Tribhuvan University

Faculty of Humanities and Social Science

Online Food Ordering System

A PROJECT REPORT

Submitted to

Department of Computer Application

Pascal National College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Sujal Poudel | 6-2-1226-26-2021

June 2025

Under Supervision of

Nawaraj Poudel

Tribhuvan University

Faculty of Humanities and Social Science

Pascal National College

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by SUJAL POUDEL entitled "**Online Food Ordering System**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

_____

Nawaraj Poudel

SUPERVISOR

Faculty Member

Department of Computer Application

Pascal National College, Satdobato, Lalitpur

Tribhuvan University

Faculty of Humanities and Social Science

Pascal National College

LETTER OF APPROVAL

This is to certify that this project prepared by SUJAL POUDEL entitled "**Online Food Ordering System**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Nawaraj Poudel Faculty Member Department of Computer Application Pascal National College Satdobato, Lalitpur	**Suresh Thapa** BCA Coordinator Department of Computer Application Pascal National College Satdobato, Lalitpur
(Internal Examiner)	(External Examiner)

ABSTRACT

The food order management system is the back-end of the system that the restaurant uses to receive and manage the orders. It enables the restaurant to track the progress of each order, update the customer on the status of their order, and coordinate the preparation and delivery of the food. The payment gateway is the component that enables the customer to pay for their order securely, using their preferred payment method. Overall, a food ordering system can enhance the customer experience, increase efficiency, and improve the profitability of a restaurant or food service business.The system is built on a user-friendly interface accessible through web and mobile applications, allowing customers to browse a diverse range of menus, place orders, and make payments effortlessly. The documentation outlines the intuitive order placement process, ensuring a smooth and enjoyable experience for users. The documentation covers the system's robust baekend, which utilizes database management systems to efficiently store and retrieve information. It also explores the use of APIs to facilitate commumcation between the frontend and backend components, ensuring real-time updates and responsiveness Additionally, the documentation discusses the implementation of feedback and rating systems, fostering a sense of trust and transparency between customers and restaurants The Food Ordering System aims to enhance customer satisfaction by providing a user-centric approach and empowering-restaurant owners with tools fo optimize their operations In conclusion, the Food Ordering System documentation serves as a comprehensive ginde for users, developers, and stakeholders mterested in understanding the architecture, functionality, and benefits of the system.

Keywords*:* Online Food Ordering System, Dotnet Backend, SqlServer Database, Food Ordering, User Authentication, Order Management, HTML/CSS Frontend

ACKNOWLEDGEMENT

I would like to sincerely thank everyone who contributed to the successful development of the Online Food Ordering System.

To begin with, my heartfelt appreciation goes to my project supervisor, Nawaraj Poudel, and our Principal, Ashok Kumar Pant, for their exceptional guidance and unwavering support. Their insights, encouragement, and expert advice played a vital role in steering the project in the right direction and motivating us to strive for excellence.

I'm also grateful to the many individuals who offered their time and shared their knowledge—whether through thoughtful discussions, valuable feedback, or access to helpful research materials. Your input has been deeply appreciated and has significantly influenced the depth and quality of my work.

Sincerely,

Sujal Poudel

LIST OF ABBREVIATIONS

API	Application Programming Interface
DFD	Data Flow Diagram
ERD	Entity Relation Diagram
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
SP	Stored Procedure
JS	JavaScript
MVC	Model View Controller
UI	User Interface
VS	Visual Studio

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Online Food Ordering System is a web-based application designed to streamline the process of ordering food from restaurants via the internet. It provides a convenient platform for users to browse menus, register, log in, place orders, make payments, and track deliveries from the comfort of their homes. The system also includes an administrative interface that allows restaurant owners or admins to manage categories, products, user details, order statuses, and customer inquiries efficiently. By automating various operations and enabling real-time interactions between users and restaurants, the system enhances customer experience and operational efficiency in the food service industry. The system eliminates the need for manual order-taking and reduces the chances of errors in food orders. It supports secure online payments and maintains a comprehensive order history for both users and administrators. The admin panel provides insightful data analytics and order tracking features to optimize restaurant operations. Overall, the Online Food Ordering System bridges the gap between restaurants and customers through an efficient, user-friendly digital solution.

1.2 Problem Statement

Traditional food ordering methods are often time-consuming, prone to errors, and lack the convenience that modern users expect. Customers face difficulties in accessing updated menus, placing accurate orders, and receiving timely delivery updates, especially during peak hours. Restaurants, on the other hand, struggle with managing a high volume of orders manually, leading to operational inefficiencies, miscommunication, and customer dissatisfaction. There is a clear need for an automated, real-time, and user-friendly platform that connects customers and restaurants seamlessly. The Online Food Ordering System aims to address these issues by providing a digital solution that simplifies the ordering process, enhances accuracy, and improves overall service efficiency. Additionally, the absence of centralized data management makes it difficult for restaurants to analyze sales trends and customer preferences. Communication gaps between users, and restaurants can lead to delays and poor service quality. This project seeks to overcome these challenges by

implementing a robust, scalable system that supports efficient order handling, real-time updates, and data-driven decision-making.

1.3 Objectives

The objectives of this project are follows as:

- To let users to order food online easily.
- To help admins to manage orders and menus.
- To keep users and admins updated with order info.

1.4 Scopes and Limitation

1.4.1 Scopes

- Users can register, log in, view menus, place orders, and make payments online.
- Admins can manage categories, products, user details, and order statuses through a dashboard.
- The system allows real-time updates for order tracking and status changes.

1.4.2 Limitations

- The system requires a stable internet connection to work properly.
- It may not support high traffic if many users access it at the same time.
- It does not handle delivery logistics (e.g., assigning drivers) in detail.

1.5 Development Methodology

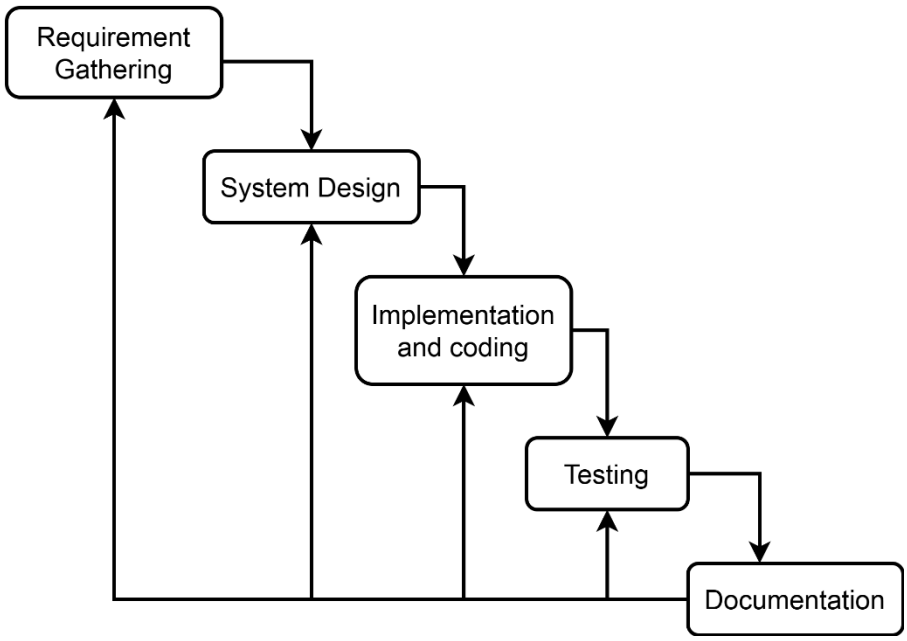


Figure 1. 1 Development Methodology of Online Food Ordering System

For this project development, Waterfall methodology is used, the functional requirements are gathered first. High level and low-level system design such as data flow diagram, system flowchart, working mechanism of system, database schema, interface design, system architecture is designed based on the requirements. After designing all the system necessary modules and components, the system is implemented based on those designs. Testing of the each modules and functions is done, further new requirements is gathered for more feature and repeat some of cycle steps.

1.6 Report Organization

The **first chapter** of the report contains the summarized introduction of the whole project report. It includes the overview, scope and limitation, problem statement and objectives of this project.

