Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

IntelliPick: AI-Enabled Smart Restaurant Ordering and Pickup Platform

Sagarika Saka[1], Darapureddy Murali Mohan[2], Dasari Navaneetha[3], Chintakuntla Lohith Rama[4], Chakinala Anvesh[5]

[1]Assistant Professor, Department of Computer Science and Engineering
Malla Reddy University, Hyderabad, Telangana, India
[2,3,4,5] Department of Computer Science and Engineering,
Malla Reddy University, Hyderabad, Telangana, India

Abstract

Online food ordering has moved from a convenience to a daily habit for many urban consumers. Yet, most commercial platforms are designed around delivery and third-party logistics, which drives up service charges, introduces additional latency, and places a commission burden on restaurants. In this work, we present IntelliPick, an AI-enabled smart restaurant ordering platform that emphasises advance ordering and secure in-person pickup. Customers can discover nearby outlets, examine menu items with images, place orders at standard in-store prices, and collect them using a one-time password (OTP) based verification procedure. The system is built on a three-tier web architecture that combines a React front end, a Spring Boot middle tier, and a MySQL database, and is augmented with components for dish recommendation, sentiment analysis of reviews, and distance-based restaurant ranking using the Haversine formula. A prototype implementation and experimental study show that IntelliPick can handle core operations within a few seconds of response time while maintaining accurate proximity ranking and useful recommendation behaviour. The results suggest that a pickup-centric, AI-assisted model can offer a cost-effective and operationally simpler alternative to heavily delivery-driven food platforms for both customers and restaurants.

Index Terms: Online food ordering, pickup systems, restaurant discovery, recommendation systems, sentiment analysis, Haversine distance, web applications, JWT authentication.

1. Introduction

Digital channels have become the primary touchpoint between diners and restaurants. Mobile applications and browser-based interfaces now handle tasks that were once manual, such as searching for nearby outlets, browsing menus, placing orders, and providing feedback. This transition has created an ecosystem dominated by large delivery platforms that coordinate customers, restaurants, and delivery partners.

The widely adopted delivery-focused model introduces several practical challenges. Customers often pay a combination of delivery fees, service charges, and context-dependent surcharges, which means the final

bill can differ substantially from the in-store price. At the same time, delivery times are at the mercy of traffic, rider availability, and weather, all of which contribute to variability in end-to-end latency.

Restaurants face a different set of constraints. Many platforms operate on percentage-based commission models, so a share of each order is redirected away from the restaurant. Over time, this can erode margins and leave restaurants economically and operationally dependent on the platform for visibility and order flow. These observations have motivated numerous works on more intelligent recommendation and feedback analysis to at least improve matching between users and restaurants within the existing paradigm [1].

An alternative approach is to retain the convenience of online ordering while removing the delivery leg altogether. In a pickup-oriented system, customers place orders ahead of time, travel to the restaurant at a suitable moment, and collect their meals at the counter. Eliminating the courier layer immediately reduces fees and simplifies logistics. Properly designed digital workflows, combined with AI-based components such as recommendation and sentiment analysis, can make this experience competitive with or even superior to conventional delivery-heavy applications [3].

This paper introduces IntelliPick, a smart restaurant ordering platform centred on secure and efficient pickup. IntelliPick provides:

- a responsive, web-based interface for customers to discover nearby restaurants and navigate image-rich menus;
- a dashboard for restaurant owners to manage menus, item images, availability, and order processing;
- administrative tools to control restaurant onboarding, category structures, and high-level platform monitoring; and
- intelligent modules for dish recommendation, sentiment analysis of textual reviews, and distance-aware restaurant visibility.

The system is implemented using a full-stack architecture based on React.js, Spring Boot, and MySQL, which is a combination that has been widely adopted for online ordering projects in practice [16], [17]. Location-based discovery relies on the Haversine formula for computing distances between customer and restaurant coordinates [7]. The recommendation and sentiment components are inspired by recent work on food-delivery and restaurant review analysis that blends machine learning with domain-specific signals [2], [8].

