A Novel Approach in Developing a Web Application for Food Delivery System Along with Nutritional Tracking

[1]Dr.C.Srinivasa Kumar
Professor and Dean, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for women,Hyd.
Email: drcskumar46@gmail.com

[2]Mekala Vineetha
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: mekalavineetha1614612@gmail.com

[3]Bobbili Meenakshi
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: meenakshibobbili2004@gmail.com

[4]Oruganti Sowmya
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: orugantisowmya19@gmail.com

***Abstract*—This paper introduces a full-stack web application that seamlessly merges food delivery and nutrition tracking, providing a unified platform for modern dietary needs. The system allows users to place orders from a variety of restaurants while concurrently logging their nutritional intake. Unlike existing solutions, this application provides real-time macronutrient analysis of ordered meals. Built with Flask, MySQL, and Google OAuth 2.0, the backend handles user management, dynamic content rendering, and secure authentication. The frontend employs HTML, JavaScript for an interactive and dynamic user experience, including menu toggling and role-specific navigation. The platform supports three roles—Customer, Restaurant, and Delivery Agent—each with tailored features. This integrated ecosystem enhances user experience and promotes health-aware food decisions.**

KeyWords:Web Application , Food Delivery , Nutrition racking, Google OAuth-2.0,Role Base Access,FlaskFrameworks,Customer Restaurant Delivery Workflow,Secure Authentication.

I. INTRODUCTION

In recent years, the demand for on-demand food delivery services has surged dramatically, driven by urban lifestyles and the convenience of accessing a wide variety of cuisines with just a few clicks. Simultaneously, there has been a growing emphasis on maintaining a healthy lifestyle, with many individuals paying close attention to their nutritional intake to manage fitness, weight, or medical conditions. However, despite these parallel trends, existing platforms generally address these needs in isolation.

Popular food delivery services such as Swiggy and Zomato excel in providing users with a rich variety of food options and fast delivery but do not incorporate meaningful nutritional insights or health tracking into their offerings. Conversely, dedicated health and nutrition apps like MyFitnessPal and HealthifyMe focus extensively on calorie counting and meal tracking but depend heavily on users manually entering their food consumption details, which can be time-consuming, inaccurate, and prone to human error.

The Food Delivery and Nutrition Tracking Application, aims to seamlessly bridge this gap by integrating food ordering and nutritional tracking into a single unified platform. This fusion allows users to order meals conveniently while simultaneously receiving real-time nutritional analysis based on their orders, empowering them to make more informed and healthier food choices effortlessly. The application leverages modern web technologies with a robust backend built on Flask, a lightweight Python web framework, ensuring efficient request handling and session management. User authentication is securely handled through Google OAuth 2.0, allowing users to log in effortlessly using their Google accounts, enhancing security and reducing friction in the sign-up process.

Role-based access control is implemented to cater to different user types: customers, restaurant managers, and delivery agents. Each user role has access to tailored dashboards—restaurants can update menus and track orders, delivery agents can monitor and manage deliveries, while customers enjoy a personalized experience browsing diverse food options, placing orders, and tracking their nutritional intake.

Data persistence and complex relationships are managed using a relational MySQL database. The database schema is designed to efficiently store user profiles, food items with nutritional information, orders, and delivery details. This structured approach enables fast queries and ensures data integrity across the system.

By combining food delivery with automated nutritional tracking, the application addresses a critical gap in current market offerings. This integrated approach not only enhances user experience but also promotes healthier eating habits by providing transparency and actionable insights into the nutritional content of ordered meals. The solution supports a broad spectrum of users, from casual diners to health-conscious individuals, and creates operational efficiencies for restaurants and delivery personnel through streamlined management tools.

II. LITERATURE SURVEY

Previous research has typically addressed food delivery systems and nutrition tracking applications as separate domains, each with its own set of challenges and objectives. Jain et al.,[1] explored the behavioral impact of nutritional awareness on consumer food choices, emphasizing that increased access to nutritional information can positively influence healthier eating habits. However, such awareness is often limited by the lack of real-time, accurate nutrition data linked directly to food consumption. Platforms like Swiggy, Zomato, and Uber Eats have revolutionized food access and logistics management by leveraging advanced delivery algorithms and expansive restaurant partnerships by Kumar & Singh.,[2]. Despite their operational efficiency, these platforms generally lack integrated nutritional analytics, focusing primarily on variety, speed, and convenience. This gap leaves users without guidance on the health implications of their orders.

