## **Full Stack Development with MERN**

## **Project Documentation format**

## **1. INTRODUCTION**

### **Project Title**

ShopSmart: Your Digital Grocery Store Experience

### **Team Members**

* Shaik Karimulla – Full Stack Developer
* Syed Ashish Ahmed - Full Stack Developer

## **2. PROJECT OVERVIEW**

### **Purpose**

ShopSmart is designed to provide a seamless online shopping experience for customers, making it convenient to explore and purchase a wide range of products. It aims to digitize traditional grocery shopping with efficient, intuitive features.

### **Features**

* User-friendly navigation
* Secure user authentication and role-based access
* Product listing and filtering
* Shopping cart and checkout system
* Real-time order tracking for users
* Admin dashboard for managing products, orders, and users

## **3. ARCHITECTURE**

### **Frontend**

* Built using Angular JS
* Tailwind CSS for UI styling
* React Router for client-side routing

### **Backend**

* Built using Node.js and Express.js
* RESTful API structure
* Authentication using JWT

### **Database**

* MongoDB with Mongoose ODM
* Schemas for Users, Products, Orders
* Hosted on MongoDB Atlas

## **4. SETUP INSTRUCTIONS**

### **Prerequisites**

* Node.js and npm:<https://nodejs.org/en/download/>
* MongoDB:<https://www.mongodb.com/try/download/community>
* Git:<https://git-scm.com/downloads>

### **Installation**

1. Clone the repository: git clone https://github.com/Karim-786/ShopSmart-
2. Navigate to project directories:  
   * Backend: cd server
   * Frontend: cd client
3. Install dependencies:  
   * Backend: npm install
   * Frontend: npm install
4. Set up environment variables (.env files) for backend:  
     
    PORT=5000

MONGO\_URI=your\_mongodb\_connection\_string

JWT\_SECRET=your\_jwt\_secret

1. Start the servers:  
   * Backend: npm start
   * Frontend: npm start

## **5. FOLDER STRUCTURE**

### **Client**

* /src  
  + /components: Reusable UI components
  + /pages: Feature-based pages
  + /services: API service files
  + /context: Context API for state management
  + /App.js, /index.js: App initialization

### **Server**

* /controllers: Handles API logic
* /models: Mongoose models for DB
* /routes: Defines API routes
* /middleware: JWT & role-based middleware
* /config: DB connection setup

## **6. RUNNING THE APPLICATION**

### **Frontend**

cd client

npm start

### **Backend**

cd server

npm run dev

## **7. API DOCUMENTATION**

### **Authentication**

* POST /api/auth/register: Register a new user
* POST /api/auth/login: Login user

### **Products**

* GET /api/products: Fetch all products
* POST /api/products: Add product (admin only)
* PUT /api/products/:id: Update product (admin only)
* DELETE /api/products/:id: Delete product (admin only)

