CANTEEN MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted By

KRISHNA SHAH

In partial fulfillment for the award of degree of

BACHELOR OF ENGINEERING

in

Computer Engineering

Madhuben and Bhanubhai Patel Institute of Technology, Anand

The Charutar Vidya Mandal (CVM) University,
Vallabh Vidyanagar- 388120

April- 2025

Madhuben and Bhanubhai Patel Institute of Technology, Anand

New Vallabh Vidyanagar, PO Box 8, Vitthal Udyognagar, Anand 388121

CERTIFICATE

This is to certify that **Krishna Shah** has submitted the Industrial Internship report based on internship undergone at Tech Elecon Pvt. Ltd. for a period of **16** weeks from **01/01/2025** to **30/04/2025** in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering, Madhuben and Bhanubhai Patel Institute of Technology at The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar during the academic year 2024– 25.

Prof. Jayna Donga
Internal Guide

Dr. Gopi Bhatt
Head of the Department

TO WHOM IT MAY CONCERN

This is to certify that **SHAH KRISHNA SUNILBHAI**, a student of **BACHELORS OF COMPUTER ENGINEERING of CVM UNIVERSITY, ANAND** has successfully completed her internship in the field of **React.js from 01/01/2025 to 30/04/2025** under the guidance of **Mr. Satyam Raval**, Deputy General Manager at **Tech Elecon Pvt Ltd**.

Her internship activities include successful completion of the assigned project at the given period of time along with abiding by companies' rules and regulation.

During the period of her internship program with us, she had been exposed to different processes and was found diligent, hardworking, and inquisitive.

We Wish her every success in her life and career.

For Tech Elecon Pvt. Ltd,

Madhuben and Bhanubhai Patel Institute of Technology, Anand

New Vallabh Vidyanagar, PO Box 8, Vitthal Udyognagar, Anand 388121

DECLARATION

I, **Krishna Shah ,** hereby declare that the Industrial Internship report submitted in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering, Madhuben and Bhanubhai Patel Institute of Technology, The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar, is a bonafide record of work carried out by me at Tech Elecon Pvt. Ltd. under the supervision of **Sr. Satyam Raval** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student **Sign of Student**

Krishna Shah

ACKNOWLEDGMENT

I hereby, would like to have the privilege to show our gratitude to all the persons, helped me in whatever way for the successful completion of this internship without hindrance. I am grateful to all our mentors who inspired me by setting an example of them for the kind purpose of motivating me to reach my targeted objective. Without their knowledge and wisdom along with experience and specialization in their specific field, I would not have been able to think of doing or completing this work. All the persons who have contributed directly or indirectly with their kind support and humble approach are highly appreciative and I would always remain indebted to them in all the ways. I am especially thankful of our internal guide and HOD respectively, for their kind support and motivation. I extend my heartfelt thanks to **Mr. Raval**, Senior Manager for his co-operation in our project work.

Krishna Shah

ABSTRACT

I carried out my internship at Tech Elecon Pvt. Ltd, Anand. Internship is an opportunity to relate what has been covered in class and what is applicable in the field in an operational environment. The purpose of the program is to fulfil the core equipment for the award of bachelor's degree in Computer Engineering to get a practical aspect of theoretical work studied at the university and to understand the operations in the IT industry and to enable students gain experience in different tasks.

During Internship period I have worked on the Canteen Management System (Cafeteria Project), as React Intern.

In conclusion, this was an opportunity to develop and enhance the skills and competencies in my career field which I achieved.

List of Figures

List of Tables

Table of Contents

CHAPTER 1: OVERVIEW OF THE COMPANY

1.1 HISTORY

Tech Elecon Pvt. Ltd is the IT division of the Elecon group of companies and has more than 25 years of experience in the fields of hardware, software, and networking solutions.

It is situated in the heart of Vitthal, Udyognagar Industrial Estate and in the proximity of the educational town of Vallabh Vidyanagar.

Tech Elecon is all set to reach new heights in the field of IT solutions. Tech Elecon is ready with all sorts of solutions and delivers any application that is web based and further our solutions are designed to adapt your business rather than your business adapting the software. Their solutions are 100% fruitful and empower you to take control of client's business online and in real time.

Tech Elecon have more than 100 employees with specialized skills in software development, custom software development, and e-commerce software development using custom software programming including.NET, C#.NET, PHP, and Open Source and Oracle.

Tech Elecon delivers quality products and services with a focus on integrating the same with existing technologies, providing the required automation to our customers to help them achieve their business objectives.

Mr. Nilesh Naik, the company's Vice President, is at the helm of the Tech Elecon organization. Mr. Satyam Raval, as Deputy General Manager, and after that, Manager and Associate Manager Positions are listed. At the bottom, there are trainees at entry level, who follow up to engineer, senior engineer, also executive and senior executive engineer.

Company Name	Tech Elecon Pvt. Ltd.
Company Type	Service Base
Company Address	Anand - Sojitra Road, Vithal Udyognagar, Vallabh Vidyanagar-388120, Anand, Gujarat.
Contact No.	+91 90990 36045
Website	**http://www.techelecon.com/**
Location	Anand, Gujarat
Company Manager	Mr. Satyam Raval

1.2 DIFFERENT PRODUCTS OF THE COMPANY

Tech Elecon has extensive experience in providing IT services and has successfully adapted to technological advancements, making it the leading IT infrastructure management service provider in the region. Our cutting-edge delivery model covers all stages of the solution lifecycle, including planning, deploying, managing, maintaining, auditing, upgrading, and improving.

Tech Elecon recognize that each client has unique needs and expectations when it comes to infrastructure and service providers. Our clients have the flexibility to choose from a wide range of IT infrastructure management and performance services based on their specific requirements. They can opt for on-site services or hybrid solutions that include on- site troubleshooting and support services.

Tech Elecon Provide Following Service for Business

- Hardware maintenance and repairing
- Service desk management
- Desktop management
- Network management
- Messaging administrator
- Back-up

management Other

Services

- Software Development Services
- Software Licensing
- Microsoft Product Implementation
- Linux Servers / Desktop Implementation

1.3 CAPACITY OF THE COMPANY

Currently our company holds over more than 100 employees. But as the company is growing rapidly, capacity is going higher and higher.

CHAPTER 2: OVERVIEW OF DIFFERENT DEPARTMENT

2.1 DIFFERENT DEPARTMENT

The development of the Cafeteria Management System involves collaboration across various departments or roles that represent essential responsibilities in a software project. These include:

Frontend Development Department: Responsible for designing and implementing the user interface using React.js. This includes building reusable components such as navigation bars, login forms, product cards, and dashboards.

Backend Development Department: Handles server-side logic using Node.js and Express. It includes API creation for user authentication, product management, order processing, and administrative functionalities.

Database Department: Uses MongoDB to store all system data such as user information, product listings, order history, and feedback. MongoDB Compass was used for visual management of the database.

UI/UX Design Department: Focuses on user experience, dark/light themes, and interactive design. CSS frameworks and animations were used to enhance the look and feel.

Project Management Department: Oversees the planning, scheduling, and integration of tasks. Ensures deadlines are met and the scope of the project is maintained.

2.2 SEQUENCE OF OPERATION FOR MANUFACTURING OF END PRODUCT

The development workflow followed a structured sequence:

1. **Requirement Analysis:** This is the foundation of the project where the functional and non-functional requirements of the Cafeteria Management System are studied. Key features such as digital ordering, cart management, payment simulation, feedback collection, and admin dashboard functionalities were identified. Understanding user behavior and expectations was crucial during this phase.

2. **UI/UX Design:** At this stage, visual layouts and interactive components were planned to ensure a smooth user experience. Wireframes and mockups were created to define the user journey from login to order placement. Special focus was given to modern aesthetics, dark mode support, and intuitive navigation.

3. **Frontend Development:** Frontend Development, React.js and Redux were used to build a responsive UI and manage app-wide state effectively.

4. **Backend Development:** The Backend Development involved creating RESTful APIs using Node.js and Express to handle user, product, and order data securely.

5. **Database Integration:** Database Integration was done using MongoDB and Mongoose to store and manage all application data with the help of MongoDB Compass.

6. **Authentication System:** A simple Authentication System was implemented on the frontend to differentiate between user and a dmin views.

7. **Testing & Debugging:** In the Testing & Debugging stage, each module and API was checked for bugs and edge cases to ensure proper functionality.

2.3 DIFFERENT STAGES OF PRODUCTION

The production of the Cafeteria Management System followed several key stages. It began with the **Planning Stage**, where user needs were analyzed, goals were set, the tech stack was chosen, and a timeline was defined. In the **Design Stage**, UI wireframes were created, and the frontend-backend data flow was mapped out. The **Development Stage** involved building the frontend with React.js, the backend with Node.js, and integrating MongoDB for data storage. Important features like QR code ordering and cart management were implemented. In the **Testing Stage**, all modules were tested to ensure smooth functionality. The **Documentation Stage** focused on preparing detailed system documentation for future use. Finally, during the **Review & Feedback Stage**, suggestions were collected from users and mentors to improve the system.

