1. INTRODUCTION

1.1 Project Overview

ShopSmart: Your Digital Grocery Store Experience is a full-stack web application designed to revolutionize the way people shop for groceries online. The application offers a seamless and intuitive shopping experience tailored for diverse users — from tech enthusiasts and fashion-forward individuals to homemakers searching for daily essentials.

Customers can effortlessly navigate through multiple product categories, view detailed product information, add items to their shopping cart, and securely complete their purchases. The system prioritizes user experience, aiming for a hassle-free and secure checkout process.

Beyond customer features, ShopSmart also caters to sellers and administrators. Sellers can efficiently manage their product listings, inventory, and orders, while administrators are equipped with tools to handle customer service, monitor transactions, and oversee app performance.

1.2 Purpose

The primary purpose of ShopSmart is to provide an easy-to-use, secure, and efficient platform for grocery shopping. It bridges the gap between local sellers and online customers, offering:

- Convenience and accessibility for customers to purchase groceries online.
- A centralized platform for sellers to manage inventory and orders.
- Secure handling of user data and transactions.
- A scalable solution that can grow with increasing user demand.

2. IDEATION PHASE

2.1 Problem Statement

General Problem Statement
Many individuals, especially the elderly, face challenges when accessing essential services due to inefficiencies in current systems. Key issues include:

- Lack of transparency or updates for customers and complainants.
- Delayed responses and unclear resolution timelines.
- Poor tracking mechanisms for repeated or high-priority issues.
- Absence of data-driven insights to enhance service quality.

Customer Problem Statement

In today's fast-paced lifestyle, customers encounter numerous difficulties with traditional grocery shopping, including:

- Long queues and limited store hours.
- Inconsistent product availability.
- Time constraints preventing physical store visits.

Customers are increasingly seeking a digital solution that enables them to:

- Browse a wide variety of high-quality groceries.
- Compare prices and place orders conveniently from home.
- Enjoy a seamless checkout and timely delivery experience.
- Receive fresh produce, transparent pricing, and personalized recommendations.
- Get real-time stock updates and flexible delivery options tailored to their lifestyle.

2.2 Empathy Map Canvas

An **Empathy Map** is a visual tool that helps project teams understand user behavior, needs, goals, and pain points by seeing the world from their perspective. It promotes user-centered thinking during product development.

Creating an empathy map for this project allowed the team to step into the customers' shoes and ask:

- **What does the user think and feel?**

They desire convenience, freshness, and trust in online grocery delivery.

- **What does the user hear?**

Mixed feedback from others about the reliability of online services.

- **What does the user see?**

Competing platforms with varying service quality and UI complexity.

- **What does the user say and do?**

Express concerns about delivery delays, price mismatches, and low product variety.

- **What are their pains?**

Uncertainty in product quality, limited technical knowledge (especially for older users), and confusion in navigating online platforms.

- **What are their gains?**

Time saved, home delivery convenience, and consistent availability of products.

This canvas serves as a foundational tool in ensuring our solution is empathetic, inclusive, and effective.

2.3 Brainstorming & Idea Prioritization

Brainstorming is a collaborative approach that encourages all team members to think freely and creatively when addressing a problem. In this project, the process followed these steps:

- **Step 1: Team Gathering & Problem Selection**

The team assembled to analyze the challenges of traditional grocery shopping and agreed on developing an online grocery platform as the core solution.

- **Step 2: Idea Listing and Grouping**

Team members contributed diverse ideas, including features like real-time inventory updates, personalized suggestions, multi-seller integration, secure payments, and userfriendly navigation. Similar ideas were grouped together.

- **Step 3: Idea Prioritization**

Ideas were evaluated based on feasibility, user impact, and implementation effort. High-priority features were chosen for the MVP (Minimum Viable Product), such as:

- Secure checkout and payment
- Admin and seller dashboards
- Cart management and order tracking
- Category-based product browsing

Reference Template: Mural - Brainstorm and Idea Prioritization

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

While not explicitly detailed, the typical customer journey in the **Shop Smart Grocery Web App** involves the following steps:

1. **Visit Website:** The user lands on the homepage.
2. **User Registration/Login:** New users register or log in using credentials or OAuth.
3. **Product Discovery:** Users browse by category or search specific grocery items.
4. **Cart Management:** Users add products to the cart, review, and modify quantities.
5. **Checkout:** Users complete payment securely via integrated gateways.
6. **Order Confirmation:** Users receive confirmation and delivery timeline.

7. **Order Tracking:** Real-time order status and delivery updates are shown.
8. **Feedback:** Users can rate the product and provide feedback.

