International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Local Food Finder App System

Mr. J.Raghunath[1], Mr.P.Dineshkumar[2]

[1]*Student, III BCA, Department of Computer Applications, Dr.N.G.P. Arts and Science College, Coimbatore -641048.*
[2]*Assistant Professor, Department of Computer Applications, Dr.N.G.P. Arts and Science College, Coimbatore -641048.*
DOI : https://doi.org/10.55248/gengpi.6.0425.14174

ABSTRACT

The Local Food Finder app is a user-friendly platform designed to help food enthusiasts discover and explore nearby eateries, restaurants, and food stalls. By leveraging GPS technology, the app provides real-time location-based recommendations, personalized according to user preferences, dietary requirements, and reviews. Users can browse menus, read ratings, and even access exclusive deals or discounts. Whether searching for popular street food or hidden culinary gems, the app simplifies the dining experience by connecting users with diverse food options in their vicinity. With its intuitive interface and community-driven feedback system, the Local Food Finder app ensures an enjoyable and convenient food discovery journey.

Keywords: AI Driven, Openstreet Map, Open ERP,GPS Technology..

1. INTRODUCTION

In recent years, the demand for locally sourced food has increased as consumers seek fresh, sustainable, and high-quality food options. However, many people face challenges in discovering local food vendors, farmers' markets, and small-scale eateries in their vicinity. The Local Food Finder web application addresses this gap by providing a user-friendly digital platform that connects consumers with nearby food businesses, enabling easy access to fresh and locally available food.

This web-based platform is developed using HTML, CSS, and JavaScript for the frontend, ensuring an interactive and visually appealing user experience. The backend is powered by Flask (Python) and MySQL, facilitating efficient data management and secure transactions. The application offers location-based search functionality, enabling users to discover vendors and food products near them using Google Maps or OpenStreetMap integration. Customers can search by cuisine, price range, and vendor ratings, making it easier to find food that suits their preferences.

The system incorporates user authentication, allowing customers to create accounts, browse listings, and place orders. Vendors, on the other hand, can register their businesses, upload food listings, update availability, and manage orders through a dedicated dashboard. A built-in review and rating system enhances transparency, ensuring quality control and trust among users. Additionally, an admin panel is included to manage vendors, customers, and reported issues, ensuring smooth platform operation.

By promoting local food consumption, this application supports small businesses, reduces food miles, and encourages sustainable eating habits. The Local Food Finder not only benefits consumers by providing convenient access to fresh food but also empowers local vendors by offering them a digital presence. With potential future enhancements such as AI-driven recommendations and real-time order tracking, the platform aims to revolutionize the way people discover and purchase local food.

2.SYSTEM STUDY

2.1 EXISTING SYSTEM

The current landscape of local food discovery is fragmented and inadequate, primarily relying on outdated methods that fail to meet the nuanced needs of modern users. Traditional approaches, such as word-of-mouth, are inherently limited in reach and objectivity, while general online searches often yield overwhelming and poorly filtered results. General-purpose apps like Google Maps and food delivery platforms, though providing basic location and rating information, lack the ability to personalize recommendations based on individual dietary restrictions, taste preferences, or budget. This gap leaves smaller food vendors and street food stalls, vital components of local culinary scenes, significantly underrepresented, hindering the discovery of authentic and unique experiences..

2.2 PROBLEM IDENTIFICATION

In today's fast-paced world, consumers are increasingly seeking fresh, locally sourced food for health, sustainability, and economic reasons. However, the lack of a centralized platform for discovering and purchasing local food creates significant challenges for both consumers and vendors. Small-scale food businesses, such as local farmers, home-based cooks, and small eateries, often struggle to reach potential customers due to limited digital presence and visibility. They primarily rely on word-of-mouth, social media, or traditional marketplaces, which restricts their growth and customer base

2.3 PROPOSED SYSTEM

The proposed Local Food Finder App seeks to revolutionize local food discovery by transcending the limitations of current, generalized platforms. Its core strength lies in its specialized, location-based design, employing precise GPS integration to map not only established restaurants but also the often-missed culinary treasures of small food stalls and hidden local spots. By prioritizing personalization, the app moves beyond simple location listings, incorporating user-defined dietary restrictions, preferred cuisines, and real-time reviews to curate tailored recommendations. This ensures that each user's food exploration is uniquely aligned with their tastes and needs. Furthermore, the app enhances user engagement through dynamic features such as live availability updates, exclusive vendor deals, and direct communication channels, fostering a more interactive and connected food experience.

