"ZORKO – AI-Based Food Delivery Web Application"

Uchit Kumar | **Satyam Panday** | **Mr. Ankit Patel (HOD)**
Uchitkumar34@gmail.com | satyampanday9651@gmail.com | gangwarankit4@gmail.com

Department of Information Technology
[DR. M.C. Saxena College of Engineering and Technology] | Lucknow

How to Cite this Article:
Panday, S. & Kumar, U. (2026). "ZORKO – AI-Based Food Delivery Web Application". International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.119

https://doi.org/ 10.55041/ijcope.v2i5.119

Abstract - This project presents ZORKO, an AI-powered online food delivery web application designed to address the growing demand for intelligent, efficient, and personalized food ordering experiences. Built on a modern technology stack—React.js for the frontend, ASP.NET Core for the backend, SQL Server for relational data storage, and Entity Framework as the ORM—ZORKO delivers a robust and scalable platform for both customers and restaurant administrators. The primary innovation of ZORKO lies in its AI-driven food recommendation engine, which employs content-based filtering to suggest dishes tailored to each user's preferences, order history, and dietary patterns. Beyond recommendations, ZORKO integrates real-time order tracking, a dedicated delivery agent module, and secure payment processing via Stripe, providing a comprehensive end-to-end food delivery solution. The administrative panel empowers restaurants with dynamic menu management, real-time order processing, and operational analytics. Security is enforced through JWT-based authentication and encrypted communication. The system demonstrates how integrating artificial intelligence into a conventional food delivery platform significantly enhances the user experience, operational efficiency, and business scalability.

Key Words: AI Food Delivery, ZORKO, Content-Based Filtering, React.js, ASP.NET Core, SQL Server, Entity Framework, Real-Time Tracking, Stripe Payment, JWT Authentication, Web Application.

1. INTRODUCTION

1.1 Background

The food delivery industry has witnessed explosive growth over the past decade, driven by increasing smartphone penetration, improved internet connectivity, and a fundamental shift in consumer behaviour toward on-demand services. Platforms such as Swiggy, Zomato, and Uber Eats have normalized the concept of food delivery, yet they largely operate as aggregators with limited personalization or intelligent assistance for end users.

ZORKO was conceptualized to bridge this gap—combining the core functionality of a food delivery platform with the intelligence of modern AI. By incorporating a content-based filtering recommendation system, ZORKO moves beyond simple menu browsing and actively curates' food suggestions for each user based on their individual taste profiles and ordering behaviour.

1.2 Problem Statement

Traditional food delivery systems, whether managed manually over phone calls or through basic online interfaces, suffer from critical limitations: lack of personalization, absence of real-time order visibility, manual error in order handling, and limited administrative control for restaurants. Even sophisticated existing platforms often fail to provide truly personalized recommendations—they rely on popularity metrics rather than individual user preferences.

There is a clear need for a web-based, AI-augmented food delivery system that not only streamlines the end-to-end ordering process but also learns from user behaviour to deliver meaningful, personalized food discovery.

1.3 Objectives

- To design and develop a full-stack AI-based food delivery web application using React.js, ASP.NET Core, SQL Server, and Entity Framework.
- To implement a content-based filtering engine for personalized food recommendations.
- To integrate real-time order tracking for customers and delivery agents.
- To build a dedicated delivery agent module for last-mile logistics management.
- To enable secure payment processing through Stripe integration.
- To provide a comprehensive administrative panel for restaurant management.
- To implement secure user authentication using JWT tokens.

1.4 Scope of the Project

ZORKO covers the following core areas: user registration and authentication, AI-powered food recommendations, restaurant and menu browsing, cart management and order placement, real-time order tracking, delivery agent management, Stripe payment gateway integration, and an administrative dashboard for restaurants. Advanced features such as voice ordering, AR-based food visualization, and multi-vendor franchise management are outside the current scope but are identified as directions for future enhancement.

1.5 Methodology

The project follows an Agile development methodology with iterative sprints. A modular, component-based frontend architecture in React.js is paired with a RESTful API backend in ASP.NET Core. Entity Framework Core serves as the ORM layer for structured data access against SQL Server. The AI recommendation engine is built as a dedicated service module. GitHub is used for version control and collaborative development, with CI/CD practices ensuring continuous integration of new features.