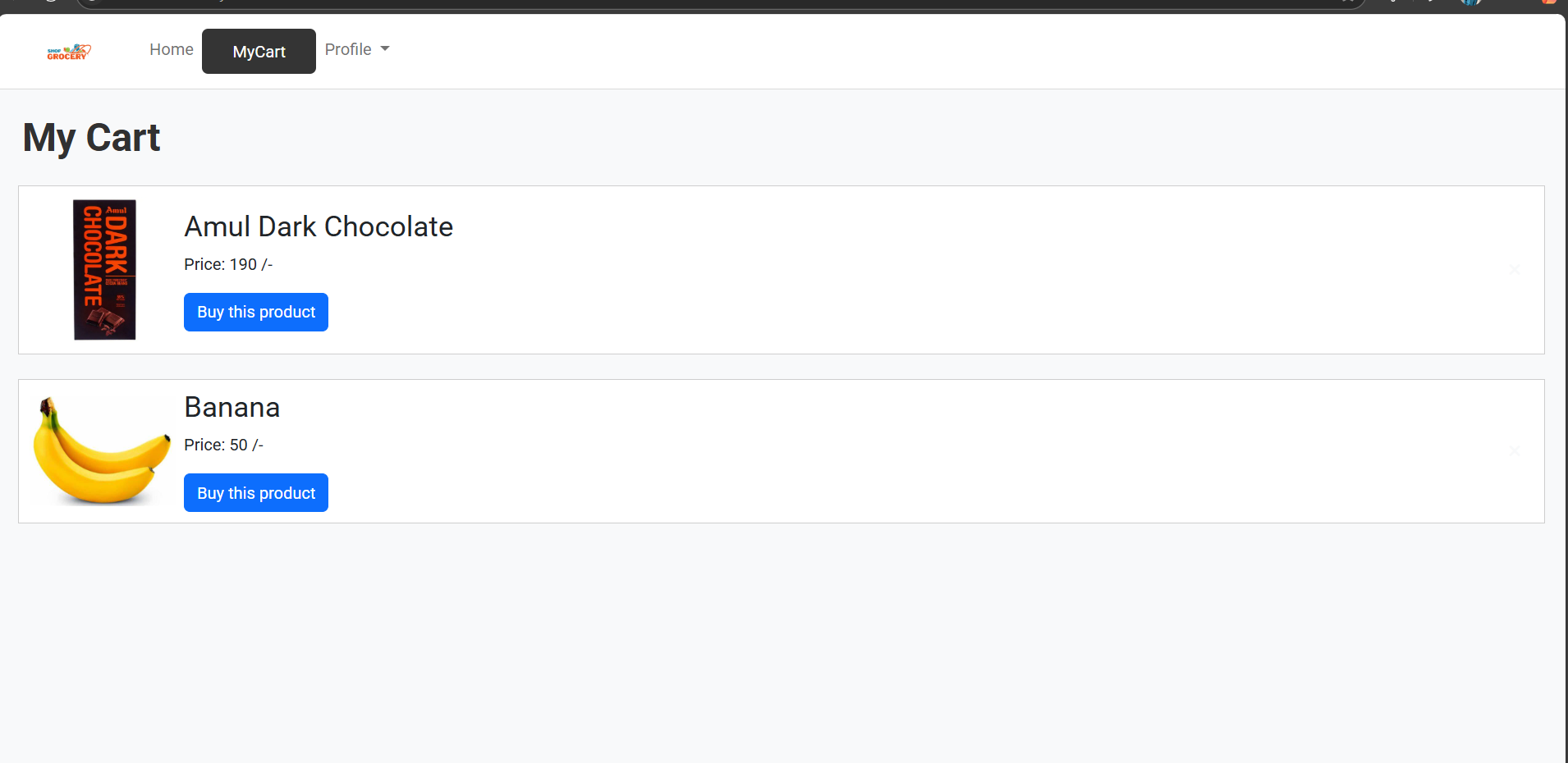
