

1. INTRODUCTION TO PROJECT

The web-based “Canteen Management System” project is an attempt to simulate the basic concepts of an online canteen ordering platform. The system enables different types of users—Admin, Caterer, and Customer—to perform their respective roles such as adding caterers, adding food items, browsing menu items, placing orders, making payments, and collecting orders using a generated token.

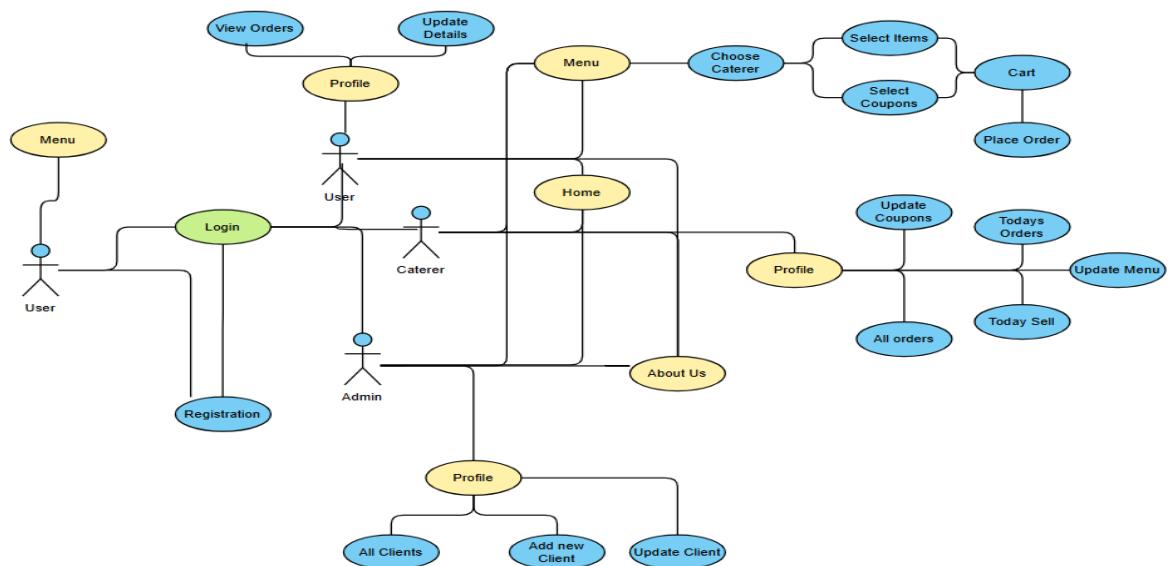
The system provides a Quick View facility that displays details about available food items without login. However, if a customer wants to place an order, it is mandatory to log in to their account.

The system allows the customer to browse all food items offered by the caterers, view details such as item name, category, price, and availability, and then select the desired items. Here we have provided a quick browsing facility which displays the list of available food items and allows the customer to choose a particular item. The system then checks the availability of the item in the inventory. If the item is available, the system allows the customer to proceed with the order; otherwise, it prompts the user to choose another item.

To place an order, the system asks the customer to enter details such as name, contact number, delivery location (if applicable), and payment information. After verifying the payment, the system confirms the order, generates a unique token number for the customer, and updates the inventory database as well as the order database.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS



2.1 User Account

The Canteen Management System supports three types of users: **Admin**, **Caterer**, and **Customer**. Each type of user is presented with a different set of functionalities upon logging into the system.

Admin

- The Admin is the highest-level user with complete control over the system.
- On logging in, the Admin can:

1. Add new caterers into the system, providing their business details, contact information, and login credentials.
2. Update existing caterer profiles.
3. View all registered clients, including caterers and customers.
4. Remove a caterer if they no longer operate within the canteen.
5. Monitor all system activities and view reports on total sales and active orders.

Caterer

- Caterers are responsible for maintaining their own menus and processing orders.
- After logging in, a Caterer can:
 1. Update their profile details, such as business name, contact information, and operating hours.
 2. Add new food items to their menu, specifying name, category, price, quantity available, preparation time, and item image.
 3. Update existing menu items (price, availability, etc.).
 4. View today's orders, including token numbers, ordered items, and customer details.
 5. Update order status as "Prepared," "Ready for Pickup," or "Delivered."

6. Apply or update special coupons or discounts.
7. View total sales for a specific date or time range.

Customer

- Customers are end-users who place orders and make payments through the system.
- A customer may be a:
 1. **Guest User** – Can browse available caterers and menu items through the **Quick View** feature without logging in, but cannot place an order.
 2. **Registered Customer** – Can log in, place orders, make payments, and view order history.
- After logging in, a Registered Customer can:
 1. Update personal details such as name, contact number, and address.
 2. Choose a caterer from the available list.
 3. Browse the caterer's menu and select items to add to the cart.
 4. Apply available coupons for discounts.
 5. Place an order by confirming the cart.
 6. Make secure online payments via card or UPI.

7. Receive a **unique token number** upon successful payment.
8. Show the token to the caterer to collect the order.
9. View order history, including past token numbers, purchased items, and payment amounts.

2.2 Registration and creation of user profile

Before placing any order, a customer or caterer must register on the platform.

- **Customer Registration** requires:

- Name
- Username
- Password
- Contact number
- Email address
- Optional delivery address

- **Caterer Registration** requires:

- Business name
- Username
- Password
- Business address
- Contact number
- Email address
- Food service category

The system stores this information in the respective user database. Each profile can be updated later by the user.

2.3 Quick Search

Here we have provided a **Quick View** facility for any user to browse available food items without logging into an account. This feature allows a user to search for menu items from different caterers and compare their prices, categories, and availability.

After logging in, a customer (either a registered user or a newly registered user) will be asked to select a **caterer** from the available list. The caterer name will be chosen from a dropdown menu displaying all registered caterers in the system.