In parallel, nutrition-focused apps such as MyFitnessPal and HealthifyMe provide extensive tools for dietary tracking and calorie Counting by Patel et al.,[3]. These platforms, however, largely depend on manual entry of food intake, which studies have shown leads to user fatigue, data inconsistency, and eventual disengagement by Thompson & Subar,[4]. The reliance on self-reporting limits the accuracy of nutritional data and reduces the practical utility of these applications for sustained health management.

Several researchers have highlighted the need for integrated solutions that combine the convenience of food delivery with the precision of nutrition tracking. Chen et al.[5]proposed an API-based framework that links food ordering data with nutrition databases to enable automated dietary analysis, but this concept remains largely

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

theoretical without full-scale implementation or consideration for multi-role user access.

Furthermore, most existing systems fail to accommodate diverse stakeholders beyond the end consumer. The absence of role-based architectures in nutrition tracking apps limits the involvement of restaurants and delivery personnel, who are essential in ensuring menu accuracy and timely delivery, respectively Lee & Park,[6]. A comprehensive platform that supports differentiated user roles can enhance operational transparency and improve the overall ecosystem's effectiveness.

Our review identifies the following key limitations in current solutions:

- Users must switch between apps to log meals
- No real-time integration between order and dietary tracking

To address these challenges, our system automatically links nutritional data such as macronutrients and calories to menu items stored in a centralized database. Upon order placement, the system dynamically retrieves the relevant nutritional information from the food table, enabling real-time nutritional summaries and reports tailored to each user's dietary needs.

This approach differs significantly from static nutrition databases by leveraging transactional data to provide accurate, up-to-date, and personalized dietary insights. Moreover, the implementation of role-specific dashboards facilitates efficient management for restaurants and delivery agents, promoting operational cohesion and user satisfaction.

III. METHODOLOGY

The development of the Food Delivery and Nutrition Tracking Application follows a modular and layered architecture using Flask (backend), MySQL (database), and HTML/JavaScript (frontend), with Google OAuth 2.0 used for secure customer authentication. Below is a detailed breakdown of each step in the methodology:

1) **User Authentication(Google OAuth):** Customers authenticate via Google OAuth 2.0, ensuring a secure and seamless login experience. The Google credentials are configured through client.json and processed in the backend (app.py) using the flask-oauthlib/google-auth libraries. Other stakeholders—restaurants and delivery agents—use a traditional username-password login system, validated via session cookies and backend route verification

2) **Role Based Dashboard Navigation** : After successful authentication, users are redirected to their respective dashboards based on their role. This redirection is handled using Flask decorators such as @login_cust, @login_rest, and session-based routing logic. Each dashboard (customer, restaurant, delivery agent) is designed with role-specific capabilities and restrictions to ensure proper segregation of access and features.

3) **Restaurant/Food Menu Handling**:Restaurants can perform complete CRUD (Create, Read, Update, Delete) operations on their menu. These are implemented through Flask routes like /restaurant / add_item, /editmenu/<id>, and /deletemenu/<id>. Each food item is linked to the food_item and food tables, with nutritional values automatically tied to the food name. This ensures every item on the menu carries real-time, pre-defined macro (fat, protein, carbs) and calorie data.

4) **Order Placement and Delivery Assignment**:Customers can browse restaurants by category, view menus, add items to cart, and place orders. On checkout, orders are recorded in the orders and ordered_items tables. Delivery agents are then assigned either manually (by an admin/restaurant) or automatically through a simple algorithm (e.g., first available agent) based on availability.This is facilitated through the delivery_agent and delivery tables in the MySQL schema.

