply filters like price range, brand, and customer ratings This feature improves shopping speed and accuracy, enhancing the overall user experience.

Admin Dashboard

For administrators, the app provides a robust Admin Dashboard with the ability to:

- Manage products, inventory, and orders
- Access customer profiles and feedback
- Monitor sales performance, stock levels, and customer behavior This helps in making informed decisions and running operations efficiently.

Order Management

- The app helps users manage their orders from start to finish.
- You can place orders easily.
- Track your order status (like packed, shipped, or delivered).
- See your past orders in one place.
- If needed, you can return or cancel an order without any trouble.

Search & Filtering

- Finding products is very simple with our app.
- Just type a keyword (like "milk" or "rice") in the search bar.
- Use filters to narrow your search—for example, by price, brand, or customer ratings.
- This helps you quickly find the exact item you're looking for.

Pre-Requisites:

To build a modern, full-stack e-commerce application using React.js, Node.js, Express.js, and MongoDB, the following tools and technologies must be set up:

1️⃣ Node.js and npm

Purpose: Enables server-side JavaScript execution and manages project dependencies.

Download: Node.js Official Website

Installation Guide: Install Using Package Manager

2️⃣ MongoDB

Purpose: Acts as the NoSQL database for storing data like products, users, and orders.

Download MongoDB Community Edition: MongoDB Download

Installation Guide: MongoDB Manual

Note: You can also use MongoDB Atlas (Cloud) instead of local setup.

3️⃣ Express.js

Purpose: A lightweight web framework for Node.js, used to create server APIs and manage routing.

Installation Command: npm install express

4️⃣ React.js

Purpose: A JavaScript library used to create interactive, component-based user interfaces.

Recommended Setup: Use Vite for faster build and development.

Quick Start with Vite:

```
npm create vite@latest

cd my-app

npm install

npm run dev
```

Open your browser at: http://localhost:5173

React Official Documentation: https://react.dev/

If you previously installed create-react-app globally, remove it:

```
npm uninstall -g create-react-app
```

5️⃣ HTML, CSS, and JavaScript

Purpose: Basic front-end technologies to build structure, style, and interactivity.

HTML: For structuring pages

CSS: For styling the user interface

JavaScript: For adding client-side logic and dynamic behavior

6️⃣ Database Connectivity (Mongoose)

Purpose: Mongoose is used to connect the Node.js backend with MongoDB and perform CRUD operations.

Install Mongoose: npm install mongoose

7️⃣ Front-End Interface (React)

Usage:

- Display product listings
- Build user dashboards and forms
- Create admin panels for managing inventory and orders

8️⃣ Version Control System (Git & GitHub)

Purpose: Track code changes, collaborate with others, and maintain code history.

Download Git: Git SCM

Use platforms like GitHub or Bitbucket for remote repositories

9️⃣ Code Editor / IDE

Recommended Development Tools:

- Visual Studio Code
- Sublime Text
- Web Storm

Choose an editor that best suits your coding workflow and supports JavaScript development.

Roles and Responsibilities:

<u>**User Responsibilities**</u>

Users interact with the application to perform shopping-related activities. Their responsibilities include:

1. Registration & Authentication
   - Create a personal account on the platform.
   - Log in securely using valid credentials to access the app’s features.
2. Browsing & Shopping
   - Browse the product catalog by category or search.
   - Add selected products to the shopping cart.
   - Proceed to checkout to place orders.
3. Payment

- Choose a preferred payment method and complete the purchase.
- Ensure payment is made securely and correctly.

4. Order Management
   - View order history and details.
   - Track the delivery status of active orders.
   - Update account information as needed.
5. Feedback & Reviews
   - Provide ratings and write reviews on purchased products.
   - Share feedback to help improve the platform and assist other users.
6. Compliance
   - Follow the app's terms & conditions and privacy policy.
   - Use the app responsibly and ethically.

Admin Responsibilities

Admins manage and maintain the platform to ensure smooth operation. Their responsibilities include:

1. User Management
   - Create, update, or delete user accounts when required.
   - Monitor user activities to ensure policy compliance.
2. Product Management
   - Add new products to the catalog.
   - Edit existing product details (price, availability, images).
   - Remove outdated or inactive products.
3. Order Management
   - Oversee all orders placed on the platform.
   - Manage payments, shipping updates, returns, and refunds.
4. Content Management
   - Maintain and update informational content across the app.
   - Manage blogs, FAQs, terms of service, and promotional content.
5. Analytics & Reporting
   - Generate performance reports on user activity, sales, and inventory.
   - Analyze data to improve platform efficiency and customer satisfaction.
6. Compliance & Security
   - Ensure all operations follow legal and data protection regulations.
   - Protect user data and prevent unauthorized access.
7. Customer Support

- Address user queries, complaints, and technical issues.
- Provide timely and helpful customer service.