3.SOFTWARE DESCRIPTION

FRONTEND

HYPERTEXT MARKUP LANGUAGE INTRODUCTION TO HTML

HTML, which stands for Hypertext Markup Language, is the standard markup language for creating web pages. It provides the structure for web documents by using a system of tags and attributes to define elements within the page. These elements can include headings, paragraphs, images, links, forms, and more.

Working Process

HTML documents are text files that contain a series of elements enclosed in angle brackets (<

>). These elements are organized in a hierarchical structure, with the <html> element at the top, followed by <head> and <body> elements. The <head> section typically contains meta- information about the document, such as its title and links to external resources like stylesheets and scripts. The <body> section contains the content visible to the user.Within the <body> section, elements like <p> for paragraphs, <h1> to <h6> for headings,<img> for images, and <a> for links are used to create the desired layout and functionality of the webpage. Attributes can be added to these elements to provide additional information or modify their behavior.Once an HTML document is created, it can be viewed in a web browser, which interprets the HTML code and displays the content according to the specified structure and formatting. Additionally, HTML can be enhanced with the use of CSS (Cascading Style

Sheets) for styling and JavaScript for interactivity, allowing for more dynamic and visually appealing web pages .

CASCADING STYLE SHEETS INTRODUCTION TO CSS

CSS, short for Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML. It controls the layout, formatting, and appearance of web pages, allowing developers to define the visual aspects such as colors, fonts, spacing, and positioning.

Working Process

CSS works by targeting HTML elements and applying styling rules to them. These rules consist of selectors that identify which elements to style and declarations that specify the styling properties and values. Selectors can target elements based on their tag names, classes, IDs, attributes, or even their relationship with other elements in the document. Once selected, CSS properties such as color, font-size, margin, padding, and border can be applied to change the appearance of the elements.

CSS can be applied to HTML documents in three ways: inline styles, internal styles, and external stylesheets. Inline styles are applied directly within the HTML tags using the "style" attribute, internal styles are defined within the <style> element in the head section of the HTML document, and external stylesheets are separate CSS files linked to the HTML document using the <link> element. When a web browser renders an HTML document, it interprets the CSS rules and applies the specified styles to the corresponding elements, resulting in the desired visual presentation of the webpage. CSS also supports various features such as inheritance, specificity, and cascading, which enable developers to efficiently manage and organize their styles across multiple pages or components. In summary, CSS plays a crucial role in web development by allowing developers to control the appearance and layout of web pages, thus enhancing the user experience and creating visually appealing websites.

3.2.BACKEND PYTHON

INTRODUCTION

Python is a high-level, versatile programming language that has gained immense popularity among developers and businesses due to its simplicity, readability, and extensive standard library. It was created by Guido van Rossum and first released in 1991, and since then, it has become one of the most widely used programming languages for a wide range of applications. Python's design philosophy emphasizes code readability and maintainability, making it an excellent choice for beginners and experienced developers alike. Python is an interpreted language, meaning it does not require compilation before execution, making it easy to write, test, and debug code. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, giving developers the flexibility to approach problems from various angles. Python's extensive standard library provides a wealth of modules and packages that facilitate a wide range of tasks, from web development and data analysis to artificial intelligence and scientific computing. Python's simplicity and readability have made it a preferred choice for many domains, from web development using frameworks like Django and Flask to data analysis and machine learning with libraries such as NumPy, Pandas, and Tensor Flow. Its versatility, combined with a vibrant and active community of developers, has solidified Python's position as a top programming language, making it an excellent tool for creating software solutions across diverse industries and applications. Whether you're a beginner learning to program or an experienced developer seeking a powerful and efficient language, Python offers a rich ecosystem and a supportive community to help you succeed in your software development endeavors.

PYTHON IDLE

Python default IDE is known as IDLE (Integrated Development and Learning Environment). There is no need to install this IDE separately (via Python PIP) as it comes as default with Python installation. Although there are plenty of IDE which you can download separately on your system, still it is considered as a super choice for a newbie. IDLE comes by default on Windows and Mac but Linux user has to download it using the package manager. You have learned to write a Python code in Interactive environment, where you get the instant result of an expression. Now it’s time to write a few lines of code to solve a problem. You can write multiple lines of code in the Interactive environment as well, but it is not favoured because of the debugging reasons.

