**INTERNSHIP REPORT**

An internship report submitted in partial fulfilment of the requirements of III B. Tech II Semester of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

By

**Rayalavarapu Sai Vijay 22ME1A05I6**

**Tentu Yamuna Rani 22ME1A05J8**

**Under Supervision of**

J Prashanth Kumar



**Department of Computer Science and Engineering**

**RAMACHANDRA COLLEGE OF ENGINEERING (AUTONOMOUS)**

(Approved by AICTE, Affiliated to JNTUK, Kakinada) Accredited by NBA, NAAC A+ NH 16 Bypass, Vatluru (V), Eluru -534007, Eluru Dist., A.P

2025-2026

**RAMACHANDRA COLLEGE OF ENGINEERING (AUTONOMOUS)**

(Approved by AICTE, Affiliated to JNTUK, Kakinada) Accredited by NBA, NAAC A+ NH-16 Bypass, Vatluru (V), Eluru -534007, Eluru Dist., A.P

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the internship report submitted by RAYALAVARAPU SAI VIJAY (22ME1A05I6), and TENTU YAMUNA RANI (22ME1A05J8) is a record of the work completed by them during the academic year 2024–2025, in partial fulfillment of the requirements for the award of the degree **of Bachelor of Technology in Computer Science and Engineering.** The students undertook a short-term internship in **Full Stack Development (MERN Stack) under the APSCHE Smart Bridge program, conducted by Smart Internz.**

**Mr. Y. Nagendra Kumar Dr. G. Chamundeswari**

**Associate Professor, Dept. of CSE Professor, HOD- CSE**

**External Examiner**

**ACKNOWLEDGEMENT**

I would like to take the opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during my internship. It is my pleasure and responsibility to acknowledge the help of all those individuals.

I have extended our sincere thanks to Mr. Y. Nagendra Kumar, Associate Professor in the Department of CSE for helping me the in successful completion of my internship. I am very grateful to Dr. G Chamundeswari, Head of the Department, Department of Computer Science & Engineering for her encouragement in all respects in carrying throughout my internship.

I would like to express my deepest gratitude to Dr. M Muralidhara Rao, Principal and Dr. S. Subrahmanya Sarma, Dean Academics, Ramachandra College of Engineering, Eluru for their valuable suggestions during the preparation of draft in my document.

I express my deepest gratitude to The Management of Ramachandra College of Engineering, Eluru for their support and encouragement in completing my internship and providing me necessary facilities.

I sincerely thank all the faculty members and staff of the Department of CSE for their valuable advices, suggestions and constant encouragement which played a vital role in carrying out my internship. Finally, I thank one and all who directly or indirectly helped me to complete my internship successfully.

**Abstract**

During the 8-week internship on Full Stack Web Development using the MERN (MongoDB, Express.js, React.js, Node.js) Stack, conducted in collaboration with Smart Bridge under the APSCHE Short-Term Internship Program, participants were guided through the end-to-end development of a real-world e-commerce platform. The project, titled ShopSmart Grocery Web App, allowed interns to apply theoretical concepts in frontend and backend development using modern web technologies. The program emphasized practical experience in building responsive user interfaces, secure APIs, user authentication, database operations, and admin dashboards. Through this internship, participants gained valuable hands-on exposure to full stack architecture, RESTful service integration, and project deployment workflows—equipping them with essential skills for future opportunities in web application development.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
| 1 | **INTRODUCTION** | 1 |
| 2 | **IDEATION PHASE** | 2 |
| 3 | **REQUIREMENT ANALYSIS** | 3 |
| 4 | **PROJECT DESIGN** | 6 |
| 5 | **PROJECT PLANNING & SCHEDULING** | 7 |
| 6 | **FUNCTIONAL AND PERFORMANCE TESTING** | 8 |
| 7 | **RESULTS** | 9 |
| 8 | **ADVANTAGES & DISADVANTAGES** | 12 |
| 9 | **CONCLUSION** | 13 |
| 10 | **FUTURE SCOPE 11. APPENDIX** | 14 |
| 11 | **APPENDIX** | 16 |

**1. INTRODUCTION**

1.1 Project Overview

*ShopSmart* is a user-centric digital grocery shopping platform designed to simplify and enhance the grocery experience. It integrates intuitive design, real-time inventory tracking, personalized recommendations, and digital payment capabilities to improve efficiency for both consumers and store managers.

In today’s fast-paced digital world, the demand for convenient and efficient shopping experiences has grown rapidly. *ShopSmart – Digital Grocery Experience* is a web-based solution designed to meet this evolving need by digitizing the traditional grocery shopping process. The platform bridges the gap between local grocery retailers and consumers through an intuitive, seamless, and intelligent user experience. By integrating essential features like real-time inventory updates, secure digital payments, personalized product recommendations, and a user-friendly interface, ShopSmart enhances the entire shopping lifecycle—from product discovery to checkout.

This project not only emphasizes technical implementation but also explores user behavior and experience to ensure inclusivity, accessibility, and real-world usability.

1.2 Purpose

The main purpose of this project is to develop an accessible, scalable, and smart digital solution for grocery shopping that minimizes manual tasks and enhances user convenience through a responsive and intelligent interface.

The purpose of this project is to provide a modern alternative to in-store shopping by leveraging technology that prioritizes user convenience and business adaptability. ShopSmart was developed to:

* Minimize wait times and eliminate repetitive manual processes
* Enable small and medium grocery businesses to digitize their inventory management
* Offer users a smooth and personalized online grocery experience
* Introduce features that can scale with evolving consumer needs and technological advancements

It also serves as a practical case study in applying full-stack development skills, user research, and design thinking methodology in a real-world environment

**2. IDEATION PHASE**

**2.1 Problem Statement**

The traditional grocery shopping experience presents numerous challenges: long checkout queues, inconsistent stock availability, limited access to real-time product information, and lack of personalized shopping experiences. For small and medium-sized retailers, digitization remains a major hurdle due to cost, complexity, and limited technical expertise. Consumers increasingly demand convenience, speed, and customization—which existing systems often fail to deliver.