3.2 Solution Requirements

Functional Requirements

The key functional capabilities of the platform include:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Authentication**	- Sign up, Login, Password Reset - OAuth login (Google / GitHub)
FR-2	**Product Management (Admin)**	- Add/Edit/Delete grocery products - View product inventory
FR-3	**Product Browsing & Search (User)**	- Browse products by category - Search specific products

Non-Functional Requirements

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Intuitive and user-friendly UI for users of all age groups.
NFR-2	**Security**	Encrypted data transfer; role-based access; JWT/tokenbased authentication.
NFR-3	**Performance**	Fast page load (≤ 2 seconds); responsive actions.
NFR-4	**Availability**	System should ensure 99.9% uptime.
NFR-5	**Scalability**	Microservice support for high concurrency and easy feature integration.

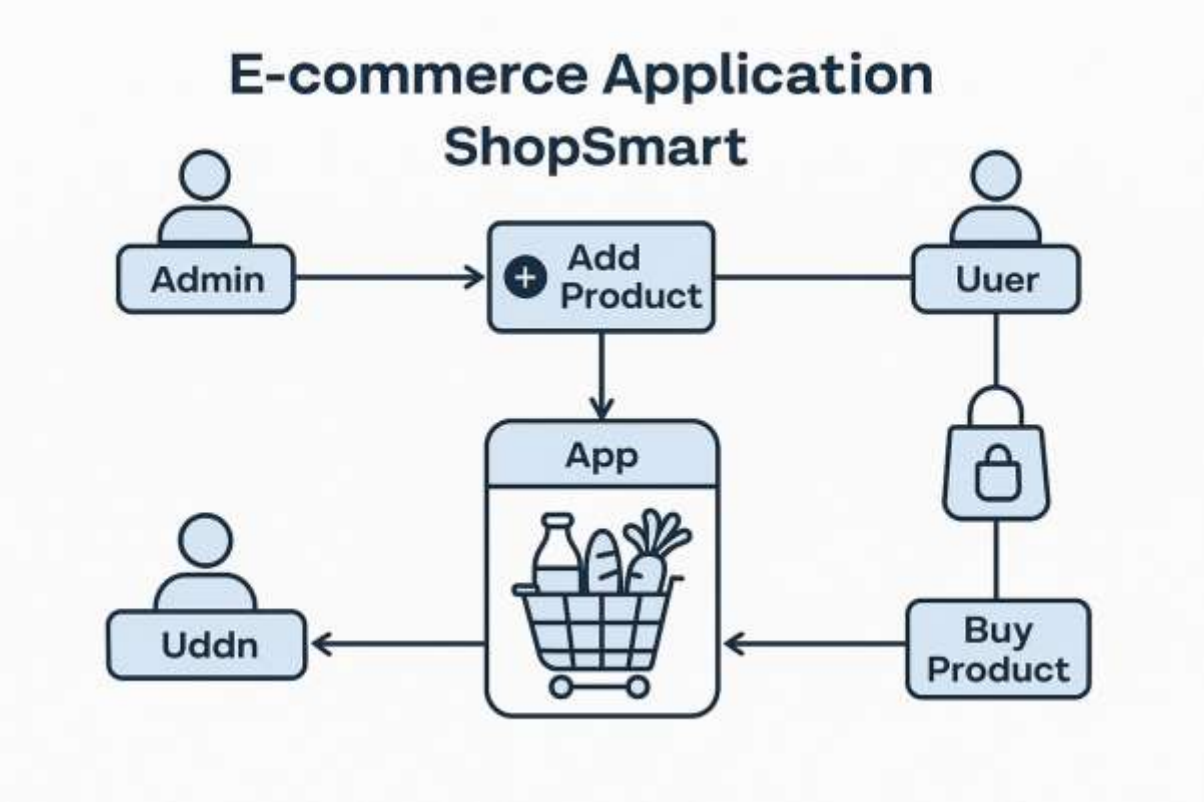
3.3 Data Flow Diagram (DFD)

The DFD represents the flow of information within the system. It shows how users (customers, admins, sellers) interact with different system modules.

Level 0 DFD Summary:

- **User Inputs:** Login, product search, checkout
- **Processes:** Authentication, product listing, payment processing

- **Data Stores:** Users, products, orders
- **Outputs:** Product display, order confirmation, user feedback



This diagram helps visualize the data lifecycle, ensuring that all interactions are efficiently handled.

3.4 Technology Stack

Technical Architecture

ShopSmart uses a **3-tier scalable architecture**:

1. **Presentation Layer (Frontend):** Web-based interface with a focus on responsive design.
2. **Business Logic (Backend):** Handles authentication, product operations, admin controls.
3. **Data Storage Layer:** Manages user, order, and product data securely.