Once a caterer is selected, the system will access the **Menu Database** and retrieve the list of food items provided by that caterer. The system shall then ask the customer to select:

- **Food Items** – Chosen from a list with name, category, price, and availability.
- **Quantity** – The desired number of portions for each selected item.
- **Coupons** – Optional discount codes provided by the caterer, selected from a dropdown list if available.

After the customer has chosen the items and quantity, the system will check for any invalid entries, such as requesting more quantity than available in stock. In such cases, the system will prompt the customer to adjust the order.

The system then queries the **Inventory Database** to ensure that the selected items are in stock. It displays the results in a suitable tabular form, with the following information for each selected item:

- Item ID
- Item Name
- Category
- Unit Price
- Quantity Selected

- Total Price for that item

The customer can then add these items to the **Cart** for final review. If the customer wishes to proceed, they can place the order by confirming the cart. The system will then ask for the following details:

- Customer Name
- Contact Number
- Payment Method (e.g., UPI, debit/credit card)

Upon successful payment, the system will:

- Generate a **unique token number** for order pickup.
- Update the inventory to reflect reduced stock.
- Save the order details in the **Orders Database**.

The token will then be shown to the customer, who will present it to the caterer at the time of food collection.

2.4 Order Placement / Payment / Confirmation

After taking the customer through the step 2.2, **Quick View and Menu Selection**, the system will now ask the customer if they wish to proceed with placing the order.

If the customer has been browsing as a **guest**, they will first have to **register** and become a registered user, and then log into the system.

If the customer is already a registered user and has logged in, they can proceed directly to place the order. However, if they have been acting as a guest, they will still need to log in before continuing.

Once the system has ensured that the user is logged in validly, it verifies the selected menu items against the inventory to confirm availability.

- If any selected item is **out of stock**, the system will notify the customer and prompt them to either remove or replace that item.
- If all items are available, the customer can proceed to checkout.

At checkout, the system will display the following order details for confirmation:

- Caterer Name
- Selected Items (with quantity and price)
- Subtotal Price
- Discounts or Coupons Applied
- Final Total Amount to be Paid

The customer will then be asked to select a **payment method** from the available options (UPI, debit card, credit card).

After the payment method is chosen, the system:

1. Validates the payment details (e.g., card number, UPI ID).
2. Processes the payment through the integrated payment gateway.

3. On successful payment, immediately generates a **unique token number** for the order.

This token acts as the customer's proof of purchase and must be shown to the caterer while collecting the food.

Once payment is completed successfully, the system performs the following updates:

- **Orders Database** – Inserts a new order record containing the customer ID, caterer ID, order details, total amount, payment status, and token number.
- **Inventory Database** – Reduces the available quantity for each ordered item by the purchased amount.
- **Sales Records** – Updates the daily sales report for the respective caterer.

Finally, the system displays an **Order Confirmation Screen** to the customer, which includes:

- Order ID
- Token Number
- Ordered Items and Quantity
- Total Paid Amount

2.5 View Order History

The system shall allow a customer to view all information about their previous orders. After logging in, the customer can access the **Order History** section, where they may search using their **token number** or **order ID**. The system then accesses the **Orders Database** and retrieves the details of the order, presenting them in a **tabular format**.

The displayed details will include:

- Token Number
 - Caterer Name
 - Item Names and Quantities
 - Total Price Paid
 - Order Date and Time
 - Order Status (e.g., Completed, Ready for Pickup, Cancelled)
-

Admin Functionalities

The **Admin** should be able to:

- Log in to the **Admin Dashboard**.
- Add caterer information (business details, contact info, login credentials).
- Add and update caterer menu information if needed.
- Delete a caterer from the system.
- View detailed information about any registered customer or caterer using their **User ID**.
- Monitor and view all orders across all caterers in the system.

2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface

Go to Appendix B for user interface designs and screenshots of the Canteen Management System.

2.2.2 Performance

- **Number of Concurrent Users:**

The CMS shall be able to handle at least 500 concurrent transactions/inquiries per second, ensuring smooth performance during peak ordering hours such as lunch and dinner time.

- **Order Placement and Payment:**

The system is designed to be resilient to temporary server failures. As it uses the robust features of Spring Boot, React JS, and MySQL, the transaction process is designed to resume from the last successful step even if the server is briefly disconnected during order placement or payment processing.

2.2.3 Constraint

The CMS shall be able to handle at least 500 transactions/inquiries per second without performance degradation. The system should be optimized to manage simultaneous requests from multiple customers, caterers, and the admin.

2.2.4 Other Requirements:

- **Hardware Interfaces**

The CMS is expected to function efficiently on:

- Intel Core i3 Processor (or equivalent) or above
- 4 GB RAM (minimum)
- 50 GB HDD/SSD storage
- Network adapter supporting high-speed internet connectivity

- **Software Interfaces**

The CMS shall operate on Windows, Linux, or macOS operating systems.

Backend: Spring Boot framework running on Apache Tomcat server.

Frontend: React JS application running in any modern browser (Google Chrome, Mozilla Firefox, Microsoft Edge).

Database: MySQL relational database.

Compatible Browsers: Chrome 80+, Firefox 70+, Safari 13+.

Optional integration with UPI / Payment Gateway APIs for online transactions.

3. DESIGN

3.1 Database Design

The following table structures depict the database design.