The remainder of the paper is organised as follows. Section II reviews related efforts on food-ordering platforms, recommendation techniques, and sentiment analysis in the restaurant context. Section III describes the IntelliPick system model and architecture. Section IV outlines the core algorithms used in the recommendation and distance modules. Section V explains key implementation aspects and user workflows for customers, restaurant owners, and administrators. Section VI discusses evaluation metrics and the experimental setup. Section VII presents and interprets the results, including a comparison with conventional delivery-focused applications. Section VIII concludes with a summary and directions for future work.

2. Related Work

A. Sentiment Analysis for Food Delivery and Restaurants

There is a growing body of work that studies sentiment analysis in the context of food delivery and restaurant reviews. Hussain et al. compiled a systematic review of sentiment analysis approaches for customer reviews on food-delivery platforms, highlighting the diversity of datasets, model architectures, and evaluation criteria used in the field [1]. Their survey emphasises that deep learning models can achieve strong performance but also argues that explainability remains an important concern for practitioners.

A number of systems explicitly combine sentiment analysis with restaurant recommendation. Alagha and Samson proposed a recommendation framework in which sentiment scores extracted from textual reviews are used alongside ratings to refine prediction accuracy for restaurant choices [2]. Their results show that integrating review text can help overcome some of the limitations of rating-only collaborative filtering. Ghorbani et al. investigated deep hierarchical architectures tailored to restaurant reviews from food applications, reporting competitive sentiment classification performance on real-world data [8]. These works collectively suggest that sentiment-aware analysis can provide additional signals for understanding customer satisfaction and guiding recommendations in food-related platforms.

B. Recommendation in Food-Delivery and Marketplace Platforms

Beyond pure sentiment analysis, several studies focus on recommendation mechanisms for online food ordering. Arifin et al. proposed using the number of orders as a central signal in a delivery application, demonstrating that simple frequency-based statistics can support effective recommendation policies in a production setting [3]. Their work reflects the practical value of order data and aligns with the frequency-based heuristics adopted in IntelliPick.

Industrial case studies offer another window into how large-scale food-delivery platforms design their recommendation pipelines. Uber has publicly described the recommendation architecture for Uber Eats, where machine learning models learn to rank restaurants and dishes given user and marketplace context [5]. In related work, a follow-up technical discussion provides more detail on the use of graph-based and multi-objective methods to balance personalisation with marketplace health and fairness [13]. Similarly, a report on iFood's platform discusses the creation of dish and restaurant collections that are personalised based on user behaviour and contextual features [6]. These large-scale deployments underline the importance of recommendation for food discovery, even though they remain primarily delivery-oriented.

C. Pickup-Oriented and Direct-Ordering Solutions

While many well-known services are built around delivery, a range of systems explicitly support pickup or order-ahead use cases. Some commercial platforms provide commission-free online ordering portals where customers can schedule pickup orders directly with restaurants, which reduces reliance on third-party marketplaces and associated fees [14]. Other providers offer self-pickup modules that allow restaurants to accept and manage click-and-collect orders through dedicated dashboards [15]. These solutions show that there is demand for digital tooling that focuses on pickup and direct interaction between restaurants and customers.

D. Web Architectures for Food-Ordering Applications

From an implementation standpoint, many academic and practical projects have converged on a similar technology stack for online ordering. Murad documented a multi-restaurant food-ordering system based on a Spring Boot backend, React front end, and MySQL database, combined with JWT authentication and role-based access control [16]. Iluma presented a full-stack example that integrates Spring Boot and React with payment gateways and cloud deployment, again demonstrating the suitability of this stack for interactive ordering workloads [17]. Open-source repositories for food-delivery web applications often follow comparable patterns, confirming the popularity of this architecture [18], [19].

E. Geospatial Distance Computation

Accurate yet efficient distance calculation is an essential ingredient for location-based restaurant discovery. Sinnott discussed the advantages of the Haversine formula in computing great-circle distances on a sphere, especially for applications where numerical stability at small distances is important [7]. The Haversine formula is widely applied in location-based systems to compute the distance between user and restaurant coordinates.

3. System Model and Architecture

A. Problem Statement

Most existing online food-ordering platforms share a similar operational pattern: the platform exposes restaurants and menus to customers, collects orders, and then orchestrates delivery through a network of drivers or riders. Although this model delivers convenience, it raises a number of recurring issues.

