SMART CANTEEN SYSTEM

A PROJECT REPORT

Submitted by

RAJEV SHREE U	**2303811710422130**
ROHINI D	**2303811710422131**
SAGANA S	**2303811710422134**
SINDHUJA S	**2303811710422150**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

CSB1303 – OBJECT ORIENTED ANALYSIS AND DESIGN

In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621112

DECEMBER - 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS)
SAMAYAPURAM - 621112

BONAFIDE CERTIFICATE

The work embodied in the present project report entitled " **SMART CANTEEN SYSTEM** " has been carried out by the students **RAJEV SHREE U, ROHINI D, SAGANA S, SINDHUJA S**. The work reported here in is original and we declare that the project is their own work, except where specifically acknowledged, and has not been copied from other sources or been previously submitted for assessment.

Date of Viva Voce: ……………………

Mrs. V. KALPANA, M.E., (Ph.D.,)
SUPERVISOR
Assistant Professor
Department of CSE
K. Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112.

Mr. R. RAJAVARMAN M.E., (Ph.D.,)
HEAD OF THE DEPARTMENT
Assistant Professor (Sr. Grade)
Department of CSE
K. Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112.

INTERNAL EXAMINER

EXTERNAL EXAMNIER

ABSTRACT

The Smart Canteen Management System is an advanced digital platform developed to transform traditional canteen operations into a faster, more efficient, and user-friendly process. It replaces manual ordering and billing methods with an automated system that supports real-time data handling, accurate menu management, and smooth customer interactions. Customers can easily register or log in, browse a wide range of meal options, filter by categories such as vegetarian or non-vegetarian, and add selected items to their cart. The system ensures convenience by allowing users to review their selections, modify quantities, and proceed seamlessly to payment and order confirmation. For administrators, the system offers a powerful set of tools to manage the entire backend, including adding new items, removing outdated ones, updating prices, monitoring orders, and maintaining inventory records. All information is stored in a centralized database, enabling quick retrieval, error-free updates, and consistent data flow across different modules.

Keywords: Smart Canteen System, Online Food Ordering System, Canteen Management Application, Object-Oriented Analysis and Design (OOAD), Web-Based Application, SQL Database, User Authentication, Menu Management, Order Placement, Inventory Management, Cashless Payment System, Secure Transactions, Order Tracking, Report Generation, Notification System.

ACKNOWLEDGEMENT

We thank our **Dr. N. VASUDEVAN**, Principal, for his valuable suggestions and support during the course of my research work.

We thank our **Mr. R. RAJAVARMAN**, Head of the Department, Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, for his valuable suggestions and support during the course of my research work.

We wish to record my deep sense of gratitude and profound thanks to my Guide **Mrs. V. KALPANA,** Assistant Professor , Department of Computer Science and Engineering**,** for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this thesis into fruition.

We are extremely indebted to our project coordinator **Mrs. V. KALPANA,** Assistant Professor, Department of Computer Science and Engineering, for her valuable suggestions and support during the course of my research work.

We also thank the faculty and non-teaching staff members of the Department of Computer Science And Engineering, K. Ramakrishnan College Of Technology, Samayapuram, for their valuable support throughout the course of my research work.

Finally, we thank our parents, friends and our well wishes for their kind support.

SIGNATURE

____

____

____

____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SRS	–	System Requirement Specification
UML	–	Unified Modeling Language
UX	–	User Experience
API	–	Application Programming Interface
OTP	–	One-Time Password
PNG	–	Portable Network Graphics
SQL	–	Structured Query Language

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO DOMAIN

The domain of a Smart Canteen System focuses on modernizing and automating traditional canteen operations within environments such as educational institutions, corporate offices, and public organizations. Conventional canteen processes often involve manual ordering, long queues, slow service, and inefficient inventory tracking. To address these issues, the Smart Canteen System integrates digital ordering, automated billing, real-time stock monitoring, and data-driven management to enhance both user convenience and operational efficiency. The system interacts with various stakeholders, including customers, staff, chefs, administrators, and suppliers, and models real-world entities such as menu items, orders, payments, and inventory using object-oriented principles. By streamlining key processes like order placement, menu browsing, queue management, and report generation, this domain emphasizes the need for accuracy, speed, automation, and user-friendly interfaces. Through OOAD, the Smart Canteen System can be developed as a scalable, modular, and maintainable solution that significantly improves the overall functioning of canteen services.

1.2 PROBLEM DESCRIPTION

Traditional canteen operations often suffer from several inefficiencies, including long queues, delayed service, manual order handling, and poor inventory management. Customers frequently experience long waiting times, lack of real-time information about menu availability, and delays in receiving their orders. Canteen staff also face challenges such as managing peak-time rush, tracking stock levels accurately, preventing order confusion, and maintaining proper billing records. These issues lead to reduced customer satisfaction, increased workload for staff, and potential financial losses due to errors or wastage. The lack of automation and real-time coordination between ordering, payment, and food preparation processes highlights the need for a Smart Canteen System that can streamline operations, enhance service efficiency, and provide a smooth, technology-driven experience for both customers and administrators. On the management side, staff struggle to maintain accurate records of daily sales, track inventory levels, and predict demand efficiently, often resulting in food wastage or shortages. Additionally, manual billing increases the chances of calculation errors and lacks transparency, creating dissatisfaction and difficulties in maintaining financial reports. The absence of automation further limits the ability to generate meaningful insights about peak hours, top-selling items, customer preferences, and resource requirements. These challenges collectively highlight the need for a Smart Canteen System that integrates digital ordering, automated billing, real-time inventory updates, and centralized management to ensure faster service, reduced waiting time, enhanced user satisfaction, and better operational control for administrators.

1.3 OBJECTIVE OF THE PROJECT

The primary objective of the Smart Canteen System project is to design and develop an efficient, user-friendly, and automated platform that streamlines the entire canteen operation— from ordering and billing to inventory management and service delivery. The system aims to eliminate the drawbacks of traditional manual processes by enabling customers to browse menus digitally, place orders quickly, and make secure digital payments without waiting in long queues. It also seeks to support canteen staff with real-time order notifications, accurate billing, and automated stock tracking to reduce errors, minimize wastage, and improve resource utilization. The Smart Canteen System is to design and develop an efficient, user-friendly, and automated platform that streamlines canteen operations from ordering and billing to inventory management and service delivery. The system aims to reduce customer waiting time by enabling quick digital ordering and secure cashless payments, thereby eliminating long queues and manual billing errors. It improves service efficiency through automated order handling and clear communication with kitchen staff, ensuring faster food preparation and delivery. Real-time inventory tracking helps prevent shortages, overstocking, and food wastage while maintaining accurate records of orders and stock. The system also enhances customer convenience by allowing users to view menus, check availability, and place orders easily.

1.4 SCOPE OF THE PROJECT

The scope of the Smart Canteen System includes designing and developing a digital platform that simplifies and automates all major activities of a canteen. The system will allow customers to browse the menu, check item availability, place orders, and make payments online. It will also support canteen staff by displaying incoming orders instantly, helping them prepare and serve food more efficiently. The system will manage inventory by updating stock levels automatically after every order, reducing the chances of shortages and wastage. Administrators will be able to view daily sales reports, track orders, update menu items, set prices, and monitor overall canteen performance. The project covers the creation of user interfaces for customers, staff, and administrators, along with secure login and data management.

CHAPTER 2
SYSTEM REQUIREMENT SPECIFICATION (SRS)

2.1 FUNCTIONAL REQUIREMENTS

2.1.1 USER REGISTRATION AND AUTHENTICATION

The smart canteen system allows users to register and log in securely using credentials such as email, phone number, student/employee ID, or RFID. Authentication ensures that only authorized users can access the system and place orders. It also helps in maintaining personalized order history and secure payment records.

2.1.2 DIGITAL MENU MANAGEMENT

Digital menu management is a key functional component of the smart canteen system that allows the canteen to display and manage food items electronically instead of using traditional printed menus. The system enables administrators to add, update, or remove menu items easily, including details such as item name, price, description, availability, and category. Any changes made to the menu are reflected in real time across all user and staff interfaces, ensuring that customers always view the most accurate and updated information.

2.1.3 ONLINE FOOD ORDERING

The Users can place food orders through a mobile app or web interface by selecting items and quantities. The system allows modification of orders before confirmation and supports both instant orders and pre-orders. This reduces waiting time and improves order accuracy.

2.1.4 ORDER TRACKING

Users mustOnce an order is placed, users can track its status, such as order placed, preparing, ready for pickup, or completed. Real-time updates improve transparency and help users manage their time efficiently. The system must confirm the order status. Users should be able to track their orders, including Processing, Shipped, and Delivered. Order history must be accessible to all registered users.

2.1.5 CASHLESS PAYMENT SYSTEM

The smart canteen system supports digital payment methods such as UPI, debit/credit cards, campus wallets, or prepaid accounts. Cashless payments reduce manual handling of money and speed up the checkout process. Payment confirmation is automatically linked to the order.

2.2 NON-FUNCTIONAL REQUIREMENTS

2.2.1 PERFORMANCE REQUIREMENTS

The smart canteen system should respond quickly to user actions such as login, menu loading, and order placement. Orders must be processed in real time without delays, even during peak hours like lunch breaks. The system should support multiple users simultaneously without performance degradation. Overall system performance should support smooth and uninterrupted ordering.

2.2.2 SECURITY REQUIREMENTS

User data, payment information, and transaction records must be protected using secure authentication and encryption techniques. Only authorized users should be allowed access based on their roles (user, staff, admin). The system should prevent unauthorized access, data leakage, and fraud.

2.2.3 USABILITY REQUIREMENTS

The user interface should be simple, intuitive, and easy to navigate for all users, including students and staff with minimal technical knowledge. Clear menus, buttons, and instructions should be provided to reduce errors and training time.The system should also support error prevention and recovery. For example, confirmation messages should be shown before final order placement, and meaningful error messages should guide users if something goes wrong, such as payment failure or unavailable items. Users should be able to easily modify or cancel orders within allowed time limits.

2.2.4 RELIABILITY REQUIREMENTS

Reliability refers to the ability of the smart canteen system to function correctly and consistently over time without failures. The system should ensure that all operations such as order placement, payment processing, and inventory updates are performed accurately. Once an order or payment is confirmed, the data must be stored safely without loss or duplication.

2.2.5 SCALABILITY REQUIREMENTS

The system architecture should be designed to allow easy expansion, such as adding more food items, integrating additional payment gateways, or supporting multiple canteens under the same platform. Scalability also enables future enhancements like mobile apps, analytics modules, or AI-based demand forecasting.

2.3 HARDWARE REQUIREMENTS

2.3.1 SERVER HARDWARE

The server requires at least an Intel i3 or higher processor to ensure smooth system operations. A central server is required to store and manage data such as user information, menu details, orders, payments, and reports. The server should have sufficient processing power, memory, and storage to handle multiple simultaneous users and real-time transactions reliably.

2.3.2 CLIENT SYSTEM

Client devices such as smartphones, tablets, or desktop computers are required for users and canteen staff to access the system. Customers use these devices to place orders, while staff use them to manage orders and update order status. These devices should support modern web browsers or mobile applications.

2.3.3 INTERNET CONNECTIVITY

The system A reliable internet connection is essential for real-time order processing, payment transactions, and system synchronization. High-speed broadband or Wi-Fi should be available throughout the canteen area to ensure uninterrupted service.HTML5.

2.3.4 KITCHEN DISPLAY SYSTEM

The Kitchen Display System (KDS) is a critical hardware and functional component of the smart canteen system that is used by kitchen staff to manage and prepare food orders efficiently. It replaces traditional paper-based order slips with a digital, real-time display of customer orders, thereby reducing errors and improving preparation speed.The KDS displays complete order details such as order ID, food items, quantities, customization requests, and pickup time.

2.3.5 PAYMENT HARDWARE

Optional payment hardware such as QR code scanners, card readers, or POS machines may be used to support digital payments. These devices help in fast and secure transaction processing.

