

TADKADISH

A PROJECT REPORT

Submitted in partial fulfilment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

By

Ms. HARSHADA MANOJ KOTAWADEKAR

Client Certificate

Tadka

To whomsoever it may concern

Respected Sir,

I am writing to confirm that “Miss Harshada Manoj Kotawadekar” has been actively engaged with our team for the development of the “TadkaDish” an Online Food Delivery System web application over the past 2 months.

During this period, she has shown great interest in understanding the intricacies of our system and has been a regular visitor to our development center. The information and insights gathered from our organization have been crucial in shaping the project.

Please note that while the data and design presented in this report are based on our system, they have been adapted and tailored to fit the project report format.

Regards,

(Client signature)

ABSTRACT

The “TadkaDish” is an Online Food Delivery System is an innovative web and mobile-based platform designed to connect customers with restaurants for quick, easy, and efficient food delivery. The system provides users with the ability to browse various restaurants, explore diverse menus, place orders, and receive food at their doorstep. Users can customize their orders, view estimated delivery times, and track their deliveries in real-time.

The platform offers multiple secure payment options, including credit/debit cards, and cash on delivery, ensuring seamless transactions. For restaurants, the system provides a management interface to handle orders, update menus, and monitor deliveries. The system is equipped with features such as user profiles, order history, ratings, and reviews, which help customers make informed decisions.

Additionally, the delivery process is streamlined, allowing drivers to receive real-time notifications and follow optimized routes for faster deliveries. The Online Food Delivery System aims to improve convenience, speed, and customer satisfaction while empowering restaurants to expand their reach and enhance their operational efficiency.

SYNOPSIS

Introduction:

The Online Food Delivery System is a digital platform that facilitates the delivery of food from restaurants to customers through an efficient, user-friendly interface. With the rise of online services and the increasing demand for convenience, food delivery platforms have become a vital part of the food industry.

This system offers customers the ability to browse menus, place orders, pay online, and track deliveries from various restaurants, all from the comfort of their homes or workplaces. By integrating advanced technologies, the platform aims to provide fast, reliable, and seamless food delivery services, thus improving customer satisfaction and supporting the growth of local restaurants.

Current System:

In the current food delivery landscape, many restaurants rely on traditional methods of receiving orders, such as phone calls or manual order-taking. Orders are often processed manually by the restaurant staff, leading to potential errors, delays, and inefficiencies. Additionally, customers have limited options to track the progress of their orders, and payment methods are often restricted to cash on delivery. The delivery process can also face challenges like delays in food preparation and navigation, resulting in longer wait times. Furthermore, there is minimal feedback collection or review mechanisms to ensure the quality of service.

Problems faced in Current System:

- Inefficiency and Errors
- Limited Payment Options
- Long Wait Times
- Limited Communication
- Lack of Order Customization

Proposed System:

The proposed Online Food Delivery System aims to modernize the way food orders are placed, processed, and delivered by introducing an integrated, digital platform. The system will allow customers to browse restaurants, view menus, customize their orders, and make secure payments through various options, including digital wallets, credit cards, and online banking. The platform will offer real-time tracking, enabling customers to follow their order's progress from preparation to delivery.

Additionally, the system will include an efficient communication system between customers, restaurants, and delivery drivers to ensure smooth order processing. With an intuitive user interface, the system will also allow users to rate and review their experiences, helping restaurants to improve service quality. By automating key processes, such as order management and delivery tracking, the system will reduce errors, increase speed, and enhance customer satisfaction.

The proposed system will provide a comprehensive solution for both customers and restaurants, offering a faster, more secure, and customer-centric food delivery experience.

Software Requirements:

- Operating System: Windows 8 and above.
- Backend: MySQL or MongoDB.
- Frontend: HTML5, CSS, JavaScript.

Hardware Requirements:

- Memory: - Minimum 4 GB RAM and above.
- Hard Disk: - minimum 128 GB and above.
- Processor: - Maximum 64bit Processor.

TABLE OF CONTENTS

SR NO.	TITLE	PAGE NO.
1.	CHAPTER 1: INTRODUCTION	1 - 4
	1.1 Background	01
	1.2 Feasibility Study	01
	1.3 Project objective	03
	1.4 Purpose, scope, applicability	04
2.	CHAPTER 2: SURVEY OF TECHNOLOGY	5 - 6
	2.1 Tools and Technology	05
3.	CHAPTER 3: REQUIREMENTS AND ANALYSIS	7 - 12
	3.1 Problem definition	07
	3.2 Requirements Specification	07
	3.3 Planning and scheduling (SDLC, GANTT CHART)	07
	3.4 Hardware and software requirements	11
4.	CHAPTER 4: SYSTEM DESIGN	13 - 21
	4.1 Basic Module	13
	4.2 Database Design	13
	4.3 Data Integrity and Constraints	15
	4.4 Logic Diagram	15

	4.5	Security Issues	21
	4.6	Test Case Design	21
5.		CHAPTER 5: Implementation and Testing	22 - 29
	5.1	Testing Approach	22
	5.2	Coding Details and Code Efficiency	24
6.		CHAPTER 6: Results	30 - 37
7.		CHAPTER 7: CONCLUSION AND FUTURE SCOPE	38 - 39
	7.1	Advantages over Current System	38
	7.2	Future Enhancement	38
	7.3	Conclusion	39
	7.4	System Maintenance	39
8.		CHAPTER 8: REFERENCE	40

1. INTRODUCTION

1.1 BACKGROUND: -

The Online Food Delivery System has transformed the way people order and receive food, driven by advancements in technology, e-commerce, and mobile platforms. Traditionally, customers relied on in-person visits or phone orders, which were often time-consuming and error-prone. However, with the rise of internet-based platforms and smartphones, food delivery services became more efficient, offering customers the ability to browse menus, place orders, and track deliveries with ease.

The growth of major players like Uber Eats, DoorDash, and Zomato, alongside improvements in logistics and payment methods, has further fueled the industry. The widespread use of mobile phones, high-speed internet, and changing consumer lifestyles have contributed to the rapid expansion of online food delivery, creating a dynamic market that continues to innovate and meet the needs of modern consumers.

1.2 FEASIBILITY STUDY: -

Feasibility Study is a critical step to assess whether a project, like developing a restaurant management system (RMS), is viable, practical, and worthwhile. The study evaluates technical, economic, legal, operational, and scheduling aspects of the system, considering the potential costs and benefits.

There are various measures of feasibility that helps to decide whether a particular project is feasible or not. These measures include: -

1) Technical Feasibility

Technical Feasibility assesses whether the technology and technical resources required for a project are available and capable of meeting the project's needs. It involves evaluating the suitability of hardware, software, and system integration to ensure the project can be executed successfully.

- This Web Application makes use of HTML, CSS, and JavaScript.
- Proposed system makes use of user-friendly interfaces that allow restaurant staff and customers to interact with the system seamlessly

- Server-Side Programming is using PHP/MySQL to handle logic, user requests, and database communication.
- Also the proposed system makes use of RESTful API for handling communication between the frontend (web and mobile apps) and the backend.
- MySQL for structured data like customer details, order history, staff schedules, and inventory.
- Regular software updates, ongoing system monitoring, and customer support for troubleshooting.
- Track system activities and user actions for security and accountability.

2) Economic feasibility

Economic feasibility assesses whether a project is financially viable, examining the costs and benefits to determine if the project is a worthwhile investment. It involves evaluating the cost of development, implementation, and maintenance against the expected benefits and potential returns.

- Order Accuracy: Reduced human error in taking orders, resulting in fewer mistakes and quicker order fulfillment. Estimated reduction in errors: 20-30%.
- Time Savings: Quicker order-taking and processing lead to faster table turnover. This can increase daily capacity, reducing wait times, and improving overall efficiency.
- Reduced Inventory Losses: Real-time tracking of inventory and automated stock alerts reduce wastage and over-ordering. Estimated savings on inventory management: 10-15%.
- A faster and smoother process improves customer satisfaction, leading to higher tips and repeat customers. Estimated revenue boost from faster service: 10-20%.
- The project is totally being created on Open Source platform and MySQL so it does not require much cost. Hardware which are going to be used are already available to us.

- The project being developed also requires very less maintenance cost. The only valuable things required in building this project is time and Labour. Hence the project is totally Economic Feasible.

3) Operational feasibility

Operational feasibility assesses how well a proposed project can be integrated into current operations and how easily it can be managed and maintained once deployed. It focuses on the system's practicality, ease of use, maintenance requirements, and overall impact on operations.

- **Ease of Use:** The platform will have an intuitive interface for customers, restaurants, and delivery drivers, ensuring easy navigation and seamless interactions.
- **Real-Time Operations:** Orders will be processed instantly, with real-time tracking for customers and optimized delivery assignments for drivers.
- **Customer Experience:** The system will ensure accurate orders, fast delivery, and responsive customer support to enhance satisfaction.
- **Scalability:** The platform can easily expand to new cities, handle increased user demand, and integrate new features as the business grows.
- **Maintenance:** Regular system updates, dedicated technical support, and continuous performance monitoring will ensure smooth operations.
- **Performance:** The system will prioritize speed, reliability, and high availability, handling traffic efficiently during peak periods without compromising service quality.

4) Schedule feasibility

A schedule feasibility study assesses whether the project can be completed within the required time frame, considering the time needed for development, implementation, testing, and full deployment.

- Maintain a buffer of 1-2 weeks for unexpected delays in system integration or hardware delivery.
- Estimate the development and implementation timeline for the online old age home management system.
- The Online Food Delivery system can be successfully developed, tested, and deployed within 3 months, meeting the restaurant's operational needs and allowing for minimal disruption.

5) Legal feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like data protection acts etc. it saves considerable amount of time and effort by learning that their project was not feasible right from the beginning.

Though our system is getting developed on open source and free platform but all of these platforms are legal and provide all kind of support required to their users.

All platforms which we are using are very renowned and are used on Industry level by big Association. We have made sure that the project undertaken will meet all legal and ethical requirements before the project is on table.

1.3 PROJECT OBJECTIVE: -

- **Customer Convenience:** Create an easy-to-use platform for seamless ordering, secure payments, and real-time tracking.
- **Efficient Delivery Management:** Optimize delivery assignments and ensure timely, accurate deliveries by leveraging proximity-based routing.
- **Restaurant Integration:** Enable restaurants to manage orders, improve operations, and expand their customer reach.
- **Scalability & Maintenance:** Design the system for growth, supporting new locations and easy updates, while ensuring reliable performance.
- **Enhanced Customer Experience:** Focus on fast delivery, exceptional customer support, and overall satisfaction for users.

1.4.1 PURPOSE: -

The purpose of the online food delivery system is to bridge the gap between customers and local restaurants by offering a convenient, efficient platform for placing food orders. It aims to provide customers with a seamless experience, allowing them to browse restaurant menus, place orders, and track deliveries in real time, all from the comfort of their homes.

For restaurants, the system serves as a tool to expand their reach, streamline order management, and enhance operational efficiency. By connecting with delivery drivers through the platform, restaurants can provide timely service to a larger customer base, boosting sales and improving customer satisfaction.

1.4.2 SCOPE: -

The scope of this project is to design and develop an **Online Food Delivery System**, which provides an easy, efficient, and seamless experience for customers to order food online.