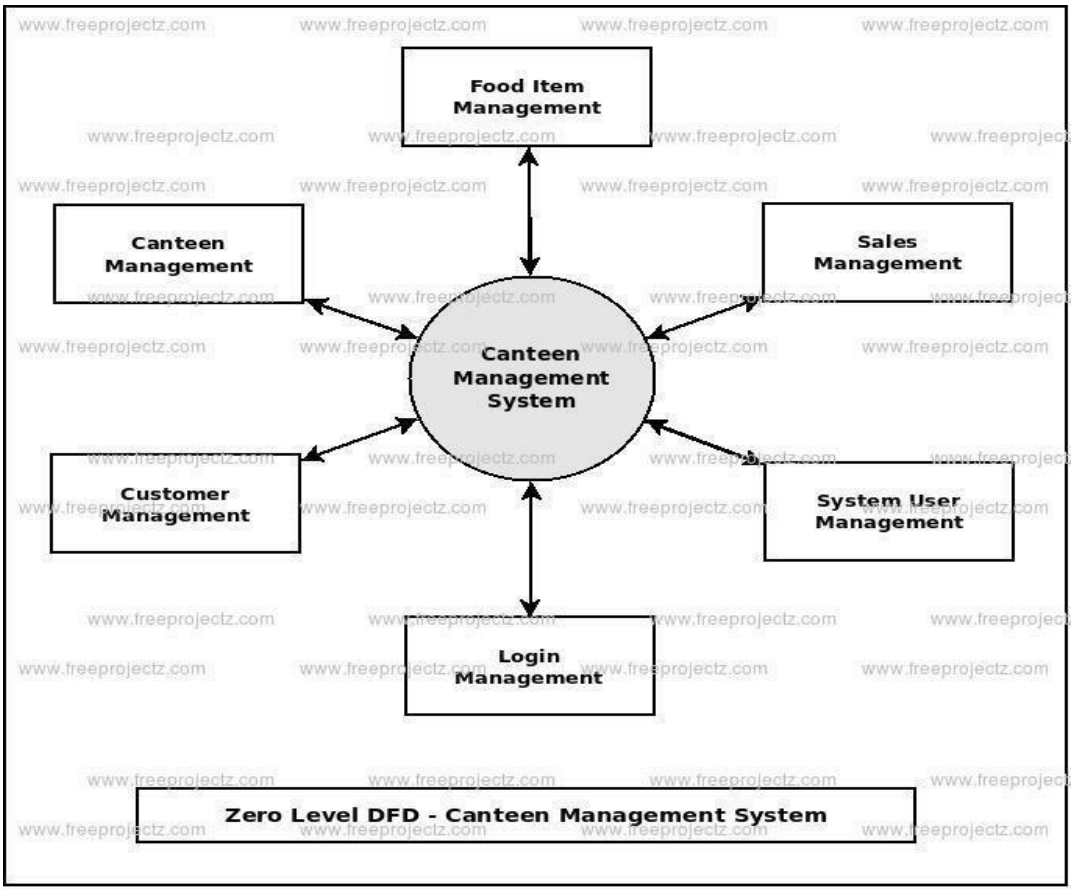

Figure 1 Production Process

CHAPTER 3: INTRODUCTION TO INTERNSHIP AND PROJECT

3.1 INTERNSHIP SUMMARY

Project Title: Canteen Management System

As part of the academic curriculum, I undertook an internship to gain practical exposure to real-world software development. During this internship, I worked on a **Cafeteria Management System**, a full-stack web development project that simulates the operations of a modern digital cafeteria, offering a seamless user experience for ordering and managing food services online. The internship provided valuable hands-on experience in both frontend and backend technologies, including **React.js, Node.js, MongoDB, and Express.js**, along with tools like **Vite** for fast frontend development and **Axios** for efficient HTTP communication.

This internship helped bridge the gap between theoretical learning and practical implementation. I was actively involved in various phases of the **Software Development Life Cycle (SDLC)**, including requirement gathering, interface design, system architecture planning, coding, testing, and documentation. It also strengthened my understanding of industry workflows such as **agile development**, task breakdown, time estimation, team collaboration, planning, and version control through tools like Git.

Key Features of the Canteen Management System:

- **Real-time order tracking** to keep users informed about their order status
- **Secure payment processing with multiple payment options**
- **Admin dashboard for managing inventory, orders, and customer feedback**

  User authentication and role-based access for enhanced security and control

3.2 PURPOSE

The primary purpose of this project is to modernize and enhance canteen operations by implementing an automated management system. Traditional canteen processes involve manual order-taking, cash transactions, and inefficient tracking, which often lead to delays, errors, and mismanagement of resources. By integrating a digital system, this project aims to eliminate these inefficiencies, ensuring a smooth and organized workflow.

The system is designed to minimize human intervention in routine tasks such as order placement, payment processing, and order tracking, thereby reducing the workload for staff and improving overall service quality. With features. customers can browse the menu, place orders, and make payments seamlessly through an intuitive interface. This not only enhances user experience but also speeds up the entire transaction process.

Additionally, the project aims to improve operational efficiency by providing real-time updates on order status, reducing wait times, and optimizing kitchen workflows. Through an admin dashboard, management can monitor daily transactions, track inventory levels, and analyze sales trends, leading to better decision-making and resource allocation. The introduction of secure digital payments will also ensure accuracy in financial transactions, reducing the risk of cash handling errors.

Overall, this project seeks to transform traditional canteen operations into a more efficient, technology-driven system that benefits both customers and management, ensuring a seamless and hassle-free dining experience.

3.3 OBJECTIVE OF PROJECT

The main objective of this project was to develop an interactive and efficient web-based platform that enables users to browse food items and place orders digitally with ease. The system was designed to enhance the overall user experience by offering functionalities like real-time cart management, order status tracking, and post-order feedback submission. By digitizing the ordering process, the platform helps eliminate delays, improves service speed, and brings convenience to both customers and cafeteria staff.

On the administrative side, the project aimed to provide a robust dashboard that allows for smooth management of orders, menu updates, and feedback monitoring. Role-based login was implemented to ensure secure and appropriate access for users and administrators, with dedicated features tailored to each role. A significant focus was placed on creating a clean, animated, and modern user interface that supports dark mode and adapts well to various devices. The project also simulated digital payment workflows and order tracking processes, reflecting the operations of a real-life cafeteria. Scalability and flexibility were considered during development to support future upgrades and deployment on larger platforms.

3.4 SCOPE

The **Canteen Management System** is designed to enhance the overall efficiency of canteen operations by incorporating various digital features.

- Separate dashboards for users and admins, ensuring role-based access and functionality.
- Real-time management of orders, item listing, cart updates, and feedback collection.
- Secure storage of user details, order history, and feedback using MongoDB with schema validation via Mongoose.
- A clean, responsive UI that works smoothly across various devices and screen sizes, ensuring accessibility and ease of use.
- Although it is currently hosted locally for testing purposes, the system is structured to be easily deployed on cloud platforms and expanded with additional features like live payments, multi-admin access, or advanced analytics in future iterations.

.

3.5 TOOLS AND TECHNOLOGY

- **Frontend Technologies:** The frontend is responsible for the user interface and interactions, providing a responsive and interactive experience.
  - **React.js:** A JavaScript library used to build dynamic and efficient user interfaces, ensuring a smooth and interactive user experience.
  - **HTML & CSS:** These core web technologies define the structure and styling of the application, ensuring a visually appealing layout.
  - **Bootstrap:** A CSS framework that helps in designing a responsive and mobile-friendly user interface with prebuilt components.
  - **Redux:** A state management library used for handling global states like the cart system and order tracking, ensuring smooth data flow across the application.
  - **Vite** – for fast and modern frontend build and development.
  - **Axios** – for API communication with the backend.

- **Backend Technologies**: The backend is responsible for processing Requests handling business logic, and communicating with the database.
  - **Node.js:** A runtime environment that enables the execution of JavaScript on the server side, ensuring fast and scalable backend performance**.**
  - **Express.js:** A lightweight framework for Node.js that simplifies API development, request handling, and middleware integration, making the backend more efficient and modular.

- **Database Technology**: The system uses a NoSQL database for storing and managing data efficiently.
  - **MongoDB:** as the database to store users, products, orders, etc.
  - **Mongoose** : for schema modeling and database management.

/.

3.6 PROJECT PLANNING

Project planning played a vital role in laying the foundation for the successful execution of the Cafeteria Management System. This phase involved defining the project scope, identifying core functionalities, selecting the development methodology, and choosing the tools and technologies best suited for the system. The project was divided into well-structured modules including user and admin dashboards, authentication, order management, feedback system, cart functionality, and payment simulation. Adopting an agile-inspired approach, the team set weekly goals to ensure consistent progress, quick feedback incorporation, and continuous improvement throughout the development process. Each task was strategically scheduled to follow the SDLC stages—Requirement Analysis, Design, Development, Testing, and Deployment.

To support efficient development, modern and scalable technologies were chosen. React.js combined with Vite was used for a fast and dynamic frontend, while Redux managed global application states such as user sessions and cart data. Axios was implemented for smooth API communication. On the backend, Node.js with Express handled server-side logic and routing. MongoDB served as the NoSQL database, and Mongoose was used for schema design and data validation. Project planning also included assigning clear responsibilities—UI/UX design, coding, database structuring, testing, and documentation—ensuring every aspect of the system was properly managed. This structured and detailed planning ensured that the project stayed on track and aligned with the intended functionality and quality standards.

3.7 PROJECT SCHEDULING

The entire development of the Cafeteria Management System was planned across **16 weeks**, with a structured timeline as follows:

- **Week 1-2: Requirement Analysis & Planning**

- Understand project scope, features, and objectives.
- Finalize technology stack and development tools.
- Create a project roadmap with defined milestones.

- **Week 3-4: UI/UX Design**

- Create wireframes, user flow diagrams, and interface mockups.
- Plan layouts for both user and admin dashboards.
- Decide color schemes, themes (including dark mode), and design responsiveness.
-

- **Week 5-6: Frontend Setup & Component Design**

- Initialize the frontend using Vite.
- Design reusable components for login/signup, item listing, cart, etc.
- Integrate routing and Redux for state management.

- **Week 7-8: Backend Setup & API Development**

- Set up Node.js and Express server.
- Create RESTful APIs for products, users, orders, and feedback.
- Test API endpoints using tools like Postman.

- **Week 9–10: Database Integration**

- Design MongoDB schemas with Mongoose for all modules.
- Integrate backend APIs with the frontend.
- Store and retrieve data for login, orders, and feedback.

- **Week 11–12: Authentication & Role-Based Access**

- Implement login and signup features (frontend-only).
- Add dashboard segregation for users and admins.
- Protect routes and simulate session handling.

- **Week 13–14: Testing & Debugging**

- Perform unit testing and manual testing on each module.
- Fix bugs, handle edge cases like empty inputs or invalid logins.
- Ensure smooth flow between frontend and backend.

- **Week 15: Final Touches & UI Enhancements**

- Add animations, polish UI/UX, and finalize responsiveness.
- Add dark mode and test the app on different devices.
- Simulate digital payment and order flow.

- **Week 16: Documentation & Project Review**

- Prepare final project report and system documentation.
- Present the project and gather feedback for improvement.
- Reflect on challenges faced and lessons learned.