Table1: Users

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	role_id	Number	4	0
0	name	Varchar2	255	0
0	email	Varchar2	255	0
0	password	Varchar2	255	0

0	mobile	Varchar2	15	1
0	created_at	Date		0
0	updated_at	Date		1
0	created_by	Number	4	1
0	updated_by	Number	4	1

Table2: Roles

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	type	Varchar2	100	0

Table3: Caterer

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	name	Varchar2	255	0
0	user_id	Number	4	0
0	created_at	Date		0
0	updated_at	Date		1
0	created_by	Number	4	1
0	updated_by	Number	4	1

Table4: Item

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	caterer_id	Number	4	0
0	name	Varchar2	255	0
0	price	Float		0
0	created_by	Number	4	1

Table5: Orders

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	user_id	Number	4	0
0	created_at	Date		0
0	total_amount	Float		0
0	razorpay_payment_id	Varchar2	255	1
0	razorpay_order_id	Varchar2	255	1
0	razorpay_signature	Varchar2	255	1
0	payment_method	Varchar2	50	0
0	created_by	Number	4	1

Table6:Order_details

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	item_id	Number	4	0
0	order_id	Number	4	0
0	quantity	Number	4	0
0	price_per_unit	Float		0

Table7:_coupon_types

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	caterer_id	Number	4	0
0	type	Varchar2	100	0
0	min_count	Number	4	0
0	original_price	Float		0
0	discount_per_coupon	Float		0

Table8:Coupon

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	coupon_type_id	Number	4	0
0	user_id	Number	4	0
0	count	Number	4	0
0	validity	Date		0
0	created_at	Date		0
0	updated_at	Date		1
0	created_by	Number	4	1
0	updated_by	Number	4	1

Table9:Coupon_usage

Key Type/Constraint	Column Name	Data Type	Length	Allow Null (1=Yes;0=No)
3	id	Number	4	0
0	coupon_id	Number	4	0
0	created_at	Date		0
0	updated_at	Date		1
0	created_by	Number	4	1

E-R Diagram,Dataflow diagram and Class Diagram:

Go to Appendix A

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in the Canteen Management System project. Pascal casing is used mainly (i.e., capitalize the first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	UserProfile, CanteenMenu, OrderService	Class names should be based on 'objects' or 'real things' and should generally be nouns. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	getOrderDetails, updateMenuItem	Methods should use verbs or verb phrases.
Parameter	Camel	userId, catererId, paymentMode	Use descriptive parameter names so that the parameter name and its type clearly indicate its meaning.
Interface	Pascal with "I" prefix	IPaymentGateway, IUserService	Do not use the '_' sign.
Property	Pascal	TotalAmount, MenuItem, OrderStatus	Use a noun or noun phrase to name properties.
Associated Private Member Variable	_camelCase	_orderId, _menuList	Use underscore camel casing for private member variables.

Exception Class	Pascal with "Exception" suffix	PaymentFailedException, ItemNotAvailableException	Exception classes should end with the word "Exception" and represent error conditions.
-----------------	--------------------------------	---	--

Comments

- Comment each class, interface, method, and non-public type member to describe its purpose and usage within the Canteen Management System.
- Add comments for each region declaration to explain the logical grouping of code.
- Use end-line comments only on variable declaration lines.
- End-line comments are comments that follow code on the same line.
- Separate comments from comment delimiters (// in Java/Spring Boot or /* ... */ for multi-line comments) with one space.
- Begin each comment text with an uppercase letter.
- End each comment with a period.
- Write comments to explain the intent and logic of the code, not to restate what the code already does.

5. TEST REPORT

GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
4				Nothing

Canteen Management System

5				Nothing
6				Nothing
7				Nothing
8				Nothing
9				Nothing
10				Nothing
11				Nothing
12				Nothing
	STATIC TESTING			
SR-NO	Deviation	Program		
1	Commenting not followed	All Web Application		

6. PROJECT MANAGEMENT RELATED STATISTICS

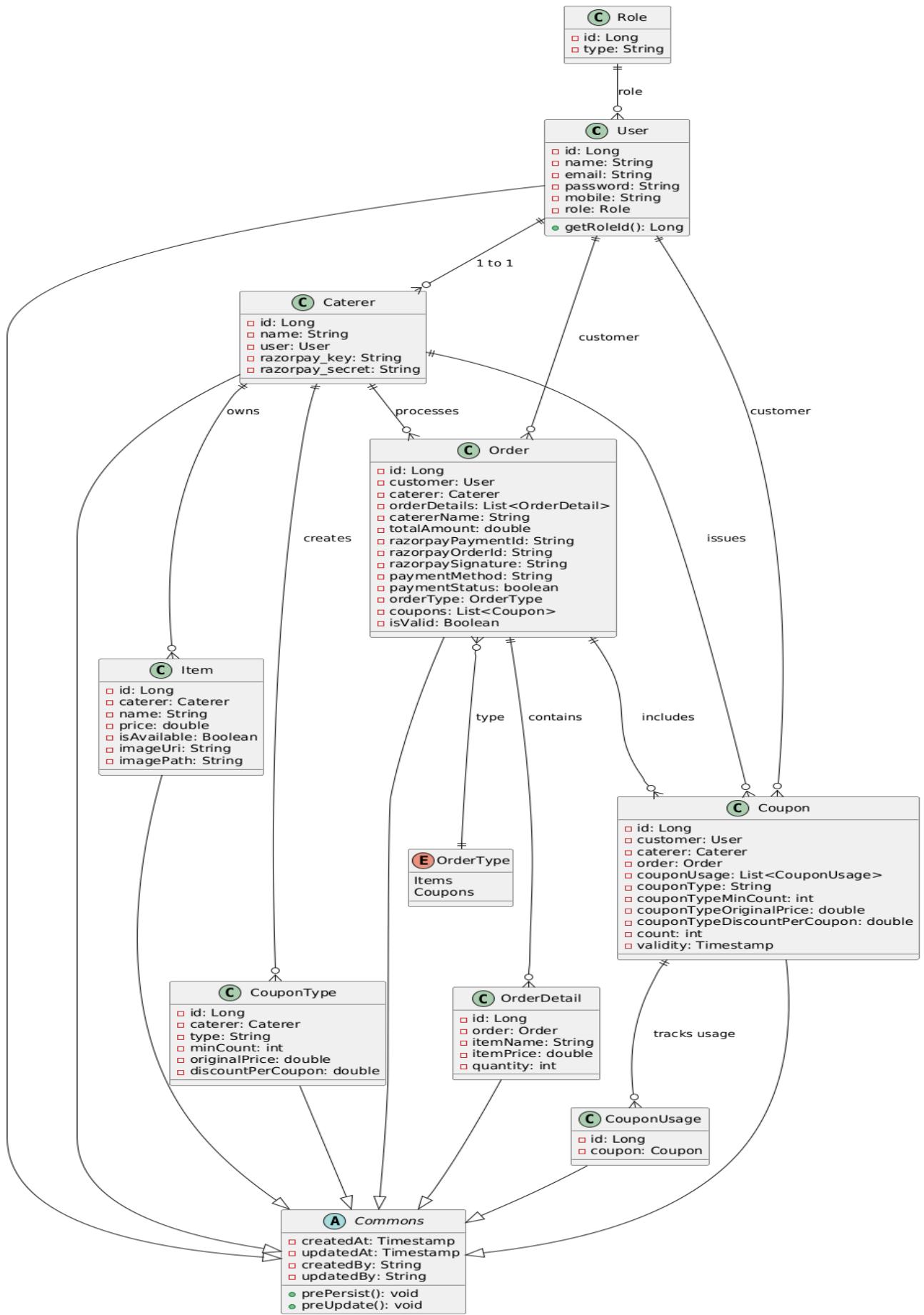
DATE	WORK PERFORMED	SLC Phase	Additional Notes
July 19,2025	Project Allotment and User Requirements Gathering	Feasibility Study	Our team met the client Mr. Nitinkudale (CEO, SIIT Pune) to know his requirements.
July 20,2025	Initial SRS Document Validation And Team Structure Decided	Requirement Analysis (Elicitation)	The initial SRS was presented to the client to understand his requirements better
July 21,2025	Designing the use-cases, Class Diagram, Collaboration Diagram, E-R Diagram and User Interfaces	Requirement Analysis & Design Phase	Database Design completed
July 22,2025	Business Logic Component design Started	Design Phase	-----
July 23,2025	Coding Phase Started	Coding Phase	70% of Class Library implemented.
July 24,2025	Implementation of Web Application and Window Application Started	Coding Phase	Class Library Development going on.
Aug 03, 2025	Off	Off	Off
Aug 04, 2025	Implementation of Web Application and Server Side Testing	Coding Phase and Unit Testing	--
Aug 05, 2025	After Ensuring Proper Functioning the Required Validations were Implemented	Coding Phase and Unit Testing	Module Integration was done by the Project leader