- **User Registration and Login:** The system will allow customers, restaurant owners, and delivery personnel to register, log in, and manage their profiles.
- **Menu Display:** Restaurants will be able to list their available food items, including details like description, price, and photos.
- **Order Placement and Customization:** Customers can place food orders, customize items, and add special instructions.
- **Payment Gateway Integration:** The system will integrate secure online payment methods (such as credit/debit cards, etc.) for transactions.
- **Reviews and Ratings:** Customers can rate and review restaurants and delivery services based on their experience.
- **Admin Panel:** The admin panel will allow for managing users (customers, restaurants, and delivery personnel), monitoring orders, and generating reports.

1.4.3 APPLICABILITY: -

The Online Food Delivery System is applicable to both urban and suburban areas, where a high demand for food delivery services exists. It caters to customers, restaurant owners, and delivery personnel, ensuring a seamless interaction for all users involved.

This system is highly applicable for businesses looking to expand their customer base through digital platforms. It also facilitates quick, secure, and efficient food delivery, enhancing customer satisfaction and improving service quality.

1.4.4 ACHIEVEMENTS: -

After working on this project, I got to know how a project is managed and completed successfully before the deadline to meet its goals. I also got to know new information related to the increasing technology and software. Objectives stated of this project are fulfilling the client needs.

2. Survey of Technology

A technological survey is a method of gathering insights on software, tools, and relevant tech skills in a particular context. The tools and technologies used for developing the project are given below:

➤ **Tools: -**

- 1) Visual Studio code
- 2) Xampp Server

1) Visual Studio Code: -

Visual Studio Code, also commonly referred to as VS Code is a source code editor made by Microsoft with the Electron framework for windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages including HTML, CSS, JavaScript, PHP, C, C#, JAVA and many more.

2) XAMPP Server: -

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

➤ TECHNOLOGY:

- 1) PHP:** - PHP is a powerful server – side scripting language designed for creating dynamic and interactive websites. It is perfectly suited for web development and can be embedded directly into the HTML code. It can be deployed on most web servers and on almost every operating system and platform free of charge. Figure 12: Waterfall method (4.3.1) 25 Server support PHP generally runs on a web server like Apache. Database support PHP supports many databases like MySQL, Oracle, Solid, PostgreSQL, etc. Most popular used is MySQL.
- 2) HTML:** - HTML stands for Hypertext Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. HTML is a markup language that is used by the browser to manipulate text, images, and other content to display it in the required format. It is a simple markup language, and its implementation is easy. It is used to create a website structure. It helps in developing fundamentals about web programming.
- 3) CSS:** - CSS (Cascading Style Sheets) is a stylesheet language used to design webpages to make it attractive. The reason for using CSS is to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. There are three types of CSS which are given below:
Inline CSS: In Inline CSS, we add the style to the tags using the “style” attribute inside the tag which we want to design.
Internal or Embedded CSS: Internal CSS allows us to style our page by adding the tag. Inside the External CSS lets us add style to our HTML page externally. We can add our styles in a different file with extension.css and link this page to our HTML page.
- 4) JavaScript:** - “JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow clientside script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.”

3. REQUIREMENTS AND ANALYSIS

3.1 PROBLEM DEFINITION: -

The increasing demand for convenience and time-saving solutions has led to a growing need for food delivery services. In urban areas, people often face challenges in finding reliable, quick, and efficient ways to order food from their preferred restaurants. Additionally, traditional food ordering methods—such as phone calls or in-person visits—can be time-consuming, inefficient, and prone to errors.

The lack of a streamlined and automated food delivery platform results in customer dissatisfaction due to long wait times, incorrect orders, and limited payment options. Furthermore, restaurants often struggle with managing customer orders, deliveries, and payments manually. This problem highlights the need for a comprehensive **Online Food Delivery System** that integrates ordering, payment, and delivery management in a user-friendly, efficient manner.

3.2 REQUIREMENTS SPECIFICATION: -

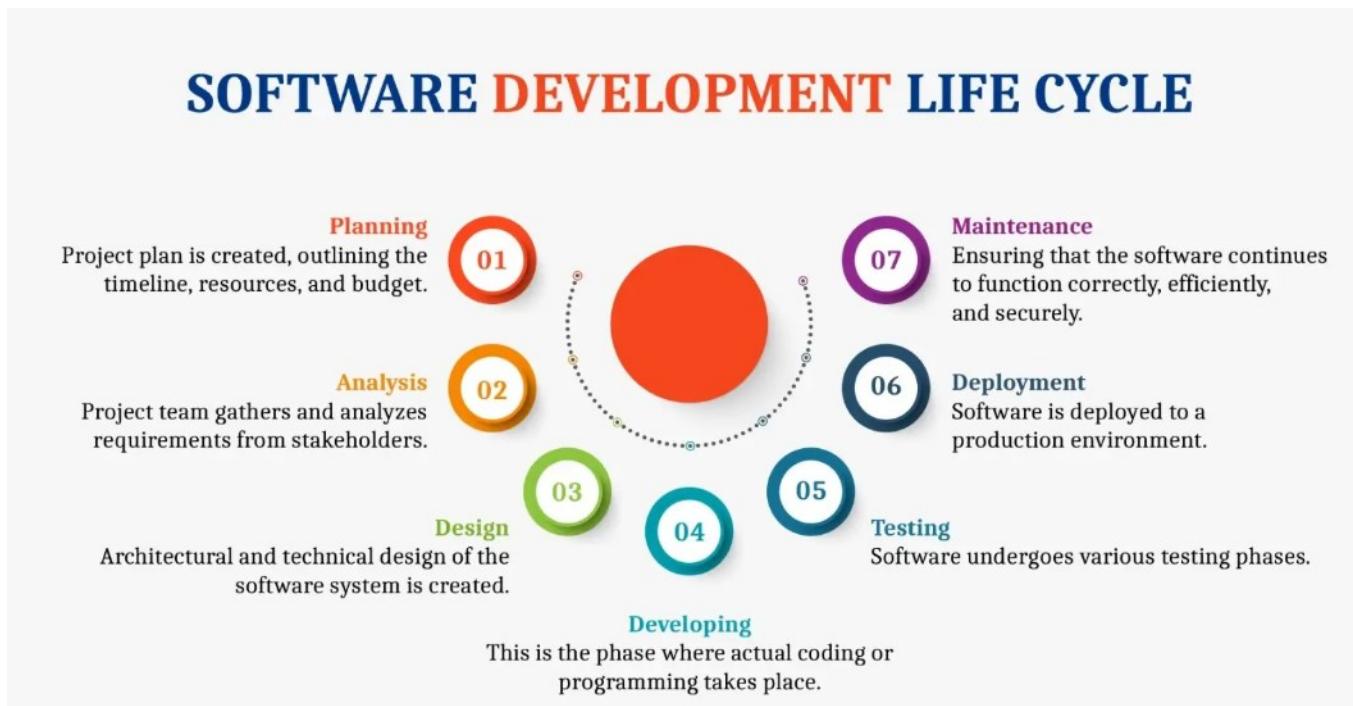
The **Online Food Delivery System** must include key features such as user registration, menu management, search and filter options, seamless order placement, and real-time order tracking for both customers and delivery personnel. It should also integrate secure payment gateways and provide an intuitive interface for all users.

Non-functional requirements focus on security, scalability, usability, and performance, ensuring the system handles growing demands efficiently. The system should be compatible with both web and mobile platforms, utilizing a relational database to manage user and order data.

3.3 Planning and Scheduling

3.3.1 Software Development Life Cycle (SDLC): - Software Development Life Cycle (SDLC) is a process used to design, develop and test high quality software. The SDLC aims to produce high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

The following figure is a graphical representation of the various stages of a typical SDLC.



For development of the Online Food Delivery System we are going to use the WATERFALL MODEL.

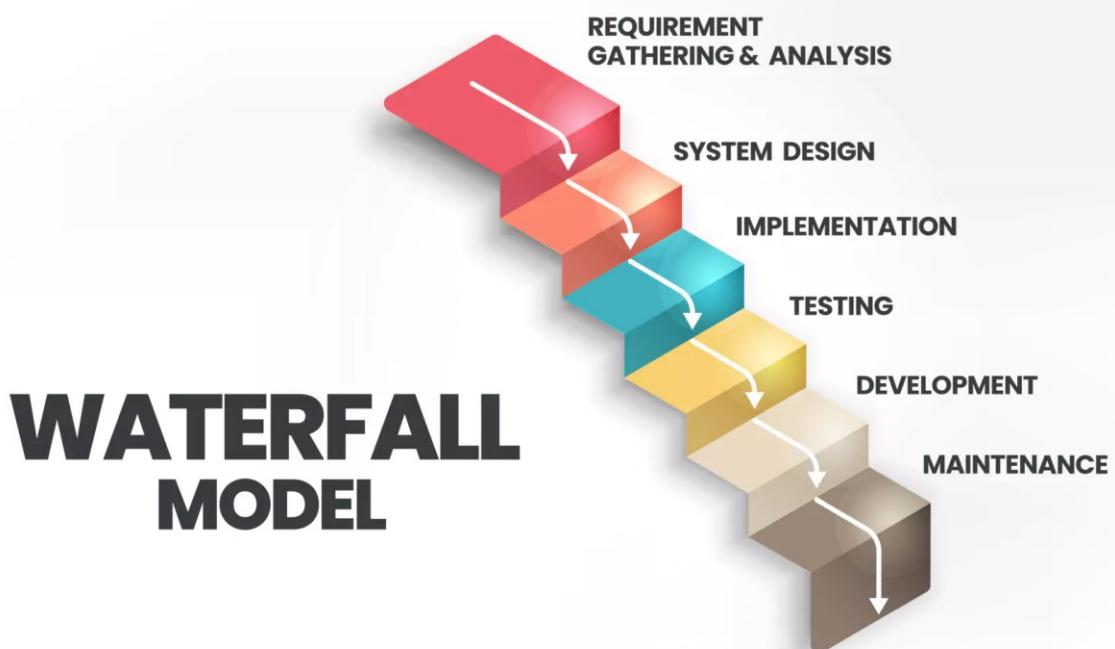
Waterfall Model: -

The waterfall model is a project management methodology based on a sequential design process in which progress is seen as flowing downwards (like a waterfall) through the different phases.

This model is simple and easy to understand and use. It is easy to manage due to the rigidity of the model. In this model phases are processed and completed one at a time.

The waterfall model involves a sequential approach to software development, where each phase of the project is completed before moving on to the next one.

The waterfall model depended on documentation to ensure that the project is well-defined and the project team is working towards a clear set of goals.



The Phases of Waterfall Model: -

1.3.1 Requirement Definition: -

- For the development of the proposed system the requirement of the client is: - Client wants a Login page for User as well as for Admin also.
- Clients want to update and delete students whenever they want to on their admin page.
- Client needs features for implementing the different schedule, offers for user and apply them.

1.3.2 System and Software Design: -

- This phase allocates the requirements to either hardware or software system by establishing an overall system architecture.
- For designing the system, we required Html, CSS, JavaScript languages, Visual Studio Code software for frontend and XAMPP server for both PHP & MySQL (database) for backend.
- And as a part of software design there will be modules in the system like:-
 1. Login Module
 2. Admin Module
 3. User Module

4. Homepage Module
5. Update/Delete Module

1.3.3 Implementation and Unit Testing: - During the implementation phase, the software design is realized as a set of programs or program unit. Unit testing involves each unit that needs its specifications.