CHAPTER 4: SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

The existing cafeteria management system is primarily manual and relies on traditional methods for order taking, payment processing, inventory management, and order tracking. This system is not only labor-intensive but also prone to human errors, which affects the overall efficiency of cafeteria operations. The manual approach has several limitations that hinder the cafeteria from achieving optimal performance and scalability. Below are the key components of the current system:

1. **Order Taking:** Orders are manually taken by staff, either through verbal communication or paper forms. This leads to several issues:
    - **Human Error**: Misinterpretation or incorrect recording of orders can result in wrong food being prepared, leading to customer dissatisfaction.
    - **Time Consumption**: Manual order taking is slow, particularly during peak hours, causing delays and longer wait times.
    - **Limited Customization**: Special requests or customizations are difficult to accommodate, which impacts the customer experience.
2. **Payment Process:** Payments are manually processed through cash or card, causing challenges:
    - **Transaction Delays:** Cash payments take time to count, and card payments may face processing delays due to reader or connectivity issues.
    - **Accounting Errors:** Manual entry increases the risk of mistakes, such as incorrect amounts or misplaced receipts.
    - **Limited Payment Options:** The system may not support digital wallets or mobile payments, which could reduce convenience for some customers.

3. **Inventory Management:** Inventory is manually tracked through physical records or spreadsheets, leading to:
    - **Manual Tracking**: Updating inventory is time-consuming and prone to errors.
    - **Lack of Real-Time Visibility**: Staff may be unaware of stock shortages until it's too late, resulting in unavailable items and customer dissatisfaction.
    - **Waste and Overordering**: Without automation, overordering or stock-outs may occur, affecting profitability.

4. **Order Tracking:** There is no system for tracking order status, causing:
    - **Customer Frustration**: Lack of updates on order status leads to confusion and frustration, especially during long waits.
    - **Inefficient Communication**: Staff members are frequently interrupted for updates, affecting workflow and delaying other orders.

4.2 PROBLEM AND WEEKNESS OF CURRENT SYSTEM

The existing system faces several challenges that limit its efficiency and customer satisfaction.

- **Human Error:** As the system relies heavily on manual processes, there is a high risk of errors such as incorrect order entries, billing mistakes, and inaccurate inventory tracking. These errors can lead to customer dissatisfaction and operational losses.
- **Time Inefficiency:** Manual order-taking and payment processing are slow and tedious, especially during busy hours. This results in long queues, increased customer wait times, and delays in service, which ultimately reduce the cafeteria's overall efficiency.
- **Lack of Real-Time Tracking:** There is no system in place for real-time tracking of orders. Neither staff nor customers can monitor the status of ongoing orders, which often leads to confusion, repeated inquiries, and disrupted workflow.
- **Limited Scalability:** The current system becomes harder to manage as the volume of customers or orders increases. Scaling up operations with a manual system is inefficient and may require more manpower, increasing operational costs.
- **Poor User Experience:** The lack of digital interfaces such as online ordering, self-service kiosks, or mobile apps leads to a dull and outdated customer experience. This can be a drawback, especially for younger or tech-savvy users who expect faster and more interactive service.

These problems collectively limit the cafeteria's ability to grow, optimize operations, and meet evolving customer expectations in a competitive environment.

4.3 REQUIREMENT OF NEW SYSTEM

To overcome the limitations of the traditional canteen management system, the proposed **Canteen Management System** must fulfill several key requirements to ensure efficiency, accuracy, and a seamless user experience. The primary requirements include:

- **Automated Order Processing:** Customers should be able to place orders online through a digital platform (e.g., website or app), eliminating the need for manual order-taking.
- **Real-time Order Tracking:** The system should provide **real-time updates** on order status (e.g., Processing, Ready for Pickup, Completed). Customers should be able to check their order status from their mobile devices without manual inquiries.
- **Payment Integration:** A secure, seamless payment system that supports multiple payment methods (e.g., credit cards, mobile wallets) for faster and more accurate transactions.
- **Inventory Management:** The system should automatically update inventory levels based on orders placed, providing real-time visibility into stock levels.
- **User Authentication:** A login and registration system for customers to create personalized profiles and manage their orders, payment history, and preferences.
- **Admin Dashboard:** A robust admin panel that allows the cafeteria staff to manage orders, update the menu, track inventory, and generate reports.
- **Responsive Design:** The system should be accessible from both desktop and mobile devices, ensuring a seamless user experience across platforms.

4.4 SYSTEM FEASIBILITY

The feasibility study assesses the practicality of implementing the new Cafeteria Management System based on the following factors:

- **Technical Feasibility:** The proposed system will be built using modern web technologies such as React.js for the frontend and Node.js for the backend. These technologies are widely used and offer scalability and flexibility. Given the skillset of the development team, this technology stack is technically feasible.
- **Operational Feasibility:** The new system is easy to integrate into the current cafeteria operations. The cafeteria staff can be trained to use the new system effectively, and customers will find it intuitive, especially if they are familiar with similar systems.
- **Financial Feasibility:** The costs associated with developing and implementing the new system are relatively low compared to the operational benefits it offers. The use of open-source technologies reduces licensing costs, and the system can be hosted on affordable cloud platforms.
- **Legal Feasibility:** The system will comply with data protection regulations (such as GDPR) to ensure that customer data is secure and handled responsibly. Payments will be processed through secure gateways, ensuring compliance with financial standards.
- **Risk Analysis**: Potential risks include: System Downtime → Implement cloud-based backups and a contingency plan. Payment Failures → Integrate secure payment gateways with error handling. User Data Security → Implement encryption and secure authentication to prevent unauthorized access.

4.5 FEATURES OF SYSTEM

The Cafeteria Management System will include the following features:

- **User Registration and Login:** Customers can create accounts to save their preferences and view past orders.
- **Order Placement:** Customers can browse the menu, add items to their cart, and place orders directly from the website or app.
- **Cart Management:** Users can view, update, or remove items in their cart before completing the purchase.
- **Order Tracking:** Customers can track the status of their order in real-time, from "Order Received" to "Ready for Pickup."
- **Payment Gateway:** A secure and simulated payment process allows customers to finalize their order using various payment methods.
- **Order History:** Customers can view their past orders and track their spending.
- **Admin Dashboard:** The admin panel allows staff to manage orders, track inventory, add/update the menu, and process payments.
- **Notifications:** Customers will receive notifications regarding order status updates, new menu items, or special offers.

These features aim to streamline operations, improve the customer experience, and provide the cafeteria management with the tools they need to enhance their services.

4.6 SELECTION OF HARDWARE AND SOFTWARE

To develop and deploy the Cafeteria Management System, the following hardware and software will be used:

Software Requirements:

- **Operating System**: Windows, Linux or macOS
- **Frontend**: React.js for building the UI, styled with Tailwind CSS or Bootstrap. Vite is used as the frontend build tool for faster development and optimized performance.
- **Backend**: Node.js with Express.js for server-side logic.
- **Database**: MongoDB for storing user, order, and menu data.
- **API Communication:** Axios for handling API requests.
- **Payments:** Integration with Stripe or PayPal for secure transactions.
- **Deployment (Optional)**: Heroku, Firebase, AWS

Hardware Requirements:

- **Processor**: Intel Core i5 or higher
- RAM: 8GB or more
- Hard Drive Space: 50GB or more
- **Network**: Stable internet, local/cloud hosting

CHAPTER 5: SYSTEM DESIGN

5.1 SYSTEM DESIGN & METHODOLOGY

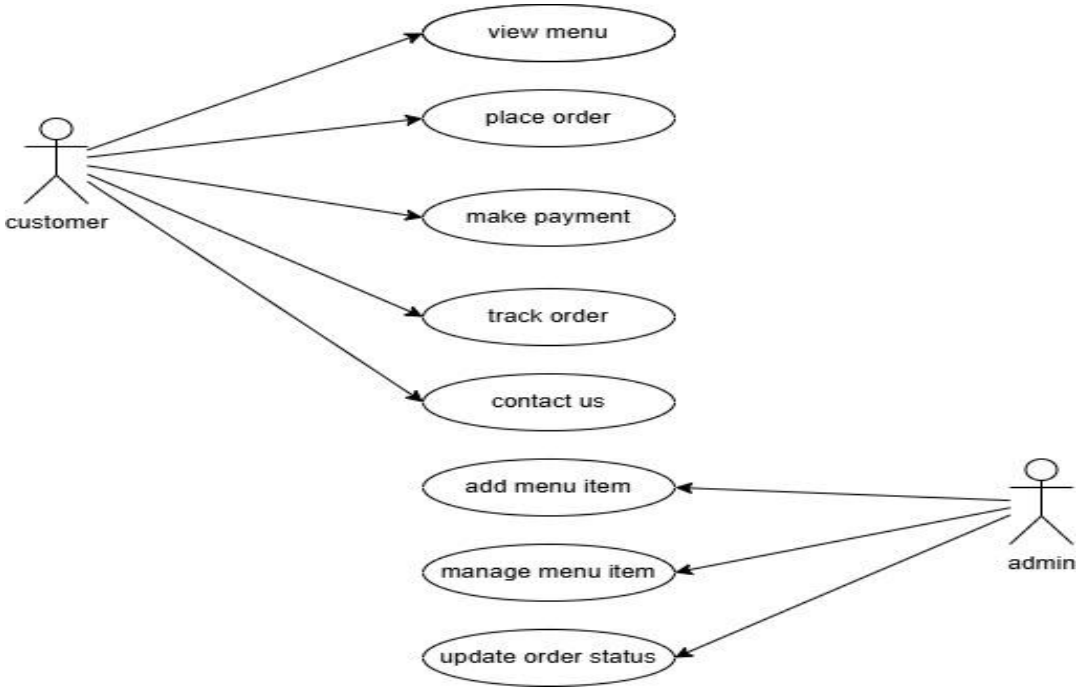

Figure 2 Use Case Diagram

5.1.1 Use Case Diagram:

The **Canteen Management System** involves three primary actors: **Customers, Admin, and Staff**. Each actor interacts with the system to perform various tasks, as shown in the use case diagram. The key processes and interactions are described below:

1. **Customers (Actor):**
   Customers interact with the system to place orders and manage their canteen experience. Their use cases include:

   - **View Menu: Customers can browse the available food items.**
   - **Search/Filter Menu**: Users can search or filter menu items based on

categories, price, or availability.

- **Place Order**: Customers can add items to their cart and place an order.
- **Make Payment**: The system allows users to simulate payments via available methods.
- **Track Order**: Customers can check the status of their orders.
- **Contact Support**: Customers can reach out to staff for assistance.
- **Receive Notifications**: Customers get notified about order updates and promotions.

2. Staff (Actor):

The staff interacts with customers and admin to manage orders efficiently. Their use cases include.

- **Contact Support: Staff members assist customers with queries or issues.**
- **Notify Customers**: Staff can update customers about their order status.

