FOODZA - **Online Food Ordering System   
  
MCA Final Semester Project Report Submitted by :**

 Dhyanee

 Divya Maheta



# Table of Contents :

1. Abstract
2. Introduction
3. System Architecture
4. Data Flow Diagrams
5. Entity Relationship Diagram
6. Data Dictionary
7. System Features and Functionalities
8. Implementation Details
9. Screenshots
10. Future Scope
11. Conclusion
12. References



# Abstract :

The "Foodza" project is an online food ordering system developed to provide a seamless platform for customers to order food from various restaurants. In today's fast-paced world, digital food ordering solutions have become essential for both customers and restaurants. Foodza addresses this need by offering a comprehensive web-based application that connects customers with multiple restaurants through a single platform.

The system enables users to browse restaurants, view menus, place orders, and track delivery status. For restaurant owners and administrators, it provides tools to manage menus, track orders, and maintain customer relationships. Built using HTML, CSS, JavaScript for frontend development, and PHP and MySQL for backend processing and database management, Foodza represents a fully functional e-commerce solution specifically tailored for the food service industry.

This project demonstrates the practical application of web development technologies and database management systems to create a solution that addresses real-world needs in the food delivery ecosystem.



# Introduction :

## Purpose :

The primary purpose of the Foodza online food ordering system is to bridge the gap between food establishments and customers through a digital platform. This system aims to streamline the food ordering process, reduce human errors, and enhance customer experience by providing a user-friendly interface for browsing and ordering food items from multiple restaurants.

## Project Scope :

Foodza encompasses :

 User registration and authentication system  Restaurant and menu browsing capabilities  Shopping cart functionality

 Order placement and tracking

 Admin panel for system management  Restaurant management interface

 Order processing and status updates  Customer feedback mechanism

## Project Objectives :

 Develop a responsive web application accessible on various devices  Create an intuitive user interface for seamless navigation

 Implement secure user authentication and data storage

 Establish efficient database design for storing and retrieving information  Provide comprehensive admin controls for system management

 Enable restaurant categorization for better user experience  Implement order tracking functionality for customers

 Develop reporting capabilities for business analytics

## Technologies Used :

### Frontend Technologies :

 HTML5 for structuring web pages

 CSS3 for styling and responsive design  JavaScript for client-side interactivity

### Backend Technologies :

 PHP 7.4 for server-side logic

 MySQL for database management

### Development Tools :

 XAMPP/WAMP for local development environment  Visual Studio Code for code editing

 Git for version control

 phpMyAdmin for database administration



# System Architecture :

Foodza follows a three-tier architecture consisting of:

## Presentation Layer :

The frontend interface that users interact with, developed using HTML, CSS, and JavaScript. This layer is responsible for displaying information to users and collecting their inputs.

## Application Layer :

The PHP-based business logic layer that processes user inputs, implements business rules, and manages the application workflow. This layer serves as the intermediary between the presentation layer and the data layer.

## Data Layer :

The MySQL database that stores all system data, including user information, restaurant details, menu items, and order data. This layer is responsible for data persistence and retrieval.

The three-tier architecture provides several advantages, including :  
  Separation of concerns

 Enhanced security  Scalability

 Maintainability

 Reusability of components



# Data Flow Diagrams :

## Context Level Diagram (Level 0) :

## Level 1 DFD :

## Level 2 DFD (Order Processing Subsystem) :



# Entity Relationship Diagram :



# Data Dictionary :

* 1. **Table : users**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| user\_id | INT | Unique identifier for each user | Primary Key, Auto-increment |
| username | VARCHAR(100) | User's login name | Not Null, Unique |
| password | VARCHAR(255) | Encrypted password | Not Null |
| email | VARCHAR(100) | User's email address | Not Null, Unique |
| phone | VARCHAR(20) | User's contact number | Not Null |
| address | TEXT | User's delivery address | Not Null |
| reg\_date | TIMESTAMP | Date and time of registration | Default: Current timestamp |
| C |  |  | C |

* 1. **Table: admin**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| admin\_id | INT | Unique identifier for each admin | Primary Key, Auto-increment |
| username | VARCHAR(100) | Admin's login name | Not Null, Unique |
| password | VARCHAR(255) | Encrypted password | Not Null |
| date | TIMESTAMP | Date and time of account creation | Default: Current timestamp |
| C |  |  | C |

