KIGALI INDEPENDENT UNIVERSITY

ULK SCHOOL OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

P.o Box: 2280 KIGALI

ONLINE FOOD ORDERING SYSTEM. CASE STUDY: MAALE RESTAURANT

By: CHUOL YUANES TOANG CHUOL

ROLL NUMBER: 202110385

Supervisor name: RUTARINDWA JEAN PIERRE

Dissertation Submitted to the School of Science and Technology in Partial Fulfilment of the requirements for the award of Bachelor's Degree in Computer Science

SEPTEMBER, 2024

Declaration

This dissertation titled Online Food ordering system is my original work, it has never been submitted before for any other degree award to any other University ‟.

Student Name: **CHUOL YUANES TOANG CHUOL**

Signature ……..

Date...../...../2024

Approval

This thesis/dissertation titled **ONLINE FOOD ORDERING SYSTEM** has been done under my supervision and submitted for examination with my approval

Supervisor Name: **RUTARINDWA JEAN PIERRE**

signature. …….

Date……/……/ 2024

DEDICATION

This work is dedicated, first and foremost, to God, whose guidance and strength have been my foundation throughout this journey.

I also dedicate this to my uncle, whose generosity and support in paying my school fees made this education possible.

Finally, to my family, my mother and father who have been my pillars of love and encouragement, always believing in me and motivating me to achieve my best.

Thank you all for your unwavering support and belief in my dreams.

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to Almighty God for granting me the strength, wisdom, and perseverance to successfully complete this research. Without His guidance, this work would not have been possible.

I am profoundly thankful to the founder of ULK **Prof. Dr Rwrigamba Balinda** for establishing a platform that promotes quality education and research. Your vision continues to inspire us to strive for excellence.

A special thank you to my supervisor **RUTARINDWA JEAN PIERRE**, whose unwavering support, valuable insights, and constructive feedback have been instrumental in guiding this research to completion. Your expertise and dedication have been essential throughout this journey.

I extend my sincere appreciation to the respondents who took time from their busy schedules to provide valuable information for this research. Your input has greatly contributed to the success of this study.

To the academic and administrative staff of ULK, your support and dedication to the growth of students have played a crucial role in shaping my academic journey. I am deeply grateful for your efforts in creating a conducive learning environment.

Finally, I would like to express my heartfelt gratitude to my family and friends for their encouragement, understanding, and unwavering support throughout my academic endeavors. You have been my pillars of strength throughout this journey, and I am eternally thankful for your belief in me.

Table of Contents

List of Tables

List of Figures

ABBREVIATIONS AND ACRONYMS

AJAX: Asynchronous JavaScript and XML

CSS: Cascading Style Sheets

DBMS: Database Management System

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

JS: JavaScript

JSON: JavaScript Object Notation

MySQL: My Structured Query Language (A popular open-source relational database management system)

PHP: Hypertext Preprocessor (originally Personal Home Page)

SQL: Structured Query Language

URL: Uniform Resource Locator

UX: User Experience

ABSTRACT

This dissertation presents the design and implementation of an online food ordering system aimed at improving the efficiency and customer experience at Maale Restaurant. The system was developed using modern web technologies including HTML, CSS, JavaScript, PHP, and MySQL. It provides an intuitive user interface that allows customers to easily browse the menu, place orders, and manage their profiles. The system also offers robust administrative functions such as order management, and reporting.

The project is driven by the need to address operational inefficiencies at the restaurant, such as long wait times, order inaccuracies, and a lack of real-time order tracking. By automating key processes, the system reduces errors, enhances customer satisfaction, and streamlines restaurant operations. A Cash on Delivery payment method was implemented to align with the current payment practices at the restaurant.

Comprehensive testing, including unit, integration, and acceptance testing, was conducted to ensure the system meets functional and non-functional requirements. The system was evaluated for performance, security, and compatibility, and successfully met all expected standards. Additionally, the administrative interface allows restaurant staff to manage orders efficiently, update status, and generate detailed reports for business insights.

This research concludes that the implemented system significantly enhances the operational efficiency of Maale Restaurant, offering a scalable solution that can be extended in the future to incorporate additional payment options and advanced order tracking features. Recommendations for future work include expanding the system to integrate with mobile applications and adding customer feedback functionality for continuous improvement**.**

Keywords: Online food ordering system, customer experience, restaurant operations, Maale Restaurant, HTML, CSS, JavaScript, PHP, MySQL, order management, Cash on Delivery, system efficiency, reporting, real-time tracking, scalability.

CHAPTER ONE: GENERAL INTRODUCTION

1.0. Introduction

The proliferation of digital technologies has fundamentally transformed various industries, and the culinary sector is no exception. Online food ordering systems have emerged as a disruptive force, reshaping the way people engage with food services worldwide (Smith, 2019; Johnson, 2020).

In the context of South Sudan, a country marked by cultural diversity and rapid urbanization, the adoption of such systems presents both challenges and opportunities (Jones & Williams, 2020).

South Sudan, with its capital city Juba as the focal point, is experiencing significant demographic shifts and economic growth. Urbanization is leading to changes in lifestyle and consumer preferences, including the demand for convenient dining options. However, the country faces infrastructural constraints and logistical challenges, which can hinder traditional food distribution channels (Brown, 2021).

Against this backdrop, the introduction of online food ordering systems in Juba holds promise for addressing these challenges while catering to the evolving needs of consumers. By leveraging digital platforms and mobile technology, these systems offer a convenient and efficient way for residents to access a diverse range of cuisines from local restaurants and eateries (White & Black, 2022; Garcia, 2023).

This research aims to investigate the implementation and impact of online food ordering systems in Juba, South Sudan. Through a comprehensive analysis of case studies and existing literature, the study will examine the challenges and opportunities associated with the adoption of online food ordering platforms in the region. By focusing on Juba as a case study, the research seeks to provide valuable insights into the role of technology in shaping the culinary landscape of South Sudan and to inform strategies for the development and expansion of online food ordering systems in similar contexts.

This chapter mainly discusses the background of the study, the problem statement, the objectives, Scope of the study, Significance of the study Methodology, Ethical consideration and Limitations of the study.

1.1. Background to the study

In recent years, the global food industry had witnessed a transformative shift towards online food ordering systems. This trend, driven by advancements in technology and changing consumer behaviors, had seen significant adoption in many parts of the world (Jones & Livingstone, 2020). However, Juba, the capital of South Sudan, remained largely untouched by this digital revolution. The then-current state of food delivery services in Juba presented a unique set of challenges and opportunities, highlighting the urgent need for a comprehensive study to understand and address the gaps in the existing system (Keen, 2019).

Juba's food service sector had been predominantly characterized by traditional, offline operations. Restaurants and eateries relied heavily on walk-in customers and phone orders, with minimal integration of digital platforms (Akech, 2018). The city's infrastructure posed significant challenges to the implementation of modern food delivery systems. Poor road conditions, limited internet connectivity, and unreliable power supply were major barriers to efficient service delivery (United Nations Development Programme, 2021). Additionally, the socioeconomic conditions in Juba contributed to a low penetration of smartphones and internet usage among the population, further impeding the adoption of online food ordering (World Bank, 2019).

Despite these challenges, there had been a discernible shift in consumer behavior in Juba. With a growing expatriate community and an increasing number of local consumers who were becoming more tech-savvy, there was a rising demand for convenience-oriented services, including food delivery (Johnson, 2020). Urban residents, especially those in the middle and upper-income brackets, were beginning to seek the convenience of ordering food online. This latent demand represented a significant market opportunity that remained largely untapped (Gurtong, 2020).

Infrastructure Deficiencies: The most glaring gap had been the lack of adequate infrastructure to support an online food ordering system. Poor road networks and unreliable electricity and internet services created a challenging environment for both service providers and consumers. These infrastructural limitations necessitated innovative solutions tailored to the local context (UNDP, 2021).

The penetration of smartphones and the internet had been relatively low in Juba compared to global standards. This technological gap limited the potential customer base for online food ordering services. Furthermore, many local businesses lacked the technical expertise and resources to develop and maintain digital platforms (World Bank, 2019).

The regulatory framework in South Sudan had not been well-developed, particularly concerning e-commerce and digital transactions. This regulatory ambiguity created uncertainty for businesses looking to invest in online food ordering systems. Additionally, the economic instability and high cost of doing business in Juba were significant deterrents for potential investors (African Development Bank, 2018).

There had been a noticeable lack of consumer trust and awareness regarding online transactions in Juba. Many potential customers were wary of digital payment methods and concerned about the reliability of online services. Building trust through secure, reliable, and user-friendly platforms was essential for the success of any online food ordering system (Johnson, 2020).

Internationally, the food and beverage industry has undergone significant transformation in recent years, driven by technological advancements and changing consumer preferences. One notable trend is the growing popularity of online food ordering systems, which allow customers to place orders conveniently from their mobile devices or computers (Mintel, 2021). This shift towards digital dining experiences has been accelerated by factors such as increased internet penetration, rising smartphone usage, and changing lifestyles characterized by busy schedules and a preference for convenience.

Across the globe, restaurants and eateries are increasingly adopting online food ordering systems to streamline operations, enhance customer satisfaction, and gain a competitive edge in the market. These systems offer benefits such as faster order processing, reduced errors, and improved efficiency, leading to higher customer retention and increased revenues. As a result, online food ordering has become a standard feature of modern dining experiences, shaping consumer expectations and driving innovation in the food service industry (Statista, 2021).

1.2. Problem Statement

Despite its reputation and popularity, Maale Restaurant has been experiencing challenges in managing customer orders, leading to longer wait times, increased errors in order processing, and customer dissatisfaction. The traditional method of taking orders in person is becoming increasingly inefficient, especially during peak hours. This inefficiency not only affects the restaurant's operational efficiency but also negatively impacts the overall customer experience.

The primary problem that this research seeks to address is the lack of an integrated online food ordering system at Maale Restaurant, which is leading to operational inefficiencies and customer dissatisfaction. Specifically, the research aims to address the following aspects:

Operational Efficiency: How the current order-taking process impacts the restaurant's operational efficiency, including the time taken to process orders, frequency of order errors, and staff workload.

1.3. Research Objectives

1.3.1 **General Objective:**

To design and implement an efficient and user-friendly Online Food Ordering System for Maale Restaurant that enhances operational efficiency, improves customer satisfaction, and positions the restaurant as a competitive player in the food industry.