3. Admin (Actor):

The admin manages the entire canteen system, including orders, users, and reports. Their use cases include**:**

- **Update Order Status**: Admins can change order statuses (e.g., preparing, ready, delivered).
- **Manage Menu Items**: Admins can add, remove, or update food items.
- **Manage Users**: Admins can add or remove users, including staff and customers.
- **View Sales Reports**: Admins can generate reports on sales and customer orders for analysis.

5.2 DATA DICTIONARY:

5.2.1 Food Table:

Field Name	Data Type	Description	Example Value
_id	ObjectId	Unique identifier for the food item	680b6df07fb18f19010cd9f7
name	String	Name of the food item	cake
description	String	Short description of the food item	desert
price	Integer	Price of the food item in currency (e.g., INR)	80
image	String	Image file path or filename of the food item	1745579504958174499840 4283food_10.png
category	String	Category to which the food item belongs	Deserts
_v	Integer	Version key (for Mongoose document versioning)	0

Table 1 Food Table

5.2.2 Orders Table:

Field Name	Data Type	Description	Example Value
_id	**ObjectId**	**Unique identifier for the order**	**680b6b6c7fb18f19010cd9e2**
userId	**ObjectId (String)**	**ID of the user who placed the order**	**680b69c17fb18f19010cd9cd**
items	**Array of Objects**	**List of food items included in the order**	**See item subfields below**
amount	**Integer**	**Total price of the order**	**270**
address	**Object**	**Delivery or billing address details**	**See address subfields below**
status	**String**	**Current status of the order**	**Food Processing**
date	**Date (Timestamp)**	**Order placement date and time**	**2025-04-24T06:23:34.109Z**
payment	**Boolean**	**Payment status (true = paid, false = unpaid)**	**true**
_v	**Integer**	**Version key (Mongoose document versioning)**	**0**

Table 2 Orders Table

5.2.3 Items Table:

Field Name	Data Type	Description	Example Value
_id	ObjectId	Food item ID	680b69377fb18f19010cd9c6
name	String	Name of the food item	big burger
description	String	Description of the food item	Full meal burger for perfect meal
price	Integer	Price of the food item	100
image	String	Filename or path of food image	1745578295030174500548621 1food_4.png
category	String	Food category	Burger
_v	Integer	Version key	0
quantity	Integer	Quantity ordered	1

Table 3 Items Table

5.2.4 Address Table:

Field Name	Data Type	Description	Example Value
firstName	**String**	**First name of the customer**	**Krishna**
lastName	**String**	**Last name of the customer**	**Shah**
email	**String**	**Email address**	krishna**@gmail.coom**
employee_email	**String**	**Company email**	krishna**@gmail.coom**
department	**String**	**Department in organization**	**ITTechnology**
sub_department	**String**	**Sub-department**	**IT Support & Helpdesk**
floor	**String**	**Office floor number**	**5**
desk_location	**String**	**Desk location within the floor**	**124**
phone	**String**	**Contact number**	**9867545672**

Table 4 Address Table

5.2.5 Users Table:

Field Name	Data Type	Description	Example Value
_id	ObjectId	Unique identifier for the user	680b65b47fb18f19010cd9be
name	String	Full name of the user	chesta
email	String	Email address of the user	chesta@gmail.com
password	String	Hashed password (bcrypt hash used for authentication)	$2b$10$Yl3gqeoODAA5YVCd.Dn7XuWMXIXdcJrluWg8K97odrXYo/RYbk7KW
Cart Data	Object	Stores user-specific cart-related information (currently empty)	{}
__v	Integer	Version key (Mongoose document versioning)	0

Table 5 Users Table

CHAPTER 6: IMPLEMENTATION

6.1 IMPLEMENTATION PLATFORM

- Visual Studio Code (VS Code): Visual Studio Code (VS Code) is a lightweight, open-source Code editor developed by Microsoft. It provides extensive support for JavaScript, React.js, and Node.js, making it an ideal choice for web development. The integrated terminal, debugging tools, and support for extensions simplify the development process.

    o Supports React.js and Node.js for frontend and backend development.
    o Offers an inbuilt Git integration for version control.
    o Provides an extensive marketplace for extensions like ESLint, Prettier, and REST Client.

- Node.js: Node.js is an open-source, cross-platform runtime environment used to build backend services. It enables server-side JavaScript execution, making it efficient for handling real-time applications like the Canteen Management System.

    o Uses Express.js for routing and API management.
    o Supports asynchronous, event-driven architecture, improving performance.
    o Works seamlessly with MongoDB for database management.

- MongoDB: MongoDB is a NoSQL database that efficiently manages unstructured and semi-structured data. It is used to store user details, orders, menu items, and payment records for the Canteen Management System.

    o Ensures consistent state management across the application.
    o Improves performance with efficient data handling.
    o Enables debugging and logging through Redux DevTools.

6.2 TECHNOLOGIES

The system was built using a modern MERN stack (MongoDB, Express.js, React.js, and Node.js), which provides a seamless development experience across the frontend, backend, and database.

- **Frontend:**
  - **React.js:** Used for building the interactive and responsive user interface.
  - **Vite:** A fast development build tool that improves the speed of project setup and hot module replacement during development.
  - **Tailwind CSS / Bootstrap:** Used for efficient and clean UI design and layout customization.
  - **React Router:** Enabled navigation between pages such as login, home, cart, admin dashboard, and order status.
- **Backend**:
  - **Node.js & Express.js**: Used to handle API requests, route management, authentication, and CRUD operations.
  - **MongoDB:** NoSQL database used for storing user data, order information, menu items, and admin records.
  - **MongoDB Compass:** GUI tool used to visualize and manage database collections.
  - **Axios:** A promise-based HTTP client used in the frontend to communicate with backend APIs securely and efficiently.
- **Additional Tools:**
  - **Git & GitHub:** For version control and collaboration.
  - **Deployment Platforms:** Vercel (for frontend) and Render or Heroku (for backend).

6.3 RESULTS

The Canteen Management System was successfully implemented with the following key features:

6.3.1 Home Page:

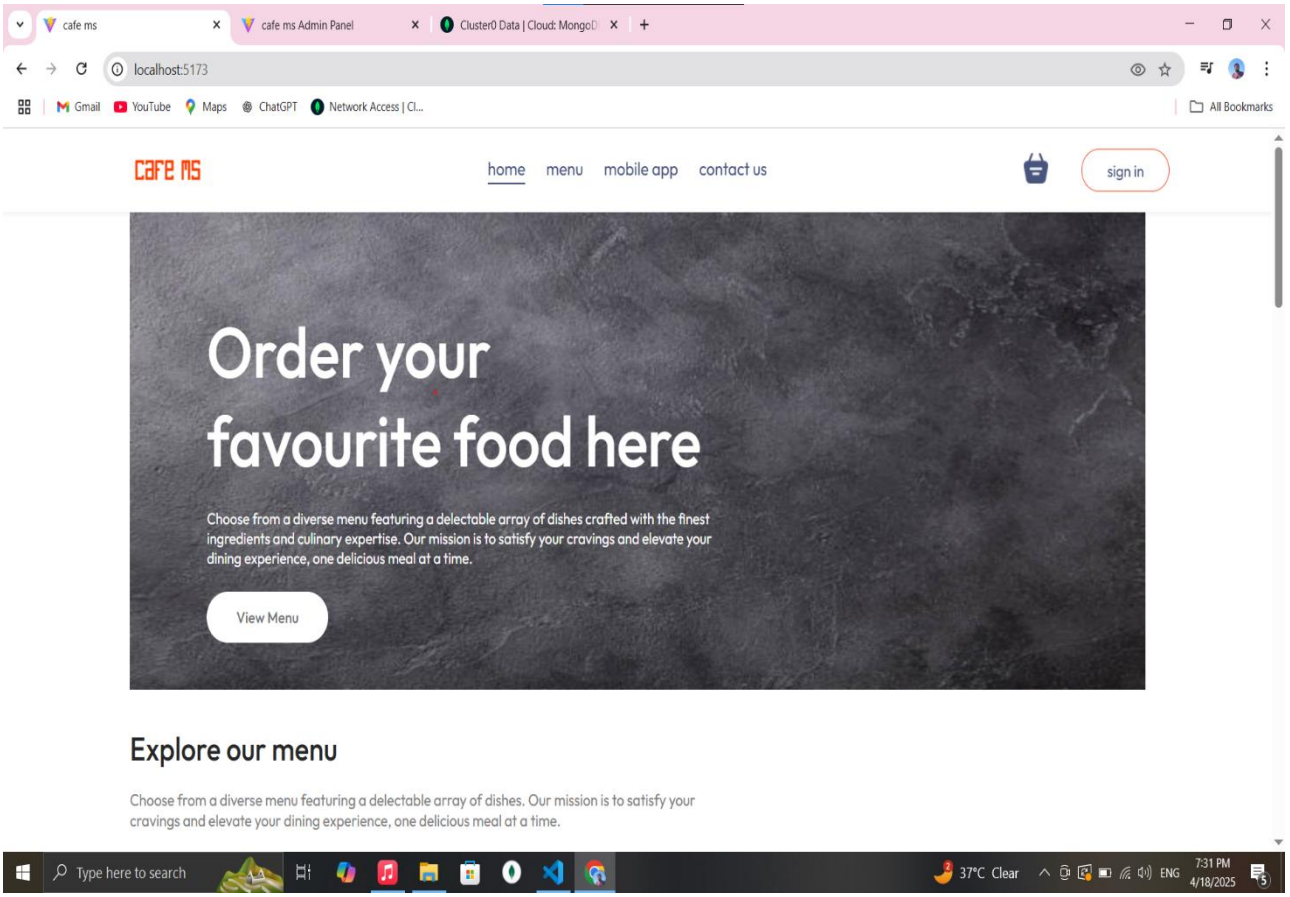

Figure 3 Home Page

- The Home Page serves as the entry point for users, providing an overview of the canteen services.
- It includes navigation links to key sections such as the menu, cart, contact page, and login/signup.
- The UI is designed with a modern and interactive layout to enhance the user experience.