1.3.4 Integration and System Testing: - In this phase the individuals program units are integrated and tested as a complete system to ensure that the requirements meet its specifications. After this phase the system is delivered to the client.

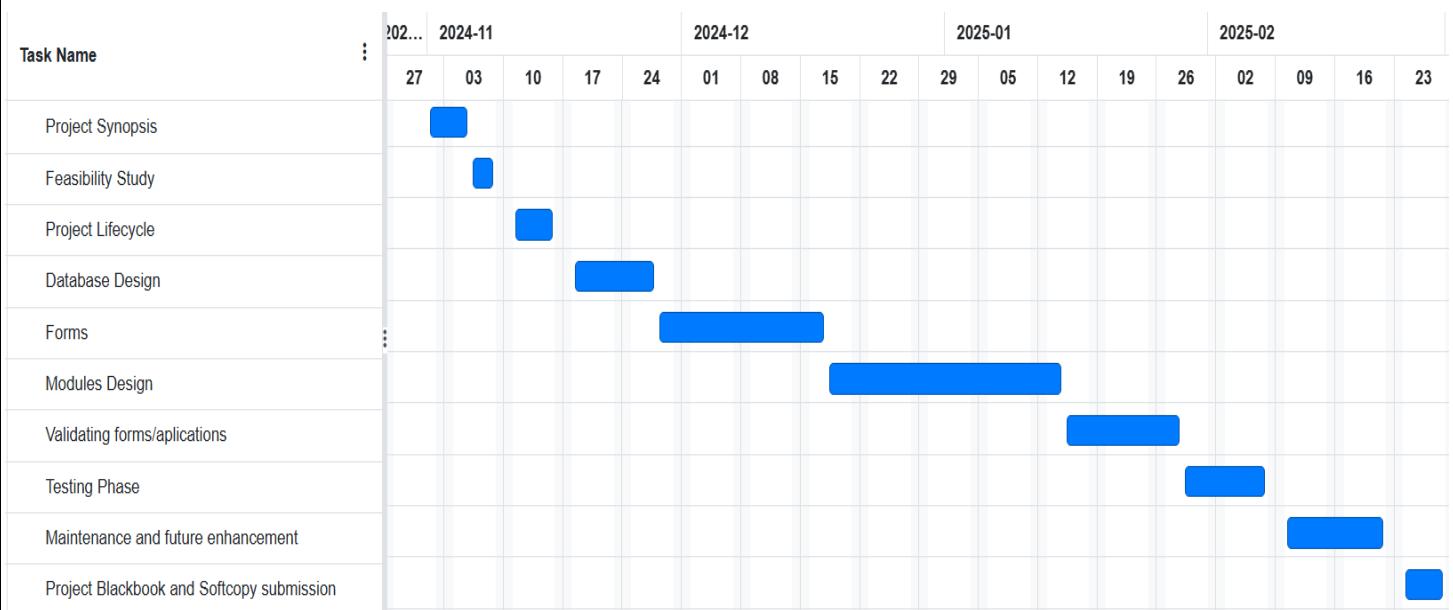
1.3.5 Operation and Maintenance: - This is the longest phase in SDLC. The system is installed on the client machine and put to practical use. Maintenance involves correcting the errors and providing the better software for the client.

3.3.2 Gantt Chart: -

A Gantt chart is a horizontal bar chart developed as a production control tool. Gantt charts are useful for planning and scheduling projects. They help you assess how long a project should take, determine the resources needed, and plan the order in which you'll complete tasks. They are also helpful for managing the dependencies between tasks.

A Gantt chart is constructed with horizontal axis representing the total time span of the project, broken down into increments (for example days, weeks, or months) and vertical axis representing the tasks that make up the project.

The primary benefit of a Gantt chart is that it provides a clear overview of a project's workflow. It helps project managers see how tasks are related, whether they overlap, or if they depend on the completion of others. This visibility aids in resource allocation and identifying potential bottlenecks or delays early in the process.



3.4 Software and Hardware Requirement: -

Software Requirements: -

- Operating System: Windows 8 and above.
- Backend: MySQL or MongoDB.
- Frontend: HTML5, CSS, JavaScript.
- Platform: Visual Studio code

Hardware Requirements: -

- Disk-space: Minimum 5GB or more
- RAM: Minimum 1GB
- Processor: Minimum 32-bit processor

3.5 Event Table: -

Event	Trigger	Source	Activity	Response	Destination
Customer places an order	Customer submits order online	Customer	Validate order details, calculate total price	Display order confirmation	Customer
Chef prepares the order	Order is received in the kitchen	Order management system	Check inventory, start cooking	Update order status to "In Progress"	Order management system
Payment processed	Customer chooses to pay online	Payment gateway	Verify payment details, process payment	Update order status to "Paid"	Order management system
Order delivered	Delivery staff picks up the order	Delivery staff	Track delivery status, confirm delivery	Update order status to "Delivered"	Customer, Order management system
Inventory updated	Chef uses ingredients from inventory	Kitchen staff	Deduct used items from inventory	Update inventory levels	Inventory management system
Customer feedback submitted	Customer completes order and provides feedback	Customer	Collect feedback, analyze satisfaction levels	Store feedback for analysis	Customer feedback system
Table reservation made	Customer reserves a table online	Customer	Validate reservation details, update reservation calendar	Send reservation confirmation	Customer, Reservation system
Employee shifts scheduled	Manager schedules shifts for staff	Manager	Assign shifts, notify employees	Update employee schedule	Employee management system
Daily sales report generated	End of day	Sales system	Aggregate sales data, generate report	Display daily sales report	Manager

4. SYSTEM DESIGN

4.1 Basic Modules: -

- **Login module** – The login module validates the email and password of the residents after successfully signing in. By login module only the authorized persons can access the system, it will alert for the invalid credentials entered by user.
- **Admin module** - Admin logs into the system and validates the admin email and password notification. Admin can register donor and register resident information.
- **Update/Delete module** - In Update/Delete module admin can update and delete records related to residents and donor present on dashboard as well as in database.

4.2 DATABASE DESIGN: -

A)Customer Details

1) Users details table

Sr. No	Field Name	Data Type	Size	Description
1	Id	Int	11	Primary Key
2	Full Name	Varchar	255	Not Null
3	Email	Varchar	255	Not Null
4	Password	Varchar	255	Not Null

2) Contact Form Details

Sr. No	Field Name	Data Type	Size	Description
1	Id	Int	11	Primary Key
2	Name	Varchaar	255	Not Null
3	Email	Varchaar	255	Unique Key

4	Message	Text	-	Not Null
5	Created at	Timestamp	-	Not Null

3) Orders

Sr. No	Field Name	Data Type	Size	Description
1	Order id	Int	11	Primary Key
2	Food items	Text	-	Not Null
3	Total price	Decimal	10,2	Not Null
4	Order date	Timestamp	-	Not Null
5	Name	Varchar	255	Not Null
6	Email	Varchar	255	Unique Key
7	Address	Text	-	Not Null
8	Status	Varchar	100	Not Null

4) Payment

Sr. No	Field Name	Data Type	Size	Description
1	Payment id	Int	11	Primary key
2	Order id	Int	11	Primary key
3	Payment amount	Decimal	10,2	Not Null
4	Payment method	Varchar	50	Not Null
5	Payment status	Varchar	50	Not Null
6	Payment date	Timestamp	-	Not Null

B) Admin Details Table:

Sr. No	Field Name	Data Type	Size	Description
1	Admin id	Int	20	Primary key
2	Admin name	Varchar	50	Not Null
3	Admin email	Varchar	50	Not Null
4	Admin username	Varchar	50	Not Null
5	Admin password	Varchar	50	Not Null

4.3 DATA INTEGRITY AND CONSTRAINT: -

Data integrity ensures that all information in the online food delivery system is accurate, consistent, and reliable. Maintaining integrity is vital for customer trust, operational efficiency, and financial accuracy.

For instance, when a customer places an order, ensuring the correct order details like customer ID, restaurant, items, and quantities are stored correctly is key to delivering the right experience.

Referential integrity ensures that relationships between different entities in the database are maintained. In an online food delivery system, this means that data like customer orders, delivery addresses, and payment information must be consistently linked.

A foreign key constraint ensures that orders can only be made by existing customers, and payments must correspond to valid orders, preventing data inconsistencies.

Constraints play a crucial role in preserving data integrity by enforcing rules at the database level. Primary key constraints ensure that each entity, such as an order or customer, has a unique identifier.

Foreign keys establish connections between related tables, while unique constraints prevent duplicate records, like ensuring each customer email is unique, and the not-null constraint ensures critical fields, such as payment status, are always filled.

4.4 LOGIC DIAGRAM: -

A logic diagram for an online food delivery system visually represents the flow of operations, highlighting key processes such as order placement, payment, and delivery. It begins with the customer browsing the menu, selecting items, and placing an order.

The system then verifies the payment details and confirms the availability of the items before passing the order to the restaurant and the delivery team for fulfillment.

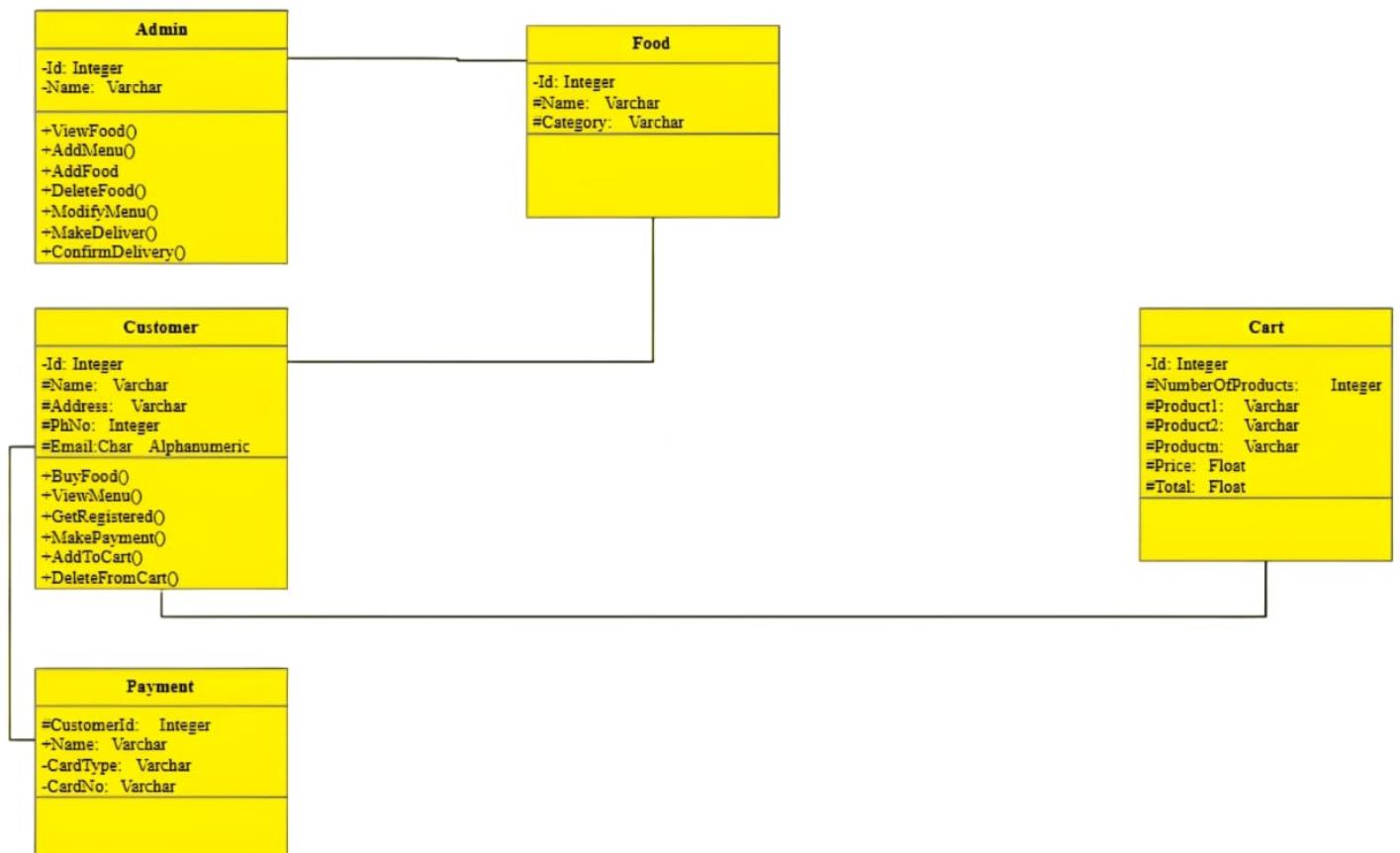
The diagram also incorporates important decision points, like order confirmation, payment success, and delivery tracking. Each stage ensures data flows correctly between the customer, restaurant, and delivery service, maintaining consistency and clarity.