1.3.2. Specific Objectives:

i. To Design a database to store all necessary information, including user profiles, menu items, orders, payment details, and order history. Tables.
ii. To design a user-friendly and intuitive user interface for Maale Restaurant's online food ordering system, ensuring ease of navigation and seamless user experience.
iii. To integrate secure and efficient payment gateways into the online food ordering system, facilitating smooth and reliable transactions for customers.
iv. To develop a robust order management system that streamlines order processing, minimizes errors, and optimizes workflow for Maale Restaurant staff.

1.3.3. Research Questions

i. How can a database be designed to efficiently store all necessary information, such as user profiles, menu items, orders, payment details, and order history for Maale Restaurant?
ii. What are the key design principles for creating a user-friendly and intuitive interface that ensures ease of navigation in Maale Restaurant's online food ordering system?
iii. What are the most secure and efficient payment gateways that can be integrated into Maale Restaurant's online food ordering system to facilitate smooth and reliable transactions?
iv. What features and functionalities should an order management system have to streamline order processing, minimize errors, and optimize workflow for Maale Restaurant staff?

1.4. Scope of the study

1.4.1. Content scope

The factors and variables to be considered in the study include:

System Design and Features: The design of the online food ordering system, focusing on user interface design, integration of payment gateways, and development of an efficient order management system.

Implementation Process: Steps involved in implementing the system at Maale Restaurant, including technical setup, staff training, and customer onboarding.

User Feedback and Testing: Methods for conducting usability testing, collecting user feedback, and identifying areas for improvement to ensure high levels of user satisfaction.

Operational Impact: Evaluation of the system's impact on Maale Restaurant's operations, including changes in order processing times, error rates, and overall workflow efficiency.

Customer Experience: Analysis of customer satisfaction with the online food ordering system.

1.4.2. Geographical Scope:

The study focuses on Maale Restaurant, located in the Jebel area of Juba, South Sudan. This geographical scope is selected due to the strategic importance of the Jebel area as a vibrant and growing community within Juba, where there is a significant demand for dining services and potential for digital innovation.

1.4.3. Time Scope

The research was conducted within the context of Maale Restaurant's significant expansion and the increasing demand for online services between 2019 and 2023. This period was marked by the restaurant's rapid growth and the subsequent need to adopt digital solutions to manage the higher volume of orders and enhance customer satisfaction. This timeframe was crucial for capturing the restaurant's transition from traditional order-taking methods to a more efficient online food ordering system, ensuring that the research addressed relevant and timely operational challenges and opportunities.

1.5. Significance of the study.

1.5.1. Personal Interest:

The development of an online food ordering system for Maale Restaurant presents a unique opportunity for the researcher to deepen their technical knowledge and practical experience in various aspects of software development and systems integration. Through this project, the researcher will gain valuable hands-on skills in designing and implementing databases, creating intuitive user interfaces, and integrating secure payment gateways. This experience is not only critical for personal growth but also for professional development, as it allows the researcher to apply theoretical knowledge in a real-world context, bridging the gap between academic learning and practical application. The challenges encountered during the project—such as ensuring data security, optimizing user experience, and managing large volumes of orders—will enhance problem-solving abilities and technical proficiency, equipping the researcher with expertise that can be applied in future projects. Ultimately, this project will serve as a stepping stone toward career advancement in the fields of information technology, e-commerce, and digital transformation, opening doors to new professional opportunities and specialized roles.

1.5.2. Institutional Interest:

For Maale Restaurant, the implementation of an online food ordering system offers numerous strategic advantages that align with its goals of improving operational efficiency and enhancing customer satisfaction. By digitizing the order management process, the restaurant can streamline its operations, reducing manual errors and speeding up service delivery, particularly during peak hours. This efficiency boost allows the restaurant to handle higher volumes of orders without compromising service quality, leading to smoother operations and less strain on staff. Additionally, the system's user-friendly design will enhance the customer experience by providing a convenient and seamless way to browse the menu, place orders, and make secure payments. Customers are likely to appreciate the ease of use, which will translate into higher satisfaction levels and increased customer loyalty. From a business perspective, adopting this technology also provides a competitive edge in the local market, positioning Maale Restaurant as an innovative and modern establishment. This can help attract a broader customer base, increase order volumes, and ultimately drive revenue growth, reinforcing the restaurant's brand and market position.

1.5.3. Public Interest:

Beyond the benefits to Maale Restaurant and the researcher, the implementation of an online food ordering system holds broader implications for the local community and economy. For customers, the system introduces a convenient and efficient way to place orders, reducing the time and effort required to dine at Maale Restaurant. This meets the growing demand for digital services in the food and beverage industry, especially as consumers increasingly prefer the convenience of online transactions. More broadly, the successful adoption of such technology can serve as a model for other local businesses, encouraging the wider adoption of digital systems and contributing to the growth of the local economy. As more restaurants and service providers embrace online ordering and payment systems, there will likely be an increase in job creation within both the tech and food delivery sectors. Moreover, the promotion of digital innovation in the local market can attract investors and technology providers, fostering an environment that supports further technological advancement and entrepreneurship. In this way, the project can stimulate technological growth in Juba, South Sudan, helping to modernize the local business landscape and contribute to the region's long-term economic development.

1.6. Software Development Methodology

The methodology for the project titled Online Food Ordering System employed a combination of data collection techniques and structured development approaches to ensure a smooth and efficient implementation. Data was gathered using documentation and observation methods, where existing sales records, customer feedback, and operational manuals were reviewed to identify key trends, while direct observation of the restaurant's operations provided insights into the current workflow and challenges. For the software development process, the Agile Model was utilized, specifically a simplified version tailored to the project's scope and timeline. This involved breaking the work into short, iterative sprints based on Scrum principles, where manageable tasks (user stories) were completed incrementally, allowing for continuous improvement and regular feedback from stakeholders. Additionally, the Structured Systems Analysis and Design Methodology (SSADM) was employed to guide the system's analysis and design. This began with a feasibility study to assess the viability of the online system, followed by requirements analysis to determine the necessary features. The system architecture was then designed, including the user interface, database, payment gateway integration, and order management functionalities. The system was developed and tested to ensure it met performance requirements and was evaluated using user

feedback to identify areas for further refinement. This comprehensive approach ensured that the system was both functional and aligned with the project's objectives.

1.7. Ethical Considerations:

Throughout the dissertation research and software development process, ethical considerations were paramount to ensure the integrity and fairness of the study. Key ethical considerations included:

- **Avoidance of Bias:** Steps were taken to minimize bias in data collection, analysis, and interpretation. Questions in surveys and interviews were designed to be neutral and unbiased, and efforts were made to include diverse perspectives and voices in the research. Data analysis was conducted objectively, and findings were presented accurately, without manipulation or distortion.
- **Transparency and Accountability:** Throughout the research process, transparency and accountability were maintained by documenting procedures, methods, and decisions. Any conflicts of interest or potential sources of bias were disclosed, and steps were taken to mitigate their impact on the research outcomes. The dissertation adhered to academic standards of honesty, integrity, and intellectual rigor.

1.8. Limitations

While every effort was made to conduct the research and software development project with rigor and accuracy, several limitations should be acknowledged:

1. **Resource Constraints:** As a student researcher with limited resources, including time, budget, and access to advanced technologies, there were constraints on the scope and depth of the research and software development efforts. This may have impacted the comprehensiveness and sophistication of the final online food ordering system prototype.
2. **Subjectivity and Bias:** Despite efforts to minimize bias and subjectivity, there may have been inherent biases in the data collection, analysis, and interpretation processes. Personal perspectives, experiences, and preconceptions could have influenced the research outcomes, potentially leading to skewed results or conclusions.

Technological Limitations: The software development project was constrained by the student researcher's technical skills and expertise, as well as the available software development tools and platforms. This may have limited the complexity and functionality of the online food ordering system prototype compared to commercially available solutions.

3. **External Factors**: External factors, such as changes in market conditions, regulatory requirements, or unexpected events (e.g., global pandemics), could have influenced the research outcomes and the feasibility of implementing the online food ordering system at Maale Restaurant. These factors were beyond the control of the researcher and may have introduced uncertainties or limitations.
4. **Time Constraints:** The project was conducted within a relatively short timeframe, which may have restricted the depth of analysis and the opportunity for thorough testing and validation of the online food ordering system prototype. This could have impacted the robustness and reliability of the final solution.

CHAPTER TWO: LITERATURE REVIEW

2.0. Introduction

This chapter reviews the existing literature on online food ordering systems, focusing on their development, implementation, and impact on the food and beverage industry. The chapter defines key concepts, analyzes similar systems, identifies gaps in current solutions, and describes how the proposed system for Maale Restaurant will address these gaps while introducing new innovations. The review aims to provide a comprehensive understanding of the current state of online food ordering systems and establish a foundation for the development of an effective solution for Maale Restaurant.

2.1. Definition of Concepts

To ensure clarity and a common understanding, the following key terms relevant to this research are defined:

2.1.1. Online Food Ordering System

A digital platform that allows customers to place food orders from a restaurant via the internet using a web or mobile application. It typically includes features for menu browsing, order placement, payment processing, and order tracking (Smith & Khosrowpour, 2021).

2.1.2. User Interface (UI)

The part of the online food ordering system that users interact with, including visual elements like buttons, menus, forms, and icons. A well-designed UI is crucial for providing a seamless and intuitive user experience (Johnson, 2018).

2.1.3. Usability Testing

The process of evaluating a product by testing it with representative users to gather feedback on its ease of use and overall user experience. Usability testing helps identify areas for improvement and ensures that the system meets user needs (Nielsen, 2012).

2.1.4. Database Design

The process of creating a detailed data model of a database, which includes the logical and physical design of the database structure. This involves defining the data elements and their relationships,

ensuring data integrity, and optimizing performance. A well-designed database supports efficient data storage, retrieval, and management (Elmasri & Navathe, 2017).

2.1.5. Order Management System (OMS)

Software that manages and tracks orders from customers, ensuring that orders are processed accurately and efficiently. An effective OMS integrates with the restaurant's operations to optimize workflow and reduce errors (Doe & Smith, 2020).

2.1.6. Payment Gateway

A service that processes credit card payments and other forms of electronic transactions for online and traditional brick-and-mortar stores, ensuring secure and efficient transactions. Examples include PayPal, Stripe, and Square (Williams & Shapiro, 2019).

2.2. Other related literatures

2.2.1. Literatures related to database

2.2.1.1. Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. Databases are managed using a database management system (DBMS). Collectively, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system. According to Elmasri and Navathe (2016), a database is designed to manage and store large amounts of information efficiently and securely.

