Proposal for the Rendezvous Restaurant Management System

Table of Contents

1. Project Overview

The Rendezvous Restaurant Management System aims to provide a comprehensive solution for managing restaurant operations, including an enhanced menu visibility feature, a reservation system, online ordering, and efficient order and table management.

2. Project Scope

The project includes the following key features:

- **Enhanced menu visibility** with a digital, dynamic menu.
- **Reservation system** for advance bookings.
- **Online ordering and takeaway** feature.
- **Order and table management system** to optimize operations.

3. Design Phase

Understand the Project Scope

A thorough review of the project proposal will ensure the final product meets client expectations and user needs.

System Architecture & Wireframes

- **System Architecture**: The architecture consists of a frontend built with Vue.js communicating with a backend API developed using Node.js and Express, with MySQL as the database.
- **Wireframes and UI/UX Design**: Wireframes will be created for key pages, such as:
  - Menu Page
  - Reservation Form
  - Order Summary Page
  - Staff Dashboard

  The design will prioritize user-friendliness and mobile accessibility.

Technologies & Tools

- **Frontend**: Vue.js, Tailwind CSS for responsive styling.
- **Backend**: Node.js with Express, MySQL for database management.
- **Deployment Tools**: Vite for development, Docker (optional) for containerization, Heroku for deployment.

- **Security**: Basic encryption for sensitive data and secure login for staff.

4. Development Phase

Development Steps

1. **Frontend**:
    - Develop dynamic pages for the Menu, Reservation, and Order Summary.
    - Utilize Vue Router for navigation between views.
    - Implement Axios for HTTP requests to the backend API.
2. **Backend**:
    - Create RESTful API endpoints for:
        - Menu visibility
        - Order processing
        - Reservations
    - Utilize Express for routing and MySQL for data storage.
    - Use bcrypt for password hashing and express-session for user session management.
3. **Database Structure**:
    - Structure the database according to the relational schema in Appendix A, normalizing entities such as Orders, Reservations, Customers, and Staff.

Testing

- Conduct unit tests on individual components.
- Implement integration tests to ensure the entire ordering and reservation process works as intended.
- Perform security testing to protect sensitive customer data and order information.

5. Deployment Phase

Deployment Steps

- Host the application on cloud platforms like Heroku, DigitalOcean, or AWS.
- Use Git and GitHub for version control to manage code updates.
- Set up a Continuous Deployment (CD) pipeline to automate deployments from the main branch of the repository.
- Monitor system performance using tools like Google Analytics or server-side monitoring.

Post-Launch Maintenance

- Establish a plan for ongoing maintenance to address bugs and implement new features.
- Conduct a training session for restaurant staff to familiarize them with the dashboard for managing orders and reservations.