Full Stack Development with MERN

Project Documentation: GreenCart – Online Grocery Web Application

1. Introduction

• Project Title: GreenCart – Online Grocery Web Application

• **Team Members**: Masineni Aswitha
Guddam Bhavana
Kanagala Anil Kumar
Kallamadi Anirudh
Lalbandh Babafakruddin

2. Project Overview

• Purpose: To provide users with a seamless online platform to browse, add, and purchase groceries.

• Features: Product listing, search/filter, cart, user authentication, admin product management.

3. Architecture

• Frontend: Built using Angular with Bootstrap for responsive design.

• Backend: Developed using Node.js and Express.js to handle business logic and APIs.

• Database: MongoDB is used for storing product, user, and cart information.

4. Setup Instructions

• Prerequisites: Node.js, MongoDB, Angular CLI.

• Installation:
1. Clone the repository.
2. Run `npm install` in both frontend and backend folders.
3. Setup environment variables.
4. Run the servers.

5. Folder Structure

• Client: Angular frontend organized into components, services, and routes.

• Server: Node.js backend with separate routes, controllers, and models.

6. Running the Application

• Frontend: `npm run dev` in the client directory.

• Backend: `npm run server` in the server directory.

7. API Documentation

• Document all endpoints exposed by the backend.
• Include request methods, parameters, and example responses.
Frontend: http://localhost:5173
Backend: http://localhost:4000

8. Authentication

• Authentication is handled using JWT (JSON Web Tokens).
• After login, a token is issued and stored in the browser (typically in localStorage).
• Protected backend routes require the token in the Authorization header.
• The server validates the token before processing the request.

9. User Interface

• Built with Angular and styled using Bootstrap for responsiveness.
• Clean design with sections for homepage, products, cart, and authentication.
• Built with Angular and styled using Bootstrap for responsiveness.
• Clean design with sections for homepage, products, cart, and authentication.
• Includes:

- Product grid with categories and prices
- Login/Signup pages
- Cart preview and checkout button

10. Testing

• Manual testing was performed for all workflows:

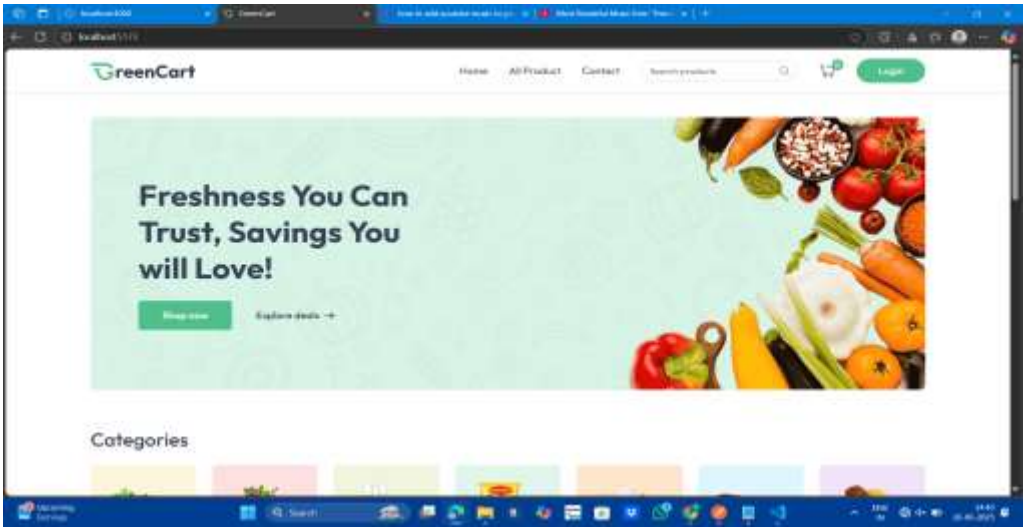
- User registration
- Login
- Adding/removing from cart
- Product navigation
  • Postman was used to validate backend API endpoints.
  • Unit testing can be enhanced using Jasmine and Karma (Angular default).

