
S. NO.	TITLE	PAGE NO.
1	Introduction	4
2	Problem Definition	6
3	Problem Statement	7
4	Objectives	8
5	Software Requirements Specifications	10
6	Design(DFD Diagram)	13
7	Implementation	15
8	Database Design	18
9	Output Screenshots	21
10	Reference	31

1. Introduction

In the age of digitization, web technologies are the backbone of modern applications and services. From online banking to food delivery, nearly every interaction we perform is facilitated by a web-based interface. Web development has evolved significantly, enabling developers to create dynamic, user-friendly, and highly scalable applications. This project, titled "Web-Based Café Management System," exemplifies the use of modern web technology tools to streamline café operations and enhance customer experiences.

Web Technology includes a broad set of tools and standards, such as HTML, CSS, JavaScript for front-end design, and server-side scripting with PHP, Python, or Node.js. These technologies enable the creation of interactive and responsive web applications that run seamlessly on a variety of devices. On the back-end, databases like MySQL are used to store and retrieve persistent data, allowing users and admins to interact with stored records securely and efficiently.

This project is primarily developed using:

- HTML5 and CSS3 for structuring and styling the interface.**
- JavaScript for interactivity and basic client-side logic.**
- PHP for server-side operations, session handling, and communication with the database.**
- MySQL to store and manage data like users, menu items, orders, and employee records.**

The aim is to replace outdated, paper-based café management methods with an interactive, centralized, and reliable system that improves efficiency and customer satisfaction. Users can browse categorized menus (food, coffee, shakes, desserts), place orders, view order history, and manage their profiles. Admins can manage employees, assign servers, and monitor all order activities from a central dashboard.

Additionally, this system integrates API services for location auto-fill using pincodes, and employs secure session handling to ensure safe user logins. The project aligns closely with the core topics of the Web Technology subject, including:

- Client-server interaction,**
- Form submission and validation,**
- Session-based authentication,**
- File upload handling,**
- SQL integration for dynamic content,**
- and responsive UI design.**

By the end of the development cycle, the system is expected to fully automate the front and back-end operations of a café using practical web development skills acquired through the Web Technology course.

2. Problem Definition

Many small cafés and restaurants continue to rely on manual systems for order taking, billing, employee tracking, and customer data management. These paper-based or offline systems introduce a range of inefficiencies. Waiters must remember or write down orders, which can easily result in incorrect or forgotten items. Physical menus become outdated and require reprinting. Customers cannot track their orders or history, and café owners lack real-time data for operational decisions.

This lack of digitization also limits the business's potential to expand or integrate with modern platforms like online delivery services, mobile ordering, or customer feedback systems. Additionally, it restricts personalization, since customer preferences, order history, and feedback aren't stored for future engagement.

From a technology perspective, the problem lies in the absence of a centralized system that can manage the following:

- Real-time order placement and processing,
- Automatic assignment of serving staff,
- Secure login and profile management,
- Dynamic menu updates,
- Historical tracking of orders and reviews.

These gaps hinder the ability to scale or even maintain service quality in peak hours. Moreover, there's no user-friendly way to view order details, apply taxes dynamically, or notify servers about incoming orders.

In this project, we identify these challenges and address them using a Web Technology-based approach. Our proposed solution creates a centralized, digital café management system that provides:

- Customers with a web interface to browse menus and place orders.
- Admins with tools to manage staff, users, and orders.
- A back-end database to store and analyze order history and performance metrics.

By transitioning café operations from manual to digital, we aim to significantly reduce human error, speed up service, and offer valuable insights to business owners using the data stored in the system.

3. Problem Statement

To develop a fully functional web-based café management system using Web Technologies that can enhance and automate all core functions of a small or mid-sized café. The system must serve both customers and administrators by offering dynamic features and backend integration with a structured MySQL database.

The key problems to solve through this project are:

- 1. Order Management:** Automate the process of placing and recording customer orders. Instead of handwritten or verbal orders, customers should place orders through a digital interface.
- 2. Employee Assignment:** Orders should automatically be assigned to available employees with a "Server" role in the system. This reduces manual intervention and confusion.
- 3. Menu Display and Updates:** The menu should be dynamically rendered from a database, categorized, and easily modifiable by the admin without changing HTML files manually.
- 4. Cart Functionality:** Items added to the cart must be saved server-side using sessions and tables, ensuring persistence even if the page reloads.
- 5. Secure Login and User Management:** Customers should register, log in securely, and manage their profiles, including uploading a profile picture and updating their address.
- 6. Order History Tracking:** Each placed order must be saved in the database with details like items ordered, price, tax, date/time, and assigned server.
- 7. Admin Dashboard:** A centralized admin view is needed to monitor all activities—new users, employee roles, order statuses, and login information using SQL views.

By solving these, the system provides a robust platform that handles the real-world problems faced by cafés using core Web Technology principles like client-server communication, form handling, session-based authentication, database integration, and responsive UI development.

4. Objectives

The main objective of this project is to demonstrate the real-world application of Web Technology concepts through the development of a fully operational web-based café management system. The system is expected to automate café operations such as order taking, employee assignment, and user profile management, while also offering a rich user experience through interactive web pages.