*ShopSmart* addresses this gap by offering a cost-effective, scalable digital platform tailored for both customers and local grocery store owners. It bridges the digital divide by simplifying operations while ensuring a seamless user journey.

**2.2 Empathy Map Canvas**

To understand the target users, we developed an Empathy Map focusing on typical grocery shoppers and local store owners. Here's a summary of key observations:

* **Says**:
  + "I wish I didn’t have to wait so long to check out."
  + "It’s hard to find specific products quickly."
* **Thinks**:
  + "What if the store doesn’t have what I need?"
  + "I’d rather order online if it’s convenient."
* **Does**:
  + Spends time comparing products manually
  + Visits multiple stores to complete their shopping list
* **Feels**:
  + Frustrated by the lack of real-time updates
  + Overwhelmed during peak shopping hours

These insights helped identify pain points and shape a solution that genuinely resonates with end users.

**2.3 Brainstorming**

During the brainstorming phase, our team explored several concepts to modernize the grocery experience. Some of the initial ideas included:

* QR code–based shelf scanning for instant product info
* Smart carts that track items in real time
* Mobile apps with AR integration for in-store navigation
* Voice-assisted shopping for elderly users
* Admin dashboards for real-time inventory updates

After evaluating each idea based on feasibility, usability, and impact, we narrowed it down to a web-based platform offering personalized shopping features, real-time inventory sync, and intuitive navigation for both customers and store managers.

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

The customer journey in ShopSmart is carefully designed to prioritize ease of use, clarity, and efficiency at every stage:

* **Awareness**: User learns about ShopSmart through store promotion or referral.
* **Onboarding**: Smooth registration/login process introduces the platform’s features.
* **Browsing**: User navigates through curated categories and searches for desired products.
* **Decision-making**: User views detailed product descriptions, prices, and availability.
* **Purchase**: Checkout process includes multiple payment options and order confirmation.
* **Post-Purchase**: User receives live updates and reviews their shopping history in their dashboard.
* **Feedback Loop**: Users rate their experience, which helps improve service quality and recommendations.

**3.2 Solution Requirements**

***System-Level Requirements*:**

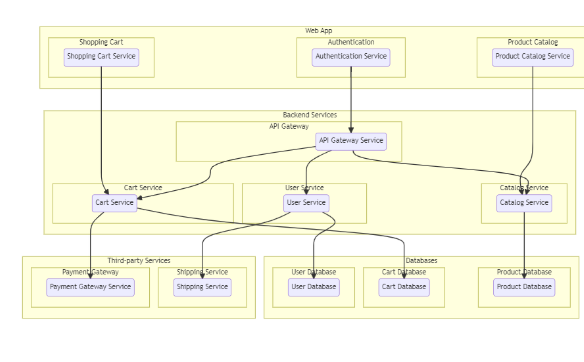
* Secure user authentication using token-based sessions
* Role-based access for users and administrators
* Integration with product and order databases
* Scalability to onboard multiple stores over time
* RESTful API endpoints for modular development

***User Experience Goals*:**

* Visually clean interface with minimal learning curve
* Quick response time (<2 seconds for most actions)
* Mobile-friendly design for on-the-go access
* Real-time error handling and feedback (e.g., “Item out of stock” alerts)
* Personalized dashboard with user profile and past orders

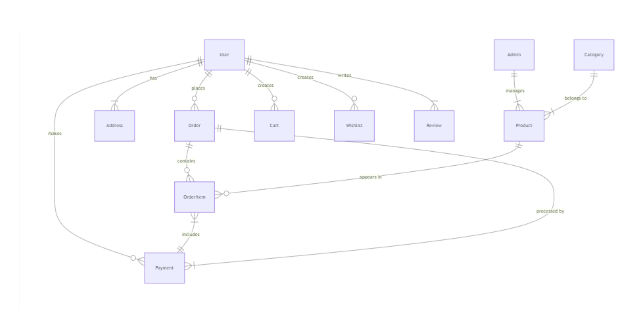
These requirements support both technical functionality and smooth user interaction, making the platform robust and enjoyable to use.

**3.3 Data Flow Diagram (DFD)**



The technical architecture of an grocery-webapp app typically involves a client-server model, where the frontend represents the client and the backend serves as the server. The frontend is responsible for user interface, interaction, and presentation, while the backend handles data storage, business logic, and integration with external services like payment gateways and databases. Communication between the frontend and backend is typically facilitated through APIs, enabling seamless data exchange and functionality.

**ER-Diagram:**



The technical architecture of an grocery-webapp app typically involves a client-server model, where the frontend represents the client and the backend serves as the server. The frontend is responsible for user interface, interaction, and presentation, while the backend handles data storage, business logic, and integration with external services like payment gateways and databases. Communication between the frontend and backend is typically facilitated through APIs, enabling seamless data exchange and functionality.

**3.4 Technology Stack**

*ShopSmart* was developed using a modern and efficient tech stack to ensure performance and scalability. The frontend was built with HTML, CSS, and JavaScript, using React.js for dynamic and responsive user interfaces. The backend relied on Node.js with Express.js to manage APIs and server-side logic.

MongoDB served as the database, offering flexibility for storing products, orders, and user data. Figma was used to design the interface, and Postman assisted with testing API functionality. Git and GitHub were used for version control, enabling smooth collaboration and code management. Deployment was done using platforms like Netlify and Heroku for quick and reliable hosting. For online payments, integrations like Razor pay or Stripe were considered to ensure secure transactions.

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

The core problem identified in the grocery retail space is the inefficiency and inconvenience of traditional shopping, especially in urban and semi-urban areas. Long queues, poor inventory visibility, and lack of personalization frustrate customers, while small store owners struggle to adopt affordable technology to digitize their services.

