

Catering Reservation and Ordering System

1. Project Title

Catering Reservation and Ordering System

2. Domain

Hospitality / Food Services / Web Application

3. Technologies Used

- HTML5 – Structure and layout of the application
- CSS3 – Styling, layout design, and responsiveness
- JavaScript (ES6) – Business logic and interactivity
- Browser Memory (In-memory arrays) – Data handling for demo purposes

4. Project Overview

The Catering Reservation and Ordering System is a web-based application designed to manage catering product selection, cart handling, order placement, and administration in a single platform. The system supports two roles: **User** and **Admin**. Users can browse catering items, place orders, and manage their bookings, while admins can manage products and view all orders.

This project is developed as an internship-level assignment to demonstrate frontend development skills, logical flow design, and role-based system implementation.

5. Objectives

- To automate catering order and reservation process
- To allow users to place and manage catering orders
- To provide admin control over products and orders
- To demonstrate secure login and role-based access
- To build a clean and interactive UI using core web technologies

6. System Modules

6.1 Authentication Module

- User registration
- User login
- Role selection (User / Admin)
- Logout functionality

6.2 User Module

- View available catering products
- Add or remove items using quantity controls
- View cart and total amount
- Place orders with delivery and payment details
- View personal order history
- Cancel orders with penalty rules

6.3 Admin Module

- Add new catering products
- View all products
- View all customer orders
- Monitor order status

6.4 Logging Module

- Records all major system activities
- Displays timestamped logs in system log panel

7. Functional Requirements

- System shall allow users to register and login
- System shall allow users to add catering items to cart
- System shall calculate total order amount
- System shall allow order placement with delivery details
- System shall allow order cancellation with conditions
- System shall allow admin to add products
- System shall allow admin to view all orders

8. Non-Functional Requirements

- User-friendly interface
- Safe and controlled operations
- Easy to maintain and extend
- Responsive layout
- Clear role-based access control

9. System Architecture

The system follows a **client-side layered architecture**: - Presentation Layer: HTML & CSS - Logic Layer: JavaScript - Data Layer: In-memory JavaScript objects (can be replaced with database)

10. Code Organization

- HTML: UI structure for authentication, dashboards, and forms

- CSS: Styling, layout management, and visual indicators
- JavaScript:
- Authentication handling
- Cart and product management
- Order processing logic
- Admin operations
- Logging utility

11. Logging Mechanism

A logging function records system actions such as: - User login/logout - Product addition - Order placement
- Order cancellation

Logs are displayed in a dedicated system log section with timestamps.

12. Data Handling Strategy

- User data stored in JavaScript arrays
- Product data managed in arrays with quantity tracking
- Orders stored in memory during runtime
- Suitable for demonstration and can be upgraded to database or Firebase

13. Business Rules Implemented

- Cart cannot be empty during checkout
- Mandatory delivery and payment details required
- Orders cannot be cancelled within 2 days of delivery
- Cancellation penalty of 10% applied

14. Test Cases

| Test Case ID | Description | Expected Result |
|--------------|----------------------|------------------------------|
| TC01 | User login | Login successful |
| TC02 | Add product to cart | Cart updated |
| TC03 | Place order | Order created |
| TC04 | Cancel order (valid) | Order cancelled with penalty |
| TC05 | Admin add product | Product added |
| TC06 | Admin view orders | Orders displayed |

15. Optimization Techniques

- Modular JavaScript functions

- Reusable UI components
- Minimal DOM manipulation
- Role-based rendering

16. Deployment

- Can be run locally in any modern browser
- No server required
- Can be deployed as a static website

17. GitHub Repository Guidelines

- Public repository
- Well-structured commits
- README file with setup instructions
- Code comments for clarity

18. Future Enhancements

- Database or Firebase integration
- Online payment gateway
- Order status tracking
- Email/SMS notifications
- Responsive mobile UI

19. Conclusion

The Catering Reservation and Ordering System demonstrates a complete role-based web application using HTML, CSS, and JavaScript. The project showcases logical design, UI interaction, and business rule implementation, making it suitable for internship submission and technical evaluation.

Prepared for Internship Submission