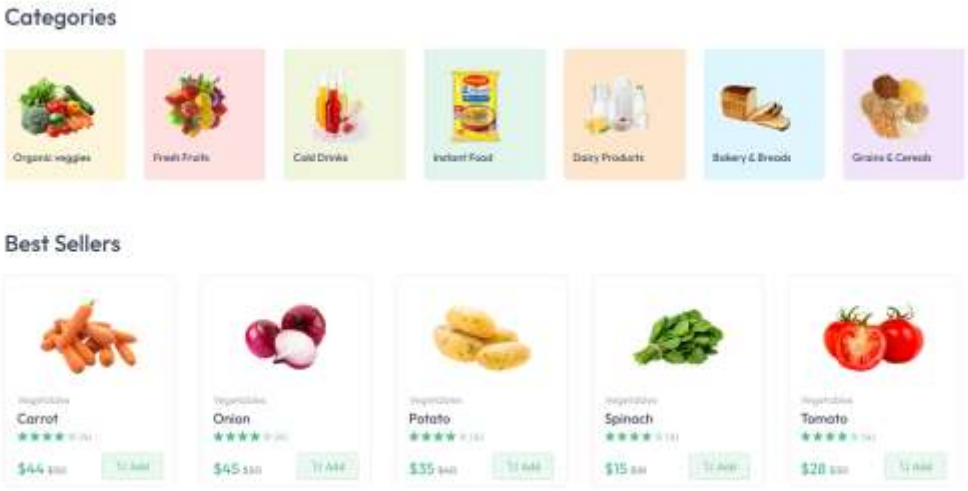
11. Screenshots or Demo

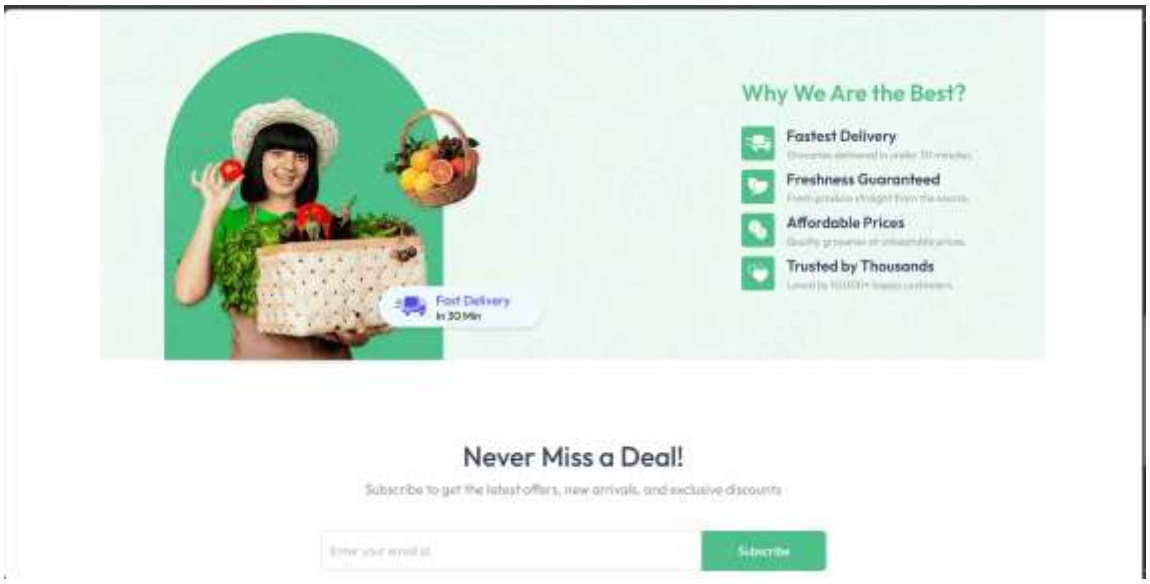
Demo Link:

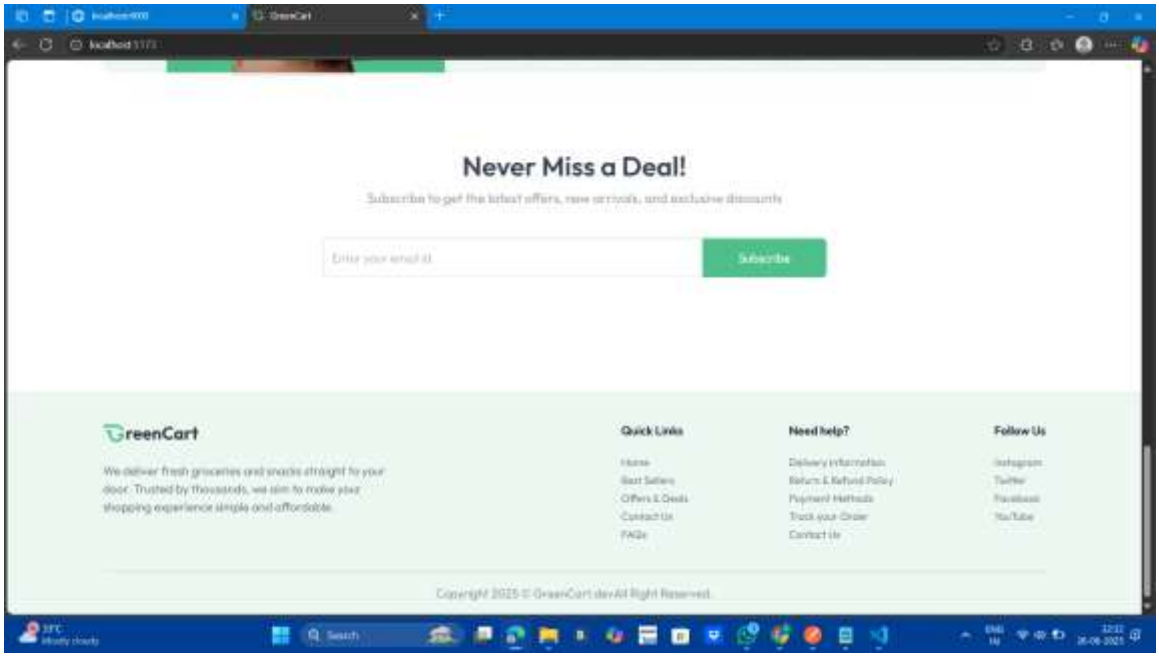
https://github.com/aswitha-06/greencart

Main page:

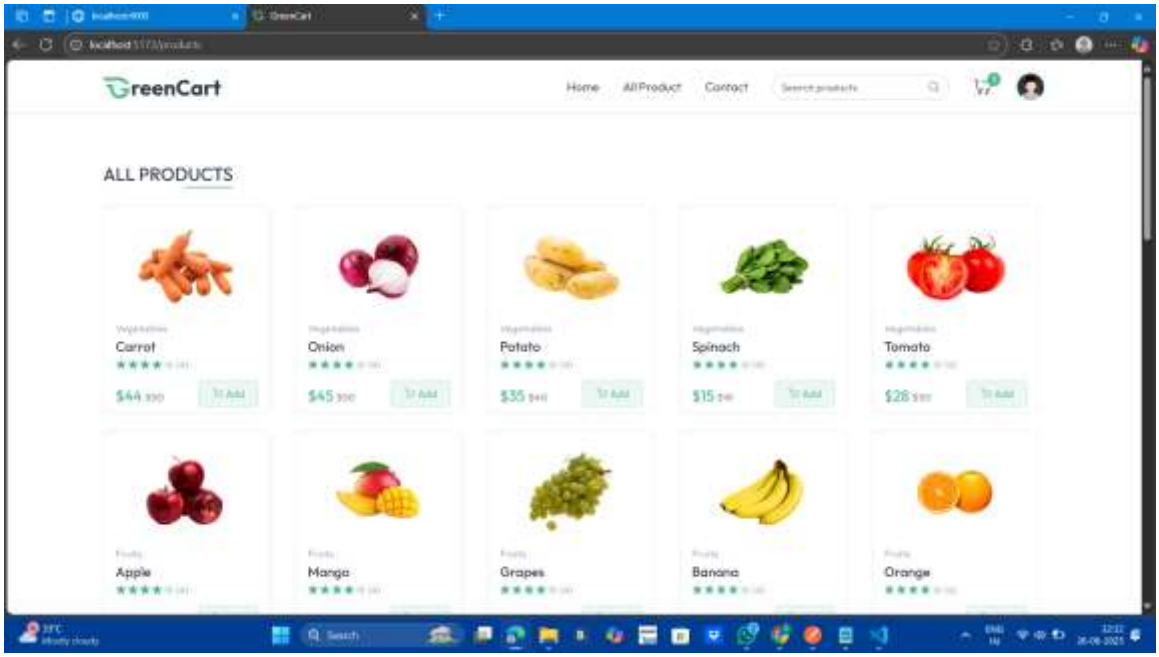

GreenCart
Home
All Product
Contact
Freshness You Can Trust, Savings You will Love!
Categories


