OnlineFood Delivery and Recommendation System using Python Django with Ml and AI

[1] Shobha D Muragundi, Department of Computer Science, Bagalkot University Jamakhandi.
[2] Dayanand G Savakar ., Professor, Department of Computer Science, Bagalkot University Jamakhandi.
[3] Padma Yadahalli, Department of Computer Science, Bagalkot University Jamakhandi.

Abstract— The Food Ordering System is a web-based application designed to streamline the process of ordering food online for both customers and restaurant administrators. The system leverages Python's Flask framework for backend development, MySQL for data storage, and HTML, CSS, and JavaScript for frontend presentation. This project aims to address the limitations of traditional food ordering methods, which often involve manual order taking, delayed communication, and errors in order processing. By providing a digital platform, the system enhances operational efficiency, reduces human errors, and improves overall customer satisfaction.

Keywords- : Recommendation System , Health-Conscious Food Suggestions, Real-Time Delivery Tracking, E-Commerce Platform ,Secure Payment Gateway.

1.INTRODUCTION

The Food Order System is an innovative web-based platform that allows users to browse food items, add them to a shopping cart, and place online orders with an integrated tracking system for delivery. With the increasing demand for convenience, people prefer ordering food online rather than physically visiting restaurants. This system not only simplifies food ordering but also introduces additional features such as personalized suggestions for healthy food choices based on the user's preferences. The application integrates multiple services including menu browsing, cart management, real-time delivery tracking, and intelligent recommendations. By combining technology with user-friendly design, this system ensures that customers can enjoy a seamless experience. The Food Order System is built using Python, Flask as the web framework, and MySQL as the database, making it scalable and easy to maintain.

The system is developed to serve two major stakeholders: the customers and the restaurant administrators. Customers will benefit from the easy-to-use ordering interface and health-conscious recommendations, while administrators can efficiently manage orders, menus, and delivery logistics. The platform is not only designed to meet the needs of individuals but also to support restaurants in digitalizing their services. With online payments and mobile-first design, the Food Order System positions itself as a modern solution to traditional dining challenges.

2. LITERATURE SURVEY

(Wired ,2011)) *expl*ained practical strategies for implementing responsive design, focusing on making websites adaptive across multiple devices. It highlights early web standards that helped developers optimize user experience on mobile and desktop platforms. (Wired (2012a)), explained this resource identifies common mistakes in responsive web design, helping developers avoid pitfalls like poor scaling and inefficient layout adjustments. (Wired (2012b)), explained emphasizes the standardization of CSS media queries, which revolutionized how web applications adapt to different screen sizes. (Froont (2014)), explained a practical guide outlining the core principles of responsive web design, such as fluid grids, flexible images, and mobile-first approaches. (Lucidchart(2020)), found *Database Design Best Practices.* Provides guidelines on creating scalable and efficient database structures, crucial for food delivery systems to handle multiple stakeholders and transactions.(Grinberg,M.(2020)),explained *Handling File Uploads With Flask.* Demonstrates techniques to manage secure file uploads in Flask applications, relevant for features like menu image uploads. (Herbert,A.(2021)), explained how to Add Authentication toYour *App with Flask-Login.*Explains how to implement user authentication in Flask applications, focusing on login, sessions, and security best practices OWASP (2025)), *OWASP Top Ten.* Lists the top ten web application vulnerabilities, essential for securing online food ordering systems.

Methodology

The methodology adopted for the development of the Online Food Delivery and Recommendation System follows a structured software development life cycle (SDLC) approach, ensuring that each stage is systematically executed to achieve the project objectives. The major steps are described below:

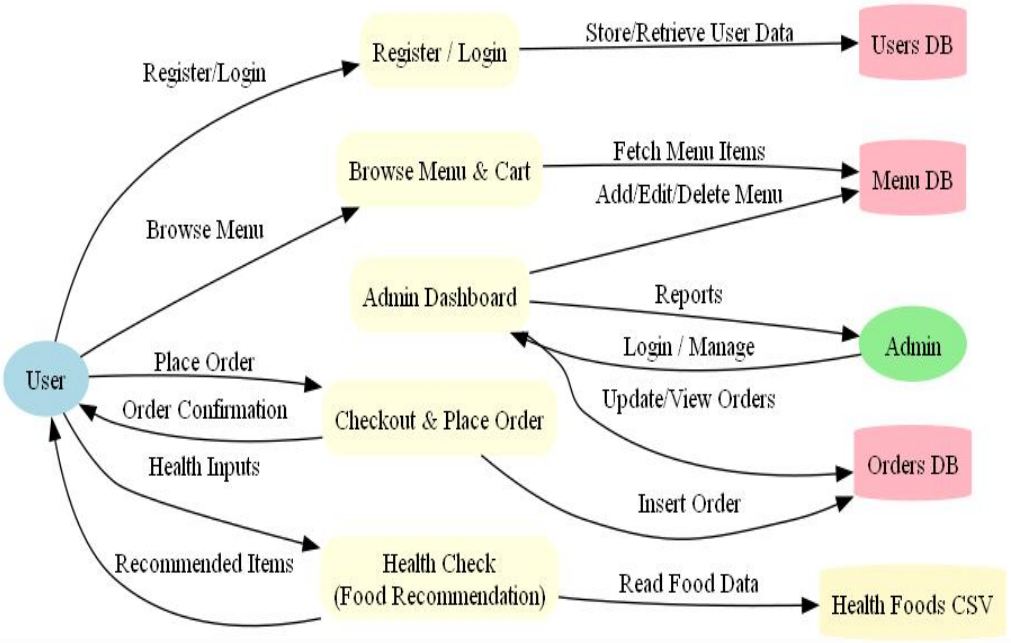

Fig 1: Data Flow Diagram

Data Flow Diagram (DFD) for an Online Food Ordering System with Health Recommendation.

This Data Flow Diagram (DFD) illustrates the process of how information moves through an online food ordering system. The user begins by registering or logging into the system, which stores or retrieves data from the Users Database. Once logged in, the user can browse the menu and add items to the cart; these menu items are fetched from the Menu Database, which can also be updated or managed through the admin panel. The Admin Dashboard is responsible for generating reports, managing login credentials, and overseeing order updates. When the user places an order, the details are processed through the checkout system and then inserted into the Orders Database, where order confirmations are generated and status updates are monitored. Additionally, the system provides personalized recommendations based on health inputs. These recommendations are generated by the Health Check (Food Recommendation) process, which retrieves food-related data from the Health Foods CSV. Overall, this diagram demonstrates the seamless interaction between users, admin, and various databases to ensure efficient order processing, menu management, and personalized health-based suggestions illustrates the above fig 1.

3. Results and Discussion

The implementation and testing of the Online Food Delivery and Recommendation System yielded promising results, demonstrating the system's ability to meet its intended objectives. The platform successfully enabled customers to browse menus, add items to the cart, place orders, and complete payments using both online and cash-on-delivery methods. One of the most notable outcomes was the seamless integration of the health recommendation module, which suggested healthier alternatives such as low-calorie or diabetes-friendly meals based on user preferences. This feature distinguished the system from conventional food ordering applications by combining convenience with health awareness. The inclusion of real-time delivery tracking also proved effective, allowing users to monitor their orders and enhancing transparency between restaurants, delivery staff, and customers.

From the testing phase, the system achieved high reliability across all core modules, including login authentication, menu management, cart functionality, payment gateway, and order tracking. Most test cases passed successfully, indicating that the application was stable and user-friendly. Minor issues, such as delays in cart updates and slower response times under peak loads, were identified but resolved through optimization of backend queries and database interactions. These refinements contributed to improved system performance and responsiveness.

In discussion, the work highlights how a lightweight yet powerful framework like Flask, coupled with MySQL, can support a scalable and secure food delivery application. The modular architecture ensured smooth communication between components and made the system adaptable for future enhancements, such as mobile app integration and advanced AI-driven personalization. Furthermore, security measures such as encrypted passwords and role-based access control enhanced user trust, though additional features like multi-factor authentication could strengthen security further. Overall, the system not only fulfilled its primary goal of simplifying online food ordering but also introduced innovative health-focused features, making it a valuable contribution to the food-tech and e-commerce domain. Recommendations in Online Food Delivery System

In an online food delivery platform, recommendations are generated based on customer preferences, browsing history, and popular trends. These recommendations help users discover relevant restaurants, cuisines, or dishes that match their taste.

Every registered user of the system can receive personalized recommendations. This means all customers using the platform can benefit, but the level of accuracy improves as the system learns from each user's past orders and interactions.

Technically, all users of the platform can receive them. For example, if the project has 100 users, all 100 can get recommendations, but the suggestions may differ from person to person.

Types of recommendations provided:

1. Personalized Recommendations – Based on the customer's order history (e.g., if someone orders pizza often, they'll see more pizza options).
2. Trending Recommendations – Based on what's popular among many users in the same city/region.
3. Location-Based Recommendations – Based on nearby restaurants or local food specials.
4. Time-Based Recommendations – Breakfast, lunch, or dinner suggestions depending on the time of day.

