**A REPORT**

**ON**

**DAILYGROCER v5.5: A SMART GROCERY AND ORDER HISTORY MANAGER**

**By**

YARAPA VENU GOPAL AP23110011440

B.Bharadwaj AP23110010254

G.V.Raghavendra AP23110010444

Ch.Sai Santosh AP23110010313

***Prepared in the partial fulfillment of the***

**FULLSTACK**

**AT**

**SMART BRIDGE**



**SRM UNIVERSITY, AP**

**(July, 2025)**

+

**Introduction:-**

GrocerEase is a modern, responsive, and intelligent **online grocery shopping platform** built using **React.js, TypeScript, Tailwind CSS, and JSON-Server**.  
The platform provides users with a smooth online shopping experience, complete with:

* Real-time product browsing
* Cart management
* Order placement
* Category-wise filtering
* Detailed product pages

In addition to customer-facing features, GrocerEase includes a full **Admin Product Management System** powered by **CRUD operations (Create, Read, Update, Delete)**.  
This allows administrators to:

* Add new grocery products
* Modify product details
* Delete unwanted items
* Manage all inventory from a single dashboard

The project is designed to be simple, fast, clean, and scalable — suitable for academic evaluation and real-world deployment.

**Scenario-Based Intro:-**

Imagine a customer opening the GrocerEase website.  
A clean, modern interface appears with fresh groceries arranged beautifully into categories such as fruits, vegetables, daily essentials, snacks, beverages, and dairy.

Users can:

* Browse products
* Filter items
* Add them to cart
* Adjust quantity
* Place real orders
* View previous purchases

Now imagine an admin logging into their backend panel.  
From the **Admin Products Page**, they can:

* Add new products with image, price, and description
* Edit the details of existing products
* Delete outdated products
* View the complete catalogue in a well-structured grid

Every change the admin makes is instantly reflected on the customer side because GrocerEase uses **JSON-Server + Axios** to simulate a real-time backend.

This project demonstrates **full-stack thinking on the frontend**, modern UI, and real CRUD integration.

**Target Audience**

GrocerEase is designed for:

**1. Everyday Users**

To shop groceries efficiently online.

**2. Store Owners / Admin Users**

Who need a lightweight product management system.

**3. Full-stack Developers / Students**

To practice React + Tailwind + CRUD operations.

**4. Academic Evaluators**

Assessing technical skills in:

* UI/UX
* Component architecture
* API integration
* State management
* JSON-Server backend simulation

**Project Goals & Objectives**

**Goals**

* Build a **fully functional grocery shopping system**.
* Implement an **Admin CRUD panel** for product management.
* Use a **modern technology stack** suitable for industry workflows.
* Provide a visually appealing, fast, and responsive UI.

**Objectives**

**✔ User-Friendly Frontend**

Clean pages for:

* Home
* Shop
* Product Details
* Cart
* Checkout
* Order History

**✔ Admin CRUD Module**

Admins can:

* Add new products
* Edit existing ones
* Delete products
* View inventory

**✔ Modern Tech Stack**

* React + TypeScript
* React Router
* Tailwind CSS
* JSON-Server
* Axios

**✔ Component Reusability**

Cards, modals, buttons, quantity selectors, layout shell.

**✔ Performance Optimization**

Fast load time, responsive design, clean animations.

**Key Features:-**

**1. Grocery Website Interface**

* Category-based browsing
* Product cards with image, price, rating
* Add-to-cart functionality
* Product detail page
* Search & filter options
* Stylish, responsive UI

**2. Cart & Checkout System**

* Add / remove items
* Update quantity
* Auto price calculation
* Order validation
* Local storage persistence

**3. Full Admin CRUD Panel**

Admin can:

**Create**

Add a new product (name, price, category, image URL)

**Read**

View all products in a grid/table

**Update**

Edit any product using a custom form

**Delete**

Remove products permanently

All CRUD actions communicate with the backend using Axios + JSON-Server.

**4. Tailwind CSS UI Design**

* Modern layouts
* Hover effects
* Transitions
* Dark/light styling
* Clean spacing & typography

**5. Client-side Routing**

Pages handled via React Router:

* /
* /shop
* /product/:id
* /cart
* /checkout
* /admin/products

**6. JSON-Server Backend**

A fake backend storing:

* products
* orders
* cart states

**PRE-REQUISITES:-**

**✔ Node.js & npm**

To run the React project.

**✔ React + Vite Setup**

npm create vite@latest

npm install

npm run dev

**✔ JSON-Server Setup**

npm install json-server --save-dev

npm run server

**✔ Axios for API Operations**

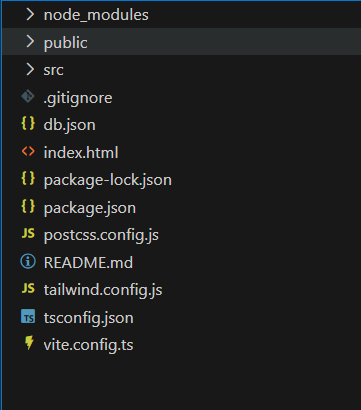
npm install axios

**✔ Tailwind CSS**

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

**Project structure:**

****

The project structure may vary depending on the specific library, framework, programming language, or development approach used. It's essential to organize the files and directories in a logical and consistent manner to improve code maintainability and collaboration among developers.