First, customers face several additional charges over and above the base price of food. Delivery fees, service charges, and occasionally dynamic surcharges increase the effective cost of ordering. Second, delivery introduces new sources of delay and uncertainty. Travel times depend on road conditions and driver availability, making arrival times difficult to predict in busy or adverse conditions. Third, the commission structure applied by many platforms means that a portion of each order's value is diverted from the restaurant, which can be particularly challenging for smaller businesses.

In addition, intelligence in these systems is often used primarily to optimise conversion or marketplace metrics rather than to expose detailed feedback to restaurant owners. While there is clear evidence that recommendation and sentiment analysis can be used to learn from customer behaviour [2], such insights are not always made directly accessible to restaurants. As a result, owners may not fully benefit from the data their customers generate.

These limitations suggest that a complementary model centred on pickup-based ordering could address both cost and control concerns. In this model, customers still place orders online but collect them in person. If the digital platform is designed to clearly indicate preparation and pickup times, and if it offers intelligent tools for discovery and feedback analysis, it can provide a streamlined experience without the complexity and overhead of managing deliveries.

B. Overview of IntelliPick

IntelliPick is a web-based platform that operationalises this pickup-centric view. It supports three principal user roles:

- Customer: registers or logs in, discovers restaurants nearby, browses menus with images, receives recommendations, places orders, and monitors status until pickup.
- Restaurant owner: manages restaurant information, defines and maintains menu items with images, sets availability, and processes incoming orders.
- Administrator: vets and approves restaurant registrations, maintains category structures, and oversees platform activity at a high level.

Beyond basic ordering, IntelliPick integrates AI-based modules that analyse order histories and textual reviews. The recommendation module highlights dishes that are popular or otherwise likely to be of interest to a given customer, following the intuition demonstrated by frequency-based and hybrid methods in prior work [3]. The sentiment analysis component processes customer feedback to derive polarity scores, which are then surfaced to restaurant owners through dashboards, in line with findings from restaurant-review studies [8]. Location-based discovery ranks restaurants by distance from the customer, guided by Haversine-based distance estimates [7].

C. Three-Tier Architecture

IntelliPick adopts a three-tier architecture that separates presentation, application logic, and data management, as illustrated in Figure 1.

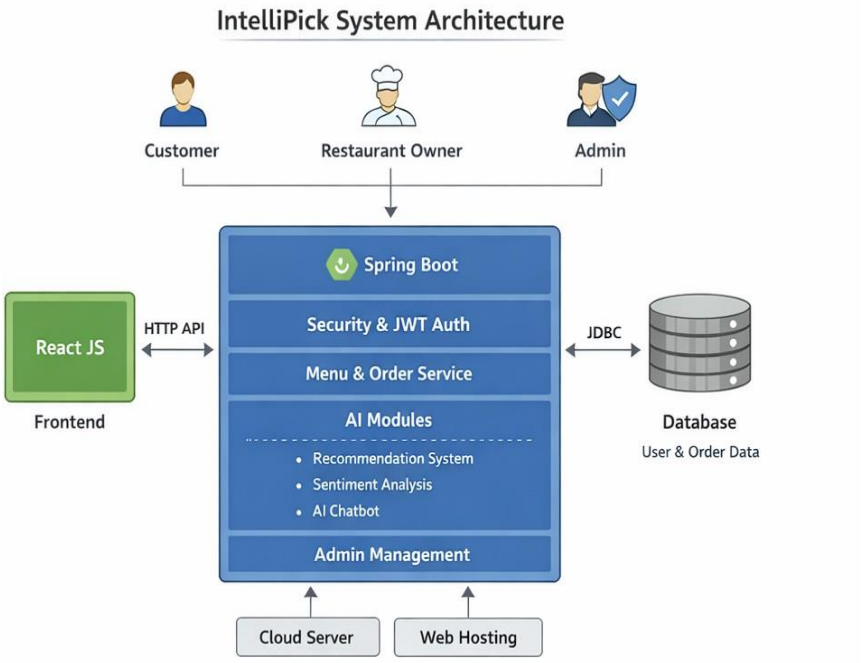

Figure 1. High-level IntelliPick three-tier architecture. Customers, restaurant owners, and administrators access the React-based presentation layer, which communicates with a Spring Boot application layer and a MySQL data layer.