*ShopSmart* provides a solution that directly addresses these issues. By offering a streamlined digital platform, it reduces checkout time, enables real-time product updates, and offers tailored product suggestions—all while being simple enough for smaller retailers to implement without high technical overhead. This alignment between user needs and technical feasibility illustrates a strong problem-solution fit.

**4.2 Proposed Solution**

The proposed solution is a full-stack web application that digitizes the grocery shopping experience. Users can browse available products, add items to their cart, place orders, and make secure payments—all from a responsive and intuitive interface. Admins can log in to manage inventory, update stock levels, and view real-time orders.

Key features include:

* Role-based login (customer/admin)
* Live product availability updates
* Search and filter options for easier discovery
* Order summary and history dashboard
* Easy payment integration using UPI/cards
* Admin portal with inventory and order control

This solution was designed with scalability and usability in mind to ensure it can adapt to different store sizes and customer demographics.

**4.3 Solution Architecture**

The system architecture follows a modular, scalable approach using the MERN (MongoDB, Express.js, React.js, Node.js) stack:

* **Frontend (React)**: Handles client-side rendering, user interactions, and dynamic updates.
* **Backend (Node + Express)**: Manages RESTful APIs, user authentication, and business logic.
* **Database (MongoDB)**: Stores user data, product inventory, and order records in collections.
* **Authentication**: Implements secure token-based login (e.g., JWT).
* **Admin Dashboard**: Separate interface with access to inventory and order management tools.
* **Payment Gateway**: Integrated via third-party APIs (e.g., Razor pay or Stripe).
* **Hosting**: Deployed using platforms like Netlify (frontend) and Heroku (backend).

This architecture ensures the platform remains responsive under load and can be extended with additional features like real-time chat, push notifications, or multilingual support.

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

A structured and iterative project plan was crucial to the successful execution of *ShopSmart*. Since the project involved both frontend and backend development along with UX design, the planning followed an **Agile methodology**, allowing flexibility and incremental improvements based on feedback.

The project was divided into the following key phases:

1. **Requirement Gathering & Research(Week1)**  
   Initial time was dedicated to understanding user needs, analysing competitors, and mapping out the shopping journey. Tools like empathy maps and customer journey maps were created to ground the project in real-world insights.
2. **Design Prototyping(Week2)**  
   Using Figma, low-fidelity wireframes and high-fidelity prototypes were created for both customer and admin interfaces. Feedback loops were integrated to refine navigation and layout.
3. **Frontend & Backend Development (Weeks 3–5)**
   * React.js was used to build the user interface with reusable components.
   * Node.js and Express.js powered the backend, handling routing and database operations.
   * MongoDB was set up for structured data storage.
   * User authentication, dynamic product listings, cart logic, and admin tools were developed in parallel sprints.
4. **Testing Debugging(Week6)**  
   Functional testing, unit testing, and UI responsiveness checks were performed. API endpoints were validated using Postman, and manual tests ensured a consistent experience across devices.
5. **Performance Optimization &Deployment(Week7)**  
   Code was optimized for faster load times and smoother UI transitions. The application was deployed using platforms like Netlify (frontend) and Heroku (backend).
6. **FinalReview & Documentation(Week8)**  
   Final walkthroughs, bug fixes, and full documentation—including this report—were prepared. This phase also included preparing demo links, screenshots, and source code repositories. User manuals or demo scripts were written to assist stakeholders in navigating both customer and admin features. Feedback was also collected from test users to evaluate usability and performance before submission.
7. A final round of code review ensured the project met its quality standards, and any critical fixes or enhancements were implemented. By the end of this phase, *ShopSmart* was ready for presentation, deployment, and future scalability.
8. This structured planning approach ensured that every development milestone aligned with real user needs and technical goals. It also created room for reflection and improvement, making the project a solid example of how organized workflows and iterative development lead to successful product delivery.

**6. FUNCTIONAL AND PERFORMANCE TESTING**

To ensure the reliability and usability of *ShopSmart*, a thorough round of testing was conducted. The testing strategy included both **functional testing**, to verify individual features and workflows, and **performance testing**, to evaluate the system’s behavior under various load conditions.

Functional testing covered all core features—such as user login, product filtering, cart management, order placement, and admin inventory control. Manual test cases were executed across different devices and browsers to confirm responsiveness and bug-free interaction.