* 1. **Table: restaurant**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| rs\_id | INT | Unique identifier for each restaurant | Primary Key, Auto-increment |
| c\_id | INT | Category ID | Foreign Key (res\_category.c\_id) |
| title | VARCHAR(255) | Restaurant name | Not Null |
| email | VARCHAR(100) | Restaurant email address | Not Null |
| phone | VARCHAR(20) | Restaurant contact number | Not Null |
| website | VARCHAR(255) | Restaurant website URL | Nullable |
| o\_hr | VARCHAR(50) | Opening hours | Not Null |
| c\_hr | VARCHAR(50) | Closing hours | Not Null |
| o\_days | VARCHAR(100) | Operating days | Not Null |
| address | TEXT | Restaurant physical address | Not Null |
| image | VARCHAR(255) | Restaurant image filename | Not Null |
| date | TIMESTAMP | Date and time of registration | Default: Current timestamp |
| C |  |  | C |

* 1. **Table: res\_category**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| c\_id | INT | Unique identifier for each category | Primary Key, Auto-increment |
| c\_name | VARCHAR(100) | Category name | Not Null, Unique |
| date | TIMESTAMP | Date and time of creation | Default: Current timestamp |
| C |  |  | C |

* 1. **Table: dishes**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| d\_id | INT | Unique identifier for each dish | Primary Key, Auto-increment |
| rs\_id | INT | Restaurant ID | Foreign Key (restaurant.rs\_id) |
| title | VARCHAR(255) | Dish name | Not Null |
| slogan | VARCHAR(255) | Short description of the dish | Nullable |
| price | DECIMAL(10,2) | Dish price | Not Null |
| img\_name | VARCHAR(255) | Dish image filename | Not Null |
| C |  |  | C |

* 1. **Table: users\_orders**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| order\_id | INT | Unique identifier for each order | Primary Key, Auto-increment |
| user\_id | INT | User ID | Foreign Key (users.user\_id) |
| dish\_id | INT | Dish ID | Foreign Key (dishes.d\_id) |
| quantity | INT | Number of items ordered | Not Null, Default: 1 |
| price | DECIMAL(10,2) | Total price for the order | Not Null |
| status | VARCHAR(50) | Order status (Pending/Delivered/etc.) | Not Null, Default: 'Pending' |
| order\_date | TIMESTAMP | Date and time of order placement | Default: Current timestamp |
| C |  |  | C |

* 1. **Table: remark**

| **Field** | **Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| id | INT | Unique identifier for each remark | Primary Key, Auto-increment |
| frm\_id | INT | Order ID | Foreign Key (users\_orders.order\_id) |
| status | VARCHAR(50) | Status update | Not Null |
| remark | TEXT | Comment or note about the order | Nullable |
| date | TIMESTAMP | Date and time of remark | Default: Current timestamp |
| C |  |  | C |

# System Features and Functionalities :

## Client-Side Features :

### User Registration and Authentication :

 Account creation with email verification

 Secure login system with password encryption   
 User profile management

 Password recovery mechanism

### Restaurant Browsing :

 View list of all available restaurants  Filter restaurants by category

 Search restaurants by name

 View restaurant details including operating hours, location, and ratings

### Menu Exploration :

 Browse menu items by restaurant

 View detailed dish information including description, price, and images  Filter dishes by price range or category

 View featured and recommended dishes

### Order Management :

 Add items to cart

 Adjust item quantities  Review order summary

 Place orders with delivery address  
 View order history

 Track current order status  Cancel pending orders

### User Experience :

 Responsive design for mobile and desktop  Intuitive navigation

 Real-time order status updates  Order notifications

## Admin-Side Features :

### Dashboard :

 Overview of system statistics  
 Recent orders summary

 Total earnings calculation  User registration trends

### User Management :

 View and manage user accounts  Block/unblock users

 Reset user passwords

 View user order history

### Restaurant Management :

 Add/edit/delete restaurants

 Manage restaurant categories

 Update restaurant operating hours  Upload restaurant images

### Menu Management :

 Add/edit/delete food items  Assign items to restaurants  Update pricing information  Upload food images

### Order Processing :

 View all incoming orders

 Update order status (Pending, Preparing, Out for Delivery, Delivered, Cancelled)  
 Generate order reports

 View order details



# Implementation Details :

## Frontend Implementation :

The frontend of Foodza was developed using HTML5, CSS3, and JavaScript to create a responsive and user-friendly interface. Key implementation details include:

### Responsive Design :

 Utilized CSS media queries to ensure compatibility across devices

 Implemented a mobile-first approach for better user experience on smaller screens  
 Used flexbox and grid layouts for responsive content organization