**PROJECT FLOW:-**

**Project demo:**

Demo link:  
<https://github.com/BollapalliBharadwaj/Grocer_Ease_Demos>  
Use code in:  
<https://github.com/BollapalliBharadwaj/Grocer_Ease>

**Project Flow**

**Milestone 1 – Setup**

* **Vite + React installed**
* **Tailwind configured**
* **JSON-Server created**
* **Basic routing added**

**Milestone 2 – Grocery UI**

* **Product grid**
* **Product card component**
* **Cart page**
* **Order page**
* **Responsive design**

**Milestone 3 – CRUD Features**

**Create**

**Add Product → POST to /products**

**Read**

**Fetch + display all products**

**Update**

**Edit product → PUT request**

**Delete**

**Remove product → DELETE request**

.

**PACKAGE.JSON**:

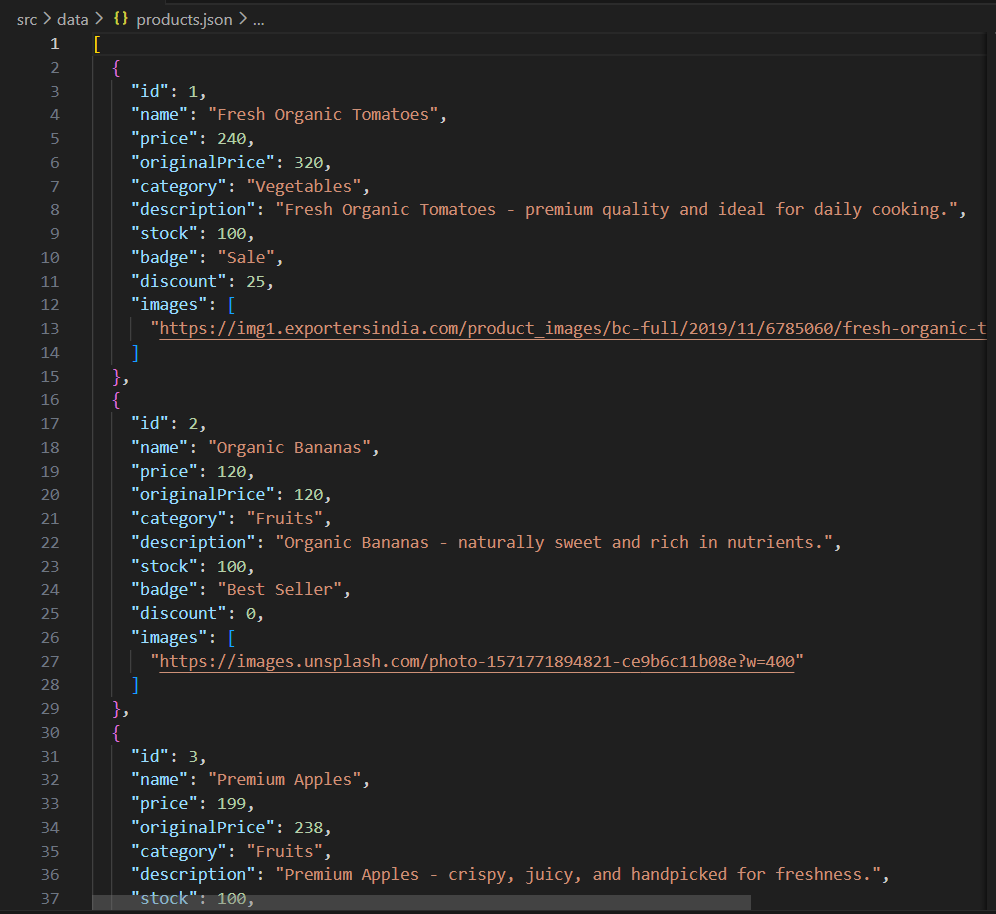


.

