Smart Multi-Restaurant Food Ordering And Delivery System

Lalitha. C[1], B. Shabin Thanga Poocharam[2], S. Abinaya[3]
[1, 2, 3]Dept of Computer Science and Engineering
[1, 2, 3] Holycross Engineering College

Abstract- *The Smart Multi-Restaurant Food Ordering and Delivery System is an Advanced food delivery platform built to overcome the limitations of traditional single-restaurant ordering applications. The system allows users to order from multiple restaurants in a single transaction, eliminating multiple payments, separate order tracking, and redundant delivery charges. The system is developed using React.js as the frontend framework deployed on Vercel, Python Flask as the backend REST API, and PostgreSQL as the relational database. Payment processing is handled through razorpay, and location services including restaurant discovery, delivery tracking, and route optimization are powered by the Mapbox API. The platform integrates Artificial intelligence and Machine Learning technologies for food pairing recommendations using Collaborative Filtering, and a Convolutional Neural Network(CNN) based hygiene monitoring system achieving approximately 91% classification accuracy. A multi-stop route optimization module ensures efficient delivery path planning for orders spanning multiple restaurants. The system adopts a three-tier architecture comprising a React.js Presentation Layer, a Python Flask Application Layer, and a PostgreSQL Data Layer. Key features include a Smart Multi-Restaurant Cart with single-checkout, AI-powered recommendations, realtime Mapbox order tracking, and JWT-based role-based access control. The platform supports 10,000 or more concurrent users with a 99.5% uptime target.*

Keywords: Multi-Restaurant Food Ordering system, Collaborative Filtering, Convolutional Neural Network, Real-Time Delivery Tracking, Route Optimization, Three-Tier Architecture, RESTful API, JWT Authentication.

I. INTRODUCTION

The food service industry has undergone profound transformation in recent years, driven by the widespread adoption of smartphones, high-speed internet, and digital payment systems. Traditional methods of phone-based food ordering have been largely supplanted by mobile and web-based platforms that offer greater convenience, transparency, and variety. Despite the availability of several commercial food delivery applications such as Zomato, Swiggy, and UberEats, existing platforms are limited in their ability to support simultaneous multi-restaurant ordering, provide truly personalized experience, or offer advanced delivery route optimization. The research addresses these critical gaps by proposing a Smart Multi-Restaurant Food Ordering and Delivery System - a full-stack digital platform that enables customers to order from multiple restaurants in a single transaction. The system leverages artificial intelligence, real-time data processing, and cloud-based microservices to deliver a seamless experience for customers, restaurant partners, and delivery agents alike. The primary objectives of this work: (1) to design an integrated architecture that supports multi-vendor cart management, (2) to implement an AI-powered recommendation engine for personalized food suggestions, (3) to develop a real-time GPS-based delivery tracking module, (4) to build a secure and scalable payment gateway integration, and (5) to evaluate system performance under simulated load conditions. The remainder of this paper is organized as follows. Section II reviews related work and existing platforms. Section III describes the proposed system architecture and design. Section IV details the implementation methodology. Section V presents experimental results and performance evaluation. Section VI discusses conclusions and future research directions.

II. LITERATURE REVIEW

A substantial body of research has been conducted on digital food ordering systems, mobile commerce platforms, and delivery logistics optimization. This section reviews the most relevant prior work and identifies the gaps that the proposed system aims to address. A. Existing Food Ordering Platforms Moa et al. [7] provides a large-scale operational study of on-demand meal delivery platforms, releasing a dataset from a platform i n china that includes order placements, deliveries, restaurant data, driver behaviour, and weather conditions. Their findings highlight the critical role of real-time data in improving delivery performance and platform operations, identifying inconvenience and time consumption as primary user pain points when switching between separate apps - a gap the proposed system directly addresses through unified multi-restaurant ordering. B. Recommendation Systems in Food Tech Min, Jiang, and Jain[1] presented a comprehensive framework for food recommendation systems, surveying existing solutions and identifying key challenges including data sparsity, context-awareness, and health-conscious personalization. Their taxonomy covers content-based, collaborative filtering, and hybrid approaches. Building on this foundation, Sharma et al. [2] developed a three-pronged recommendation system using collaborative filtering, ingredient-based similarity, and association rule mining with the Apriori algorithm to increase both personalization accuracy and revenue per order. The proposed system adopts a similar hybrid strategy enhanced with contextual signals such as time of day user location history. Patel and Chokshi [3] conducted a systematic review of food recommender systems, examining machine learning algorithms including content-based filtering,, Collaborative filtering and hybrid models, and highlighting future research directions around deity restrictions and real-time context. Their findings inform the design of the recommendation engine in the proposed system. C. Delivery Route Optimization Tiwari and Sharma [6] proposed an optimization model for the vehicle outing problem(VRP) in last-mile delivery, comparing exact algorithms with metaheuristic approaches and demonstrating measurable improvements in delivery time and cost efficiency. Muthukalyani [9] further explored AI-powered routing and scheduling for last-mile retail delivery, showing that intelligent dispatch systems significantly reduce operational costs and improve customer experience. The proposed system's route optimization module is informed by the findings, applying dynamic routing decisions at order assignment time. D. Microservices and API Architecture Pandiya and Charankar [4] demonstrated that integrating microservices architecture with AI enables scalable, fault-tolarant real-time data processing - a design pattern directly adopted in the proposed system. Ortiz et al. [5] validated this approach in an IoT smart ports context, showing that microservice-based systems offer strong modularity and reusability. Landutama et al. [8] further confirmed the suitability of a React Native and cloud backend stack for food delivery platform integration. E. Research Gap Despite the advances documented above, no existing system combines multi-restaurant ordering, AI-based personalization, real-time tracking, and route optimization in a single unified platform accessible to small and medium-scale restaurant operators. This research fills that gap by presenting an end-to-end solution with empirical performance evaluation.