Functional Objectives:

1. User Account System

- Registration and login features using PHP sessions.
- Ability for users to manage personal information and upload profile images.
- Manual Address Filling.

2. Menu and Cart Features

- Dynamically display categorized menu items pulled from a MySQL database.
- Provide “Add to Cart” functionality with quantities and price calculations.
- Cart items are stored server-side (in the `cart_items` table), not in localStorage.

3. Order Processing

- On placing an order, the system saves data in the `orders` and `order_items` tables.
- Automatically assign an employee (role = "Server") randomly to each order.
- Include tax calculations, timestamp, and order confirmation notification.

4. Admin Functions

- View and manage users, orders, and employees.
- SQL view for displaying login data while hiding sensitive info (like passwords).
- Admin panel interface for order and employee monitoring.

Technical Objectives:

- Utilize HTML, CSS, JavaScript for front-end development.
- Implement PHP for back-end logic and form processing.
- Use MySQL to build and query the relational database schema.

- **Ensure secure session handling and form validation.**
- **Organize code modularly for scalability and maintainability.**
- **Design a responsive layout using CSS3 to ensure cross-device compatibility**

5. Software Requirements Specification (SRS)

1. Purpose

To define the complete software requirements for the development of a web-based café ordering system, which allows users to register, browse menu items, place and track orders, and for admin to manage backend operations.

2. Overall Description

2.1 Product Perspective

- This is a web-based system for a café business.
- Consists of two main roles: User and Admin.
- Data is stored and retrieved from a MySQL database.
- The system connects the frontend (HTML/CSS/JS) to backend (PHP) and database (SQL).

2.2 User Characteristics

- Users: Customers of the café; not technically skilled.
- Admins: Café staff or managers; moderate technical knowledge.
- Employees: Assigned automatically to orders.

2.3 Operating Environment

- Platform: Web browser (Chrome, Firefox, Edge)
- Server: Apache (XAMPP, LAMP, Wamp64etc.)
- Languages: PHP, HTML, JavaScript, CSS, JSON
- Database: MySQL

3. Functional Requirements

A. User Module

- Register (create account)
- Login (authentication)
- View & update profile
- Add items to cart

- Place an order
- Track order status
- View order history
- Submit reviews
- Contact the café

B. Admin Module

- Add/Delete menu items
- View users and their login details
- Receive and assign orders to employees
- Add employees to the system
- Display ads on home page
- View reviews and contact messages

C. System Functions

- Automatically assign an employee (role = Server) to each order
- Save all user activities (login, order, review) in the database
- Display bills and assign order IDs

4. Non-Functional Requirements

4.1 Performance

- The system should respond to user actions within 1–2 seconds.
- Able to handle multiple users simultaneously (scalable for 50+ users).

4.2 Usability

- Simple and intuitive interface.
- Accessible on desktops, tablets, and mobile phones.

4.3 Reliability

- Data should be stored securely and persist across sessions.
- Users should not lose order/cart data once logged in.

4.4 Security

- Passwords should be hashed.
- Inputs should be sanitized to avoid SQL injection.

- Session-based authentication for secure access.

4.5 Maintainability

- Modular code structure to easily update menu, employee data, or UI design.

4.6 Portability

- Should run in any modern browser with internet access.

5. External Interfaces

5.1 User Interface

- Web-based UI with navigation to login, menu pages, cart, order history, and profile.
- Admin panel to access system backend functions.

5.2 Database Interface

- Communicates with MySQL to perform CRUD operations for users, items, orders, reviews, and employees.