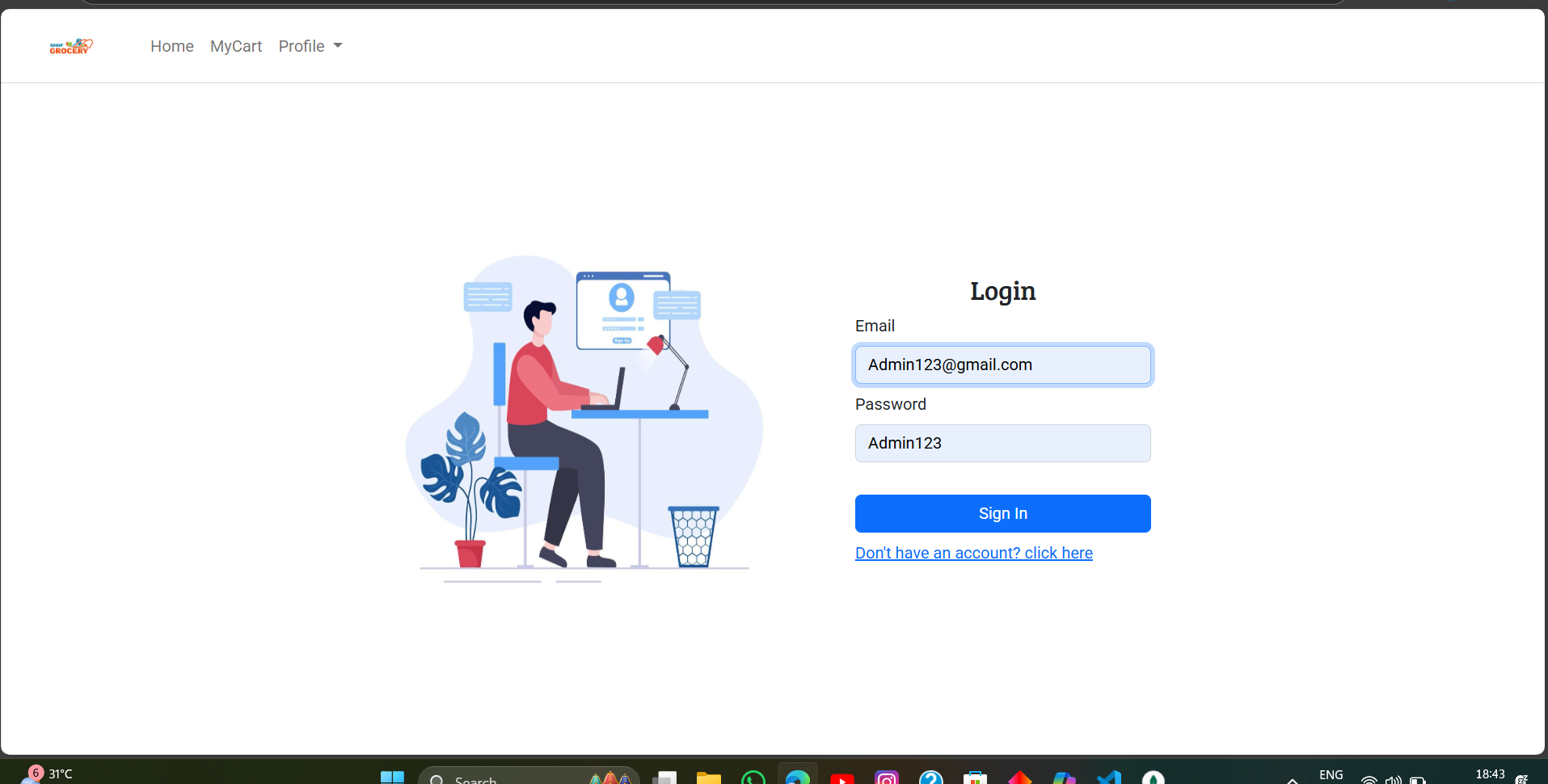
### **Orders**