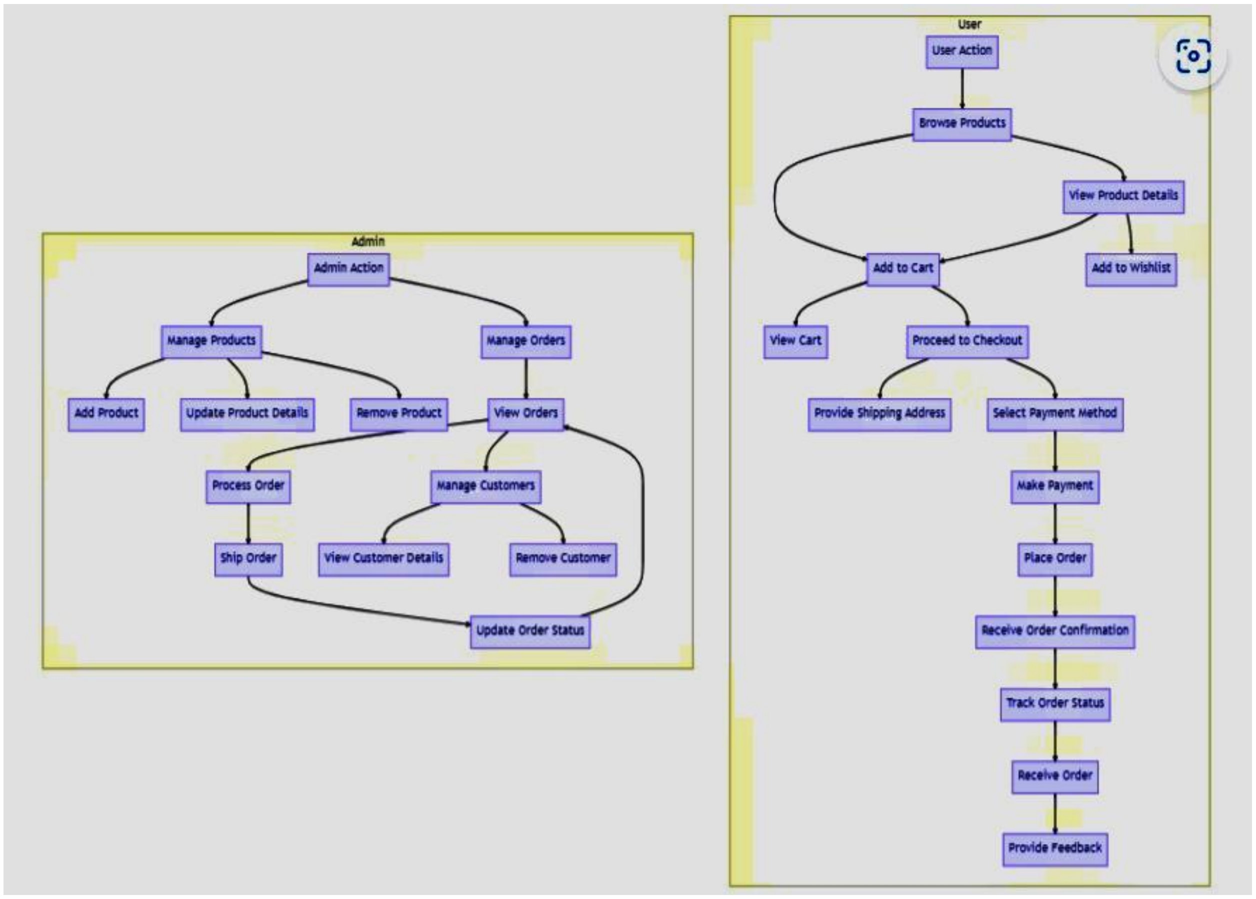
8. Marketing & Promotions
    - Plan and run campaigns, discounts, and special offers.
    - Monitor the impact of promotional activities to boost user engagement.

Admin & User Flow:

The Grocery Web App works in two parts: one for users (customers) and one for admins (managers). Both have different roles and actions in the app.

For users, the journey starts with creating an account or logging in. After that, they can look through different categories of products, search for items, and apply filters to find what they need easily. They can add items to their cart and, once done shopping, go to the checkout page. There, users enter their delivery address and choose how they want to pay (like using UPI, debit/credit card, or cash on delivery). After payment, they get a confirmation message and tracking details to know where their order is. Users can also check past orders and write reviews about the products they bought.

For admins, the app provides a dashboard to manage everything. Admins can add new products, update product details (like price or stock), and remove items that are no longer available. They can see all the orders placed by users and update the status (like packed, shipped, or delivered). Admins also manage user accounts, help with customer questions, and handle returns or cancellations. They can view reports to check how well the app is doing, monitor sales, and run offers or promotions. Admins also make sure the app is safe and follows all rules properly.



Project flow:

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

Demo link:-

https://drive.google.com/file/d/1wvMsw6MyRXnmKIZxHoYuBt4WJ6ndFj0Y/view?usp=sharing

Use the code in:-

https://drive.google.com/drive/folders/1RP-29p9mf-bbLAK5r7S4HSF_Itai0i1G?usp=drive_link

or follow the videos below for better understanding.

Milestone 1: Project Setup and Configuration:

1. **Install required tools and software:**
   - Node.js.
   - MongoDB.
   - Create-react-app.
2. **Create project folders and files:**
   - Client folders.
   - Server folders.
3. **Install Packages:**
   - Frontend npm Packages
   - Axios.
   - React-Router –dom.
   - Bootstrap.
   - React-Bootstrap.
   - React-icons.
   - Backend npm Packages
   - Express.
   - Mongoose.
   - Cors.

Reference Link:-

https://drive.google.com/file/d/1Acv3Lx3PtJcOYkUjREWAzIoC-i6w96Tl/view?usp=drive_link

Milestone 2: Backend Development:

1. **Setup express server**
   - Create index.js file in the server (backend folder).
   - Create a .env file and define port number to access it globally.
   - Configure the server by adding cors, body-parser.
2. **User Authentication:**
   - Create routes and middleware for user registration, login, and logout.
   - Set up authentication middleware to protect routes that require user authentication.
3. **Define API Routes:**
   - Create separate route files for different API functionalities such as users orders, and authentication.

- Define the necessary routes for listing products, handling user registration and login, managing orders, etc.
- Implement route handlers using Express.js to handle requests and interact with the database.

4. **Implement Data Models:**
   - Define Mongoose schemas for the different data entities like products, users, and orders.
   - Create corresponding Mongoose models to interact with the MongoDB database.
   - Implement CRUD operations (Create, Read, Update, Delete) for each model to perform database operations.
5. **User Authentication:**
   - Create routes and middleware for user registration, login, and logout.
   - Set up authentication middleware to protect routes that require user authentication.
6. **Error Handling:**

   • Implement error handling middleware to catch and handle any errors that occur during the API requests.

   • Return appropriate error responses with relevant error messages and HTTP status codes.

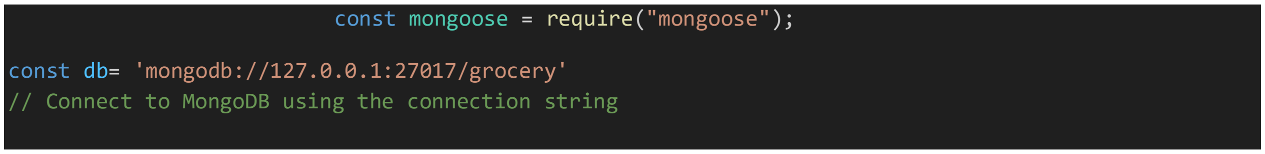
Reference Link:-

https://drive.google.com/file/d/18-gnCVAZdojBhl7QM0_yShH1JL5A6v32/view?usp=drive_link

Milestone 3: Database:

1. **Configure MongoDB:**
   - Install Mongoose.
   - Create database connection.
   - Create Schemas & Models.
2. **Connect database to backend:**

   Now, make sure the database is connected before performing any of the actions through the backend. The connection code looks similar to the one provided below.