The **second chapter** includes background study i.e., description of fundamental theories, general concepts and terminology related to the project. It also includes the literature review i.e. review of the similar projects, research and theories done by others.

The **third chapter** includes the system analysis and design phase in which the report of functional and non-functional requirements of the project is stated using use case and system diagrams. It includes the feasibility study, shows the technical, operational and economic feasibility of the project development phase. The explanation of the designing of the system is also done in this chapter. It includes data modeling and process modeling which is explained by using ER diagram and Data Flow Diagram. The architectural design, database designs, user interface designs and algorithms are also listed in this chapter.

The **fourth chapter** includes the implementation and testing phase includes how the tools like CASE tools, programming languages and database platforms are implemented. In testing, it includes the test cases tables for unit testing and system testing.

The **fifth chapter** includes conclusion and future recommendation. It contains the final paragraphs of the report and in this phase the overall outcome and the developer point of view is written. The lesson learned through all the phases is included in this chapter.

CHAPTER 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

In recent years, the demand for online services has significantly increased, especially in the food industry. People prefer ordering food online due to its convenience, speed, and flexibility. Traditional ordering methods, such as phone calls or in-person visits, are often slow, prone to errors, and less efficient. Online food ordering systems have emerged to address these issues by offering user-friendly platforms where customers can view menus, place orders, and make payments digitally. Many popular food delivery apps like Swiggy, Zomato, and Uber Eats have set industry standards, inspiring smaller businesses to adopt similar systems. This project aims to create a simple and effective online food ordering system tailored to meet the needs of both customers and administrators.

2.2 Literature Review

For this project, we researched and reviewed some of the related websites Throughout the research, I get to find out that there are very few website related to ordering.

If some website are available for users, I found the websites are looking many features regarding the ordering system. The users are convinced with the services.

The authors off!]proposed an online food ordering system which sets up a food menu online and customers can easily place the order as per their wish that enables case for the customers which overcomes the disadvantages of the traditional queuing system.The authors of [2] Food panda are limited with only restaurants and cafe where we can place orders. All the time it is not affordable and healthy for people to order the food from the restaurants, so there is need of an application which will provide the facility of home-made food and mess too.

The authors of [3] An. Online Food Ordering System is proposed here which simplifies the food ordering process. The proposed system shows an user interface and update the menu

with all available options so that it eases the customer work Customer can choose more than one item to make an order and can view order details before logging off.

The author of [4] this proposed system is an online food ordering system that enables case for the customers. In addition to these systems, some recent research has explored integrating GPS tracking, mobile notifications, and secure payment gateways to enhance reliability and user engagement. However, most existing systems still lack features such as customer feedback loops, multi-vendor support (home chefs, mess owners), and AI-driven recommendations, which could significantly improve personalization and service quality.

Overall, there is a clear gap in the market for a comprehensive online food ordering system that not only includes restaurant services but also supports home-made food delivery, customizable orders, real-time status updates, and a robust admin panel for management. This project aims to fill that gap by developing a system that is inclusive, efficient, and tailored to user needs. Another significant challenge identified in existing systems is the lack of scalability and customization for different user groups. Most current platforms focus only on urban populations and do not address the needs of students, office workers, or people in semi-urban or rural areas who may rely on local mess services or home kitchens. Systems with rigid backend management structures also struggle to accommodate menu changes, availability updates, and order modifications in real-time. This limitation affects both the customer experience and vendor flexibility. Recent studies have emphasized the importance of adaptive design and backend scalability to ensure that food ordering systems can cater to a diverse range of users and service providers. Moreover, security and data privacy are increasingly important aspects of food ordering platforms. Many systems do not implement strong encryption or secure payment mechanisms, putting users at risk. Researchers have suggested integrating secure gateways and multi-factor authentication to build user trust. Additionally, artificial intelligence and machine learning are being explored to personalize the user experience by analyzing order history and preferences. These advancements could lead to smarter recommendation engines and more efficient delivery logistics. However, despite these emerging trends, many existing platforms still operate with basic functionality, indicating a strong need for innovation and broader feature integration in online food ordering systems.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

3.1.1 Requirement Analysis

I. Functional Requirement:

- The system should be able to register and login users.
- The system should be able to show Menu to Users.
- The users should be able to order Foods.
- The system should be able to place limit food categories.

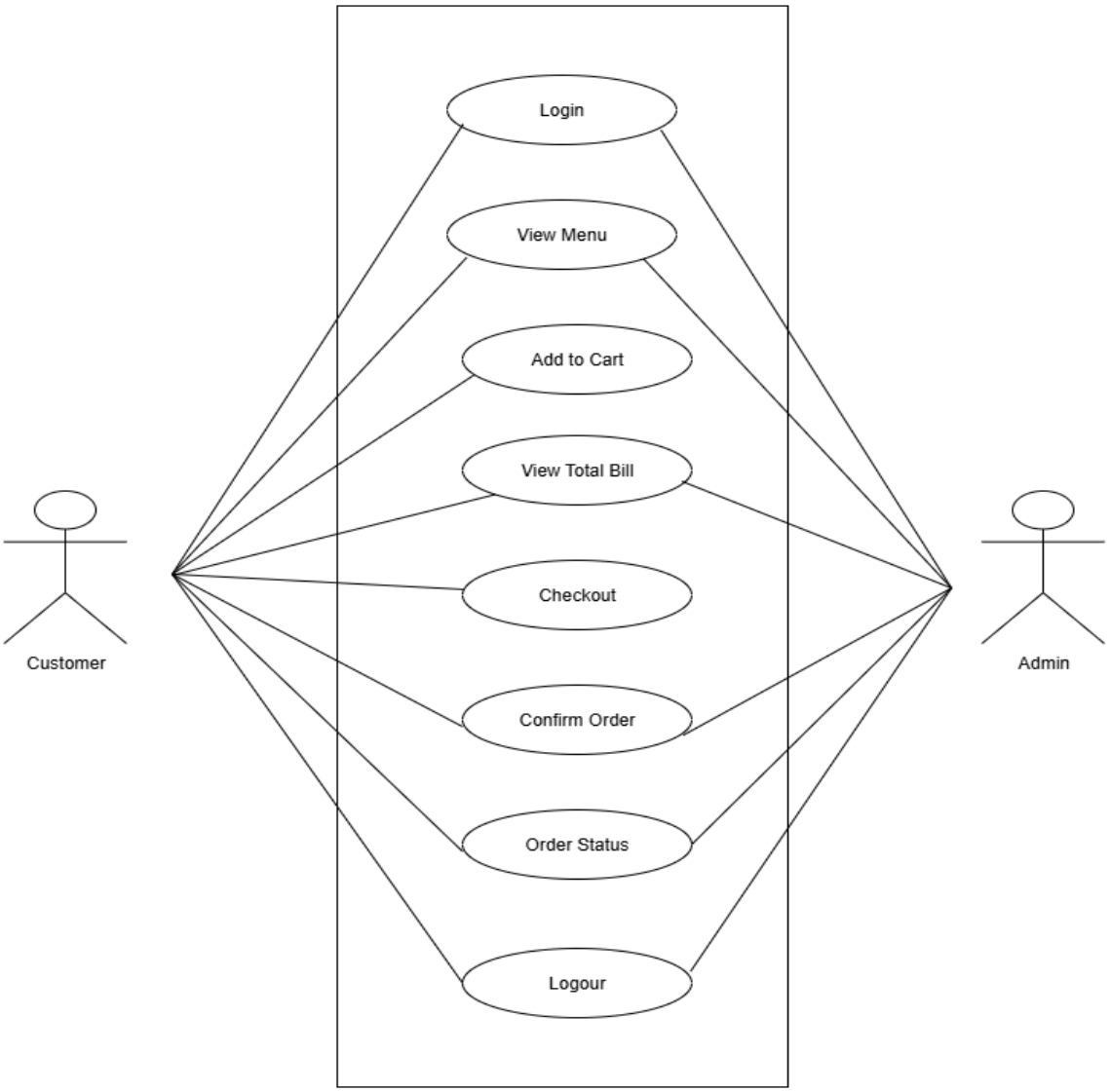


Figure 3. 1 Use case diagram of Online food ordering system

II. Non-Functional Requirement:

- The application should be secured.
- The application should have easy to use and interactive user interface.
- The application should be faster in performance.