5.3 Hardware Interface

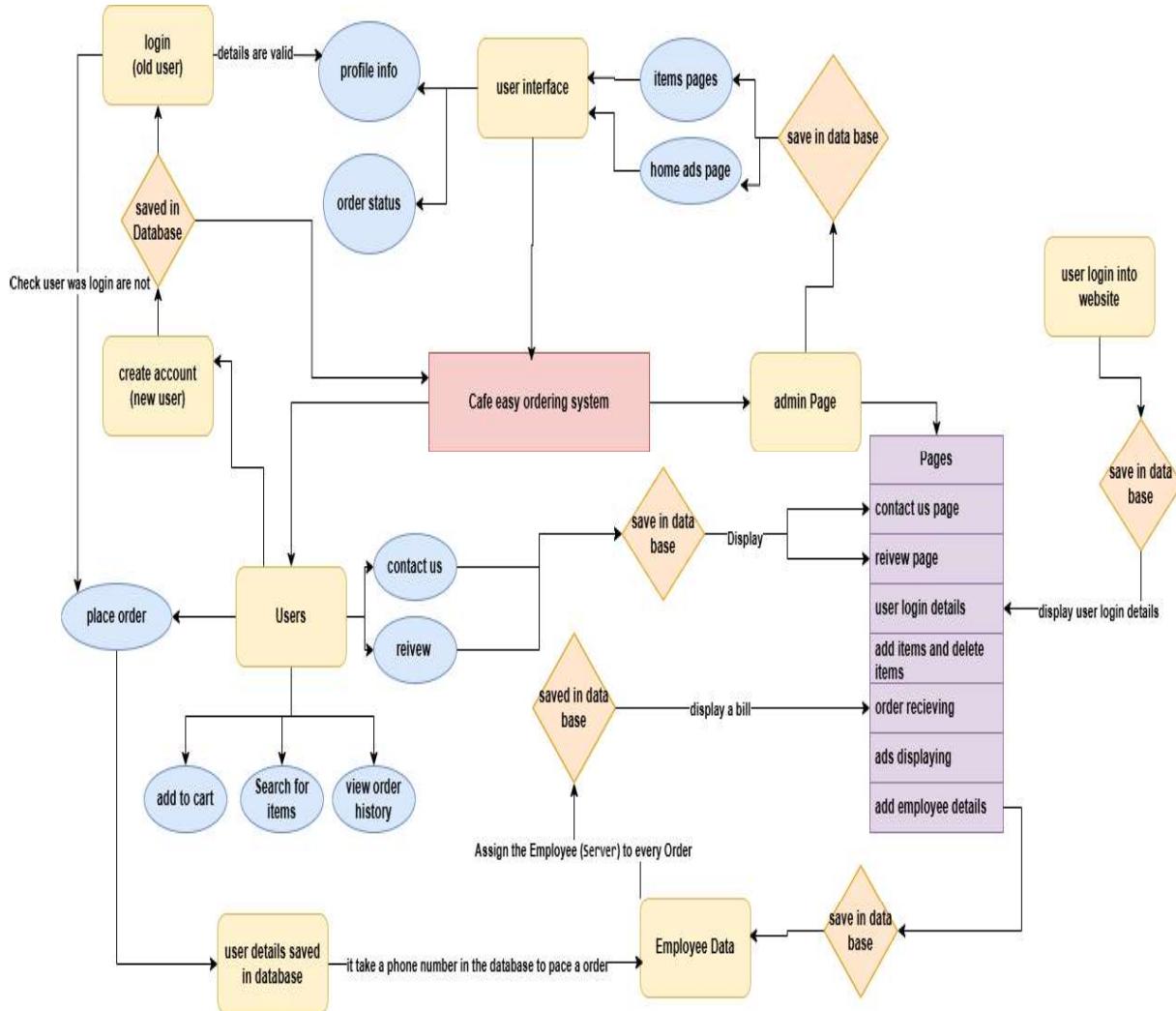
- No specific hardware needed; any device with a browser is sufficient.

6. Data Requirements

Main tables:

- **cafe_users:** user details
- **menu_items:** food/drinks info
- **orders:** order summary
- **order_items:** items in each order
- **employees:** staff info
- **cart_items:** current cart for users
- **customer_reviews:** reviews submitted
- **contact_messages:** contact queries

6. Data Flow Diagram



Flow Explanation (Step-by-Step)

1. User Login

- o If the user exists in the DB, show profile info and order status.
- o Else, they are redirected to create an account.

2. Main User Actions

- o Add to cart
- o Search for items
- o View order history

- Place order → order is saved → employee is assigned

3. Admin Actions

- Can access all system pages
- Add/delete items, assign employees, and view orders

4. Feedback

- Users can leave reviews or contact support.

5. Order Flow

- Order is placed → saved → assigned employee → bill is displayed

7. Implementation

The implementation of the Web-Based Café Management System involves the integration of all planned components into a functional and interactive web application. The development is carried out using the WAMP64 stack, which includes Windows (OS), Apache (web server), MySQL (database), and PHP (server-side scripting). This stack provides a robust environment for running, testing, and deploying dynamic web applications locally.

1. Front-End Implementation

The front-end is designed using HTML5, CSS3, and basic JavaScript for validation and interaction. Key UI components are spread across multiple pages:

- **Home Page:** Acts as an entry point, directing users to log in or browse the menu.
- **Menu Pages (food.php, coffee.php, shakes.php, desserts.php):** Menu items are retrieved from the database and displayed in structured categories. Each item includes a photo, name, and price, and can be added to the cart.
- **Login & Registration Pages (login.php, create_account.php):** Handle secure user authentication.
- **Profile Page (profile.php):** Displays user details with options to update address, upload a profile picture, and use pincode-based location auto-fill.
- **Cart and Order Pages (mycart.html, or.html):** Display cart items and order history using server-stored session data and MySQL queries.

CSS is used for styling without affecting the existing layout. JavaScript handles basic interactivity, while PHP does all dynamic data handling.

2. Back-End Implementation

PHP scripts form the core of the back-end logic:

- **User Session Management:** Secure session handling to track logged-in users.
- **Dynamic Data Rendering:** PHP scripts fetch menu items and order history from MySQL.
- **Order Placement:** Cart items are submitted via a form, and PHP inserts the data into orders and order_items tables.
- **Random Server Assignment:** When an order is placed, a random employee with the role "Server" is assigned using a MySQL query inside the PHP script.

The backend includes the following key files:

- **save_order.php:** Receives POST data and processes the order.