Component Overview

S.No	Component	Description	Technology
1	**User Interface**	Interface for users and admins	HTML, CSS, JavaScript, React.js

2	**Application Logic-1**	Product browsing & interaction logic	Node.js, Express.js
S.No	**Component**	**Description**	**Technology**
3	**Application Logic-2**	Admin panel functionality	React.js, Node.js
4	**Database**	Persistent data storage	MongoDB

Application Characteristics

S.No	Characteristics	Description	Technology
5	**Open-Source Frameworks**	Web development frameworks	React.js, Node.js, Bootstrap, Tailwind CSS
6	**Scalable Architecture**	RESTful services and microservice readiness	Node.js-based microservices

References:

- React.js Documentation
- https://nodejs.org/en/learn/getting-started/introduction-to-nodejs
- Technical Architecture Guide

4. PROJECT DESIGN

4.1 Problem–Solution Fit

The **Problem–Solution Fit** confirms that the **ShopSmart** platform is well-suited to address common issues faced by traditional shoppers, especially the elderly, by offering a secure, efficient, and user-centric online grocery experience.

Purpose:

- Build a quick-commerce platform enabling individuals and organizations to conveniently purchase grocery products.
- Provide wishlist management for saving favorite products.
- Enable real-time updates and automated notifications to enhance user engagement.
- Empower administrators with analytics, product tracking, and inventory monitoring.

- Ensure secure payment gateways to build trust and protect customer data.

Identified Problems:

- Lack of transparency and order updates.
- Delayed response and unclear resolution times.
- Difficulty tracking recurring or high-priority product/service issues.
- Absence of data insights to improve shopping efficiency and customer satisfaction.

4.2 Proposed Solution

ShopSmart, a full-stack e-commerce grocery application, provides the following core solutions:

- **Seamless Shopping Experience:** A smooth and intuitive interface for product browsing and ordering.
- **Wide Product Range:** Access to fresh produce, pantry items, dairy, and daily-use essentials.
- **User Authentication:** Role-based sign-up/login system for customers and admins.
- **Wishlist Management:** Users can save products for future purchases.
- **Admin Dashboard:** Allows administrators to manage inventory, product listings, user data, and order analytics.
- **Secure Payment Gateway:** Integrated payment system with encryption and fraud prevention.
- **Real-Time Engagement:** Status updates and alerts for orders, deliveries, and payments. **Solution Overview Table**

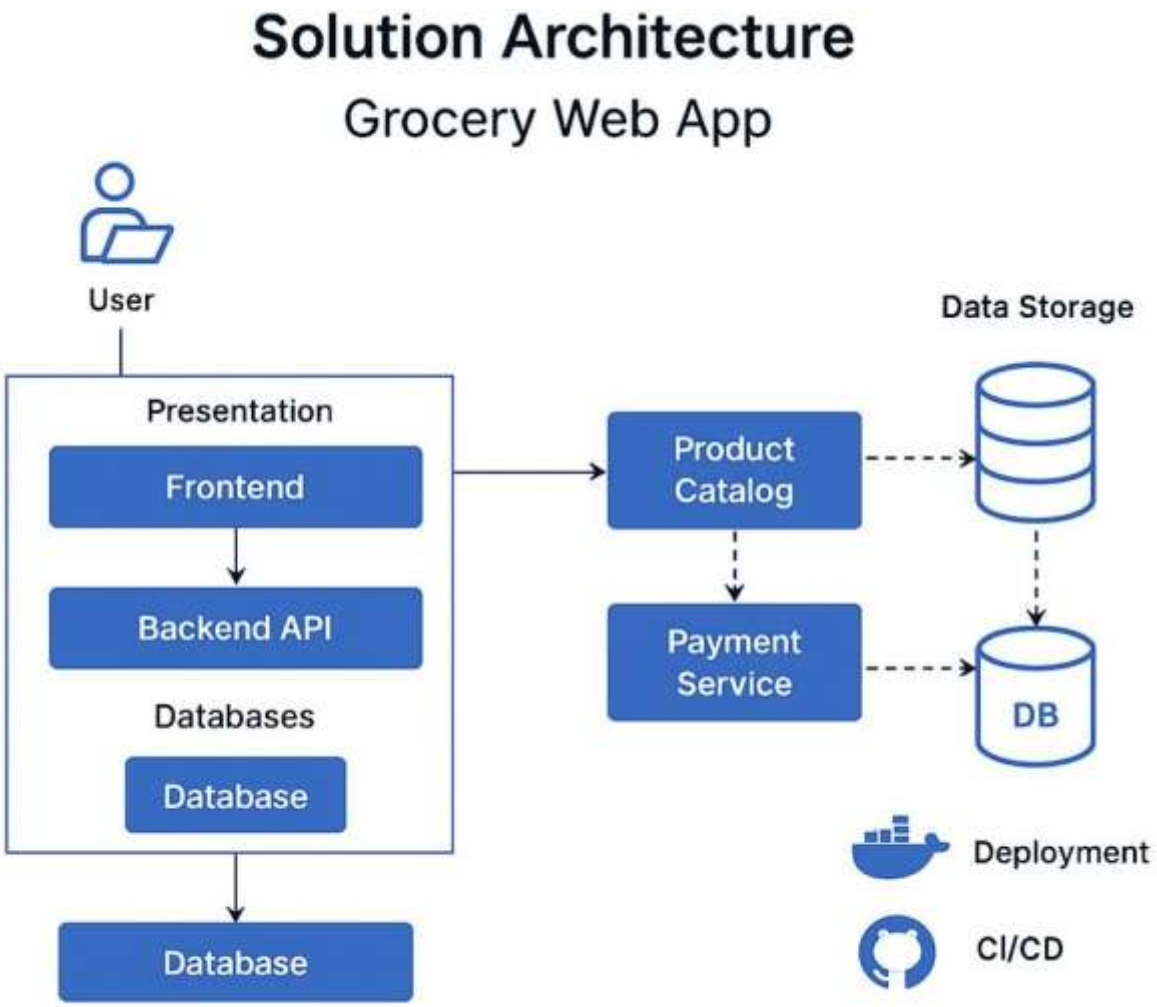
S. No	Parameter	Description
1	**Problem Statement**	In a fast-paced world, users prefer quick, reliable platforms for shopping. ShopSmart addresses this need with a streamlined experience.
2	**Solution Description**	A full-stack grocery web app (React + Node.js) that enables