Presentation Layer: The presentation layer is implemented in React.js and comprises separate interface flows for each role. Customers access views for restaurant lists, detailed restaurant pages, menu browsing, cart management, and order tracking. Restaurant owners use views for menu and order administration. Administrators have access to sections for approvals and monitoring. All views are designed to be responsive so that they can be used comfortably on both desktop and mobile browsers. The front end communicates with the backend via RESTful APIs. Axios is used to issue HTTP requests and manage responses.

Application Layer: The application layer is built on Spring Boot and exposes endpoints for authentication, restaurant and menu operations, order processing, recommendation queries, sentiment analysis tasks, and admin functions. Spring Security and JWT are used together to enforce role-based access, a pattern also adopted in earlier full-stack food-ordering implementations [16]. Each endpoint checks the caller's JWT, which encodes role information, before performing the requested operation.

Data Layer: The data layer uses MySQL to persist application entities. The main categories of data are: user accounts, profiles, and roles; restaurant records, including names, addresses, and geographic coordinates; menu items with attributes such as name, description, price, category, status, and image path; orders with associated items, totals, statuses, and timestamps; and customer reviews and optional ratings. The AI components read from and write to this database through well-defined interfaces, avoiding any direct coupling between machine learning logic and transactional operations.

4. Algorithms and Intelligent Components

A. Dish Recommendation Logic

The dish recommendation module in IntelliPick is intentionally grounded in signals that can be readily obtained from order histories. It focuses on identifying popular and contextually appropriate items rather than attempting to fully model user taste, a strategy that is consistent with practical approaches in several food-delivery applications [3].

The module considers the following aspects:

- overall popularity of menu items as measured by order counts;
- co-occurrence of items within orders, which captures frequently ordered combinations;
- category-level popularity within each restaurant; and
- when sufficient history exists, a customer's own ordering patterns.

A simplified version of the recommendation procedure can be described as:

- Retrieve past order records from the database.
- Compute the order frequency for each menu item.
- Rank items by frequency, optionally restricting to a specified restaurant or category.
- For a given customer and restaurant, take the top-N ranked items, applying simple rules to avoid recently repeated selections if desired.

Although this method is straightforward, it aligns with how frequency-based heuristics have been used to generate effective baselines in practice [3]. The design also leaves room to incorporate more advanced models, such as those that integrate sentiment-aware features [2] or graph-based representations used in large platforms [5], should data availability and deployment constraints permit.

B. Distance Calculation with the Haversine Formula

Restaurant discovery in IntelliPick is based on geographic proximity. To rank restaurants by distance from the customer, the system uses the Haversine formula, which provides a numerically stable way to compute great-circle distances on a sphere [7]. Let (φ_1, λ_1) denote the latitude and longitude of the customer and (φ_2, λ_2) that of a restaurant, all expressed in radians. The distance d between these points is:

$$d = 2R \cdot \arcsin\left(\sqrt{\left[\ \sin^2(\Delta\varphi/2) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2(\Delta\lambda/2)\ \right]}\right)$$

where $\Delta\varphi = \varphi_2 - \varphi_1$, $\Delta\lambda = \lambda_2 - \lambda_1$, and R is the Earth's radius (approximately 6371 km). In operational terms, the distance calculation proceeds as follows: the customer's coordinates are obtained from the browser's geolocation API with explicit user permission; restaurant coordinates are retrieved from the database; the Haversine formula is applied for each restaurant to compute d; and the resulting list is sorted in ascending order of d. This approach yields accurate ordering for the typical distances encountered in local restaurant searches and can scale to many restaurants per query when implemented efficiently.

C. Sentiment Analysis of Customer Reviews

Customer reviews contain a mixture of factual comments and subjective impressions about food quality, service, ambience, timing, and pricing. IntelliPick uses sentiment analysis to extract an overall polarity for each review (positive, neutral, or negative) and to compute aggregate indicators at the restaurant level.

A typical processing pipeline includes:

- text preprocessing (tokenisation, lowercasing, removal of stop words and punctuation, and optional lemmatisation);
- feature representation, using either classical schemes such as term-frequency–inverse-document-frequency or embeddings drawn from pre-trained language models; and
- classification with a supervised model trained on labelled sentiment data.