2.4 SOFTWARE REQUIREMENTS

2.4.1 OPERATING SYSTEM

The smart canteen system requires a stable and reliable operating system to support smooth execution of all software components. Server-side operations may run on operating systems such as Windows Server or Linux, which provide strong performance, security, and multitasking capabilities. Client-side systems, including user devices and staff terminals, should support Android, iOS, or web-based platforms to ensure wide accessibility. Regular updates should be enabled for better security and performance.

2.4.2 FRONT-END TECHNOLOGIES

Front-end technologies are responsible for providing the user interface through which customers, staff, and administrators interact with the smart canteen system. Technologies such as HTML, CSS, and JavaScript, along with modern frameworks like React, Angular, or Flutter, enable the creation of responsive and visually appealing interfaces. The front end should be designed to work smoothly across different screen sizes and devices.

2.4.3 BACK-END TECHNOLOGIES

The back end forms the core of the smart canteen system by handling business logic, data processing, and communication between the front end and the database. Server-side technologies such as Java, Python, Node.js, or PHP are used to process user requests, manage authentication, handle order workflows, and ensure secure transactions. The back end must be efficient, scalable, and capable of handling multiple requests simultaneously, especially during peak hours.

2.4.4 WEB SERVER

The smart canteen system requires a web or application server to host the application and manage incoming requests from users and staff. Servers such as Apache, Nginx, or Tomcat handle client requests, route them to the appropriate back- end services, and deliver responses efficiently. The server must be configured for high availability, security, and optimal performance to support real-time order processing and user interactions.

2.4.5 SECURITY SOFTWARE

Security software components are essential to protect sensitive user information and system data from unauthorized access. This includes authentication systems, role- based access control, and encryption technologies such as SSL/TLS. Security software ensures that only authorized users can perform specific actions within the system. Regular security updates and monitoring are necessary to safeguard against cyber threats and data breaches.

2.5 USER CHARACTERTICS

The users of the smart canteen system consist of students, employees, canteen staff, and system administrators, each with varying levels of technical knowledge and system interaction. Most users, such as students and employees, are expected to have basic familiarity with smartphones and web applications, allowing them to easily browse menus, place orders, and make digital payments. These users typically expect quick response times, simple navigation, and minimal steps to complete tasks, especially during peak hours. Canteen staff users interact with the system more frequently and require a clear and efficient interface to manage incoming orders, update order statuses, and coordinate food preparation without disrupting kitchen workflow.

2.6 CONSTRAINTS

The smart canteen system is developed and operated under several technical, operational, and organizational constraints that influence its design and implementation. One major constraint is hardware availability, as the system must function effectively using existing devices such as smartphones, tablets, kitchen displays, and standard servers without requiring expensive or specialized equipment. Network dependency is another significant constraint, since real-time order processing, order tracking, and digital payment transactions rely heavily on stable internet connectivity; any interruption in the network may temporarily affect system performance or availability. Budget limitations also act as a constraint, restricting the use of advanced technologies, third-party services, or high-end infrastructure. As a result, the system must be designed using cost-effective hardware and open-source or widely available software tools. The system must comply with institutional policies, data protection rules, and digital payment regulations, which may limit certain features or data storage practices. These constraints require careful system planning and design to ensure that the smart canteen system remains reliable, secure, and user-friendly while meeting organizational requirements.

CHAPTER 3

ANALYSIS AND DESIGN

3.1 USE CASE DIAGRAM

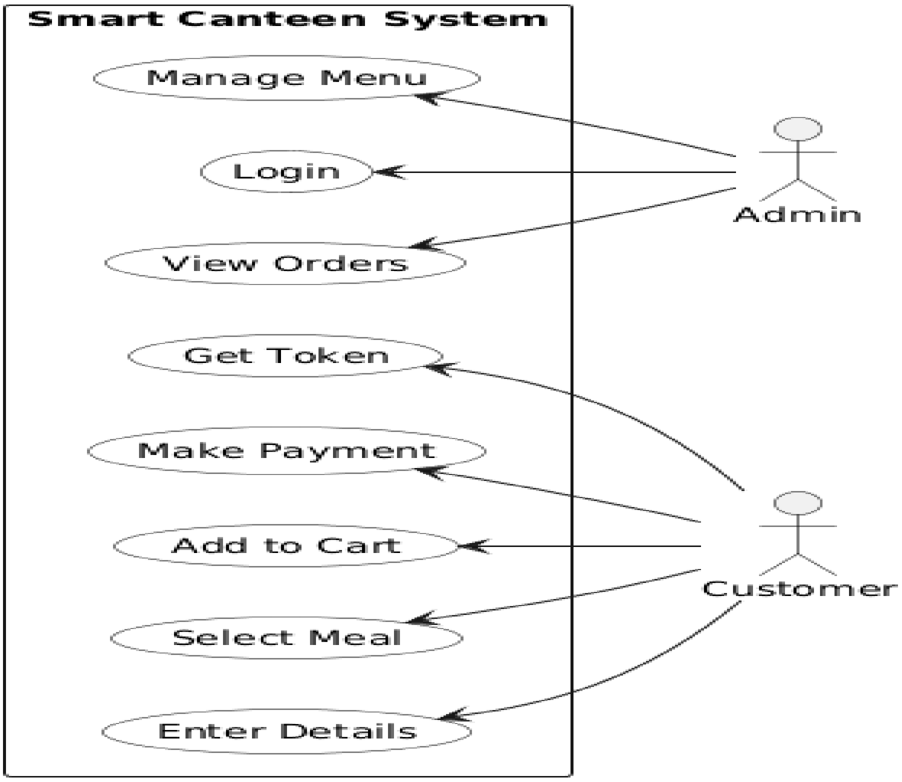


Figure 3.1 Use Case Diagram

3.1.1 USE CASE DESCRIPTION

The diagram represents a Use Case Diagram for a Smart Canteen System, showing interactions between the system and its users. There are two main actors: Admin and Customer. The Admin is responsible for managing the system, which includes logging in, managing the menu by adding or updating items, and viewing all customer orders. The Customer interacts with the system to place orders, starting with entering personal details and selecting meals from the menu. Once the meals are selected, they can be added to a cart and payment can be made through the system. After payment, the customer receives a token for order collection. The lines connecting actors to use cases illustrate these interactions clearly.

3.2 CLASS DIAGRAM

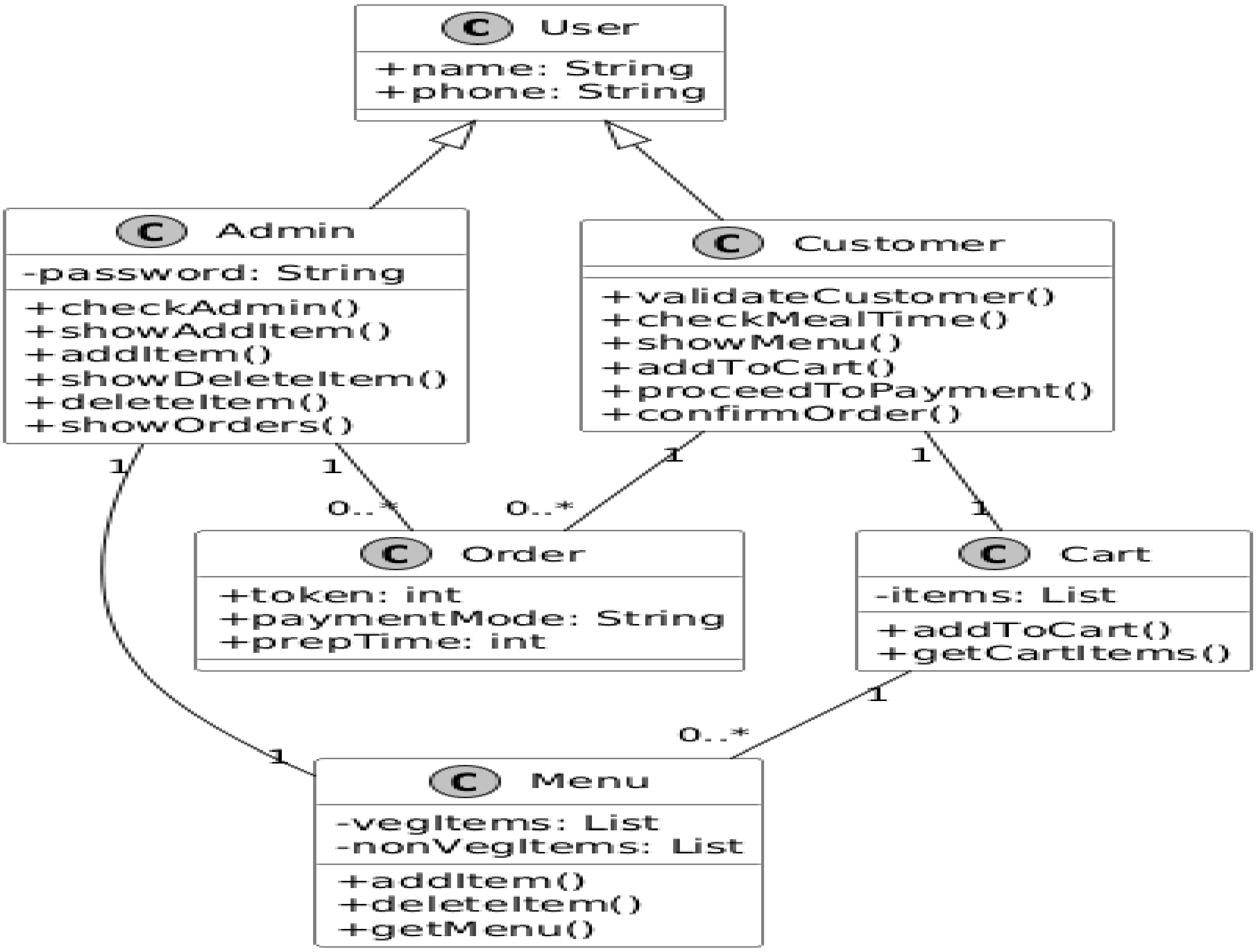


Figure 3.2 Class Diagram

3.2.1 CLASS DIAGRAM DESCRIPTION

The class diagram for the Smart Canteen Management System shows the main entities and their relationships. The User class is the parent, with subclasses Admin and Customer. Admin manages the menu and orders using methods like addItem(), deleteItem(), and showOrders(). Customer can select meals, add items to the Cart, and proceed to Order with methods like addToCart() and confirmOrder(). The Menu class stores veg and non-veg items, while Order keeps details like token, paymentMode, and prepTime. Cart maintains selected items for the customer. The diagram shows inheritance, associations, and multiplicities, clearly representing how users, menu, cart, and orders interact in the system.

