Project Kit

Title of the Project

CUISINE POPULAR AND FESTIVALS(YumYard)

Abstract of the Project

The **Cuisine popular and festivals(YumYard)** automates the process of browsing, ordering, and delivering food from various restaurants to customers. It allows customers to view restaurant menus, place orders, and track deliveries in real-time. Vendors and administrators can manage menus, process orders, and track inventory. The system integrates a payment gateway (e.g., Razorpay) for seamless transactions and includes an admin panel for order and vendor management. This project enhances operational efficiency and user experience in the food delivery industry, ensuring timely deliveries and accurate payment processing.

Generic keyword:

Food Delivery, Online Ordering, Restaurant Management, Customer Experience, Payment Integration, Delivery Tracking, Vendor Dashboard, Online Menu, Delivery Logistics, Order Management, Real-time Updates, Secure Transactions

Specific Technology keywords:

HTML, CSS, React.js, Django, REST API, Razorpay, PostgreSQL/SQLite, Payment Gateway Integration, Web Application, Database Management

Functional Components:

1. **User Registration and Authentication**
   - Allows customers and vendors to securely create accounts and log in to manage orders and food listings.
2. **Restaurant Menu Management**
   - Vendors can add, update, and remove items from their menu and manage item availability.
3. **Order Management**
   - Customers can view restaurant menus, add items to their cart, place orders, and track the delivery status in real-time.
4. **Payment Gateway Integration (Razorpay)**
   - Integrates Razorpay for processing secure payments for orders placed by customers.
5. **Order Tracking**
   - Real-time tracking of food orders from restaurant preparation to delivery.
6. **Admin Dashboard**
   - Allows administrators to manage customers, vendors, order histories, payments, and system configurations.
7. **Order History and Reports**
   - Customers and vendors can view detailed reports on past orders and earnings (for vendors).
8. **Review and Rating System**
   - Customers can rate and review vendors after receiving orders.
9. **Push Notifications**
   - Notifies customers about order status, delivery time, and promotional offers.
10. **Security and Data Protection**

- Ensures secure transactions, data encryption, and user privacy.

Functionality

- **Customers:**
  - Role: Browse restaurants, place orders, make payments, and track orders.
- **Vendors:**
  - Role: Manage menus, process orders, update availability, and monitor sales.
- **Administrators:**
  - Role: Manage users (customers/vendors), monitor transactions, and oversee system operations.

Core Functionalities:

Customer Interface:

- User-friendly navigation to browse restaurants and order food.
- Secure login system for account management.
- Ability to track order progress in real-time.

Vendor Interface:

- Management of restaurant menus, including item descriptions, pricing, and availability.
- Order processing and updating the status of the delivery.

Admin Panel:

- Oversee customer/vendor activities, monitor payments, and generate reports.
- Management of site-wide configurations, such as vendor approval and promotions.

Payment Integration:

- Secure processing of customer payments via Razorpay or any other integrated gateway.

Security Measures:

- Authentication and authorization for user and admin access.
- Encryption for sensitive data and transactions.

Steps to start-off the project:

1 **Understand the Technologies**

- Familiarize yourself with React, Django, REST API development, and payment gateway integration (Razorpay).

2 **Create User Stories and Workflow**

- Define user journeys for customers, vendors, and admins, focusing on ease of use.

3 **Design Database Schema**

- Create database models for users, orders, menu items, payments, and reviews.

4 **UI/UX Design**

- Focus on building a simple yet attractive user interface for customers and vendors.

5 **Implement Payment Integration**

- Set up Razorpay (or another gateway) to handle transactions securely.

Requirements

Hardware Requirements:

Number	Description	Alternative (if Available)
1.	Processor: x86-64 bit CPU	
2.	Ram - 4Gb, Disk Space - 5Gb.	

Software Requirements:

Numbers	Descriptions	Alternatives (if Available)
1.	Client on Intranet – User Interface, Windows OS	
2.	Development Environment – React, Django, PostgreSQL	
3	Code Editor – VS Code, Sublime Text	

Manpower requirements:

2 to 3 students can complete this in 4 – 6 months if they work fulltime on it.

Milestones and Timelines

Number	Milestonename	Milestone Description	Timeline Week no. From the start ofthe project	Remarks
1.	Requirements Specification	Define the requirements and create a specification document.	2-3	Ensure that all functionalities are covered.
2.	Technology familiarization	Understand React, Django, REST APIs, and Payment Gateway integration..	4	Demonstrate understanding with practical application..

3.	Database creation	Design and implement the database schema.	5-6	Focus on data integrity and relationships.
4.	High-Level and Detailed Design	Create flowcharts or pseudocode to map out requirements.	7-8	Make sure the design covers all features.
5.	Front-End Implementation	Implement the login system and initial restaurant menu.	9-10	Begin with wireframes for the UI
6.	Front-End and Database Integration	Connect front-end to backend for data management.	11-12	Focus on the interaction between frontend and database.
7.	Integration Testing	Thoroughly test the system, focusing on edge cases and error handling.	13-14	Fix any bugs and ensure smooth functionality.
8.	Final Review	Final demo of the complete working system.	15-16	Prepare all documents and report.

Guidelines and Reference:

- **React Official Documentation** – React Docs
- **Django Official Docs** – Django Docs
- **Razorpay Payment Gateway** – Razorpay Docs
- **Database Management Systems by Navathe** – Navathe Reference
- **Wikipedia** – Wikipedia