A well-structured database is fundamental to the success of an online food ordering system. Existing literature emphasizes the importance of designing a database that can efficiently store and manage user profiles, menu items, orders, payment details, and order history. According to Connolly and Begg (2015), a relational database management system (RDBMS) is ideal for such applications due to its ability to handle large volumes of data and support complex queries. Furthermore, Mookiah (2020) highlights the significance of database normalization in reducing data redundancy and improving data integrity.

2.2.1.1.1 Characteristics of a Database

- **Data Integrity:** Ensures that data is accurate and consistent over its lifecycle. Techniques such as normalization help in maintaining data integrity by organizing the data to reduce redundancy.
- **Data Security:** Protects the data from unauthorized access and breaches. This is crucial for sensitive information such as payment details and personal user data (Korth & Silberschatz, 2019).
- **Data Scalability:** The ability to handle increasing amounts of data. As the amount of data grows, the database should be able to scale without a loss in performance (Connolly & Begg, 2015).
- **Data Redundancy:** Minimization of duplicate data. Normalization and other techniques help reduce redundancy to ensure efficient storage and retrieval of data (Mookiah, 2020).
- **Data Consistency:** Ensures that data is the same across the database, which is particularly important for transaction management and accurate reporting.
- **Data Access:** The database should allow efficient and quick access to data through various querying techniques.
- **Backup and Recovery:** Mechanisms to recover data in case of failure. This ensures the continuity and reliability of the database system.

2.2.1.1.2 Type of Database

- **Relational Databases:** Use tables (relations) to store data. These databases use Structured Query Language (SQL) for defining and manipulating data. Examples include MySQL, PostgreSQL, and Oracle (Elmasri & Navathe, 2016).
- **NoSQL Databases:** Designed for unstructured data, these databases are highly scalable and are often used for big data and real-time web applications. Examples include MongoDB, Cassandra, and Redis (Stonebraker, 2010).
- **Distributed Databases:** Spread across multiple physical locations. Distributed databases can be either relational or NoSQL. They ensure data availability and reliability across different locations (Ozsu & Valduriez, 2011).

- **Object-Oriented Databases:** Store data in objects as used in object-oriented programming. This type of database is useful for applications that require complex data relationships (Khoshafian & Abnous, 1995).
- **Cloud Databases:** Delivered as a service via the cloud, these databases offer scalable resources and reduced operational costs. Examples include Amazon RDS, Microsoft Azure SQL Database, and Google Cloud SQL (Mather et al., 2009).

2.2.2 **Literatures related to User-Friendly Interface.**

2.2.2.1. User Interface

A User Interface (UI) is the space where interactions between humans and machines occur. It includes the screens, pages, and visual elements—like buttons and icons—that users interact with to use a digital product or service. The primary goal of a UI is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (Johnson, 2018).

UID should be useful to users. This requires an application that does not just focus on the important parts of users' tasks; it should likewise allow users to interact with the application in ways that are instinctive and normal. Hence, the UID should be simpler yet easy to understand by the users. Friendlier software with constrained abilities is seen to be more usable as the user interface greatly affects the quality of the software product (A. Sajedi, M. Mahdavi, A. Mohammadi, and M. M. Nejad, 2022)

2.2.2.1.1. Types of User Interfaces

- **Graphical User Interface (GUI):** This type of interface allows users to interact with electronic devices using graphical icons and visual indicators as opposed to text-based interfaces, typed command labels, or text navigation. Examples include Windows OS, macOS, and many web applications.
- **Command Line Interface (CLI):** This interface allows users to interact with the computer by typing commands into a console or terminal. Examples include Linux shell and Windows Command Prompt.

- **Touchscreen Interface:** Used primarily in mobile devices and tablets, this interface allows users to interact with the device using touch gestures. Examples include iOS and Android operating systems.
- **Voice User Interface (VUI):** Allows users to interact with the system through voice commands. Examples include virtual assistants like Amazon Alexa, Apple Siri, and Google Assistant.
- **Natural User Interface (NUI):** Interacts with the user through natural means of communication, such as gestures or facial expressions. Examples include gaming consoles like the Xbox Kinect.

2.2.2.1.2. Characteristics of a User-Friendly Interface

- **Simplicity:** A simple interface is free from unnecessary elements, making it easier for users to navigate and find the information they need. A simple UI minimizes cognitive load and reduces the chances of user error (Nielsen, 2012).
- **Consistency:** Consistent design elements and behavior help users understand and predict how the interface will work, thus improving usability. Consistency includes uniformity in fonts, colors, and layout, as well as in interaction patterns (Shneiderman, 2016).
- **Feedback:** The system should provide immediate and clear feedback to users about their actions. This can include visual cues like highlighting buttons when pressed, showing loading indicators, or displaying confirmation messages (Cooper et al., 2014).
- **Accessibility:** The interface should be accessible to users with disabilities. This includes supporting screen readers, providing alternative text for images, and ensuring sufficient contrast between text and background (Lazar et al., 2015).
- **Efficiency:** Users should be able to perform their tasks quickly and with minimal effort. This involves optimizing the workflow, reducing the number of steps required to complete a task, and ensuring fast response times (Dix et al., 2004).
- **Error Handling:** The interface should help users avoid errors, and when errors do occur, it should provide clear instructions on how to fix them. This includes informative error messages and providing undo options (Norman, 2013).

- **Aesthetics:** A visually pleasing interface can enhance user satisfaction and engagement. Aesthetic design involves the thoughtful use of colors, typography, and layout to create an appealing user experience (Tractinsky et al., 2021).
- **Responsiveness:** The interface should work well across different devices and screen sizes. This includes ensuring that the layout adjusts appropriately for mobile devices, tablets, and desktops (Marcotte, 2014).

2.2.3 Literatures related to Integrating Secure and Efficient Payment Gateways

Integrating Secure and Efficient Payment Gateways (ISEPG) refers to the process of incorporating third-party payment processing services into an online platform to facilitate transactions. These gateways handle the complex and secure aspects of payment processing, ensuring that sensitive payment information is encrypted and transferred securely between the customer, the merchant, and the financial institutions involved.

Integrating secure and efficient payment gateways is essential for facilitating smooth and reliable transactions in an online food ordering system. Khosrowpour (2021) highlights that payment security is a major concern for customers, and the use of secure payment gateways such as PayPal, Stripe, and Square can help build trust. Bhasin (2020) indicates that efficient payment processing reduces transaction times and enhances the overall customer experience. Additionally, adherence to the Payment Card Industry Data Security Standard (PCI DSS) is critical for ensuring that payment information is handled securely (PCI Security Standards Council, 2018).

2.2.3.1. Characteristics of Secure and Efficient Payment Gateways

- **Security:** The primary characteristic of a payment gateway is its ability to secure sensitive information. This includes encryption of data during transmission, tokenization of payment information, and compliance with security standards such as PCI DSS (Khosrowpour, 2021).
- **Reliability:** An efficient payment gateway ensures high availability and uptime, providing consistent and uninterrupted service to customers. Reliability also includes accurate transaction processing without errors or delays (Bhasin, 2020).
- **User-Friendliness:** A user-friendly payment gateway offers an intuitive and seamless experience for customers, with easy-to-navigate interfaces and straightforward payment processes (Kumar, 2019).

- **Speed:** Efficient payment gateways process transactions quickly, minimizing wait times for customers and reducing the likelihood of transaction timeouts or failures (Smith & Khosrowpour, 2021).
- **Integration Capabilities:** A good payment gateway can be easily integrated with various platforms and supports multiple payment methods, including credit cards, debit cards, and digital wallets (PayPal, Stripe, Square) (Bhasin, 2020).
- **Customer Support:** High-quality payment gateways offer robust customer support, addressing issues promptly and efficiently, thus ensuring a positive customer experience (PCI Security Standards Council, 2018).

2.2.3.2. Types of Payment Gateways

2.2.3.2.1. Hosted Payment Gateways:

These redirect customers to the payment service provider's (PSP) page to complete the transaction. Examples include PayPal and Stripe. Hosted gateways are easy to implement and maintain high security but can disrupt the user experience due to the redirection (Laudon & Traver, 2020).

2.2.3.2.2. Self-Hosted Payment Gateways:

These allow the merchant to handle the payment process on their website. The payment details are collected and then sent to the PSP for processing. Examples include WooCommerce and Magento. While this provides a seamless user experience, it requires strict compliance with security standards (Chaffey, 2019).

2.2.3.2.3. API/Non-Hosted Payment Gateways:

These integrate directly with the merchant's website via APIs, providing a seamless checkout experience. Examples include Stripe and Authorize.Net. This method offers control over the user experience but demands higher security measures and technical expertise (Mirza, 2019).

2.2.3.2.4. Local Bank Integration:

Some payment gateways are integrated directly with local banks, allowing for direct transfers and payments within the same financial institution. This type of gateway is beneficial in regions where digital payment adoption is still growing (Turban, King, & Lang, 2020).

2.2.3.2.5 Offline Payment Methods:

While not technically payment gateways, offline payment methods are still crucial for many businesses. These include:

- **Cash on Delivery (COD):** Customers pay in cash when the order is delivered to them. This method is still widely used in regions where online payment adoption is low or where customers prefer physical transactions (Stair & Reynolds, 2020).
- **Bank Transfers:** Customers manually transfer funds from their bank account to the merchant's account. This can be done through online banking, mobile banking apps, or in-person at a bank branch (Turban & Volonino, 2017).
- **Checks:** Although increasingly rare, some businesses still accept checks as a form of payment, particularly for large orders or corporate clients (Laudon & Traver, 2020).
- **Mobile Payments:** In some regions, payments are made using mobile credit through telecom operators. Customers can transfer credit from their mobile account to the merchant's account (Chaffey, 2019).

2.2.4 Literatures related to Developing a Robust Order Management System.

An Order Management System (OMS) is a digital tool used to track sales, orders, inventory, and fulfillment. It is designed to streamline the order processing workflow from order creation to delivery, ensuring that orders are processed efficiently and accurately. An OMS typically integrates with various other systems such as payment gateways, inventory management, and customer service platforms to provide a cohesive solution for managing the order lifecycle.