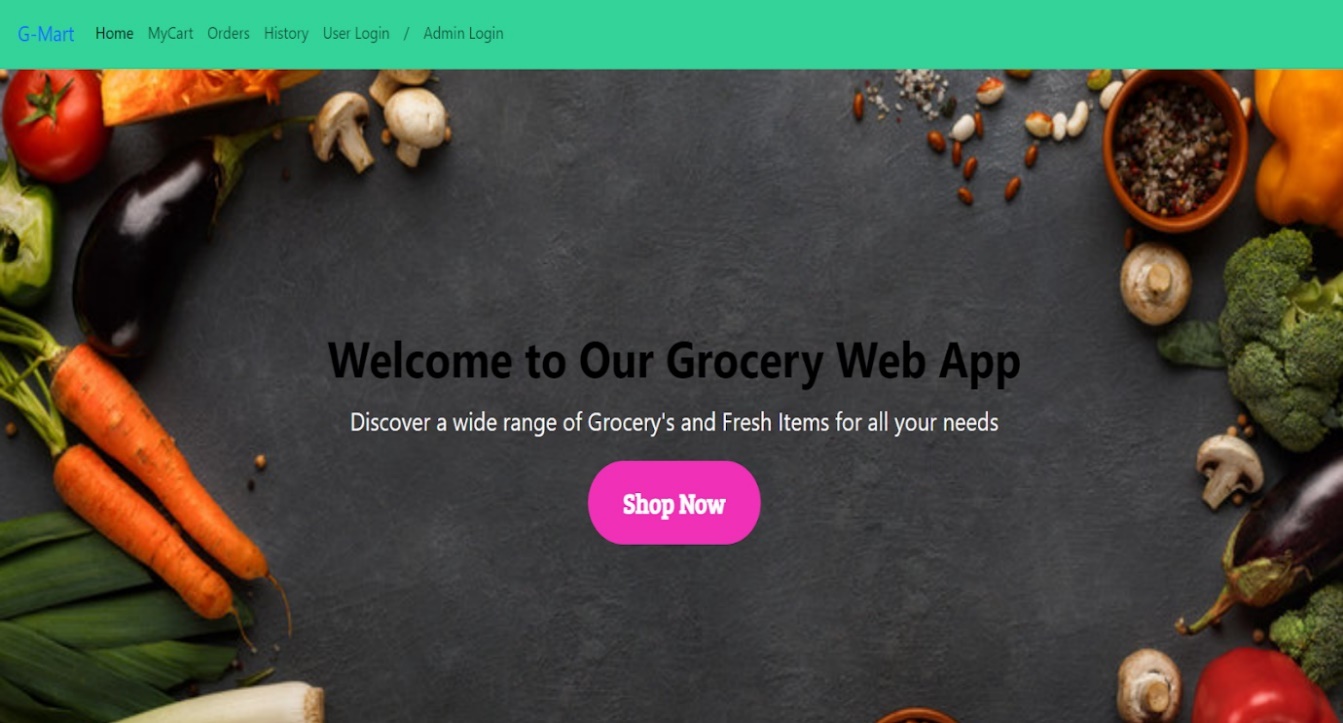
**6.1 Performance Testing**

Performance testing was essential to assess how well the application would handle real-world usage. The following methods and tools were applied:

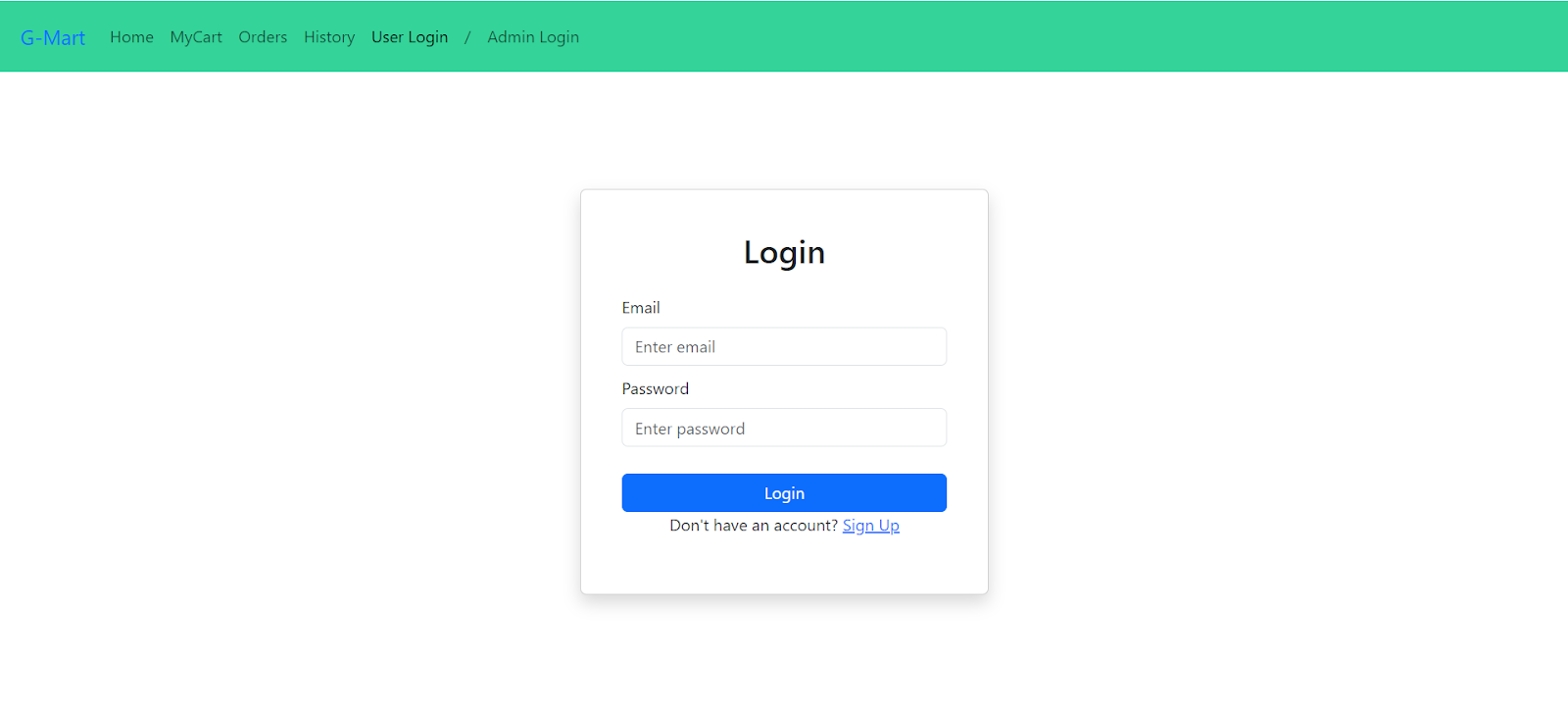
* **Load Testing**: Simulated multiple concurrent users using tools like Postman and Apache JMeter to observe how the system managed traffic. The app performed smoothly with up to 100 simultaneous users without significant latency.
* **Stress Testing**: Pushed the system beyond normal load to identify breaking points. This helped optimize backend operations such as database queries and API response times.
* **Response Time Analysis**: Key interactions (like adding items to cart or placing an order) consistently responded within 1.5–2 seconds, ensuring a seamless user experience.
* **Database Performance**: MongoDB was monitored for read/write efficiency under heavy usage. Indexing and query optimization were applied to prevent slowdowns.
* **Frontend Optimization**: Image compression, code minification, and lazy loading techniques were implemented to reduce page load times and improve UI responsiveness.