		browsing, wishlist, secure payment, and admin control.
3	**Novelty / Uniqueness**	- Admin-driven product listing - Secure payment integration -
4	**Social Impact / Customer Value**	- Saves time and effort - Easy access to essentials - Transparent reviews - Enhances digital literacy for older adults Centralized admin dispute system

S. No	Parameter	Description
5	**Business Model (Revenue)**	Freemium user access with potential future integration of vendor partnerships or premium services.

4.3 Solution Architecture

The **ShopSmart** platform is designed using a scalable and modular architecture that supports high performance and a growing user base.



Key Architectural Features:

- **Frontend (Presentation Layer):** Built using React.js, the UI supports product browsing, cart/wishlist management, reviews, and secure checkout.
- **Backend (Application Logic):** Node.js and Express handle user authentication, product operations, order processing, and admin control panels.
- **Database Layer:** MongoDB stores user profiles, product catalogs, order histories, and feedback securely.
- **Security Layer:** Implements role-based access control, encrypted transactions, and token-based authentication (JWT).
- **Responsive Design:** Ensures mobile and desktop usability with features like wishlists, reviews, and dynamic carts.

Highlighted Capabilities:

- End-to-end management of product lifecycle and user interaction
- Seamless cart and wishlist functionality
- Scalable and secure login system
- Trackable and encrypted transactions via integrated payment services

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

To ensure structured development, the project followed the Agile methodology with sprintbased planning, team collaboration, and iterative delivery. The following components detail the planning, scheduling, and estimation efforts.

Product Backlog and Sprint Schedule

Sprint	Functional Requirement (Epic)	User Story ID	User Story / Task	Story Points	Priority	Team Member
Sprint 1	User Authentication	USN-1	As a user, I can sign up and log in securely.	3	High	Lakshmi
		USN-2	As a user, I can reset my password.	2	Medium	Jayasree
Sprint 2	Product Management (Admin)	USN-3	As an admin, I can add and delete grocery products.	2	High	Harsha Vardhan
Sprint 3	Product Browsing & Cart	USN-4	As a user, I can browse products and view details.	3	High	Jyothika

Sprint	Functional Requirement (Epic)	User Story ID	User Story / Task	Story Points	Priority	Team Member
			As a user, I can add			

			USN-5	As a user, I can add products to my cart and update quantities.	2	High	Harsha Vardhan
			USN-6	As a user, I can lodge a complaint.	2	Medium	Jayasree
Sprint 4	Checkout & Order Management		USN-7	As a user, I can complete purchases through checkout and make payment.	3	High	Lakshmi
			USN-8	As a user, I can view my order history.	2	Medium	Harsha Vardhan
			USN-9	As an admin, I can view and manage all orders.	2	Medium	Lakshmi

---

Project Tracker, Velocity & Burndown Summary

	Total Story Points	Duration	Start Date	Sprint End Date	Planned Story Points	Completed	Actual Release Date
Sprint 1	20	6 Days	20 June 2025	21 June 2025	20		21 June 2025
Sprint 2	20	6 Days	22 June 2025	22 June 2025	20		22 June 2025
Sprint 3	20	6 Days	23 June 2025	24 June 2025	20		24 June 2025
Sprint 4	20	6 Days	24 June 2025	24 June 2025	20		24 June 2025

Sprint Strategy Summary

- **Methodology Used:** Agile with 4 sprints
- **Sprint Duration:** 6 days per sprint
- **Tools Used:** Manual tracking with Excel (can be substituted with Trello, Jira in production environments)
- **Velocity:** Consistent delivery of 20 story points per sprint

This structured planning allowed timely completion of all major functional modules with clear ownership and prioritization.