FEATURES OF PYTHON

Easy to Learn and Use

Python is easy to learn and use compared with other programming languages. It is developer- friendly and high level programming language.]

Interpreted Language

Python is an interpreted language because no need of compilation. This makes debugging easy and thus suitable for beginners.

Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

Free and Open Source

The Python interpreter is developed under an open-source license, making it free to install, use, and distribute.

Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

GUI Programming Support

Graphical user interfaces can be developed using Python.4.SYSTEM DESIGN

DESCRIPTION OF MODULES

User Module

The User Module allows consumers to search, browse, and order food from local vendors. Users can create accounts, manage their profiles, and access personalized recommendations based on location and preferences. The module provides a location-based search that enables users to discover nearby vendors and filter options based on cuisine, price, and ratings. Additionally, users can place orders, track their purchases, and provide reviews and ratings for vendors

Vendor Module

The Vendor Module is designed for local food sellers, farmers, and small eateries to manage their business effectively. Vendors can register, create profiles, list products, update availability, and manage orders. This module offers a dashboard where vendors can track customer interactions, order history, and sales trends.

Admin Module

The Admin Module oversees the platform’s overall functionality, ensuring a secure and smooth user experience. Administrators can monitor vendor activity, manage user accounts, verify vendor legitimacy, and handle reported issues. This module also allows admins to update the database, review customer feedback, and maintain platform security and data integrity.

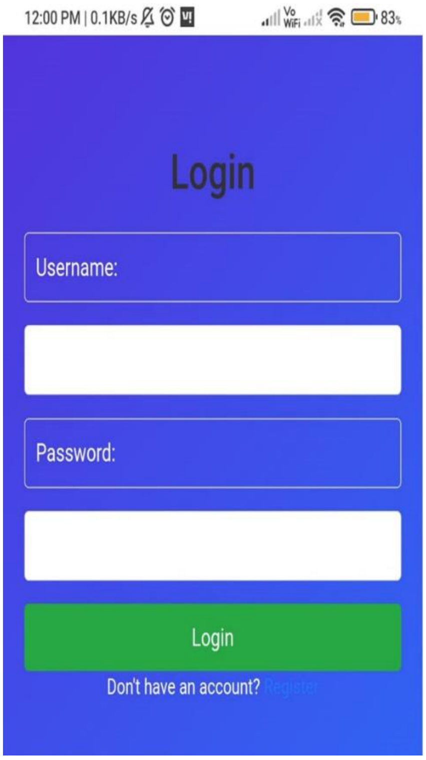
Search and Recommendation Module

The Search and Recommendation Module enhances food discovery by providing smart search filters and AI-driven recommendations. It integrates location-based services, enabling users to find vendors nearby with ease. Advanced algorithms suggest popular vendors, trending food items, and user-preferred cuisines based on past interactions.

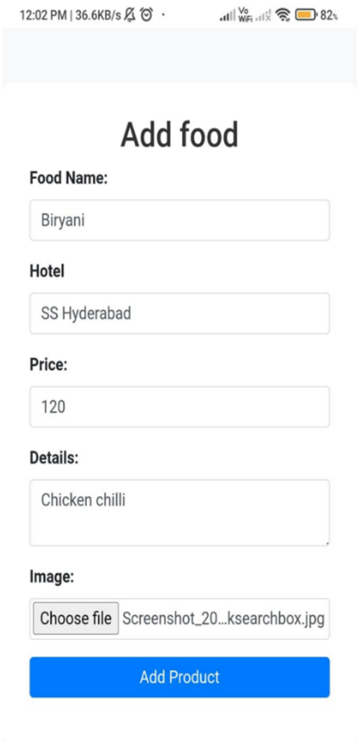
Order and Payment Module

The Order and Payment Module ensures secure and hassle-free transactions for users and vendors. Customers can place, track, and manage orders through a streamlined interface, while vendors can process and fulfill orders efficiently. The module integrates multiple payment options, including credit/debit cards, digital wallets, and cash-on-delivery. Secure encryption mechanisms protect sensitive payment information, ensuring safe transactions.