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

const db= 'mongodb://127.0.0.1:27017/grocery'
// Connect to MongoDB using the connection string
```

```
mongoose.connect(db, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
}).then(() => {
  console.log(`Connection successful`);
}).catch((e) => {
  console.log(`No connection: ${e}`);
});
```

3. Configure Schema:

Firstly, configure the Schemas for MongoDB database, to store the data in such pattern. Use the data from the ER diagrams to create the schemas.

The schemas are looks like for the Application.

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


const userSchema = new mongoose.Schema({
    firstname: { type: String},
    lastname: { type: String },
    username: { type: String, unique: true },
    email: { type: String},
    password: { type: String }
});

// category schema
const categorySchema = new mongoose.Schema({
    category: { type: String, required: true, unique: true, },
    description: { type: String, }
});

const productSchema = new mongoose.Schema({
    productname: { type: String, required: true },
    description: { type: String, required: true },
    price: { type: Number, required: true },
    image: { type: String, required: true },
    category: { type: String, ref: 'Category', required: true },
    countInStock: { type: Number, required: true, min: 0 },
    rating: { type: Number, required: true },
    dateCreated: { type: Date, default: Date.now }
});

const addToCartSchema = new mongoose.Schema({
    userId: { type: String, required: true },
    productId: { type: String, required: true },
    quantity: { type: Number, minimum: 1, required: true, default: 1 },
});

const orderSchema = new mongoose.Schema({
    firstname: { type: String, required: true },
    lastname: { type: String, required: true },
    user: { type: String, ref: 'User', required: true },
    phone: { type: String, required: true },
    productId: { type: String, required: true },
    productName: { type: String, required: true },
    quantity: { type: String, default: 1 },
    price: { type: String, required: true },
    status: { type: String, enum: ['Pending', 'Confirmed', 'Shipped', 'Delivered', 'Canceled',],
default: 'Pending' },
    paymentMethod: { type: String, required: true },
    address: { type: String, required: true },
    createdAt: { type: Date, default: Date.now }
});

const models = {
    Users: mongoose.model('User', userSchema),
    Category: mongoose.model('Category', categorySchema),
    Product: mongoose.model('Product', productSchema),
    AddToCart: mongoose.model('AddToCart', addToCartSchema),
    Order: mongoose.model('Order', orderSchema),

};

module.exports = models;
```

Milestone 4: Frontend Development:

1. **Setup React Application:**
    - Create React application.
    - Configure Routing.
    - Install required libraries.
2. **Design UI components:**
    - Create Components.
    - Implement layout and styling.
    - Add navigation.
3. **Implement frontend logic:**

    • Integration with API endpoints.

    • Implement data binding.

Reference:-

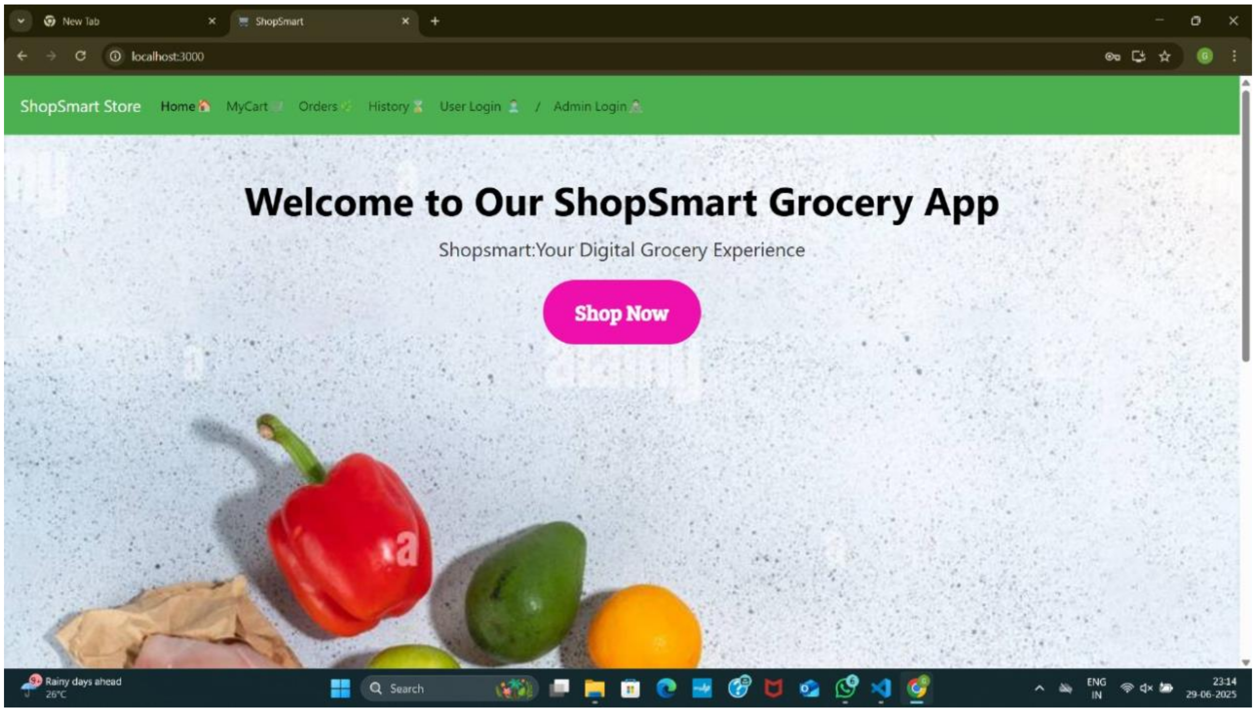
https://drive.google.com/file/d/1MFQ7NIQ02ynLioEMySPpEzdX-c0i6Wer/view?usp=drive_link

Milestone 5: Project Implementation:

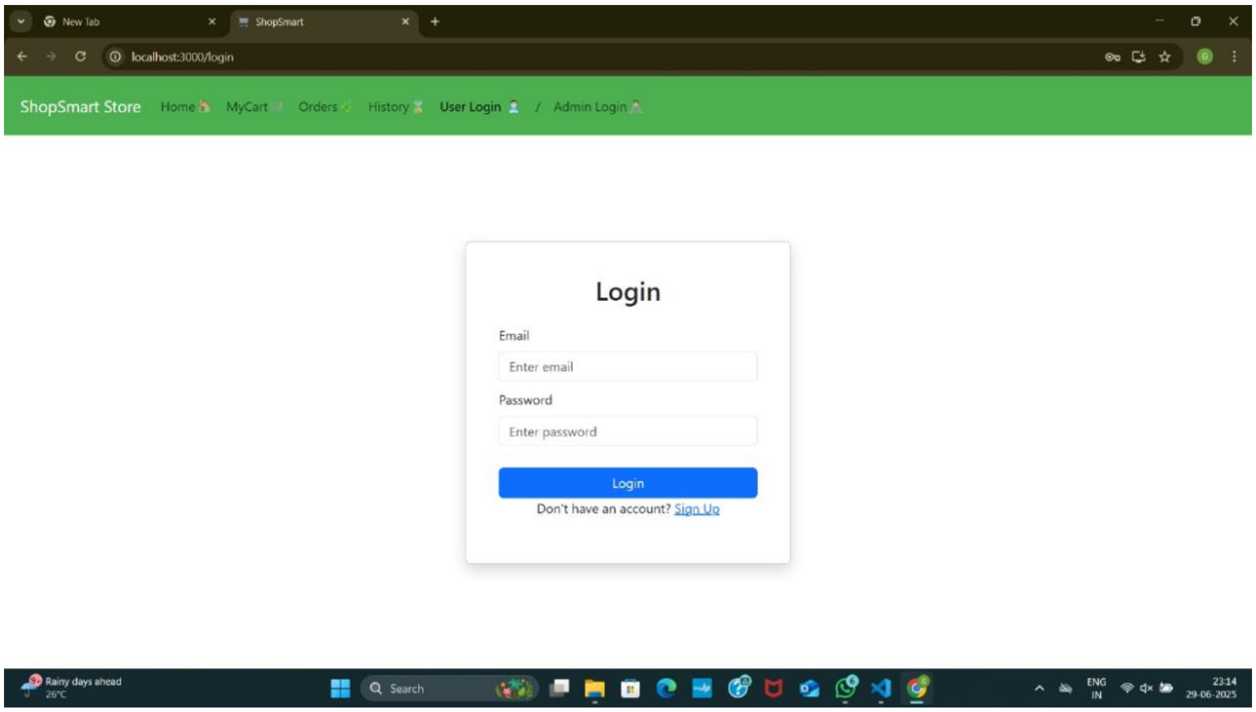
Finally, after completing the coding phase, we executed the entire project to test its functionality and identify any bugs or issues. This stage involved thorough testing of all features to ensure that each component works as expected and the system performs smoothly. Any detected errors were resolved to ensure a seamless user experience.

Now, let's take a final look at the fully functional Darshan Ease system and observe how it operates in real-time.