3.1.2 Feasibility Analysis

I. Technical

The Online Food Ordering System is technically feasible because current technologies support its development easily. Modern web and mobile frameworks allow building user-friendly apps, while backend systems and databases can handle orders and user data efficiently. Integration with payment gateways and delivery tracking is straightforward using existing APIs. Cloud services ensure the system can scale and remain reliable. Skilled developers are available to build and maintain the system, making the technical implementation practical and achievable.

II. Operational

This project is simple and intuitive, using this application doesn't require extra knowledge. It is very user-friendly and can be easily adapted by any new user that will use this website. As it uses well-known web technologies, it is easy to maintain.

III. Economic

This project will be highly cost-effective, as it relies on free and open-source technologies such as SQL Server Dotnet**,** Visual Studio, and other community-supported tools. No financial investment is required for software licensing or development tools. The primary requirement is setting up a suitable development environment along with effective supervision to ensure the project progresses efficiently and meets its objectives.

IV. Schedule

This project is completed following below schedule.

This project was scheduled as:

2025	Feb	March	April	May	June	July	Remarks
Requirement Gathering							Complete
System Design							Complete
Development Phase							Complete
Development Phase II							Complete
Testing							Complete
Documentation							Complete

Figure 3.2: Gantt Chart

3.1.3. Data Modelling

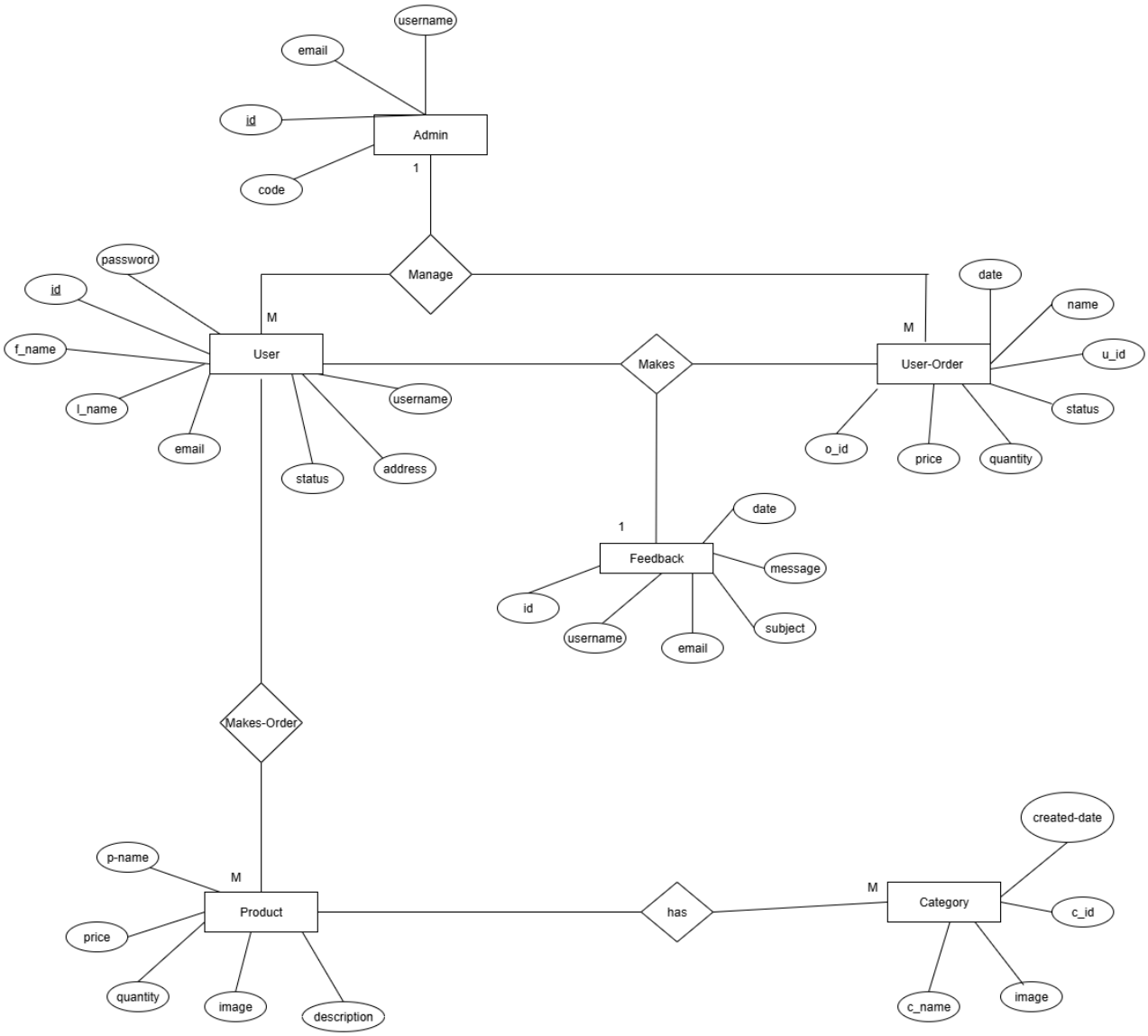


Figure 3. 3 Entity-relation diagram of Online food system

The Entity Relationship (ER) diagram illustrates the logical structure of the database for the Online Food Ordering System. It consists of two main entities: Admin and User.

3.1.4 Process Modelling

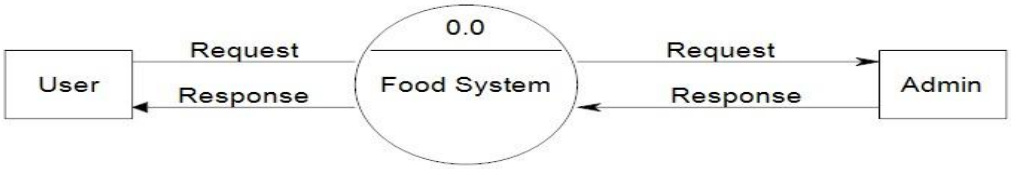


Figure 3. 4 Context level DFD of Online Food Ordering System

This context level data flow diagram provides a high-level overview of the **Online Food Ordering System**, focusing on the key entities (Admin and User) and their interactions. It serves as a foundation for developing more detailed data flow diagrams that capture the processes, data transformations, and interactions within the application.