The design of this pipeline is inspired by studies that apply deep models to restaurant and food-delivery reviews and report strong performance on sentiment classification tasks [8]. Insights from broader work on sentiment-aware recommender systems can also be incorporated when constructing features, especially when linking sentiment signals to recommendation logic [10]. Within IntelliPick, sentiment scores are presented to restaurant owners through dashboards. Over time, aggregated sentiment trends can help owners understand whether specific interventions, such as menu changes or service improvements, are reflected in customer feedback.

5. System Implementation and Workflows

A. Technology Stack

The IntelliPick implementation follows a technology stack that closely matches those of existing food-ordering projects built on modern web frameworks [16], [17]. Table I summarises the main components.

TABLE 1 Technology Stack

Layer	Technology
Frontend	React.js, JavaScript, HTML5, CSS3, Axios
Backend	Java, Spring Boot, Spring Security, REST APIs, JWT Authentication
Database	MySQL
AI Features	Recommendation logic, sentiment analysis libraries
Location Services	Browser Geolocation API, Haversine

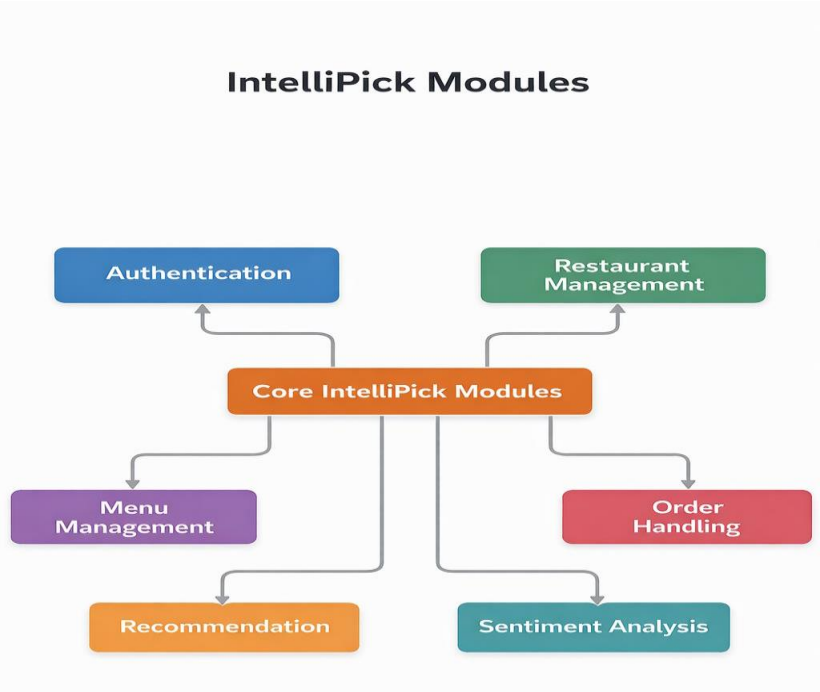

Figure 2. Illustrative IntelliPick modules: authentication, restaurant management, menu management, order handling, recommendation, and sentiment analysis.

B. Authentication and Role Management

Authentication is implemented using JWT tokens controlled by Spring Security. During registration, users supply basic profile information and are assigned one of the defined roles (customer, restaurant owner, or administrator). On successful login, the backend generates a signed JWT that encodes the user's identity and role. Subsequent API calls include this token, and middleware verifies its signature and extracts the embedded role information. This pattern is similar to those employed in other Spring Boot and React-based online ordering systems, where JWTs offer a stateless and scalable means of managing user sessions

[16]. Endpoint-level security constraints ensure that each role can only perform actions aligned with its responsibilities.

C. Menu and Image Management

Restaurant owners interact with IntelliPick through a dashboard that allows them to create, modify, and delete menu items. Each item is described by: a name and short description; a price and category (for example, starters, mains, desserts); an availability flag; and an optional image path. Images are uploaded from the browser, stored on the server or a dedicated storage service, and referenced in the database by path. The React front end then uses these paths to render images in both restaurant and customer views. Visual cues of this sort are known to improve user engagement in menu browsing contexts, as seen in commercial deployments where images and contextual information guide food choices [4].

D. Order Management and OTP-Based Pickup

Customers assemble orders by selecting items from a restaurant's menu and placing them in a cart. Once an order is submitted, the backend creates a corresponding record and updates the restaurant's dashboard. The restaurant can then accept the order, begin preparation, and update its status through several intermediate states (for instance, accepted, in preparation, ready).