3.3 ACTIVITY DIAGRAM

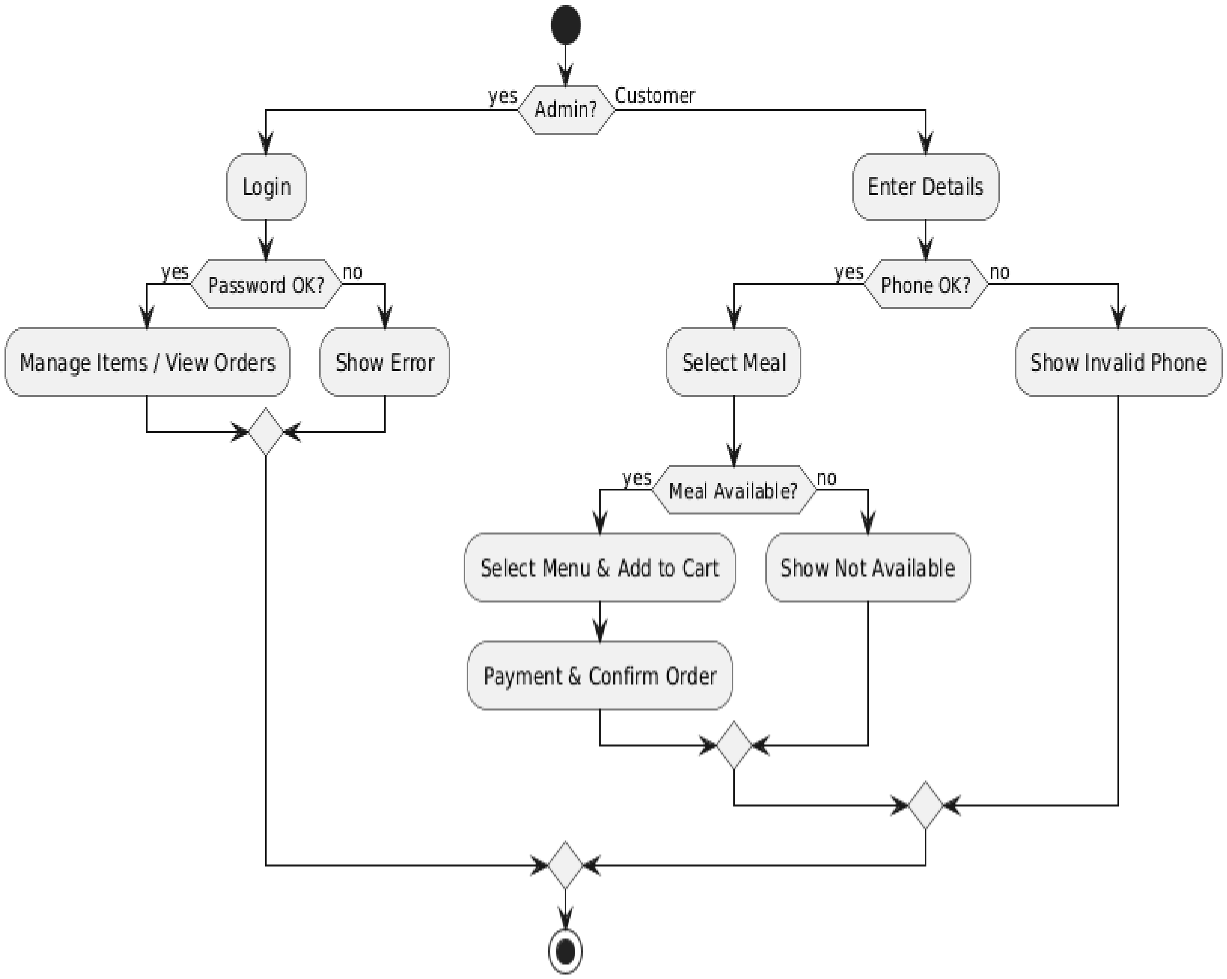


Figure 3.3 Activity Diagram

3.3.1 ACTIVITY DIAGRAM DESCRIPTION

The diagram represents the flow of a Smart Canteen Management System. It starts with identifying whether the user is an admin or a customer. If the user is an admin, they must log in and the system checks if the password is correct. Upon successful login, the admin can manage items or view orders; if the password is incorrect, an error message is displayed. For customers, they first enter their details and the system validates the phone number. If the phone number is invalid, an error message is shown. Once verified, the customer can select a meal, and the system checks its availability. If the meal is available, it is added to the cart, followed by payment and order confirmation. If the meal is not available, a “not available” message is displayed.

3.4 SEQUENCE DIAGRAM

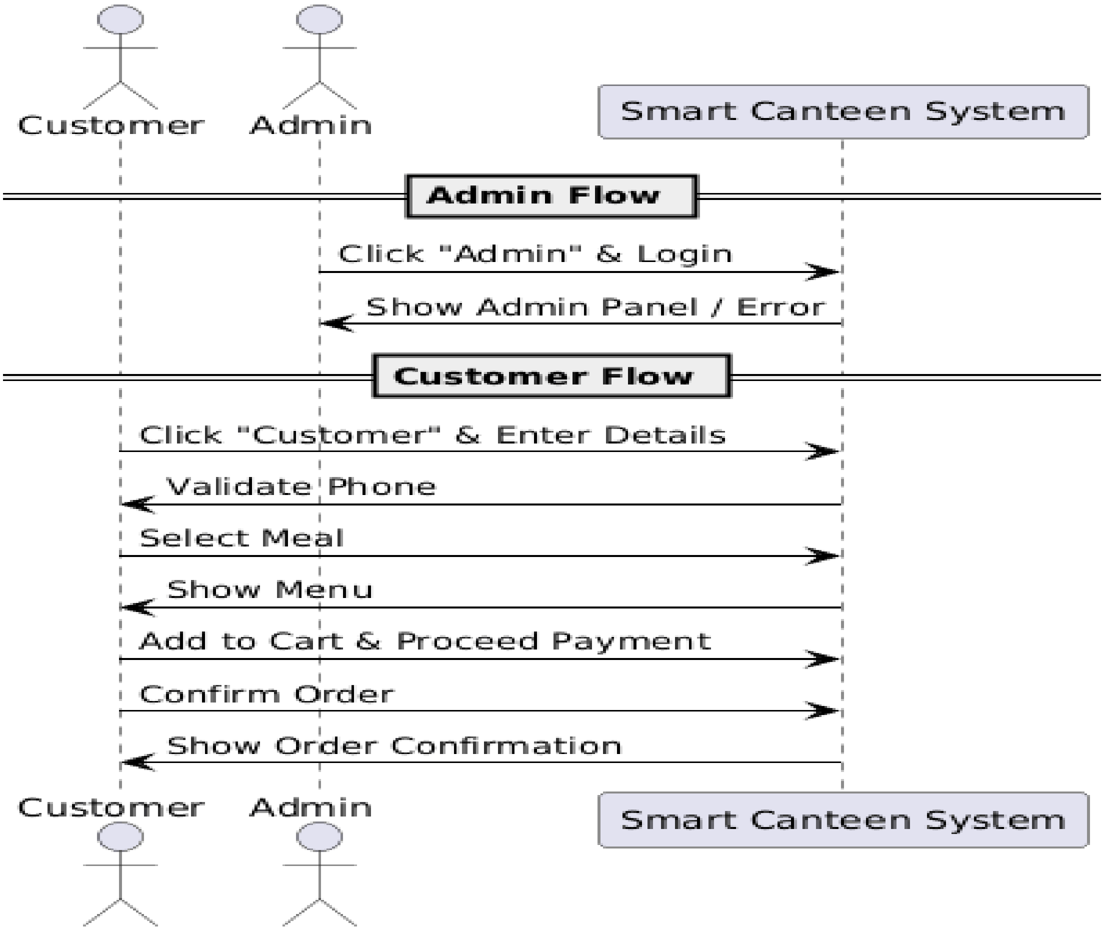


Figure 3.4 Sequence Diagram

3.4.1 SEQUENCE DIAGRAM DESCRIPTION

The sequence diagram shows how a customer interacts with the Smart Canteen System from login to order confirmation. The customer first logs into the system, which verifies the credentials and confirms successful login. The customer then views the menu, and the system retrieves and displays item details from the Menu component. When the customer places an order, the system checks item availability with the Menu and receives confirmation. After this, the system sends a payment request to the Payment Gateway, which processes the payment and returns a success message. Finally, the system confirms the order to the customer, completing the entire ordering and payment workflow.

3.5 STATE MACHINE DIAGRAM

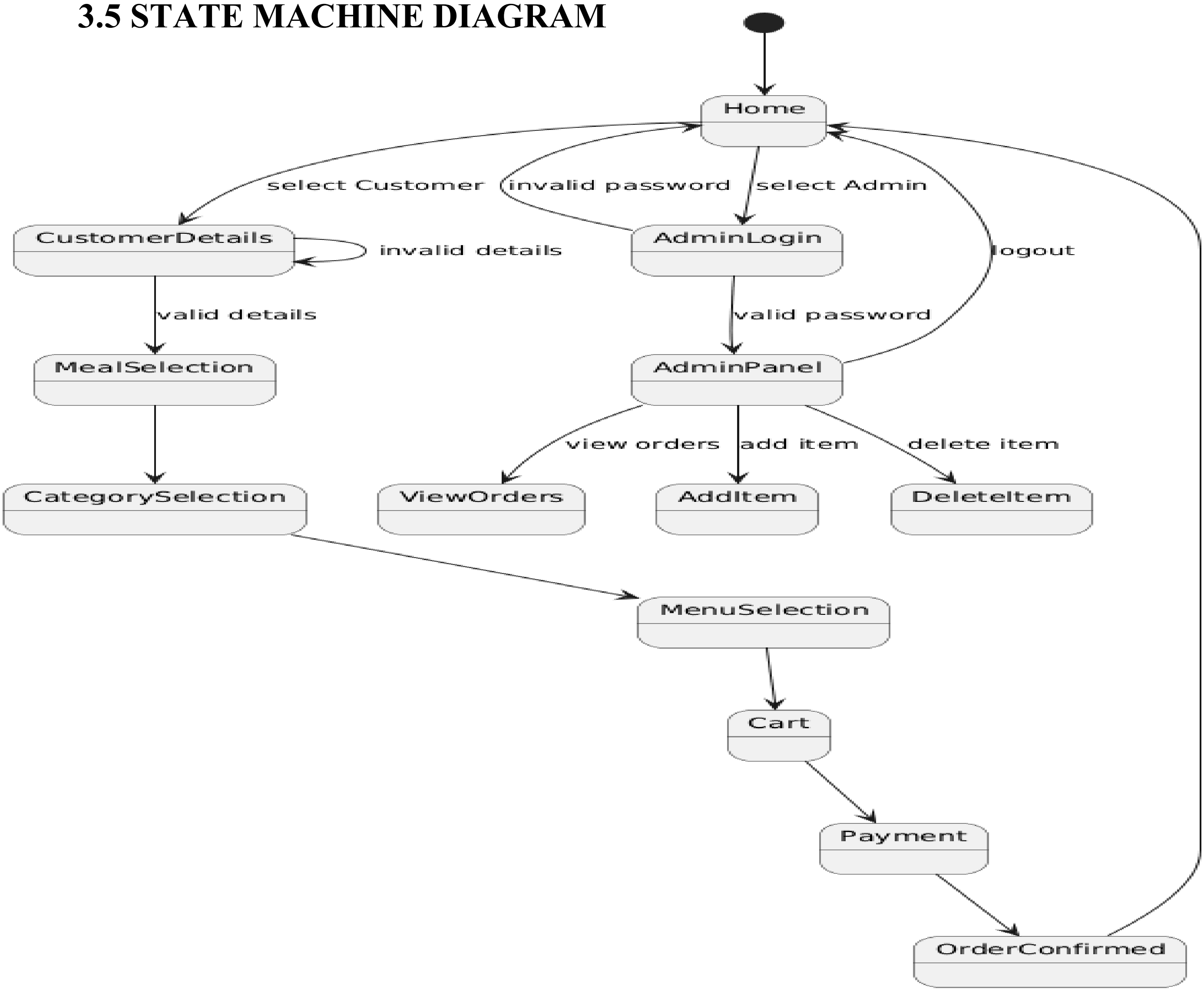


Figure 3.5 State Machine Diagram

3.5.1 STATE MACHINE DIAGRAM DESCRIPTION

The state machine diagram shows how the Smart Canteen system moves between different states based on user actions. The system begins at the Home state, where users choose either Admin or Customer roles. Admins log in to access the Admin Panel and manage items or view orders, then return to Home. Customers enter details, select meals, add items to the cart, and proceed to payment. After order confirmation, the system returns to the Home state, completing the workflow.

