**Name:** Muhammad Fareed

**PROJECT REPORT**

**Project Overview:**

This project is a dynamic Food Ordering Website developed to streamline the process of browsing, ordering, and managing food items online. The website provides functionalities like user registration, login/logout, displaying available food items, and placing orders. It's built using a combination of front-end and back-end technologies along with a relational database (MySQL) for data storage.

**Languages and Technologies Used:**

The following technologies were used in the development of this project:

* HTML, CSS, JavaScript: For creating the layout, style, and interactivity of the website.
* PHP: For server-side scripting and handling user interactions with the database.
* MySQL: For creating and managing the database.
* CSS File: Used to style headers and navigation.
* JavaScript File: Provides interactivity to user interface components.

**Working of the Project:**

The website operates by interacting with users through web forms and sessions:

* register.php: Allows new users to sign up and saves their credentials in the database.
* login.php: Authenticates existing users and starts a session.
* logout.php: Ends the user session.
* unauthorized.php: Prevents unauthorized access to restricted pages.
* index.html: Acts as the homepage for viewing food options and interacting with the system.
* main.js: Enhances user interaction and responsiveness.

**Project Enhancements:**

To make the food ordering system more scalable, user-friendly, and industry-ready, the following enhancements can be implemented in future development phases:

**1. Admin Panel for Managing Users and Menu Items**

An admin dashboard should be introduced to manage the core operations of the platform:

* View, add, edit, or delete users and food items.
* Manage order statuses (e.g., pending, in-progress, delivered).
* Track user activity and perform role-based access control.
* Enable or disable menu items based on availability.

This would require building a secure interface with restricted access and proper session handling for admin users.

**2. Search and Filtering System for Food Items**

Implement an advanced search bar and filter functionality that allows users to:

* Search food items by name or description.
* Filter by category (e.g., Fast Food, Beverages, Desserts).
* Sort by price, ratings, popularity, or availability.  
  This enhancement significantly improves the user experience and helps users quickly find what they need.

**3. Advanced CRUD Features for Menu Management**

Instead of hardcoding food items, enable full CRUD (Create, Read, Update, Delete) operations via a user-friendly interface:

* Admins can add new dishes with descriptions, prices, and images.
* Update pricing or descriptions as needed.
* Delete discontinued items.  
  This will make the menu dynamic and easy to manage without modifying backend code.

**4. Order History and Tracking for Users**

Allow logged-in users to view their past orders with details such as:

* Items ordered, quantity, and price.
* Order date and status (delivered, in-progress, cancelled).
* Estimated delivery time or tracking info.  
  This builds transparency and trust and helps users re-order with ease.

**5. Password Reset Functionality**

Implement a “Forgot Password” feature using:

* Email verification tokens or OTP.
* Secure password reset link with expiry time.
* Validation to ensure passwords meet security standards.  
  This feature adds convenience for users and boosts platform reliability.

**6. Image Uploads and Dynamic Display**

Enable food item images to be:

* Uploaded via admin panel.
* Stored in a structured directory or cloud storage (e.g., AWS S3).
* Dynamically displayed on the website with optimized size and resolution.  
  This enhances the visual appeal and helps customers make better choices.

**7. Online Payment Integration**

Integrate secure payment gateways such as Stripe, Razorpay, or PayPal:

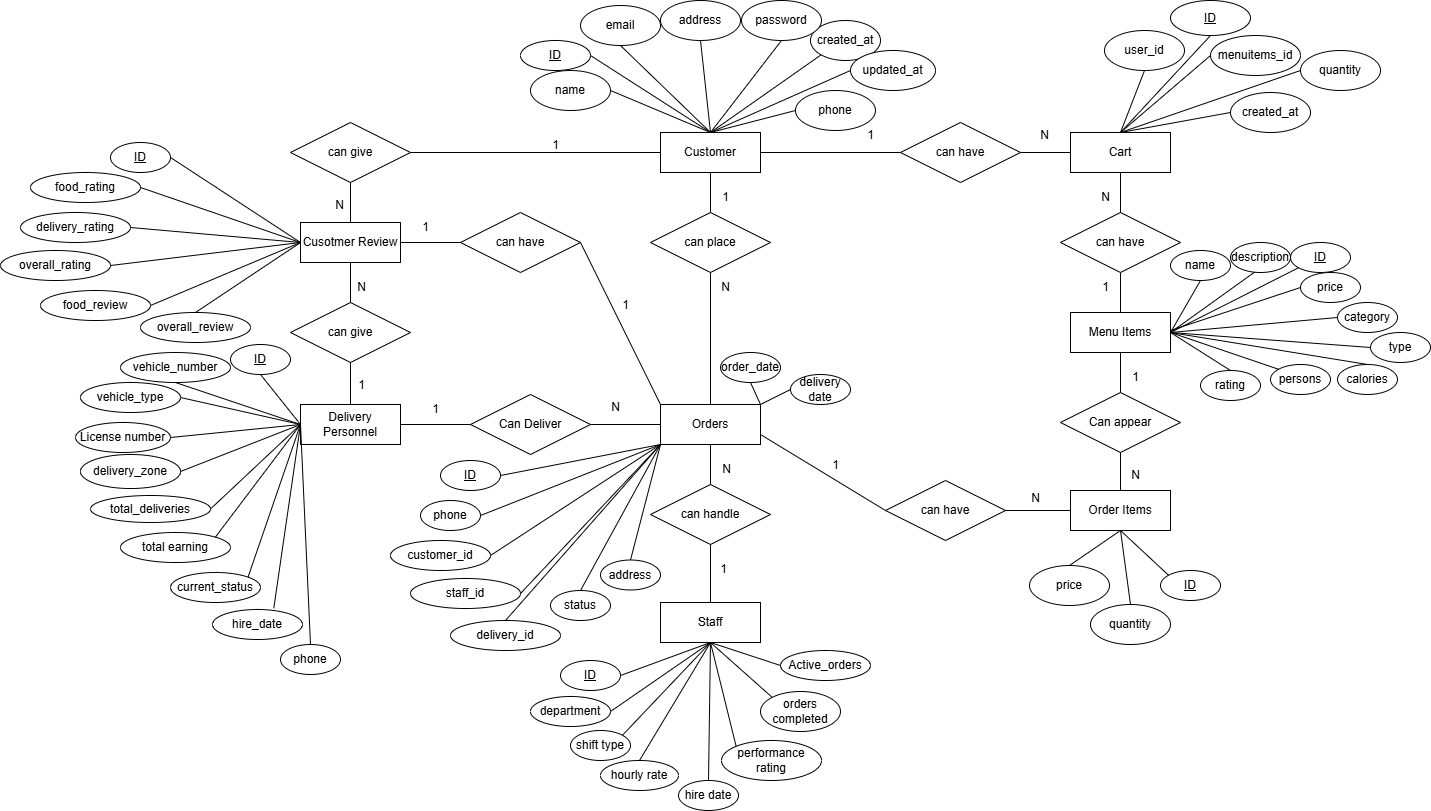
* Accept debit/credit cards, UPI, wallets, and net banking.
* Provide real-time payment verification and transaction status.
* Automatically generate digital invoices and receipts.  
  This makes the ordering process smoother and prepares the website for commercial use.

**8. Mobile Responsive Layout**

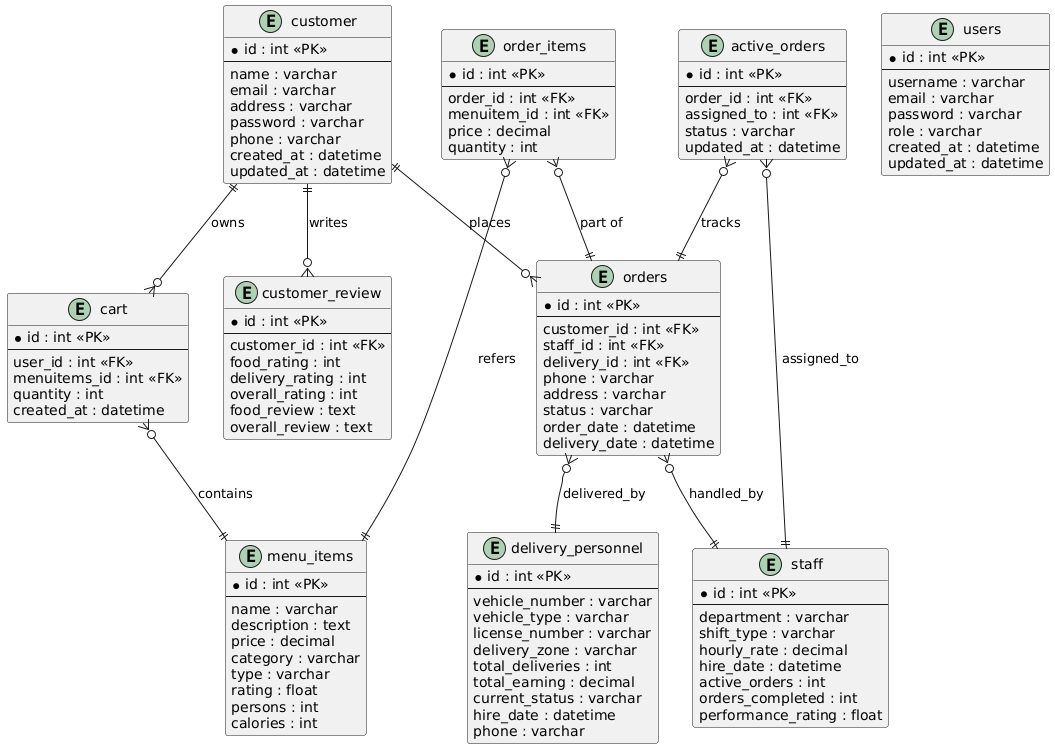
Ensure the UI is fully responsive using:

* Media queries in CSS.
* Responsive frameworks like Bootstrap or Tailwind CSS.
* Fluid layouts that adjust to phones, tablets, and desktops.  
  Given that a large percentage of users order food via mobile, this step is essential for accessibility and engagement.

**Project ERD:**

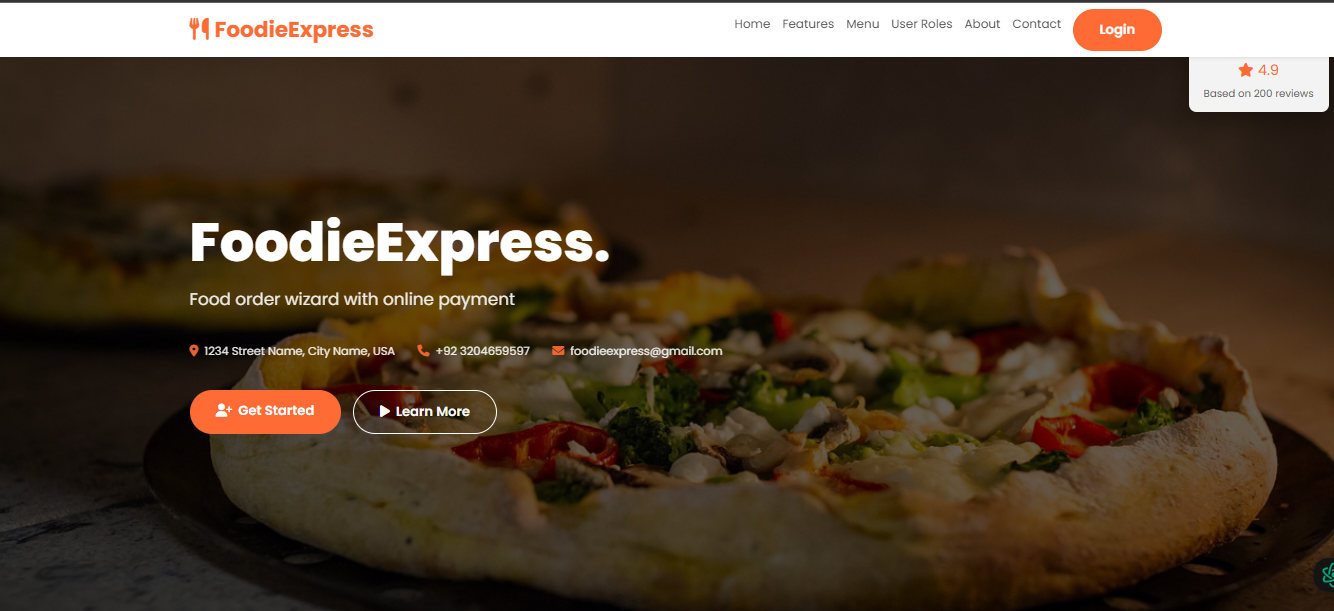
****

**Relational Schema:**

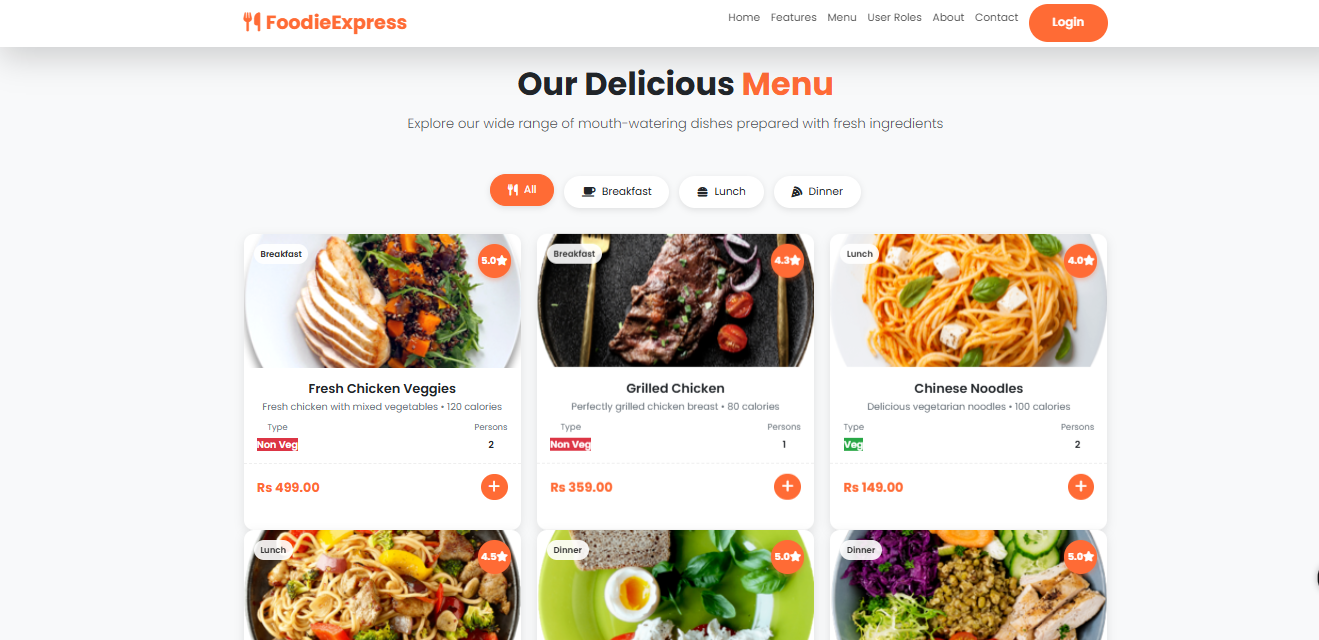


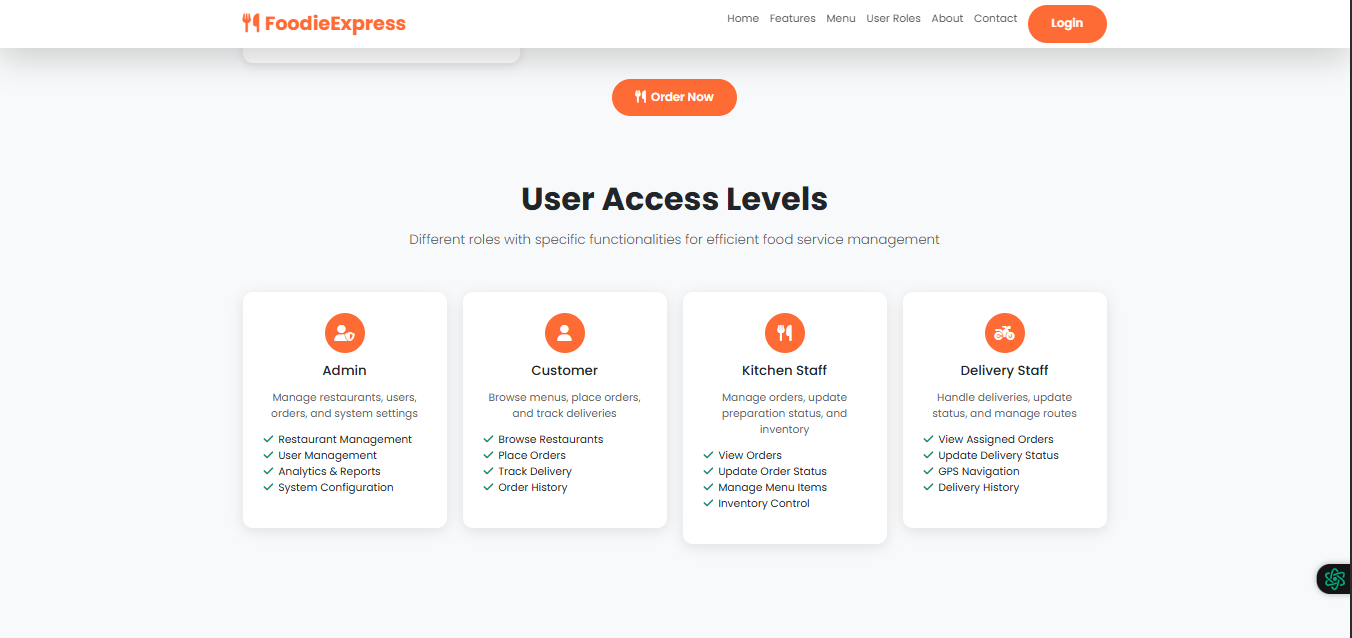
**Project Pictures:**

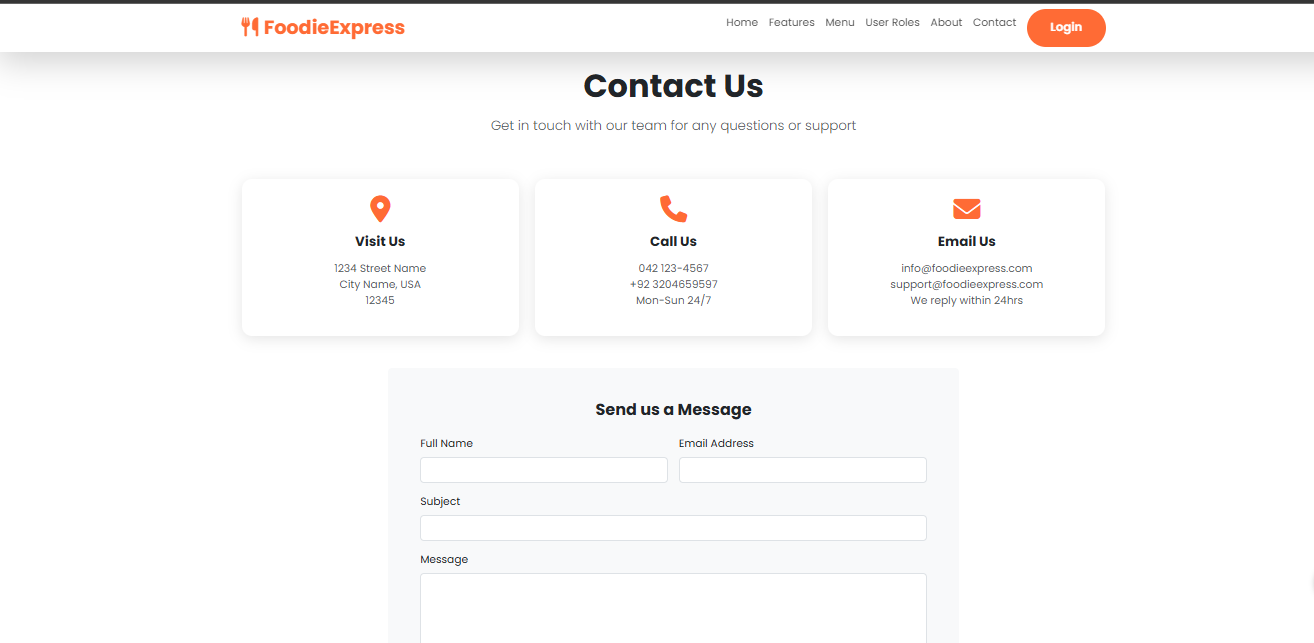