Landing page:-



Login Page:-



Login

Email

Enter email

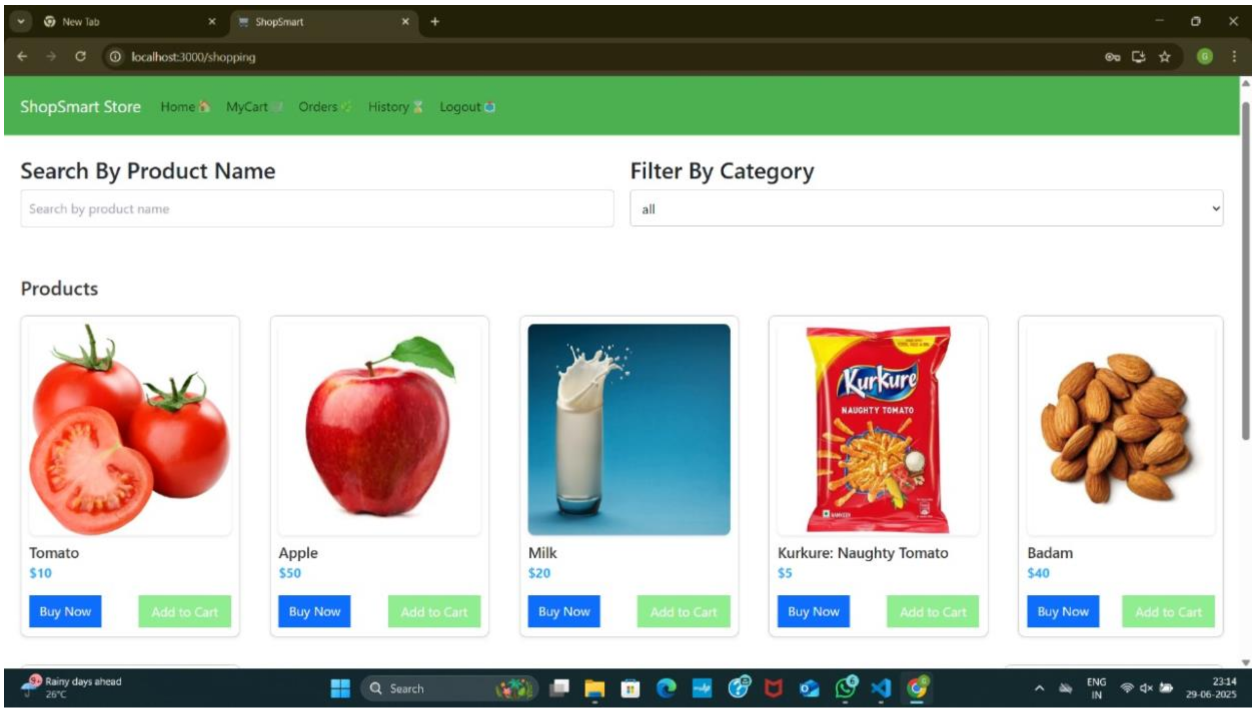
Password

Enter password

Login

Don't have an account? Sign Up

Items Page:-



Search By Product Name

Search by product name

Filter By Category

all

Products

Tomato
$10
Buy Now
Add to Cart

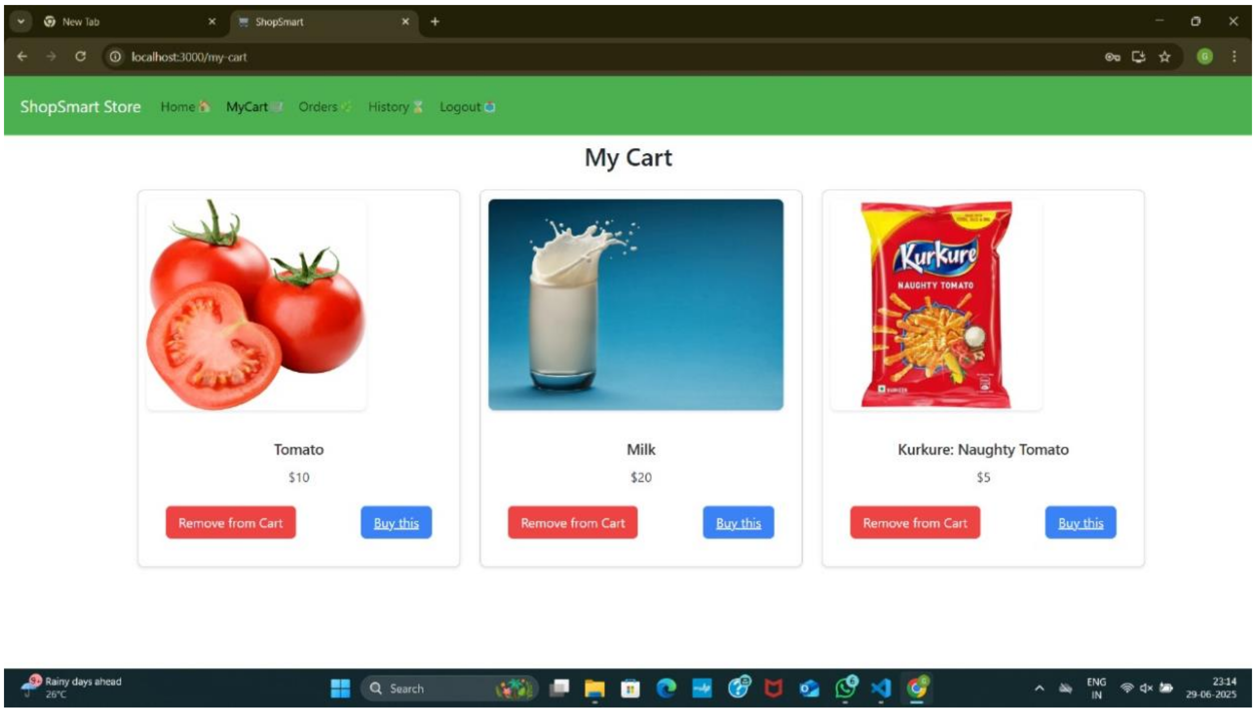
Apple
$50
Buy Now
Add to Cart

Milk
$20
Buy Now
Add to Cart

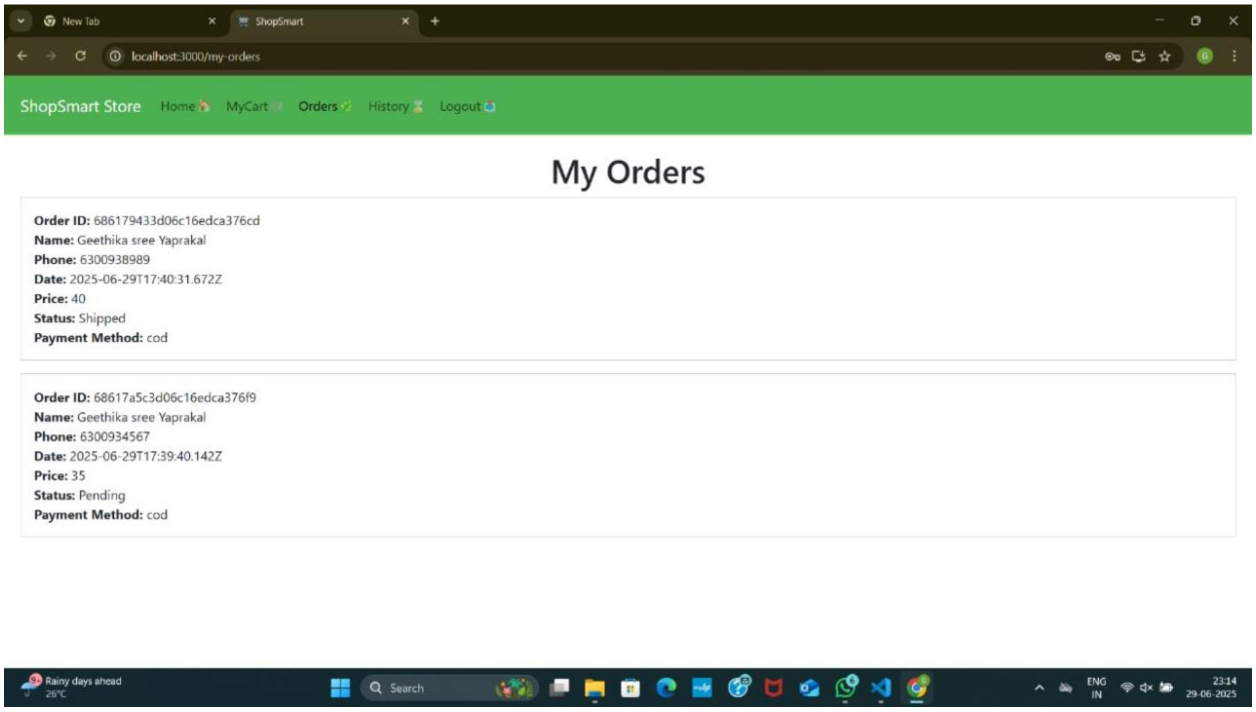
Kurkure: Naughty Tomato
$5
Buy Now
Add to Cart