3.6 COMPONENT DIAGRAM

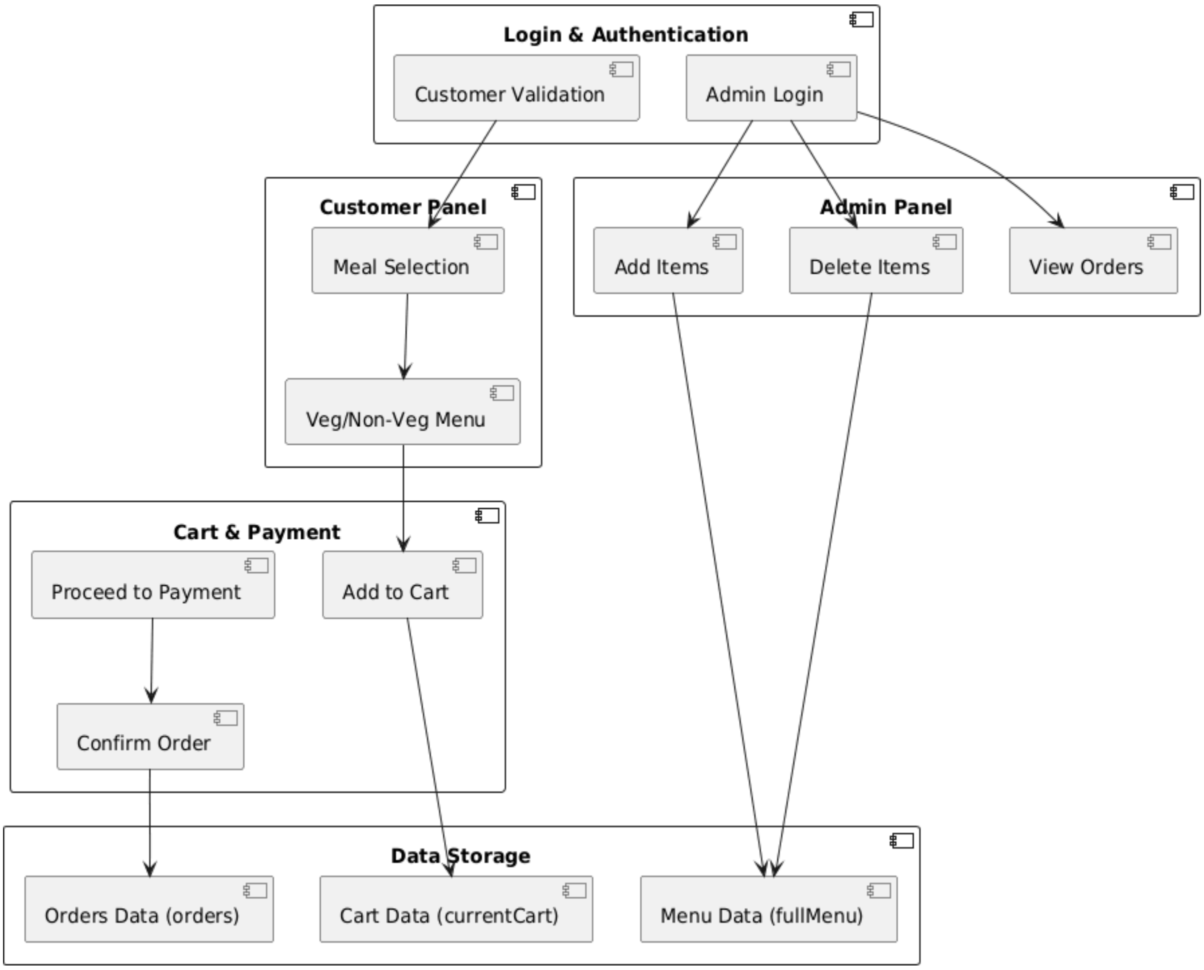


Figure 3.6 Component Diagram

3.6.1 COMPONENT DIAGRAM DESCRIPTION

The diagram shows the modular structure of the Smart Canteen System. It starts with the Login Module for admin authentication and customer verification. After login, admins access the Admin Panel to manage items and view orders. Customers move to the Customer Section to select meals and categories. Both admin and customer activities connect to the Meal Management module. This module handles cart operations and meal details. The process then moves to Order & Payment for completing purchases. Overall, the system separates roles and ensures smooth order completion.

3.7 DEPLOYMENT DIAGRAM

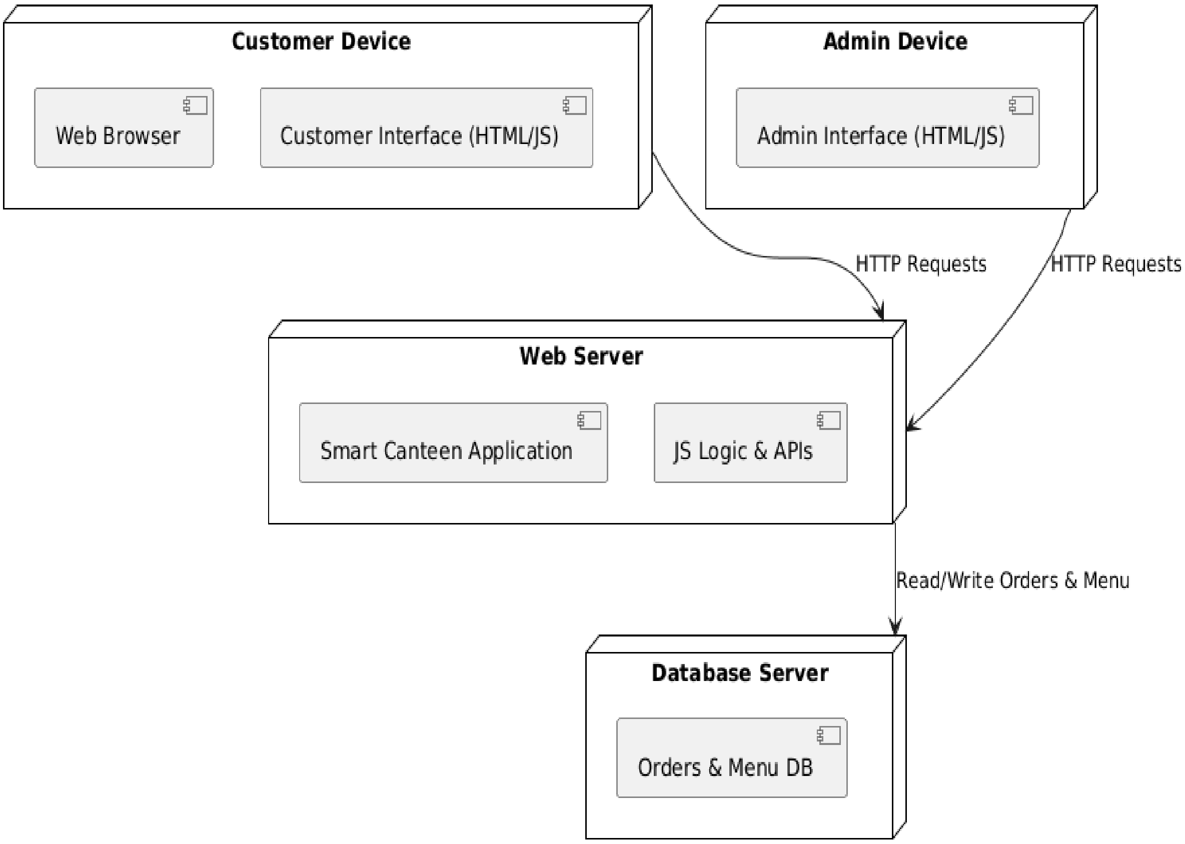


Figure 3.7 Deployment Diagram

3.7.1 DEPLOYMENT DIAGRAM DESCRIPTION

The deployment diagram shows how the Smart Canteen System is physically deployed across different devices and servers. The Customer Device runs the web or mobile app interface, which sends HTTP requests to the Canteen Server. This server hosts the Application Backend, responsible for processing user actions, managing orders, and interacting with the Database to store or retrieve data. When a payment is initiated, the backend communicates with the Payment Gateway Server, where the Payment Processor handles the transaction securely. Once the payment result is returned, the system updates the database and completes the order process.

3.8 PACKAGE DIAGRAM

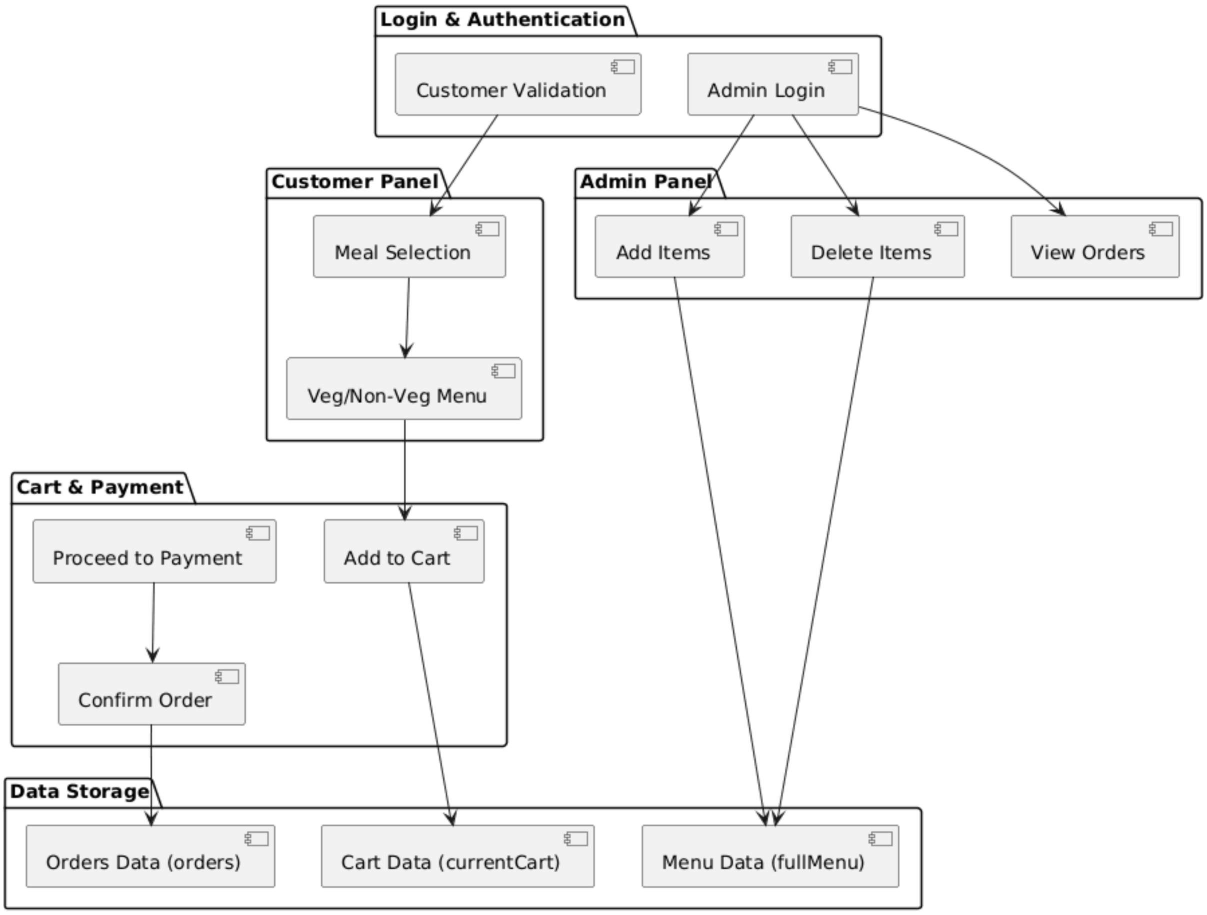


Figure 3.8 Package Diagram

3.8.1 PACKAGE DIAGRAM DESCRIPTION

The package diagram organizes the Smart Canteen Management system into five modules: Login & Authentication, Admin Panel, Customer Panel, Cart & Payment, and Data Storage. Admins log in to manage menu items and view orders, while customers validate their details to select meals and categories. The Cart & Payment module handles adding items, processing payment, and confirming orders. Data Storage keeps track of menu items, current carts, and orders. Arrows show dependencies, illustrating how user actions interact with data and different system modules, ensuring smooth workflow and modular structure.

