

# **Full Stack Development with MERN**

## **Project Documentation**

**Project Title: ShopSmart - Your Digital Grocery Store Experience**

## **1. Introduction**

Project Title: ShopSmart - Your Digital Grocery Store Experience

Team Leader : Ambati swetha

Team member : Harika Annam

Team member : Bakkamanthula Gouthami

Team member : Chedella Rajasri Leelavati

## **2. Project Overview**

### **Purpose:**

ShopSmart is a grocery web application designed to provide a seamless online shopping experience. Users can browse products, add items to cart, and place orders securely. Admins can manage products, users, and orders efficiently.

### **Features:**

- User Registration and Login
- Product Browsing and Search
- Shopping Cart Management
- Order Placement and History
- Admin Dashboard
- Product Add/Update/Delete
- Role-Based Access Control
- Secure Authentication using JWT

### **3. Architecture**

#### **Frontend:**

Developed using modern frontend framework (React/Angular). Handles UI rendering, routing, API calls, and responsive design.

#### **Backend:**

Developed using Node.js and Express.js. Implements RESTful APIs, business logic, authentication, and middleware.

#### **Database:**

MongoDB used as NoSQL database with Mongoose ODM. Collections include Users, Products, Orders, and Cart.

## 4. Setup Instructions

### Prerequisites:

- Node.js and npm
- MongoDB
- Git
- VS Code
- Web Browser (Chrome/Firefox)

### Installation Steps:

- Clone the repository from GitHub.
- Navigate to project directory.
- Run npm install to install dependencies.
- Start backend using npm run dev.
- Start frontend using npm start.
- Open <http://localhost:5100> in browser.

## **5. Folder Structure**

### **Client Folder:**

Contains src, components, pages, services, App.js, index.js.

### **Server Folder:**

Contains models, routes, controllers, middleware, config, server.js.

## 6. API Documentation

- POST /api/register - Register user
- POST /api/login - Login user
- GET /api/products - Get products
- POST /api/products - Add product (Admin)
- PUT /api/products/:id - Update product
- DELETE /api/products/:id - Delete product
- POST /api/orders - Place order
- GET /api/orders - Get orders