- **fetch_orders.php** and **fetch_received_orders.php**: Retrieve orders for display.
- **manage_employees_and_orders.php**: Admin functions to add employees and manage orders.
- **upload_profile.php**: Handles user profile updates, including image uploads.

All PHP scripts are tested using WAMP64's localhost environment, making development and debugging smooth on a Windows platform.

3. MySQL Database Integration

All persistent data is stored in a MySQL database managed through phpMyAdmin provided by WAMP. The schema is designed with proper normalization and indexing for performance.

Key tables:

- **cafe_users**: Stores user details including email, password (hashed), address, etc.
- **menu_items**: Stores menu data including item names, images, prices, and categories.
- **orders & order_items**: Store complete order records with timestamps, taxes, totals, and itemized lists.
- **employees**: Employee data including roles ("Server", "Chef", etc.).
- **cart_items**: Tracks cart items by session/user ID before placing orders.
- **login_logs_view**: SQL view to show login details for admin, with password hidden for security.

4. Session-Based Cart System

Instead of relying on localStorage, a session-based cart was implemented using PHP sessions and a **cart_items** table. This ensures:

- Cross-device persistence,
- Secure and server-side storage,
- Automatic clean-up after checkout.

When the user adds items to the cart, PHP stores them in **cart_items**. On checkout, items are moved to **order_items**, and the cart is cleared.

5. Admin Features

Admins can:

- **View user logins from a secure view (`login_logs_view`),**
- **Add new employees,**
- **Track orders and assigned servers,**
- **Access all order data in real-time using a centralized dashboard (`user_dashboard.php`).**

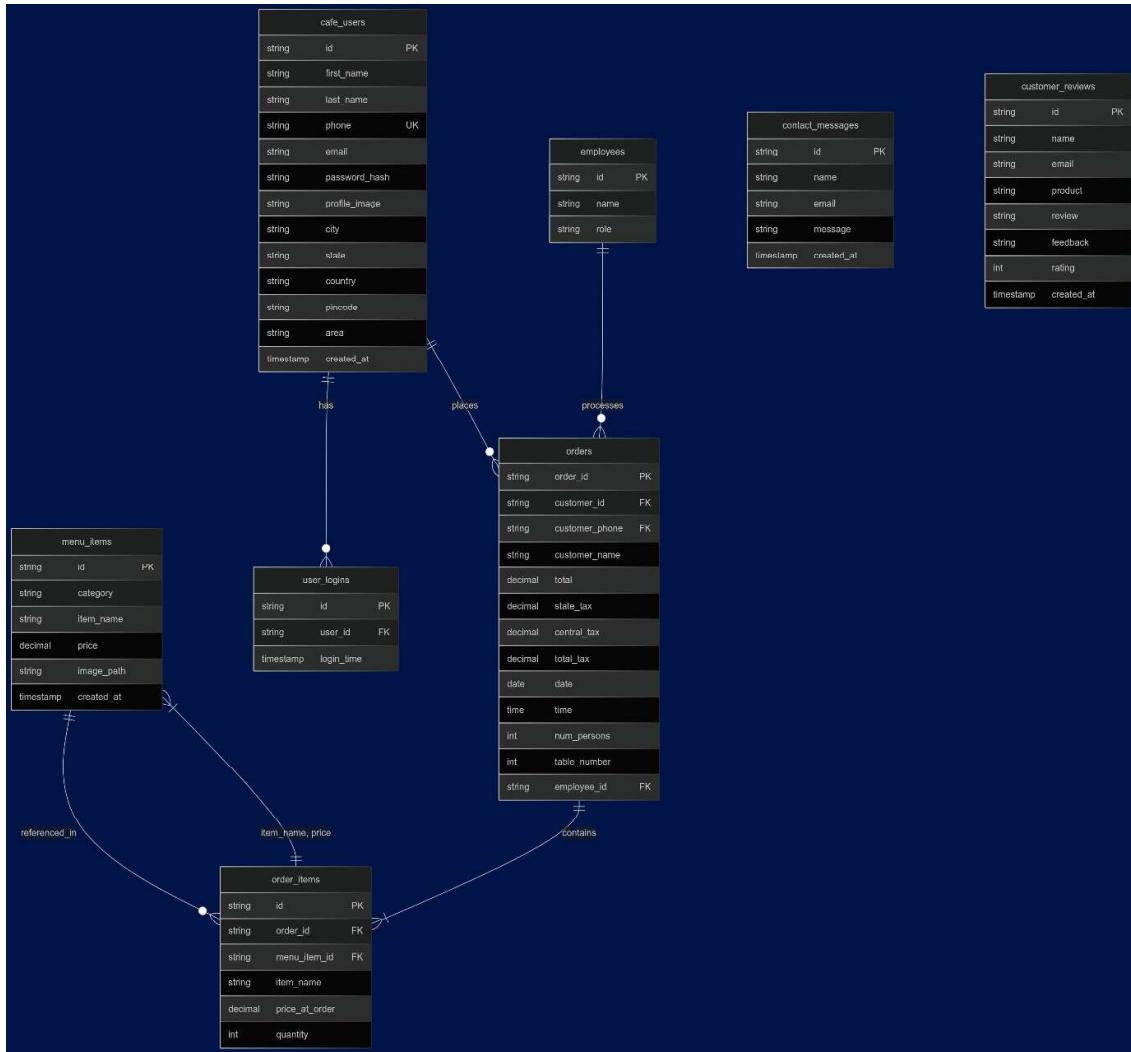
The admin system enhances café management by simplifying operations.