When an order is marked as ready, IntelliPick generates a unique OTP associated with that order. The customer receives the OTP and must present it at the restaurant. Staff verify the code using the dashboard; if the code matches the stored value, the pickup is authorised and the order is marked as completed. This sequence helps ensure that orders are handed over to the correct customer without requiring additional hardware or complex integration.

E. Location-Based Restaurant Discovery

Upon visiting the platform, customers may grant permission for their approximate location to be accessed via the browser's geolocation API. The backend then uses the stored coordinates of registered restaurants, combined with the customer's coordinates, to compute distances using the Haversine formula. The resulting ranked list is returned to the front end, which can present restaurants ordered by proximity or grouped into distance bands. This design reflects common practice in modern location-aware food and restaurant services, where proximity is a primary factor in the ranking logic [7].

F. Role-Specific Workflows

The interactions between users and the system can be organised into three main workflows, illustrated conceptually in Figure 3

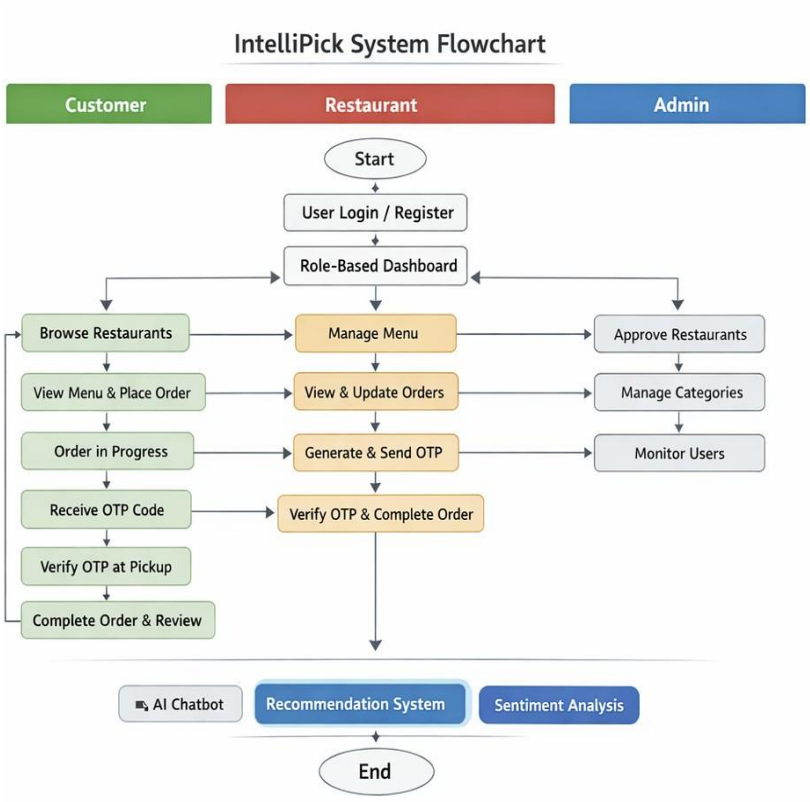

Figure 3. Simplified IntelliPick workflows for customers, restaurant owners, and administrators across registration, authentication, ordering, OTP-based pickup, and management steps.

Customer Workflow: The customer's path through the system typically follows these steps:

i. register or log in to obtain an authenticated session;

ii. optionally allow geolocation so that nearby restaurants can be identified;

iii. view a list of restaurants sorted by distance and navigate to detailed views;

iv. browse menus with images, receiving dish suggestions informed by the recommendation module;

v. place an order and monitor its status in real time;

vi. and after receiving an OTP for a ready order, travel to the restaurant and present the code at the counter.

Restaurant Workflow: The restaurant workflow proceeds as follows:

i. register as a restaurant on the platform;

ii. wait for an administrator to approve the registration;

iii. after approval, configure restaurant details, categories, and menu items with associated images;

iv. receive and manage orders, updating statuses as preparation progresses;

v. generate and communicate OTPs to customers when orders are ready;

vi. and verify the customer's OTP at pickup and close the order.