FORM DESIGN ADMIN LOGIN:



Don't have an account? Register



Add food

Food Name: Biryani

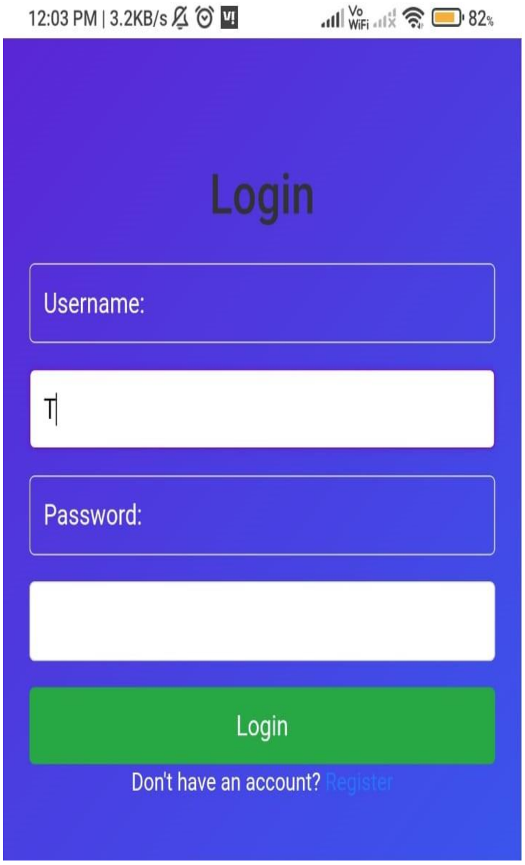
Hotel: SS Hyderabad

Price: 120

Details: Chicken chilli

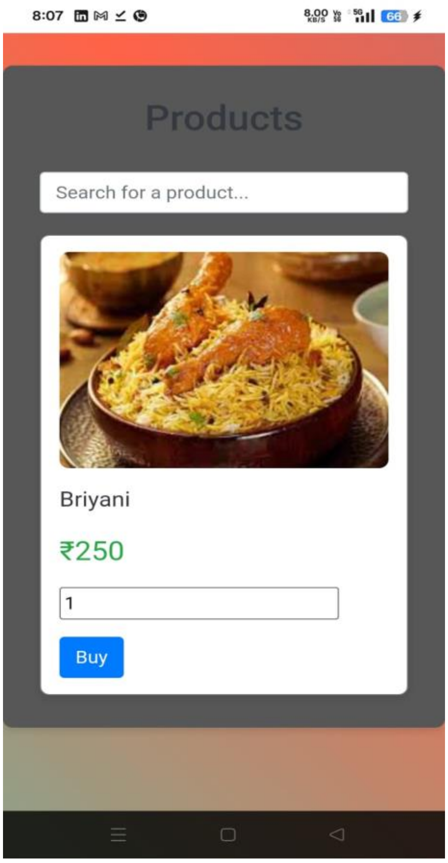
Image: Screenshot_20...ksearchbox.jpg

USER LOGIN:



Username: T

Don't have an account? Register

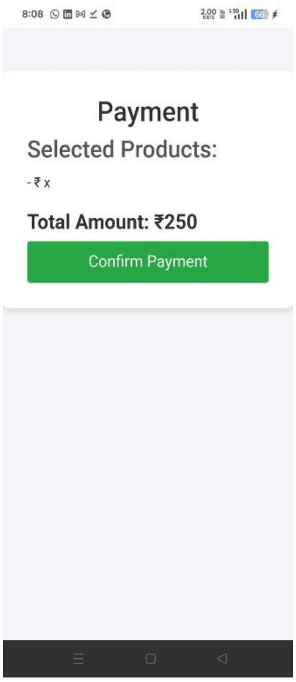


Briyani

₹250

1

PAYMENT MODULE:



Payment

Selected Products:

- ₹ x

Total Amount: ₹250

DATABASE DESIGN

The Local Food Finder web application utilizes a well-structured relational database using MySQL to manage user, vendor, order, and transaction data efficiently. The database is designed to ensure data integrity, scalability, and security while providing fast query processing for real- time updates. It consists of multiple interconnected tables that store and manage platform operations.

Key Database Tables:

Users Table – Stores user details such as User ID, Name, Email, Phone, Address, and Password (hashed) for authentication and profile management.

Vendors Table – Contains information about vendors, including Vendor ID, Name, Business Name, Location, Contact Details, and Business Type.

Food Items Table – Maintains records of food products with fields like Food ID, Vendor ID (foreign key), Name, Category, Price, Availability, and Description.

Orders Table – Tracks order transactions, including Order ID, User ID, Vendor ID, Order Status, Total Amount, and Timestamp.