III. SYSTEM DESIGN

The proposed Smart Multi-Restaurant Food Ordering and Delivery System is architecture using a three-tier client-server model comprising a presentation layer (mobile/web client), a business logic layer (RESTful API server), and a data layer (relational and NoSQL databases). The system employs a microservices design pattern to ensure modularity, independent scalability of components, and fault isolation. A. System Architecture Overview The overall system architecture consists of eight primary modules, each responsible for a distinct functional domain. Communication between services is handled via a RESTful API gateway, while asynchronous events such as order status updates and notifications are processed through a message broker (Apache Kafka). All services are containerized using Docker and deployed on a cloud platform (AWS/GCP) with auto-scaling capabilities. B. Database Design The system employs a polyglot persistence strategy. Structured transactional data (user profiles, orders, payments) is stored in a PostgreSQL relational database ensuring ACID compliance. Product catalog and session data are managed in MongoDB (document store) fro flexible schema support. Redis is used for caching, real-time session management, and rate limiting. The entity-relationship model encompassess over 20 entities with referential integrity enforced at the database level. C. AI Recommendation Engine The recommendation engine employs a hybrid filtering approach. Content-based filtering analyzes item attributes (cuisine type, ingredients, calorie count, price range) against user preferences profile derived from order history. Collaborative filtering identifies pattern across user cohorts with similar ordering behavior. A weighted ensemble of both models is computed in real time using a lightweight scoring function deployed as a serverless function. The engine delivers recommendations with sub-100ms latency at the 95th percentile. D. Real-Time GPS Tracking Module Live delivery tracking is implemented using WebSocket connections between the delivery agent's mobile application and the tracking server. GPS coordinates are published every 5 seconds to a location topic on the Kafka broker. The React Native mobile app subscribes to order-specific topics and renders the agents's position on an interactive map using Google Maps SDK. Geofencing triggers automatic notifications to customers when the delivery agent enter a 500-meter radius of the delivery address.

IV. IMPLEMENTATION

Technology Stack

The system is implemented using a modern, open-source technology stack chosen for performance, developer ecosystem maturity, and cloud-native support:

- Frontend (Customer & Restaurant): React.js (Web), React Native (iOS/Android)
- Backend API: Node.js with Express.js framework, RESTful architecture

- Database: PostgreSQL 14 (primary)
- Authentication: Firebase
- Payment Gateway: Razorpay
- Maps & Location: Mapbox
- Deployment: Vercel
- CI/CD: GitHub Actions pipeline with automated testing and deployment.

Multi-Restaurent Cart Management

A core innovation of the proposed system is the multi-restaurant cart feature. Unlike conventional platforms that restrict ordering to a single restaurant per transaction, the proposed system maintains a unified cart object in Redis that aggregates items from multiple restaurant entities. Each cart items retains its source restaurant ID, pricing, and preparation time estimate. The checkout module consolidates all items, generates sub-orders per restaurant, and coordinates delivery logistics accordingly. The total delivery fee is computed using a dynamic pricing model that considers distance, time fo day, demand index, and order weight.

Security Implementation

Security is enforced at multiple layers. All API endpoints require JWT authentication with role-based access control (RBAC) distinguishing between customer, restaurant admin, delivery agent and system administrator roles. Input validation and parameterized queries prevent SQL injection attacks. Rate limiting is applied at the API gateway to mitigate DDoS attacks. Payment data is never stored on application servers; all sensitive card information is handled exclusively through PCI-DSS complaint third-payment payment processors. HTTPS/TLS 1.3 encryption is enforced for all client-server communications.

V. RESULTS

The system was deployed in a controlled test environment and evaluated over a 30-day period with simulated concurrent users ranging from 100 to 5,000. Performance benchmarks were conducted using Apache JMeter, and user acceptance testing(UAT) was performed with a cohort of 85 volunteer participants.