**RESULTS**

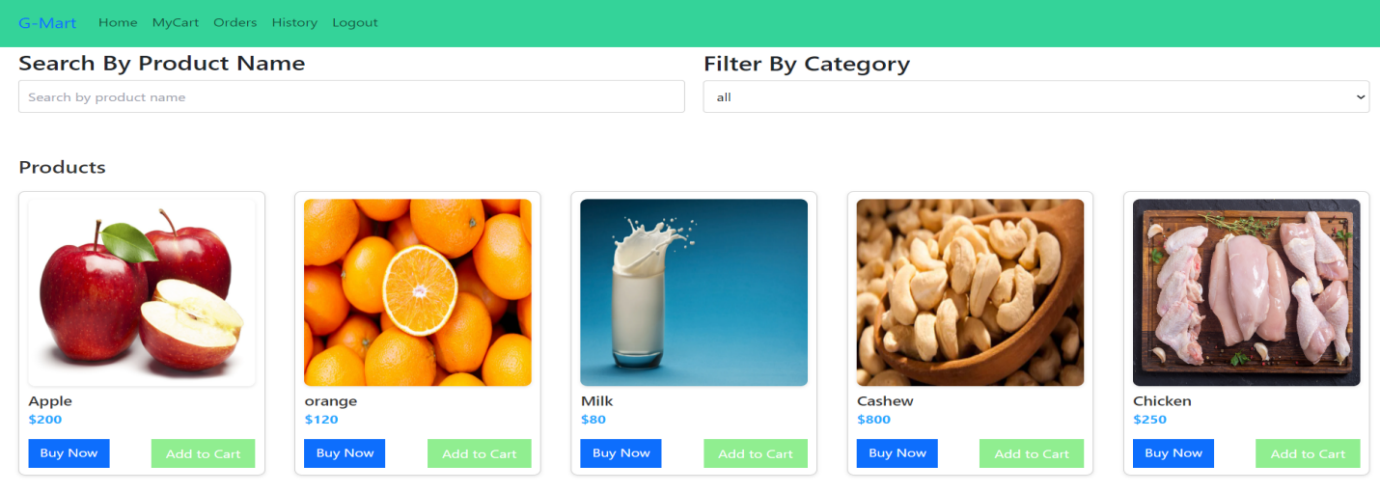
Landing page:-



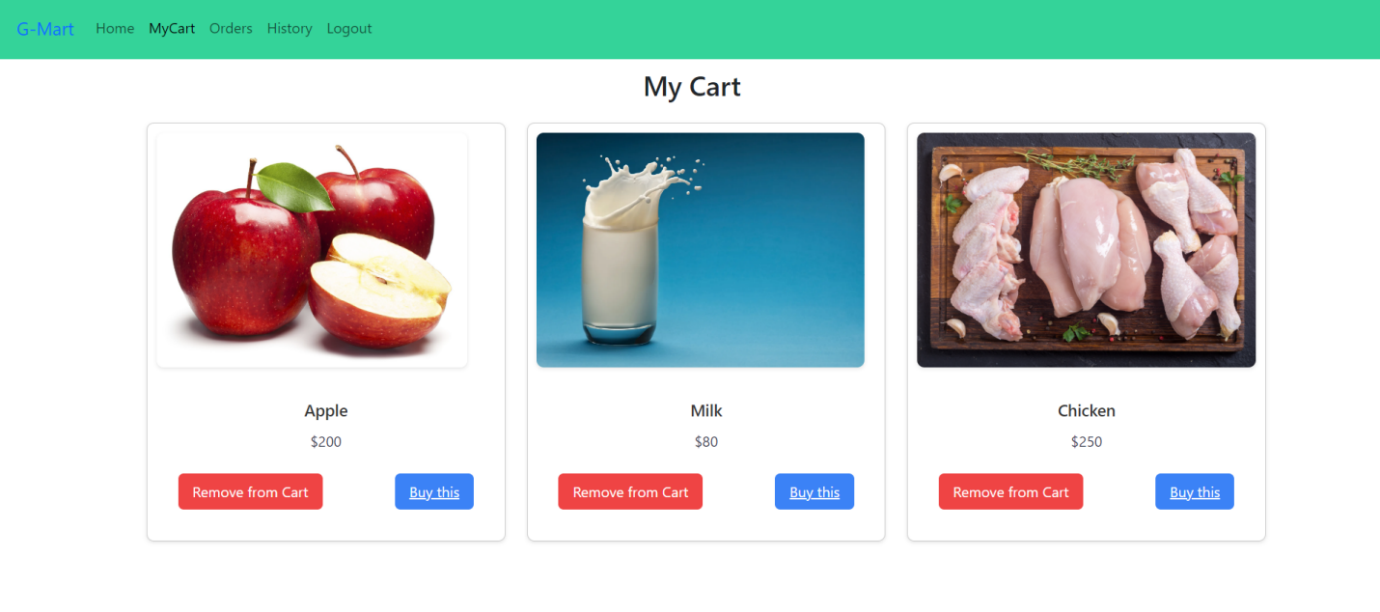