By illustrating these processes, the logic diagram helps streamline system functions and ensures smooth interactions across different system components.

4.4.1 CLASS DIAGRAM: -

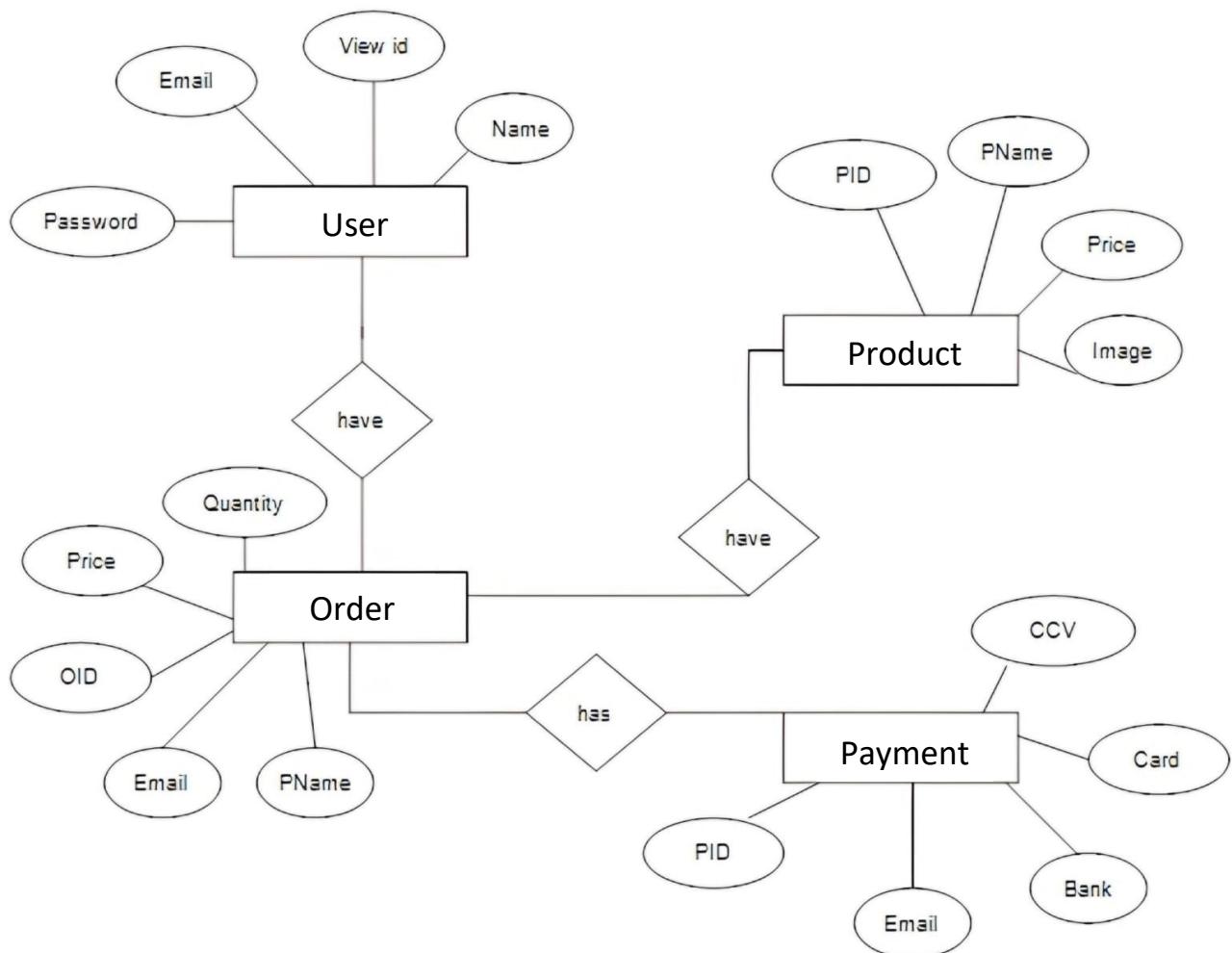
The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and it may inherit from other classes. A class diagram is used to visualize, describe, document various aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.



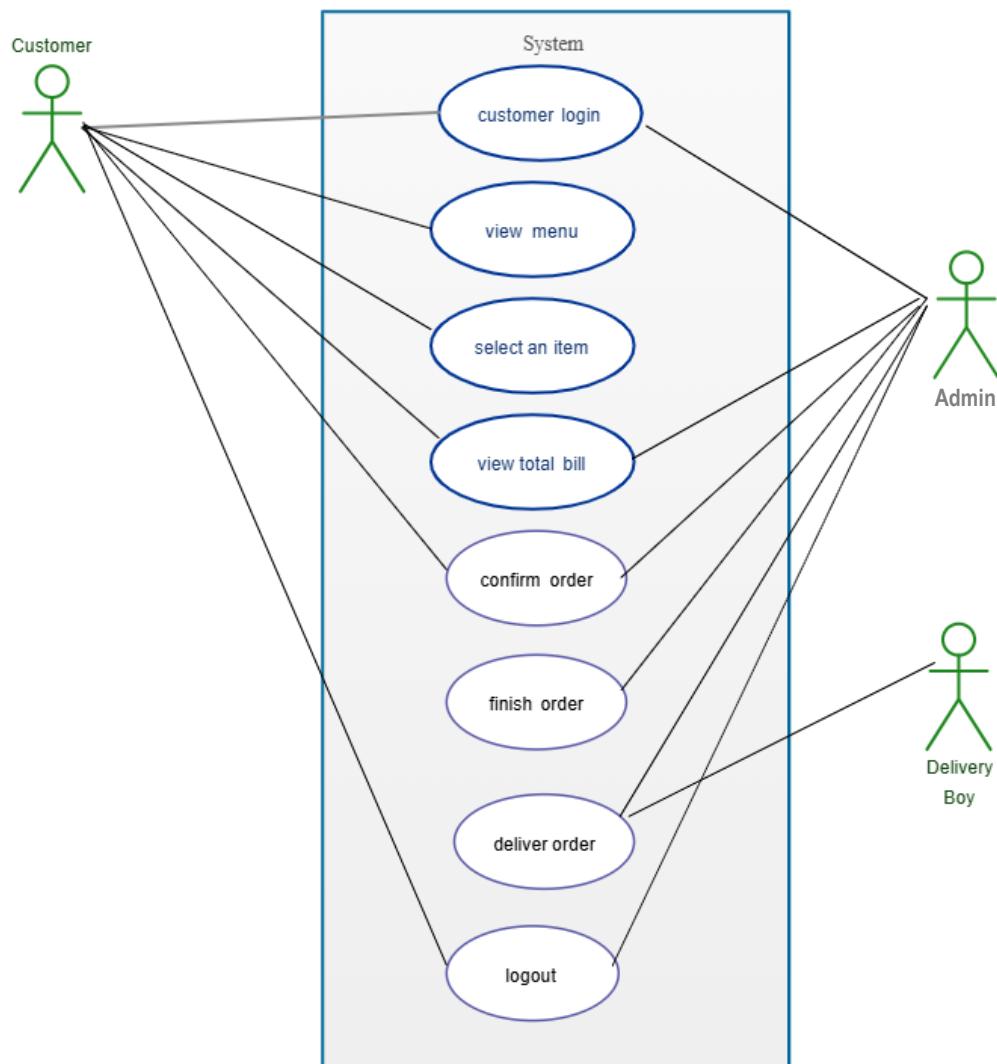
4.4.2 ER DIAGRAM: -

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. It is a tool used to design and model relational databases and shows the logical structure of the database. ER diagrams use symbols to represent entities, attributes, and relationships, which help to illustrate the relationships between the entities in the database. ER diagrams are commonly used in software engineering and database design to help developers and stakeholders understand and design complex databases.