6.3.2 Login/ Sign up Page:

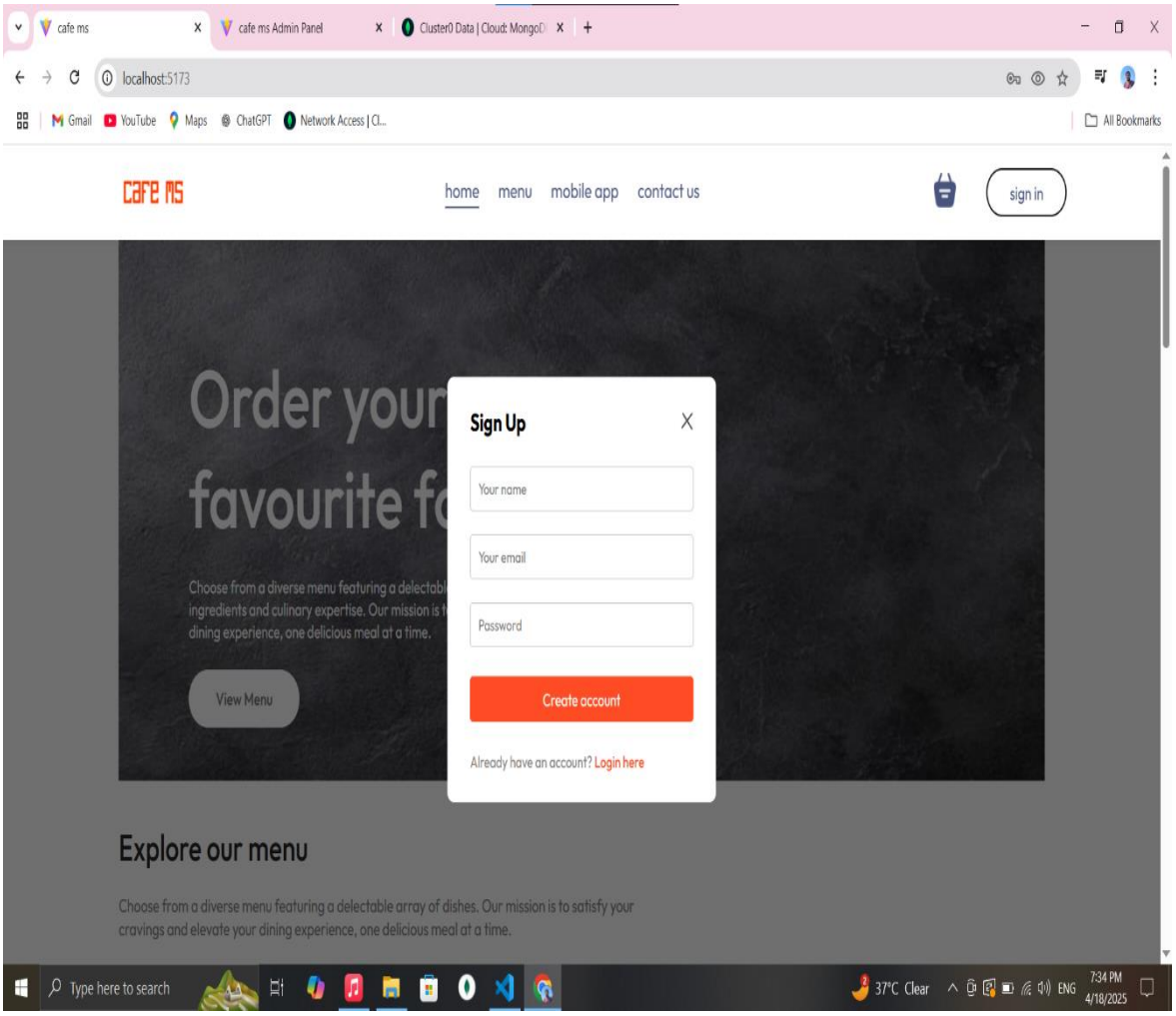

Figure 4 Sign up Page

This form allows **new users to register**. It is built using React form components and connects to the backend using **Axios**.Key features:

- Frontend validation of inputs.
- Sends user data to the backend API for MongoDB storage.
- Implements Redux to manage user authentication state (frontend-only authentication system).

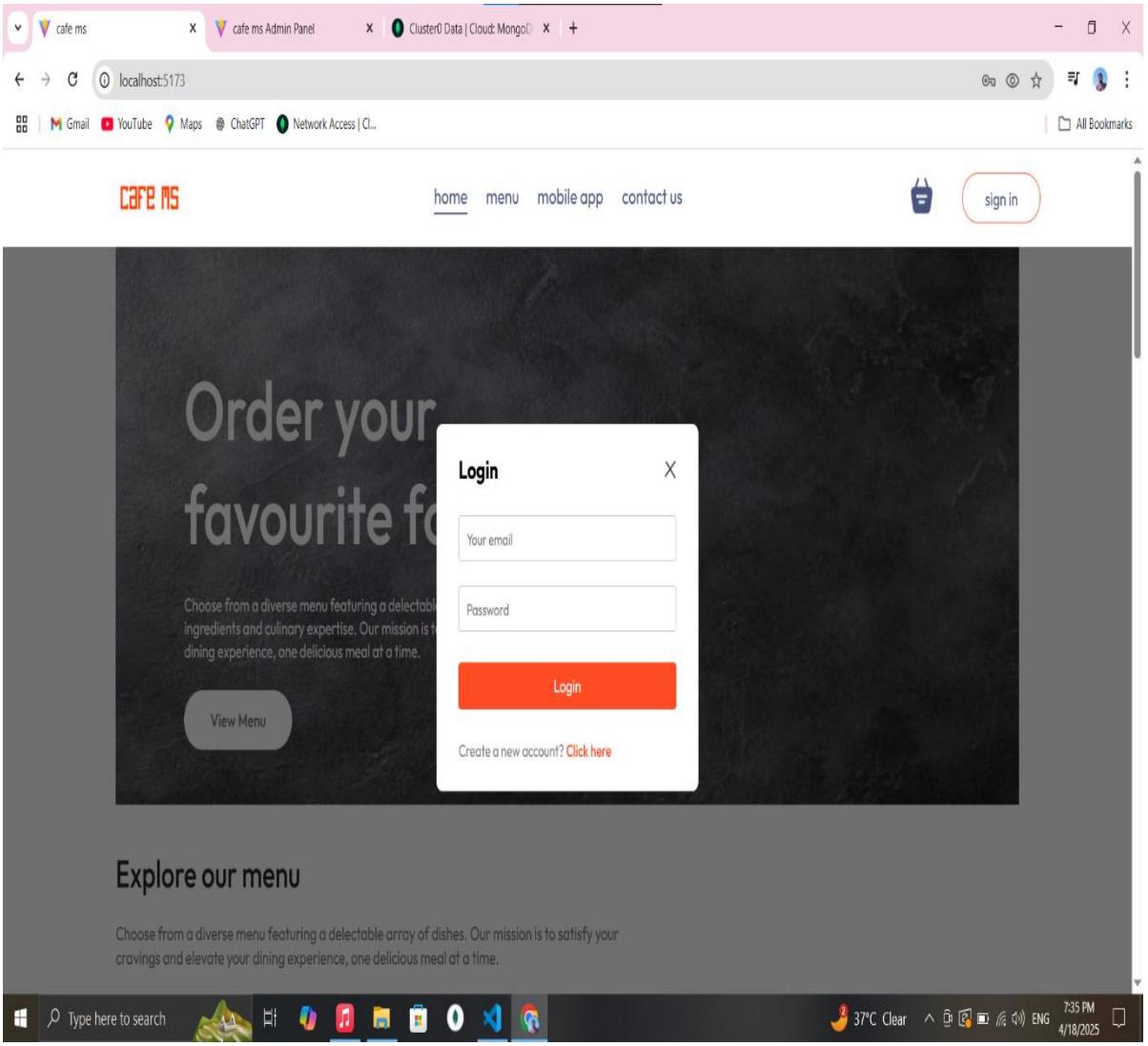

Figure 5 Login Page

Allows returning users to **log in and access their dashboard**.

- Connects to your Node.js API via Axios.
- Includes protected routes that direct users/admins to their specific dashboards based on role.
- Handles invalid credentials with proper error messages.

6.3.3 Menu Page:

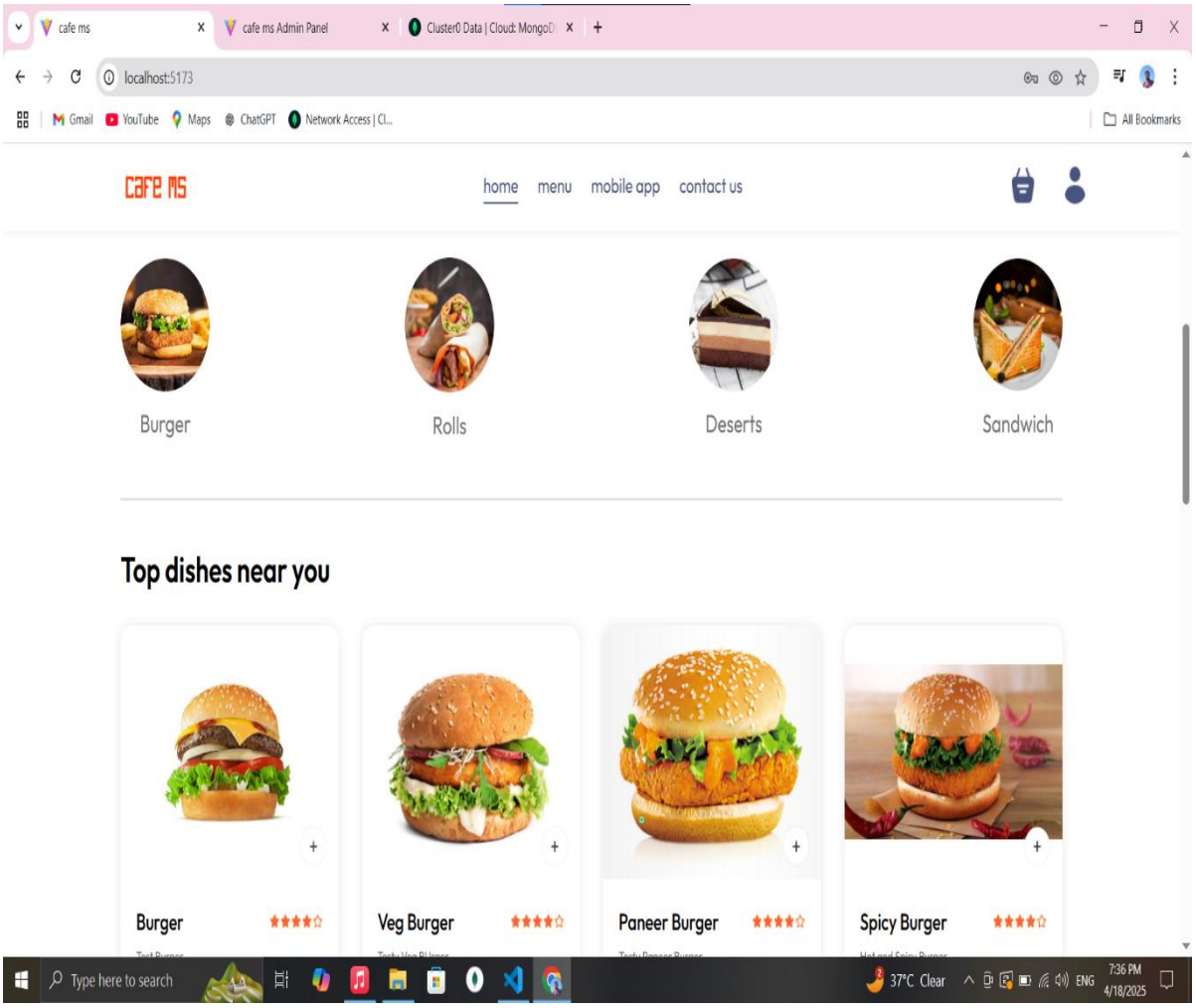

Figure 6 Menu Page

Displays the **entire food listing**, fetched from MongoDB using Axios GET request.

