## Front End Engineering-II /Artificial

## Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

**Food Express**

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Mr. Rahul Samar (2210992233)

Nandini(2210991958)Ishanesh (2210991685)

Sambhav (2210992236)

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**Abstract**

Food Express is an intuitive and efficient online food ordering platform designed to streamline the process of ordering food from restaurants. With a user-friendly interface and seamless integration with multiple food vendors, the platform enhances the customer experience by providing a smooth and hassle-free ordering process.

The platform offers real-time order tracking, personalized recommendations, and secure payment options, ensuring a convenient and safe transaction experience. Utilizing modern web technologies, Food Express caters to both customers and restaurant owners by simplifying the order management system, reducing delays, and enhancing overall efficiency.

**Table of Contents**  
**1. Introduction**

* 1. Background
  2. Objectives
  3. Significance

**2. Problem Definition and Requirements**:

2.1. Problem Statement

2.2. Software Requirements

2.3. Hardware Requirements

2.4 Data Sets

**3. Proposed Design/Methodology** :

3.1. System Architecture

3.2 Core features and Functionalities

3.3 Technology Stack

3.4 Development Methodology

3.5 Security Measure

**4. Results**:

4.1. Code Snippets

4.2 Project Snippets

**5. References**:

**1. Introduction**

**1.1 Background**

The food delivery industry has seen massive growth with the increasing penetration of digital technology and the rise of convenience-based services. Traditional food ordering methods often involve long waiting times, miscommunication, and payment hassles. With advancements in web-based platforms, there is a need for a robust, scalable, and efficient food ordering system that provides a seamless experience for both consumers and restaurant owners.

Food Express aims to bridge the gap between customers and restaurants by offering an easy-to-use digital food ordering solution. It enables customers to browse restaurant menus, place orders, track deliveries in real time, and provide feedback, all in a single platform.

* **Security and Data Privacy: To implement robust security measures that protect user data and ensure a safe blogging environment.**
* **Monetization Opportunities: To offer features that allow bloggers to generate revenue through ads, subscriptions, sponsorships, and affiliate marketing.**
* **Cross-Device Accessibility: To ensure a mobile-friendly experience, allowing users to create, edit, and manage content from any device.**

**1.2 Objectives**

The primary objectives of Food Express include:

* **User-Friendly Interface:** Ensuring easy navigation and a smooth experience for all users.
* **Real-Time Order Tracking:** Keeping users informed about the order status from preparation to delivery.
* **Secure Payment System:** Providing multiple payment options such as credit/debit cards, UPI, wallets, and COD.
* **Personalized Recommendations:** Leveraging AI to suggest food items based on user preferences and order history.
* **Restaurant Management Dashboard:** Helping restaurant owners manage menus, orders, and customer feedback efficiently.
* **Location-Based Services:** Recommending nearby restaurants based on the user's location for faster delivery.
* **Customer Support:** Offering 24/7 chat and call support for order assistance.

**1.3 Significance**

Food Express aims to revolutionize the food delivery sector by:

* Enhancing convenience by reducing waiting times and eliminating communication barriers.
* Supporting local businesses by providing them a digital platform to reach a larger customer base.
* Ensuring customer satisfaction through personalized recommendations and real-time tracking.
* Encouraging digital transactions with secure payment integrations.
* Introducing an advanced feedback system for restaurants to improve their service quality.

**2. Problem Definition and Requirements**

**2.1 Problem Statement**

Existing food ordering platforms either lack efficient tracking features, have complex user interfaces, or charge high commissions from restaurants. Customers also face issues with inaccurate estimated delivery times and limited payment options. Food Express aims to create an intuitive, cost-effective, and user-centric food ordering system that simplifies the entire process while ensuring customer and vendor satisfaction.