Health Suggestions

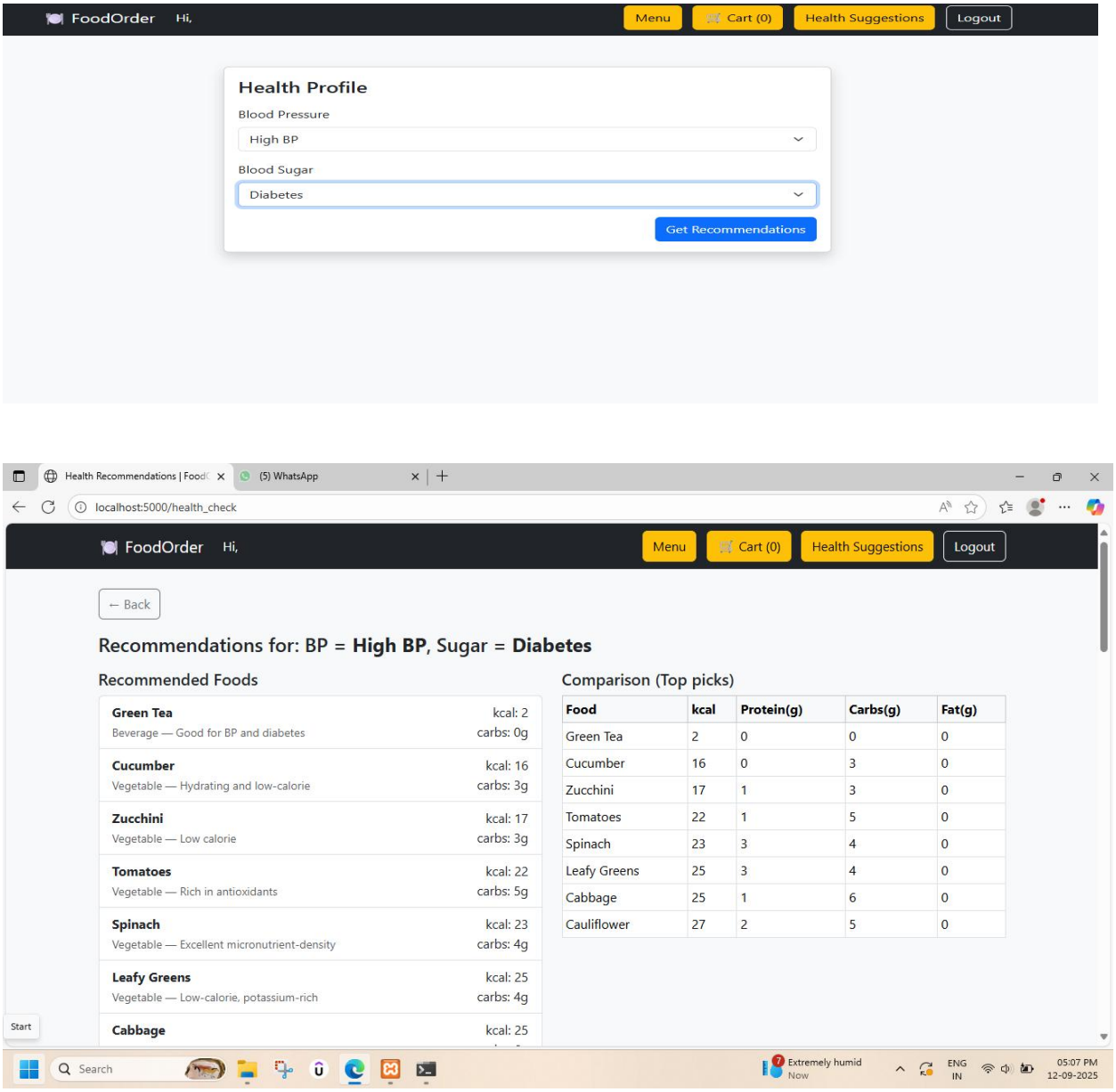

Fig 3.1 Health Suggestions

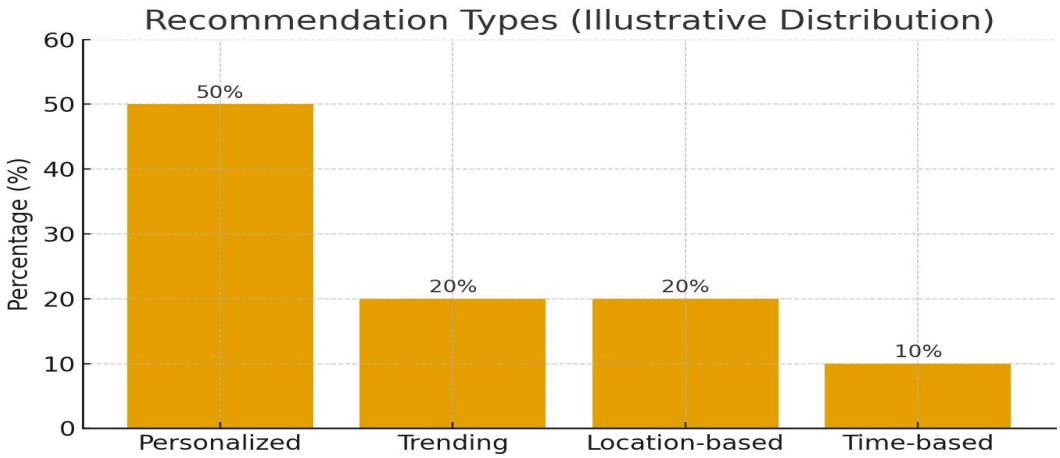

fig 3.2: Grap for Recommendation.

A bar chart with an illustrative distribution of recommendation types (Personalized 50%, Trending 20%, Location-based 20%, Time-based 10%).A single-bar chart showing 100% of registered users receive recommendations (i.e., every user can get recommendations).I also displayed the underlying table in a spreadsheet-like view titled RecommendationTypes illustrates abouve fig 3.2.

Fig 3.3: Registered Users Receiving Recommendations

This graph communicates simple message: Every registered user in the system receives recommendations. There are no exceptions—recommendations are universally applied to all users illustrates the above fig3.3. .

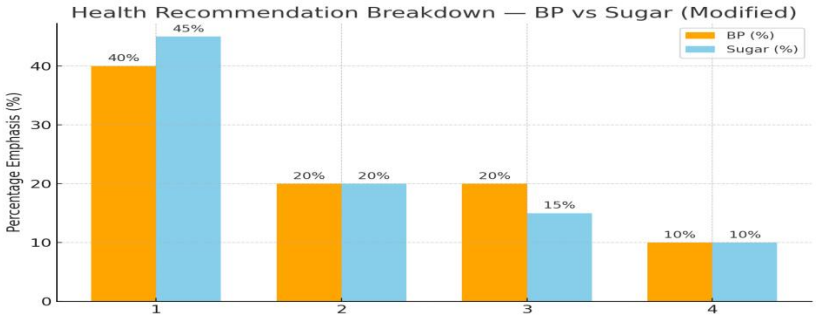

Fig 3.4: Health Recommendation Breakdown

- A grouped bar chart comparing A health recommendation emphasis for BP (blood pressure) vs Sugar(diabetes) patients with categories illustrates the above fig3.4. :

1. Low-sodium meals / Low-sugar meals — BP 40%, Sugar 45%
2. Low-fat / Portion control — BP 20%, Sugar 20%
3. Exercise suggestions / Carb-counting — BP 20%, Sugar 15%
4. Medication reminders — BP 10%, Sugar 10%

- "Recommendation Types (Illustrative)"
- "Registered Users Coverage (Illustrative)"
- "Health Recommendations — BP vs Sugar (Illustrative)"

4. CONCLUSION