Administrator Workflow: Administrators support the overall health of the platform by

i. logging in with privileged credentials;
ii. reviewing and approving or rejecting new restaurant registrations;
iii. maintaining global category lists and any platform-wide configuration;
iv. and monitoring key indicators such as numbers of active users, orders, and restaurants.

6. Evaluation Metrics and Experimental Setup

A. Evaluation Metrics

Several metrics are used to gauge IntelliPick's performance and effectiveness.

- System response time: measures the time from a user action (such as placing an order) to the receipt of the corresponding response from the backend.
- Recommendation behaviour: captures whether recommended dishes align with observed popularity patterns and elicit user interest.
- Location accuracy: evaluates whether restaurants are ordered as expected by distance in common usage scenarios.
- Scalability: reflects how well the system maintains performance when handling multiple concurrent users and restaurants.

System response time is an important practical concern in interactive web applications and is used widely in performance evaluations of related systems [3]. Recommendation behaviour is probed qualitatively by examining whether top suggestions correspond to frequently ordered dishes, a principle that underpins frequency-based systems [3]. Location accuracy is assessed by checking whether restaurants that are known to be nearby appear near the top of proximity-based lists.

B. Experimental Setup

A prototype deployment of IntelliPick was created using realistic but synthetic data. Several restaurants were populated with menus and geographic coordinates representing plausible city locations. Synthetic customers were registered, and scripted interactions were used to simulate order placement, menu browsing, and review submission.

The experiments focused on: registering and configuring restaurants and menu items; placing and updating orders under varying simulated loads; generating and verifying OTPs as part of pickup; invoking the recommendation module under accumulated order histories; and performing location-based discovery using the Haversine implementation. Average backend response times for order-related operations were observed to fall between approximately 1.5 and 2 seconds in the test environment, which is in line with expectations for a full-stack web application of this type [16]. Distance-based ranking consistently produced an ordering of restaurants that matched their intended relative positions.

7. Results and Comparative Analysis

A. Experimental Observations

Table II summarises key observations from the prototype deployment. The results indicate that IntelliPick can manage the essential operations of a pickup-centric ordering platform without noticeable delays under the tested conditions. The architecture supports efficient database queries and REST interactions, and the AI modules operate without introducing significant overhead.

TABLE 2 Representative Experimental Observations

Feature	Observation
Order placement and processing	Backend responses typically below 2 seconds
Distance calculation	Restaurant ordering matches geographic proximity
Recommendation behaviour	Popular dishes and frequently ordered items highlighted
Image handling	Menu images rendered reliably across views
OTP verification	Pickup confirmed successfully for all tested orders

B. Comparison with Delivery-Centric Applications

IntelliPick can be contrasted with traditional delivery-focused food applications along several dimensions, as shown in Table III. The large-scale platforms described in prior work have understandably optimised their recommendation and ranking logic for delivery conversion and marketplace outcomes [5]. IntelliPick, by design, reorients these intelligent components around pickup flows and direct restaurant–customer interactions. The use of sentiment and recommendation modules in service of restaurant analytics and pickup experience differentiates the platform from conventional delivery services.

TABLE 3 Qualitative Comparison with Delivery-Focused Applications

Aspect	Delivery-Centric	IntelliPick
Delivery dependency	Core to the model	Not used (pickup only)
Cost structure	Multiple fees and commissions	Direct restaurant pricing
Recommendation focus	Marketplace-wide optimisation	Restaurant-centred and extensible
Sentiment visibility	Often internal to platform	Surfaced to restaurant owners
Location-based ranking	Present but often opaque	Explicit distance-based ordering
Pickup workflow	Typically secondary	Primary interaction mode

8. Conclusion and Future Work

This paper has presented **IntelliPick**, an AI-enabled smart restaurant ordering and pickup platform intended as a complementary alternative to delivery-heavy food-ordering systems. By focusing on advance ordering and secure in-person collection, IntelliPick reduces dependence on third-party logistics and associated fees while still providing the convenience of a modern web-based interface. The platform combines a React front end, Spring Boot backend, and MySQL database with components for dish recommendation, sentiment analysis, and distance-aware restaurant discovery.

A prototype deployment and experimental study have demonstrated that IntelliPick can process orders within acceptable response times and produce proximity rankings and recommendation behaviour that align with expectations. The architecture draws on established patterns from other full-stack food-ordering projects [16] and incorporates ideas from recent research on sentiment analysis and recommendation in the restaurant domain [1], [8].