* Login Page:-

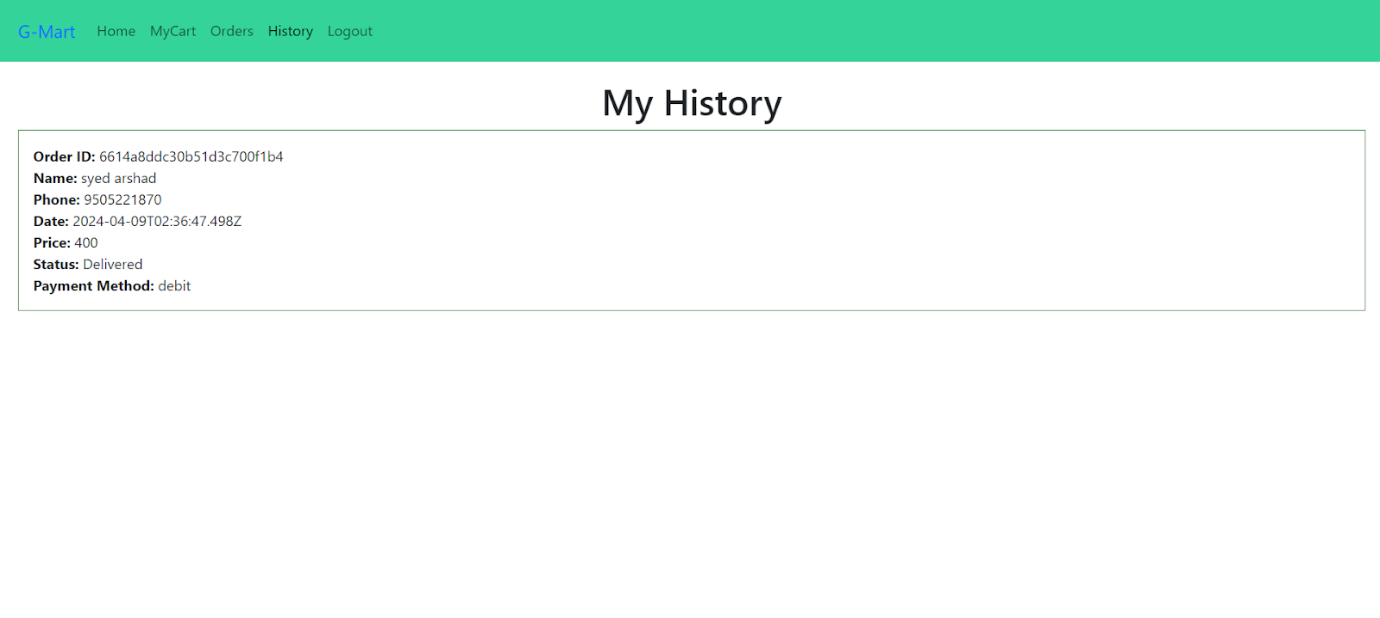
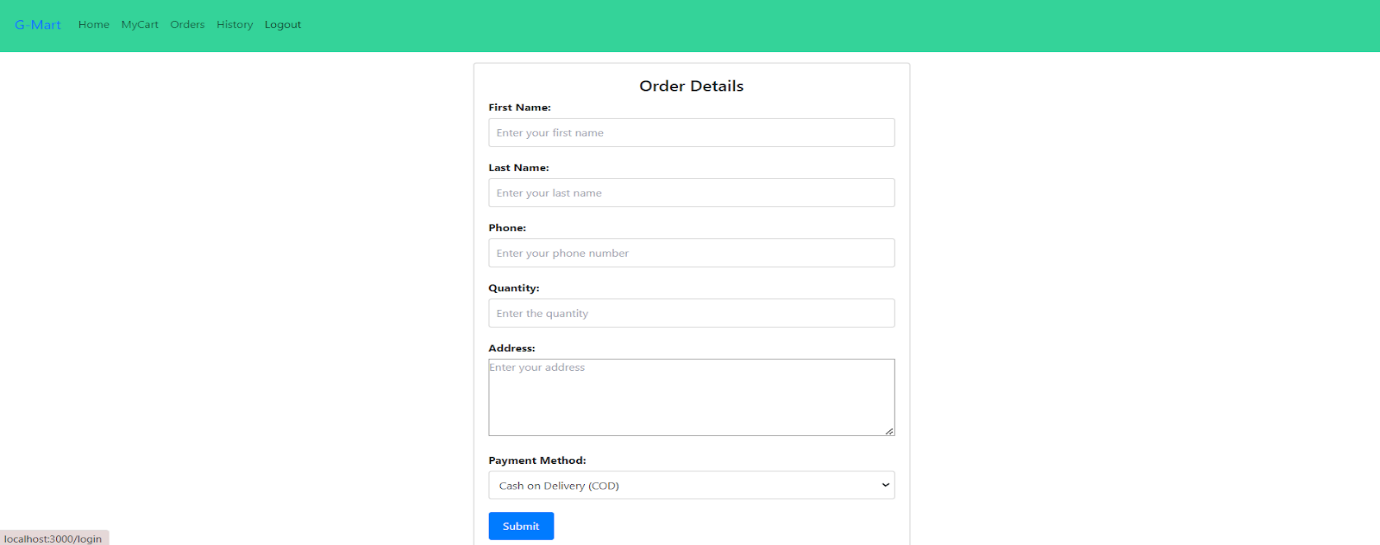