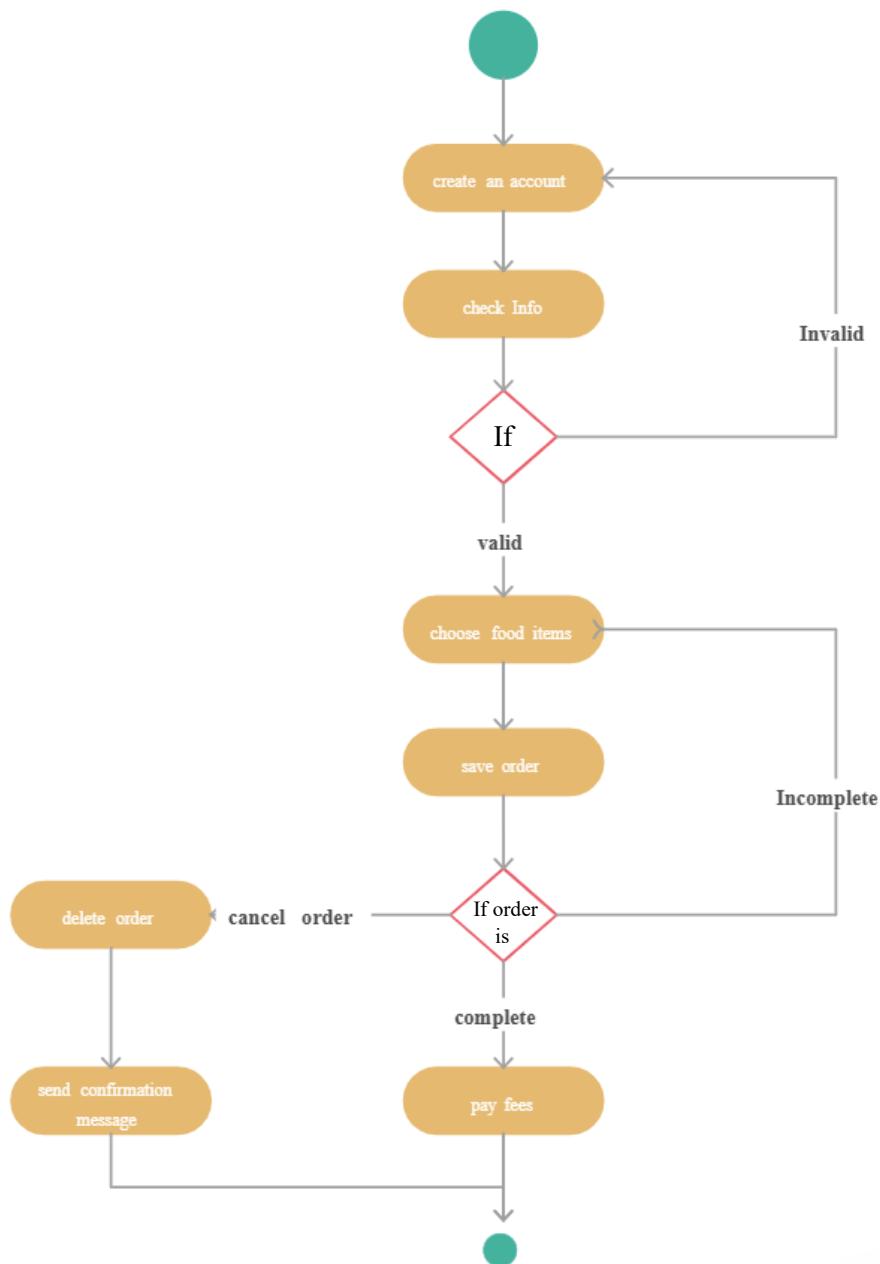
4.4.3 USE CASE DIAGRAM: -

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.



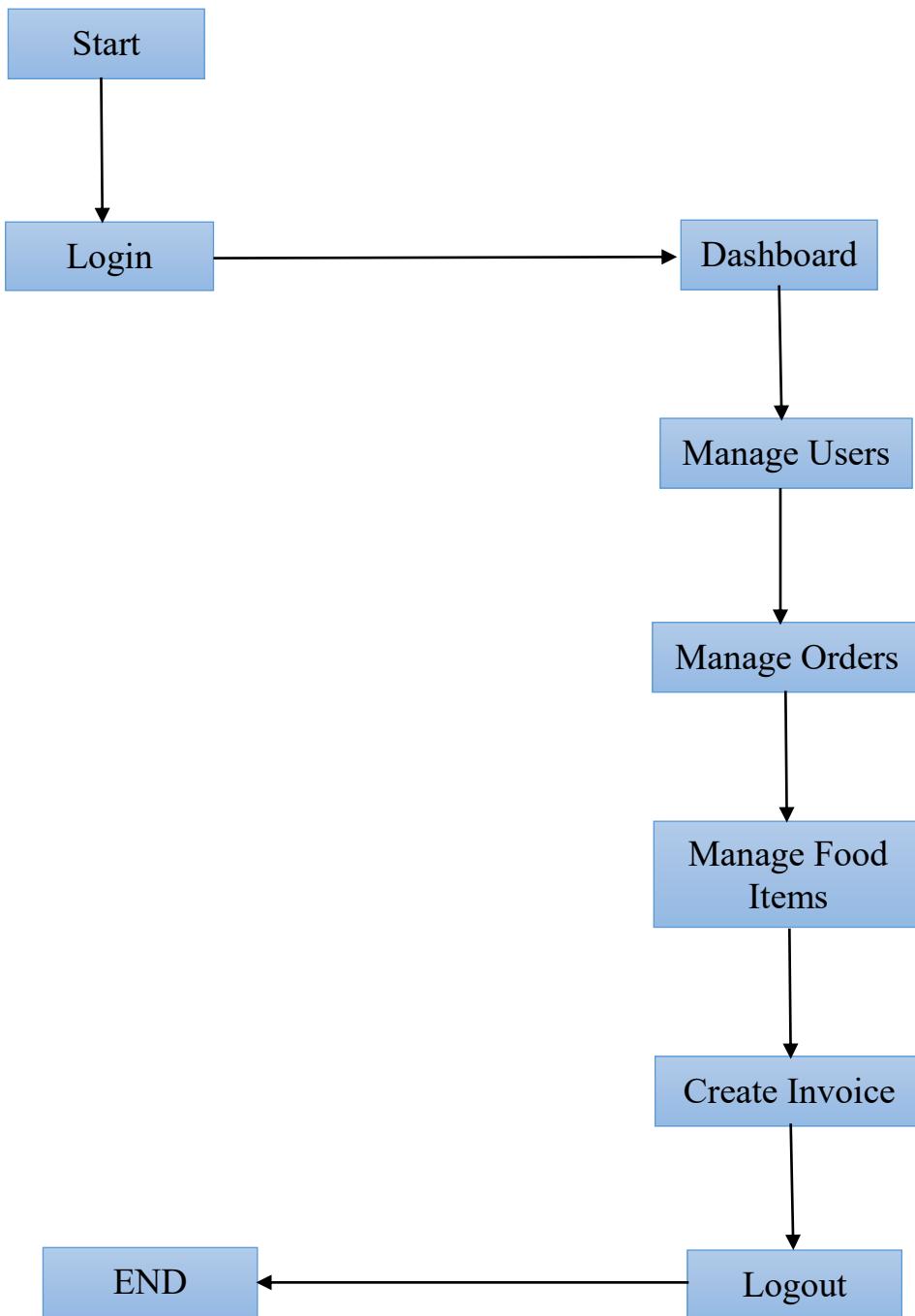
4.4.4 ACTIVITY DIAGRAM: -

Activity diagram is another important in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described in an operation of the system. The control flow is drawn from one operation to another. Activity models can be sequential and concurrent.



4.4.5 DATA-FLOW DIAGRAM: -

A Data-flow diagram (DFD) is a way of representing the flow of data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.



4.5 SECURITY ISSUES: -

In the system software the security depends upon the login authentication of the admin and another login authentication for customers. The admin password is known to admin123 only and customer password is different for each customer in the system. If a person tries to enter in software, he needs to crack first these two passwords. If he can login, then he can make changes or get details of someone.

4.6 TEST CASE DESIGN: -

Sr. no	Action	Input	Expected output	Actual output	Test result	Test comment
1.	Launch application	Click on software	Login page	Login page	Pass	Successful
2.	Enter correct email and password	email: abc@gmail.com Password: *****	User page or admin dashboard page	User page or admin dashboard page	Pass	User page or admin dashboard will display
3.	If email and password are incorrect	email: abc@gmail.com Password: *****	Login failed	Login failed	Fail	Invalid email and password
4.	If entered email is not in valid format	Enter email	Invalid email	Invalid email	Fail	Unsuccessful
5.	If entered name is in character format	Enter name	No error message	No error messages	Pass	Successful

5: IMPLEMENTATION AND TESTING

5.1 TESTING APPROACH: -

Software Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. It can also provide an objective, independent view of the software to allow the business to appreciate and understand the risk of software implementation. The basic purpose of testing is to detect the errors that may be present in the program or software. Testing is the process of executing a program with the intent of finding errors.

5.1.1 The box approach: -

Testing methods are traditionally divided into white-box and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

White-box Testing: - White box testing is also known as clear testing, glass testing, transparent box testing and structural testing. In white box testing an internal perspective of the system, as well as programming skills, are used to design test case. The testers choose inputs to exercise paths through the code and determine the appropriate outputs. While White box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level.

Here in this project, a white-box tester may look at the order flow from the moment a customer selects items to when the delivery details are processed. They would verify if the correct logic is applied at each step, such as checking inventory, calculating delivery charges, or applying discounts. Additionally, the tester ensures that the system is secure from SQL injections or other malicious attacks by reviewing the code for weaknesses and ensuring data validation is properly implemented.

Black-box Testing: - Black box testing treats the system as a “black box”, examining functionally without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the system is supposed to do, not how it does it.

For an online food delivery system, black-box testing involves evaluating the system's overall performance, including the user interface, responsiveness, and user experience during order placement, payment processing, and delivery tracking. Testers would ensure that the system meets user requirements by validating if users can complete actions like placing orders, modifying orders, and paying seamlessly.

This type of testing also checks for system behavior under various conditions, such as incorrect user inputs, slow internet connections, or unavailable delivery areas.

5.1.2 Levels of Testing: -

The levels of testing are as follows:

1. Unit Testing
2. Integration Testing
3. System Testing

1. Unit testing: - Unit testing focuses verification efforts on the smallest unit of the software design module. It involves testing individual modules of the application in isolation i.e., without interaction with other modules or external dependencies such as databases. Unit tests are often automated to run throughout development as a set of regression tests that can identify errors introduced during development.

For instance, it tests the functionalities of components like the order placement system, payment gateway, or the user authentication module. By isolating each feature and testing it with various input scenarios, developers can identify and fix bugs at the earliest stage of development, ensuring that small parts of the system, such as the calculation of total order cost or applying discounts, work as expected.

2. Integration testing: - Integration testing is a systematic testing for construction of the program structure while at the same time conducting tests to uncover errors associated with in the interface. It involves checking if modules work correctly when they are combined. This provides a wider scope for testing, and bugs that are introduced arising from module interactions are commonly detected during integration testing.

After unit testing, integration testing checks how well the order management system integrates with the payment gateway, or how the menu data is fetched from the database and displayed correctly on the app. It validates the flow of information between these interconnected components and ensures that data transitions smoothly, such as an order being correctly processed and reflected in the user's order history and kitchen system.

- 3. System testing:** - System testing involves checking if the entire integrated software is working correctly. Performed on a fully integrated system, this lets engineers check whether the system meets its high-level requirements. System testing is a comprehensive process that evaluates the complete online food delivery system as a whole to ensure all components work together and meet the specified requirements.

This includes testing all features such as user registration, menu browsing, order placement, payment processing, and delivery tracking, under various conditions like multiple users, high traffic, or simultaneous orders. It ensures the system operates efficiently and effectively, performing functional, security, performance, and usability tests to validate that the entire system is ready for production deployment.

5.2 CODING DETAIL AND CODE EFFICIENCY: -

In the software, to validate the form there are many if else and validation statements are used. The main part of software code is to perform database operation, accepting responses, form validation. I have used MySQL for database connection in this project.

All the fields such as Customer Name, Address, Email-address, Password, etc are validated and does not take invalid values.

Each for customer registration, customer login, order form, payment form cannot accept blank value fields.

- Integration of all the modules/forms in the system.
- Functionality of the entire module/forms.
- Avoiding errors in data.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Validations for user input.