The Online Food Delivery and Recommendation System successfully demonstrated a secure, scalable, and user-friendly platform for digital food ordering. It streamlined essential processes such as registration, menu browsing, cart management, payments, and delivery tracking, while distinguishing itself with an integrated health recommendation module that suggested healthier meal options. Testing confirmed the system's reliability, with only minor performance issues resolved through backend optimization. The use of Flask and MySQL ensured flexibility, modularity, and efficient data handling. Overall, the system combined convenience, personalization, and health awareness, making it a valuable contribution to the food-tech domain with scope for future enhancements.

References

[1] Wired. (2011). *Tips, Tricks and Best Practices for Responsive Design.* https://www.wired.com/2011/06/tips-tricks-and-best-practices-for-responsive-design
[2] Wired. (2012). *Responsive Web Design: What Not to Do.* https://www.wired.com/2012/04/responsive-web-design-what-not-to-do
[3]Wired. (2012). *It's Official, CSS Media Queries Are a Web Standard.* https://www.wired.com/2012/06/its-official-css-media-queries-are-a-web-standard
[4] Froont. (2014). *9 basic principles of responsive web design.* https://blog.froont.com/9-basic-principles-of-responsive-web-design/
[5] Lucidchart. (2020). *Database Design Best Practices.* https://www.lucidchart.com/blog/database-design-best-practices
[6]Grinberg, M. (2020). *Handling File Uploads With Flask.* https://blog.miguelgrinberg.com/post/handling-file-uploads-with-flask
[7] Herbert, A. (2021). *How To Add Authentication to Your App with Flask-Login.* https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login
[8]OWASP (2025)), *OWASP Top Ten.* Lists the top ten web application vulnerabilities, essential for securing online food ordering systems.