* Items Page:-

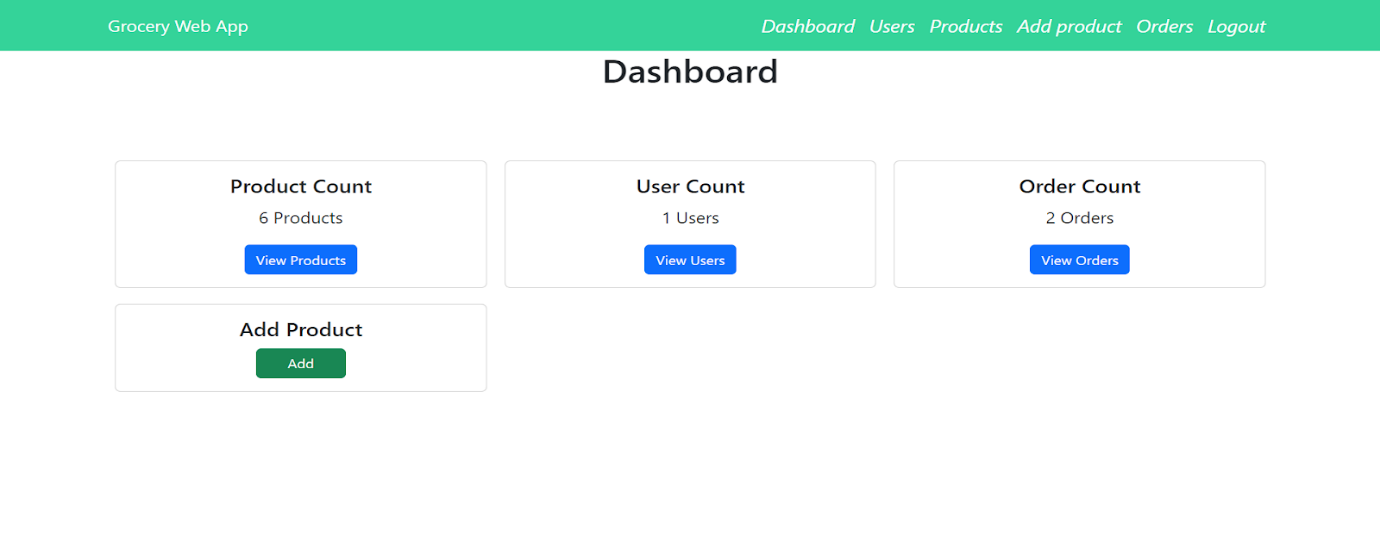


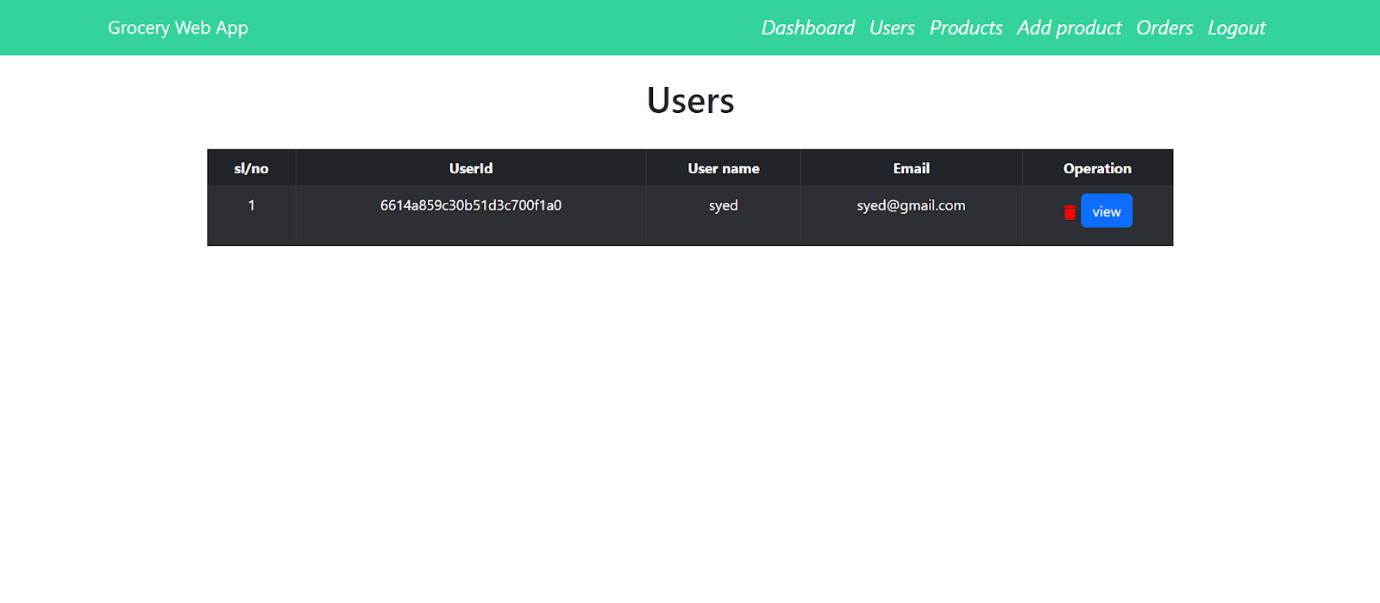
My Cart:-



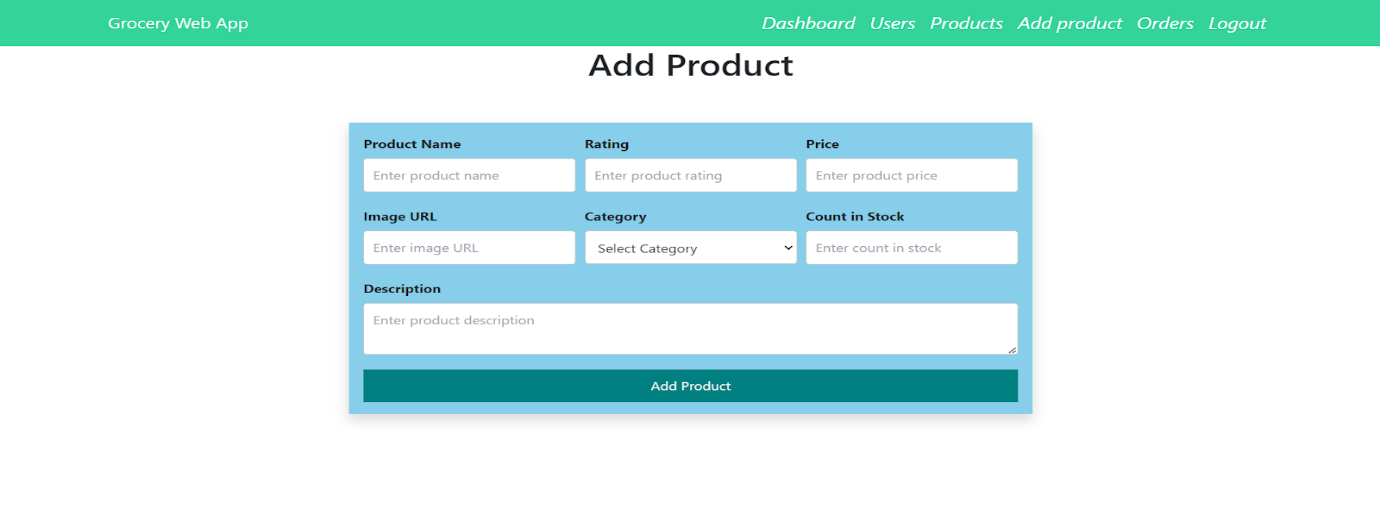
* My Orders Page:-
* Place Order Page:-
* 