## **7. Authentication**

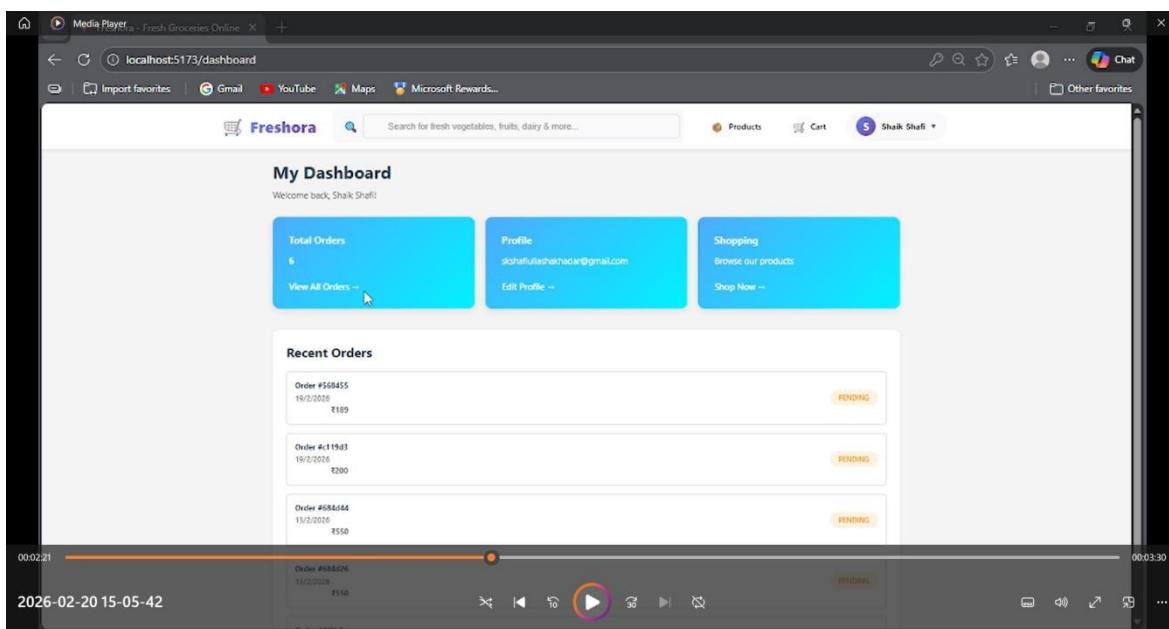
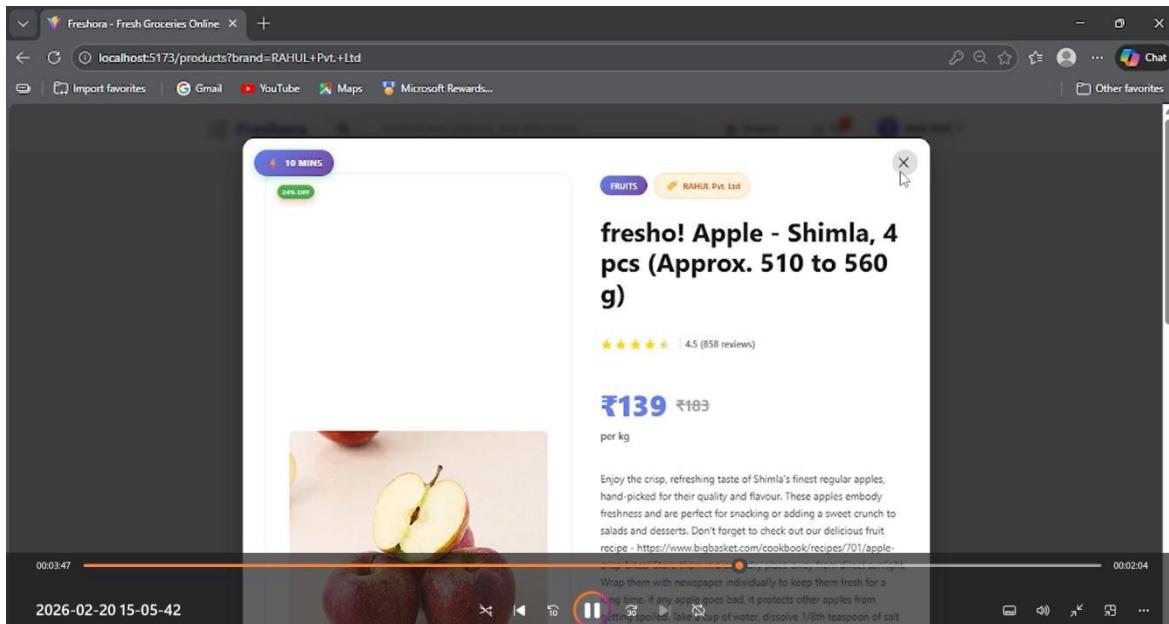
Authentication is handled using JWT tokens. Passwords are hashed using bcrypt. Role-based authorization is implemented for Admin and User access control.