3.2 System Design

3.2.1 Architectural Design

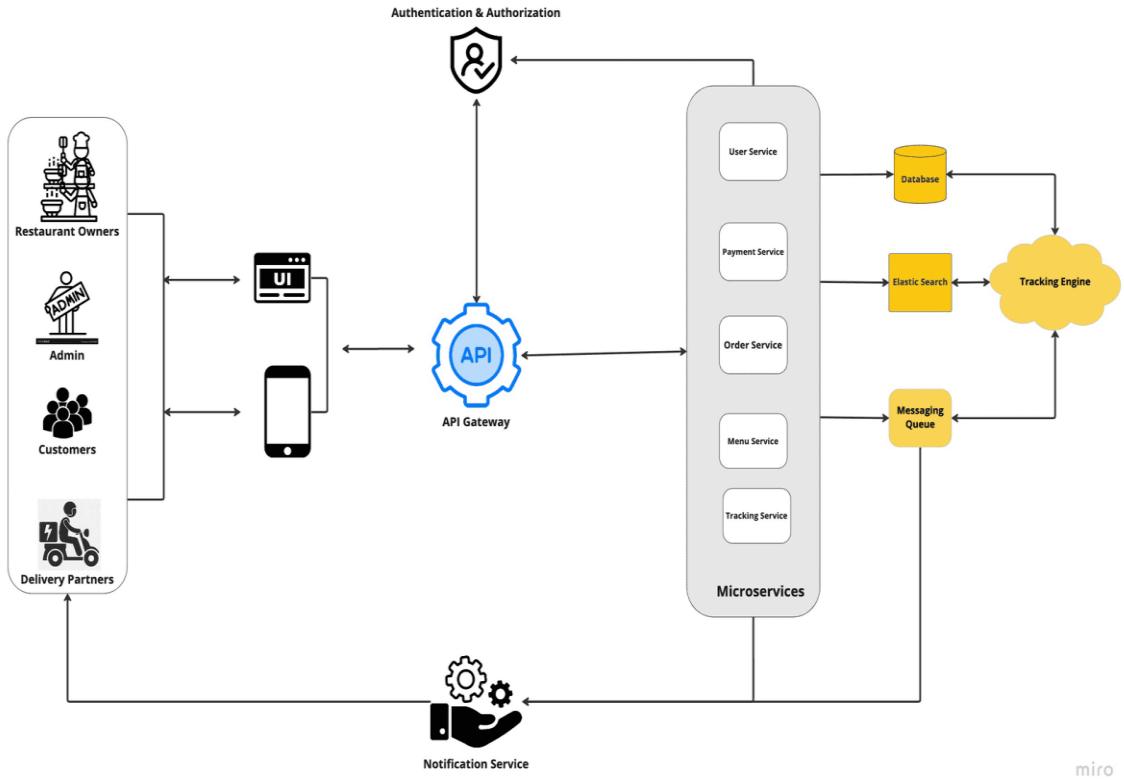


Figure 3. 5 Architecture design of Food Ordering System

3.2.2 Database Schema Design

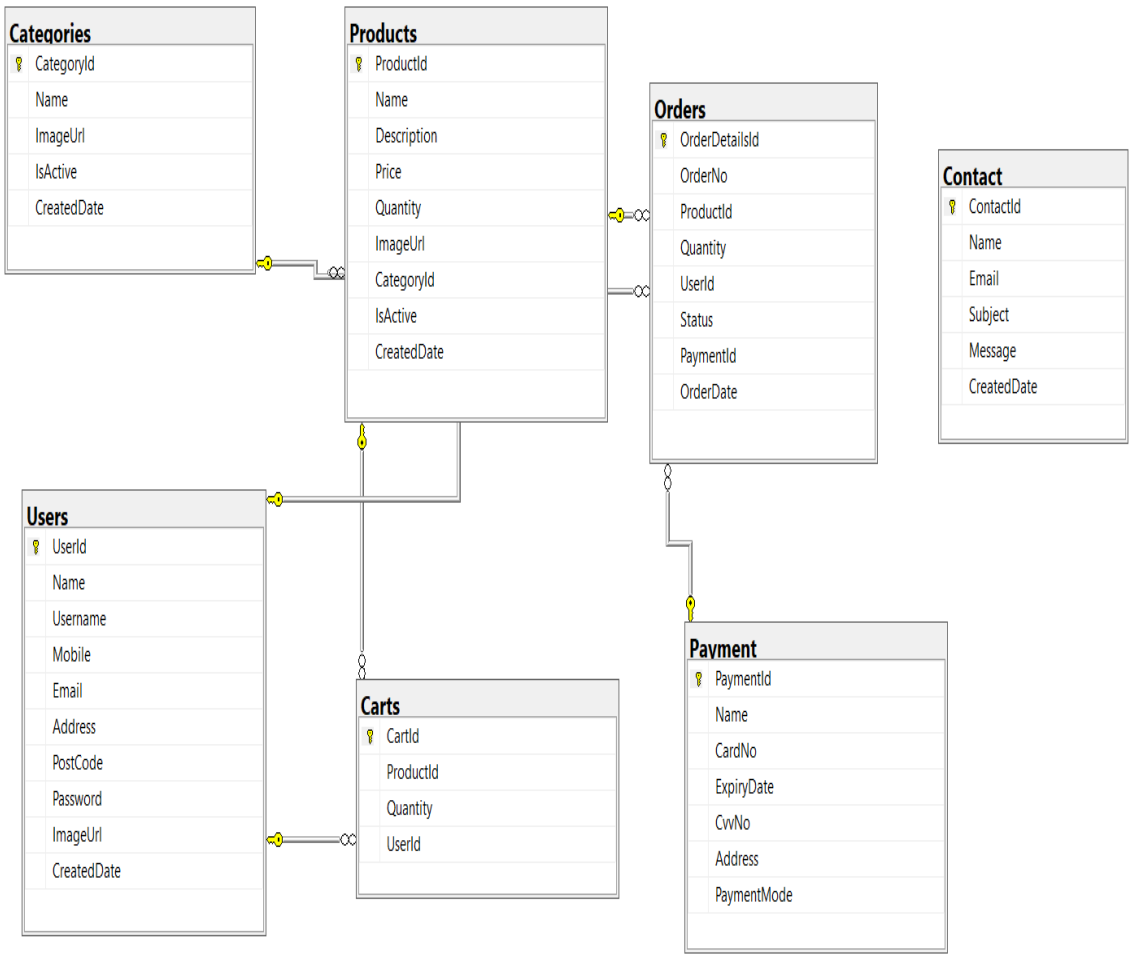


Figure 3. 6 Database schema of system

The above figure is an Database schema for Online Food Ordering System. It showcases Seven tables along with their respective fields and data types. This figure represents a relational database schema for an Online Food Ordering System. It includes key entities such as Users, Products, Categories, Carts, Orders, Payment, and Contact. Each table contains relevant attributes, and relationships are established through foreign keys. For example, Products belong to Categories, and Orders link Users, Products, and Payments. Users can also add Products to their Carts, and the Contact table stores user inquiries. This schema ensures structured data storage and supports core functionalities of an online shopping platform.

3.2.3. Physical DFD

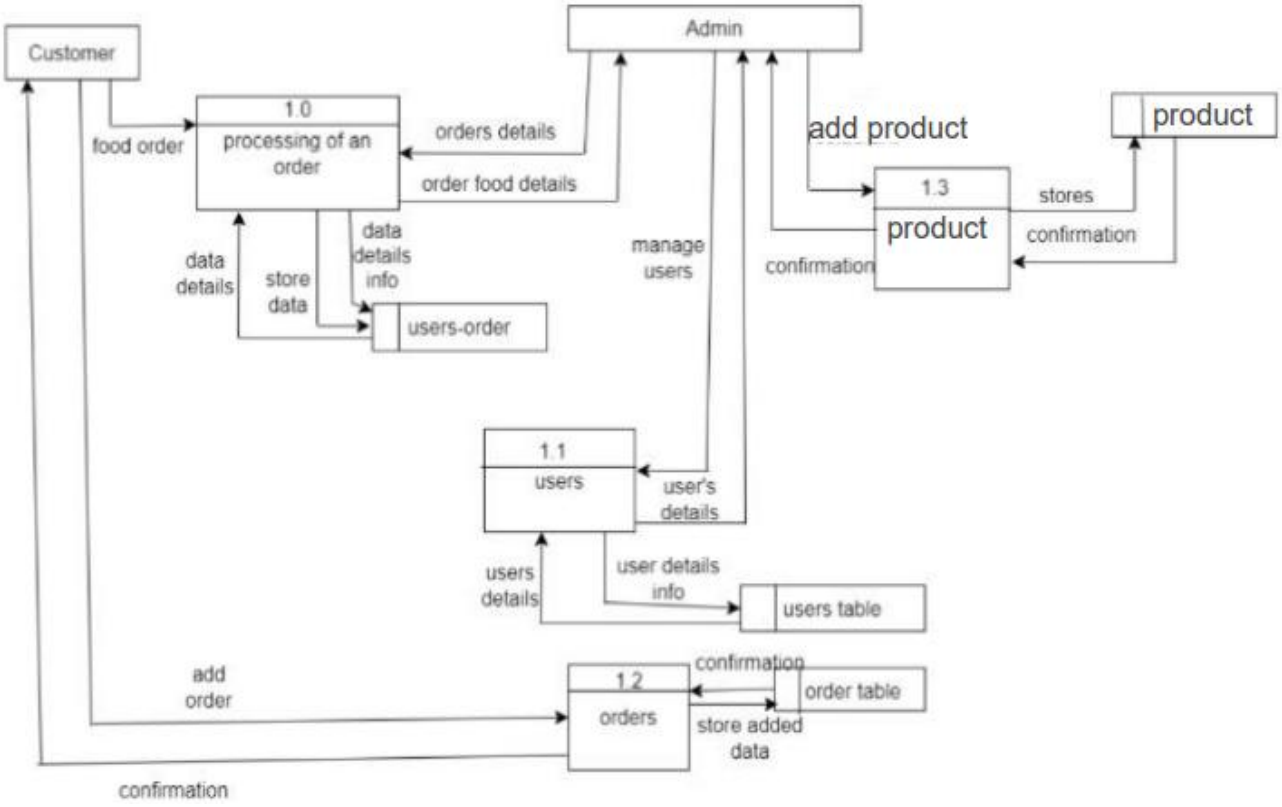


Figure 3. 7 schema Diagram for Online Food Ordering system

This diagram illustrates the Schema Diagram of a food ordering system, highlighting the interaction between different entities: Customer, Admin, and various system processes. Customers initiate the process by placing a food order, which is handled in the "Processing of an Order**"** module (Process 1.0). This process stores order data and user order details and forwards them to the Orders module (1.2), which records the data in the Order Table and sends a confirmation back to the customer. Meanwhile, Admins manage the system by handling user and product data. They can add products via the "Add Product" module (1.3), which stores product information in the Product Table and confirms successful entry. Admins also manage users through the Users module (1.1), updating and storing user details in the Users Table. Overall, the Diagram outlines how data flows between processes, external entities, and data stores to support food ordering, user management, and product management.