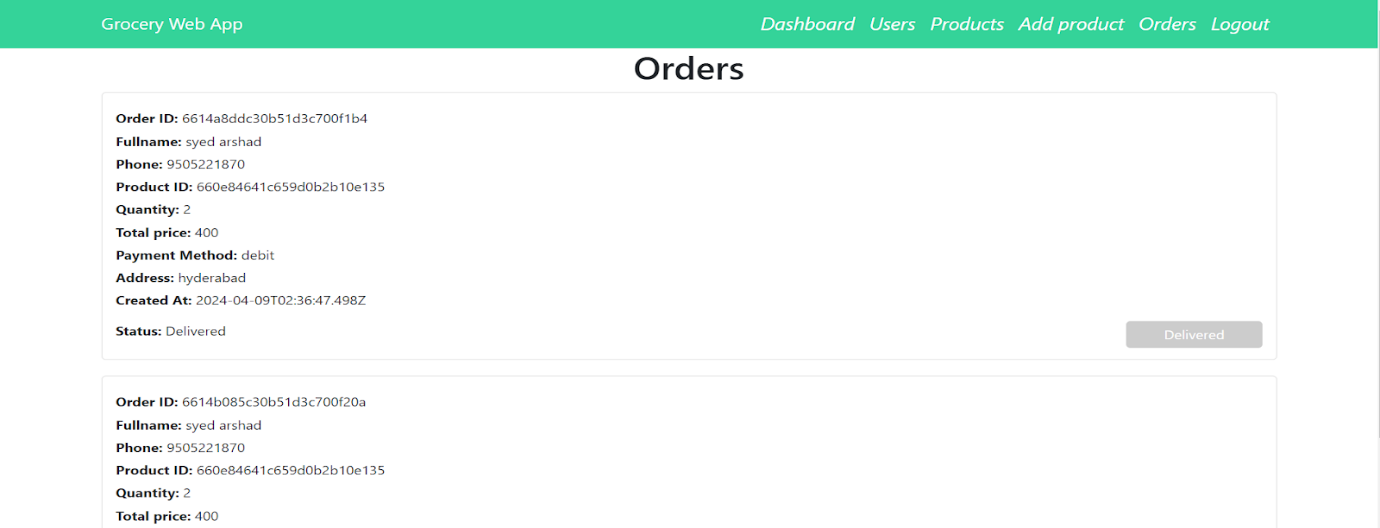
Admin Dashboard Page:



Users Pag

* Add Product page:-



* Admin Orders Page:- 

**8.ADVANTAGES & DISADVANTAGES**

**Advantages**

* Convenience for Users: ShopSmart allows customers to browse, select, and purchase groceries from the comfort of their homes, saving time and effort.
* Real-Time Inventory Management: By providing live product availability, it prevents issues like placing orders for out-of-stock items, improving customer trust.
* Scalable & Customizable: The platform can be easily scaled to support more stores and customized to fit different business models, from local shops to regional chains.
* Cost-Effective for Retailers: ShopSmart empowers smaller grocery businesses to go digital without investing in expensive enterprise systems.
* User-Friendly Interface: Designed with simplicity in mind, the app provides a smooth experience for customers of all age groups and tech familiarity levels.
* Secure Payments: Integrated payment gateways ensure safe and seamless transactions, boosting user confidence.
* Eco-Friendly Approach: Reducing the need for printed bills and physical shopping trips supports more sustainable shopping habits.

**Disadvantages**

* Internet Dependency: The platform requires a stable internet connection, which can limit its accessibility in areas with poor connectivity.
* Initial Setup for Shop Owners: Vendors with little technical background may find it challenging at first to upload products and manage the dashboard without training.
* Maintenance Requirements: Regular updates and technical support are necessary to keep the system secure and functioning smoothly.
* Limited Physical Interaction: Some users prefer the tactile experience of in-person shopping, such as checking freshness or quality, which this system cannot replicate.

**9. CONCLUSION**

The *ShopSmart* project represents a meaningful step toward transforming the traditional grocery experience through digital innovation. By identifying real-world pain points and addressing them with a practical, user-friendly platform, this project successfully bridges the gap between technology and everyday needs.

Through intuitive design, real-time features, and smooth functionality, ShopSmart simplifies the shopping journey for consumers while empowering store owners with efficient tools to manage their inventory and orders. From ideation to deployment, every phase of this project emphasized user-centric design, responsiveness, and scalability—making it a viable and impactful solution in the growing digital retail space.

This project not only highlights technical proficiency in full-stack development but also reflects thoughtful planning, design thinking, and the ability to deliver meaningful user experiences. *ShopSmart* stands as both a functional application and a valuable learning milestone in building future-ready digital solutions.

**10.FUTURE SCOPE**

The *ShopSmart* platform has the potential to evolve into a comprehensive digital retail solution. Future enhancements include developing dedicated mobile apps for Android and iOS, integrating AI for personalized product recommendations, and adding multilingual support to reach a broader audience.

Features like voice search, delivery partner integration, and a loyalty rewards system can further improve user experience. Advanced analytics dashboards for store owners and offline functionality through Progressive Web App (PWA) development would boost usability, performance, and adoption.

These upgrades would make ShopSmart more intelligent, accessible, and ready for long-term success in the digital grocery market.

**11. APPENDIX**

* **Source Code**: [https://github.com/magantipadma/Grocery-Web-App-Mern-Main]
* **Dataset Link**: [Mention if any datasets were used]
* **GitHub & Project Demo Link**:
  + GitHub Repository: [https://github.com/magantipadma/Grocery-Web-App-Mern-Main ]
  + Live Demo / Video Walkthrough:

[https://drive.google.com/file/d/1HQEkysRjQXmnwHh\_9sSn4zUmN ka6jTZU/view?usp=drivesdk ]