A robust order management system (OMS) is crucial for streamlining order processing and optimizing workflow. Chen and Popovich (2018) discuss how an effective OMS can reduce errors, improve order accuracy, and enhance operational efficiency. Davis (2019) highlights the importance of real-time order tracking and inventory management in providing a seamless order fulfillment process. Integrating OMS with other systems such as customer relationship management (CRM) and enterprise resource planning (ERP) can further enhance efficiency and data consistency (Sharma, 2021).

2.2.4.1. Characteristics of a Robust Order Management System

Real-Time Order Tracking: An effective OMS provides real-time tracking of orders, allowing both the business and the customers to monitor the status of their orders throughout the fulfillment process (Davis, 2019).

Inventory Management: A robust OMS integrates with inventory management systems to ensure that stock levels are accurately tracked and updated, preventing stockouts and overstock situations (Chen & Popovich, 2018).

Error Reduction: By automating many of the order processing tasks, a good OMS minimizes manual errors, improving overall order accuracy and customer satisfaction (Sharma, 2021).

Integration Capabilities: The ability to integrate with other business systems such as CRM, ERP, and payment gateways is crucial for an OMS, ensuring data consistency and operational efficiency (Chen & Popovich, 2018).

Scalability: A robust OMS can handle increasing order volumes as the business grows, ensuring that the system remains efficient and effective even during peak times (Davis, 2019).

Customer Communication: Effective order management systems keep customers informed through automated notifications and updates about their order status, enhancing the customer experience (Sharma, 2021).

2.2.4.2 Types of Order Management Systems

- **Standalone OMS:** These systems are dedicated solely to order management and are not integrated with other business systems. They are suitable for small businesses with straightforward order processing needs.
- **Integrated OMS:** These systems are part of a broader enterprise resource planning (ERP) system or customer relationship management (CRM) system. Integrated OMS solutions provide a unified platform for managing various business processes, ensuring better data flow and consistency.
- **Cloud-Based OMS:** These systems are hosted on the cloud, providing scalability, accessibility, and lower upfront costs. Cloud-based OMS solutions are ideal for businesses looking for flexibility and ease of access from multiple locations.
- **On-Premises OMS:** These systems are hosted on the company's own servers, providing greater control and security. On-premises OMS solutions are suitable for businesses with specific regulatory requirements or those that require significant customization.
- **Mobile OMS:** These systems are optimized for use on mobile devices, allowing businesses to manage orders on the go. Mobile OMS solutions are particularly useful for businesses

with a mobile workforce or those that require real-time order management from various locations.

2.3. Analysis of Existing similar systems

The analysis of existing literature provides insights into the development, implementation, and impact of online food ordering systems.

2.3.1. Similar Systems and Their Gaps

Numerous online food ordering systems have been developed and implemented worldwide, each with varying degrees of success and functionality. Notable examples include:

2.3.1.1. Uber Eats: A widely used platform that connects users with local restaurants, allowing for easy ordering and delivery. However, it often faces issues related to high service fees and delivery delays. Studies have shown that while Uber Eats provides convenience, it can also lead to customer dissatisfaction due to inconsistencies in service quality (Gonzalez & Smith, 2020).

2.3.1.2 Grubhub: Known for its extensive restaurant network and user-friendly interface. Despite its popularity, users sometimes report issues with order accuracy and customer service response times. Research indicates that these issues can affect customer loyalty and satisfaction (Patel & Jones, 2019). These systems, while popular, have gaps that can impact user satisfaction, such as high service fees, order inaccuracies, and limited payment options. The proposed online food ordering system for Maale Restaurant aims to address these gaps by focusing on affordability, order accuracy, and secure, diverse payment methods.

2.3.2. Innovations Brought by the New System

The new online food ordering system for Maale Restaurant will introduce several innovations to enhance user experience and operational efficiency:

2.3.2.1. Customizable User Interface

A user-friendly and intuitive interface that can be personalized according to user preferences, ensuring ease of navigation and a seamless ordering experience. This customization will cater to diverse user needs and improve overall satisfaction.

2.3.2.2. Integrated Payment Solutions

Secure and efficient payment gateways, including mobile money options, to cater to the diverse payment preferences of customers in Juba. This integration will ensure smooth and reliable transactions, reducing the risk of payment-related issues.

2.3.2.3. Advanced Order Management

A robust order management system that minimizes errors and optimizes workflow, enabling staff to handle orders more efficiently and accurately. This system will improve operational efficiency and reduce the likelihood of order mishandling.

2.3.2.4. Optimized Database Design

A well-structured database design to support efficient data storage, retrieval, and management. This design will ensure data integrity and optimize system performance, providing a seamless user experience.

These innovations aim to enhance the overall dining experience at Maale Restaurant, improve operational efficiency, and position the restaurant as a competitive player in the food and beverage industry in Juba, South Sudan

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

Systems analysis is a process of collecting factual data, understanding the processes involved, identifying problems, and recommending feasible suggestions for improving system functioning. It involves a thorough examination of the existing system to understand its workings, uncover inefficiencies, and pinpoint areas that require enhancement.

Based on the user requirements and the detailed analysis of the existing system, a new system must be designed. This phase of system designing is critical as it lays the foundation for the development of the new system. The logical system design, arrived at as a result of systems analysis, is then converted into a physical system design. This chapter covers the analysis and design processes undertaken to develop the online food ordering system for Maale Restaurant.

3.2 Analysis of the Current System

3.2.1 Problems of the Current System

The current system at Maale Restaurant involves several manual processes, which contribute to various operational inefficiencies and customer dissatisfaction. These processes include:

Order Placement: Customers place orders in person at the counter or over the phone. This method limits customer convenience, particularly in a modern, fast-paced environment where customers increasingly prefer online or mobile ordering options. It can also lead to congestion at the counter, especially during peak hours, further exacerbating wait times and customer frustration.

Order Recording: Staff manually record orders on paper or basic order pads. Manual recording is prone to human error, such as incorrect item entry or missed modifications, leading to order inaccuracies. This frustrates customers and results in wasted resources when incorrect orders must be remade.

Order Communication: Orders are verbally communicated or manually transferred to the kitchen staff. Verbal communication increases the risk of miscommunication between the front-of-house

staff and kitchen staff. This can lead to incorrect or incomplete orders being prepared, further slowing down service and impacting customer satisfaction.

Payment Processing: Payments are accepted in cash or through a traditional Point of Sale (POS) system. The reliance on cash and traditional POS systems limits the payment options available to customers. In an era where digital and contactless payments are becoming the norm, this limitation can inconvenience customers and slow down the checkout process, especially when change must be provided or card readers malfunction.

Order Fulfillment: Orders are prepared and served to customers or packed for takeaway. The manual nature of order fulfillment can lead to delays, especially if the kitchen is overwhelmed with orders or if there is a breakdown in communication between the staff. This can result in long wait times for customers and decreased overall efficiency in the restaurant's operations.

3.2.1.1. Impact of These Problems

Long Wait Times:

During peak hours, the manual process of taking and recording orders leads to significant delays.

Customers experience longer wait times, leading to frustration and dissatisfaction.

Order Errors:

Miscommunication between staff and kitchen or mistakes in manually recording orders result in incorrect orders. This affects the accuracy of order fulfillment, leading to customer complaints and potential loss of business.

Operational Inefficiency:

The manual system is labor-intensive and prone to human error. Staff spend excessive time on repetitive tasks. This reduces overall operational efficiency and increases the workload on staff, affecting productivity.

Customer Dissatisfaction:

The combination of long wait times, order inaccuracies, and inefficient service negatively impacts customer satisfaction. Unsatisfied customers are less likely to return, and negative word-of-mouth can harm the restaurant's reputation.

Lack of Real-Time Order Tracking:

Customers have no way to track the status of their orders in real-time. This lack of transparency leads to uncertainty and dissatisfaction among customers.

Limited Payment Options:

The current system primarily supports cash payments, with limited integration of modern digital payment methods. This restricts customer convenience and excludes those who prefer or rely on digital payment options.

Inefficient Data Management:

Manually recorded data on paper is difficult to manage, analyze, and retrieve. This hampers the ability to gain insights into sales trends, customer preferences, and operational performance.

This analysis highlights the key problems facing Maale Restaurant's current system, emphasizing the need for a more efficient and modernized approach to order management, payment processing, and customer service to enhance overall performance and customer satisfaction..

3.3 Analysis of the New System

3.3.1 Introduction

The new system is designed to address the issues identified in the current system by leveraging technology to streamline operations and enhance customer satisfaction. The following sections detail the requirements, functional diagrams, and methodologies used in developing the new system.

3.3.2 System Requirements

Functional Requirements	**Non-Functional Requirements**
User Registration and Login: The system must allow users to create accounts and log in securely.	Performance: The system must handle multiple simultaneous users without significant delays.
Menu Browsing: Users should be able to browse the restaurant's menu, including categories, descriptions, and prices.	Security: User data, especially payment information, must be protected using industry-standard security practices.
Order Placement: Users must be able to select items, customize their orders, and place them through the system**.**	Usability: The system should be easy to use and navigate, even for non-tech-savvy users.

Payment Processing: The system should support payment methods cash on delivery**.**	Scalability: The system should be able to accommodate future growth, including more users and additional features.
Order Tracking: Users should be able to track the status of their orders in real-time.	Reliability: The system must be reliable and available with minimal downtime.
Order Management: The system must provide tools for staff to manage orders, including preparation, packaging, and delivery.	

3.3.3 Functional Diagram

Functional diagrams illustrate the relationships between the main components of the system and their interactions. They are critical for understanding how data flows through the system and identifying the necessary processes to be implemented. The following sections detail the functional diagrams used in the design of the online food ordering system for Maale Restaurant.