3.9 DESIGN PATTERNS USED (GRASP, GOF)

The Smart Canteen Management system applies both GRASP and GoF design patterns to achieve a well-structured, scalable, and maintainable design. GRASP principles are clearly reflected through the Controller pattern, where JavaScript functions manage user interactions such as login, meal selection, cart updates, payment processing, and order confirmation. The Information Expert pattern is implemented by assigning data-related responsibilities to suitable structures like the menu list, cart, and order collections, which store and process relevant information efficiently. Low Coupling is achieved by separating admin operations, customer activities, and data management into independent modules, reducing interdependencies and simplifying maintenance. At the same time, High Cohesion ensures that each module focuses on a single, well-defined responsibility, improving clarity and reusability. From the GoF perspective, the Singleton pattern is conceptually applied to shared resources such as menu data and order records, ensuring consistency and avoiding duplication across the system. The Facade pattern is evident in the user interface, which offers a unified and simplified access point to complex internal operations like validation, order handling, and payment processing.

CHAPTER 4
IMPLEMENTATION

4.1 MODULE DESCRIPTION

4.1.1 Login and Role Selection Module

The Login and Role Selection Module is the first point of interaction in the Smart Canteen Management System. Its main purpose is to identify whether the user is a customer or an admin and provide access accordingly. Customers are allowed to directly access the ordering interface, enabling them to browse the menu and place orders efficiently. In contrast, admins must enter a secure password to gain access to management functionalities. This module ensures that the system remains secure by controlling access based on user roles, preventing unauthorized users from modifying menus or viewing sensitive order data.

4.1.2 Menu Management Module

This module allows users to explore a wide collection of 20+ flowers available in the shop. Each flower card displays its name, image, and price. Users can add flowers directly to the shopping cart with a single click. The module dynamically loads the flower data from a predefined dataset in JavaScript and ensures a visually organized layout using CSS grid. It acts as the main product display and selection interface.

4.1.3 Customer Ordering Module

The Customer Ordering Module focuses on streamlining the food ordering experience for users. Customers can view the complete menu, select the dishes they want, and add them to a virtual cart. The module also allows customers to choose a preferred payment method and confirm their order in a few simple steps. This functionality simplifies the ordering process, reduces wait times, and improves overall customer satisfaction. By integrating real- time menu updates from the admin module, the system ensures that customers always see accurate availability and pricing, making the entire ordering workflow smooth and efficient.

.4.1.4 Cart and Billing Module

The Cart and Billing Module allows customers to review and manage their selected items before placing an order. It displays a comprehensive list of the chosen food items along with their quantities and calculates the total price. Additionally, it shows the estimated preparation time for the order, helping customers make informed decisions and ensuring transparency. This module improves the user experience by allowing customers to double-check their selections and make adjustments before confirming payment.

4.1.5 Order Management Module (Admin Side)

The Order Management Module (Admin Side) provides the admin with a centralized view of all customer orders. Through this module, the admin can mark orders as “Ready” or delete them once they are completed. This functionality helps the canteen staff efficiently manage and track order fulfillment, ensuring that meals are prepared and delivered on time while maintaining an organized workflow.

4.2 TECHNOLOGY DESCRIPTION

The Smart Canteen Management System is a web-based application that provides an interactive platform for both customers and administrators. The system begins with a login and role selection interface where users choose to continue as either a customer or an admin. Admins are authenticated using a predefined password and are given access to the Admin Panel, which includes options to add new food items, delete existing items, and view all customer orders. The menu management operations dynamically update the Veg and Non-Veg lists stored in memory. Customers, on the other hand, must enter their name and a valid 10- digit phone number before proceeding. They are then presented with meal options that are time- restricted based on the system’s clock. After selecting a Veg or Non-Veg category, customers can browse the menu, choose food items, specify quantities, and add them to a cart. The Cart and Billing system displays all selected items and allows users to choose a payment mode before confirming their order. When an order is placed, the system generates a unique token number, sets a fixed preparation time of 30 minutes, and stores the order details in an internal array that is accessible to the admin. The admin can view all placed orders along with token numbers, customer details, ordered items, and payment mode, ensuring proper order management. Once an order is completed, the admin can remove it from the list. The entire system runs on HTML, CSS, and JavaScript, using dynamic DOM manipulation and in- memory data storage to provide a smooth, interactive user experience without requiring a backend database. Since the entire application runs on the client side, performance remains highly optimized with minimal loading time and no network dependency after the initial page load. The technology stack also improves maintainability and portability, making the application suitable for deployment across desktops, laptops and mobile browsers with consistent behavior and layout.

CHAPTER 5

TESTING

5.1 TESTING STRATEGY

The testing strategy for a Smart Canteen Management System is done to make sure everything works smoothly and correctly. First, each small part of the system like login, ordering, payment, and menu is tested separately. Then all parts are tested together to check if they work correctly as a whole. The entire system is tested to see if students can place orders, the kitchen can receive them, and payments are updated properly. The system is also checked for speed during busy times and tested to make sure user data and payments are safe. Finally, real users try the system to confirm that it is easy to use and works the way they need. This simple testing process ensures the canteen system is reliable and ready to use.

5.2 SAMPLE TEST CASES

Test Case 1: Login

Module

Test Case: Enter valid credentials → System logs in successfully.

Description:

Allows users to enter their username and password to access the system.It ensures only authorized users can log in securely.

Input: Valid username & password. Expected

Output: Login successful.

Status: Pass.

Test Case 2: Registration Module

Test case: Submit valid user details → New account created.

Description:

This module lets new users create an account by filling required details.It stores user information for future login and system use.

Input: New user details.

Expected Output: Account created.

Status: Pass.

Test Case 3: Menu Management Module

Test case: Add new food item → Item appears in menu.

Description:

Admin can add, edit, or delete food items in the canteen menu. It ensures the menu stays updated with correct items and prices.

Input: Add new food item.

Expected Output: Item added to menu.

Status: Pass.

Test Case 4: Order Placement Module

Test case: Select items and place order → Order is recorded.

Description:

Users select their desired food items and place an order easily.

The module confirms the order and sends it to the system for processing.

Input: Selected food items.

Expected Output: Order placed.

Status: Pass.

Test Case 5: Payment Module

Test case: Enter valid payment details → Payment is successful.

Description:

Handles secure online payments through UPI, card, or wallet.

It verifies payment details and processes successful transactions.

Input: Valid UPI/Card details.

Expected Output: Payment successful.

Status: Pass.

Test Case 6: Order Tracking Module

Test case: Enter order ID → Order status is displayed.

Description:
Allows users to check the current status of their order in real time. It updates the order as “Preparing,” “Ready,” or “Delivered.”

Input: Order ID.

Expected Output: Status displayed.

Status: Pass.

Test Case 7: Inventory Management Module

Test case: Update stock quantity → Inventory shows updated value.

Description:

Manages the stock levels of raw materials used in the canteen. Alerts admin when items are low and need restocking.

Input: Update item quantity.

Expected Output: Stock updated.

Status: Pass.

Test Case 8: Report Module

Test case: Choose date and generate → Report is generated.

Description:

Generates daily, weekly, or monthly sales and activity reports. Helps admin analyze performance and make decisions.

Input: Date range.

Expected Output: Report generated.

Status: Pass.

Test Case 9: Notification Module

Test case: Order ready event occurs → User receives notification.

Description:
Sends alerts such as order-ready messages or payment confirmations.Ensures users receive important updates instantly.
Input: Order ready trigger.
Expected Output: Notification sent.
Status: Pass.

5.3 TEST RESULTS

The testing of the Smart Canteen Management System demonstrated that all major modules functioned correctly and efficiently. The Login module allowed users to access the system with valid credentials, while the Registration module successfully created new user accounts. Menu Management operated smoothly, enabling the addition and updating of food items, and the Order Placement module accurately accepted selected items and generated orders. The Payment module processed online transactions successfully with valid details, and Order Tracking displayed the correct status for each order. Inventory Management updated stock levels precisely whenever changes occurred, and the Report Generation module produced sales reports without errors. The Notification module reliably sent alerts to users when their orders were ready. Additionally, the system's interface was intuitive and responsive across devices, proper error messages were displayed for invalid inputs, and it handled multiple simultaneous users without lag. Security measures ensured the safe handling of user credentials and payment details, and the application maintained compatibility across various browsers and operating systems. Data validation prevented incorrect entries, backup and recovery mechanisms functioned properly during simulated failures, and user notifications, including promotions and updates, were sent accurately and timely, reflecting the overall reliability and efficiency of the system.

CHAPTER 6
CONCLUSION & FUTURE ENHANCEMENT

6.1 CONCLUSION

The Smart Canteen System is designed with a clear focus on modularity, efficiency, and user-centric functionality. Through the various UML diagrams, the system's structure and behavior are visualized in a comprehensive and organized manner. The class, package, and component diagrams highlight how responsibilities are distributed across different modules such as Menu, Cart, Order, and Payment, ensuring that each part of the application remains independent yet well-connected. The use case, sequence, and activity diagrams illustrate the smooth interaction flow between customers and the system, from browsing the menu to placing orders and making secure payments. The deployment diagram further demonstrates how the application functions across devices and servers, showcasing real-world implementation. Overall, these diagrams collectively reflect a robust, scalable, and maintainable system architecture that supports a seamless digital canteen experience for users.

6.2 FUTURE ENHANCEMENT

The Smart Canteen Management System can be enhanced to include mobile apps for Android and iOS, enabling real-time ordering and notifications. Integration with multiple payment gateways like UPI, cards, and digital wallets will simplify transactions. AI-based meal recommendations can suggest dishes based on customer preferences and dietary needs. Dynamic menu management can update available items automatically according to ingredient stock. Inventory management will help monitor supplies, reduce wastage, and alert staff when items run low. Analytics and reporting features can track popular meals, peak hours, and revenue trends. Multi-canteen support allows centralized management of multiple locations. Automated notifications can inform customers when orders are ready. IoT integration with smart counters and kitchen equipment can improve efficiency. Loyalty and reward systems can encourage repeat orders. Feedback mechanisms will help enhance food quality and service. Cloud hosting enables multi-platform access and real-time data synchronization. Overall, these enhancements make the system more intelligent, efficient, and scalable.