Aug 06, 2025	The Project was Tested by the Team Leader.	Testing Phase (Module Testing)	--
Aug 07, 2025	The Project was Submitted to remote Repository For Testing	Testing Phase (Acceptance Testing)	One of The team members Taken It for Testing
Aug 09-10, 2025	The Errors Found were Removed	Debugging	The Project was complete for submission
Aug 11, 2025	Final Submission of Project		

Appendix A

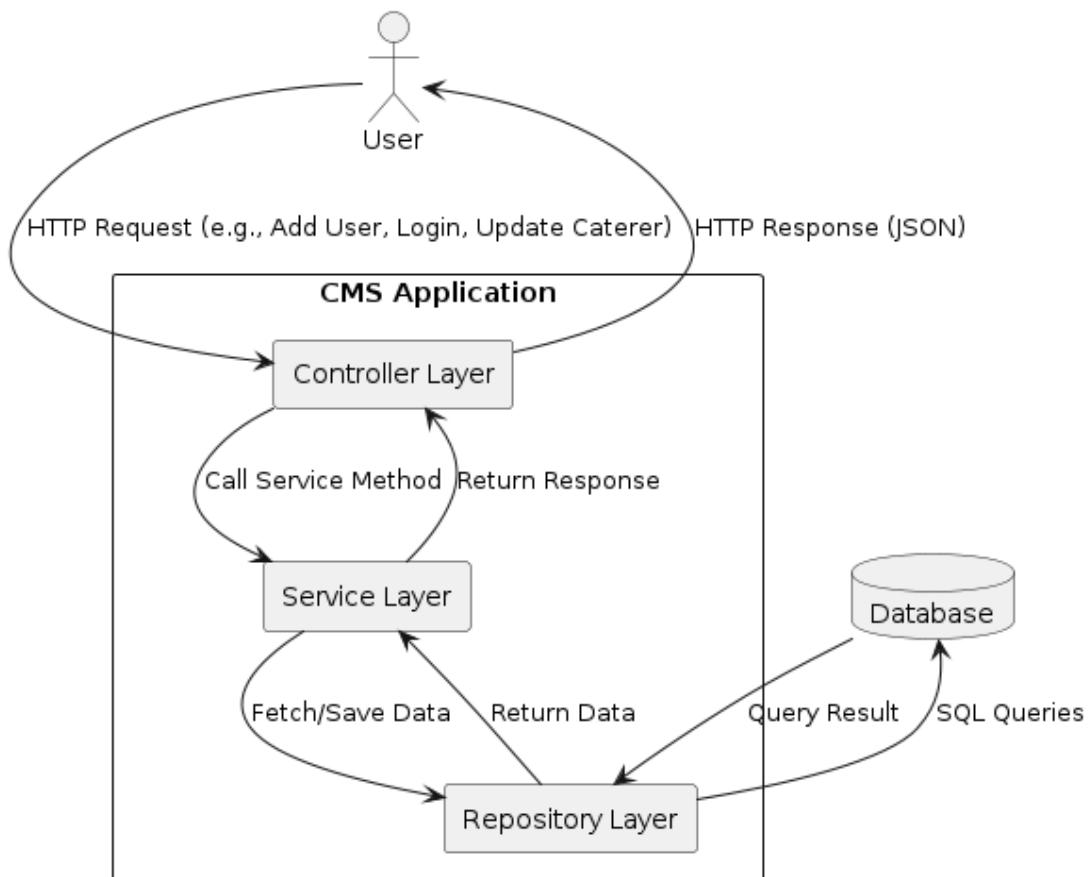
Entity Relationship Diagram

Canteen Management System

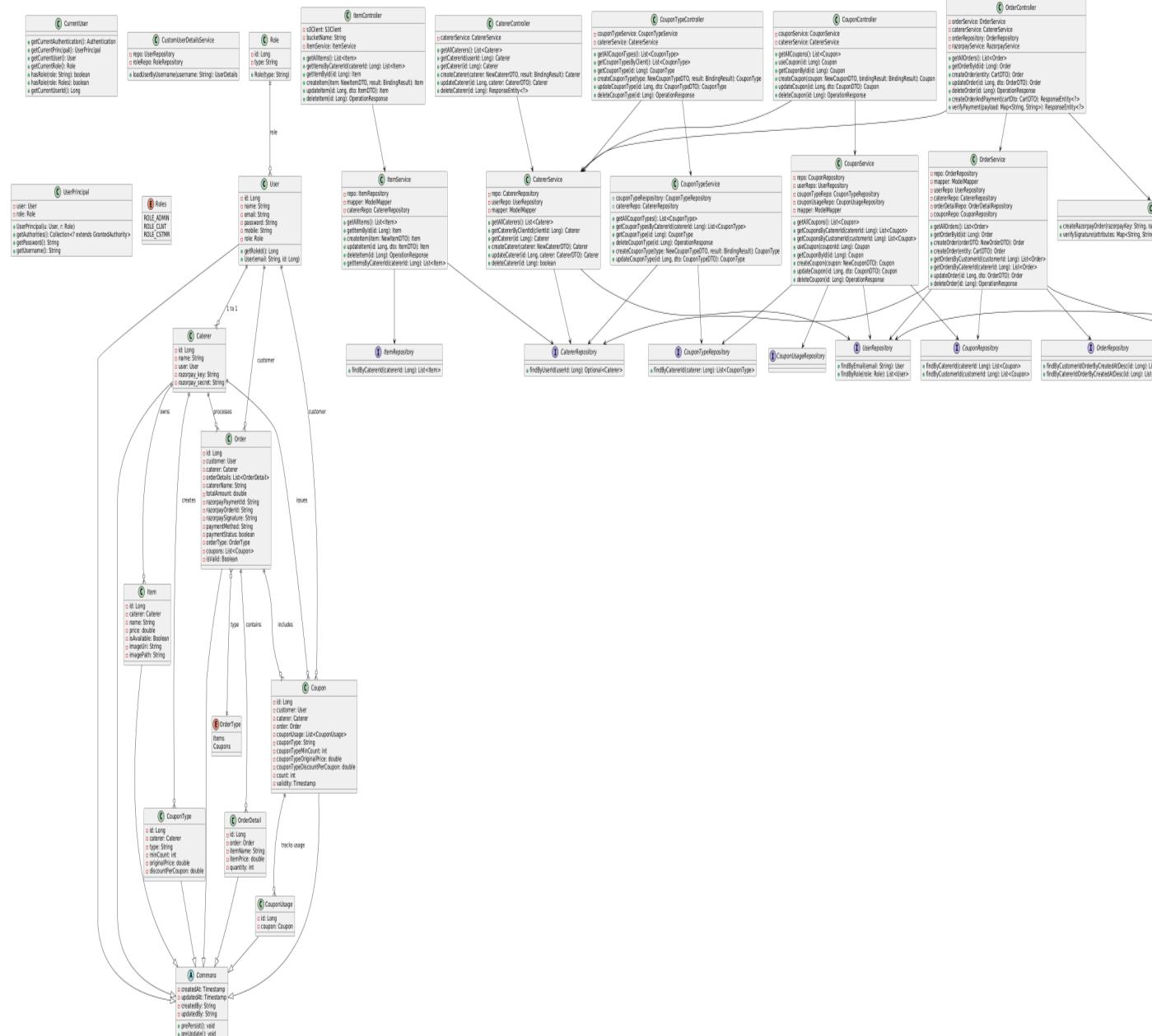


Data Flow Diagram:

Data Flow Diagram - Caterer Management System

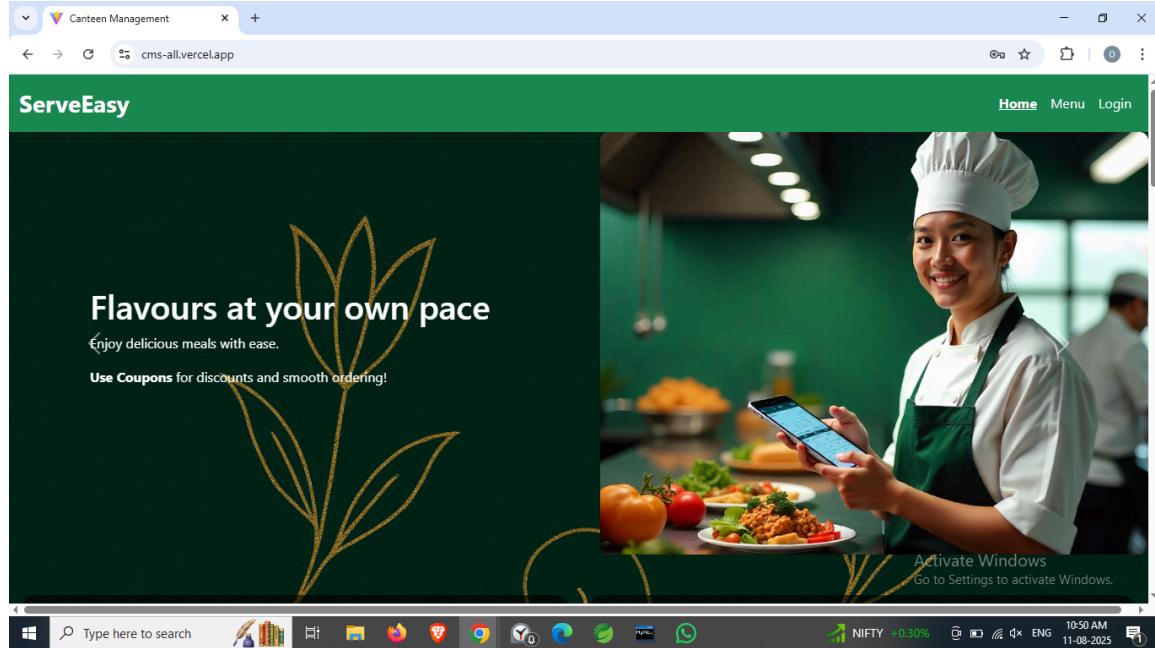


Class Diagram:

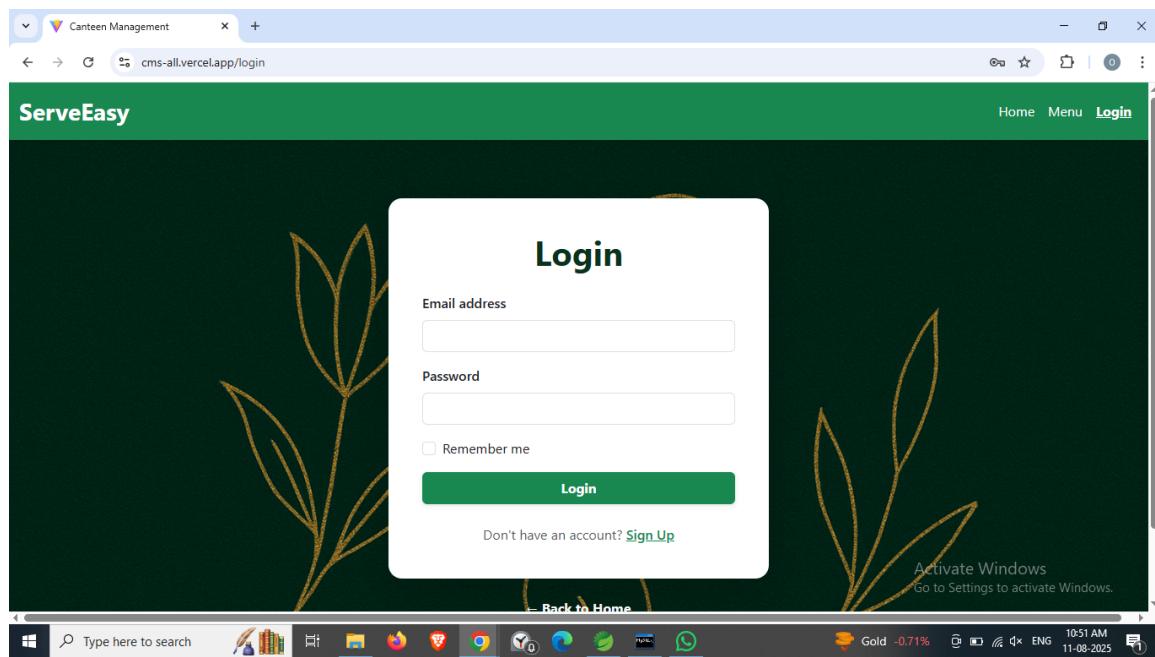


Appendix B

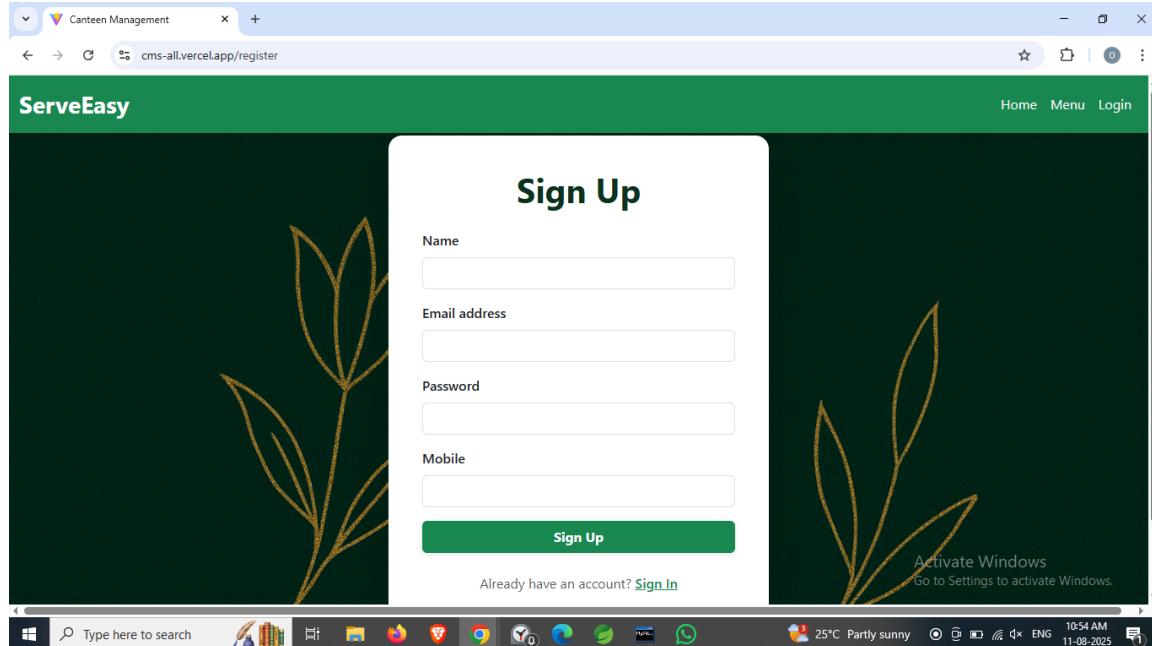
Homepage:



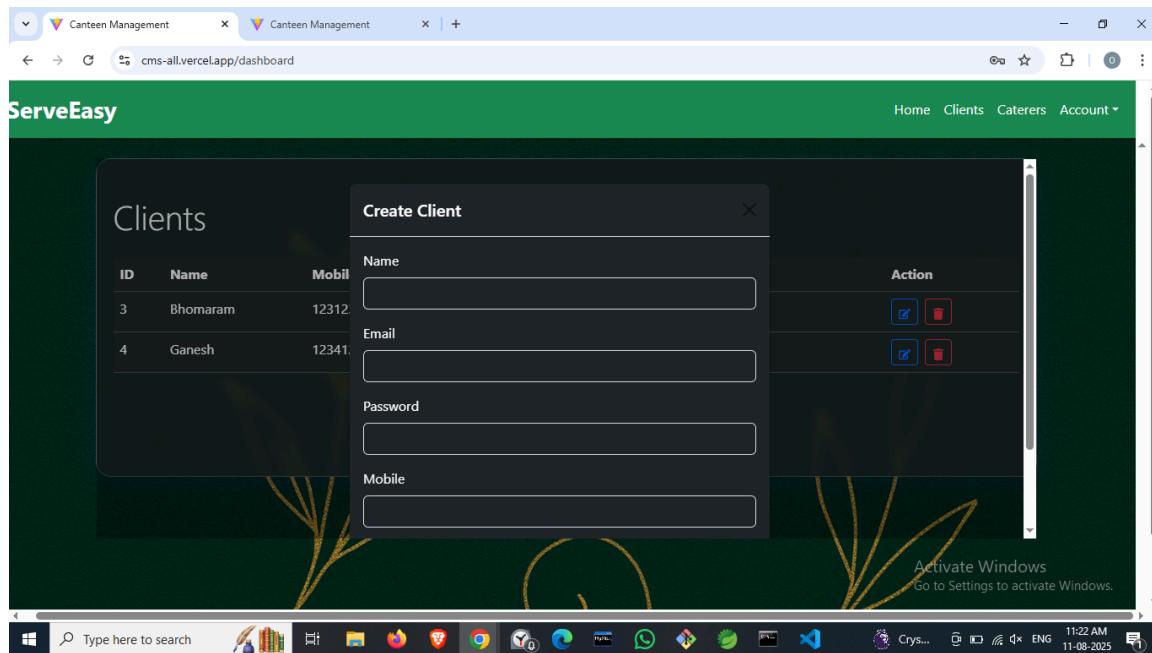
Login Popup:



User Registration:



Add New Client:



See All Caterers:

Client Dashboard:

Order ID	Customer	Amount	Status	Date	Notes
#15	Bhuvan	₹55	Paid	11 Aug, 08:06 am	Activate Windows Go to Settings to activate Windows.
#14	Naveen	₹1950	Paid	11 Aug, 06:02 am	

Add FoodItem:

Name	Price (₹)
Poha	₹25
Vadapav	₹25
Idli	₹30
Samosa	₹20

Add Coupons:

ID	Type
3	Breakfast
4	LunchDinner
5	LunchDinnerHalf

View All Sold Coupons:

Type	Caterer	Created	Min Count	Original Price	Discount	Remaining	Total Value	Status
Lunch	Ganesh Catering	11 Aug 2025	30	₹55	₹5	30	₹1,500	Active
LunchDinner	Laxmi Catering	11 Aug 2025	30	₹70	₹5	30	₹1,950	Active

View All Orders:

Order ID	Customer	Date	Items	Total Amount	Payment Status
#16	Bhuvan	11 Aug 2025, 11:31 am	Poha (1)	₹25	Paid
#15	Bhuvan	11 Aug 2025, 08:06 am	Poha (1), Idli (1)	₹55	Paid
#14	Naveen	11 Aug 2025, 06:02 am	LunchDinner (30)	₹1950	Paid
#13	Bhuvan	11 Aug 2025, 05:46 am	BreakFast (30)	₹900	Paid
#11	Bhuvan	11 Aug 2025, 05:33 am	LunchDinnerHalf (30)	₹1350	Paid
#8	Naveen	11 Aug 2025, 05:23 am	null (0)	₹0	Paid
#7	Naveen	11 Aug 2025, 05:15 am	Idli (1)	₹30	Paid
#6	Naveen	11 Aug 2025, 05:13 am	Vadapav (2)	₹50	Paid
#4	Naveen	11 Aug 2025, 05:06 am	Samosa (1)	₹20	Paid
#3	Naveen	11 Aug 2025, 05:04 am	Poha (1), Idli (1)	₹55	Paid

View Menu:

Select menu type

Food Items

Select caterer

All Caterers

Select item availability

All Items



Poha

Caterer: Laxmi Catering

Price: ₹25

Add to cart



Vadapav

Caterer: Laxmi Catering

Price: ₹25

Add to cart



Idli

Caterer: Laxmi Catering

Price: ₹30

Add to cart

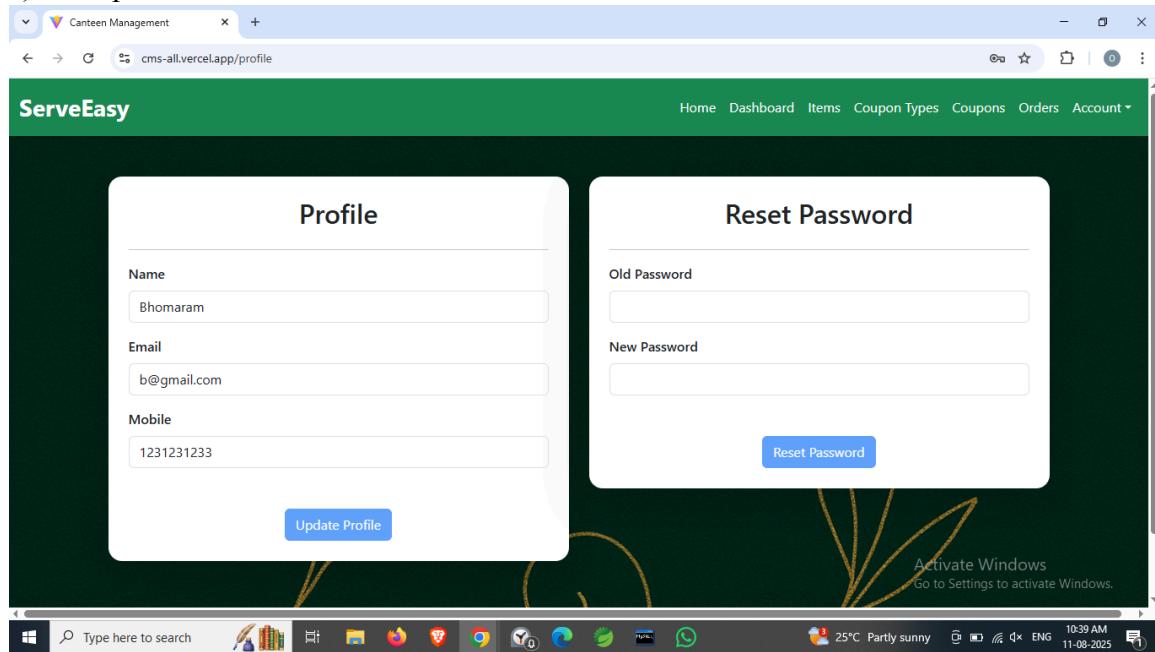
User's bought coupons:

UserProfileDetails:

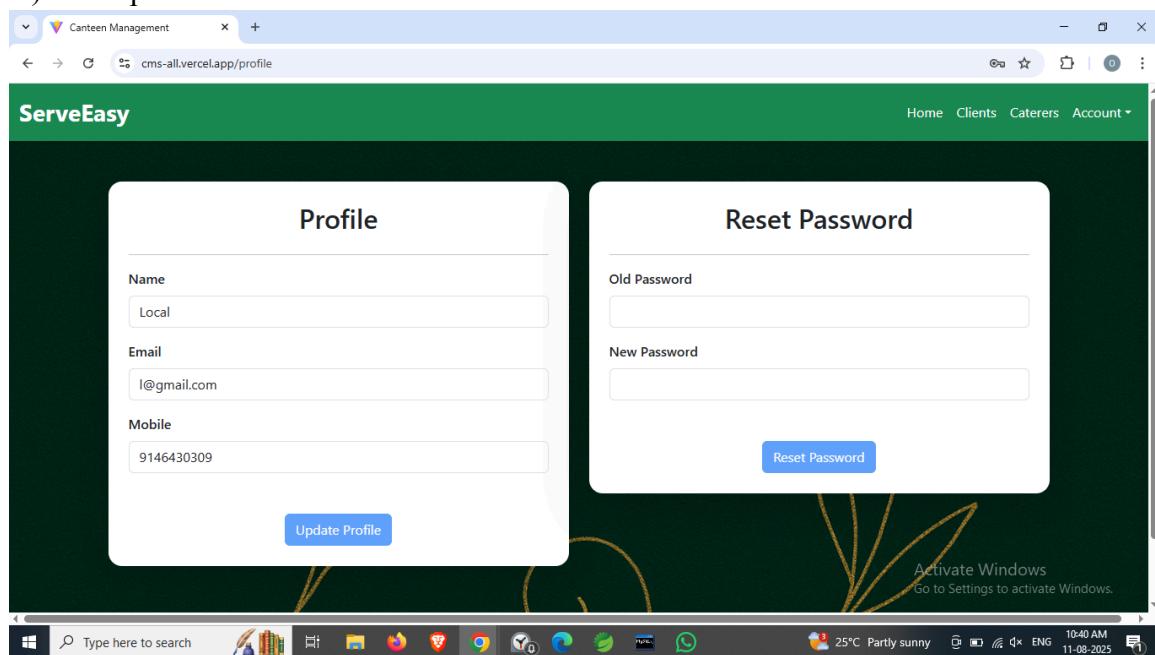
There are three types of users i.e customer, client and admin.

i) customer profile

ii)client profile



iii)admin profile



Add To Cart:

The screenshot shows a web browser window titled "Canteen Management" with the URL "localhost:5173/cart". The page has a green header with the logo "ServeEasy". On the left, there's a "Cart Items" section showing a single item: "Poha" from "Caterer: Laxmi Catering" at ₹25, with a quantity of 2. On the right, there's a "Totals" section showing "Total Items: 2" and "Total Amount: ₹50". A message says "Please review your cart before checkout." with a "Checkout" button. The background features a decorative illustration of leaves. The taskbar at the bottom shows various application icons.

Payment:

The screenshot shows a web browser window titled "Canteen Management" with the URL "localhost:5173/cart". The page displays a "Price Summary" of ₹25 and a note about using a phone number for payment. It then lists "Payment Options" including "Cards", "Netbanking", "Wallet", and "Pay Later". The "Netbanking" section is expanded, showing a list of suggested banks: Bank of Baroda - Retail Banking, Canara Bank, Punjab National Bank - Retail Banking, PNB (Erstwhile-United Bank of India), and IDBI. A red ribbon in the top right corner says "Test Mode". The taskbar at the bottom shows various application icons.

7. REFERENCES:

<https://www.google.com>

<https://spring.io/> (Spring Boot Official Site)

<https://react.dev/> (React Official Documentation)

<https://www.mysql.com/> (MySQL Database Official Site)

<https://www.w3.org/> (World Wide Web Consortium)

<https://developer.mozilla.org/> (MDN Web Docs)

<https://razorpay.com/docs/> (Razorpay Payment Gateway Documentation)

<https://www.wikipedia.org>