6. Error Handling

Error messages and validations are handled on both the client and server side:

- **PHP uses `isset()`, `empty()`, and try-catch for handling data and connection errors.**
- **File uploads are validated for type and size.**
- **JavaScript handles real-time form validation (e.g., required fields, valid email format).**

8.Database Design



The **Café Management System** uses a relational database built with **MySQL** under the WAMP64 stack. The design includes normalized tables and foreign key relationships to manage users, menu items, orders, employees, and additional features like reviews and contact messages. Below is a summary of each table and its role in the system:

1. `cafe_users`

Stores customer account details.

- **Primary Key:** `id`
- Contains name, phone, email (unique), password hash, profile image, and full address (city, state, country, pincode, area).
- Includes `created_at` timestamp to track registration.

2. `user_logins`

Tracks each login activity.

- **Primary Key:** id
- **Foreign Key:** user_id → cafe_users(id)
- Captures login time for admin dashboard and user analytics.

3. menu_items

Holds all available food, drinks, and desserts.

- **Primary Key:** id
- Includes category (e.g., Food, Coffee), item_name, price, and image_path.

4. orders

Stores full order details.

- **Primary Key:** order_id
- **Foreign Keys:**
 - customer_id → cafe_users(id)
 - employee_id → employees(id) (assigned server)
- Includes customer name/phone, table number, number of persons, total cost, taxes (central, state), and timestamp.

5. order_items

Stores individual items in each order.

- **Primary Key:** id
- **Foreign Keys:**
 - order_id → orders(order_id)
 - menu_item_id → menu_items(id)
- Includes item name, price, and quantity ordered.

6. employees

Contains details of all café staff.

- **Primary Key:** id
- Fields: name, role (e.g., Server, Chef).
- Linked to orders via employee_id for staff assignment.

7. contact_messages

Handles messages sent via the contact form.

- **Primary Key:** id
- Fields: name, email, message, created_at

8. customer_reviews

Stores customer feedback and ratings.

- **Primary Key:** id
- Fields: name, email, product, review, feedback, numeric rating, and timestamp.

Design Highlights

- All relationships are normalized to avoid redundancy.
- Foreign keys enforce data integrity.
- Designed to support advanced queries like order history, staff assignments, and user analytics.

This well-structured design supports the complete functioning of a real-world café web application with reliability and scalability.

9.Output Screenshots

USER PAGES:-

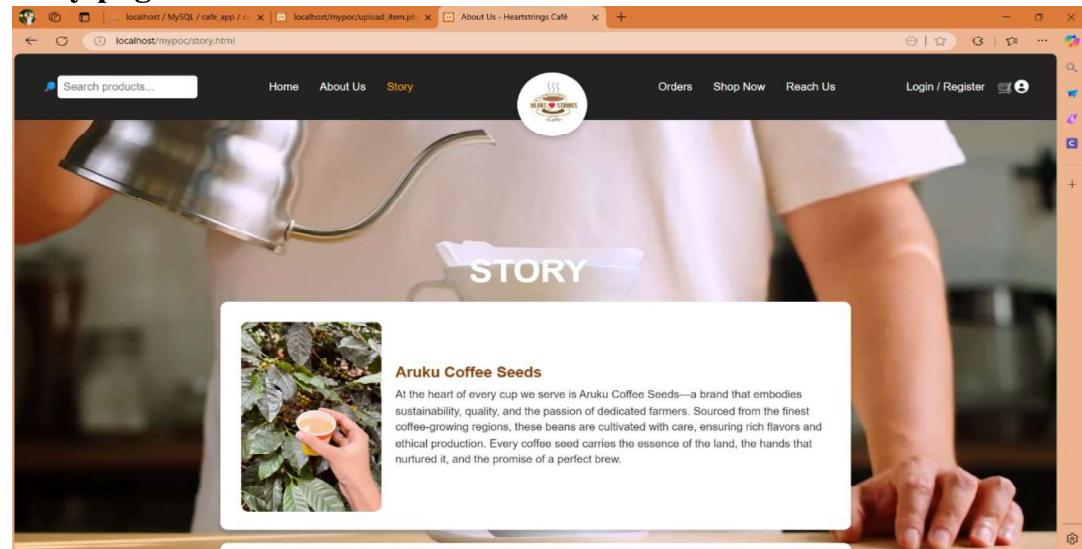
Home page



About us page



story page



order history page

The screenshot shows a web browser window with the title "Order History". The main content area displays a large image of a coffee cup with latte art. Overlaid on the image is a white rectangular box containing the following order details:

Order Number: 20
Customer ID:
CUST11485
GST No:
G59945AF85GNT9
Customer Name:
[REDACTED]
Date:
2023-04-13
Time:
4:58:22 pm
Number of Persons:
3
Table Number:
B1
Mocha - ₹190 x 1
State Tax (5%):
₹9.50
Central Tax (10%).

food menu page

The screenshot shows a web browser window with the title "COFFEE MENU". The main content area displays a grid of eight coffee images with their names and prices:

Item	Price
Nitro	₹210.00
Cartado	₹200.00
Iced Coffee	₹190.00
Cappuccino	₹180.00
Latte Coffee	₹160.00
Mocha	₹190.00
Americano	₹180.00
Affogato coffee	₹150.00

Each item has an "Add to Cart" button below it.