Badam
$40
Buy Now
Add to Cart

My Cart:-



My Orders Page:-

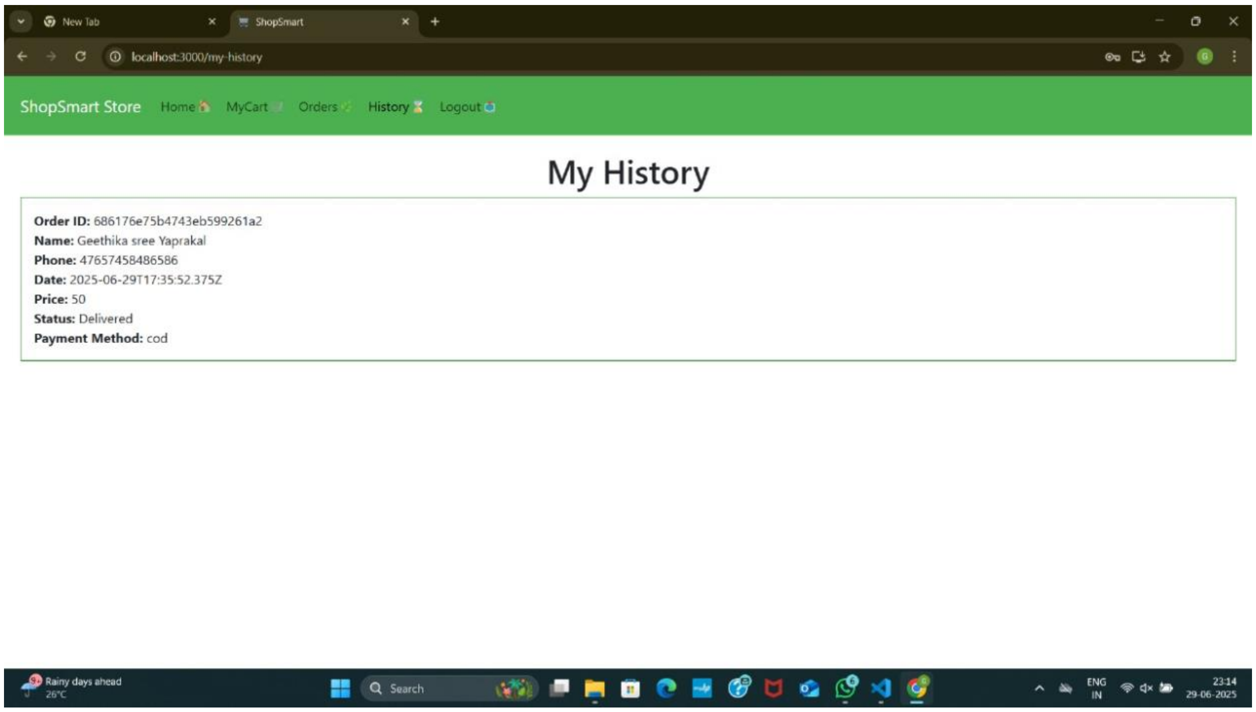


My Orders

Order ID: 686179433d06c16edca376cd
Name: Geethika sree Yaprakal
Phone: 6300938989
Date: 2025-06-29T17:40:31.672Z
Price: 40
Status: Shipped
Payment Method: cod

Order ID: 68617a5c3d06c16edca376f9
Name: Geethika sree Yaprakal
Phone: 6300934567
Date: 2025-06-29T17:39:40.142Z
Price: 35
Status: Pending
Payment Method: cod

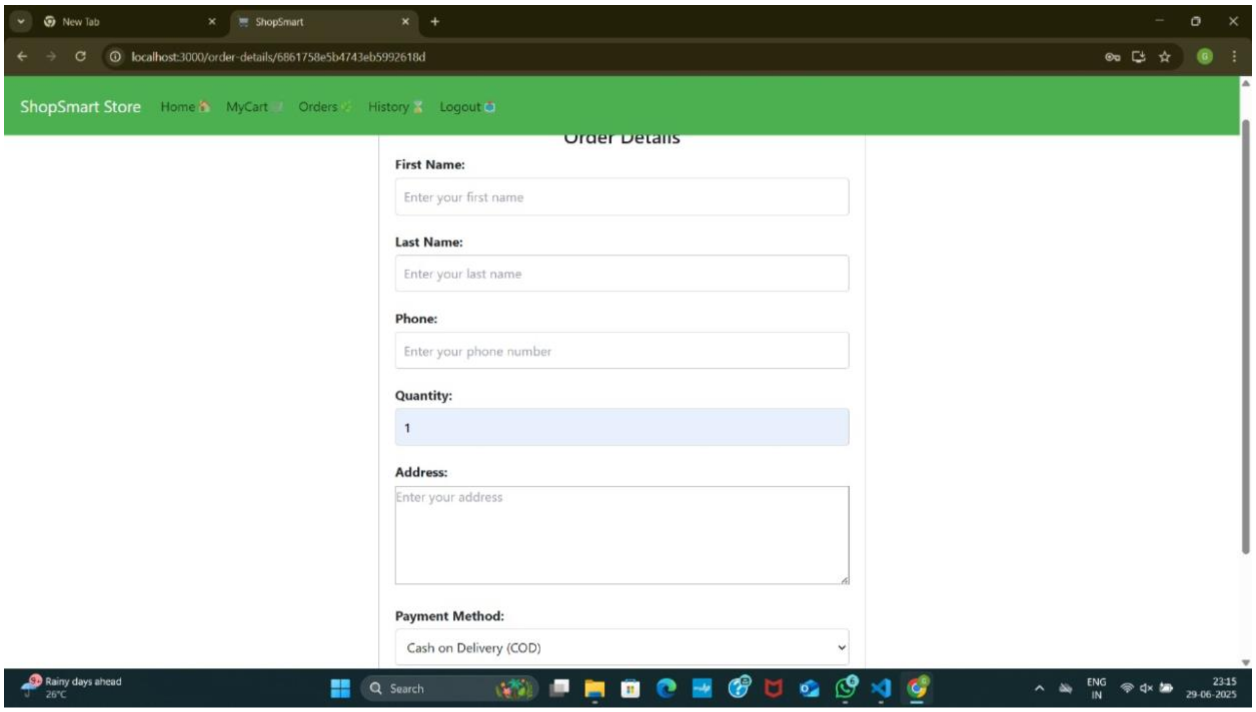
My History Page:-



My History

Order ID: 686176e75b4743eb599261a2
Name: Geethika sree Yaprakal
Phone: 47657458486586
Date: 2025-06-29T17:35:52.375Z
Price: 50
Status: Delivered
Payment Method: cod