6. FUNCTIONAL AND PERFORMANCE TESTING

Project Overview

- **Project Name:** ShopSmart: Your Digital Grocery Store Experience
- **Version:** v1.0.0
- **Description:** A full-stack grocery e-commerce platform enabling customers to browse, wishlist, and purchase grocery items. Admins manage inventory and monitor user orders.
- **Frontend Framework:** Angular 16.0.0
- **Backend:** Node.js with Express, MongoDB
- **Testing Period:** 26 May 2025 – 2 June 2025

Testing Scope

The testing phase focused on validating core functionalities and ensuring platform stability across user and admin roles. The following areas were covered:

- User registration and login
- Product browsing and cart management
- Checkout and secure payment processing
- Order history viewing
- Admin functionalities (product and order management)

Requirements to Be Tested

- **User Requirements:**
    - Register, log in securely, and purchase groceries ○
      Add items to the cart and complete transactions
- **Admin Requirements:**
    - Add, edit, and delete grocery products ○
      View and manage all user orders

Testing Environment

- **Testing URL:** https://drive.google.com/file/d/1fVxVPUTlEW3ovT0Jd_tdfvGPMki7mpj/view?usp=drive_link

---

Test Cases

Test Case	Test Scenario ID	Test Steps	Expected Result	Actual Result	Status
TC-001	User Registration	1. Visit site 2. Click "Sign Up" 3. Fill registration form and submit	Account is created and user is redirected to dashboard	User successfully registered and redirected	✅ Pass
TC-002	Add to Cart	1. Login as user 2. Browse products 3. Click “Add to cart”	Item added and reflected correctly in the cart	Product is added to cart	✅ Pass

---

Bug Tracking

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Error on job posting form	1. Login as client 2. Submit empty form	Medium	Open	Form should show appropriate validation message
BG-002	Cart not updating item quantity	1. Add item 2. Click “+” to increase quantity in cart preview	High	Open	Quantity not updating in preview

---

Sign-off

- **Tester Name:** Shaun Rohi

- **Date:** 26 May 2025
- **Signature:** *Shaun Rohi* **Notes:**
- All test cases were validated for both positive and negative input scenarios.
- Detailed tester feedback was encouraged for continuous improvement.
- Bugs are tracked by severity and include reproduction steps for efficient debugging.
- Final deployment requires sign-off from both the project manager and product owner.

7. RESULTS

This section highlights the successful implementation of core features and demonstrates the final output of the **ShopSmart** web application through key UI screens.

7.1 Output Screenshots

Below are the major output screenshots showcasing key functionalities of the application:

---

1. Home Page

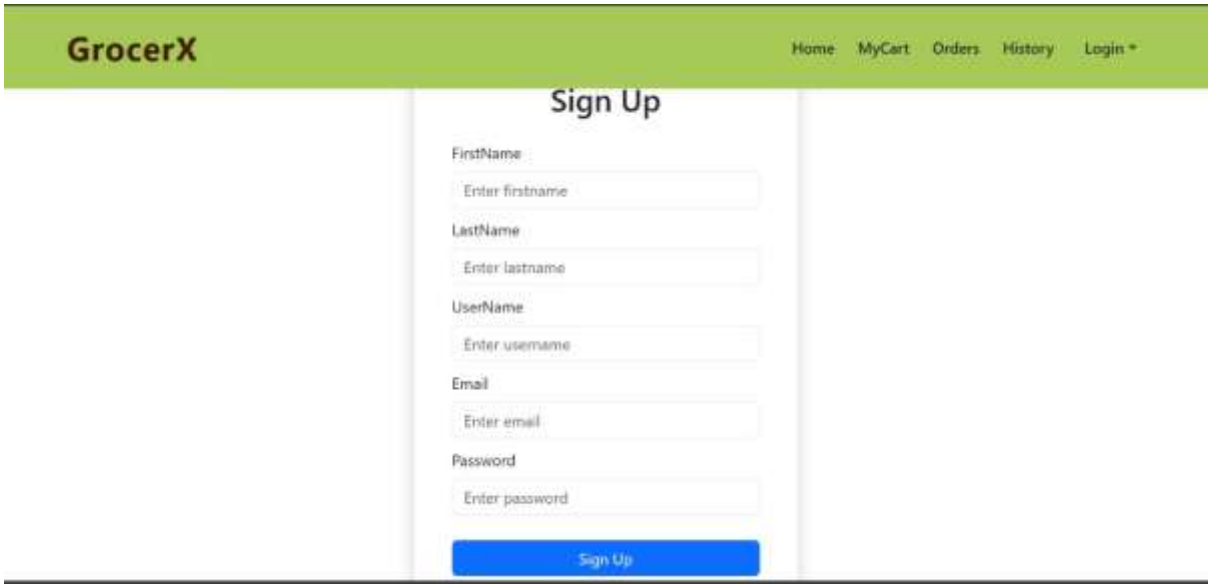
Description:
Displays a clean layout of product categories, promotional banners, and navigation menu for login, cart, and search.
Screenshot:

---

2. User Registration / Login

Description:
User registration form and login interface with email/password and OAuth support (Google/GitHub).