3.3. Algorithm Details

3.3.1.Sales Data Aggregation Algorithm

The Sales Data Aggregation Algorithm is implemented to generate a selling report based on a user-defined date range. It filters orders within the specified period and aggregates the total number of items ordered and the total cost per user. This algorithm helps in analyzing user purchases and tracking sales performance efficiently.

3.3.2.Working Steps of Sales Data Aggregation Algorithm

1. **Input Collection:**
    - Accept From Date and To Date from the user via the UI.
2. **Fetch Orders:**
    - Retrieve all order records from the database.
3. **Filter Orders by Date:**
    - Iterate through the order list and select only those orders where the order date is within the selected date range.
4. **Group by User:**
    - For each filtered order, group the data by user (using UserId or Email).
5. **Aggregate Order Data:**
    - For each user:
        - Count the total number of items ordered.
        - Sum the total cost of all their orders.
6. **Store Results:**
    - Store each user's name, email, total item count, and total cost in a temporary result structure (e.g., list or table).
7. **Calculate Total Sales:**
    - Sum all users' individual order costs to compute the overall Sold Cost.
8. **Display Report:**

    - Present the final aggregated data in a table format under the Selling Report section.
    - Show total Sold Cost below the table.

3.2.4 Interface Design

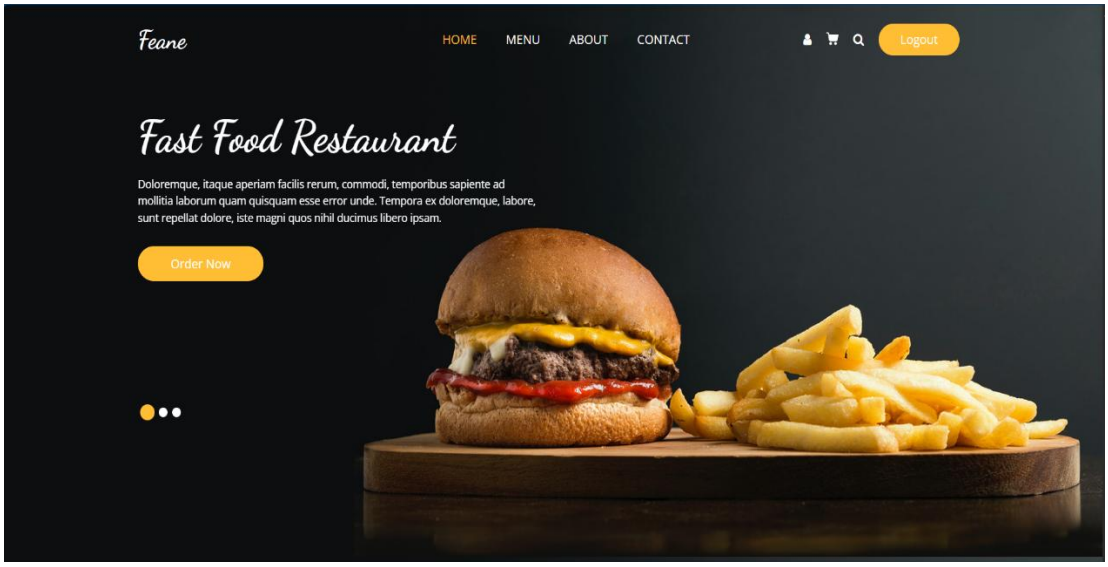


Figure 3. 7 Interface design for Online Food Ordering System

The above diagram showcases a user interface of an **Online Food Ordering** system, designed for simplicity, accessibility, and clear financial insight.

At the top, there's a **navigation bar** that includes tabs such as **Home**,**Menu Login**, and **Register** etc allowing users to quickly access essential areas of the application.

Below the navigation bar, the interface features a **prominent header section** with the title *"Order Now"* and a brief description encouraging users to manage their Foods in one centralized place.

Further down, the **Features section** is organized into three clearly defined cards:

- **User Details**: Allows users to Edit their provided Details
- **Add To Cart**: Allows users to ADD their selected foods in Cart
- **Search** : Users can search their favourite Menus .

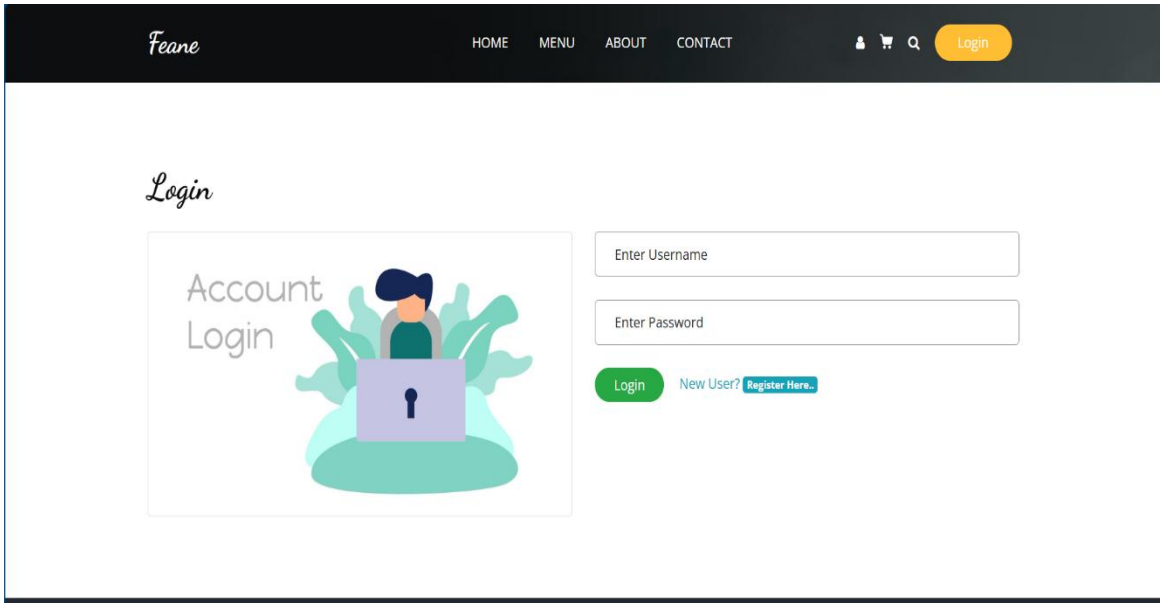


Figure 3. 2 Interface design for user login

The above diagram is the interface design of the order user login page which takes user inputs and login to user side.