- Each item is shown with image, price, and "Add to Cart" button.
- React components are used for dynamic rendering.
- Filtering or sorting features may be present depending on your implementation.

6.3.4 Add to Cart Page:

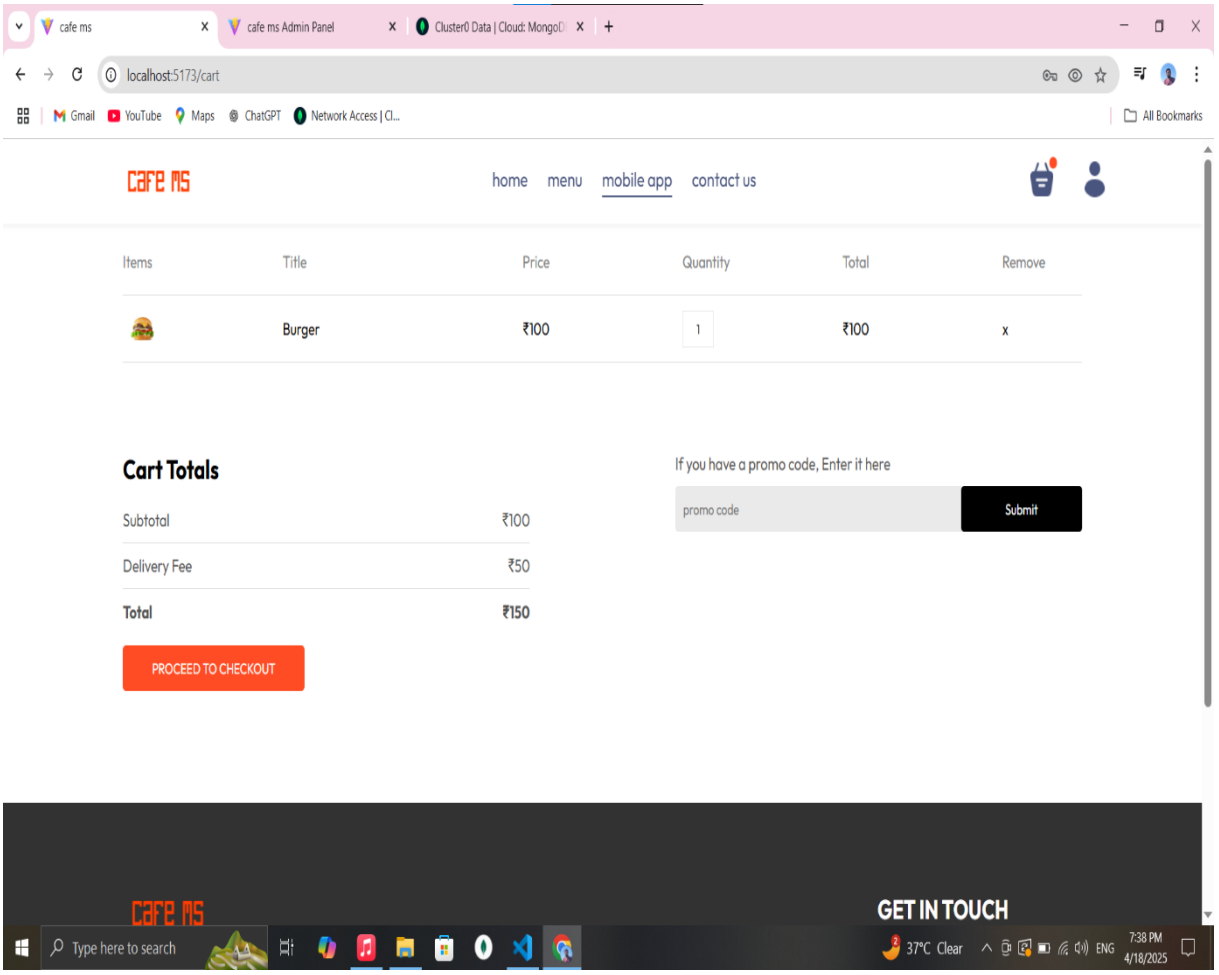

Figure 7 Add to Cart Page

This page shows items that users have added to their cart.

- Redux is used to store and manage the cart data globally, allowing consistent access and Updates across the app.
- Users can modify the quantity of items or remove them entirely from the cart based on their preference.
- The system automatically updates and shows the total price of the cart as items are added, removed, or quantity changest.

6.3.5 Payment Checkout Page:

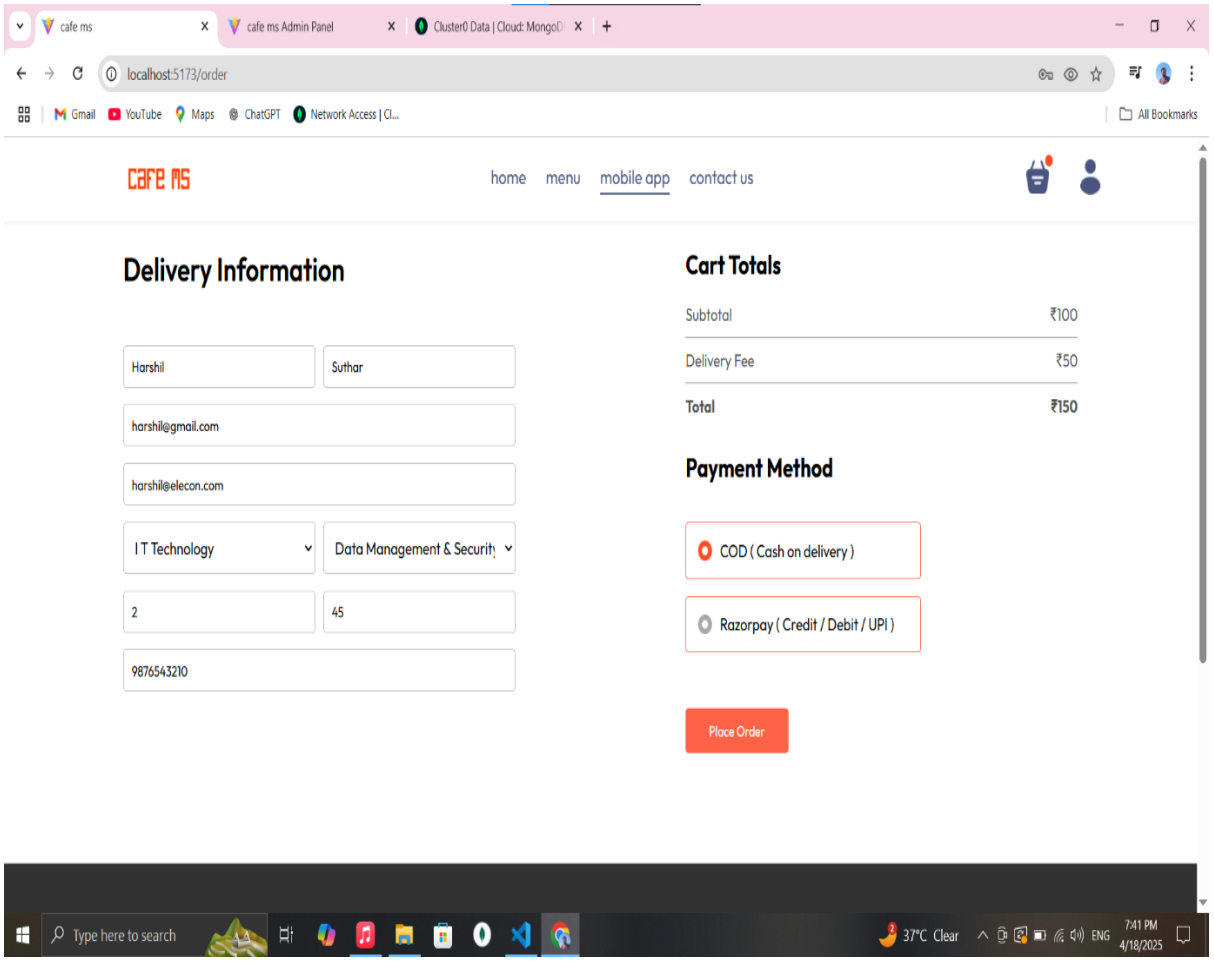

Figure 8 Payment Checkout Page

This screen allows users to **simulate payment** after finalizing their order.

- It may include dummy input for payment options.
- Once confirmed, order data is sent to the backend API and stored in MongoDB.
- Future scope: Integrate real payment gateways like Stripe or Razorpay.