System Performance Metrices

Under a load of 1,000 concurrent users, the system maintained an average API response time of 187ms with a 99th percentile response time of 412ms. The order processing pipeline demonstrated a throughput of 340 orders per minute before horizontal scaling was triggered. The recommendation engine returned personalized results in an average of 78ms. System uptime measured 99.94% over the evaluation period.

User Acceptance Testing

UAT results revealed high satisfaction scores across key dimensions: ease of navigation (4.6/5.0), ordering experience (4.5/5.0), delivery tracking accuracy (4.7/5.0), recommendation relevance (4.2 /5.0), and payment security confidence (4.8/5.0). 91.7% of participants indicated they would prefer the proposed system over their current food ordering application. Common feedback included appreciation for the multi-restaurant cart feature and the live tracking map.

Comparison with Baseline Systems

Compared to baseline single-restaurant ordering systems, the proposed platform demonstrated a 42% reduction in average order completion time, a 35% increase in average order value attributed to multi-restaurant ordering capability, and a 28% improvement in delivery route efficiency due to the route optimization module. These results validate the effectiveness of the proposed architectural and algorithmic design decisions.

VI. CONCLUSION

This paper has presented the design, implementation, and evaluation of a Smart Multi-Restaurant Food Ordering and Delivery System. The platform address key limitations of existing food delivery solutions by integrating multi-vendor cart management, AI-powered recommendations, real-time GPS tracking, dynamic pricing, and a scalable microservices architecture into a single cohesive system. Experimental results confirm that the system achieves substantial improvements in order processing optimization, and user satisfaction compared to conventional single-restaurant platforms. The successful integration of AI-based personalization and route optimization demonstrates the viability of intelligent techniques in enhancing food delivery operations at scale. Future work will focus on four primary directions: (1) integration of deep learning-based food image recognition for enhanced menu search, (2) implementation of predictive demand forecasting to assist restaurant inventory management, (3) expansion of the recommendation engine to incorporate real-time social signals and nutritional performance evaluation.

REFERENCES

[1] W. Min, S.Jiang, and R.Jain, “Food Recommendation: Framework, Existing Solutions, and Challenges, “IEEE

Transactions on Multimedia, vol. 22, no. 10,pp. 2659-2671, Oct.2020, doi: 10.1109/TMM.2019.2958761.

[2] K. Sharma, K. V. Mandapatti, M. C. Pattekar, K. K. T. Kumar, and G. Srinivasa, "Food Recommendation System Based on Collaborative Filtering and Taste Profiling," in Proc. 3rd Int. Conf. on Intelligent Technologies (CONIT), Hubli, India, Jun. 2023, pp. 1–6. doi: 10.1109/CONIT59222.2023.10205379.

[3] Z. Patel and H. J. Chokshi, "Food Recommender System: Methods, Challenges, and Future Research Directions," International Journal of Engineering Trends and Technology (IJETT), vol. 73, no. 5, pp. 267–276, May 2025. doi: 10.14445/22315381/IJETT V73I5P123.

[4] D. K. Pandiya and N. Charankar, "Integration of Microservices and AI for Real-Time Data Processing," International Journal of Computer Engineering and Technology (IJCET), vol. 14, no. 2, pp. 240–254, 2023.

[5] G. Ortiz, A. Garcia-de-Prado, J. Boubeta-Puig, and G. Diaz, "A Microservice Architecture for Real-Time IoT Data Processing: A Reusable Web of Things Approach for Smart Ports," Computer Standards & Interfaces, vol. 81, p. 103604, Apr. 2022. doi: 10.1016/j.csi.2021.103604.

[6] K. V. Tiwari and S. K. Sharma, "An Optimization Model for Vehicle Routing Problem in Last-Mile Delivery," Expert Systems with Applications, vol. 222, p. 119789, Jul. 2023. doi: 10.1016/j.eswa.2023.119789.

[7] W. Mao, L. Ming, Y. Rong, and C. S. Tang, "On-Demand Meal Delivery Platforms: Operational Level Data and Research Opportunities," Manufacturing & Service Operations Management, vol. 24, no. 5, pp. 2535–2542, 2022. doi: 10.1287/msom.2022.1112.

[8] Landutama, R. Suryono, and D. Irfan, "Designing Platform for Food Delivery Services Integration," Journal of System and Management Sciences, vol. 13, no. 5, pp. 574–584, 2023.

[9] A. Muthukalyani, "Optimizing Last-Mile Delivery in Retail Through AI-Powered Routing and Scheduling," International Journal of Computer Engineering and Technology (IJCET), vol. 14, no. 2, pp. 194–204, 2023.

[10] OWASP Foundation, "OWASP Top Ten 2021: The Ten Most Critical Web Application Security Risks," [Online]. Available: https://owasp.org/Top10/. Accessed: Mar. 2025.