Payments Table – Stores payment details such as Payment ID, Order ID (foreign key), Payment Mode, Transaction Status, and Timestamp.

Admin Table – Stores admin credentials with Admin ID, Username, Password (hashed), and Role Permissions for managing the platform.

Location Data Table – Saves latitude and longitude for vendors and users to enable geolocation- based searches.

TABLE STRUCTURE

Users Table

Stores information about registered users.

Column Name	Data Type	Constraints	Description
User_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for each user
Name	VARCHAR(100)	NOT NULL	Full name of the user
Email	VARCHAR(150)	UNIQUE, NOT NULL	User email (for login)
Phone	VARCHAR(15)	UNIQUE, NOT NULL	Contact number
Address	TEXT	NOT NULL	User’s address
Password	VARCHAR(255)	NOT NULL	Hashed password for security

Vendors Table

Stores information about local food vendors.

Column Name	Data Type	Constraints	Description
Vendor_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique vendor identifier
Business_name	VARCHAR(150)	NOT NULL	Name of the food business
Owner_name	VARCHAR(100)	NOT NULL	Owner's name
Email	VARCHAR(150)	UNIQUE, NOT NULL	Vendor's email

Column Name	Data Type	Constraints	Description
hone	VARCHAR(15)	UNIQUE, NOT NULL	Contact number
Location	TEXT	NOT NULL	Business location address
Password	VARCHAR(255)	NOT NULL	Hashed password for security

Food Items Table

Stores details about the food items listed by vendors.

Column Name	Data Type	Constraints	Description
Food_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for food items
Vendor_id	INT	FOREIGN KEY	Links food to the respective vendor
Name	VARCHAR(100)	NOT NULL	Food item name
Category	VARCHAR(50)	NOT NULL	Type of food (e.g., vegan, fast food)
Price	DECIMAL(10,2)	NOT NULL	Price of the food item
Availability	BOOLEAN	DEFAULT TRUE	Indicates if the item is available
Description	TEXT	NULL	Food item description

Orders Table

Column Name	Data Type	Constraints	Description
Order_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for orders
User_id	INT	FOREIGN KEY	Links to the user placing the order
Vendor_id	INT	FOREIGN KEY	Links to the vendor providing food
Order_status	ENUM('Pending', 'Processing', 'Completed', 'Cancelled')	DEFAULT 'Pending'	Tracks order progress
Total_price	DECIMAL(10,2)	NOT NULL	Total amount of the order
Timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Time order was placed

Payments Table

Column Name	Data Type	Constraints	Description
Payment_Id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for payments

Order_id	INT	FOREIGN KEY	Links to the respective order
Payment_mode	ENUM('Card', 'Wallet', 'Cash')	NOT NULL	Type of payment method used
Transaction_status	ENUM('Success', 'Failed', 'Pending')	NOT NULL	Status of the payment
Timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP	Time of transaction

Admin Table

Manages administrator accounts for platform control.

Column Name	Data Type	Constraints	Description
Admin_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for admins
Username	VARCHAR(50)	UNIQUE, NOT NULL	Admin login username
Password	VARCHAR(255)	NOT NULL	Hashed password for authentication
Role	ENUM('Super Admin', 'Moderator')	NOT NULL	Defines admin permissions

Location Data Table

Stores geolocation details for vendors and users.

Column Name	Data Type	Constraints	Description
Location_id	INT (AUTO_INCREMENT)	PRIMARY KEY	Unique identifier for location data
User_id	INT	FOREIGN KEY	Links to the user's location
Vendor_id	INT	FOREIGN KEY	Links to the vendor's location
Latitude	DECIMAL(10,8)	NOT NULL	Latitude coordinate
Longitude	DECIMAL(11,8)	NOT NULL	Longitude coordinate

Design Notations DATA FLOW DIAGRAM LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each

of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

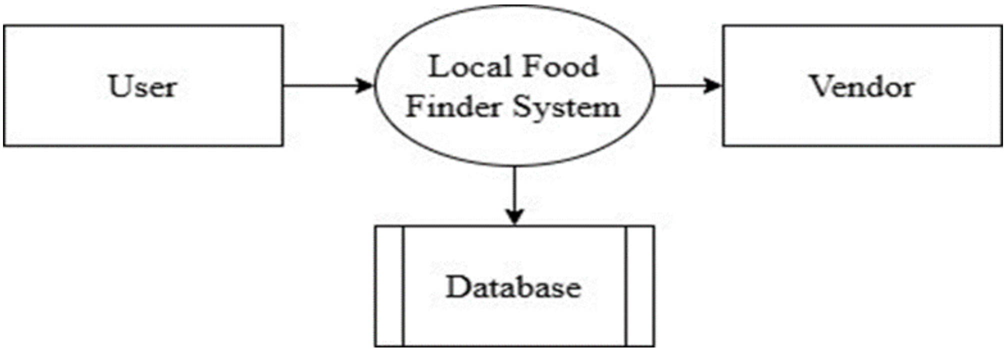


FIG 3.2.1- Data Flow Diagram level 0

LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper.