5.2.1 Code Design: -

1) signup.php

```
<?php

session_start();

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "user_db";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}

$message = "";

$showCountdown = false;

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    $fullname = $_POST['fullname'];

    $email = $_POST['newEmail'];

    $password=password_hash($_POST['newPassword'],
PASSWORD_DEFAULT);

    $sql = "SELECT * FROM users WHERE email = '$email'";

    $result = $conn->query($sql);

    if ($result->num_rows > 0) {

        $message = "<p class='error-message'>Email is already registered!</p>";

    }

}
```

```
    } else {

        $sql = "INSERT INTO users (full_name, email, password) VALUES
        ('$fullname', '$email', '$password')";

        if ($conn->query($sql) === TRUE) {

            $message = "<p class='success-message'>New record created
            successfully!</p>";

            $showCountdown = true; // Set the flag to show countdown

        } else {

            $message = "<p class='error-message'>Error: " . $sql . "<br>" . $conn-
            >error . "</p>";

        }

    }

    $_SESSION['message'] = $message;
    $_SESSION['showCountdown'] = $showCountdown;
    header("Location: signup.php");
    exit();
}

$conn->close();

if (isset($_SESSION['message'])) {

    $message = $_SESSION['message'];
    unset($_SESSION['message']); displayed

}

if (isset($_SESSION['showCountdown']) && $_SESSION['showCountdown']
== true) {

    $showCountdown = true;

}
```

```
unset($_SESSION['showCountdown']);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Sign Up Form</title>

<link rel="stylesheet" href="login.css" />

</head>

<body>

<div class="form_container">

<div class="signup_form" id="signupForm">

<form action="signup.php" method="POST">

<h3>Create a New Account</h3>

<div class="input_box">

<label for="fullname">Full Name</label>

<input type="text" name="fullname" id="fullname" placeholder="Enter your full name" required />

</div>
```

```
<div class="input_box">
    <label for="newEmail">Email</label>

        <input type="email" name="newEmail" id="newEmail"
placeholder="Enter your email" required />

    </div>

    <div class="input_box">
        <label for="newPassword">Password</label>

        <input type="password" name="newPassword" id="newPassword"
placeholder="Create a password" required />

    </div>

    <div class="input_box">
        <label for="confirmPassword">Confirm Password</label>

        <input type="password" id="confirmPassword" placeholder="Confirm
your password" required />

    </div>

    <button type="submit">Sign Up</button>

    <p class="sign_in">Already have an account? <a href="login.php"
id="showLogin">Log in</a></p>

</form>

<?php echo $message; ?>

<?php if ($showCountdown): ?>

    <div id="countdownContainer">

        <p>Registration successful! You will be redirected in <span
id="countdown">3</span> seconds...</p>

    </div>
```

```
<?php endif; ?>

</div>

</div>

<script>

<?php if ($showCountdown): ?>

let countdown = 3;

const countdownElement = document.getElementById("countdown");

const interval = setInterval(function() {

    countdown--;

    countdownElement.textContent = countdown;

    if (countdown === 0) {

        clearInterval(interval);

        window.location.href = "login.php";

    }

}, 1000);

<?php endif; ?>

</script>

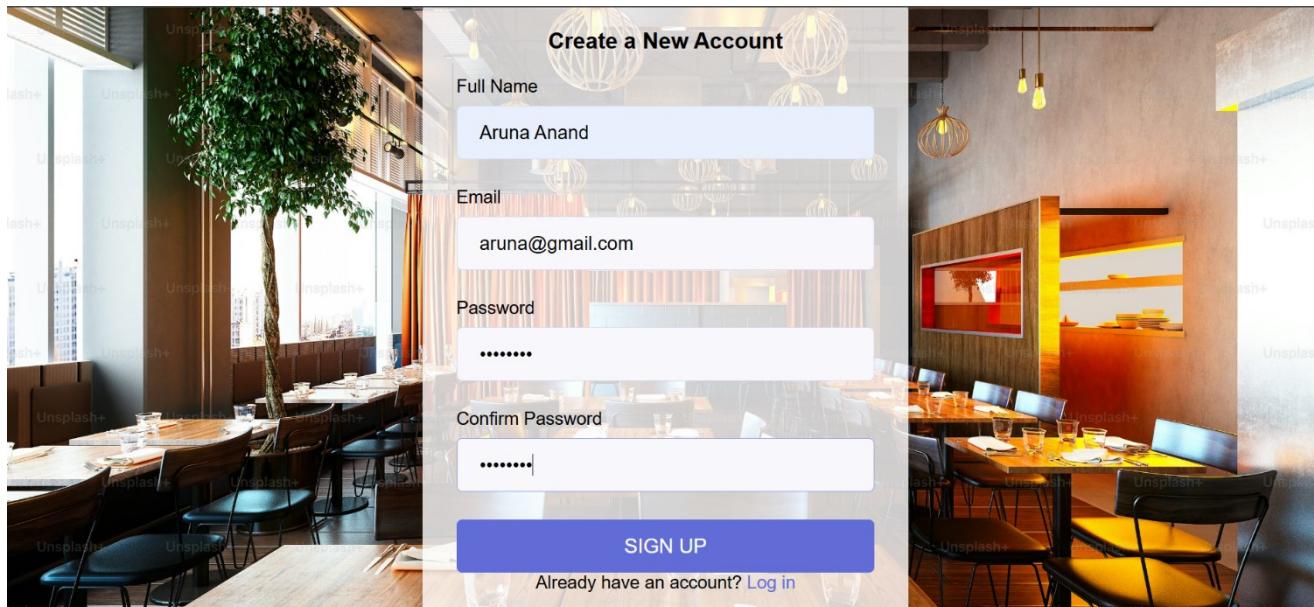
</body>

</html>
```