FAQ page

The screenshot shows a web browser window with the title "FAQs". The main content area features a large graphic with the word "FAQ" in green, surrounded by stylized figures of people. Below the graphic is a list of frequently asked questions:

- How will I manage my business with you? ▾
- Is the content on this website available in other languages? ▾
- What does it mean to be a part of our booking platform? ▾
- How to order the items? ▾
- The user manual ▾
- What if I have more questions? ▾

Contact us page

The screenshot shows a contact form for a website. At the top, there's a navigation bar with links for Home, About Us, Story, Orders, Shop Now, Reach Us, and Login / Register. A search bar is also present. The main content area has a heading "CONTACT US" and a sub-heading "Hello, CUSTOMERS!". It includes a note: "Give us a call or send a note. We will be happy to answer your questions." Below this are sections for "CONTACT" (with icons for phone and email) and "LOCATION" (with an icon for location and the text "opposite to SRM University nerukonda,mangalagiri-guntur"). To the right, there's a "SEND A NOTE" form with fields for Full Name, Email, and Question, followed by a "SEND" button.

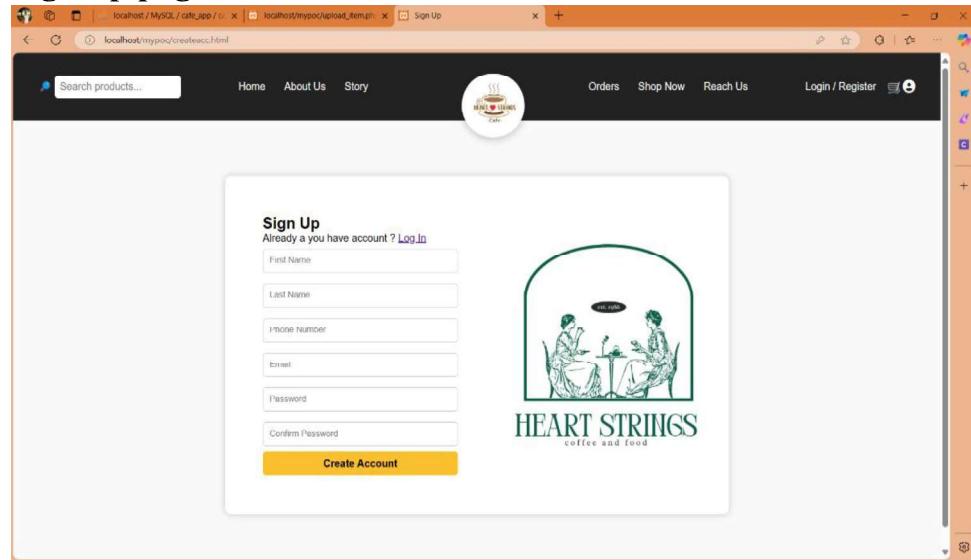
Review page

This screenshot shows a review submission form. The top navigation bar is identical to the previous one. The main form is titled "Submit Your Review". It contains fields for Name, Email, a dropdown menu for "Select Product", a text area for "Review on product:", a text area for "Feedback / Complaint:", a rating scale from 1 to 5 stars, and a "Submit Review" button at the bottom.