1.6 Organization of the Report

- Chapter 1: Introduction – Overview, objectives, and scope of the project.
- Chapter 2: Literature Review – Existing systems, AI in food delivery, and related technologies.
- Chapter 3: System Analysis and Design – Requirements, architecture, and design diagrams.
- Chapter 4: Implementation – Development process, tools used, and key implementation details.
- Chapter 5: Testing and Evaluation – Testing strategy, results, and performance analysis.
- Chapter 6: Conclusions and Future Work.

2. LITERATURE REVIEW

Singh (2023) presented a foundational online food ordering system that highlighted the core requirements of menu browsing, cart management, and order placement—forming a baseline against which AI-augmented systems such as ZORKO can be compared. Kumar and Devi (2020) demonstrated the importance of real-time tracking in improving customer satisfaction metrics, a finding that directly motivated the inclusion of live order tracking in ZORKO.

Prasad and Murthy (2021) investigated recommendation systems in food ordering platforms and showed that content-based filtering achieves high precision in personalized dish suggestions when user preference vectors are constructed from order history and item attributes. Their methodology forms the theoretical basis for ZORKO's AI recommendation engine. Kaur and Bhatia (2022) further explored AI applications in food delivery, particularly route optimization and demand prediction, underscoring the broad potential of machine intelligence in this domain.

Mishra and Chauhan (2018) addressed security challenges in payment gateway integration, establishing best practices that informed ZORKO's Stripe implementation. Ali and Han (2024) emphasized data privacy and regulatory compliance in food ordering systems, reinforcing the need for encrypted communication and secure session management—both addressed in ZORKO via HTTPS and JWT authentication.

Das and Rao (2022) established UX design principles for food ordering websites, advocating for minimal navigation depth and clear visual hierarchies, principles reflected in ZORKO's React.js frontend design. The reviewed literature collectively confirms the relevance of combining a robust CRUD-based food delivery core with an AI personalization layer, as realized in ZORKO.

---

3. SYSTEM ANALYSIS AND DESIGN

3.1 Requirement Analysis

3.1.1 Functional Requirements

- User Registration/Login: Customers, administrators, and delivery agents can register and authenticate securely via JWT.
- AI Food Recommendations: The system generates personalized dish suggestions using content-based filtering on user order history and item attributes.
- Restaurant and Menu Browsing: Users can explore restaurant listings and detailed menus with category filters.
- Cart Management: Users can add, remove, and modify item quantities before checkout.
- Order Placement and Payment: Orders are placed and processed through Stripe's payment gateway.
- Real-Time Order Tracking: Customers receive live status updates from order confirmation to delivery.
- Delivery Agent Module: Agents can view, accept, and update assigned deliveries in real time.
- Admin Panel: Administrators manage restaurants, menus, pricing, availability, and view order analytics.

3.1.2 Non-Functional Requirements

- Responsiveness: The UI must function seamlessly on mobile, tablet, and desktop viewports.
- Security: JWT tokens for authentication; bcrypt for password hashing; Stripe for PCI-compliant payments.
- Scalability: The SQL Server schema and ASP.NET Core service layer are designed for horizontal scaling.
- Performance: Optimized EF Core queries, asynchronous API endpoints, and React lazy loading ensure low latency.
- Reliability: The system maintains order integrity through transaction management in SQL Server.

3.2 System Architecture

ZORKO follows a Three-Tier Architecture comprising a React.js client layer, an ASP.NET Core API layer, and a SQL Server database layer. An additional AI Service module operates as a logical component within the backend, handling recommendation computation. The frontend communicates with the backend exclusively through RESTful API calls over HTTPS. Entity Framework Core manages all database interactions through strongly-typed models.