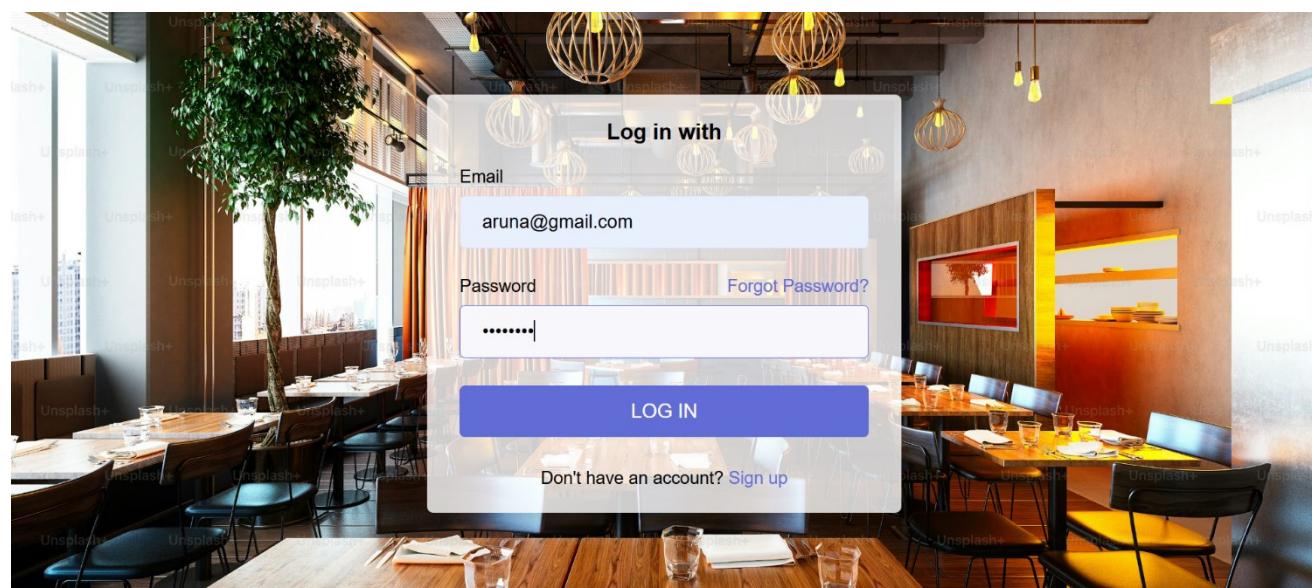
6. RESULTS

A) System Layout

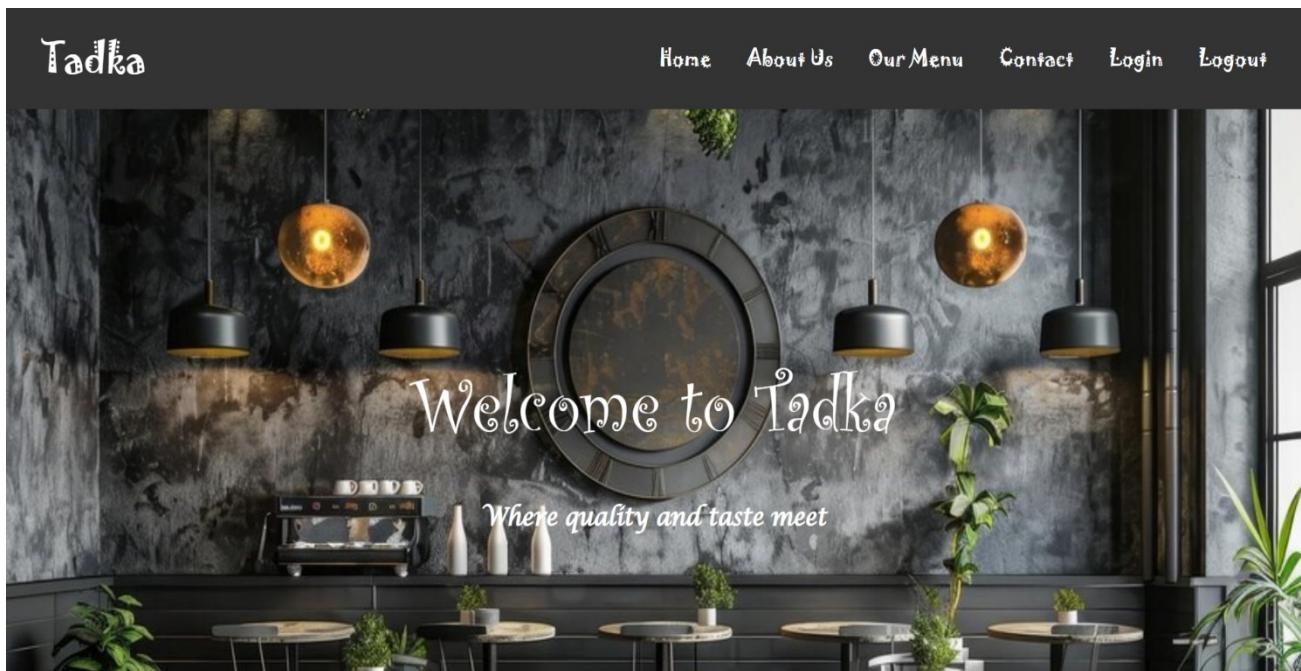
Customer Registration page: -



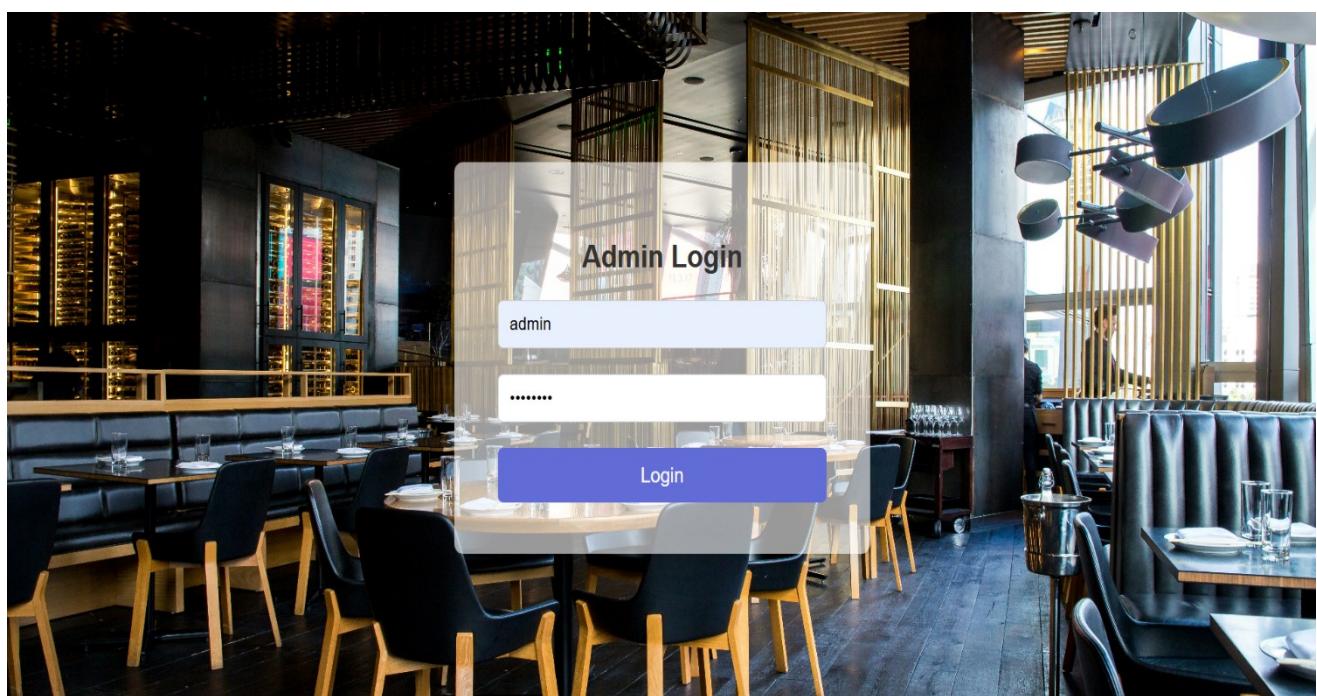
Customer Login page: -



Home page: -



Admin Login: -



Admin Dashboard: -

The screenshot shows the Admin Dashboard for a restaurant named 'Tadka'. The dashboard features a sidebar on the left with links for Dashboard, Manage Orders, Manage Users, Manage Menu, and Log Out. The main area has a background image of a restaurant interior. At the top right, there are three summary boxes: 'Total Orders' (3), 'Total Users' (3), and 'Total Sales' (₹11,480.00). Below these is a section titled 'Recent Orders' with a table:

Order ID	Customer Name	Total Price (INR)	Status
3	Aruna Anand	₹11,480.00	Completed
2	Samarth	₹6,560.00	In Progress
1	Samarth	₹18,860.00	Delivered

Manage Orders: -

The screenshot shows the 'Manage Orders' page. At the top, there is a navigation bar with links for Dashboard, Manage Users, Manage Menu, and Logout. The main title is 'Manage Orders'.

Below the title are four filter input fields: 'Search orders...', 'Order Status' (dropdown), 'Order Type' (dropdown), and 'Date Range' (dropdown), followed by a 'Apply Filters' button.

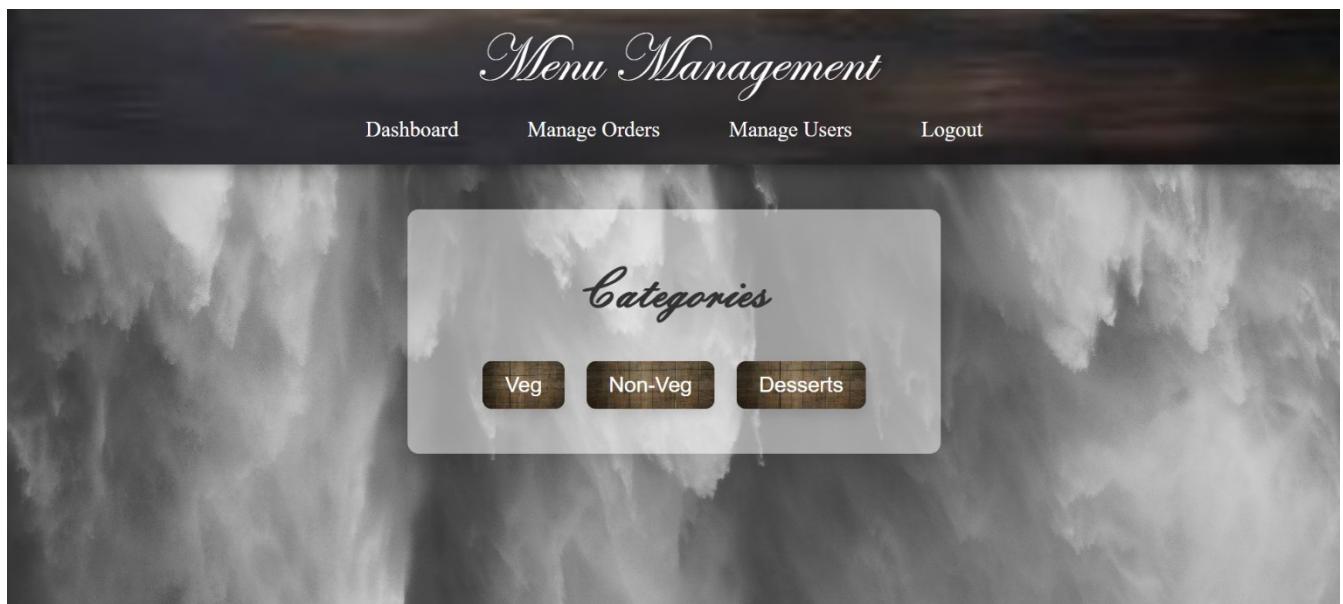
A table displays the list of orders:

Order ID	Customer Name	Items Ordered	Order Time	Total Price (₹)	Status	Actions
1	Samarth	Puranpoli, Misal Pav	2025-02-21 14:45:17	₹ 230.00	Delivered	View Update Status
2	Samarth	Dosa-Sambar	2025-02-21 16:11:39	₹ 80.00	In Progress	View Update Status
3	Aruna Anand	Kachori, Chole-Bhature, Puranpoli	2025-02-22 23:08:13	₹ 140.00	Completed	View Update Status

Manage Users: -

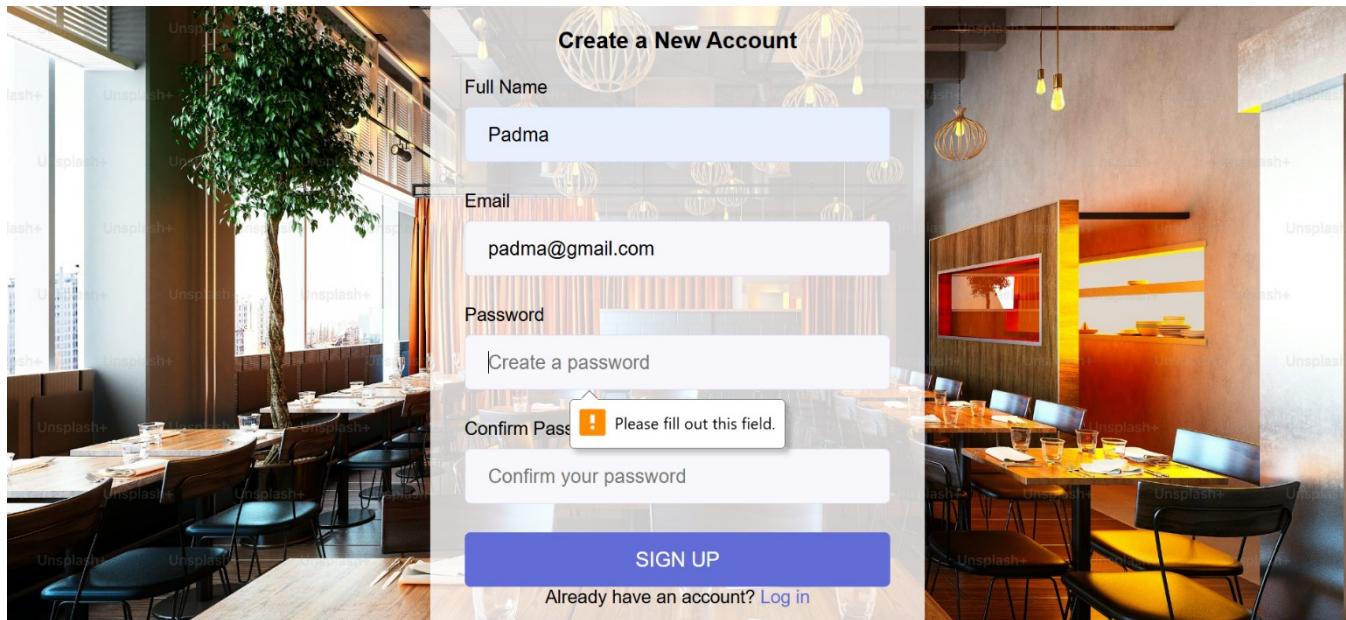
ID	Full Name	Email	Actions
1	Harshada M. Kotawadekar	harshada8k@gmail.com	View Edit Delete
2	Samarth Kotawadekar	samkot@gmail.com	View Edit Delete
3	Aruna Anand	aruna@gmail.com	View Edit Delete