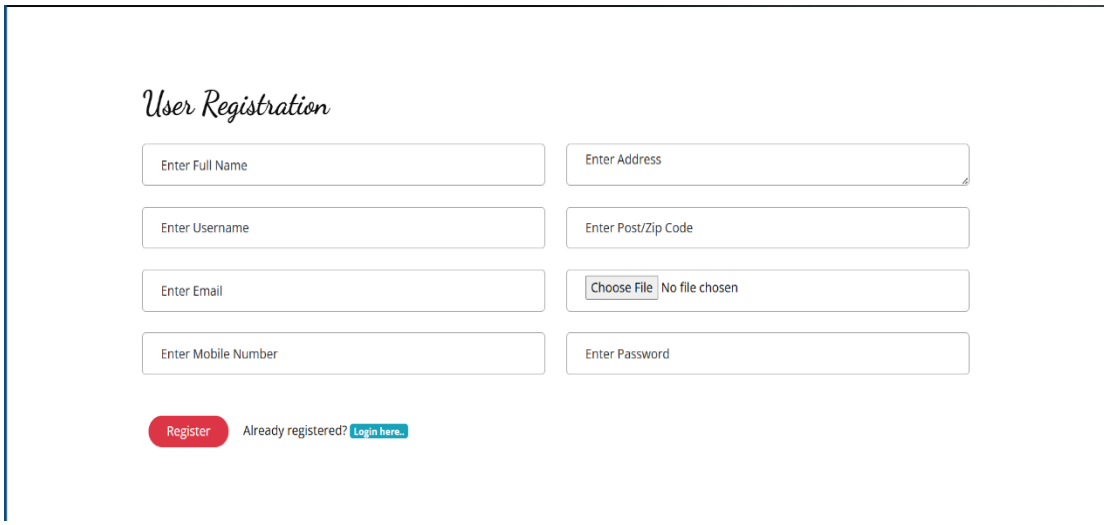


Figure 3. 3 Interface design for signup

The above diagram is the interface design for the login and sign-up page box which takes username and password field as input.

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

Integrated Development Environment:

- **DotNet**: Used as a main IDE for backend development.
- **Visual Studio** Used for developing frontend development of application.

Diagram tool:

- **Draw.io:** Used for making ER models, User Interface, Database schema and System architecture.

Backend:

- **C#:** Used as a primary language for application backend logic.

Frontend:HTML,CSS

- **Bootstrap:** Used for rendering the dynamic interface.
- **CSS:** used for all styling of webpages.

Database platform:

- **Sql Server**: Used as a database for storing data.

Documentation Tool:

- **MS Word:** Used as a documentation tool.

CASE Tools:

- **Draw.io:** Used to design system diagrams including flowcharts, ER diagrams, and architecture.
- **MS Word:** Used for creating and compiling the full project documentation.

4.2 Testing

4.2.1 Test Cases for Unit Testing

Test Cases for Signup:

T01: Check user signup with valid data

T02: Check user signup with invalid data

T03: Check user signup with null data

Table 4. 1 Test table for signup

ID	Description	Test Data	Expected Results	Actual Results	Pass/Fail
T01	Check user signup with valid data	Username= Sujal123 Password = password	User should register.	Registered Successful!	Pass
T02	Check user signup with invalid data	Username= Ram Password = pass269	User should not register.	Username invalid!	Pass
T03	Check user signup with null data	Username=Null Password = Null	User should not register.	Username required!	Pass
T04	Check user signup with same data	Username=Don Password = password	User should not register.	User already exists!	Pass

Post-condition: After successful registration, user credential is stored in database.

Test Cases for Login:

T05: Check user login with valid data

T06: Check user login with invalid data

T07: Check user login with invalid password

Table 4. 2 Test table for login

ID	Description	Pre-conditions	Test Data	Expected Results	Actual Results	Pass/Fail
T05	Check customer login with valid data	User is registered.	Username= hari Password = password	User should login into an application	Login successful!	Pass
T06	Check user login with invalid username	User is not registered.	Username =dumbcom Password = rick	User should not login.	User not found!	Pass
T07	Check user login with invalid password	User is registered.	Username= bikash Password= Null	User should not login.	Invalid Password!	Pass

Post-condition: After successful login, id and username are stored in session.

Test cases for Logout:

T08: Check user clicks logout button

Table 4. 3 Test table for logout

ID	Description	Expected Results	Actual Results	Pass/Fail
T08	Check user clicks logout button	User logged out of application.	Redirected to homepage	Pass

Pre-condition: User is already logged in.

Post-condition: After successful logout, user session is cleared.

Test cases for placing order:

T09: Check user adds with valid order

T10: Check user places order with invalid data

Table 4. 4 Test table for order

ID	Description	Test Data	Expected Results	Actual Results	Pass/Fail
T09	Check user places valid order	Order request data	Order should be placed.	Order placed!	Pass
T10	Check user inputs invalid order	Order request data	Order should not be placed.	Order rejected!	Pass

Pre-condition: User is already logged in.

Post-condition: After successful order, user orders is placed.

Test cases for Update:

T10: Check user updates with invalid password

T11: Check user updates with valid password

Table 4. 5 Test table for update

ID	Description	Test Data	Expected Results	Actual Results	Pass/Fail
T10	Check user updates with invalid password	Password = nsudffd	User detail should not be updated.	Update successful!	Pass
T11	Check user updates with valid password	Password = password	User detail should be updated.	Password Invalid!	Pass

Pre-condition: User is already logged in whose valid password is “password”.

Post-condition: After successful update, user credentials are updated in database.

Test cases for Delete:

T12: Check user deletes with invalid password

T13: Check user deletes with valid password

Table 4. 6 Test table for delete

ID	Description	Test Data	Expected Results	Actual Results	Pass/Fail
T12	Check user deletes with invalid password	Password = nsudd	User detail should not be deleted.	Update successful!	Pass
T14	Check user deletes with valid password	Password = password	User detail should be deleted.	Password Invalid!	Pass

Pre-condition: User is already logged in whose valid password is “password”.

Post-condition: After successful delete, user credentials are deleted from database.

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Lesson Learnt/Outcome

1. Food Ordering System is an ensure a user-friendly interface to make ordering easy and intuitive for customers.

2. Regularly update and manage the menu to reflect accurate prices, availability, and options.

3. Prioritize Order accuracy to minimize errors and customer dissatisfactions.

4. Seamlessly integrate with payment gateway, delivery services.

5.2 Conclusion

In conclusion, the Online Food Ordering System is viable and promising from technical, economic, and operational perspectives. With accessible technology, clear revenue potential, and growing market demand, the project is well-positioned for success. Proceeding with development is recommended to capitalize on the increasing preference for convenient online food ordering. Proper planning and effective marketing will be key to gaining a competitive edge and ensuring long-term sustainability. Continuous monitoring and regular updates will help the system adapt to changing customer needs and technological advancements.

5.3 Future Recommendations

To ensure long-term success, the system should continuously evolve by adding features like AI-based personalized recommendations and real-time order tracking. Expanding to support multiple languages and integrating with emerging payment methods will improve accessibility and user convenience. Building partnerships with more restaurants and delivery services can widen the service area. Finally, investing in data analytics will help understand customer behavior and optimize operations.

References

[1] V. Kanade, R. Adithya, S. Abhishek and p. Salman, "Online Food Ordering System," International Journal of Computer Applications, pp. 22-24, 2017.

[2] O. KAMBLE, "Online Food Ordering System," Advanced Research in Computer and Communication Engineering, pp. 14-16, May 2019.

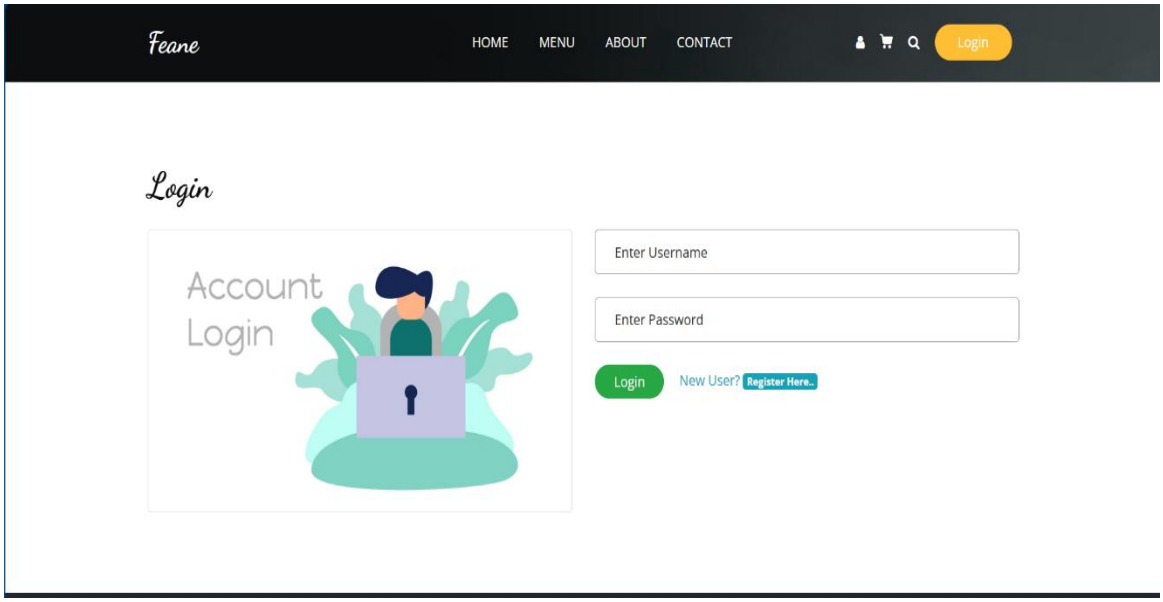
[3] R. R. R. J. B. A. S. C. P. Trupthi B, "Online Food Ordering System," Recent Technology and Engineering (IJRTE), pp. 834-836, July 2019.