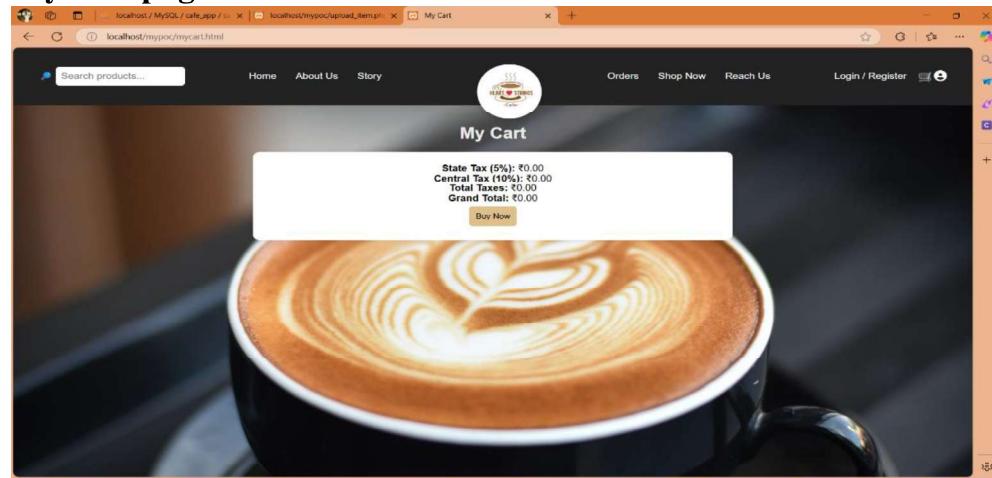
Login page

The screenshot shows a login form. The top navigation bar is consistent. The main form is titled "Sign In" and contains fields for Email and Password, a "Login" button, and links for "Forgot Password?" and "Create a New Account". Below these are social media login options for Apple, Google, and Facebook. At the bottom right of the page, there's a footer note: "Opposite to SRM University AP Guntur".

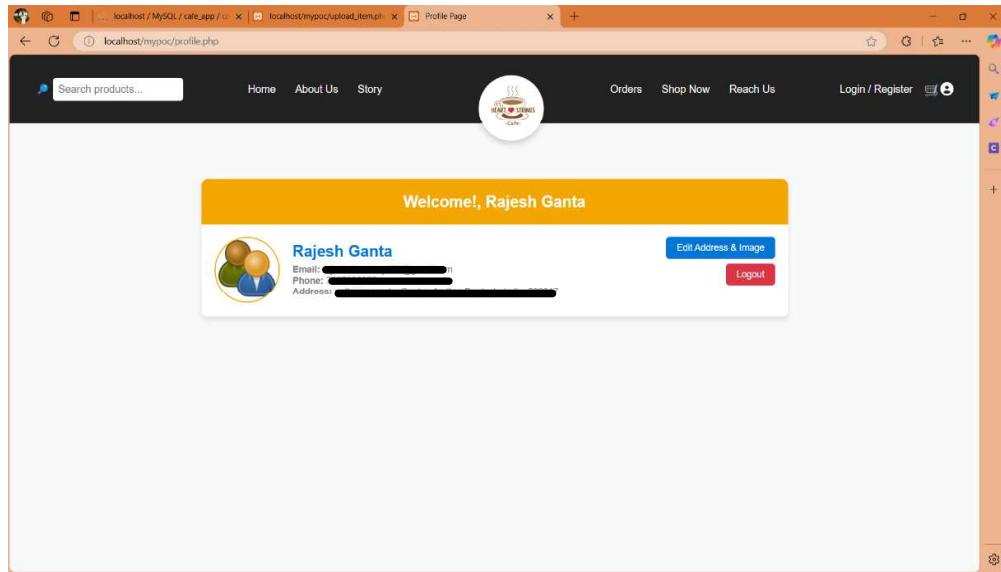
Sign-up page



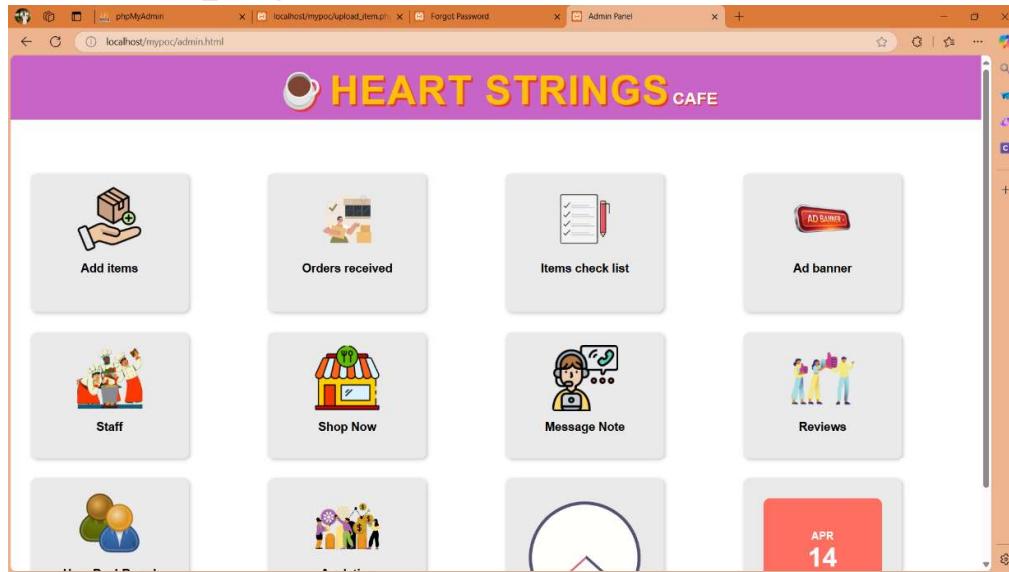
My cart page



Profile page



Admin pages



Items adding page

The screenshot shows a web browser window with the title "ADD ITEMS". The form contains the following fields:

- Category: A dropdown menu set to "Food".
- Item Name: An input field.
- Price: An input field.
- Image: A file input field labeled "Choose File" with "No file chosen" displayed. Below it is an orange "Upload" button.