### User Interface Components :

 Navigation menu with user authentication status  Restaurant and food item cards with hover effects  Shopping cart interface with real-time updates

 Order tracking visualization

 Form validation with visual feedback

### Client-Side Validation :

 Input validation for registration and login forms  Real-time feedback for user inputs

 Form submission prevention for invalid data

## Backend Implementation :

The backend functionality was implemented using PHP 7.4 and MySQL database. Key implementation aspects include:

### Database Connectivity :

 Used PDO (PHP Data Objects) for secure database connections  Implemented prepared statements to prevent SQL injection

 Connection pooling for efficient database operations

### Authentication System :

 Secure password hashing using PHP's password\_hash() function  Session management for maintaining user login state

 Role-based access control for different user types

### Order Processing Logic :

 Validation of order details before processing

 Inventory management to check item availability  Order status tracking and update mechanisms

 Email notifications for order status changes

### Data Processing :

 Server-side validation of all user inputs  Data sanitization to prevent XSS attacks  Error handling and logging mechanisms

## Security Measures :

 CSRF token implementation for form submissions  Input sanitization and validation

 Password encryption using bcrypt

 Prevention of SQL injection using prepared statements  Session management with secure cookies

 Protection against XSS attacks  Rate limiting for login attempts



# Screenshots :

[Note: Screenshots would be included here in the actual report, showing the user interface, admin panel, order flow, etc.]



# Future Scope :

## Technical Enhancements :

### Mobile Application Development :

The current web application can be extended to native mobile applications for iOS and Android platforms, providing a more seamless experience for users on mobile devices.

### Real-time Order Tracking :

Implementing GPS-based real-time tracking for delivery personnel, allowing customers to track their order's exact location during delivery.

### Payment Gateway Integration :

Expanding payment options by integrating multiple payment gateways to support credit/debit cards, digital wallets, and other online payment methods.

### Cloud Migration :

Moving the system to a cloud-based infrastructure for better scalability, reliability, and performance optimization.

## Feature Enhancements :

### Loyalty Program :

Implementing a reward system for regular customers to encourage repeat business through points, discounts, and special offers.

### Review and Rating System :

Adding functionality for customers to rate and review restaurants and food items, helping others make informed decisions.

### Subscription-based Model :

Offering subscription plans for frequent users, providing benefits like free delivery, discounted prices, or priority service.

### Dietary Preference Filtering :

Enhancing the menu filtering system to accommodate specific dietary requirements such as vegetarian, vegan, gluten-free, etc.

## Business Expansions :

### Multi-vendor Marketplace :

Expanding the platform to allow restaurants to self-register and manage their profiles, transforming it into a complete marketplace solution.

### Analytics Dashboard :

Developing comprehensive analytics tools for restaurant owners to gain insights into their business performance, popular items, peak hours, etc.

### API Development :

Creating public APIs to allow third-party integrations with other services like social media platforms, food bloggers, or review sites.

### Internationalization :

Adding multi-language support and region-specific customizations to expand the service to international markets.



# Conclusion :

The Foodza online food ordering system successfully demonstrates the implementation of a comprehensive web application that connects customers with restaurants through a digital platform. By

leveraging HTML, CSS, JavaScript for the frontend and PHP with MySQL for the backend, the project delivers a functional and user-friendly solution for online food ordering.

Throughout the development process, emphasis was placed on creating a responsive user interface, implementing secure authentication mechanisms, establishing an efficient database structure, and developing robust order processing logic. The system not only meets the basic requirements of food ordering but also provides additional features like restaurant categorization, order tracking, and comprehensive administration tools.

The modular architecture of Foodza allows for easy maintenance and future enhancements. The identified future scope highlights potential areas for expansion, including mobile application development, additional payment options, and advanced analytics capabilities.

As e-commerce continues to grow in the food service industry, systems like Foodza will play a crucial role in connecting businesses with customers and streamlining the food ordering process. This project serves as a solid foundation for further research and development in the field of online food ordering systems.



# References :

1. PHP Official Documentation - <https://www.php.net/docs.php>
2. MySQL Documentation - <https://dev.mysql.com/doc/>
3. MDN Web Docs (HTML, CSS, JavaScript) - <https://developer.mozilla.org/>
4. Bootstrap Documentation - <https://getbootstrap.com/docs/>
5. W3C Web Standards - <https://www.w3.org/standards/>
6. E-Commerce UX Best Practices - Nielsen Norman Group
7. Food Delivery Market Research - Industry Reports
8. Web Application Security Best Practices - OWASP