6.3.6 Order-Tracking Page:

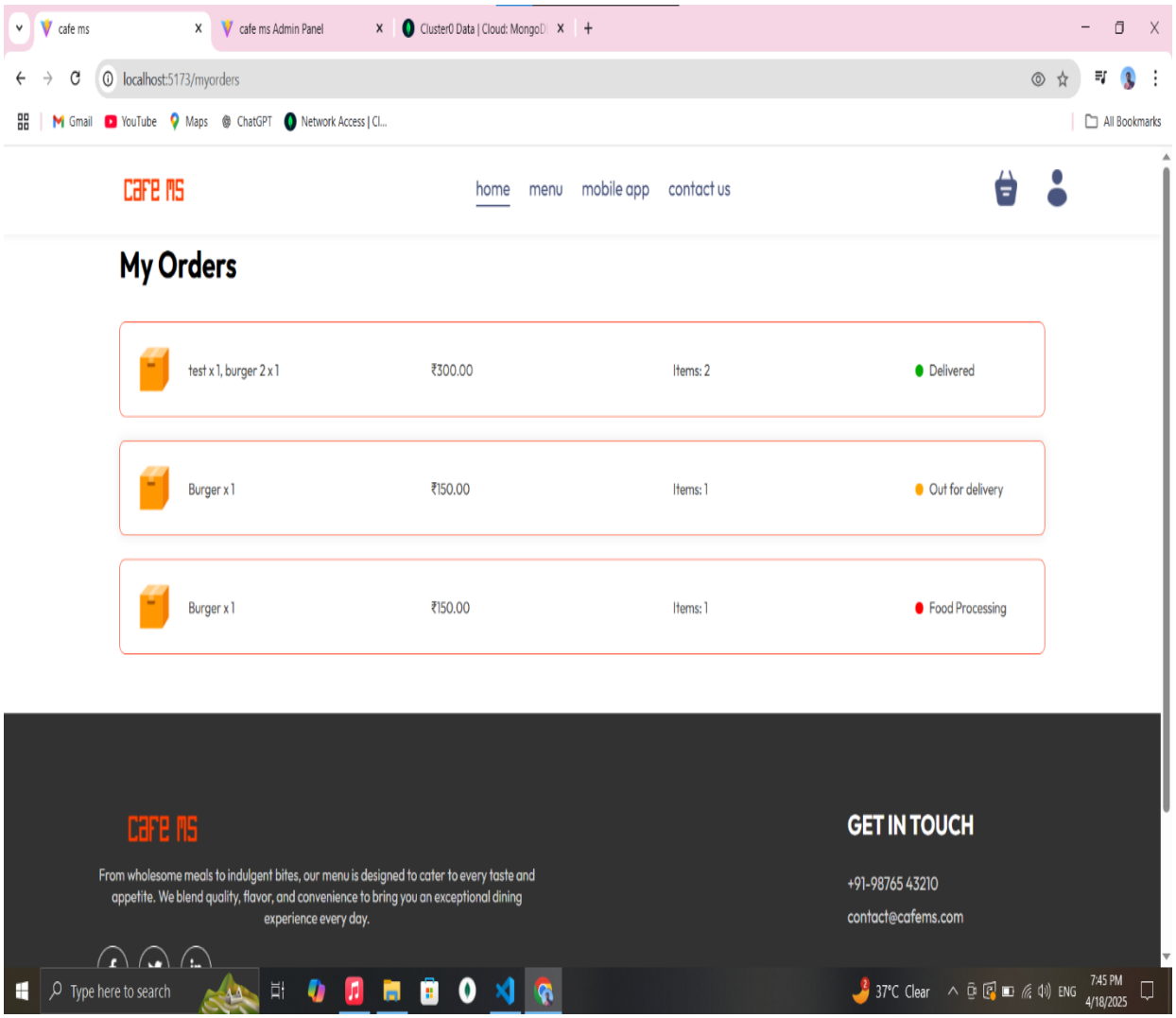

Figure 9 Order-Tracking Page

Shows **live order status updates**, e.g., “Order Received,” “Preparing,” “Ready,” etc.

- Backend updates order status, which is fetched using Axios in intervals.
- Enhances transparency and mimics real-world order tracking systems.

6.3.7 Orders Page:

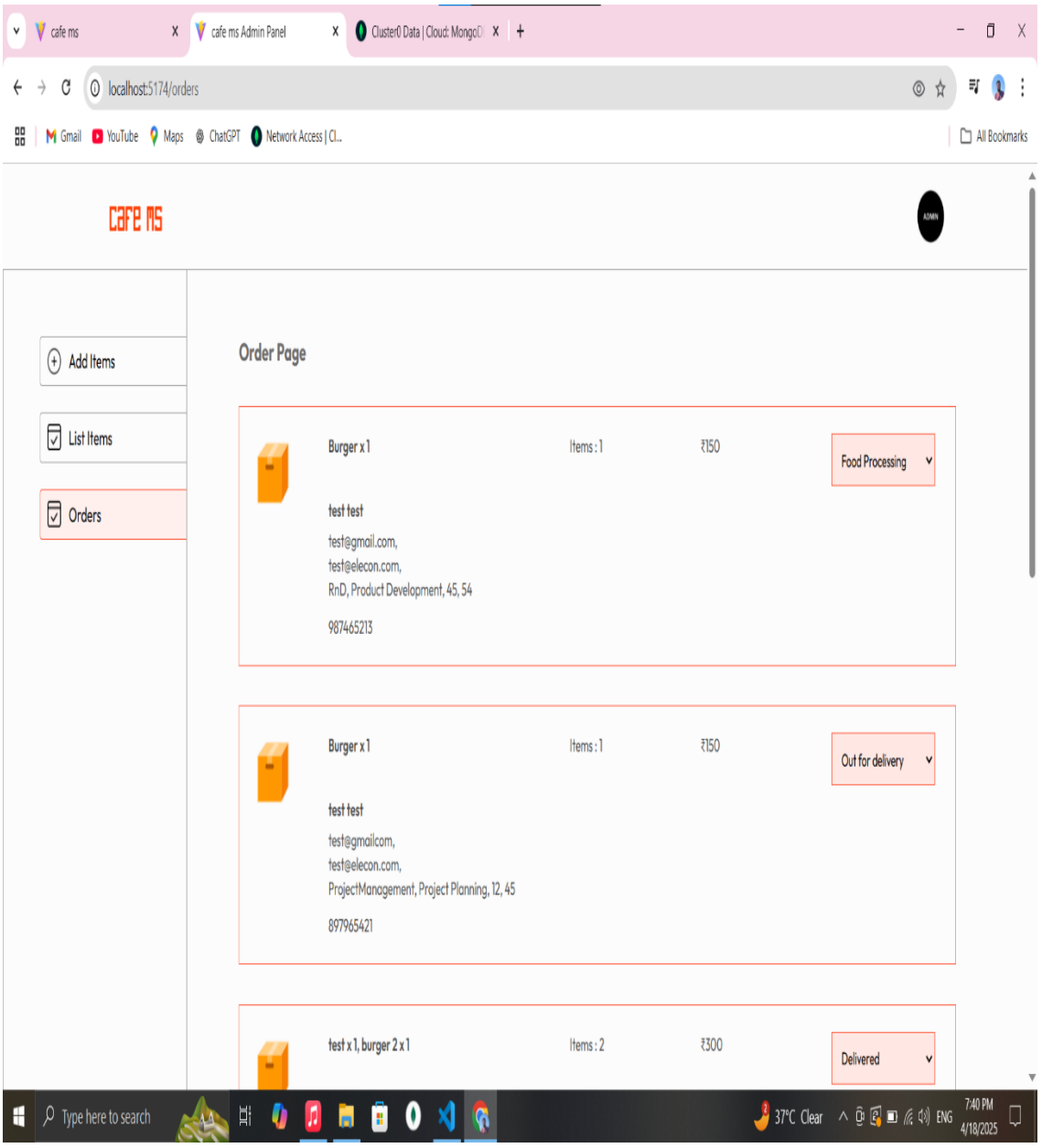

Figure 10 Orders Page

This page lists **past orders placed by the logged-in user**, fetched from MongoDB.

- Includes details like item names, total cost, status, and date.
- Helps users review their history and check previous transactions.

6.3.8 List Item Page:

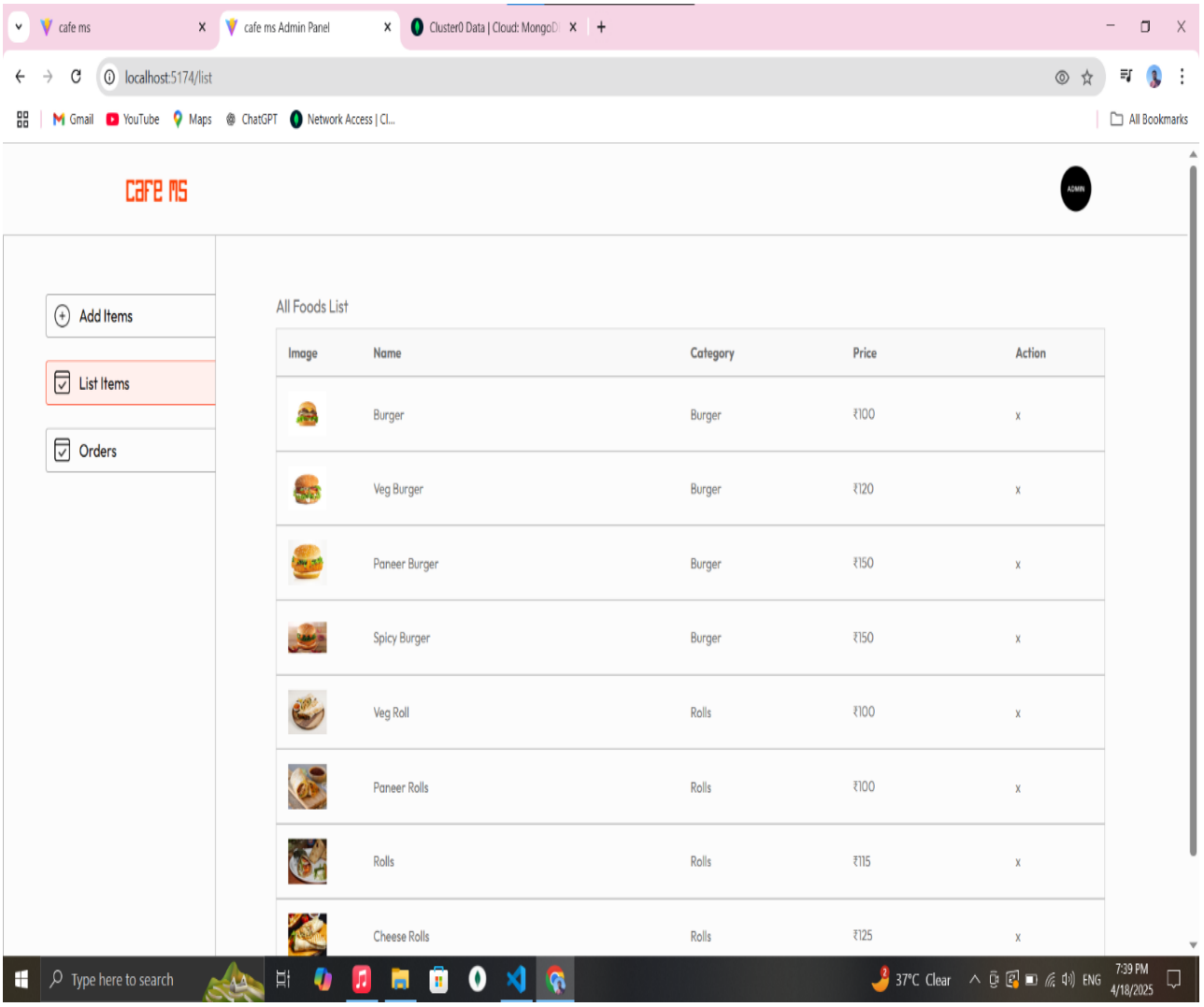

Figure 11 List Item Page

This is part of the **Admin Dashboard**.