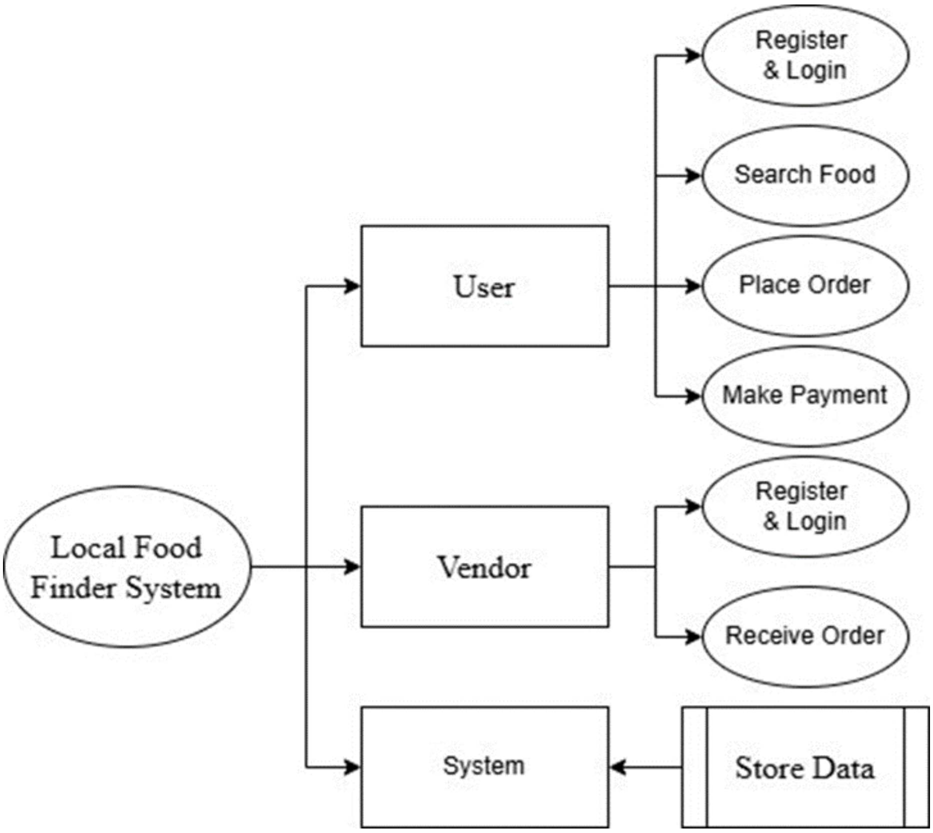


FIG 3.2.2- Data Flow Diagram level

5.TESTING AND IMPLEMENTATION

TESTING

Unit Testing

Unit testing ensures that each individual module of the Local Food Finder system functions correctly. Developers test components such as user authentication, order processing, and payment handling separately to detect errors at an early stage. Automated and manual unit tests validate input validation, database interactions, and API responses. This testing phase ensures that each feature works as expected before integration into the complete system.

Integration Testing

Integration testing examines how different modules interact. For instance, the system tests how the food search module connects with the database, how the order placement process interacts with payment processing, and how vendor management updates order status. This phase ensures seamless communication between components, eliminating any bugs in data exchange or API calls. It verifies that system modules function as a cohesive unit.

Functional Testing

Functional testing validates whether the system meets all requirements and user expectations. This involves testing key functionalities like food ordering, payment processing, vendor management, and user reviews. The test cases simulate real-world usage, ensuring that the system provides expected outputs for various user actions. Functional testing guarantees that the platform works as intended across all features.

Performance Testing

Performance testing checks how the system behaves under different loads. The Local Food Finder is tested for response time, order processing speed, and handling multiple users simultaneous

Load and stress testing identify bottlenecks, ensuring that the system remains responsive during

peak usage. This phase enhances scalability and ensures a smooth experience for users and vendors.