* POST /api/orders: Place an order
* GET /api/orders/user/:id: Get user orders
* GET /api/orders: Get all orders (admin only)

## **8. AUTHENTICATION**

* JWT tokens are generated during login
* Stored in client-side localStorage
* Admin routes protected by adminAuthenticateToken middleware
* Regular users are restricted from admin functionalities

## **9. USER INTERFACE**

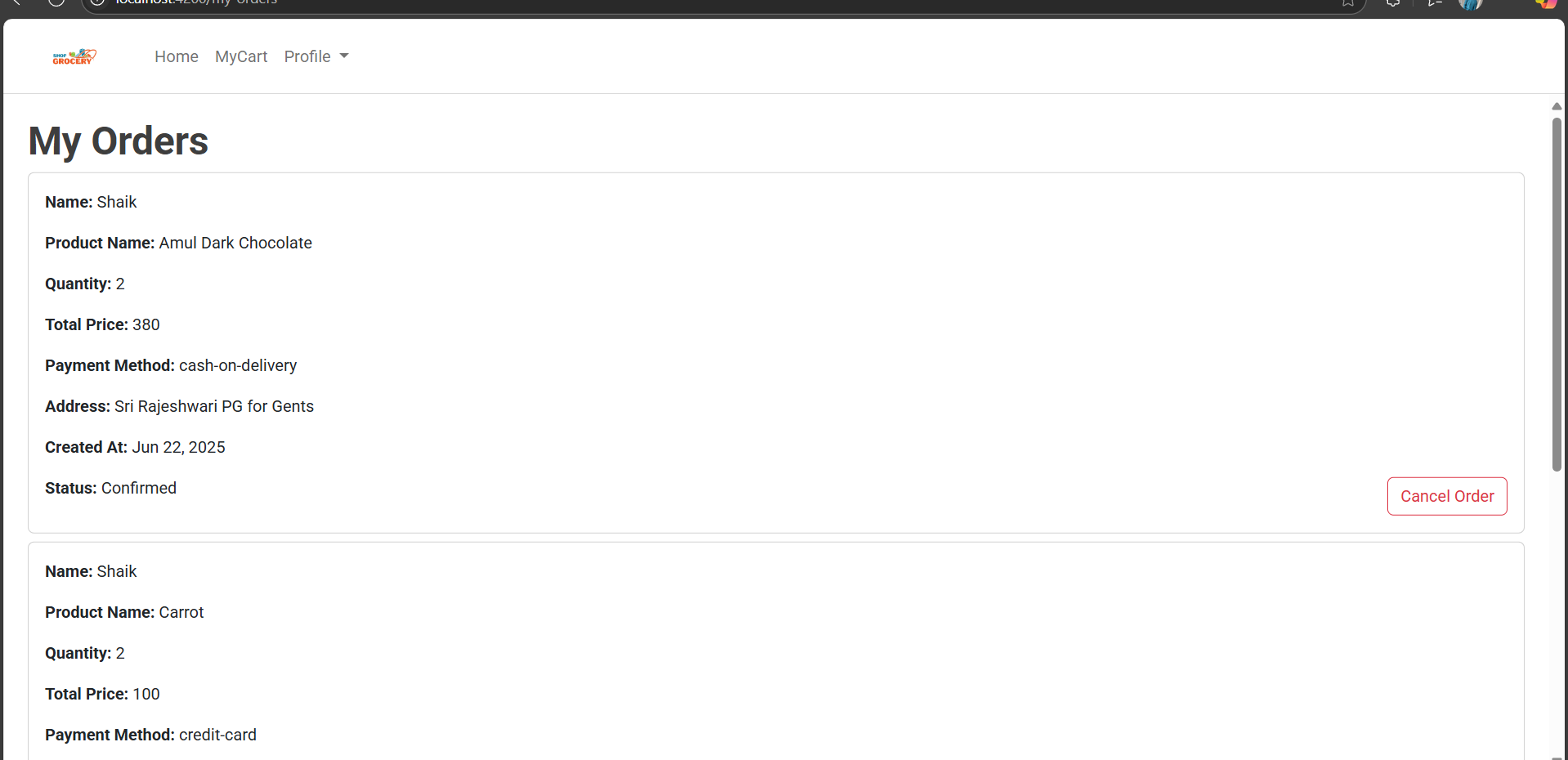
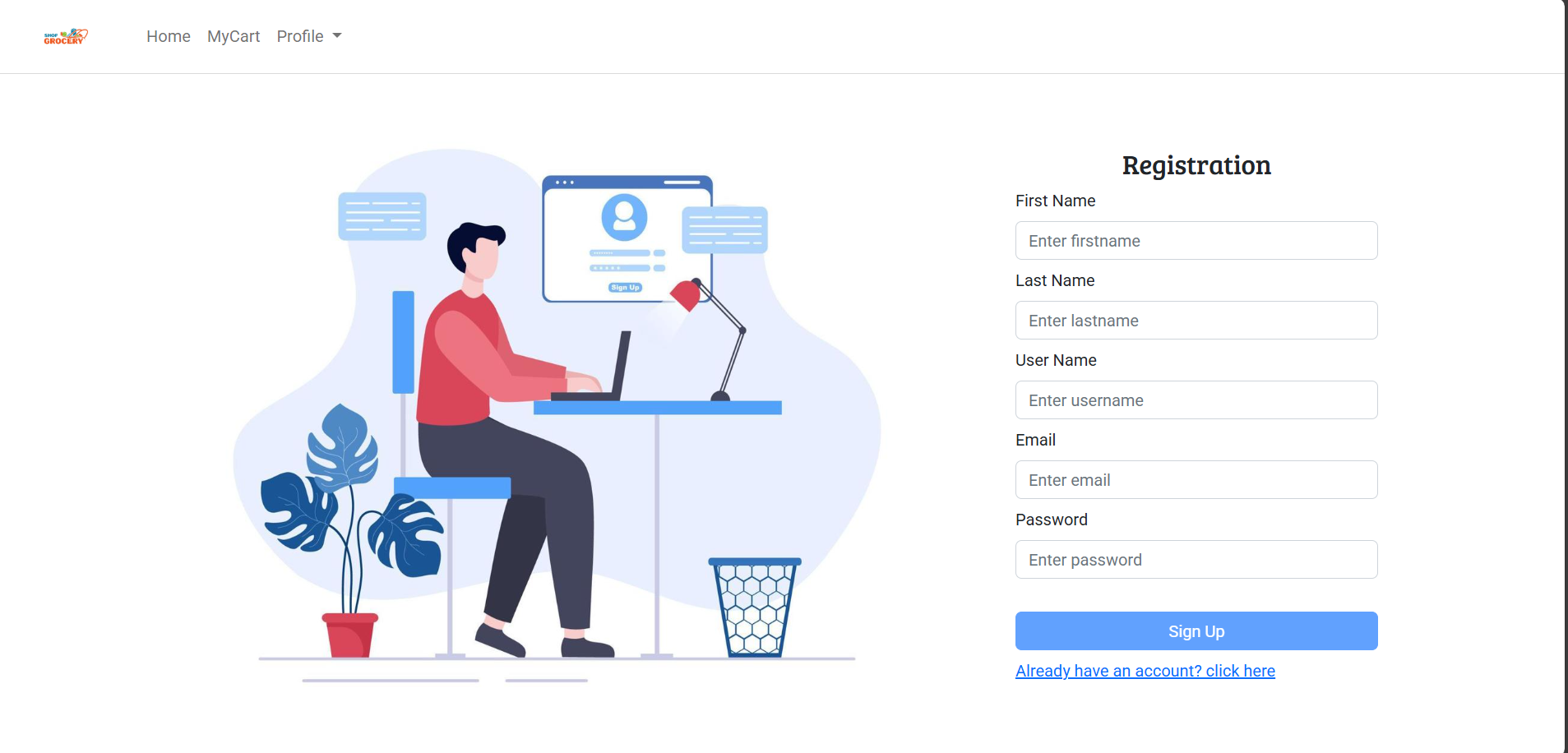