Future work will explore several directions. One is to refine the recommendation module by incorporating more detailed user and context features, potentially taking inspiration from hybrid and deep models used in other sentiment-aware recommender systems [10]. Another is to develop richer analytics dashboards for restaurant owners, using sentiment and order data to highlight trends and support decision-making. Finally, building a dedicated mobile application and integrating real-time order tracking could further improve the user experience for both customers and restaurants.

References

1. S. Hussain, M. W. A. Khan, A. A. Shah, M. Abid, and S. A. H. Shah, "Sentiment analysis of customer reviews of food delivery services using deep learning and explainable artificial intelligence: A systematic review," Foods, vol. 11, no. 15, pp. 1–27, 2022.
2. I. Alagha and M. Samson, "A restaurants recommendation system: Improving rating predictions using sentiment analysis," in Proc. 2020 IEEE Intl. Conf. on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 1–6.
3. R. R. Arifin, R. Nugraha, and S. Suriadi, "Recommendation system for a delivery food application based on number of orders," Applied Sciences, vol. 13, no. 4, pp. 1–19, 2023.
4. A. Goyal, "FoodNet: Simplifying online food ordering with contextual food combos," Swiggy Engineering Blog, 2019.
5. S. Pasumarthi et al., "Food discovery with Uber Eats: Recommending for the marketplace," Uber Engineering Blog, 2019.
6. R. Duarte, R. Sawczuk, and B. Ponce, "Personalized recommendation of dish and restaurant collections on iFood," arXiv preprint arXiv:2508.03670, 2025.
7. R. W. Sinnott, "Virtues of the Haversine," Sky and Telescope, vol. 68, no. 2, p. 159, 1984.
8. A. Ghorbani, M. A. Ahmad, and J. D. Martin, "Deep hierarchical networks for sentiment analysis of restaurant reviews from food apps," Scientific Reports, vol. 15, art. 23856, 2025.

9. A. A. Khan and S. M. Khan, "Restaurant recommender system based on sentiment analysis," International Journal of Creative Research Thoughts, vol. 11, no. 5, pp. 2308–2315, 2023.
10. M. A. Rahman, M. M. Alam, and M. N. Islam, "Integrated sentiment analysis with BERT for enhanced hybrid recommendation systems," Expert Systems with Applications, vol. 245, art. 123123, 2024.
11. A. Meydan, "Restaurant reviews: Sentiment analysis and recommendation," Analytics Vidhya, 2020.
12. S. Hussain and I. A. Babar, "An integrated framework utilizing hybrid LDA and BERT for enhanced hotel recommendation systems," in Proc. 2023 Intl. Conf. on Data Science and Advanced Analytics, 2023.
13. S. Shah, "Uber Eats food discovery: A deep dive into ML-powered recommendations – Part 2," Medium, Feb. 2026.
14. Pickup.ph, "Free online ordering system and delivery app," 2024. [Online]. Available: https://www.pickup.ph/
15. Deonde Technologies, "Self pickup ordering system: Click and collect for restaurants," 2024. [Online]. Available: https://www.deonde.co/self-pickup-ordering-system.shtml
16. M. Murad, "Multi-restaurant online food ordering system project using Spring Boot + React JS + MySQL," CodeWithMurad, Oct. 2023.
17. D. Iluma, "Build a full-stack food ordering and delivery app: Spring Boot, React, payments, and AWS cloud deployment," DEV Community, 2022.
18. "Web-based food delivery app," GitHub repository, 2021. [Online]. Available: https://github.com/Sowmiya81/Web-based-Food-Delivery-App
19. "CraveDash: Online food ordering and delivery platform," GitHub repository, 2022. [Online]. Available: https://github.com/Shobhits1/CraveDash
20. Appetier Inc., "Appetier – AI restaurant OS: Direct online ordering and local SEO," 2024. [Online]. Available: https://appetier.com/
21. Papaya, "Papaya POS – AI restaurant POS, QR ordering and payments," 2024. [Online]. Available: https://papayapos.ai/
22. T-Menu, "T-Menu – AI-powered digital menu and online ordering platform," 2024. [Online]. Available: https://tmenu.ai/