Security Testing

Security testing protects user data and transactions from vulnerabilities. It includes testing for SQL injection, cross-site scripting (XSS), and unauthorized access attempts. The system encrypts sensitive data, secures user authentication, and ensures safe payment transactions. Security testing guarantees that the Local Food Finder remains a secure platform for users, vendors, and administrators.

Usability Testing

Usability testing focuses on user experience (UX). Testers evaluate the platform's interface, navigation, and accessibility on various devices. Feedback from test users helps refine the design, ensuring that customers and vendors find the system easy to use. This phase ensures a smooth and intuitive experience, leading to higher user satisfaction.

5.2 IMPLEMENTATION

The implementation of the Local Food Finder web application follows a structured approach, ensuring efficient development, deployment, and maintenance. The system is built using HTML, CSS, and JavaScript for the frontend, Python (Flask) for the backend, and MySQL for database management. The implementation is divided into multiple stages:

Frontend Development

The frontend is designed using HTML, CSS, and JavaScript, ensuring a responsive and user- friendly interface. Features such as food search, order placement, and payment integration are implemented using JavaScript frameworks. CSS ensures a visually appealing design, while JavaScript enhances interactivity and smooth navigation.

Backend Development

The backend, developed in Flask (Python), handles server-side logic, including user authentication, order management, and database interactions. Flask routes manage requests from users and vendors, processing data and sending responses efficiently. API endpoints are created to facilitate seamless data exchange between the frontend and backend.

Database Integration

The system's database is designed using MySQL, storing user details, vendor information, food listings, orders, payments, and reviews. Proper indexing and query optimization ensure fast data retrieval. The database schema is structured to maintain data consistency and security.

Deployment and Hosting

Once development is complete, the application is deployed on a cloud platform such as AWS, Heroku, or DigitalOcean. The deployment involves setting up a secure server, configuring the database, and ensuring system scalability. Secure Socket Layer (SSL) encryption is implemented to protect user data.

Testing and Optimization

Before launch, rigorous testing is conducted, including unit testing, integration testing, and performance testing. Feedback is collected from test users, and necessary improvements are made. Optimization techniques, such as caching and database indexing, are applied to enhance performance.6.

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The Local Food Finder system offers a comprehensive, user-friendly solution to bridge the gap between consumers and local food vendors. Through its intuitive interface and robust backend, it enables seamless food discovery, order placement, and payment processing. By integrating efficient database management and secure transaction handling, the platform ensures a smooth experience for both users and vendors. The system's design incorporates features like real-time updates, secure authentication, and review systems, ensuring that customers receive timely and quality services. Moving forward, additional features and enhancements can be introduced to cater to emerging market needs, ensuring that Local Food Finder remains relevant and valuable to its users. In summary, the Local Food Finder not only improves the local food experience but also sets the stage for future advancements in food delivery technology

FUTURE ENHANCEMENT

The Local Food Finder system can be further enhanced to meet evolving user needs and industry trends. One potential enhancement is the integration of AI-powered recommendations, which could analyze user preferences and suggest personalized food options based on past orders, reviews, and dietary restrictions. Additionally, incorporating real-time delivery tracking would allow users to track the status of their orders and receive precise ETAs. Another valuable improvement could be the addition of a loyalty rewards program that incentivizes users to make repeat orders, boosting customer retention. Expanding the platform to include features like multi-language support and integration with social media for easy sharing could attract a broader audience. Furthermore, integrating sustainability features, such as eco-friendly packaging options and promoting local farm-to-table connections, would align the platform with growing environmental consciousness. The continuous optimization of the platform's performance and scalability would ensure that it can handle increasing user traffic and demand as the business grows. Through these enhancements, Local Food Finder can remain competitive and continue providing a valuable service to both consumers and vendors.

BIBLIOGRAPHY AND REFERENCE

BOOK REFERENCES

Robert W. Sebesta. *Programming the World Wide Web*. 7th Edition, Pearson, 2019.

Ethan Brown. *Flask Web Development: Developing Web Applications with Python*. 2nd Edition, O'Reilly Media, 2017.

Paul Deitel & Harvey Deitel. *Internet & World Wide Web: How to Program*. 6th Edition, Pearson, 2017.

WEB REFERENCES

Flask Documentation

W3Schools - Web Development Tutorials

MySQL Documentation