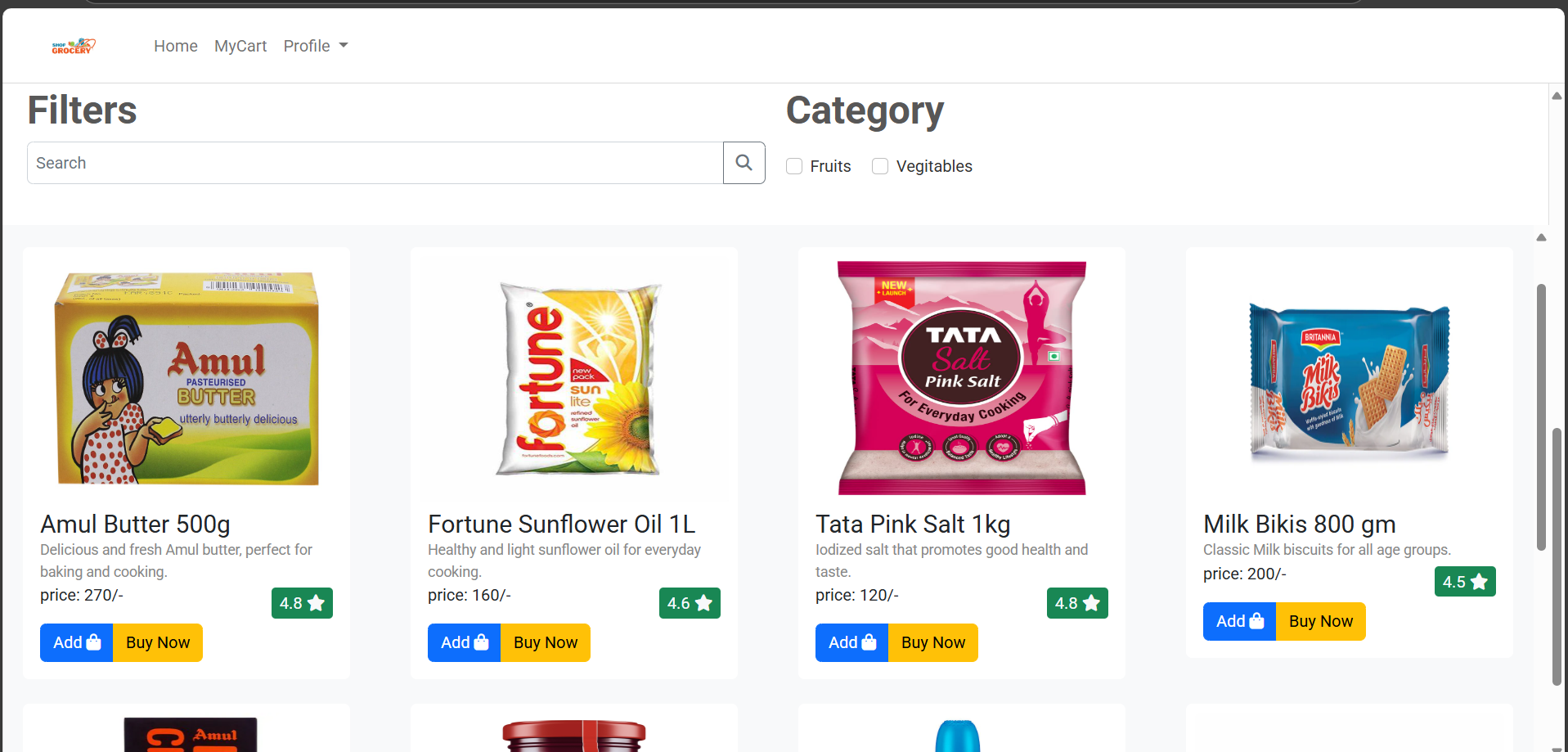
* User Login / Register
* Product Grid / Details
* Cart View
* Admin Panel
* Order Tracker