**Front Page:**

****

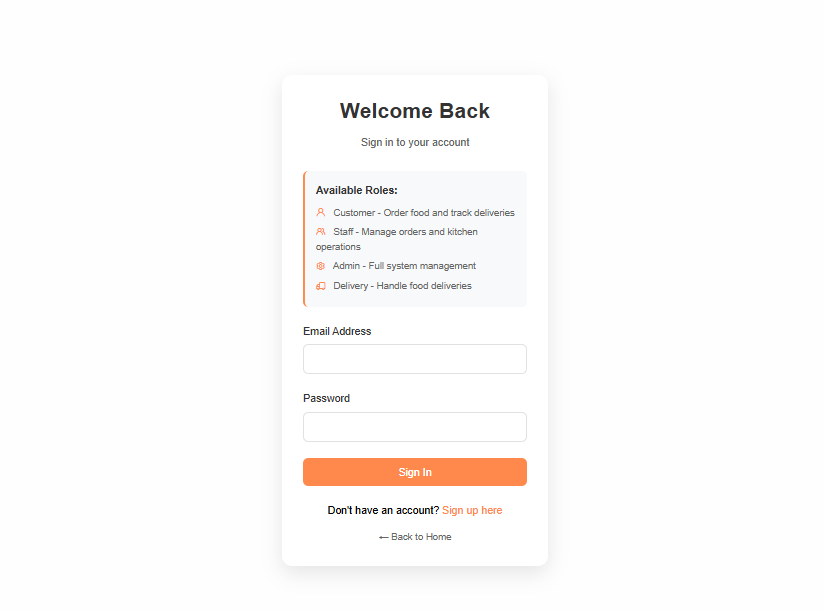
****

****

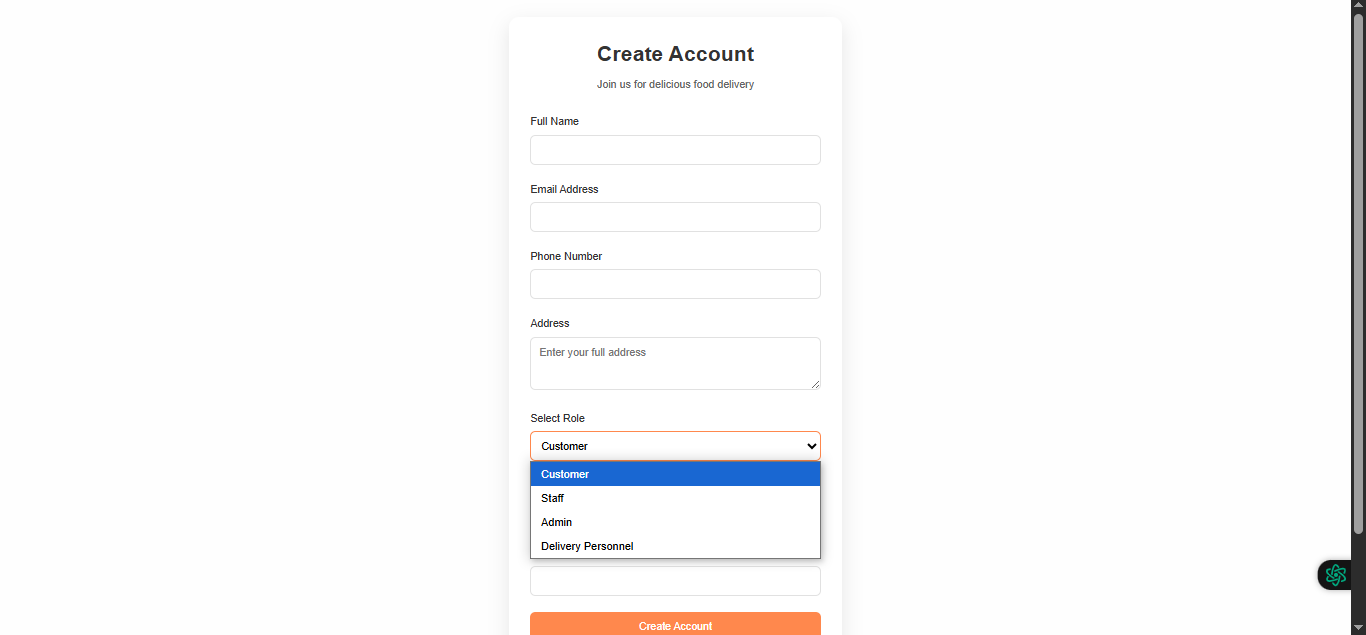
****

****

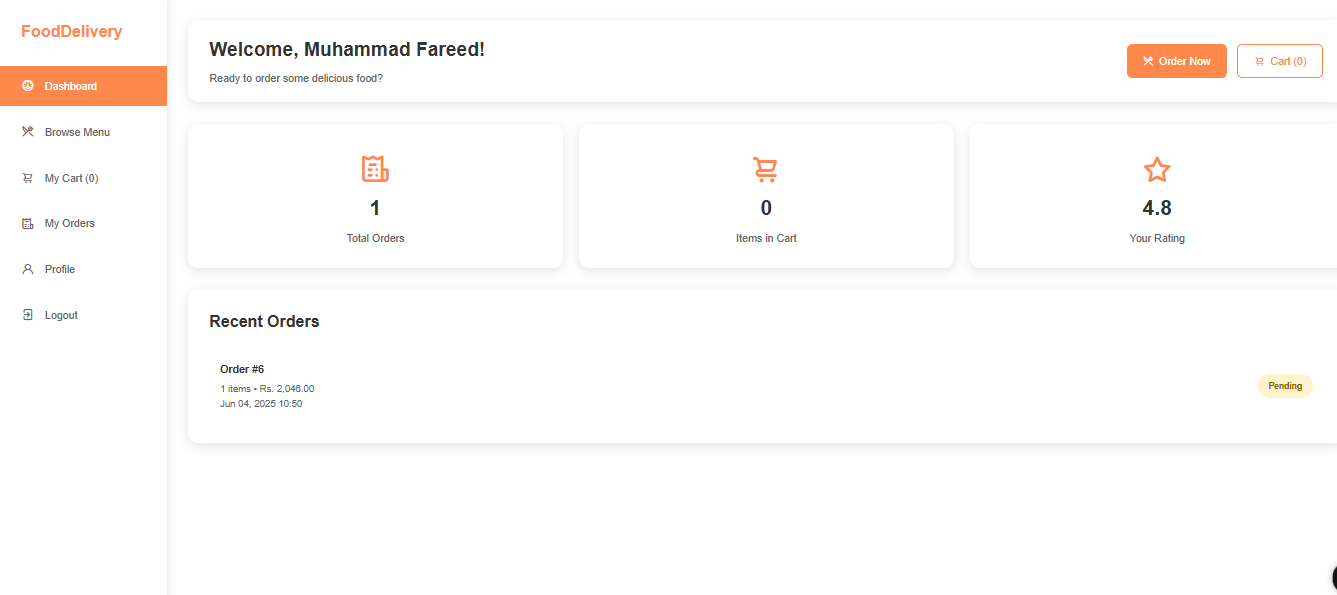
**Login Page:**

****

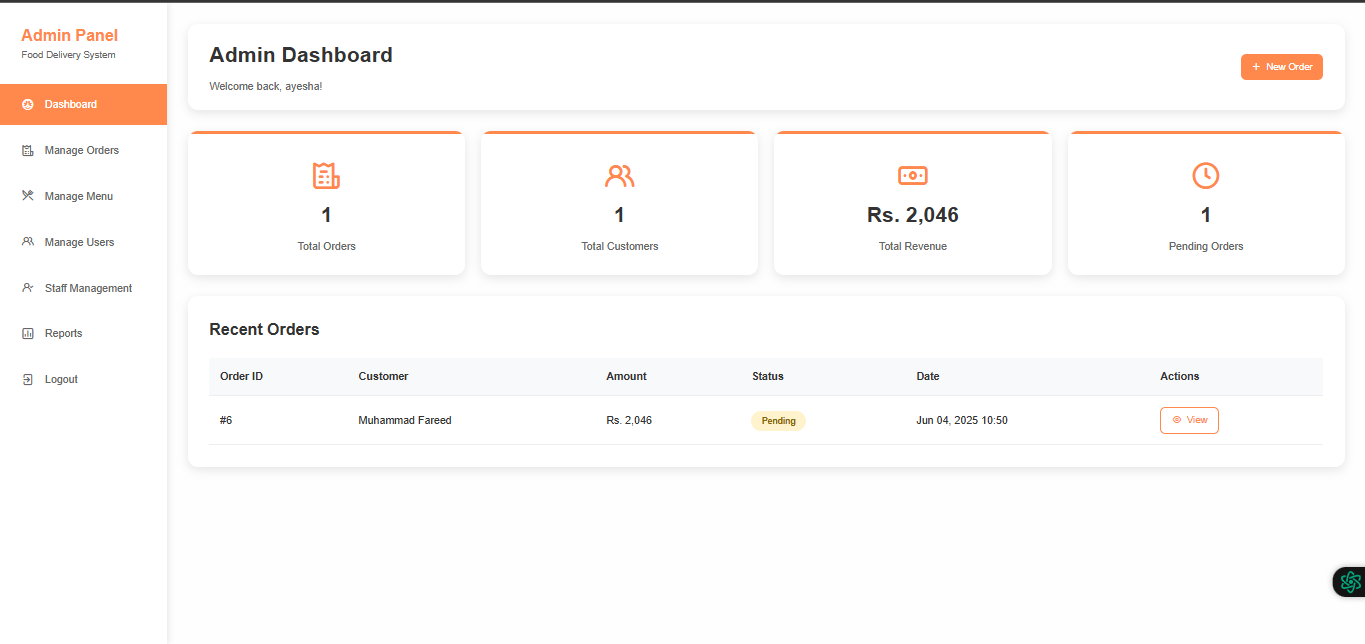