## **8. Testing**

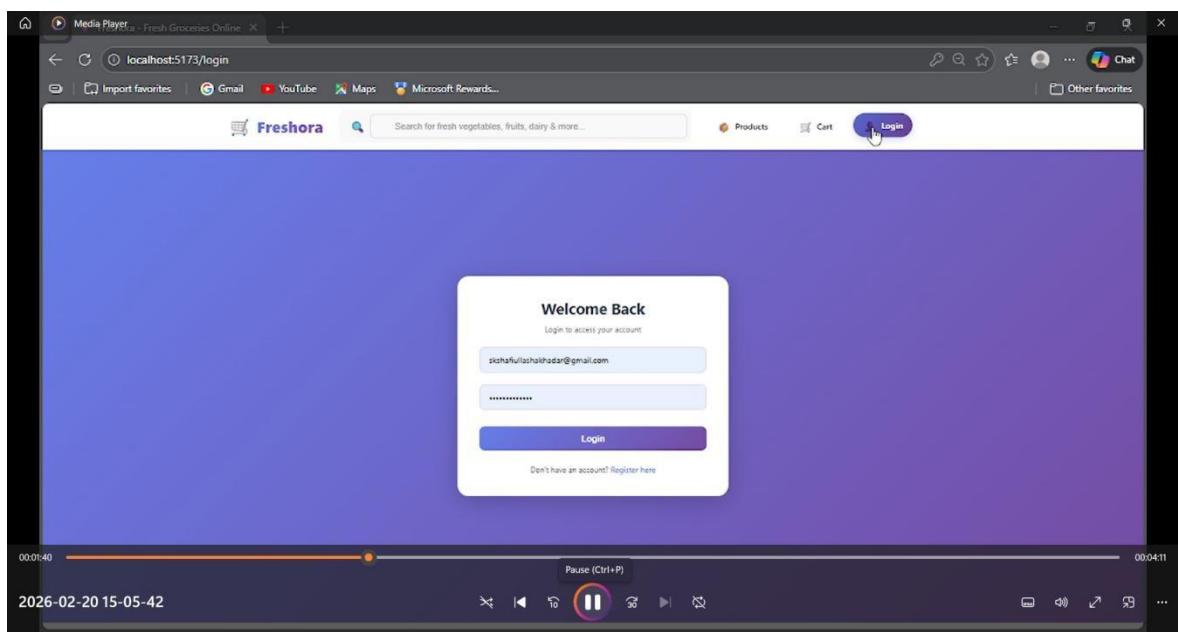
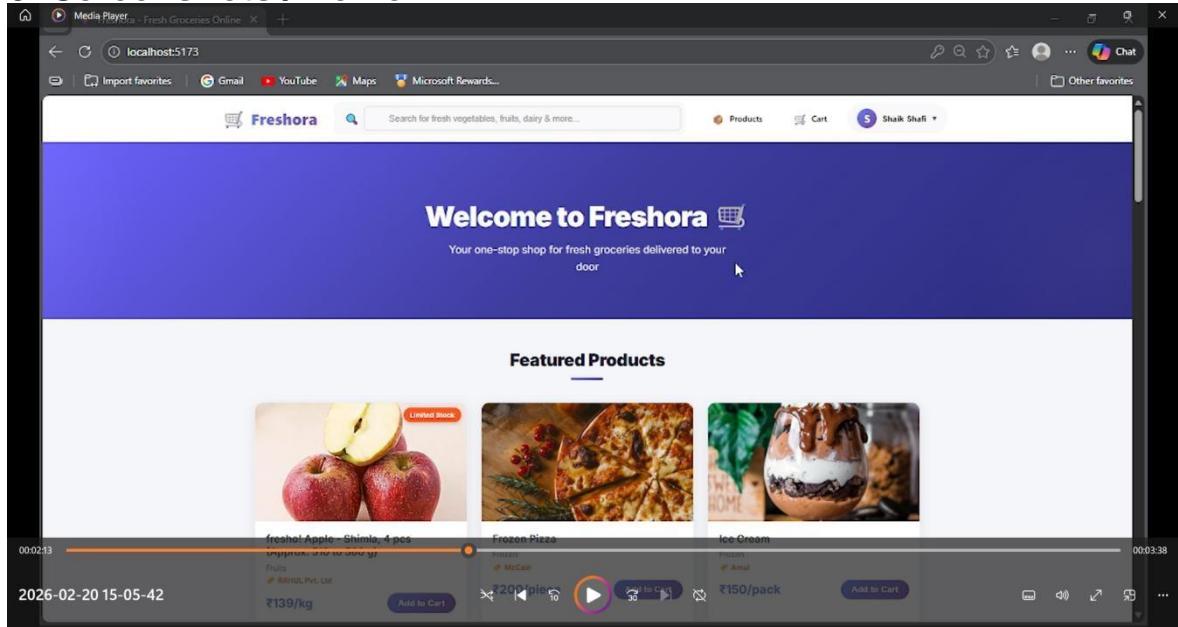
API testing performed using Postman. Manual testing conducted for user flows including registration, login, cart management, and order placement.

## 9. Screenshots / Demo

Demo link provided in project documentation Google Drive link:-



## 9. Screenshots / Demo



## **10. Known Issues**

Minor UI responsiveness issues. Payment gateway not fully integrated.

## **11. Future Enhancements**

- Online Payment Integration
- Email Notifications
- Product Reviews and Ratings
- Mobile Application Version
- Performance Optimization