Order Received page of admin

The screenshot shows a web browser window with the title "ORDERS RECEIVED". The page displays an "Admin Orders" section with the following details for Order ID: CUST1485:

Order ID: CUST1485
GST No: G99045AP65GNT9
Customer Name: [REDACTED]
Phone Number: [REDACTED]
Date: 2025-04-13 at 04:58:22
Number of Persons: 3
Table Number: B1
Assigned Server: abhi
Items:

- [REDACTED] Unknown - ₹190.00 x 1 = ₹190.00

State Tax (5%): ₹9.50
Central Tax (10%): ₹19.00
Total Tax: ₹28.50
Grand Total: ₹219.00

Added items check list

The screenshot shows a web browser window titled "Added Items" with a yellow header. Below the header, there is a "Sort: Newest First" dropdown and a "+ Add New Item" button. The main content area is titled "ADDED ITEMS". It displays a list of items in a grid format:

Category	Name	Price	Action
COFFEE	Nitro - ₹210.00	₹210.00	Delete
COFFEE	Cartado - ₹200.00	₹200.00	Delete
COFFEE	Iced Coffee - ₹190.00	₹190.00	Delete
DESSERTS	Nanaimo Bar - ₹190.00	₹190.00	Delete
DESSERTS	Jam Cake - ₹130.00	₹130.00	Delete
SHAKES	Salted Caramel-Pretzel Milkshake - ₹300.00	₹300.00	Delete
SHAKES	Oreo Milkshake - ₹170.00	₹170.00	Delete
SHAKES	Lemon pie Milkshake - ₹215.00	₹215.00	Delete

Upload ads on home page

The screenshot shows a web browser window titled "Admin Panel - Background Slides" with a yellow header. Below the header, there is a "Choose File" button, a "No file chosen" message, and an "Add to Slideshow" button. The main content area is titled "UPDATE SLIDESHOW". It displays a list of images in a grid format:

Image Preview	Action
	Delete

Review by customer

The screenshot shows a web browser window titled "Customer Reviews" with the URL "localhost/mypoc/review_display.php". The main content area has a purple header bar with the text "REVIEW BY CUSTOMER". Below this is a white form-like section containing the following data:

Name: Rajesh
Email: [REDACTED]
Product: ALL
Review: super
Feedback: no
Rating: ★★★★
Submitted On: 12-04-2025

The browser interface includes a top navigation bar with icons for search, refresh, and other functions, and a sidebar on the right with a "HOME" button and other navigation links.

Staff details page

The screenshot shows a web browser window titled "Staff Management" with the URL "localhost/mypoc/staff.php". The main content area has a purple header bar with the text "Staff & Your Team". Below this is a white form-like section containing three staff entries:

Employee Name	Role	ID	Action
Rajesh	Chef	EMP_101	Delete
abhi	Server	EMP_102	Delete
ravi	Server	EMP_103	Delete

The browser interface includes a top navigation bar with icons for search, refresh, and other functions, and a sidebar on the right with a "HOME" button and other navigation links.

User login Details

The screenshot shows a web browser window titled "User Login Dashboard" with the URL "localhost/mypoc/user_dashboard.php". The main content area has a purple header bar with the text "USER LOGIN DETAILS". Below this is a table with four columns: "Name", "Email", "Password", and "Login Time". The table contains 10 rows, each representing a login entry for a user named "rajesh Ganta". The "Email" column contains several redacted email addresses. The "Login Time" column shows various dates and times from April 13 to April 16, 2025.

Name	Email	Password	Login Time
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-16 11:01:15
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-15 14:53:09
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-15 14:27:46
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:57:30
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:14:45
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:14:36
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:05:26
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:04:21
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-14 10:03:28
rajesh Ganta	[REDACTED]	xxxxxx	2025-04-13 10:00:58

Analytics DashBoard

The screenshot shows a web browser window titled "Cafe Analytics Dashboard" with the URL "localhost/mypoc/analytics.html". The main content area has a purple header bar with the text "Cafe Analytics Dashboard". Below this is a section titled "Filter by Date" containing input fields for "Start Date" and "End Date" with "Apply Filter" and "Reset Filter" buttons. Below the filter is a section titled "Orders by Date" featuring a line chart. The chart has "Order Count" on the legend and "Number of Orders" on the y-axis (ranging from 1.0 to 3.0). The x-axis represents time, and a single data point is plotted at approximately (1.0, 3.0).

References

1. MySQL Documentation
MySQL 8.0 Reference Manual
<https://dev.mysql.com/doc/>
2. WAMP64 Official Website
Windows Apache MySQL PHP (WAMP) Stack
<http://www.wampserver.com/en/>
3. PHP Manual
PHP Language Reference
<https://www.php.net/manual/en/>

4. **w3schools – SQL and PHP Tutorials**
<https://www.w3schools.com/sql/>
<https://www.w3schools.com/php/>
5. **GeeksforGeeks – Web Development with PHP and MySQL**
<https://www.geeksforgeeks.org/php/>
<https://www.geeksforgeeks.org/mysql-introduction/>
6. **ERD Tool**
draw.io – Used for drawing the ER Diagram
<https://www.draw.io/>