Categories
Organic veggies
Fresh Fruits
Cold Drinks
Instant Food
Dairy Products
Bakery & Breads
Grains & Cereals
Best Sellers
Carrot
$44
Onion
$45
Potato
$35
Spinach
$15
Tomato
$28

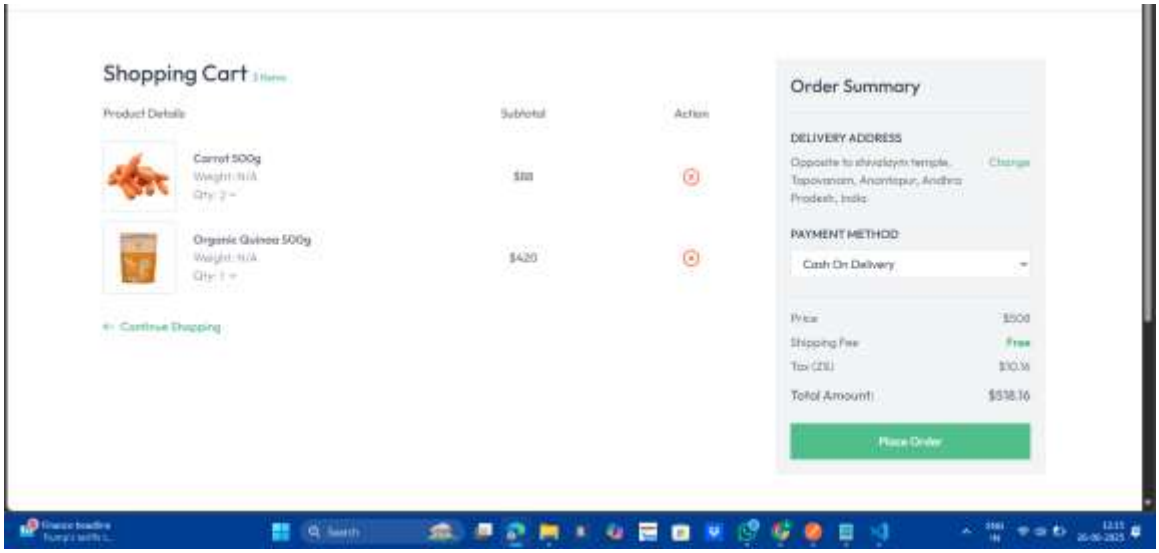

Why We Are the Best?
Fastest Delivery
Freshness Guaranteed
Affordable Prices
Trusted by Thousands
Fast Delivery
In 30 Min
Never Miss a Deal!
Subscribe to get the latest offers, new arrivals, and exclusive discounts
Subscribe