Layer	Technology	Role
Client Tier	React.js + Tailwind CSS	UI rendering, state management, routing
API Tier	ASP.NET Core (C#)	Business logic, REST API, authentication
AI Module	Content-Based Filtering Service	Personalized food recommendations
Data Tier	SQL Server + Entity Framework	Relational data storage and ORM
Payment	Stripe API	Secure online payment processing
Auth	JWT (JSON Web Tokens)	Stateless session management

Table 1: System Architecture Layers

---

4. IMPLEMENTATION

4.1 Tools and Technologies

Technology	Purpose
React.js	Frontend library for building an interactive, component-based UI
ASP.NET Core	Backend web framework for building RESTful APIs in C#
SQL Server	Relational database for persistent storage of all application data
Entity Framework Core	ORM for type-safe database queries and migrations
JWT (jsonwebtoken)	Stateless user authentication and session handling
Stripe API	PCI-compliant payment gateway integration
Axios	HTTP client for frontend-to-backend API communication
React Router	Client-side navigation and route management
TailwindCSS	Utility-first CSS framework for responsive UI styling
SignalR (ASP.NET)	Real-time WebSocket communication for live order tracking
Content-Based Filtering	AI recommendation engine based on user preference vectors

Table 2: Tools and Technologies Used

4.2 AI Recommendation Engine

ZORKO's recommendation engine employs content-based filtering to generate personalized food suggestions. Each menu item is represented as a feature vector comprising attributes such as cuisine type, flavor profile (spicy, sweet, savory), dietary category (vegetarian, vegan, non-vegetarian), price range, and average rating. A corresponding user preference vector is dynamically constructed and updated from the user's order history, explicit ratings, and browsing patterns.

Cosine similarity is computed between the user preference vector and the item feature vectors to rank dishes. The top-N highest-similarity items not recently ordered are surfaced as recommendations. The engine is

implemented as a dedicated service class within the ASP.NET Core backend, with preference vectors persisted in SQL Server and recomputed incrementally on each new order event. This approach ensures recommendations improve organically as users interact more with the platform.

4.3 Frontend Implementation

The frontend is developed in React.js using a component-based architecture. Pages include a Home/Hero page, Restaurant Listing, Menu Browse, Cart, Checkout, Order Tracking, User Profile, and the Admin Dashboard. React Router provides smooth client-side navigation without full page reloads. TailwindCSS enables a fully responsive layout compatible with all screen sizes. Axios handles all API calls with interceptors for JWT token attachment on authenticated routes. The AI recommendation widget appears on the home screen and dish listing pages, updating in real time as the user's session progresses.

4.4 Backend Implementation

The backend is built with ASP.NET Core and follows RESTful API design principles. Controllers are organized by domain: AuthController (registration, login, JWT issuance), Restaurant Controller (listings, details), MenuController (item CRUD), OrderController (placement, status updates), RecommendationController (AI suggestions), PaymentController (Stripe session creation and webhook handling), and DeliveryController (agent assignments and status). Entity Framework Core manages schema migrations and all database interactions through a strongly-typed DbContext. SignalR hubs enable real-time push notifications to customers and delivery agents on order status changes.

4.5 Payment Integration

Stripe is integrated for payment processing. Upon order confirmation, the frontend initiates a Stripe Checkout Session via the backend PaymentController. The user is redirected to Stripe's hosted payment page, ensuring that sensitive card data never passes through ZORKO's servers—achieving PCI-DSS compliance. Stripe webhooks notify the backend of successful payments, triggering order confirmation and initiating the delivery pipeline. Refund management is also handled through the Stripe API.

4.6 Delivery Agent Module

A dedicated interface for delivery agents allows them to log in, view assigned orders, update pickup and delivery statuses, and receive new order assignments in real time via SignalR. The admin panel includes a dispatch view for assigning agents to orders based on proximity and availability. Order status transitions (Confirmed → Preparing → Out for Delivery → Delivered) are synchronized across all stakeholders—customer, restaurant, and agent—in real time.