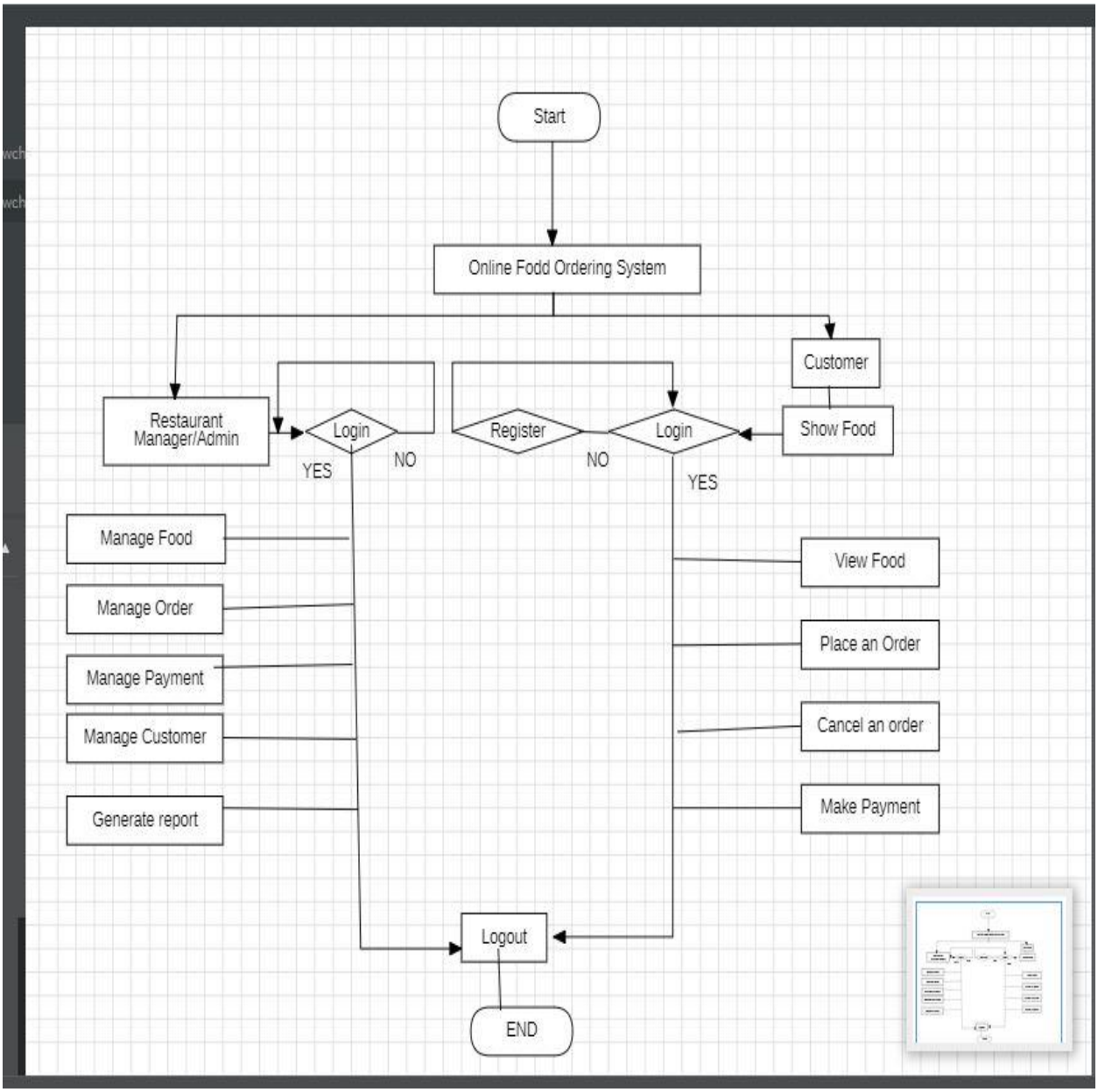


Figure 1. Functional Diagram

Functional diagrams illustrate the relationships between the main components of the system and their interactions.

3.3.4. Methodological Approach

3.3.4.1 Data Collection Techniques

As a student conducting this project within a short timeframe, I utilized streamlined data collection methods to gather necessary information efficiently. This included:

Observations: I made direct observations of the restaurant's operations and customer interactions to supplement survey and interview data and gain a deeper understanding of the current situation.

Documentation: I reviewed existing records, including order logs, customer feedback forms, and financial reports to gather quantitative data on the restaurant's operations.

3.3.4.2 Software Development Methodology

I chose the Agile model for this project due to its iterative approach, which allows for continuous feedback and improvements. The Agile model is particularly suitable for projects with tight deadlines and limited resources, as it emphasizes flexibility, collaboration, and customer involvement. The Agile process involved the following steps:

- **Planning:** Defining the project scope, objectives, and timeline.
- **Requirement Analysis:** Gathering and documenting user requirements.
- **Design:** Creating the system architecture and design specifications.
- **Development:** Writing and testing the code in small, manageable increments.
- **Testing:** Conducting usability testing and gathering user feedback.
- **Deployment:** Implementing the system and providing training to users.
- **Review:** Collecting feedback and making necessary adjustments.

3.3.4.3 System Design Methodology

I utilized the Structured System Analysis and Design Methodology (SSADM) for the system design of the online food ordering system for Maale Restaurant. SSADM is a widely recognized and systematic approach to analyzing and designing information systems. It ensures that all aspects of the system are thoroughly analyzed, documented, and designed before development begins, thereby reducing risks and increasing the likelihood of successful implementation (Connolly & Begg, 2015).

Definition of SSADM

Structured System Analysis and Design Methodology (SSADM) is a comprehensive, data-driven methodology used for the analysis and design of information systems. It was developed by the UK government in the early 1980s and has since become a standard approach in system development projects. SSADM is characterized by its rigor and attention to detail, making it particularly suitable for large and complex systems (Connolly & Begg, 2015). The methodology involves a series of stages and activities that guide the process of analyzing the current system, identifying requirements, and designing a new system. SSADM focuses heavily on data flow, data structure, and process modeling, ensuring that the system's functional and non-functional requirements are thoroughly addressed (Connolly & Begg, 2015)

3.3.4.3.1. Approaches within SSADM

SSADM can be approached using three different perspectives, each offering a unique focus on the system's design and implementation:

1. **Logical Data Modeling:**

   This approach involves creating a detailed model of the data required by the system. It focuses on defining the data entities, their attributes, and the relationships between them. Logical data modeling ensures that the data structure is well-organized and supports the system's functional requirements (Connolly & Begg, 2015).

2. **Data Flow Modeling:**

   Data flow modeling involves mapping out the flow of data within the system. This approach uses Data Flow Diagrams (DFDs) to represent how data moves between processes, data stores, and external entities. It helps in identifying the processes that need to be developed and how they interact with each other (Connolly & Begg, 2015).

3. **Entity-Behavior Modeling:**

   Entity-behavior modeling focuses on the dynamic aspects of the system. It models the behavior of data entities over time, particularly how they change state in response to different events. This approach is crucial for understanding the system's dynamic processes and ensuring that all possible scenarios are accounted for (Connolly & Begg, 2015).

3.3.4.3.2. Components of SSADM

SSADM consists of three main components, each of which is divided into specific stages and sub-divisions. These components guide the analysis and design process and ensure that all aspects of the system are thoroughly considered:

Feasibility Study:

Definition: The feasibility study is the first component of SSADM, aimed at determining whether the proposed system is viable and worth pursuing. It assesses the technical, economic, and operational feasibility of the system (Connolly & Begg, 2015).

Sub-Divisions:

- **Technical Feasibility:** Evaluates whether the technology needed to implement the system is available and suitable (Connolly & Begg, 2015).
- **Economic Feasibility:** Assesses the cost-effectiveness of the system, comparing the projected benefits with the estimated costs (Connolly & Begg, 2015).
- **Operational Feasibility:** Examines whether the system will function effectively within the existing operational environment (Connolly & Begg, 2015).

Requirements Analysis:

Definition: Requirements analysis is the component where the detailed requirements of the system are gathered and documented. It involves understanding the needs of the users and the system's functionality (Connolly & Begg, 2015).

Sub-Divisions:

- **Fact Finding:** Gathering detailed information about the current system, user requirements, and operational constraints (Connolly & Begg, 2015).
- **System Modeling:** Developing models such as Data Flow Diagrams (DFDs) and Entity-Relationship Diagrams (ERDs) to represent the system's requirements (Connolly & Begg, 2015).
- **Specification:** Documenting the system's functional and non-functional requirements in a structured format (Connolly & Begg, 2015).

System Design:

Definition: The system design component is where the logical models developed during the requirements analysis are translated into physical designs that can be implemented (Connolly & Begg, 2015).

Sub-Divisions:

- **Logical Design:** Creating a detailed design of the system's architecture, focusing on data structures, database design, and user interfaces (Connolly & Begg, 2015).
- **Physical Design:** Translating the logical design into a physical implementation, including hardware specifications, network design, and software architecture (Connolly & Begg, 2015).
- **System Integration:** Planning and designing how different components of the system will be integrated and how data will be shared between them (Connolly & Begg, 2015).

3.3.4.3.3. Tools Used in SSADM

In the context of Maale Restaurant's online food ordering system, the following SSADM tools were utilized:

Data Flow Diagrams (DFDs):

DFDs were used to map out the flow of data within the system, providing a clear representation of how data moves between different processes and entities (Connolly & Begg, 2015).

Entity-Relationship Diagrams (ERDs):

ERDs were employed to model the data structure, defining the entities involved in the system, their attributes, and the relationships between them (Connolly & Begg, 2015).

Data Dictionary:

The data dictionary was created to document all the attributes found in every table within the database, ensuring consistency and clarity in data definitions (Connolly & Begg, 2015).

By following the SSADM approach, the system design process for Maale Restaurant's online food ordering system was conducted in a structured, methodical manner, ensuring that all critical aspects were thoroughly analyzed and that the resulting system design met the project's objectives.