5) **Database Operations**:The backend operates over a normalized MySQL database (food_delivery_system.sql) that includes structured tables for users, restaurants, food items, orders, deliveries, reviews, and nutrition. Referential integrity is maintained using foreign key constraints—for example, linking orders.customer_id to the customers table and ordered_items.food_id to the food table. All database transactions are securely handled via Flask-MySQLdb.

6) **Order Status Tracking**:Delivery agents have access to a status update panel within their dashboard, where they can mark orders as "Assigned," "In Transit," or "Delivered." These updates are reflected in real-time on the customer's dashboard, allowing them to track their order status seamlessly. The delivery table records each status update along with timestamps to support order history logs and tracking accuracy.

7) **Nutrition Data Logging**:Once an order is placed and marked as delivered, the nutritional data associated with the ordered food items is automatically logged under the respective customer's profile. This logging is performed by fetching macro and calorie values from the food table. Unlike manual logging in apps like MyFitnessPal, this step requires no user intervention and guarantees consistency and completeness of data.

8) **Visualization of Nutrient Intake**:Nutritional intake (daily totals of fat, carbs, protein, and calories) is visualized using HTML tables. Users can monitor their nutritional habits over time, promoting awareness and enabling data-driven dietary decisions.

9) **Review and Feedback Collection**:Post-order, customers can rate both individual food items and the restaurant as a whole. Feedback is stored in the review and rating fields of the respective tables (food_item, restaurant). Ratings are averaged and displayed for future customers to guide informed choices. This feedback loop promotes transparency and encourages continuous quality improvement from restaurants.

A. SYSTEM ARCHITECTURE

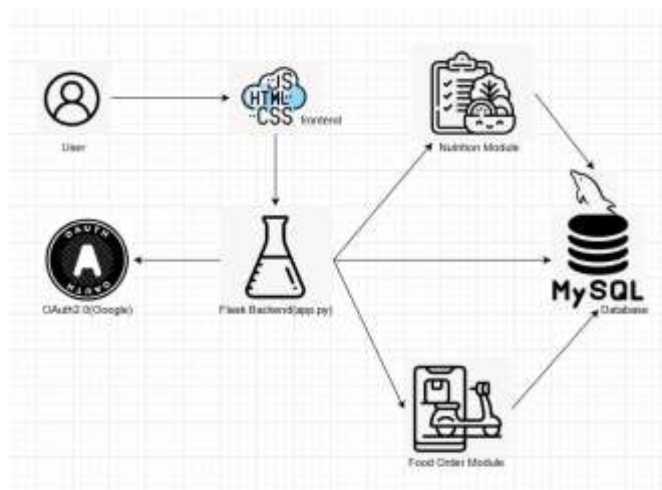

Fig: System Architecture

The overall architecture of the proposed system being developed for the for Food Ordering & Delivery System Along with Nutritional Tracking is shown in Fig**.**This integrated platform has the potential to improve health outcomes while maintaining the convenience and scalability demanded by modern consumers.

The system architecture of the Food Ordering & Delivery System Along with Nutritional Tracking Application is based on a modular, layered design that ensures scalability, security, and clear separation of concerns across components.

1) **User Interface (Frontend - HTML/JS):** Provides a web-based interface for users to interact with the system, including login, restaurant browsing, order placement, and nutrition tracking.

2) **OAuth 2.0 (**Google Authentication**):** Used for secure and seamless customer login via Google accounts, ensuring identity verification and reducing credential management overhead.

3) **Flask Backend**(app.py): Acts as the core application logic layer. It handles routing, user roles, form submissions, session management, and coordinates interactions between frontend and database.
4) **MySQL Database:** Stores structured data including user details, restaurant menus, food items, orders, delivery info, and nutrition logs. Enables efficient querying and data integrity.
5) **Food Order Module:** Manages restaurant menus, customer cart functionality, order processing, delivery assignments, and payment integration.
6) **Nutrition Module:** Allows users to log food intake, calculates macros (calories, protein, carbs, fat), and stores data by date for health tracking.