APPENDIX A- SOURCE CODE

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Smart Canteen Management</title>
<style> body {
background-image: url('https://images.unsplash.com/photo-1504674900247- 0877df9cc836');
background-size: cover;
 background-position: center; font-family: Arial, sans-serif;
margin: 0;
padding: 0;
color: white;
}
.overlay {
background: rgba(0,0,0,0.5);
width: 100%;
height: 100%;
position: absolute; top: 0;
left: 0;
z-index:1;
}
.container {
position: relative;
z-index:2;
padding: 30px;
text-align: center;
}
.box {
background: rgba(255,255,255,0.15);
padding: 20px;
border-radius: 10px;
margin: 20px auto;
width: 80%;
}
button {
```

```
padding: 18px 32px;
margin: 12px;
border: none;
 background: #ffcc00;
border-radius: 12px;
cursor: pointer;
font-size: 20px;
font-weight: bold;
}
input, select { padding:10px;
font-size:18px;
margin:5px;
}
ul{ text-align:left; }
</style>
</head>
<body>
<div class="overlay"></div>
<div class="container">
<h1>Fresh Bite Smart Canteen</h1>
<h3>Opening Time: 7:00 AM | Closing Time: 10:00 PM</h3>
<div class="box" id="loginBox">
<h2>Login As</h2>
<button onclick="showAdminLogin()">Admin</button>
<button onclick="showCustomerSection()">Customer</button>
</div>

<div id="adminLogin" class="box" style="display:none;">
<h2>Admin Login</h2>
<input type="password" id="adminPass" placeholder="Enter Secret Password" />
<button onclick="checkAdmin()">Submit</button>
<p id="adminError" style="color:red;"></p>
</div>

<div id="adminPanel" class="box" style="display:none;">
<h2>Admin Panel</h2>
<button onclick="showAddItem()">Add Items</button>
<button onclick="showDeleteItem()">Delete Items</button>
```

```
<button onclick="showOrders()">View Orders</button>
<button onclick="returnHome()">Finish & Return Home</button>
<div id="adminContent" style="margin-top:20px;"></div>
</div>
<div id="customerSection" class="box" style="display:none;">
<h2>Enter your Details</h2>
<input type="text" id="custName" placeholder="Your Name" /><br>
<input type="text" id="custPhone" placeholder="Phone Number" /><br>
<p id="phoneError" style="color:red;"></p>
<button onclick="validateCustomer()">Continue</button>
</div>

<div id="mealOptions" class="box" style="display:none;">
<h2>Select Meal</h2>
<button onclick="checkMealTime('breakfast')">Breakfast</button>
<button onclick="checkMealTime('lunch')">Lunch</button>
<button onclick="checkMealTime('dinner')">Dinner</button>
<p id="mealMsg" style="color:yellow;"></p>
</div>

<div id="vegNonVeg" class="box" style="display:none;">
<h2>Select Category</h2>
<button onclick="showMenu('veg')">Veg</button>
<button onclick="showMenu('nonveg')">Non-Veg</button>
</div>

<div id="menuSection" class="box" style="display:none;"></div>
<div id="nextPage" class="box" style="display:none;"></div>

<script>
let tokenCounter=1001; let orders=[];
 let currentCart=[]; let fullMenu={
veg:["Idly","Dosa","Poori","Veg Biryani","Meals"], nonveg:["Chicken
Biryani","Mutton Biryani","Fish Fry","Egg Curry"]
};
function showAdminLogin(){
document.getElementById('loginBox').style.display='none';
document.getElementById('adminLogin').style.display='block';
```

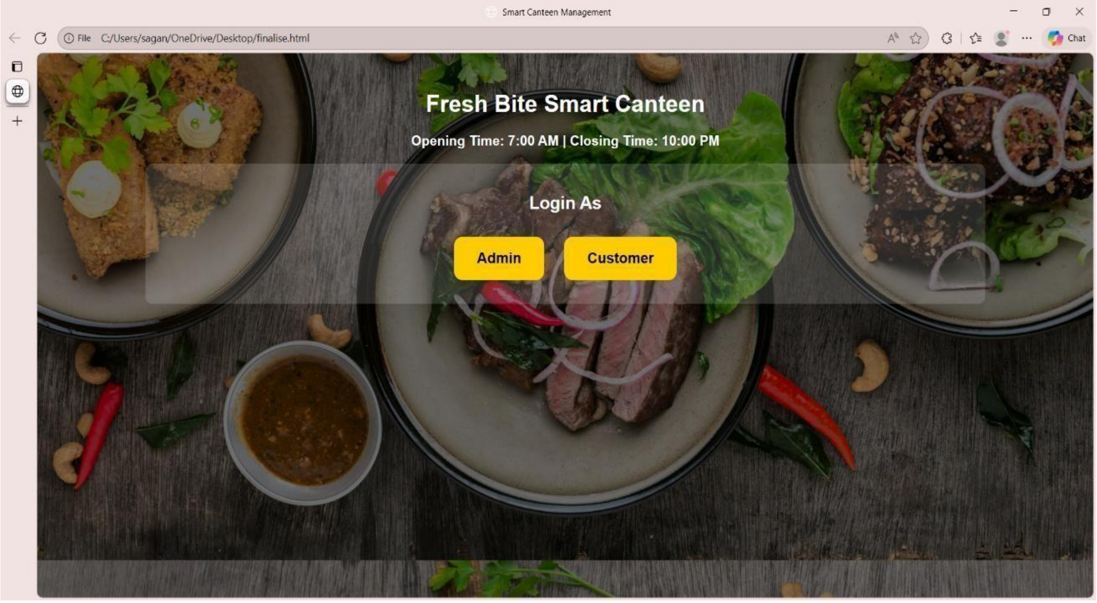
```
}
function checkAdmin(){
let p=document.getElementById('adminPass').value; if(p==='1234'){
document.getElementById('adminLogin').style.display='none';
document.getElementById('adminPanel').style.display='block';
document.getElementById('adminError').innerHTML='';
} else {
document.getElementById('adminError').innerHTML='Wrong Password';
}
}
function showAddItem(){
let content=document.getElementById('adminContent'); content.innerHTML=`
<h3>Add Item</h3>
<input type='text' id='newItem' placeholder='Item Name' />
<select id='itemType'><option value='veg'>Veg</option><option value='nonveg'>Non-Veg</option></select>
<button onclick='addItem()'>Add</button>
<p id='addMsg' style='color:lightgreen;'></p>`;
}
function addItem(){
let name=document.getElementById('newItem').value;
let type=document.getElementById('itemType').value;
if(name){ fullMenu[type].push(name);
document.getElementById('addMsg').innerText='Item Added!'; }
}
function showDeleteItem(){
let content=document.getElementById('adminContent');
let allItems=fullMenu.veg.concat(fullMenu.nonveg);
let options=allItems.map(i=><option value='${i}'>${i}</option>).join('');
content.innerHTML=`
<h3>Delete Item</h3>
<select id='delItem'>${options}</select>
<button onclick='deleteItem()'>Delete</button>
<p id='delMsg' style='color:lightcoral;'></p>`;
}
function deleteItem(){
let name=document.getElementById('delItem').value;
['veg','nonveg'].forEach(type=>{fullMenu[type]=fullMenu[type].filter(i=>i!==name);
});
document.getElementById('delMsg').innerText='Item Deleted!';
}
function showOrders(){
let content=document.getElementById('adminContent');
if(orders.length===0){ content.innerHTML='<h3>No Orders Yet</h3>';
return;}
let list='<h3>Orders</h3><ul>' + orders.map(o=><li>Token: ${o.token}, ${o.name} -
${o.phone}, Items: ${o.items.map(i=>i.name+' x'+i.qty).join(', ')}, Pay: ${o.pay},
PrepTime: ${o.prep} mins</li>).join('')+'</ul>';
content.innerHTML=list;
}
function returnHome(){ document.getElementById('adminPanel').style.display='none';
document.getElementById('adminContent').innerHTML='';
document.getElementById('loginBox').style.display='block';


}
function showCustomerSection(){
document.getElementById('loginBox').style.display='none';
document.getElementById('customerSection').style.display='block';
 currentCart=[];
}
function validateCustomer(){
let phone=document.getElementById('custPhone').value;
let phoneError=document.getElementById('phoneError');
if(phone.length!==10 || !/^[0-9]+$/.test(phone)){
phoneError.innerHTML='Please enter a valid 10-digit phone number';
return;
}
PhoneError.innerHTML=’’;
document.getElementById('customerSection').style.display='none';
document.getElementById('mealOptions').style.display='block';
}
function checkMealTime(meal){ let hr=new Date().getHours();
let msg=document.getElementById('mealMsg');
if(meal==='breakfast' && !(hr>=7 && hr<=11)){ msg.innerHTML='Breakfast only
from 7 AM to 11 AM'; return; }
if(meal==='lunch' && !(hr>=12 && hr<=15)){ msg.innerHTML='Lunch only from 12
```

```
PM to 3 PM'; return; }
if(meal==='dinner' && !(hr>=18 && hr<=22)){ msg.innerHTML='Dinner only from 6
PM to 10 PM'; return; }
msg.innerHTML=";
document.getElementById('mealOptions').style.display='none';
document.getElementById('vegNonVeg').style.display='block';
}
function showMenu(type){
let menu=document.getElementById('menuSection');
menu.style.display='block';
let list=fullMenu[type].map(i=><option value='${i}'>${i}</option>).join(");
menu.innerHTML=`
<h3>${type==='veg'?'Veg':'Non-Veg'} Menu</h3>
<select id='foodItem'>${list}</select><br><br>
<input type='number' id='qty' placeholder='Quantity' min='1' /><br><br>
<button onclick='addToCart()'>Add to Cart</button>
<button onclick='proceedToPayment()'>Proceed to Payment</button>`;
}
function addToCart(){
let item=document.getElementById('foodItem').value;
let qty=document.getElementById('qty').value;
currentCart.push({name:item, qty:qty}); alert('Item added to cart!');
}
function proceedToPayment(){
document.getElementById('menuSection').style.display='none';
let next=document.getElementById('nextPage');
next.style.display='block';
let cartHtml=currentCart.map(i=><p>${i.name} x ${i.qty}</p>).join(");
next.innerHTML=`
<h2>Payment & Cart</h2>
${cartHtml}
<h3>Select Payment Mode</h3>
<select id='payMode'>
<option value='GPAY'>GPAY</option>
<option value='Cash'>Cash</option>
</select><br><br>
<button onclick='confirmOrder()'>Confirm Order</button>`;
}
```