.3.3.4.3.1 Context Diagram (Level 0 DFD)

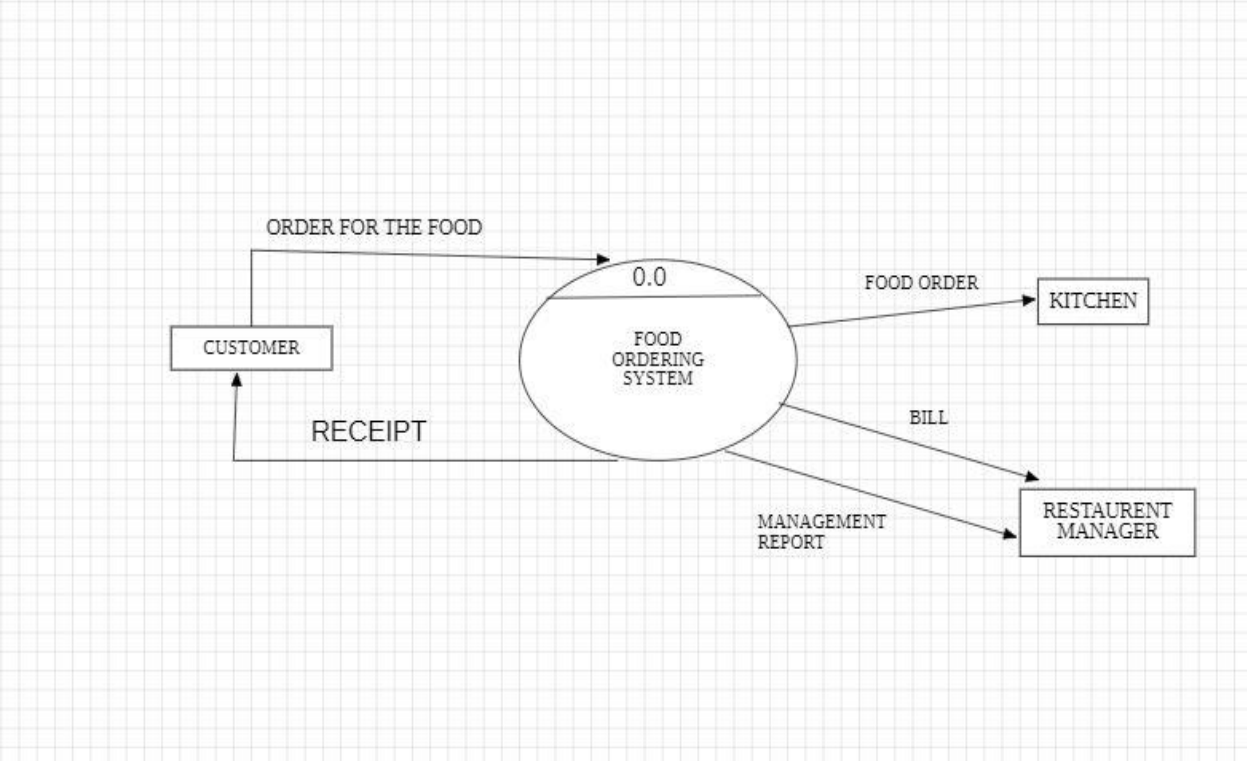


Figure 2. Context Diagram (Level 0 DFD)

This diagram shows how the system interacts with the external entities (Customer, Restaurant Staff, and Payment Gateway) and the data flows between them.

3.3.4.3.2. Level 1 DFDl

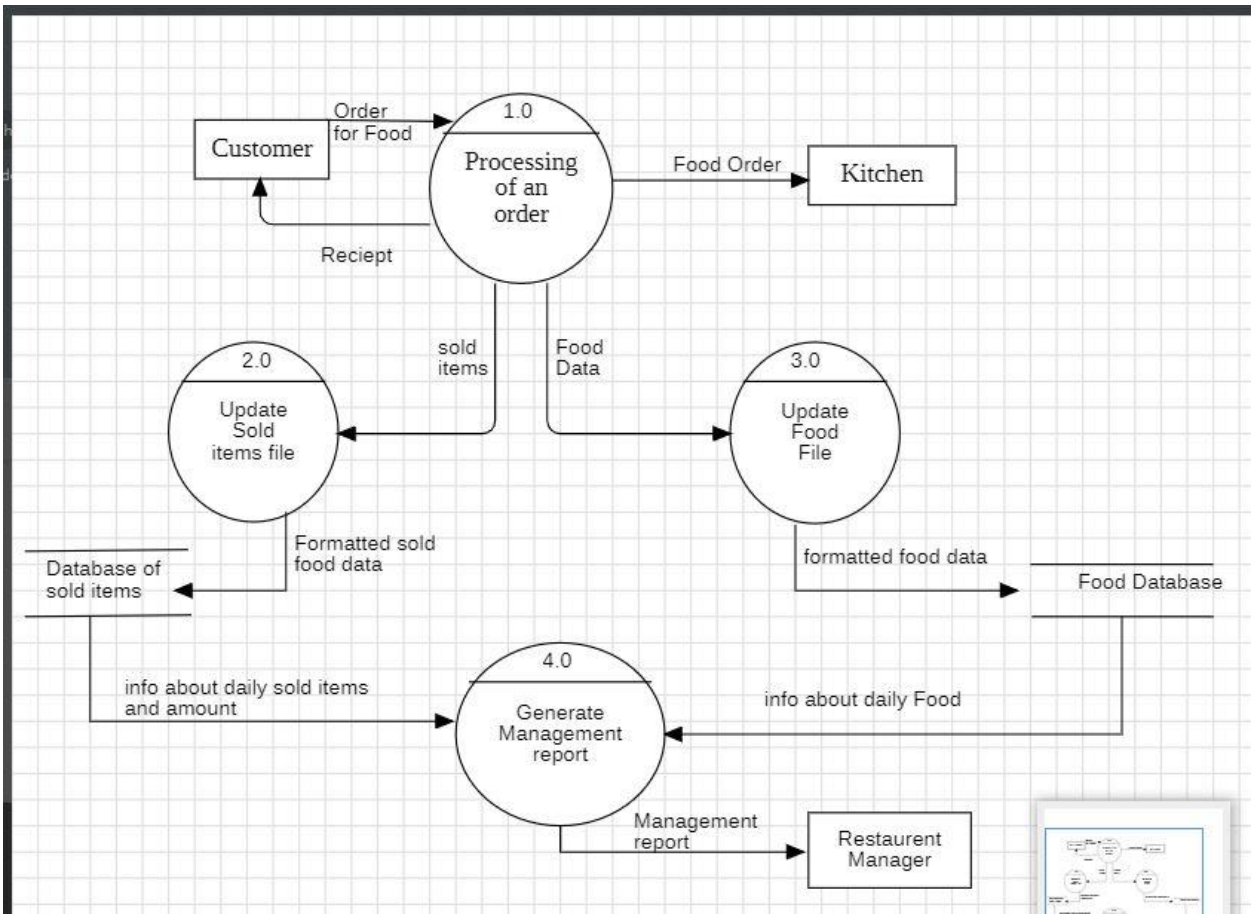


Figure 3. Level 1 DFDl

A Level 1 Data Flow Diagram (DFD) provides a more detailed breakdown of the internal processes of the system, expanding on the context diagram by showing major sub-processes and how data flows between them.

3.3.4.3.3. Entity Relationship Diagram(ERD)

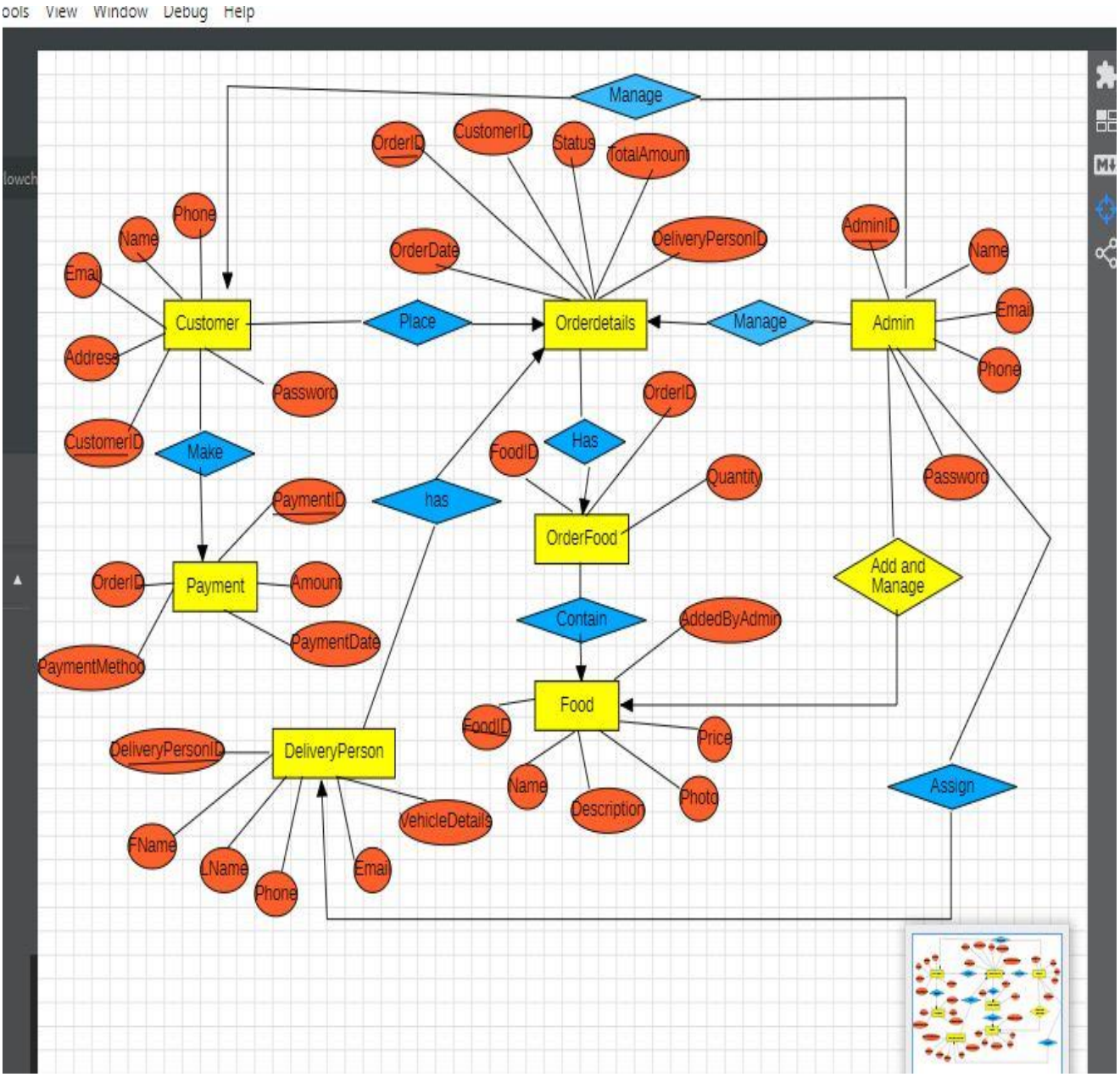


Figure 4. Entity Relationship Diagram(ERD)

An Entity Relationship Diagram (ERD) represents the data structure and relationships between entities in the Online Food Ordering System. The ERD shows how data is stored in a relational database by identifying key entities, their attributes, and the relationships between them.