Screenshot:

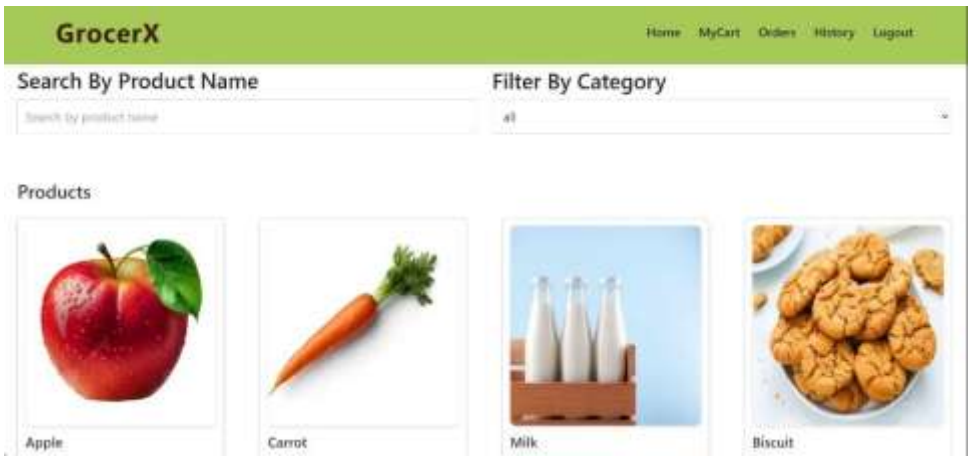


---

3. Product Detail Page

Description:
Displays detailed information about a selected product, including price, description, and "Add to Cart" option.

Screenshot:



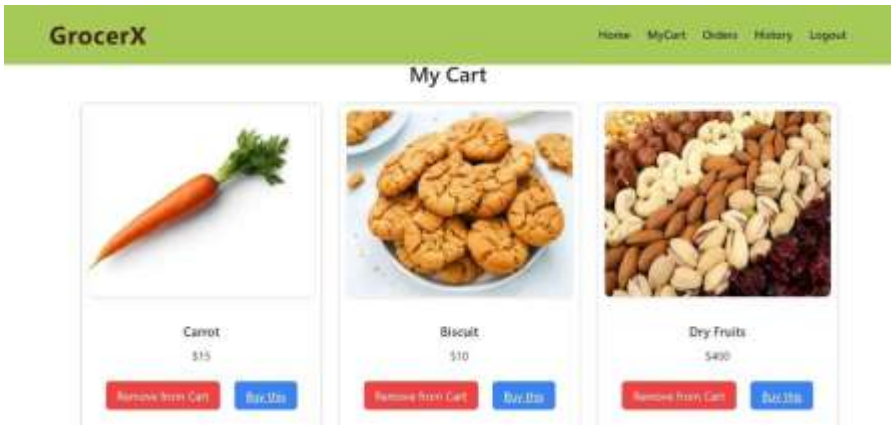
---

4. Shopping Cart

Description:

Users can view items in their cart, adjust quantities, and proceed to checkout.

Screenshot:

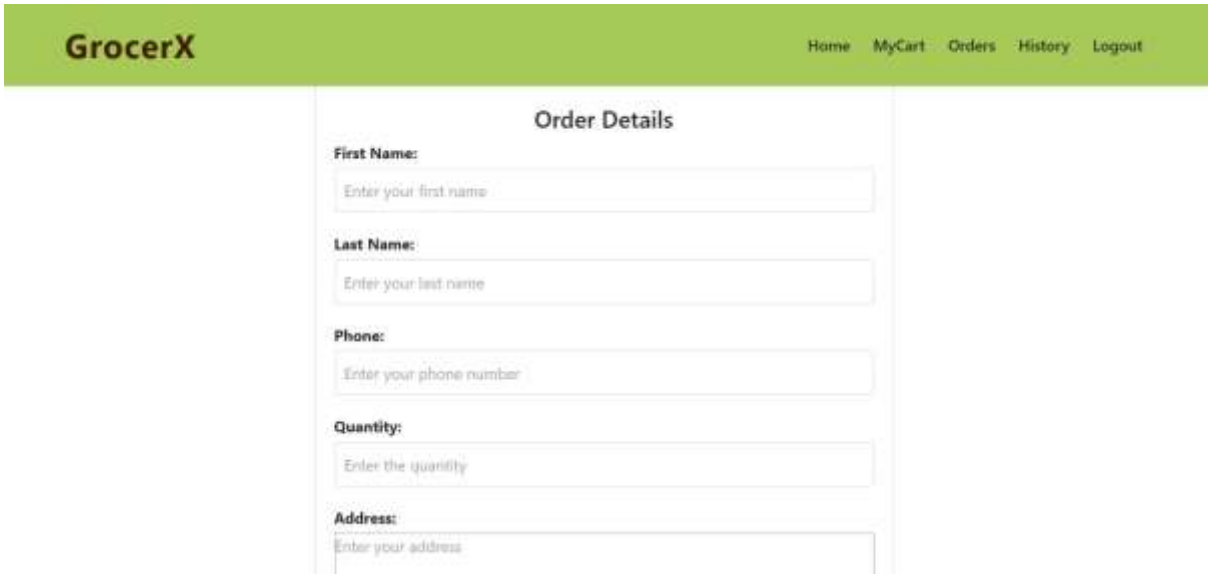


---

5. Checkout & Payment

Description:
Checkout interface with order summary and secure payment processing.

Screenshot:



---

6. Order History

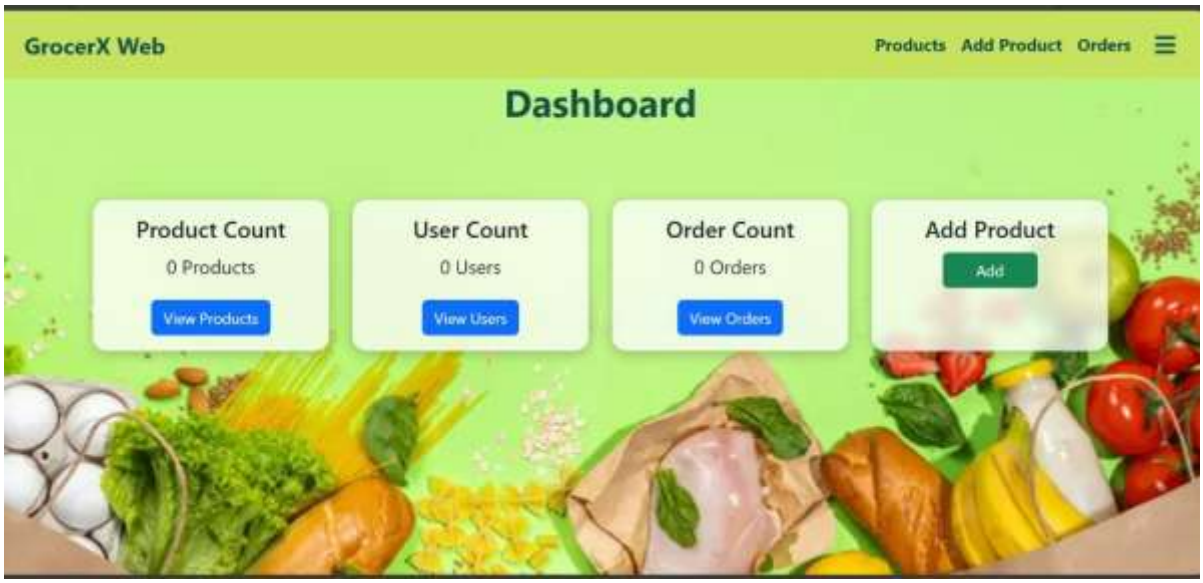
Description:
Shows a list of past orders placed by the user with date, status, and product summary.
Screenshot:

---

7. Admin Dashboard

Description:
Admin panel for managing products, viewing orders, and performing CRUD operations.

Screenshot:

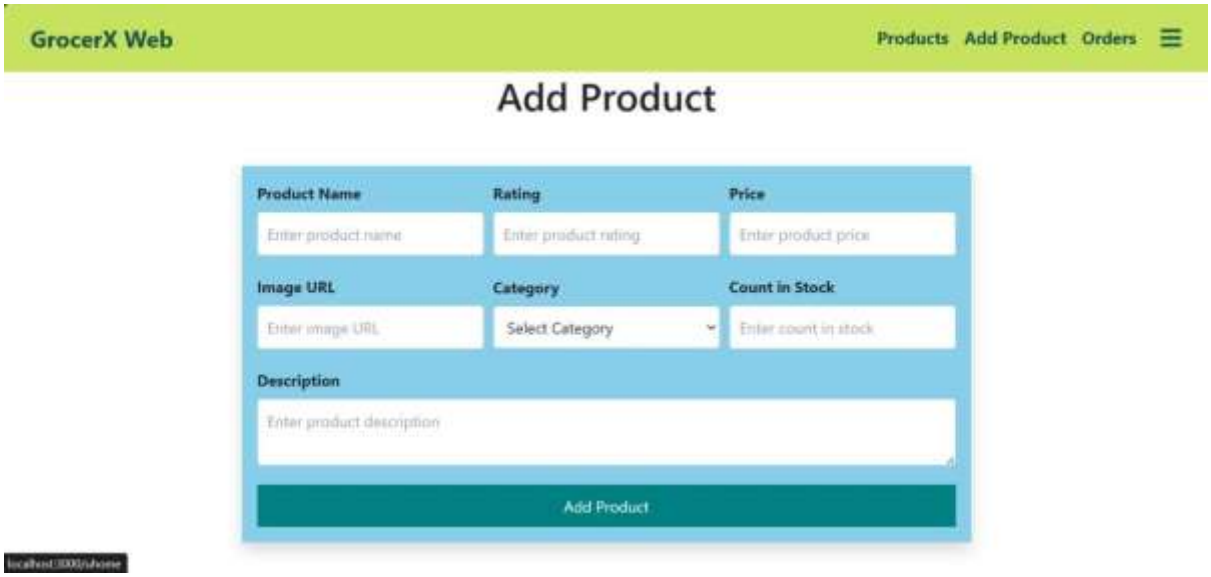


---

8. Admin's Add Products Page

Description:
Admin can add new products with details like name, price, category, stock, and image. Ensures real-time update to the product list.

Screenshot:



---

9. Admin Orders Page

Description:
Displays all user orders for admin review and management. Supports status updates and tracking.

Screenshot:

8. ADVANTAGES & DISADVANTAGES

Advantages

1. **User-Friendly Interface**
   The application provides an intuitive UI that caters to users of all ages, including elderly individuals who may be less tech-savvy.
2. **Secure Transactions**
   Integrated secure payment gateway with role-based access ensures data privacy and transaction security.
3. **Convenient Shopping Experience**
   Users can shop from the comfort of their homes, browse by category, and get realtime product availability.
4. **Admin Control Panel**
   Admins have full control over product listings and order management, streamlining backend operations.
5. **Scalable Architecture**
   Built using modern full-stack technologies (Angular, Node.js, MongoDB), the app is scalable for future enhancements.
6. **Wishlist Functionality**
   Enables users to save favorite items for later, improving engagement and return visits.

Disadvantages

1. **Internet Dependency**
   Requires a stable internet connection for both users and admins, which may not be accessible to all users.
2. **Limited Product Range Initially**
   As it's an MVP (Minimum Viable Product), the initial launch may feature a limited product catalog.
3. **No Mobile App Version Yet**
   Although mobile responsive, a native mobile app could improve usability and performance on mobile devices.
4. **Manual Inventory Updates (in MVP)**
   Currently, product stock updates are manual and may require automation in future versions.

9. CONCLUSION

The **ShopSmart** GrocerX web application successfully delivers a seamless and secure digital shopping experience tailored for modern consumers. By addressing common pain points in traditional grocery shopping—such as long queues, product unavailability, and limited store hours—ShopSmart empowers users to browse, wishlist, and purchase essentials with just a few clicks.

The platform effectively supports both end-users and administrators with key features such as user authentication, cart and checkout functionality, product management, and order tracking. It also emphasizes usability, performance, and data security, ensuring trust and satisfaction for all stakeholders.

Through structured sprint planning, collaborative development, and continuous testing, the project achieved all planned milestones within the scheduled timeframe. ShopSmart stands as a scalable and expandable solution, paving the way for digital transformation in local and online retail.

10. FUTURE SCOPE

The current version of **ShopSmart** lays a solid foundation for digital grocery shopping. However, to enhance its functionality, user experience, and scalability, the following improvements and expansions are planned for future versions:

1. Mobile Application Development

- Launch dedicated Android and iOS mobile apps for a more native experience.
- Enable push notifications for order updates and promotions.

2. AI-Based Product Recommendations

- Implement machine learning algorithms to offer personalized product suggestions based on user behavior and preferences.

3. Real-Time Inventory Sync

- Automate stock updates using IoT-enabled systems or vendor integrations for realtime inventory tracking.

4. Multilingual Support

- Add support for regional languages to improve accessibility for non-English speaking users.

5. Enhanced Admin Analytics

- Provide detailed analytics dashboards for admins including product performance, sales trends, and customer insights.

6. Delivery Partner Integration

- Integrate APIs from delivery services to enable live order tracking and route optimization.

7. Subscription Services

- Introduce subscription models for recurring purchases like milk, bread, or vegetables.

8. Chat Support / Chatbot

- Add real-time customer support via chatbot to resolve user queries instantly.

11. APPENDIX

This section provides additional resources and references related to the **ShopSmart** project.

Source Code

- **GitHub Repository:** https://github.com/Shaun67man/Shop-smart.git

---

Dataset

- **Product Dataset:**
  Manually created and entered product list for grocery items used during development and testing.

Video Demonstration

- **Demo Video (YouTube / Google Drive):**
  https://drive.google.com/file/d/1VKRh8EfpnCsY7m95Uo-6_KWP1GiP29SK/view

---

References

- Angular & React Official Docs
- Node.js & Express.js Documentation
- MongoDB Design Guidelines
- JWT Authentication Reference
- Technical Architecture Guide: https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecturediagrams-2d20c9fda90d