Manage Menu: -

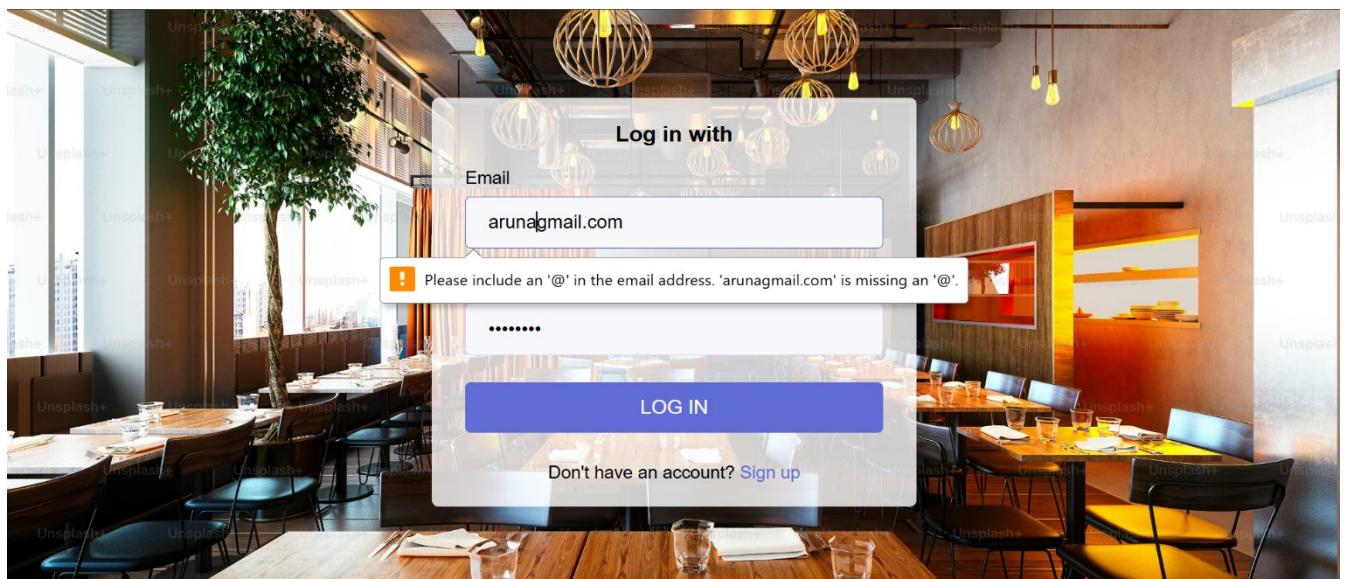


B) Validation

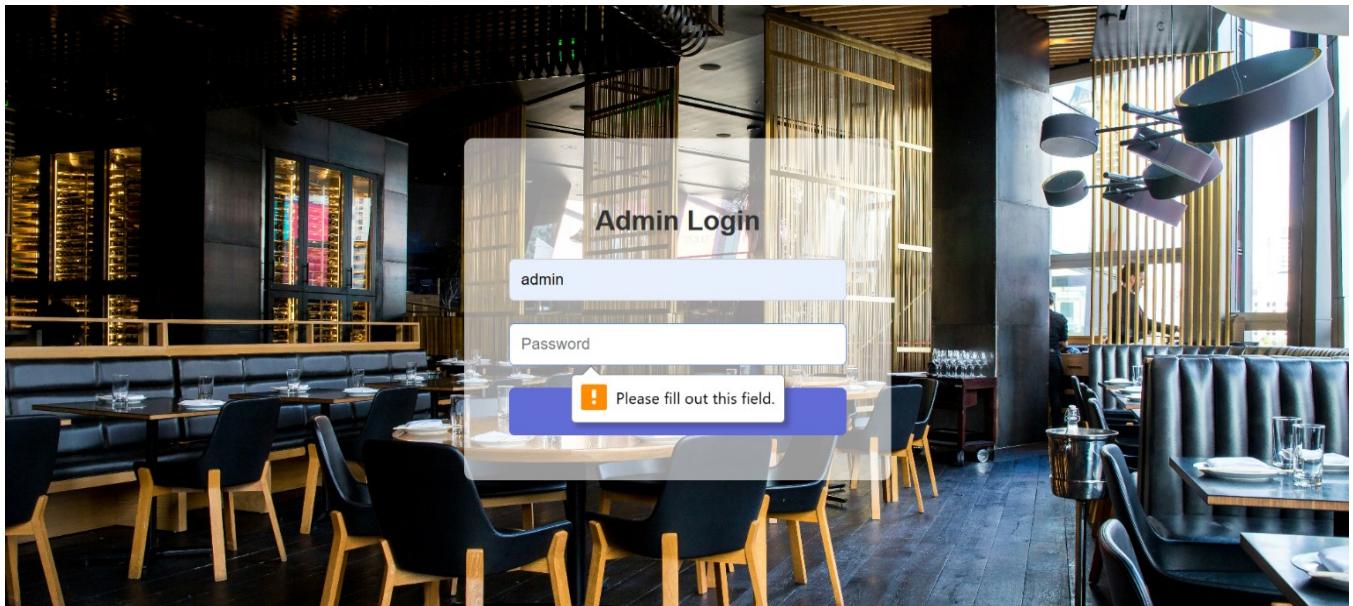
Customer Registration page: -



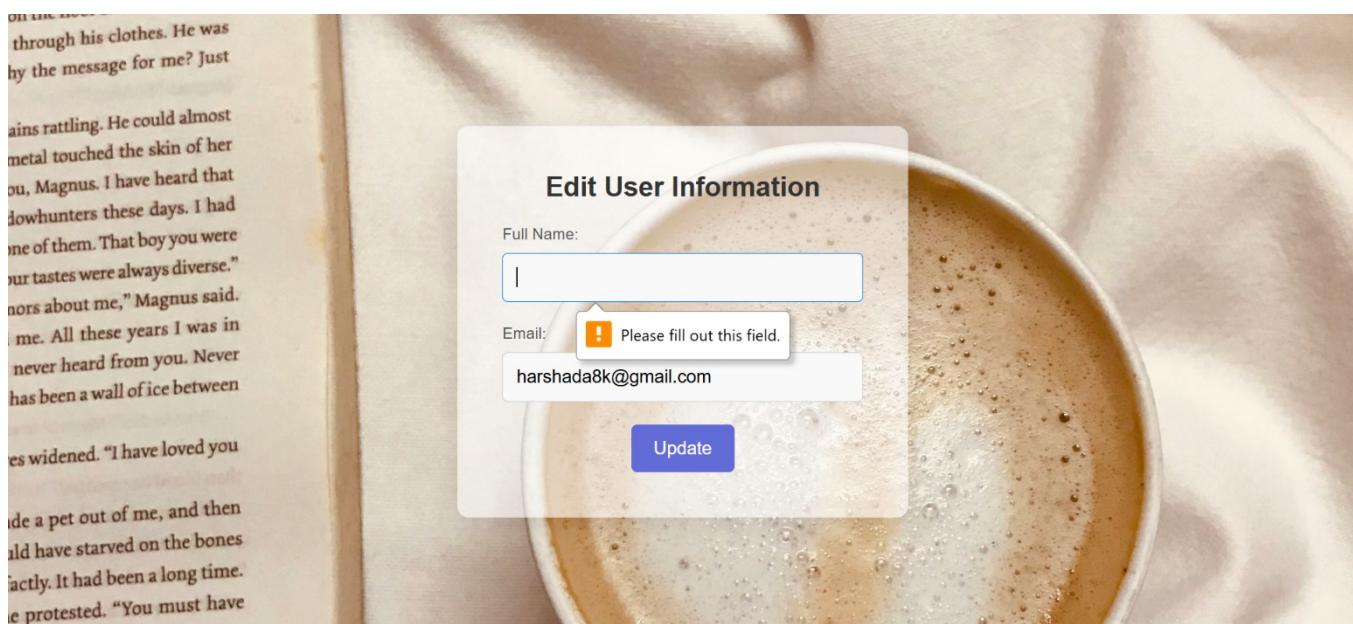
Customer Login page: -



Admin Login: -



Manage Users: -



C) DATABASE

Database Structure: -

The screenshot shows the phpMyAdmin interface for the database 'user_db'. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, test, and user_db. Under user_db, there are four tables: contact_form, orders, payments, and users. The main area displays a table of these four tables, showing details like Rows, Type, Collation, Size, and Overhead.

Table	Action	Rows	Type	Collation	Size	Overhead
contact_form	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 Kib	
orders	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 Kib	
payments	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 Kib	
users	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 Kib	
4 tables	Sum	23	InnoDB	utf8mb4_general_ci	96.0 Kib	

Users: -

The screenshot shows the phpMyAdmin interface for the 'users' table within the 'user_db' database. The left sidebar lists the same database structures as before. The main area shows the table structure with columns: id, email, password, and full_name. It includes options for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, and Operations.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)	utf8mb4_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	full_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Contact form: -

The screenshot shows the phpMyAdmin interface for the user_db database. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, test, and user_db. The right panel displays the structure of the contact_form table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	
2	name	varchar(255)	utf8mb4_general_ci		No	None			
3	email	varchar(255)	utf8mb4_general_ci		No	None			
4	message	text	utf8mb4_general_ci		No	None			
5	created_at	timestamp			No	current_timestamp()			

Orders: -

The screenshot shows the phpMyAdmin interface for the user_db database. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, test, user_db, contact_form, and orders. The right panel displays the structure of the orders table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	order_id	int(11)			No	None		AUTO_INCREMENT	Change Drop
2	food_items	text	utf8mb4_general_ci		No	None			Change Drop
3	total_price	decimal(10,2)			No	None			Change Drop
4	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop
5	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop
6	address	text	utf8mb4_general_ci		No	None			Change Drop
7	status	varchar(50)	utf8mb4_general_ci		No	None			Change Drop
8	order_date	datetime			Yes	current_timestamp()			Change Drop

Payments: -

The screenshot shows the phpMyAdmin interface for the user_db database. The left sidebar lists databases: information_schema, mysql, performance_schema, phpmyadmin, test, and user_db. The right panel displays the structure of the payments table.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	payment_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	order_id	int(11)			No	None			Change Drop More
3	payment_amount	decimal(10,2)			No	None			Change Drop More
4	payment_method	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	payment_status	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

7. CONCLUSION AND FUTURE WORK

7.1 ADVANTAGES OVER CURRENT SYSTEM: -

The Online Food Delivery System offers several advantages over traditional food ordering methods. Unlike the manual, phone-based ordering system, it provides a more efficient, convenient, and accurate way for customers to place orders. The automated system reduces the chances of errors in order details and ensures real-time tracking of both orders and deliveries.

Additionally, the system enhances customer satisfaction by offering features like customized order options, secure online payments, and multiple restaurant choices in a single platform. It also enables restaurants to reach a larger audience and manage orders more efficiently, improving overall operational performance.

7.2 FUTURE ENHANCEMENT: -

While the current system covers essential functionalities, there are several areas for future enhancement:

- 1. AI-based Recommendations:** Implementing machine learning algorithms to provide personalized food recommendations based on user preferences and past orders.
- 2. Voice-Activated Ordering:** Integrating voice assistants like Google Assistant or Alexa to allow customers to place orders hands-free.
- 3. Loyalty Programs:** Introducing a rewards system to incentivize repeat customers with discounts, loyalty points, and exclusive deals.
- 4. Global Expansion:** Expanding the platform to support international food deliveries and multi-currency payment options for a wider user base.
- 5. Sustainability Integration:** Collaborating with eco-friendly packaging vendors and offering delivery options with a lower environmental impact.

These enhancements will improve the user experience, expand the platform's reach, and add value to both customers and restaurants.

7.3 CONCLUSION: -

The **Online Food Delivery System** effectively addresses the growing demand for quick, reliable, and convenient food ordering services. It automates order placement, payment, and delivery management, significantly improving the efficiency and reliability of food delivery operations. By providing a user-friendly interface for both customers and restaurants, it streamlines the food ordering process and offers a seamless experience for all parties involved.

This system not only benefits consumers with fast and easy access to food but also helps restaurants enhance their service offerings and reduce operational inefficiencies. The overall impact is a more efficient, customer-focused, and profitable food delivery solution.

7.4 System Maintenance: -

To ensure the **Online Food Delivery System** continues to function optimally, regular maintenance is required. The maintenance activities will include:

1. **Bug Fixing:** Identifying and resolving any software bugs or issues that affect system performance or user experience.
2. **System Updates:** Regular updates to improve features, enhance security, and keep the system up-to-date with new technologies.
3. **Database Maintenance:** Routine checks and optimizations to ensure database integrity, performance, and data backups.
4. **Performance Monitoring:** Ongoing monitoring of system performance to detect bottlenecks, improve response times, and ensure scalability as the user base grows.
5. **User Support:** Providing customer support for troubleshooting, issue resolution, and addressing any user complaints or feedback.

Effective system maintenance will ensure the platform runs smoothly, remains secure, and meets the evolving needs of users over time.

8. REFERENCES

We express our sincere gratitude to all those people who helped us in gathering the information while preparing this project. For this project, I had to prepare information about how to develop the software efficiently and fulfill the client requirements. For which I had taken the reference from the following sources:

- Google: <https://www.google.com>
- YouTube: <https://www.youtube.com>