3.3.4.3.4. Data dictionary

1. Admin

Table 1. Admin

Attribute Name	Data Type	Constraints	Description
AdminID	INT	Primary Key, Auto Increment	Unique identifier for each admin
Name	VARCHAR(20)	NOT NULL	Name of the admin
Email	VARCHAR(20)	NOT NULL, UNIQUE	Admin's email address
Phone	VARCHAR(15)	NOT NULL	Admin's phone number
Password	VARCHAR(20)	NOT NULL	Admin's password

Table 2. Customer

Attribute Name	Data Type	Constraints	Description
CustomerID	INT	Primary Key, Auto Increment	Unique identifier for each customer
Name	VARCHAR(25)	NOT NULL	Name of the customer
Email	VARCHAR(30)	NOT NULL, UNIQUE	Customer's email address
Phone	VARCHAR(15)	NOT NULL	Customer's phone number

Address	VARCHAR(25)	NOT NULL	Customer's delivery address
Password	VARCHAR(20)	NOT NULL	Customer's password

Table 3. Food.

Attribute Name	**Data Type**	**Constraints**	**Description**
FoodID	INT	Primary Key, Auto Increment	Unique identifier for each food item
Name	VARCHAR(100)	NOT NULL	Name of the food item
Description	TEXT	NOT NULL	Description of the food item
Price	DECIMAL(10, 2)	NOT NULL	Price of the food item
AddedByAdminID	INT	Foreign Key (AdminID)	Admin who added the food item

Table 4. Orderdetails.

Attribute Name	**Data Type**	**Constraints**	**Description**
OrderID	INT	Primary Key, Auto Increment	Unique identifier for each order
CustomerID	INT	Foreign Key (CustomerID)	Customer who placed the order
DeliveryPersonID	INT	Foreign Key (DeliveryPersonID)	Delivery person assigned to the order

OrderDate	DATETIME	NOT NULL	Date and time when the order was placed
Status	VARCHAR(50)	NOT NULL	Current status of the order (e.g., Pending, Delivered)
TotalAmount	DECIMAL(10, 2)	NOT NULL	Total amount for the order

Table 5. Payment

Attribute Name	**Data Type**	**Constraints**	**Description**
PaymentID	INT	Primary Key, Auto Increment	Unique identifier for each payment
OrderID	INT	Foreign Key (OrderID)	Order associated with the payment
PaymentDate	DATETIME	NOT NULL	Date and time when the payment was made
Amount	DECIMAL(10, 2)	NOT NULL	Amount paid
PaymentMethod	VARCHAR(50)	NOT NULL	Method used for payment (e.g., Credit Card, Cash)

Table 6. orderfood

Attribute Name	**Data Type**	**Constraints**	**Description**

OrderID	INT	Primary Key, Foreign Key (OrderID)	Order associated with the food item
FoodID	INT	Primary Key, Foreign Key (FoodID)	Food item in the order
Quantity	INT	NOT NULL	Quantity of the food item ordered

Table 7. DeliveryPerson

Attribute Name	**Data Type**	**Constraints**	**Description**
DelID	INT	Primary Key, Auto Increment	Unique identifier for each delivery person
Name	VARCHAR(25)	NOT NULL	Name of the delivery person
Phone	VARCHAR(15)	NOT NULL	Delivery person's phone number
Email	VARCHAR(25)	NOT NULL, UNIQUE	Delivery person's email address
VehicleDetails	VARCHAR(25)	NULL	Details about the vehicle (e.g., license plate, model)

CHAPTER FOUR: SYSTEM IMPLEMENTATION

4.1. Implementation and Coding

4.1.1. Introduction

The implementation phase is a critical stage in the system development lifecycle, where the theoretical designs and plans are transformed into functional software. This phase involves translating the system design into actual code that a computer can execute. Implementation aims to develop a system that meets the specified requirements and operates efficiently in the intended environment.

In this chapter, we will discuss the steps involved in implementing the online food ordering system for Maale Restaurant. The process includes setting up the development environment, writing the code for the various system components, and integrating these components to create a cohesive and functional system. We will also cover the testing procedures used to ensure that the system operates correctly and meets all the defined requirements.

4.1.2. Description of Implementation tools and technology

The successful implementation of the online food ordering system for Maale Restaurant requires the use of various tools and technologies. These tools and technologies are selected based on their compatibility, scalability, and ability to meet the system's functional requirements. The key implementation tools and technologies used in this project include HTML, CSS, JavaScript, PHP, and MySQL.

4.1.2.1. HTML (HyperText Markup Language): HTML is the standard markup language used to create the structure of web pages. It is the backbone of web development, providing the basic framework upon which all other elements are built. In this project, HTML is used to define the structure of the web pages, including the layout of the order forms, menus, and user interface

elements. Each page of the system, from the customer ordering interface to the admin dashboard, is built using HTML.

4.1.2.2. CSS (Cascading Style Sheets): CSS is used to control the appearance and layout of HTML elements on the web pages. It allows for the separation of content from design, enabling developers to apply consistent styles across multiple pages and devices. In this project, CSS is employed to create an attractive and user-friendly interface for both customers and administrators. It is used to style the buttons, forms, navigation bars, and other visual elements, ensuring that the system is aesthetically pleasing and responsive to different screen sizes.

4.1.2.3. JavaScript: JavaScript is a versatile scripting language used to enhance the interactivity and functionality of web pages. It is particularly useful for implementing client-side logic, such as form validation, dynamic content updates, and user interactions. In the online food ordering system, JavaScript is utilized to perform real-time validation of user input, handle events like button clicks, and manage the dynamic updating of the order summary. This ensures a smooth and interactive user experience.

4.1.2.4. PHP (Hypertext Preprocessor): PHP is a server-side scripting language designed for web development. It is used to handle the backend logic of the system, including processing user requests, interacting with the database, and generating dynamic content. In this project, PHP is the primary language used to implement the core functionalities of the system, such as user authentication, order processing, and payment management. PHP scripts are responsible for connecting the front-end interfaces with the MySQL database, retrieving and storing data as needed.

4.1.2.5. MySQL: MySQL is a relational database management system (RDBMS) used to store and manage the data for the online food ordering system. It is chosen for its reliability, ease of use, and compatibility with PHP. The system's database is designed using MySQL to handle all the data related to customers, orders, food items, and payments. The database is structured to ensure

efficient data retrieval, storage, and manipulation, supporting the system's overall performance and scalability.

These tools and technologies work together to create a robust and efficient online food ordering system. HTML, CSS, and JavaScript provide a responsive and interactive front-end experience, while PHP and MySQL manage the backend processes and data management. The combination of these technologies ensures that the system is not only functional but also user-friendly and capable of handling the restaurant's operational needs.

4.1.3. Screen shorts and source codes

1. Database tables.

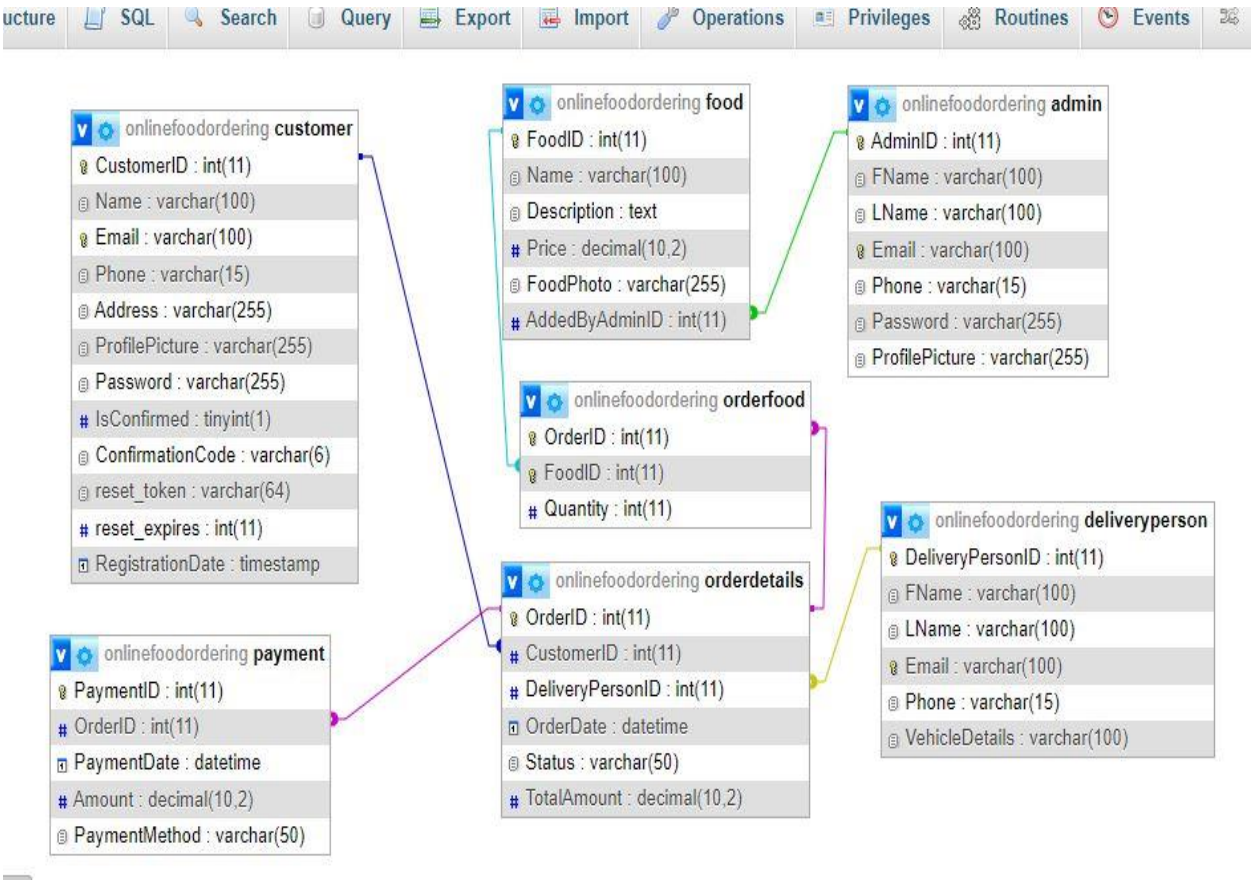


Figure 5. Database tables.

This are all database tables for Online Food Ordering System.

4.1.3.1 Customer Interface.

1. Customer Menu Side.

This is where customer order his/her food.