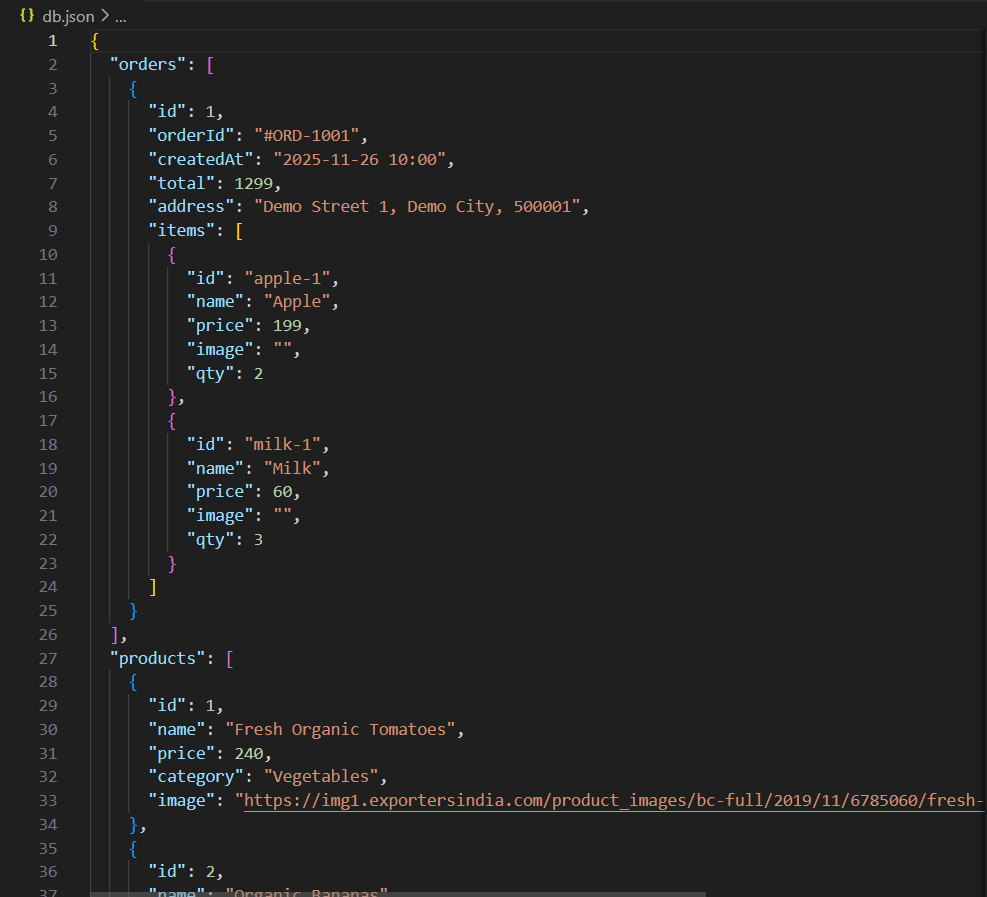
**Main.tsx:**

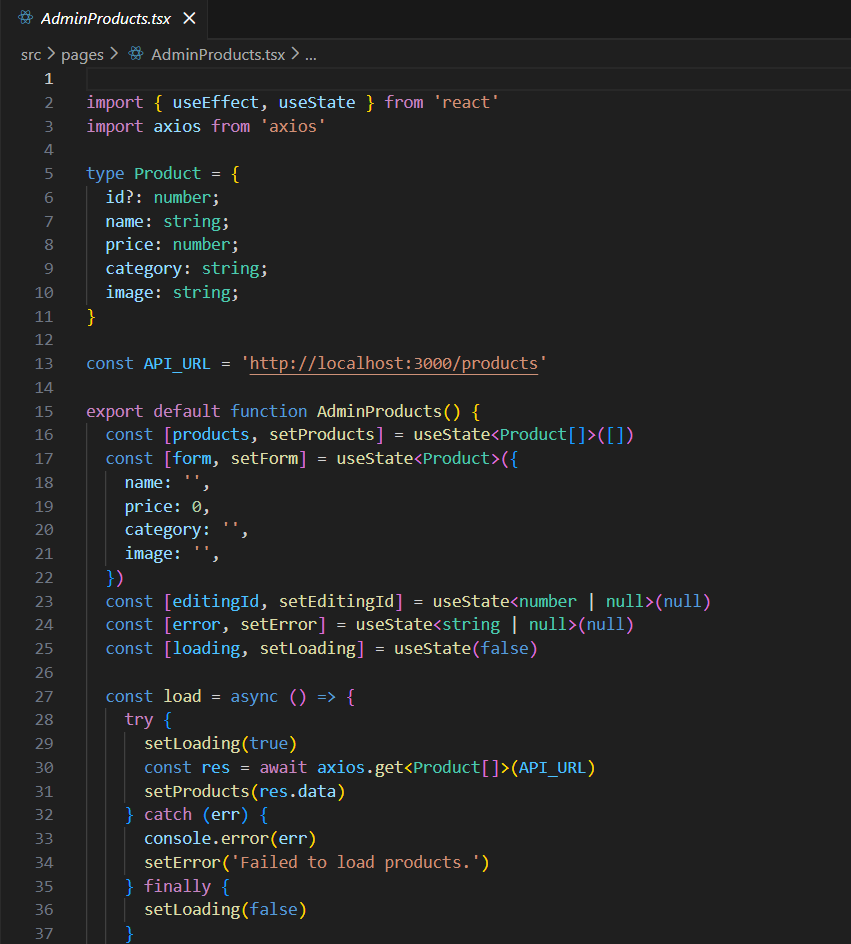


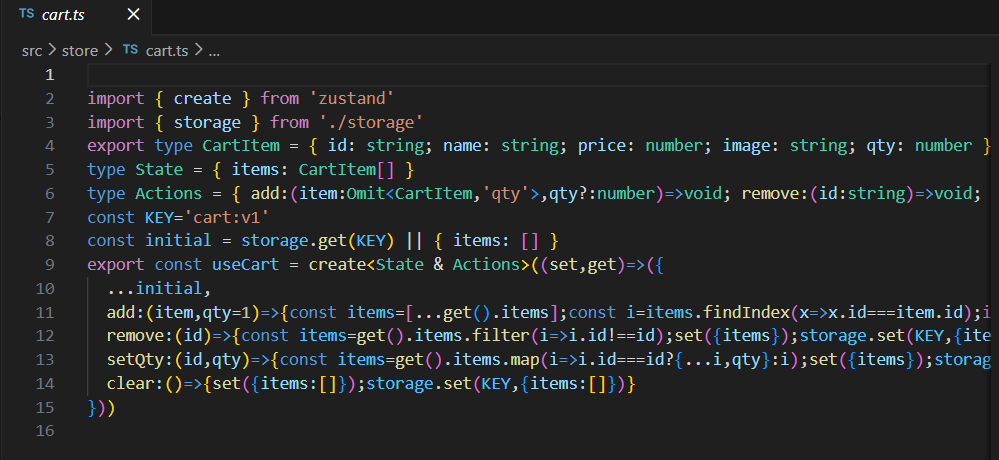
**Products.json:**



**db.json:**



**CRUD Components:  
Admin Products.tsx  
  