Never Miss a Deal!
Subscribe
Quick Links
Need help?
Follow Us

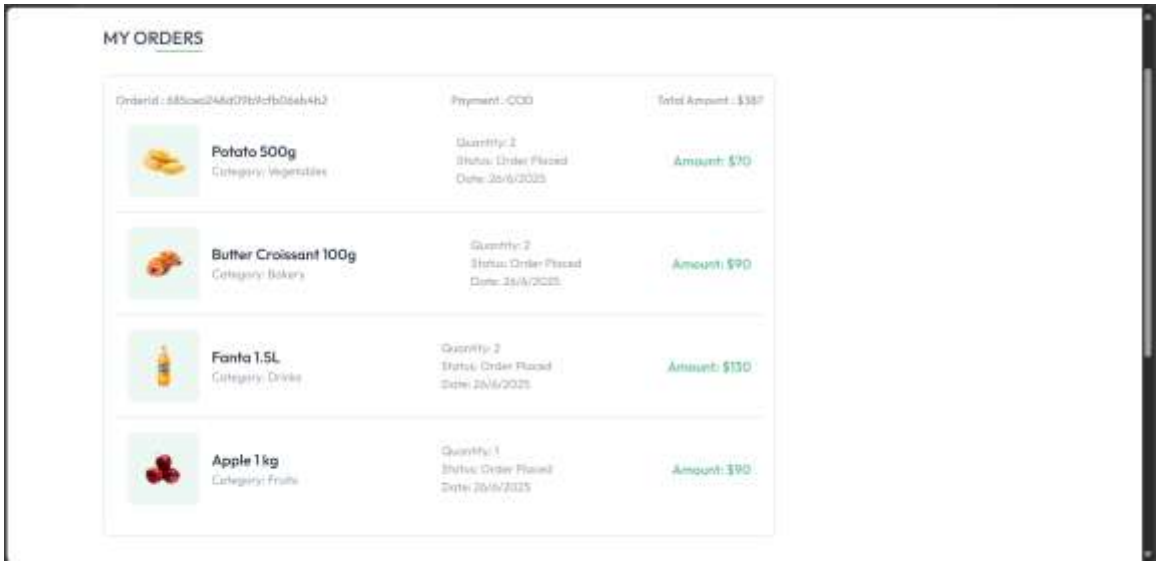
All Products



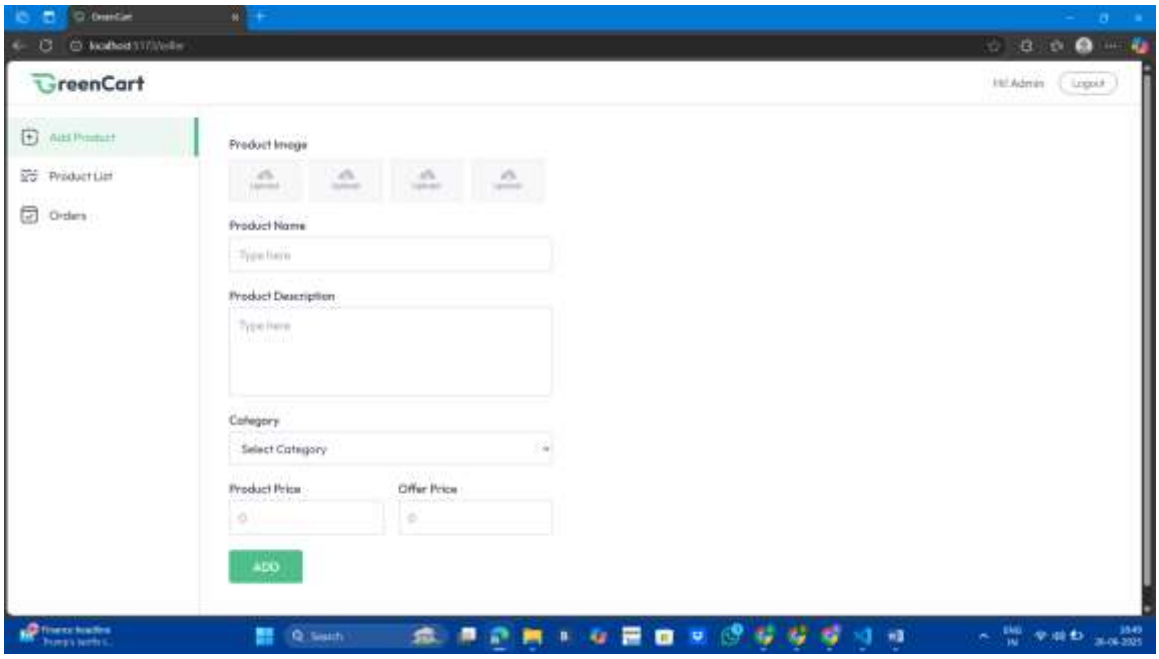
Cart



MyOrders



Admin:



12. Known Issues

• Payment gateway is not live – only simulated.
• No image upload or file storage functionality.
• No user role separation for admin tasks (if admin panel exists).
• No validation for duplicate products.

13. Future Enhancements

• Integrate real payment gateways like Razorpay or Stripe.
• Implement role-based access (admin/user).
• Enable product image uploads using Cloudinary or AWS S3.
• Add product reviews, ratings, and feedback.
• Improve search functionality with filters and suggestions.
• Enable order tracking and invoice generation.