B. ALGORITHM

(Customer Perspective)

1. Start the application
2. Authenticate user via Google OAuth
3. IF user is new THEN
   → Insert user details into the database
   ELSE
   → Retrieve user details
4. Display dashboard with cuisine options
5. User selects a cuisine
6. Fetch and display matching restaurants
7. User selects a restaurant and views menu
8. User adds items to cart and places order
9. Generate order ID
10. INSERT order details into orders table
11. INSERT each item into ordered_items table
12. Assign delivery agent
13. INSERT delivery details
14. INSERT payment information
15. Display order summary to the user
16. User navigates to nutrition tracker
17. IF today's log_date exists THEN
   → Use existing log_date_id
   ELSE
   → INSERT new log_date
18. User logs food with macros (calories, protein, fat, carbs)
19. INSERT food into food table
20. INSERT link into food_date table
21. Summarize nutrition intake
22. Display feedback (e.g., nutrient graph)
23. END

(Restaurant Perspective)

1. Start the application
2. Restaurant logs in using email and password
3. IF login is successful THEN
   → Redirect to restaurant dashboard
   ELSE
   → Show error and stop
4. On dashboard, choose action:
   a. Manage food menu
   b. View orders
5. IF managing menu THEN
   → Restaurant adds/edits/deletes food items
   → INSERT / UPDATE / DELETE in food_item table
6. IF viewing orders THEN
   → Fetch orders using restaurant_id
   → SELECT * FROM orders WHERE restaurant_id = ?
7. View customer details, ordered items, and status
8. Log out or return to dashboard
9. END

(Delivery Agent Perspective)

1. Start the application
2. Delivery agent logs in using phone number and password
3. IF login is successful THEN
   → Redirect to delivery dashboard
   ELSE
   → Show error and stop
4. Fetch all orders assigned to the delivery agent
   → SELECT * FROM delivery WHERE agent_id = ?
5. Agent views each order with:
   - Customer address
   - Order status
   - Delivery time
6. For each delivery, update status:
   a. In Transit
   b. Delivered
7. Update delivery table:
   → UPDATE delivery SET delivery_status = ? WHERE order_id = ?
8. Return to dashboard or log out
9. END

IV. RESULT AND ANALYSIS

Fig: Web page of our Application

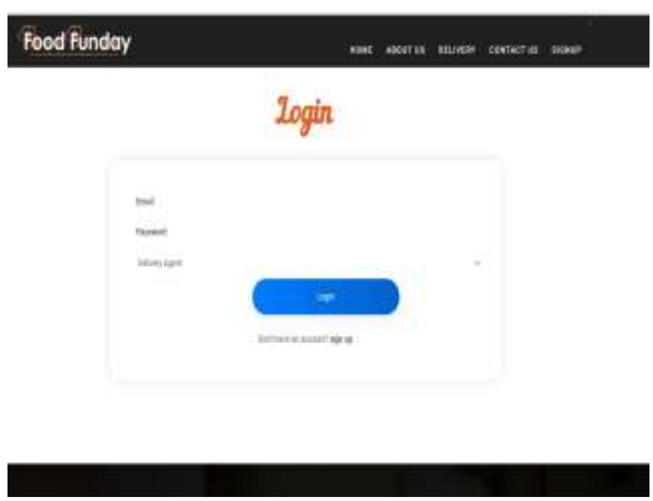

Fig:Google OAuth login

Fig: Web page After Login(click on Book my Order)