**

**Cart.ts  
  
**

**Project Execution**

**Project Execution**

**1. Start Backend**

npm run server

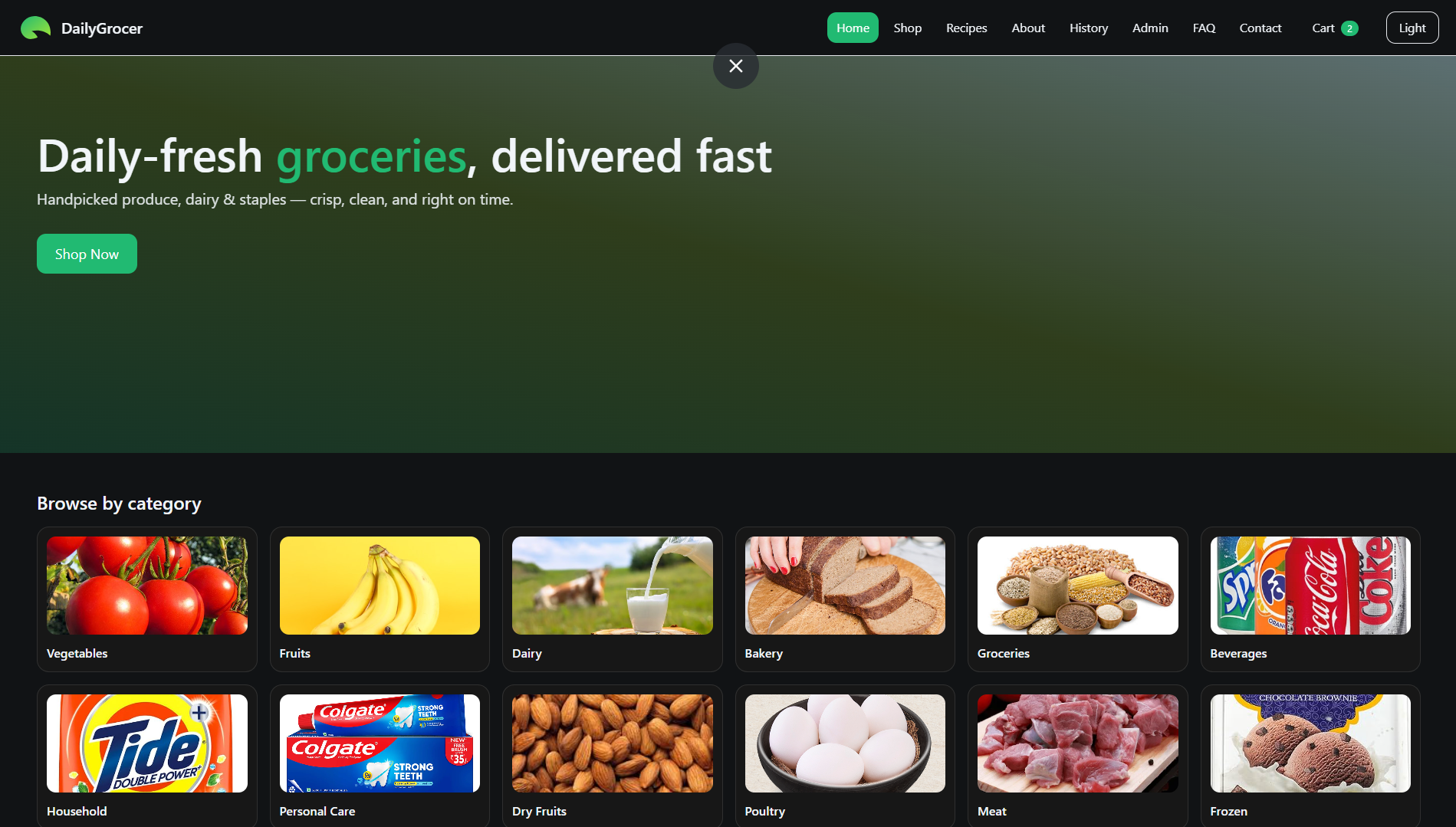
Runs on:  
http://localhost:3001/products