**Register Page:**

****

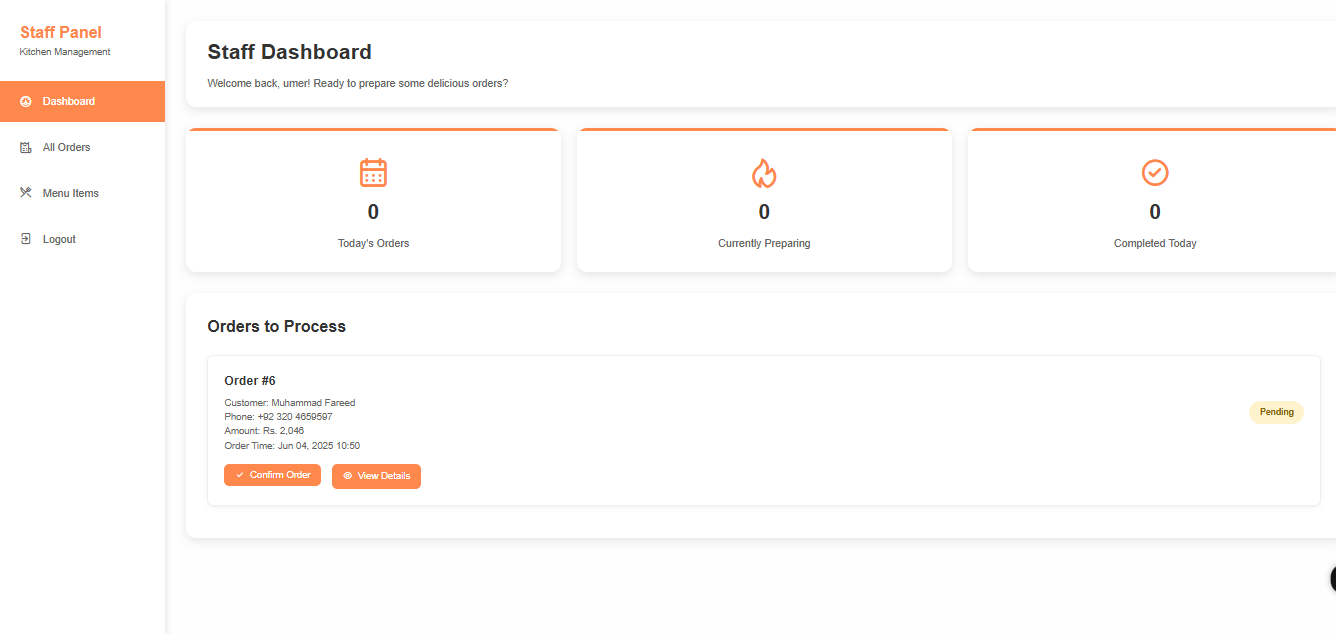
**Customer Dashboard:**

****

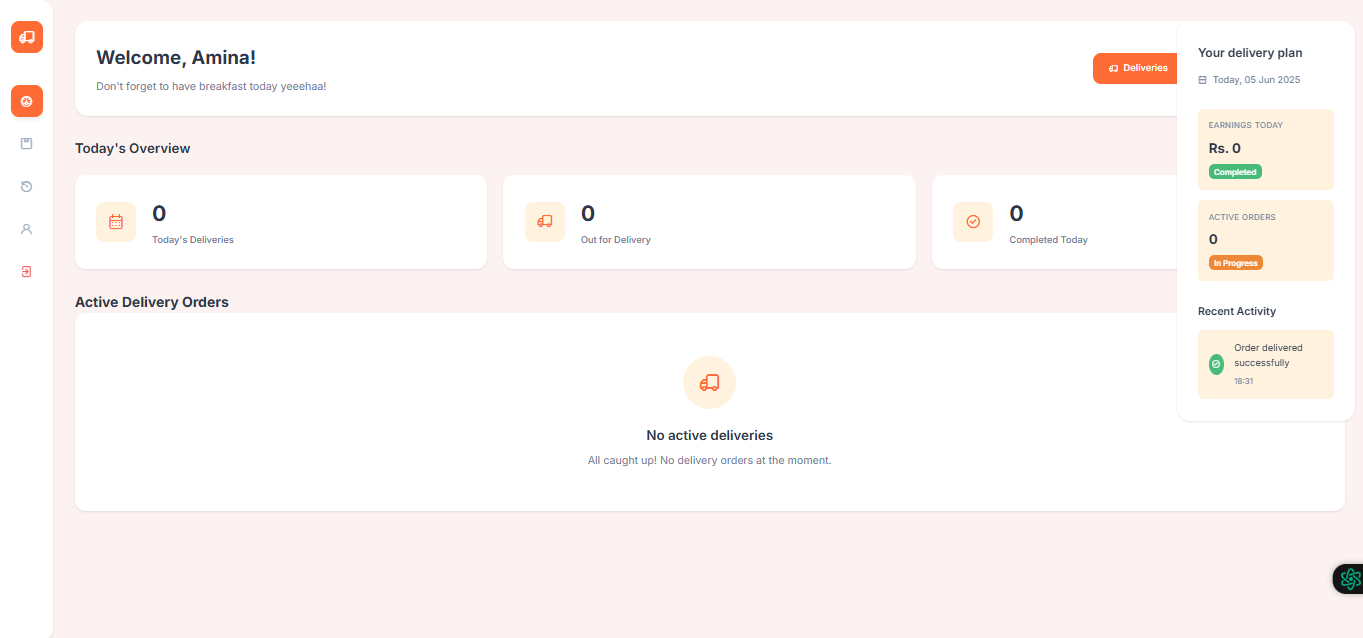
**Admin Dashboard:**

****

**Staff Dashboard:**

****

**Delivery Dashboard:**

****

**Conclusion:**

In conclusion, this project not only fulfills academic objectives but also reflects industry-level standards in database design for online food services. It lays a solid foundation for developing a fully integrated food ordering application with both frontend and backend compatibility, and it can be extended to incorporate features like real-time delivery tracking, payment gateway integration, and AI-driven recommendation systems.