```
function confirmOrder(){
let pay=document.getElementById('payMode').value;
let name=document.getElementById('custName').value;
let phone=document.getElementById('custPhone').value;
let token=tokenCounter++;
let prepTime=30; // Max prep time 30 mins
orders.push({token,name,phone,items:currentCart,pay,prep:prepTime});
alert(Order Confirmed!\nToken Number: ${token}\nItems:
${currentCart.map(i=>i.name+' x'+i.qty).join(', ')}\nPayment: ${pay}\nPreparation
Time: ${prepTime} mins); document.getElementById('nextPage').style.display='none';
document.getElementById('loginBox').style.display='block';
}
</script>
</body>
</html>
```

APPENDIX B - SCREENSHOTS

1. USER LOGIN



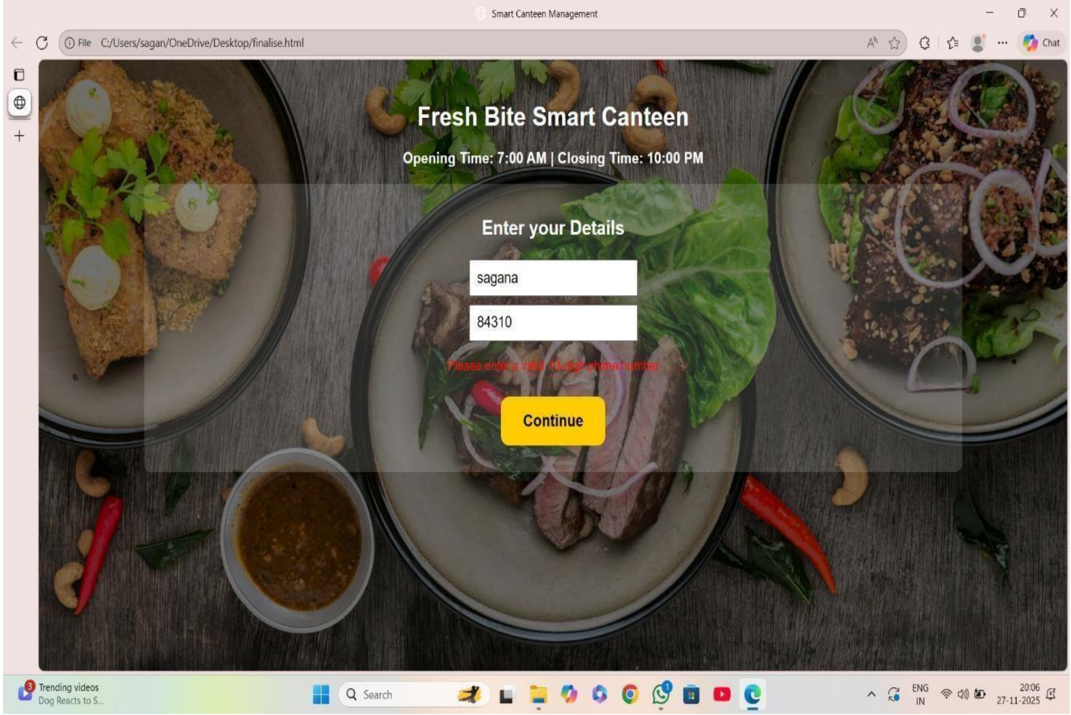
Fresh Bite Smart Canteen

Opening Time: 7:00 AM | Closing Time: 10:00 PM

Login As

Figure B 1. USER LOGIN

2. USER DETAILS



Fresh Bite Smart Canteen

Opening Time: 7:00 AM | Closing Time: 10:00 PM

Enter your Details

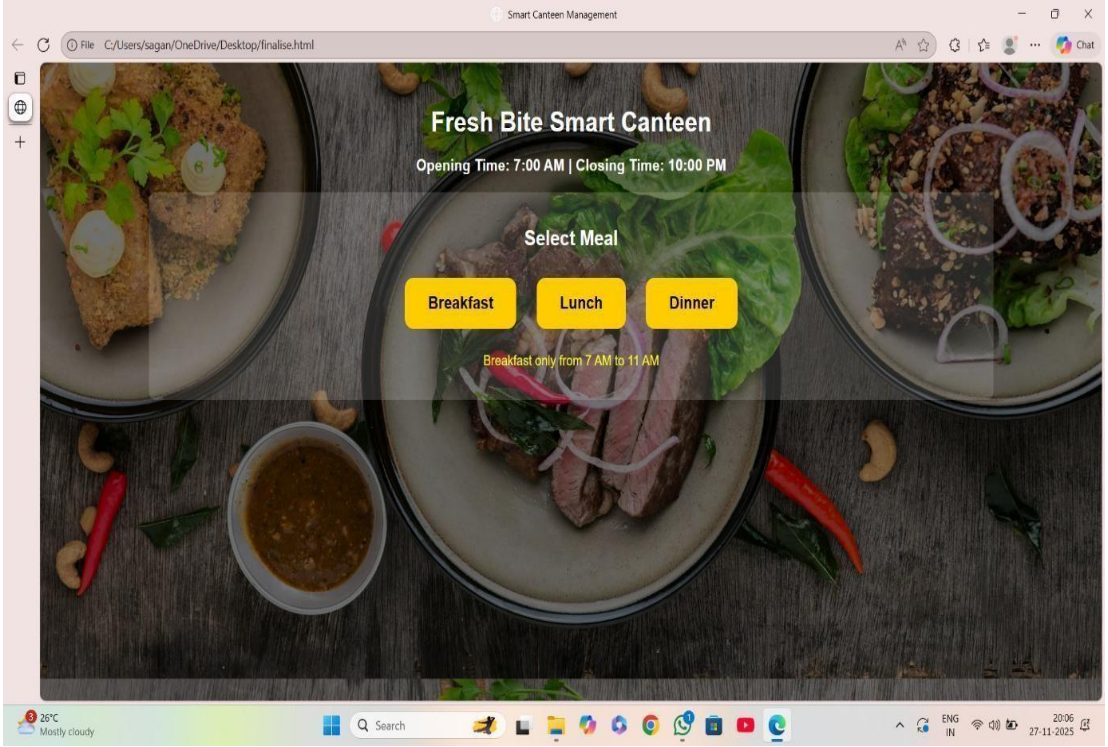
sagana

84310

Please enter a valid 10-digit phone number

Figure B 2. USER DETAIL

3. SELECT MEAL



Fresh Bite Smart Canteen

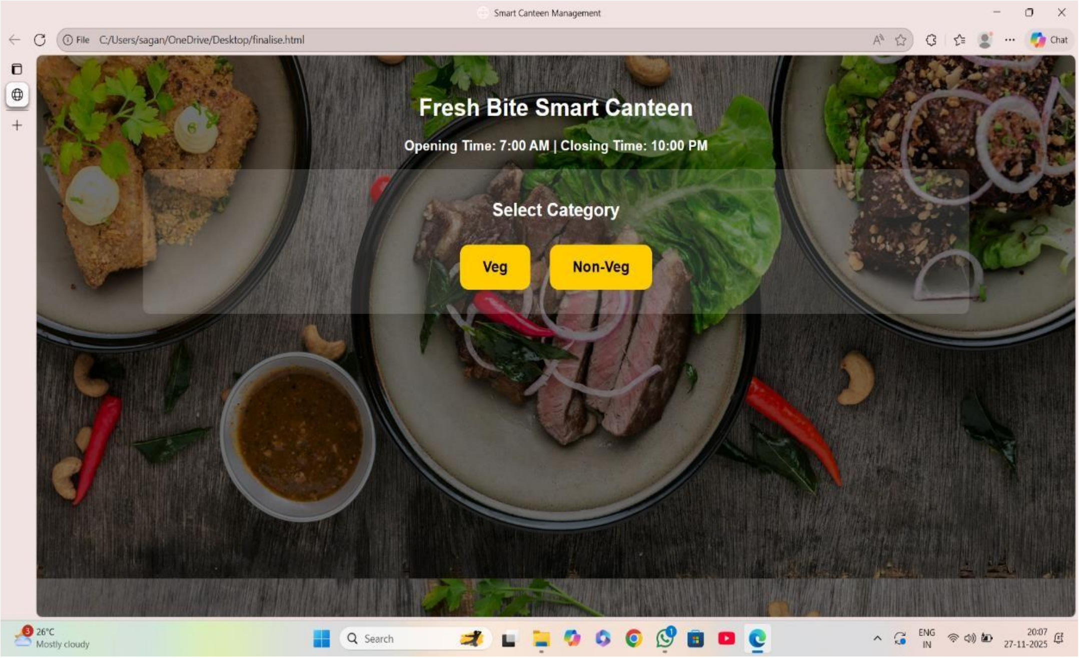
Opening Time: 7:00 AM | Closing Time: 10:00 PM

Select Meal

Breakfast only from 7 AM to 11 AM

Figure B 3. SELECT MEAL

4. SELECT CATEGORY



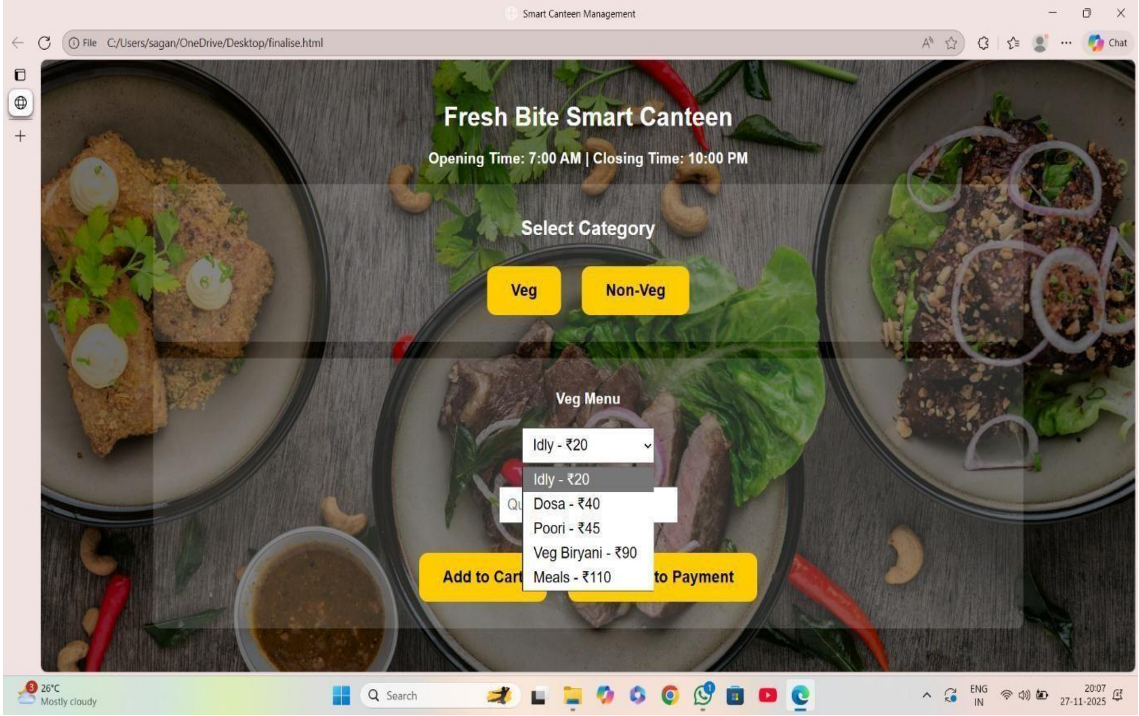
Fresh Bite Smart Canteen

Opening Time: 7:00 AM | Closing Time: 10:00 PM

Select Category

Figure B 4. SELECT CATEGORY

5. SELECT DISHES



Fresh Bite Smart Canteen

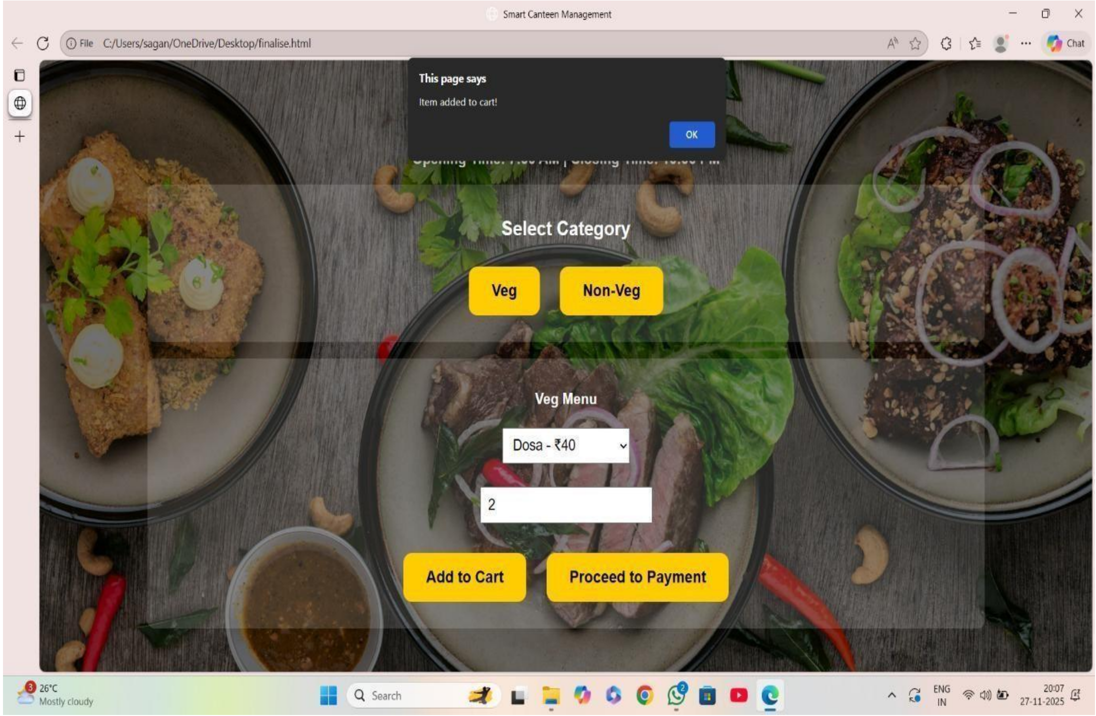
Opening Time: 7:00 AM | Closing Time: 10:00 PM

Select Category

Veg Menu

Figure B 5. SELECT DISHES

6. ADD TO CART



This page says

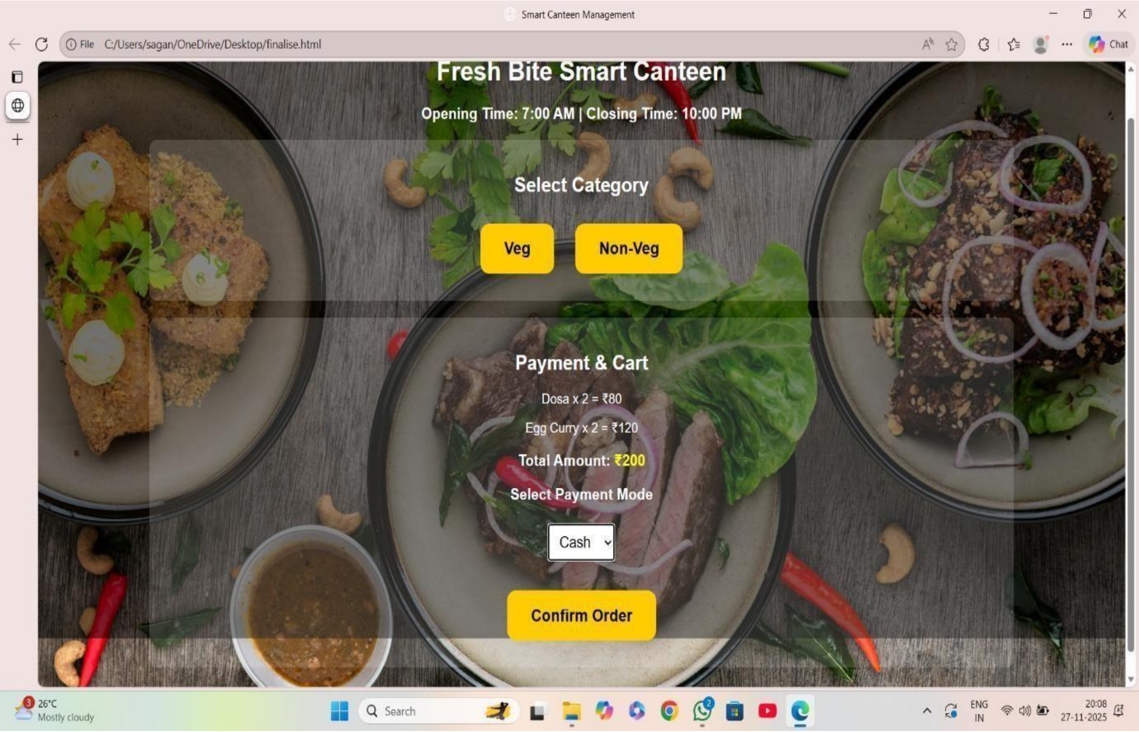
Item added to cart!

Select Category

Veg Menu