[4] A. S. Sarika R. Rathil, "Online Food Ordering System, Student, Dept of Computer Engineering, MGM's Polytechnic, Maharashtra, India, pp. 2524-2551, march,2021.

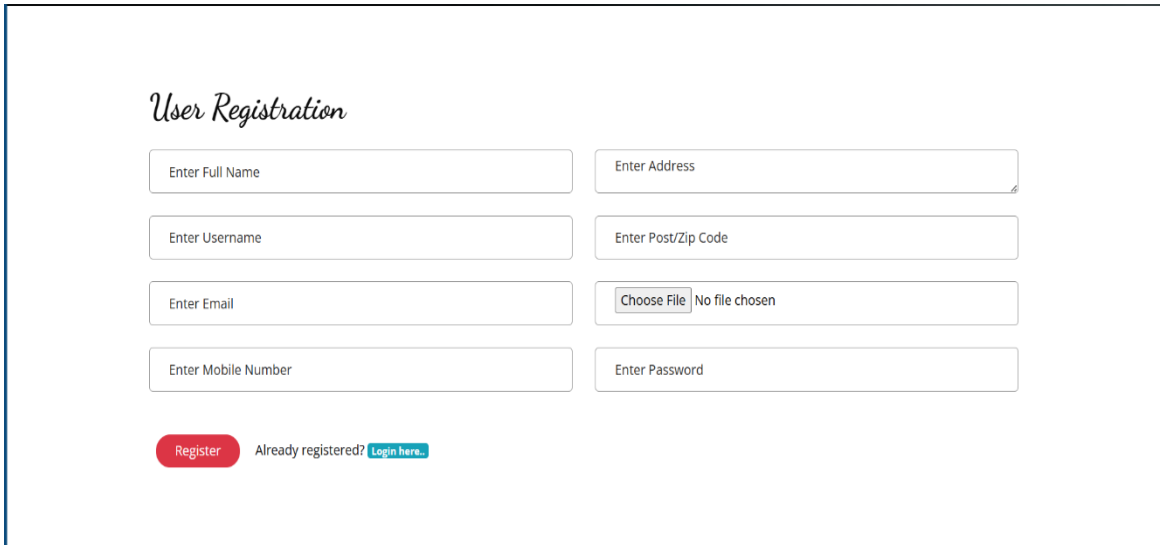
APPENDICES

Screenshots

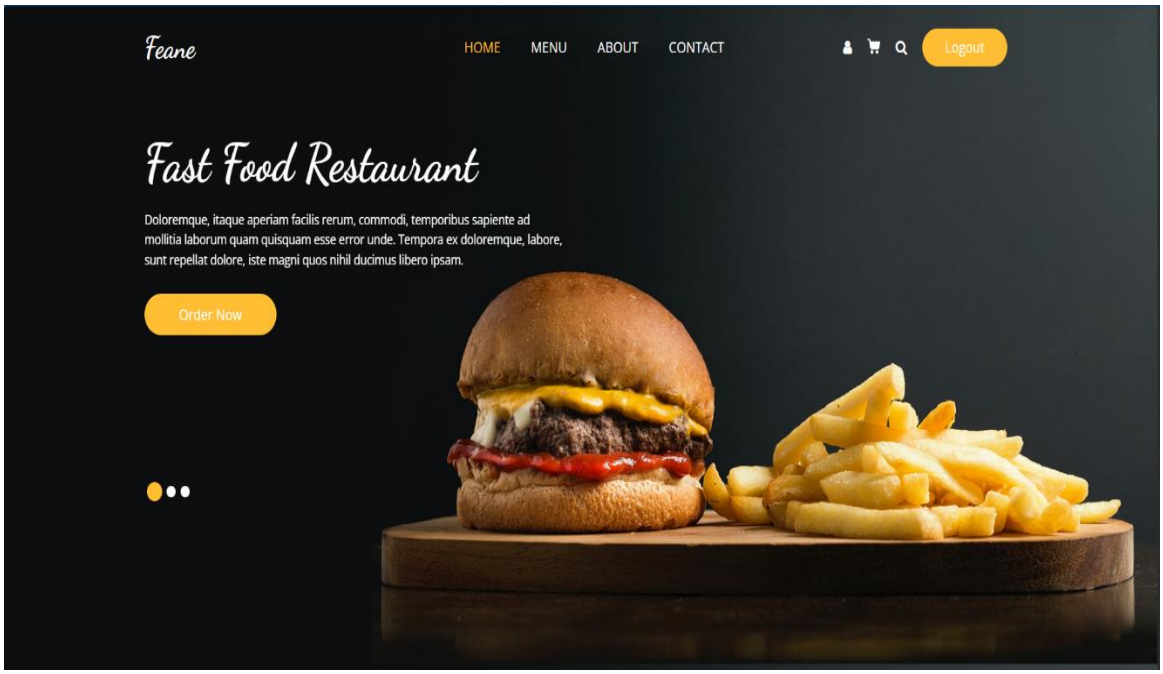
Login Page



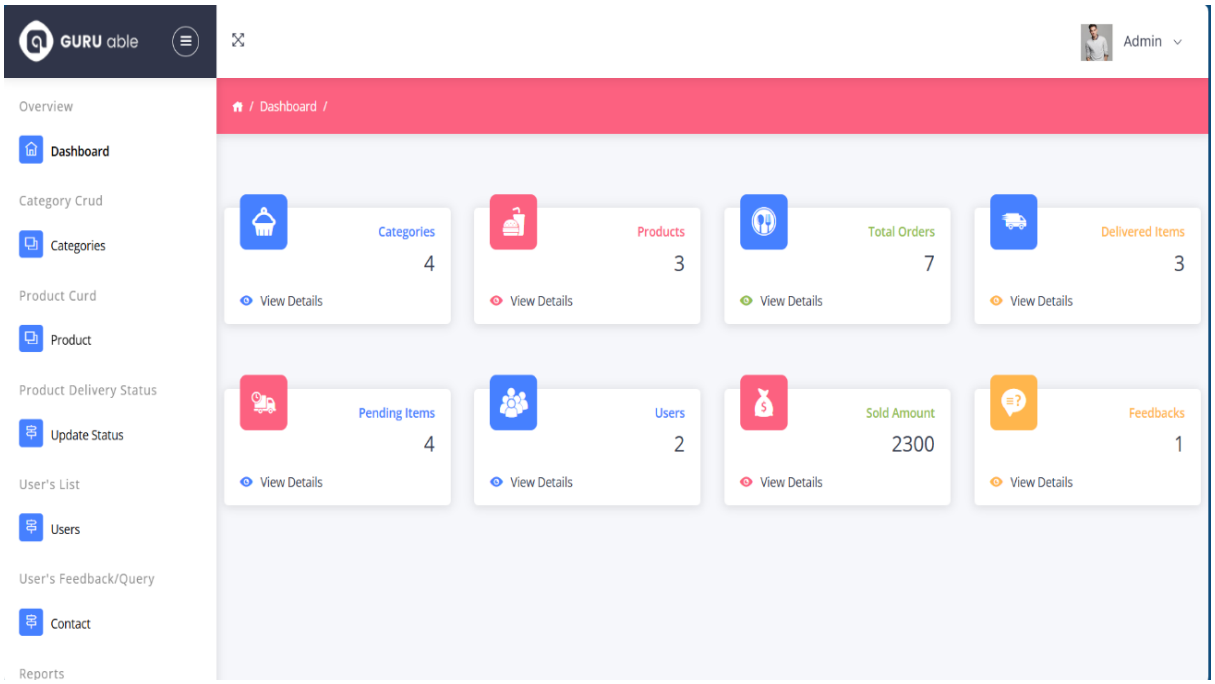
Register

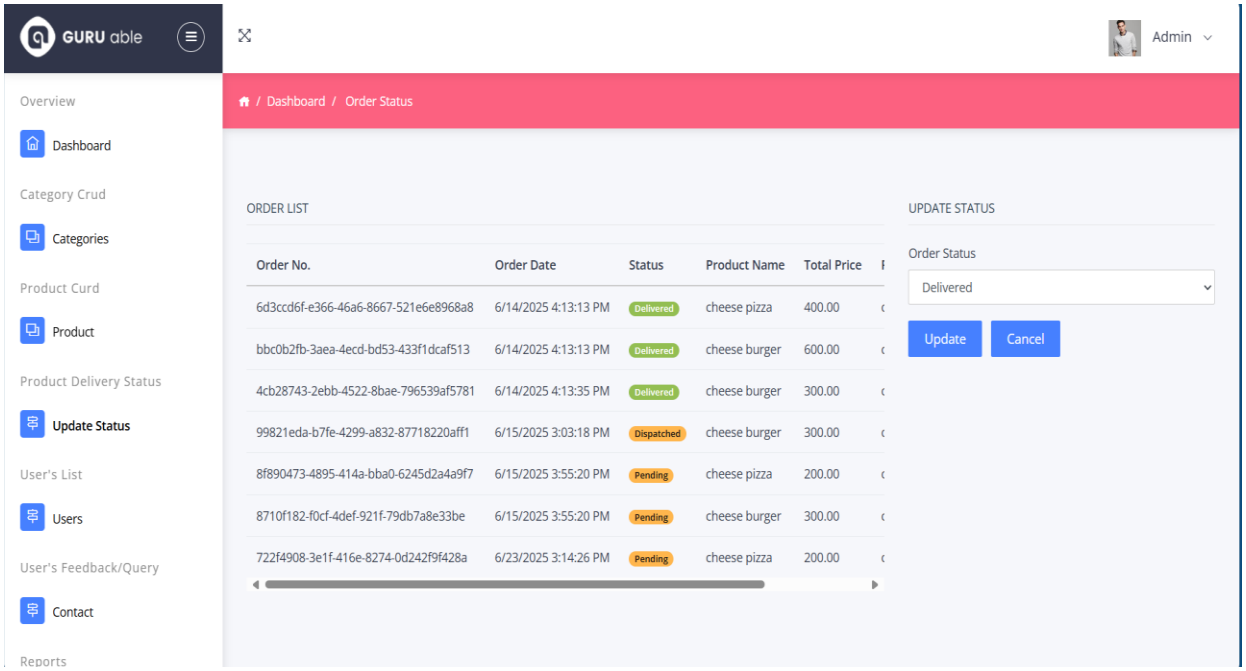


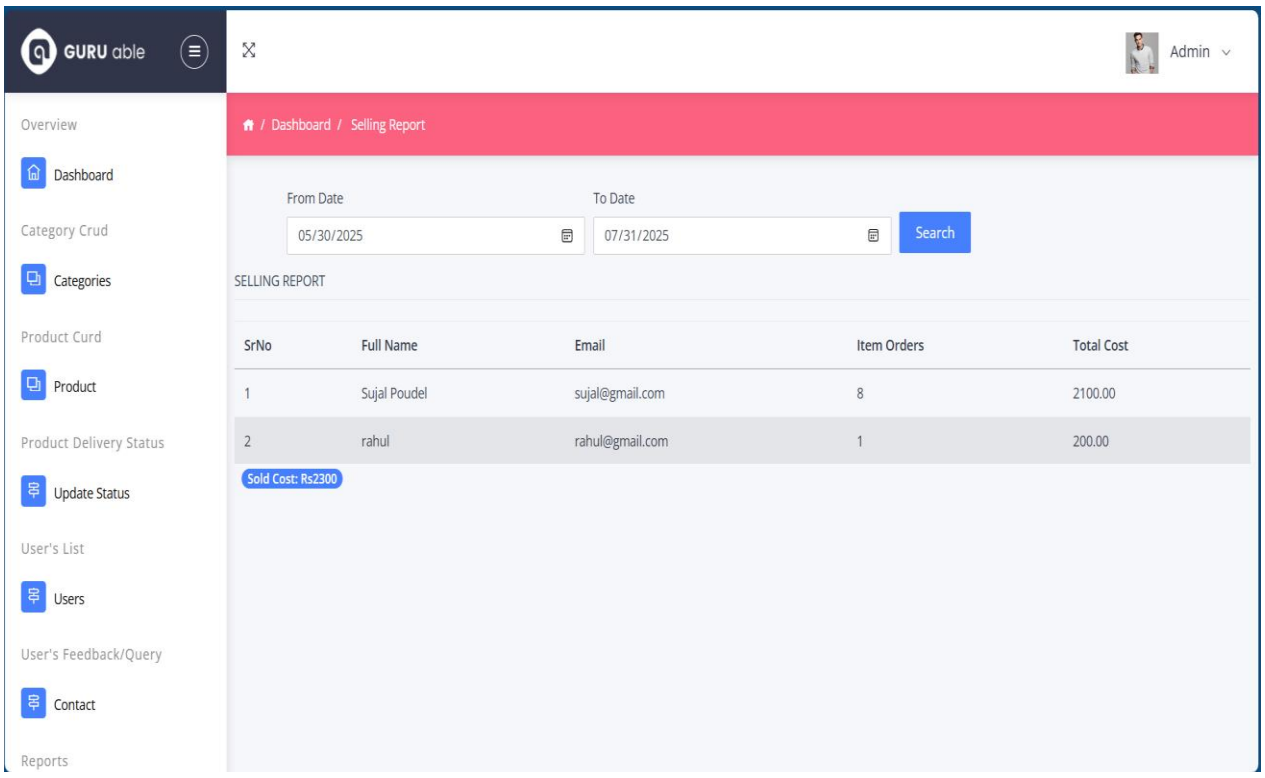
Default



Admin




GURU able
Admin
Overview
Dashboard
Category Crud
Categories
Product Curd
Product
Product Delivery Status
Update Status
User's List
Users
User's Feedback/Query
Contact
Reports
/ Dashboard / Order Status
ORDER LIST
Order No.
Order Date
Status
Product Name
Total Price
6d3ccd6f-e366-46a6-8667-521e6e8968a8
6/14/2025 4:13:13 PM
Delivered
cheese pizza
400.00
bbc0b2fb-3aea-4ecd-bd53-433f1dcaf513
6/14/2025 4:13:13 PM
Delivered
cheese burger
600.00