TESTING AND EVALUATION

4.7 Testing Methodology

Test Type	Tools Used	Purpose
Unit Testing	xUnit (.NET), Jest (React)	Validate individual components, services, and controller methods
Integration Testing	Postman, Swagger UI	Verify API route connectivity, request/response contracts, and JWT validation
UI Testing	Manual, Chrome DevTools	Ensure responsive layout and correct user interactions across devices
Payment Testing	Stripe Test Mode	Simulate successful, failed, and refunded payment scenarios
Real-Time Testing	Browser WebSocket Inspector	Validate live order status updates via SignalR
AI Recommendation Testing	Custom test dataset	Measure recommendation relevance using precision and recall metrics
Authentication Testing	Postman, Manual	Verify JWT issuance, expiry, refresh, and role-based access control

Table 3: Testing Methodology

4.8 Test Results Summary

All critical functional requirements were validated successfully. Unit tests for the recommendation engine achieved a precision score of 0.82 and recall of 0.76 on a held-out test dataset of 500 user-order pairs, confirming meaningful personalization. API integration tests confirmed correct status codes, response schemas, and error handling across all 24 API endpoints. Payment simulation tests covered successful transactions, card decline scenarios, and webhook delivery, all handled correctly. Real-time tracking updates consistently delivered within 300ms under simulated load. Authentication flows, including token expiry and role-based access enforcement, passed all boundary test cases.

---

5. CONCLUSIONS

ZORKO successfully demonstrates the integration of artificial intelligence into a full-stack food delivery web application. Built on React.js, ASP.NET Core, SQL Server, and Entity Framework Core, the platform delivers a comprehensive ordering experience enriched by a content-based filtering recommendation engine. The inclusion of real-time order tracking via SignalR, a dedicated delivery agent module, and secure Stripe payment processing positions ZORKO as a production-ready, end-to-end food delivery solution.

The project validates that content-based filtering is an effective approach for food recommendation in a cold-start-mitigated environment, achieving strong precision and recall metrics. The modular architecture ensures that each component—AI, payments, real-time communication, and data management—can be independently scaled or enhanced. ZORKO thus provides a strong foundation for future extensions including demand forecasting, route optimization for delivery agents, multi-language support, and a native mobile application.

ACKNOWLEDGEMENT

The authors express sincere gratitude to the faculty of the Department of Computer Science and Engineering for their guidance and continuous support throughout this project. We acknowledge the contributions of the open-source communities behind React.js, ASP.NET Core, Entity Framework, and TailwindCSS, whose tools and documentation were instrumental in the development of ZORKO. We also thank Stripe for their comprehensive developer resources and test environment, and Microsoft for SQL Server and the .NET ecosystem.

REFERENCES

1. A. B. Singh, "Online Food Ordering System," B.Tech. project report, Dept. Comput. Sci. Eng., Lovely Professional University, Phagwara, India, 2023.
2. J. S. Kumar and R. S. Devi, "Design and Implementation of an Online Food Ordering System with Real-Time Tracking," Int. J. Comput. Appl., vol. 178, no. 1, pp. 25–30, 2020.
3. P. G. Sharma and D. V. Reddy, "A Secure and Scalable Architecture for Web-Based Food Delivery Platforms," in Proc. Int. Conf. Adv. Comput. Commun. Technol., 2021, pp. 112–117.
4. V. N. Prasad and S. R. Murthy, "Implementing Recommendation Systems in Online Food Ordering Platforms," Expert Syst. Appl., vol. 165, p. 113945, 2021.
5. H. Kaur and P. S. Bhatia, "Artificial Intelligence in Food Delivery: Route Optimization and Demand Prediction," J. Artif. Intell. Rev., vol. 55, no. 1, pp. 1–20, 2022.
6. R. P. Mishra and S. Chauhan, "Payment Gateway Integration and Security Challenges in Online Food Ordering Systems," in Proc. National Conf. Inf. Technol., 2018, pp. 45–50.
7. L. M. Das and K. R. Rao, "User Experience (UX) Design Principles for Online Food Ordering Websites," J. Web Eng., vol. 21, no. 3, pp. 345–360, 2022.
8. F. B. Ali and C. M. Han, "Ensuring Data Privacy and Compliance in Online Food Ordering Systems," Int. J. Inf. Secur., vol. 23, no. 4, pp. 567–580, 2024.
9. S. K. Singh and A. K. Dubey, "Blockchain Technology for Enhancing Transparency and Trust in Food Supply Chains," Food Control, vol. 131, p. 108390, 2022.
10. G. V. Ramana and P. S. Kumar, "Customer Relationship Management (CRM) in Online Food Ordering Businesses," J. Mark. Technol., vol. 10, no. 1, pp. 78–85, 2021.