Figure B 6. ADD TO CART

7. PAYMENT



Fresh Bite Smart Canteen

Opening Time: 7:00 AM | Closing Time: 10:00 PM

Select Category

Payment & Cart

Dosa x 2 = ₹80

Egg Curry x 2 = ₹120

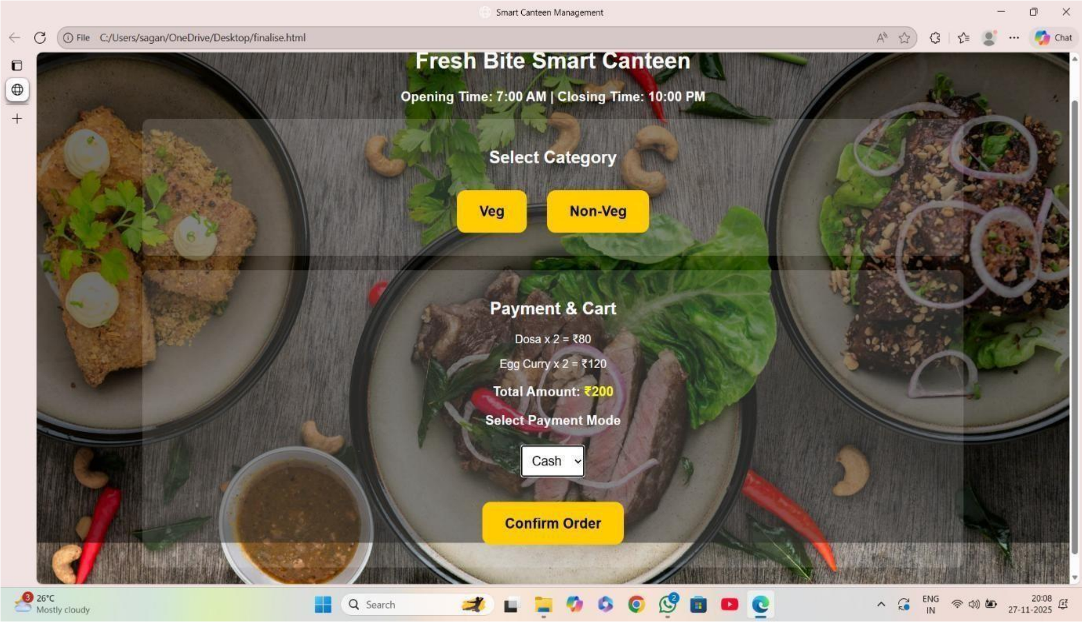
Total Amount: ₹200

Select Payment Mode

Cash

Figure B 7. PAYMENT

8. CONFIRM ORDER



Fresh Bite Smart Canteen

Opening Time: 7:00 AM | Closing Time: 10:00 PM

Select Category

Payment & Cart

Dosa x 2 = ₹80

Egg Curry x 2 = ₹120

Total Amount: ₹200

Select Payment Mode

Cash

Figure B 8. CONFIRM ORDER

9. ADMIN LOGIN

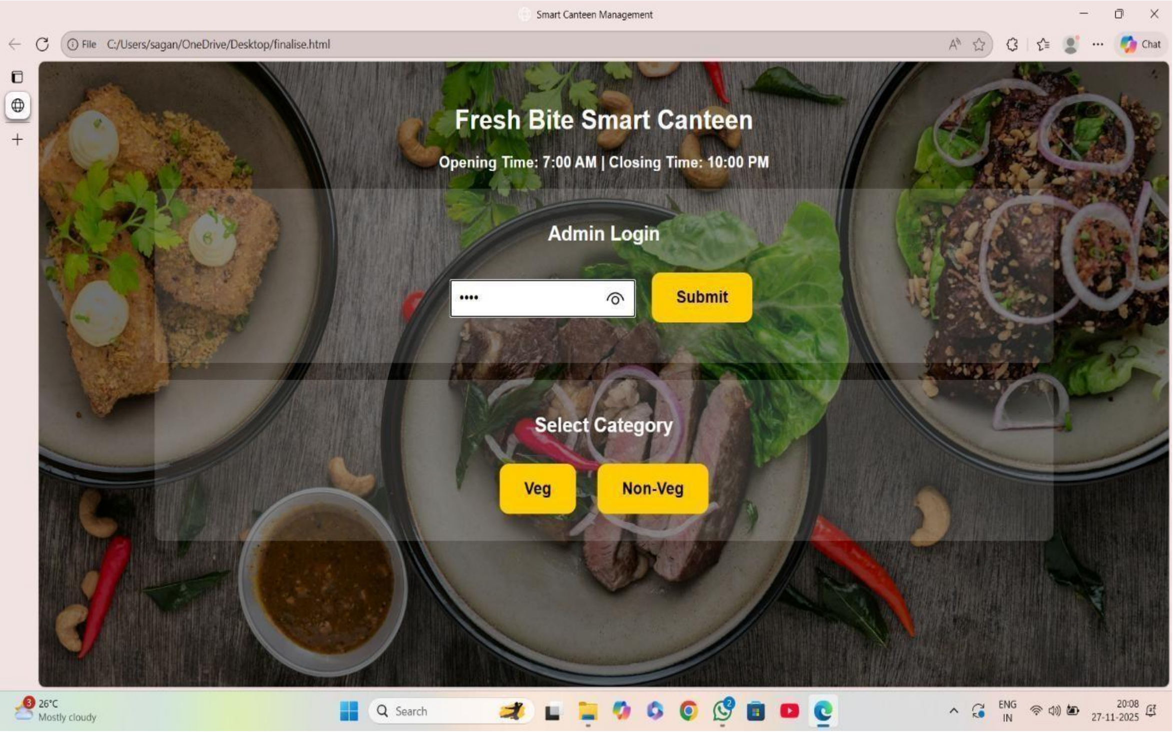
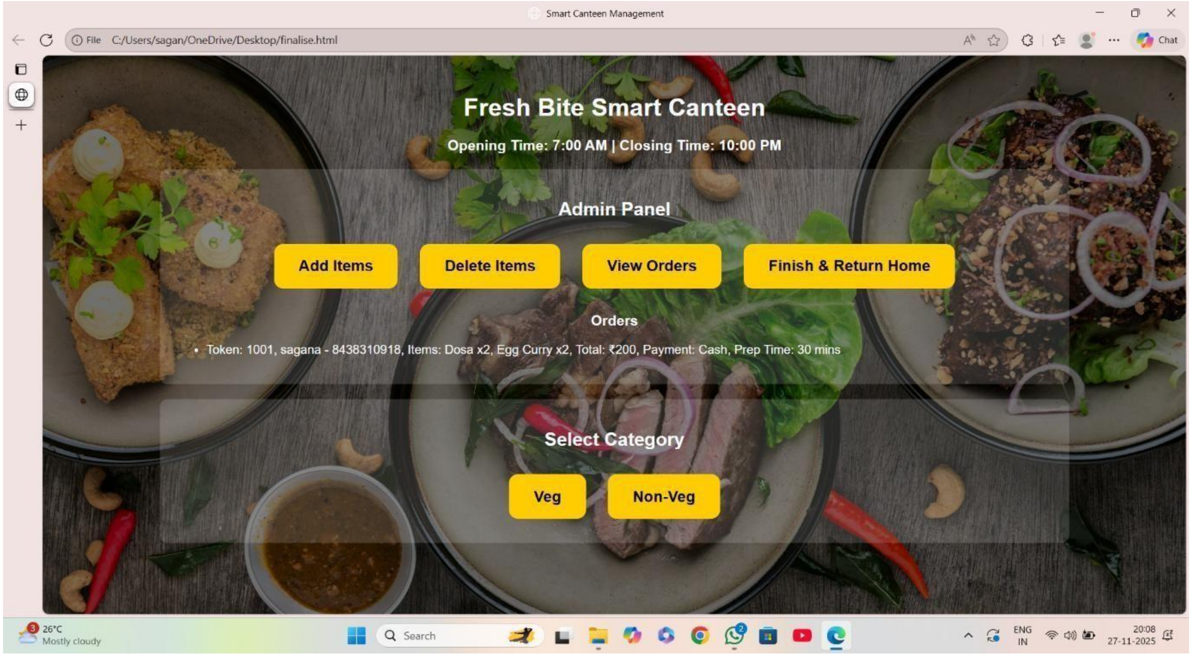


Figure B 9. ADMIN LOGIN

10. ORDER DETAILS



- Token: 1001, sagana - 8438310918, Items: Dosa x2, Egg Curry x2, Total: ₹200, Payment: Cash, Prep Time: 30 mins

Figure B10.ORDER DETAILS

REFERENCES

1. Pressman, R. S. (2020). Software Engineering: A Practitioner's Approach. McGraw-Hill Education.
2. Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide. Addison-Wesley.
3. Sommerville, I. (2016). Software Engineering (10th Edition). Pearson.
4. Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Prentice Hall.
5. Martin, R. C. (2003). Agile Software Development: Principles, Patterns, and Practices. Prentice Hall.
6. R. Sharma, S. Gupta, "Design and Implementation of Smart Canteen System Using Object-Oriented Approach," International Journal of Computer Applications, vol. 180, no. 45, 2018.
7. S. Kaur, M. Singh, "Object-Oriented Analysis and Design of Automated Canteen Management System," International Journal of Engineering and Technology, vol. 7, no. 3, 2018.
8. A. Verma, R. Tiwari, "Smart Canteen Management System Using UML and OOAD," International Journal of Advanced Research in Computer Science, vol. 9, no. 2, 2018.