4cb28743-2ebb-4522-8bae-796539af5781
6/14/2025 4:13:35 PM
Delivered
cheese burger
300.00
99821eda-b7fe-4299-a832-87718220aff1
6/15/2025 3:03:18 PM
Dispatched
cheese burger
300.00
8f890473-4895-414a-bba0-6245d2a4a9f7
6/15/2025 3:55:20 PM
Pending
cheese pizza
200.00
8710f182-f0cf-4def-921f-79db7a8e33be
6/15/2025 3:55:20 PM
Pending
cheese burger
300.00
722f4908-3e1f-416e-8274-0d242f9f428a
6/23/2025 3:14:26 PM
Pending
cheese pizza
200.00
UPDATE STATUS
Order Status
Delivered
Update
Cancel


GURU able
Admin
Overview
Dashboard
Category Crud
Categories
Product Curd
Product
Product Delivery Status
Update Status
User's List
Users
User's Feedback/Query
Contact
Reports
/ Dashboard / Selling Report
From Date
05/30/2025
To Date
07/31/2025
Search
SELLING REPORT
SrNo
Full Name
Email
Item Orders
Total Cost
1
Sujal Poudel
sujal@gmail.com
8
2100.00
2
rahul
rahul@gmail.com
1
200.00
Sold Cost: Rs2300