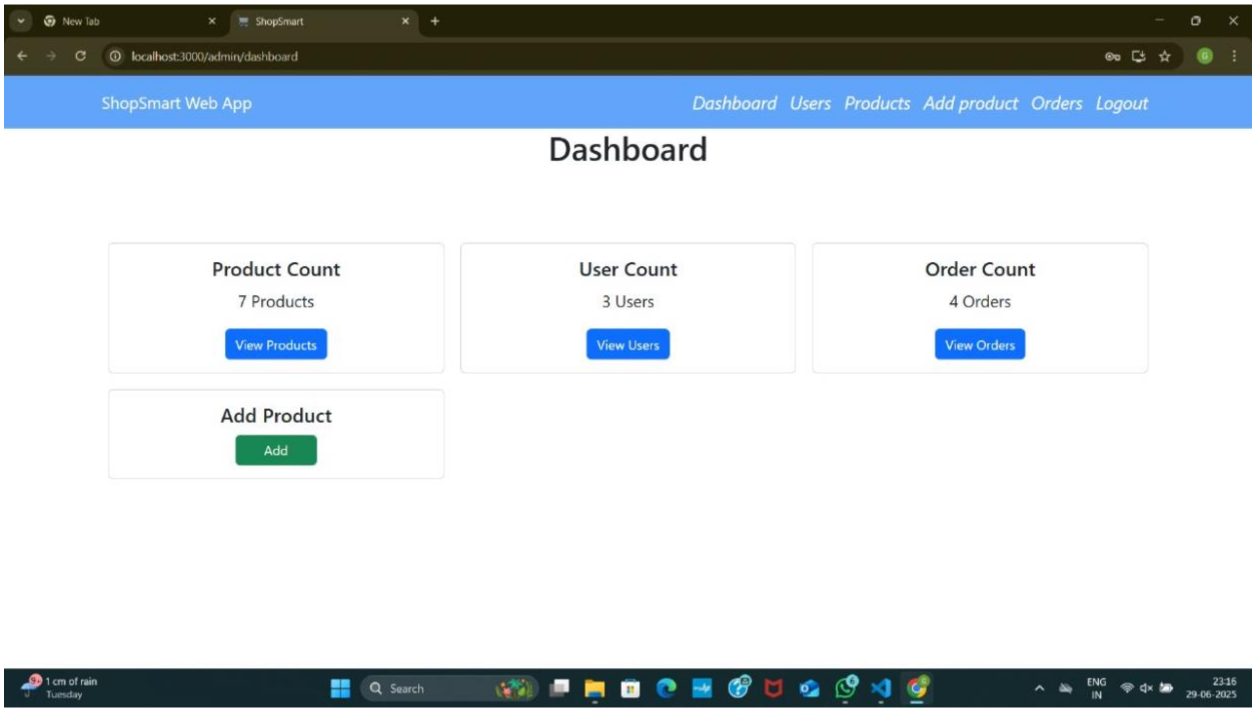
Place Order Page:-



Order Details

First Name: Enter your first name
Last Name: Enter your last name
Phone: Enter your phone number
Quantity: 1
Address: Enter your address
Payment Method: Cash on Delivery (COD)

Admin Dashboard Page:



Dashboard

Product Count

7 Products

User Count

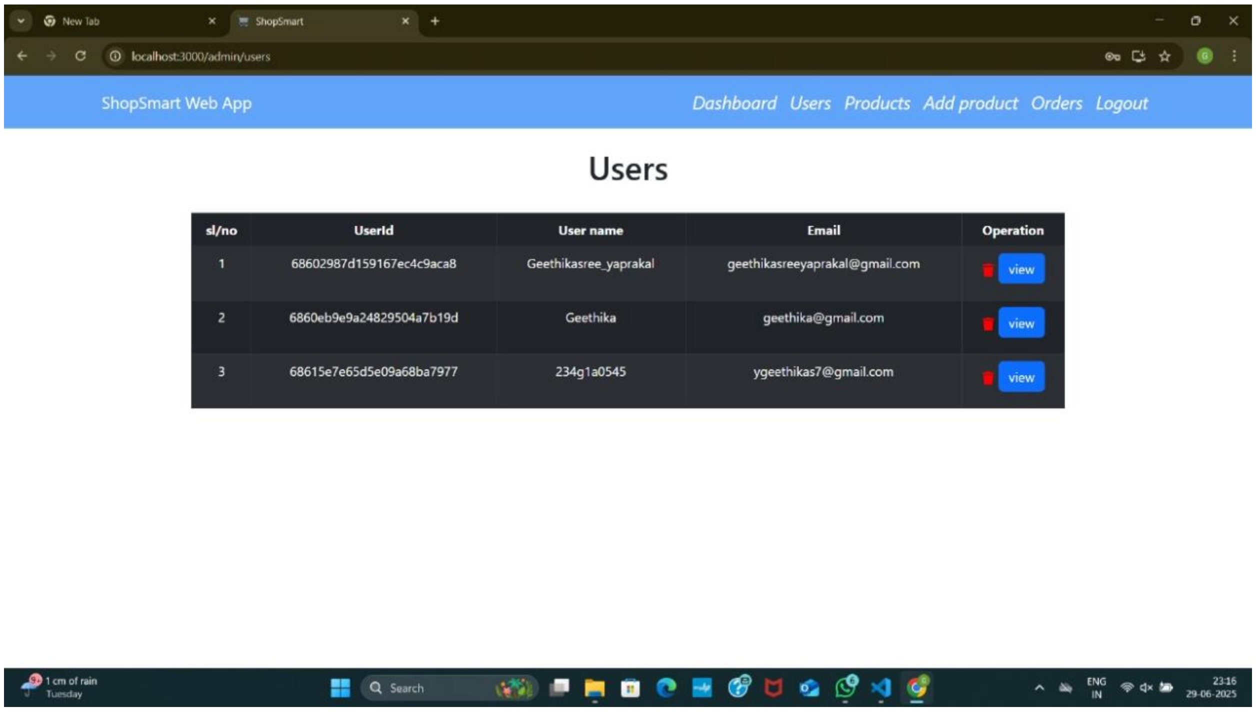
3 Users

Order Count

4 Orders

Add Product

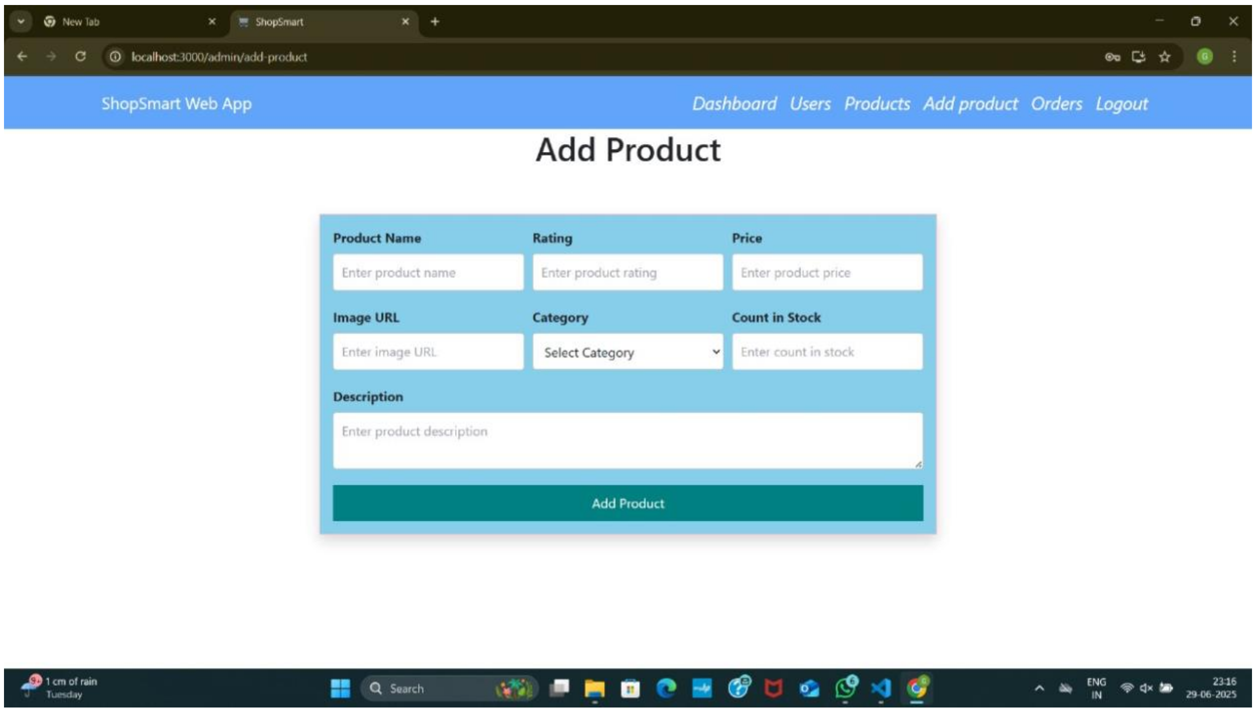
Users Page:-



Users

sl/no	UserId	User name	Email	Operation
1	68602987d159167ec4c9aca8	Geethikasree_yaprakal	geethikasreeyaprakal@gmail.com	view
2	6860eb9e9a24829504a7b19d	Geethika	geethika@gmail.com	view
3	68615e7e65d5e09a68ba7977	234g1a0545	ygeethikas7@gmail.com	view

Add Product page:-


localhost:3000/admin/add-product
ShopSmart Web App
Dashboard Users Products Add product Orders Logout

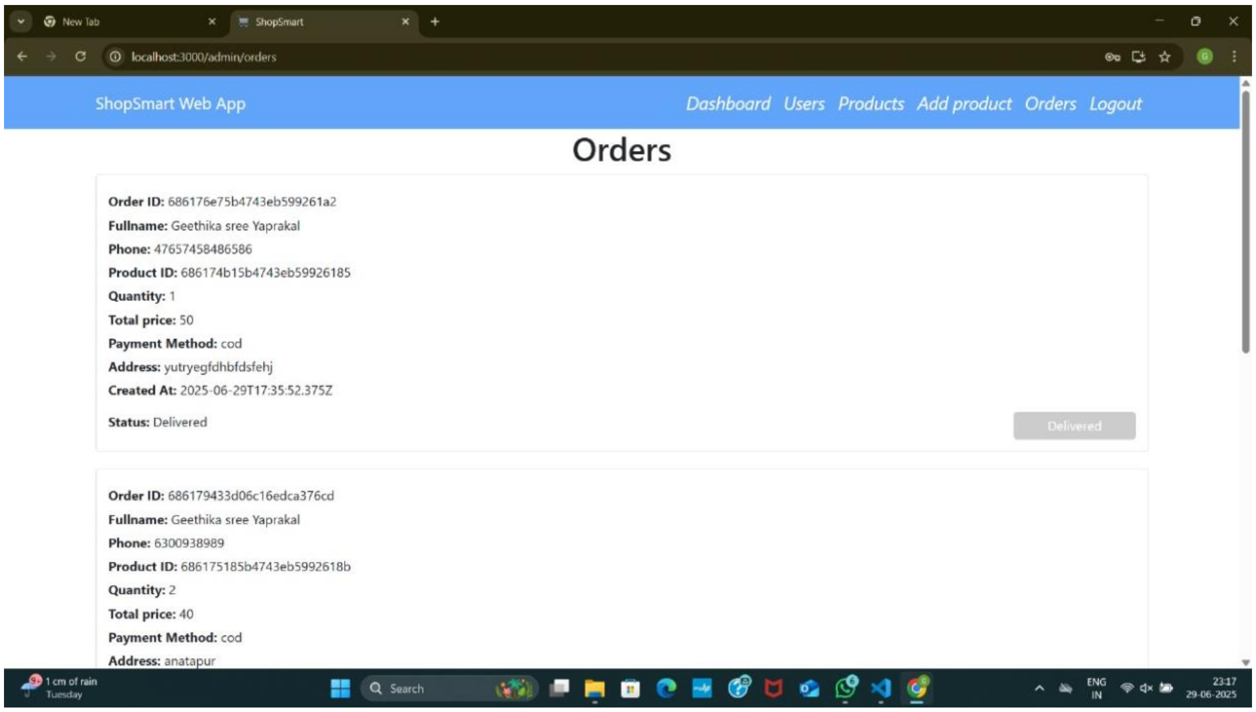
Add Product

Product Name	Rating	Price
Enter product name	Enter product rating	Enter product price
Image URL	Category	Count in Stock
Enter image URL	Select Category	Enter count in stock

Description

Enter product description

Admin Orders Page:-


localhost:3000/admin/orders
ShopSmart Web App
Dashboard Users Products Add product Orders Logout

Orders

Order ID: 686176e75b4743eb599261a2
Fullname: Geethika sree Yaprakal
Phone: 47657458486586
Product ID: 686174b15b4743eb59926185
Quantity: 1
Total price: 50
Payment Method: cod
Address: yutryegfdhbfdsfehj
Created At: 2025-06-29T17:35:52.375Z
Status: Delivered

Order ID: 686179433d06c16edca376cd
Fullname: Geethika sree Yaprakal
Phone: 6300938989
Product ID: 686175185b4743eb5992618b
Quantity: 2
Total price: 40
Payment Method: cod
Address: anatapur

The demo of the app is available at:-

https://drive.google.com/file/d/1wvMsw6MyRXnmKIZxHoYuBt4WJ6ndFj0Y/view?usp=shari
ng