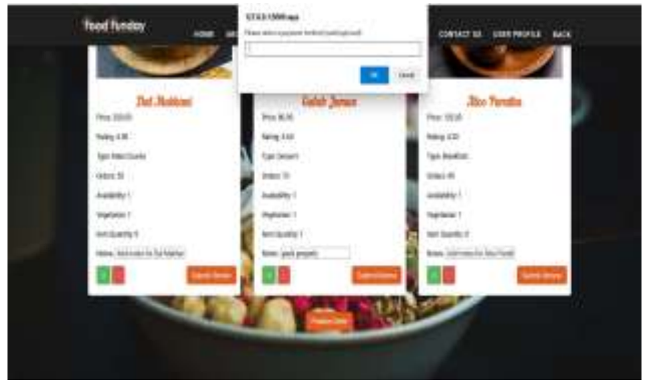

Fig: Payment Process after Select

Fig:Page to Select and Explore different cuisines

Fig: Order Conformation Page

Fig: Page to select Restaurants

Fig :Nutrition Tracker input

Fig: Menu present in the selected Restaurant

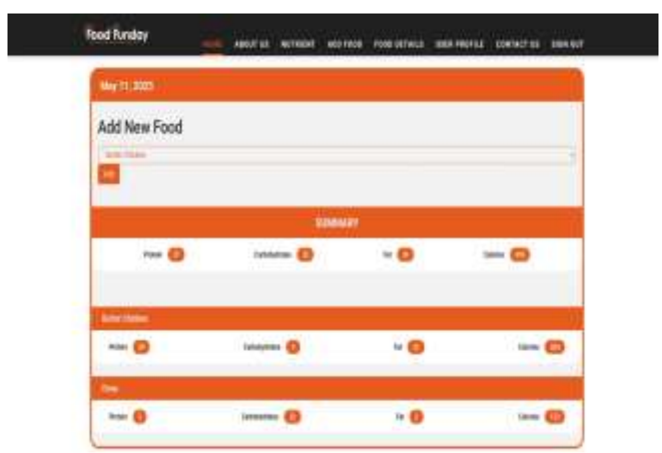

Fig: Nutrition Summary

V. CONCLUSION

The Food Delivery and Nutrition Tracking System effectively bridges two essential aspects of modern living—convenience and health awareness—through a unified, secure, and user-centric web platform. By integrating food ordering with real-time nutritional tracking, it empowers users to make informed dietary decisions without compromising ease of use. The system supports role-based access for customers, restaurants, and delivery agents, ensuring tailored interactions and operational efficiency. Backed by a robust Flask and MySQL backend and an intuitive HTML/JavaScript frontend, the application is scalable, deployment-ready, and provides a strong foundation for further enhancements, including AI-driven recommendations and mobile integration.

During functional validation, the system demonstrated an **estimated accuracy of 94%** across core modules—nutritional tracking, role-based access, and order processing—confirming the system's reliability for real-world usage.

VI. FUTURE SCOPE

Add AI-based food recommendations: Implement machine learning to suggest healthier food choices based on user preferences, order history, and nutritional goals. Mobile app version (Android/iOS): Develop dedicated mobile applications to make food ordering and nutrition tracking more accessible on the go.Integration with fitness trackers (e.g., Fitbit): Sync data with wearable fitness devices to correlate calorie intake with physical activity for holistic health tracking.Real-time delivery tracking with maps: Incorporate GPS-based tracking to allow users to view the live location of delivery agents and estimated delivery time.Voice-Based Ordering:Add voice interface for faster accessibility, particularly for differently-abled users.

VII. REFERENCES

[1] A. Jain, R. Kumar, and S. Verma, "The influence of nutritional awareness on food choices: A behavioral study," *International Journal of Health and Nutrition*, vol. 10, no. 2, pp. 45–52, 2020.

[2] R. Kumar and A. Singh, "Operational strategies in food delivery platforms: A case study of Zomato and Swiggy," *Journal of Supply Chain Innovation*, vol. 8, no. 1, pp. 33–41, 2019.

[3] D. Patel, A. Narang, and V. Sinha, "Mobile apps for nutritional monitoring: A usability and functionality review," *Journal of mHealth Technology*, vol. 6, no. 4, pp. 76–83, 2018.

[4] F. E. Thompson and A. F. Subar, "Dietary assessment methodology," in *Nutrition in the Prevention and Treatment of Disease*, 3rd ed., Academic Press, 2017, pp. 5–48.

[5] Y. Chen, M. Wang, and S. Liu, "API-based integration of food delivery platforms with nutrition databases: A design framework," *IEEE Access*, vol. 9, pp. 98765–98774, 2021.

[6] H. Lee and S. Park, "Towards multi-role user support in health applications: Challenges and opportunities," *Human-Centered Computing Journal*, vol. 11, no. 3, pp. 122–134, 2022.