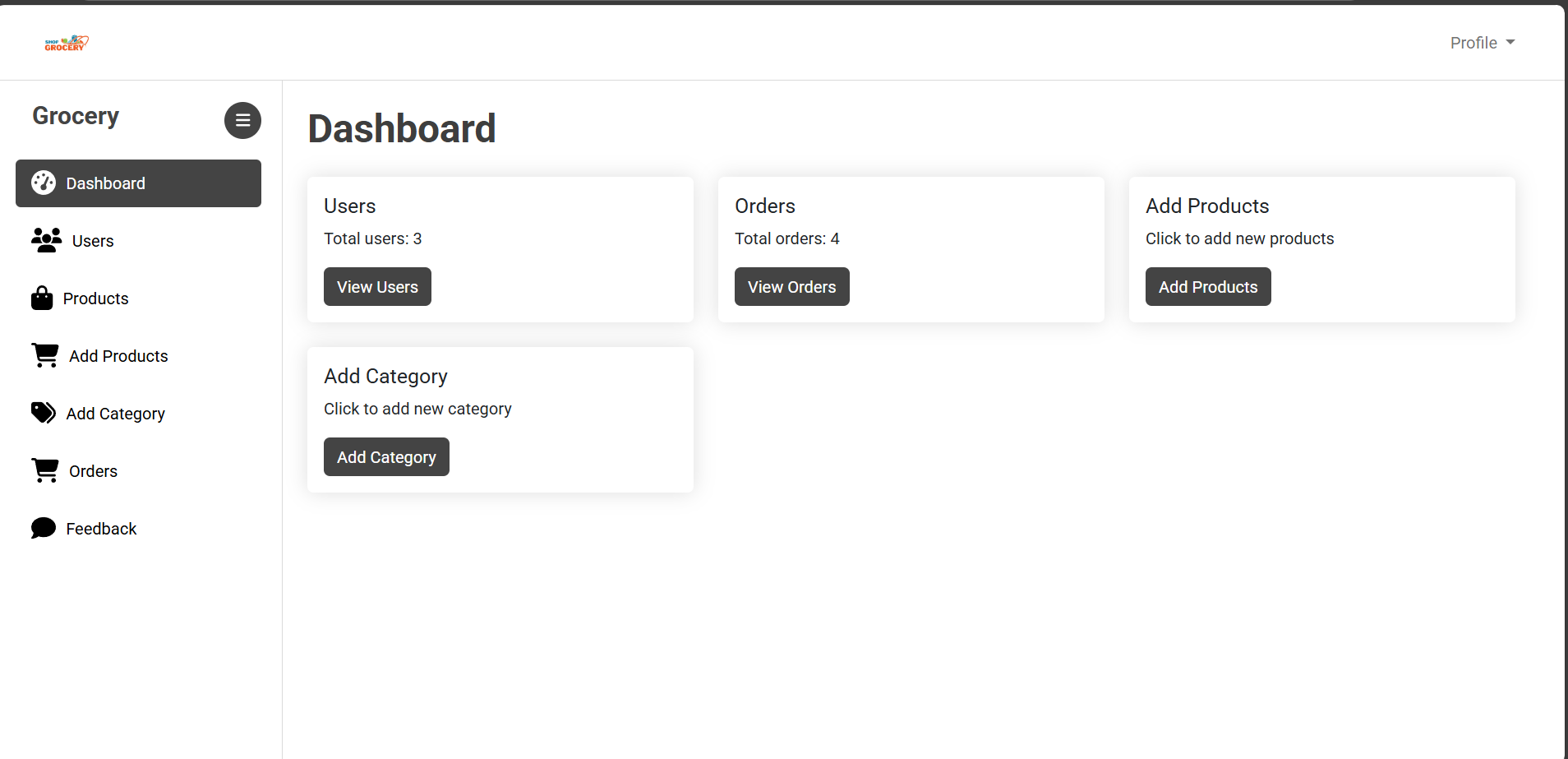
## **10. TESTING**

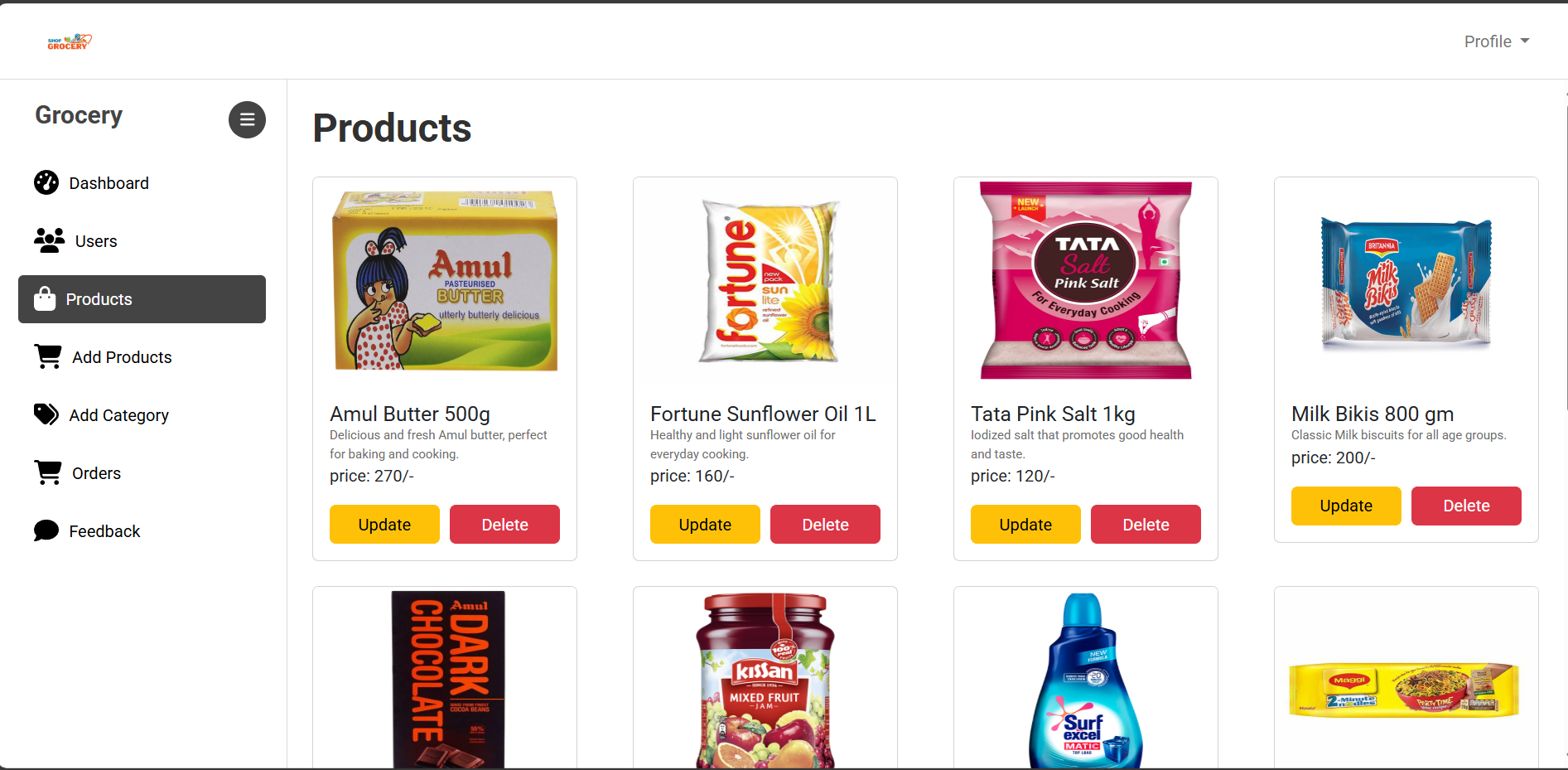
* Manual testing via Postman
* Frontend testing through Chrome DevTools
* Future scope: Jest or Mocha/Chai unit tests

## **11. SCREENSHOTS OR DEMO**









## **12. KNOWN ISSUES**

* No real payment gateway integration
* Product search may not support fuzzy matching
* Map tracking is static (future integration pending)

## **13. FUTURE ENHANCEMENTS**

* Integrate Razorpay/Stripe for secure payments
* Implement mobile responsive UI completely
* Add product recommendations
* Integrate chatbot for support
* Deploy as a Progressive Web App (PWA)

### **APPENDIX**

* GitHub Repo:<https://github.com/Karim-786/ShopSmart->
* Angular Setup (For Reference if Angular used): [https://angular.io](https://angular.io/)