**2. Start Frontend**

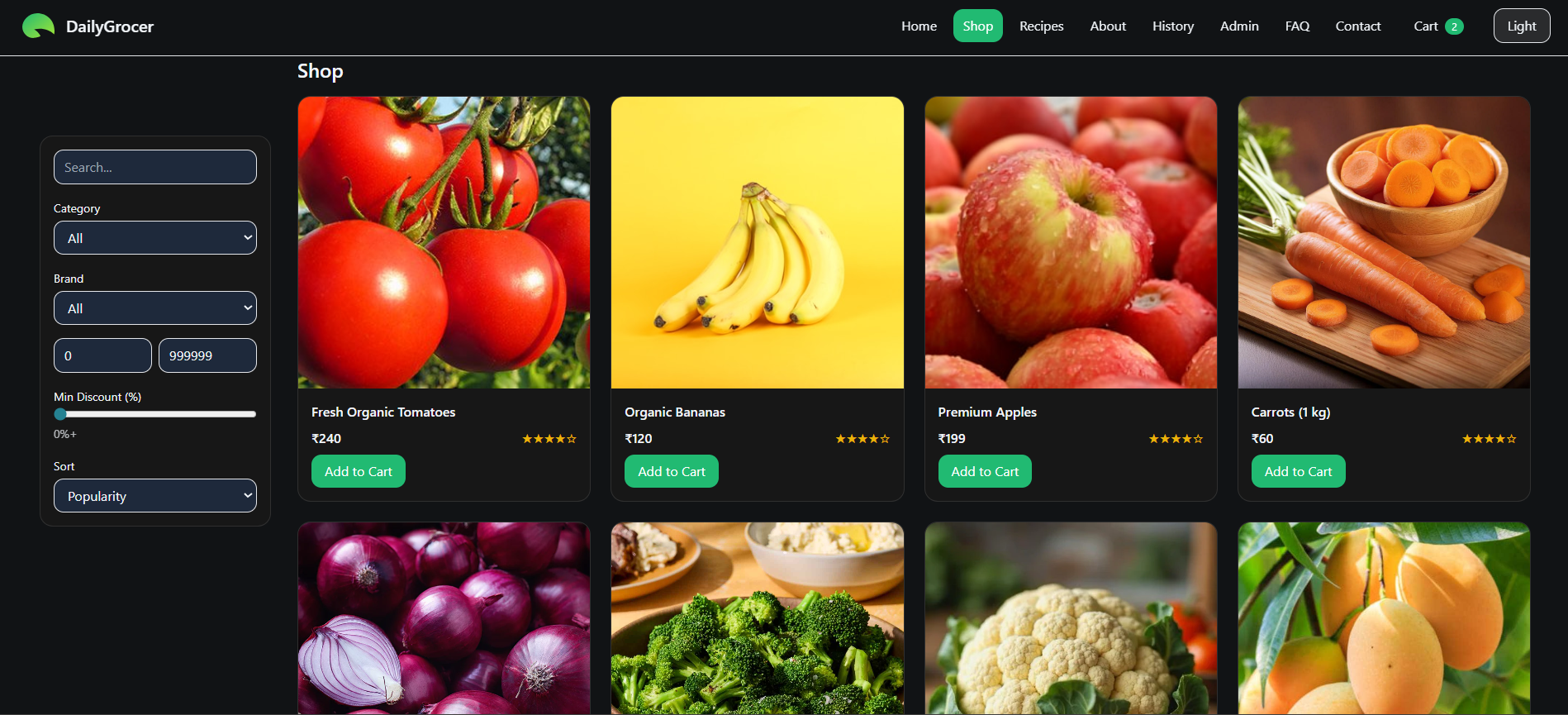
npm run dev

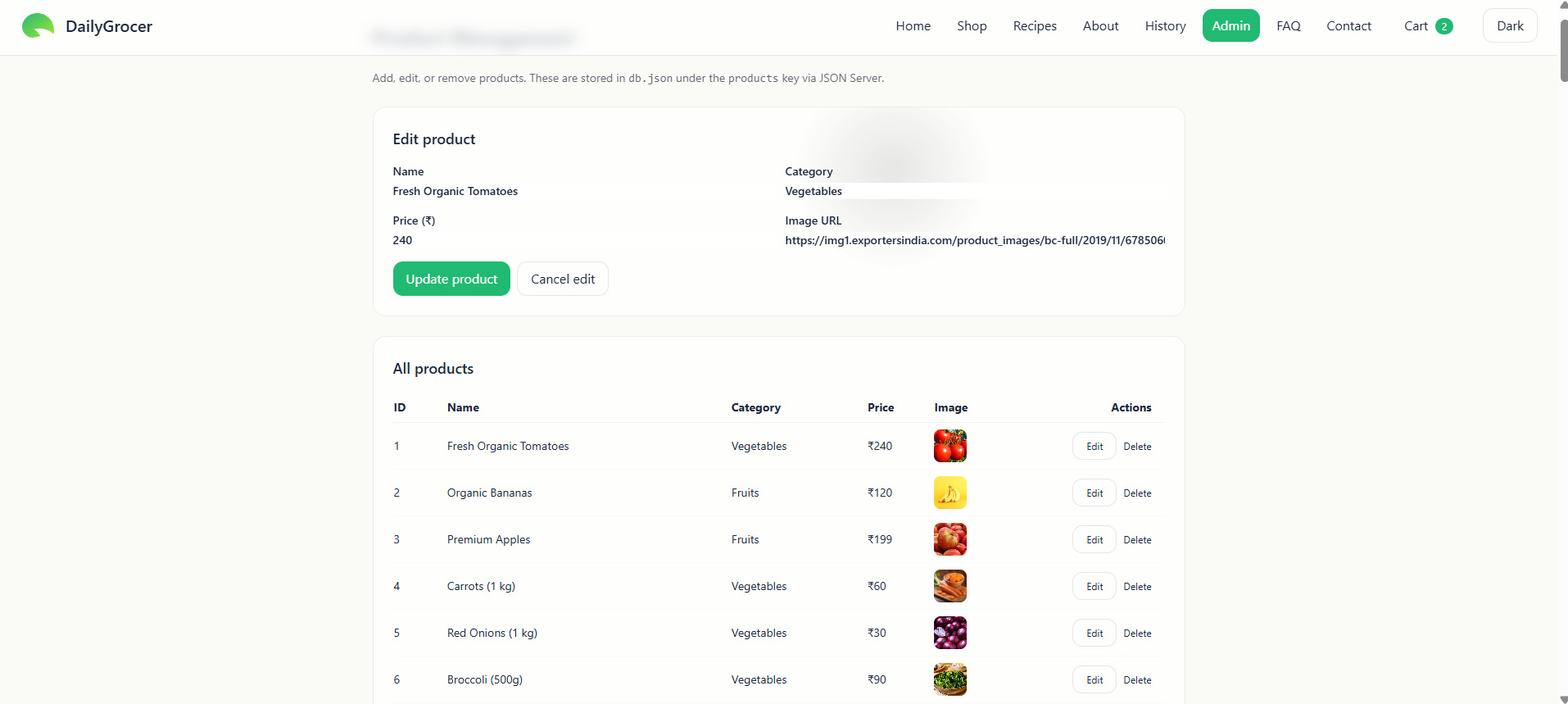