**2.2 Software Requirements**

* **Frontend Technologies:** ReactJS, HTML5, CSS3, JavaScript
* **Backend Technologies:** Node.js, Express.js
* **Database:** MongoDB or PostgreSQL
* **Payment Integration:** Stripe, PayPal, Razorpay
* **Authentication:** JWT, OAuth 2.0
* **Cloud Storage:** AWS S3 or Firebase
* **Geolocation Services:** Google Maps API for restaurant location tracking
* **Notification Services:** Firebase Cloud Messaging for order updates

**2.3 Hardware Requirements**

* **Server Requirements:**
  + Minimum 4-core CPU (Intel Xeon or AMD Ryzen)
  + 16GB RAM
  + 500GB SSD storage
  + Load balancing for high traffic handling
* **User Device Requirements:**
  + Minimum 2GB RAM (for mobile and desktop users)
  + Latest web browser support (Chrome, Firefox, Safari, Edge)
  + Stable internet connection for seamless order placement

**2.4 Data Sets**

* **Bloom will leverage various datasets to enhance user experience and content optimization:**
* **User Data: Profile information, blog posts, comments, likes, and shares**
* **Engagement Metrics: Page views, click-through rates, session duration**
* **SEO Data: Keyword rankings, metadata analysis, traffic sources**
* **Multimedia Content: Images, videos, audio files uploaded by users**
* **Advertising and Monetization Data: Ad impressions, revenue generation statistics**

**3. Proposed Design/Methodology**

**3.1 System Architecture**

Food Express follows a three-tier architecture:

* **Frontend:** Developed using ReactJS for a fast and interactive UI.
* **Backend:** Uses Node.js and Express.js for handling API requests and business logic.
* **Database:** MongoDB for storing user profiles, restaurant details, menus, and order history.
* **Caching Mechanism:** Redis is used to speed up frequently accessed data.
* **Content Delivery Network (CDN):** Cloudflare is integrated for faster page loading and enhanced security.

**3.2 Core Features and Functionalities**

* **User Registration & Authentication:** Secure sign-up/login using JWT and OAuth.
* **Restaurant Listings & Menu Display:** Users can browse available restaurants and their menus.
* **Order Management System:** Enables customers to place, modify, and track their orders.
* **AI-Based Recommendations:** Suggests dishes based on customer behavior and preferences.
* **Secure Payment Gateway:** Multiple payment options with fraud detection.
* **Feedback & Rating System:** Customers can rate and review restaurants.
* **Push Notifications:** Real-time updates about order status.
* **Admin Panel:** Provides restaurant owners with analytics and order insights.

**3.3 Technology Stack**

* **Frontend:** ReactJS, Tailwind CSS
* **Backend:** Node.js, Express.js
* **Database:** MongoDB
* **Authentication:** JWT, OAuth
* **Cloud Storage:** AWS S3, Firebase
* **Caching:** Redis
* **Geolocation:** Google Maps API
* **Payment Integration:** Razorpay, Stripe, PayPal

**3.4 Development Methodology**

* **Agile Development:** Bi-weekly sprints with iterative improvements.
* **Version Control:** GitHub for tracking code changes.
* **Testing:** Unit testing with Jest, integration testing using Postman.
* **Continuous Deployment:** CI/CD pipelines for automatic deployments.

**3.5 Security Measures**

* **SSL/TLS Encryption:** Secure communication.
* **Role-Based Access Control (RBAC):** User-specific privileges.
* **Fraud Detection System:** AI-based anomaly detection for payments.
* **Regular Security Audits:** Monitoring vulnerabilities.

**4. Result**

**4.1 Code snippets:**

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**4.2 Project Snippets:**

**A plate of food with a fork and knife

AI-generated content may be incorrect.**

**A screenshot of a menu

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

**5. References:**

* “Run JavaScript everywhere,” Nodejs.org. [Online]. Available: https://nodejs.org/en. [Accessed: 18-Sep-2024].
* “React,” React.dev. [Online]. Available: https://react.dev/. [Accessed: 18-Sep- 2024].
* “Express - Node.js web application framework,” Expressjs.com. [Online]. Available: https://expressjs.com/. [Accessed: 18-Sep-2024].
* “Documentation - Tailwind CSS,” Tailwindcss.com. [Online]. Available: https://v2.tailwindcss.com/docs. [Accessed: 18-Sep-2024].
* “Github Docs,” github.com Available: https://docs.github.com/en