- Admin can view, edit, or delete menu items from the database.
- It allows the cafeteria staff to maintain the menu dynamically.
- Uses secure routes and protected access.

6.3.9 Contact Page:

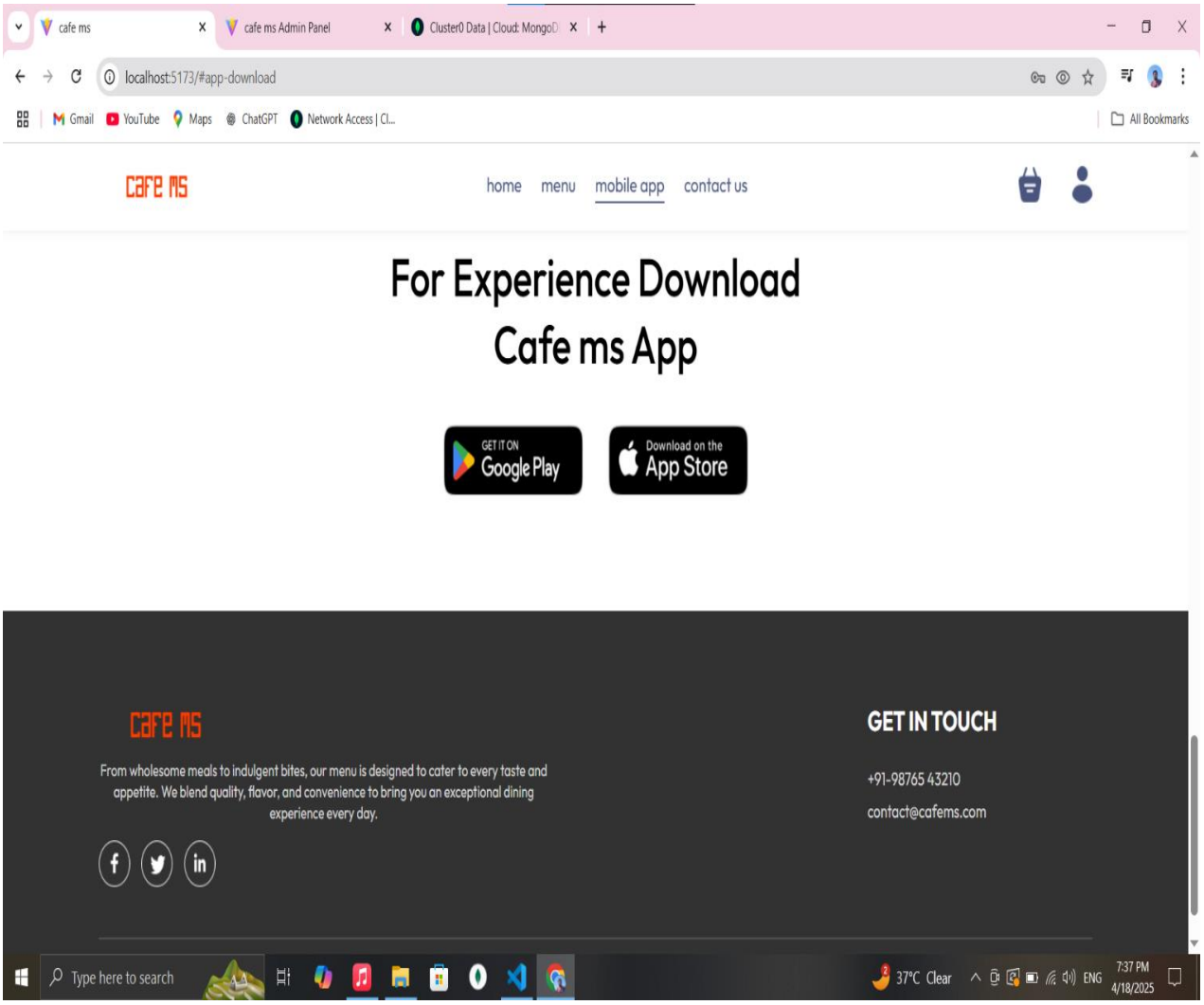

Figure 12 Contact Page

Provides users with a way to submit queries or feedback.

- Likely includes a form storing input locally or forwarding it via API to backend.
- A future scope here is integrating an email API or a feedback dashboard.

CHAPTER: 7 CONCLUSION AND DISCUSSION

7.1 LEARNING OUTCOMES

The Cafeteria Management System project was a significant learning experience that allowed for the practical application of full-stack web development skills. One of the major outcomes was gaining hands-on experience with **React.js**, a powerful frontend library, which helped in understanding component-based architecture, state management, routing, and reusability in web interfaces. Using **Vite** as the build tool made the development process faster and more efficient, especially with features like hot module replacement and rapid build times. On the backend, working with **Node.js** and **Express.js** helped in understanding how servers handle HTTP requests, manage routes, and connect with databases.

The project also involved learning how to store and retrieve data using **MongoDB**, a NoSQL database. This deepened the understanding of database design, schema structuring, and CRUD operations. Integration between frontend and backend was achieved using **Axios**, which offered a real-world understanding of handling asynchronous operations and API error handling. Additionally, the project highlighted the importance of a clean and responsive user interface, which required attention to **UI/UX design principles**, use of **Tailwind CSS** for styling, and ensuring cross-device compatibility.

Apart from technical skills, this project also improved soft skills such as **project planning**, **time management**, and **problem-solving**. Debugging errors, managing code versions with Git, and testing different modules enhanced logical thinking and attention to detail. Overall, it provided a comprehensive learning journey across both the frontend and backend of web application development.

7.2 EXPERIENCE

The overall experience of developing the Cafeteria Management System was both enriching and transformative. From the initial planning phase to the final deployment, every stage offered new challenges and opportunities to learn. One of the highlights was understanding how a real-world application is structured and how different components of a system work together to create a smooth and user-friendly experience. The hands-on implementation of login/signup systems, order management, and the admin dashboard helped solidify key development concepts.

Working on this project also brought several technical challenges, such as integrating frontend with backend APIs, maintaining proper state across different components, handling errors gracefully, and ensuring smooth navigation through routes. Each challenge was an opportunity to research, experiment, and grow as a developer. It also offered exposure to best practices in **code organization**, **modular development**, and **data security** through techniques like password hashing and secure data transmission.

Moreover, the experience of using modern tools and platforms such as **GitHub** for version control, **MongoDB Compass** for database management, and **Heroku or Vercel** for deployment gave a more industry-like experience. The journey not only improved technical proficiency but also boosted self-confidence and increased enthusiasm for building more real-world applications in the future. Overall, it was a fulfilling experience that helped bridge the gap between academic learning and practical development.

7.3 CONCLUSION

The **Cafeteria Management System** project successfully accomplished its objective of designing and developing a user-friendly, efficient, and digital platform to manage daily cafeteria operations. By replacing the outdated and error-prone manual system, the new solution significantly improved the speed, accuracy, and overall experience for both customers and cafeteria staff. The system effectively integrated key functionalities such as **user authentication**, **order placement and tracking**, **admin dashboard for order and menu management**, and a **simulated payment system**—making it a valuable step forward toward automation.

Throughout the development process, modern technologies such as **React.js**, **Vite**, **Node.js**, **Express.js**, and **MongoDB** were utilized to build a robust and scalable full-stack web application. These tools provided the flexibility and performance required to handle dynamic data, maintain secure user sessions, and enable smooth communication between the frontend and backend. Moreover, **Axios** facilitated seamless API integration, while thoughtful UI design enhanced the user experience on both desktop and mobile devices.

Although some advanced features like **QR code-based ordering**, **real-time order status updates**, and a **feedback mechanism** were not included in the current version due to time and scope constraints, the existing system lays a strong foundation for future enhancements. The modular and scalable architecture ensures that such features can be integrated with minimal refactoring.

This project not only demonstrated the application of full-stack development concepts in a real-world scenario but also highlighted the transformative impact of technology on traditional service systems. It reflects the potential of digital solutions to bring structure, transparency, and efficiency into routine business operations. Overall, the Cafeteria Management System is a practical and scalable application that opens the door to further innovation and real-time automation in similar domains.

.

REFERENCES

1. **React.js Official Documentation** – https://react.dev/
2. **Redux Documentation** – https://redux.js.org/
3. **Vite Documentation** - Getting Started | Vite
4. **Node.js Official Documentation** – https://nodejs.org/en
5. **Express.js Guide** – https://expressjs.com/
6. **MongoDB Documentation** – https://www.mongodb.com/docs/
7. **Axios Documentation** – Getting Started | Axios Docs
8. **Tailwind CSS Docs -**Installing Tailwind CSS with Vite - Tailwind CSS
9. **MDN Web Docs for JavaScript and Web Technologies** – https://developer.mozilla.org/
10. **Bootstrap CSS Framework** – https://getbootstrap.com/
11. **GitHub and Stack Overflow Discussions for Troubleshooting**
12. **Research Papers and Articles on Smart Canteen Systems and Web-Based Order Management**
13. **API Documentation for Payment Gateway and QR Code Integration**