Runs on:  
http://localhost:5173

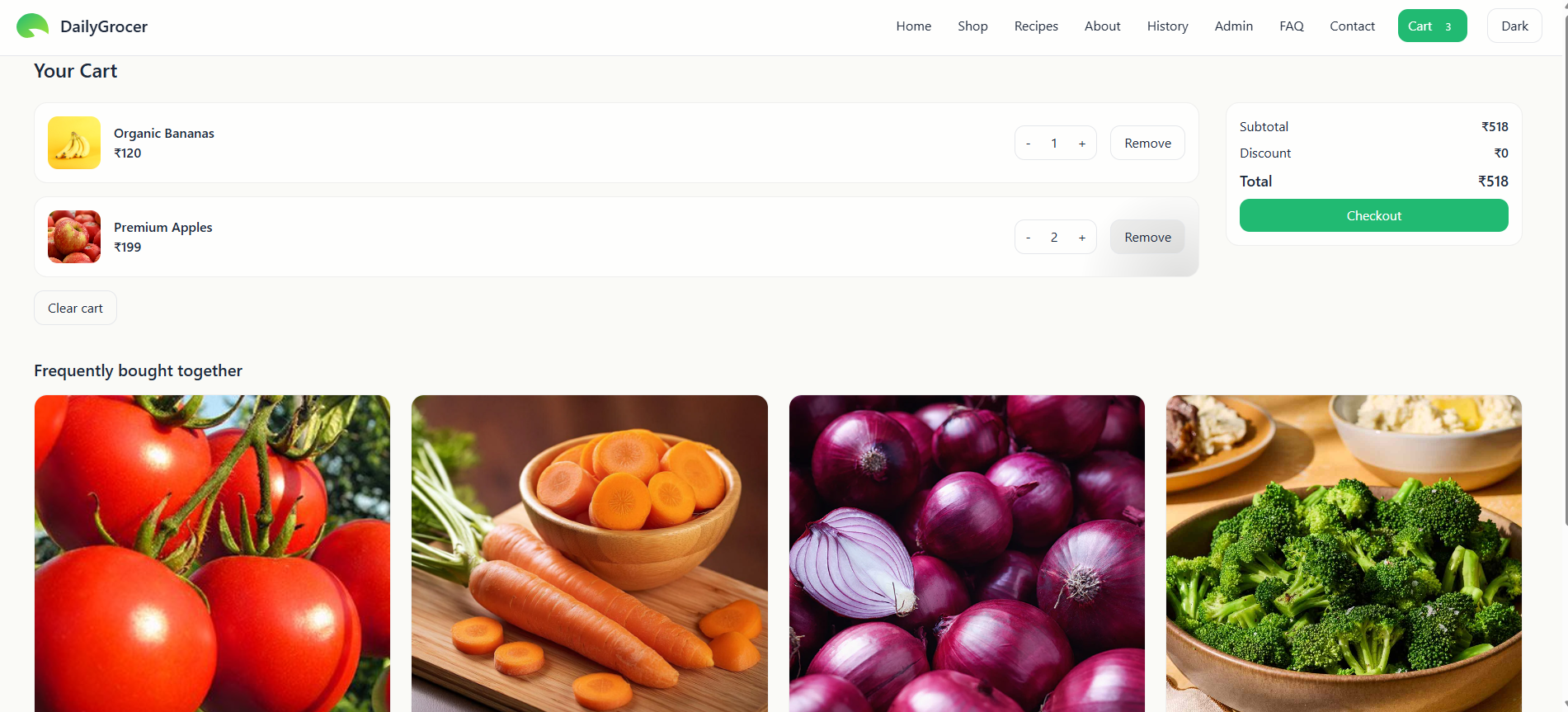
Home Page:

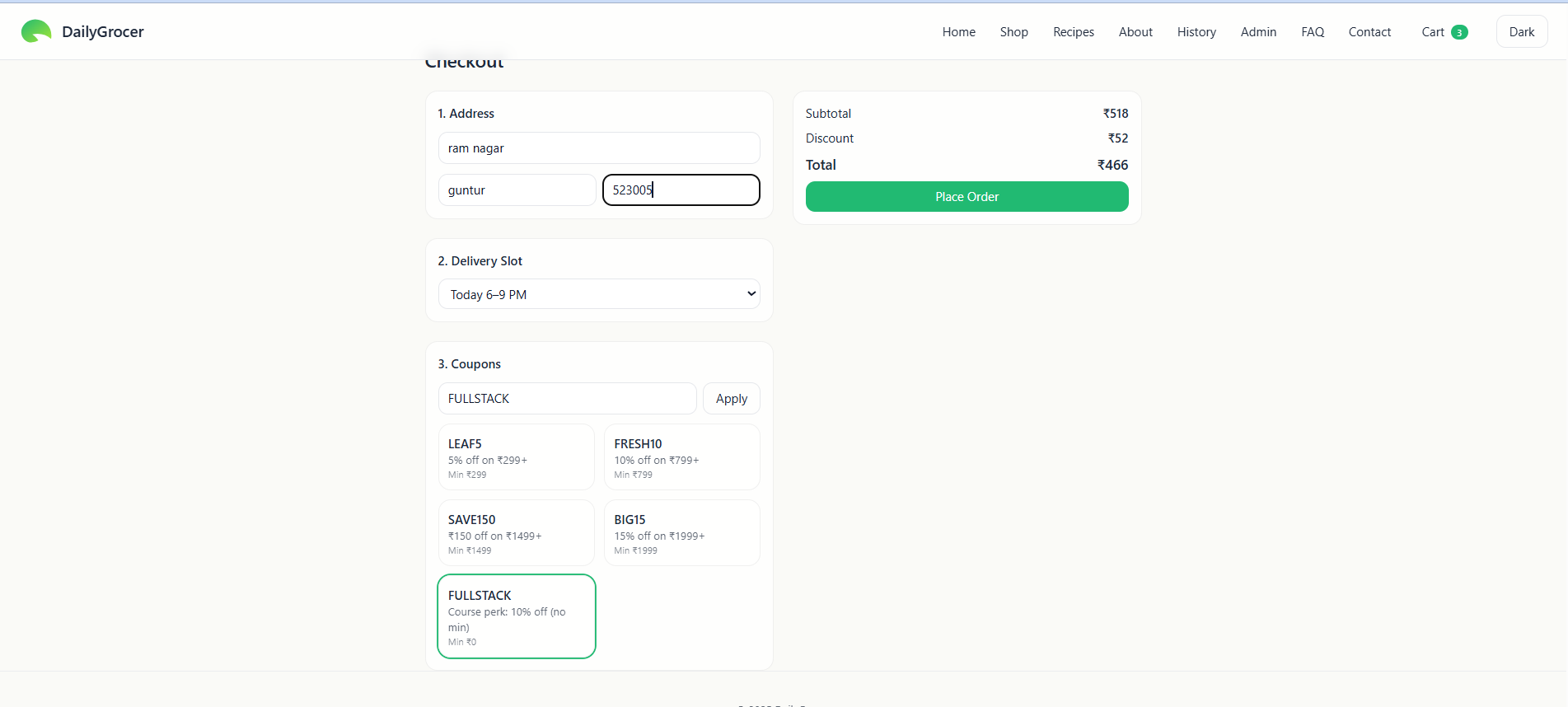


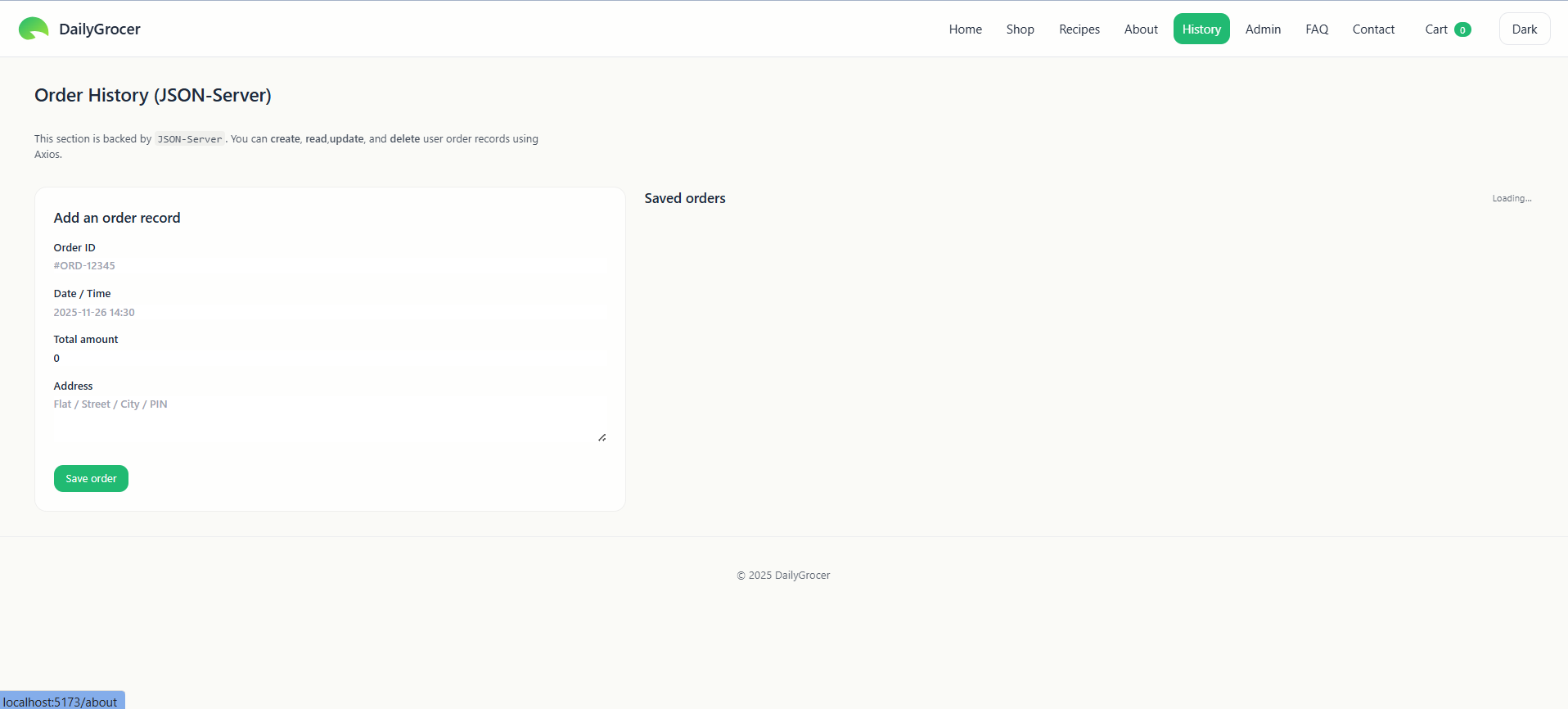
Shop page:

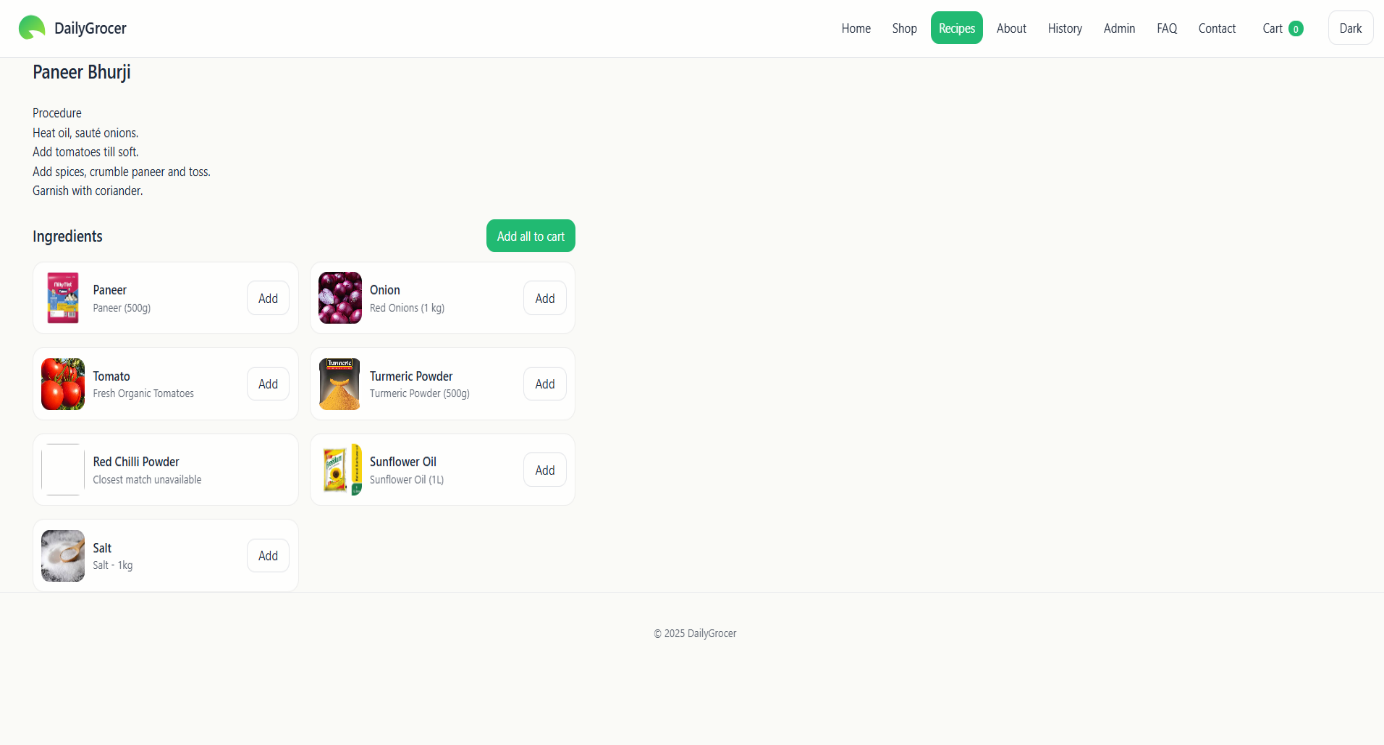


Admin edit page:  


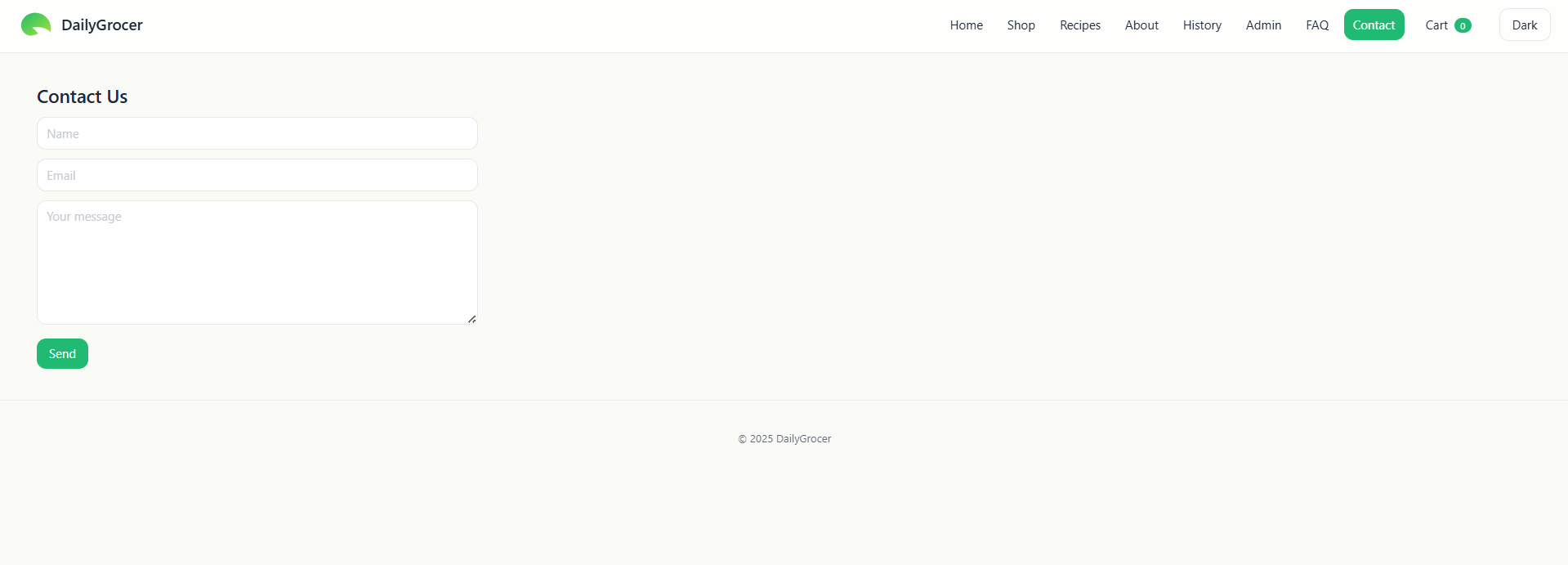
Cart page  


Check Out page:  
  


Order History  
  


Reciepes Page:  
  


Contact Page:



Project Demo Link:  
<https://github.com/BollapalliBharadwaj/Grocer_Ease_Demos>