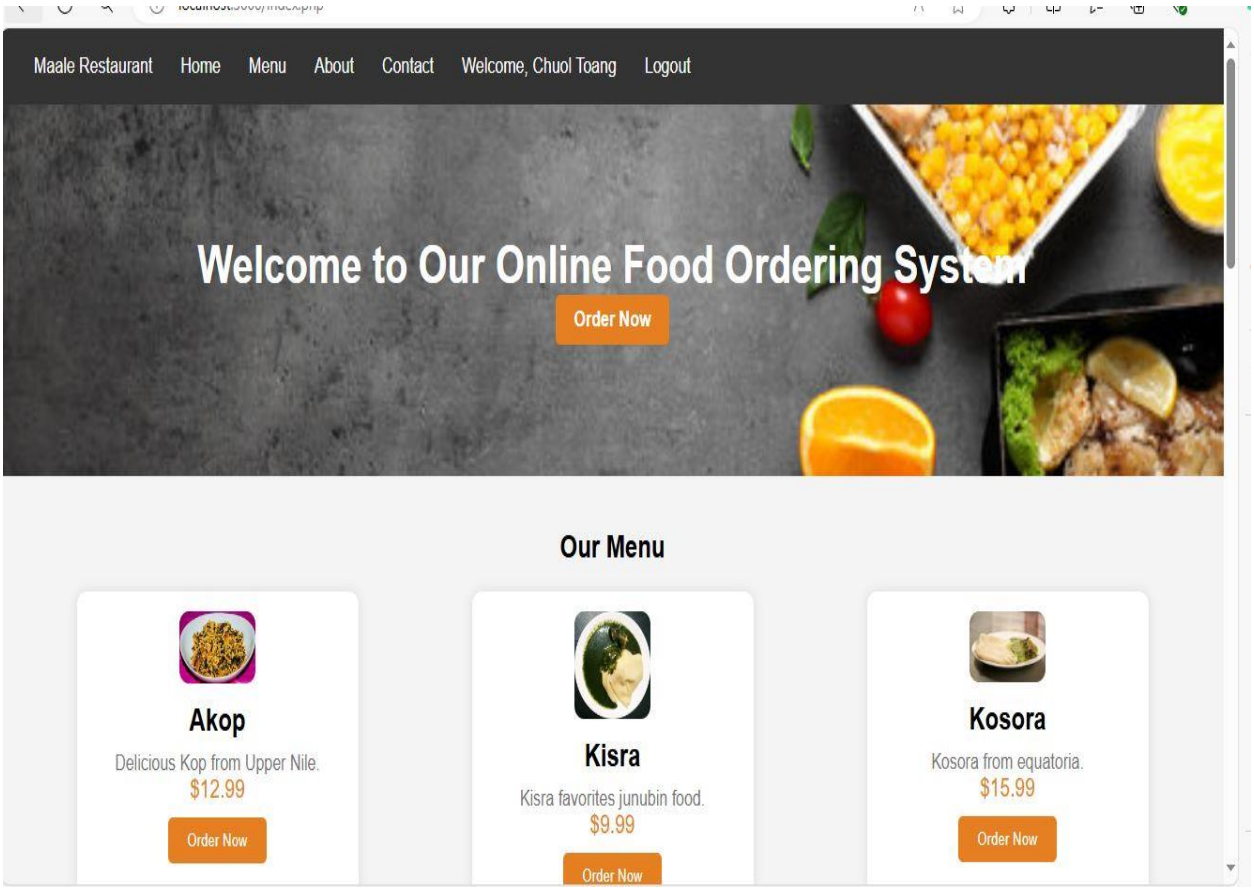


Figure 6. Customer Menu Side.

The Customer Menu Side represents the interface through which customers interact with the restaurant's online ordering system. It includes various features and options that allow customers to browse the menu, select items, and manage their orders.

2 Custpmer Registration.

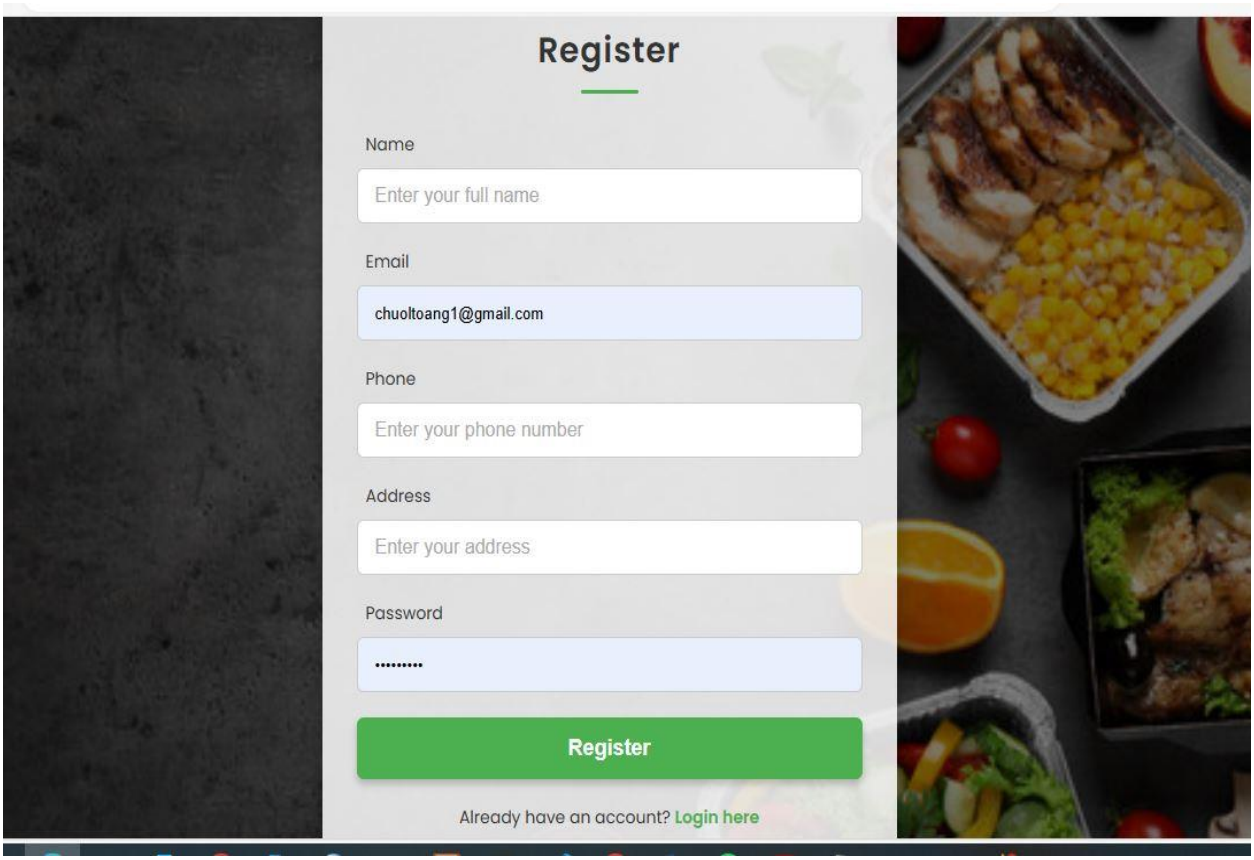


Figure 7. Custpmer Registration.

This is where new customer register to access and buy Foods.

3 Customer Login.

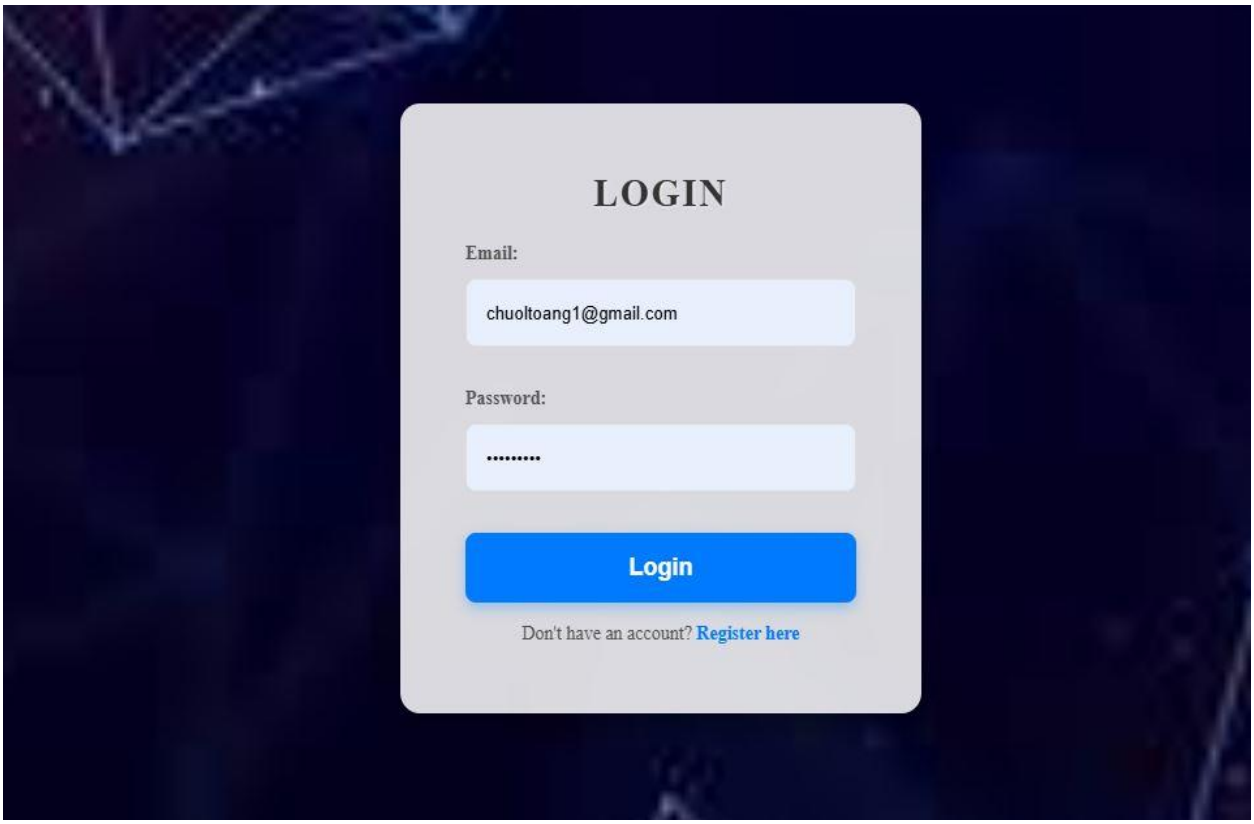


Figure 8. Customer Login.

This is where customer login when they want to buy food.

Customer order summary.

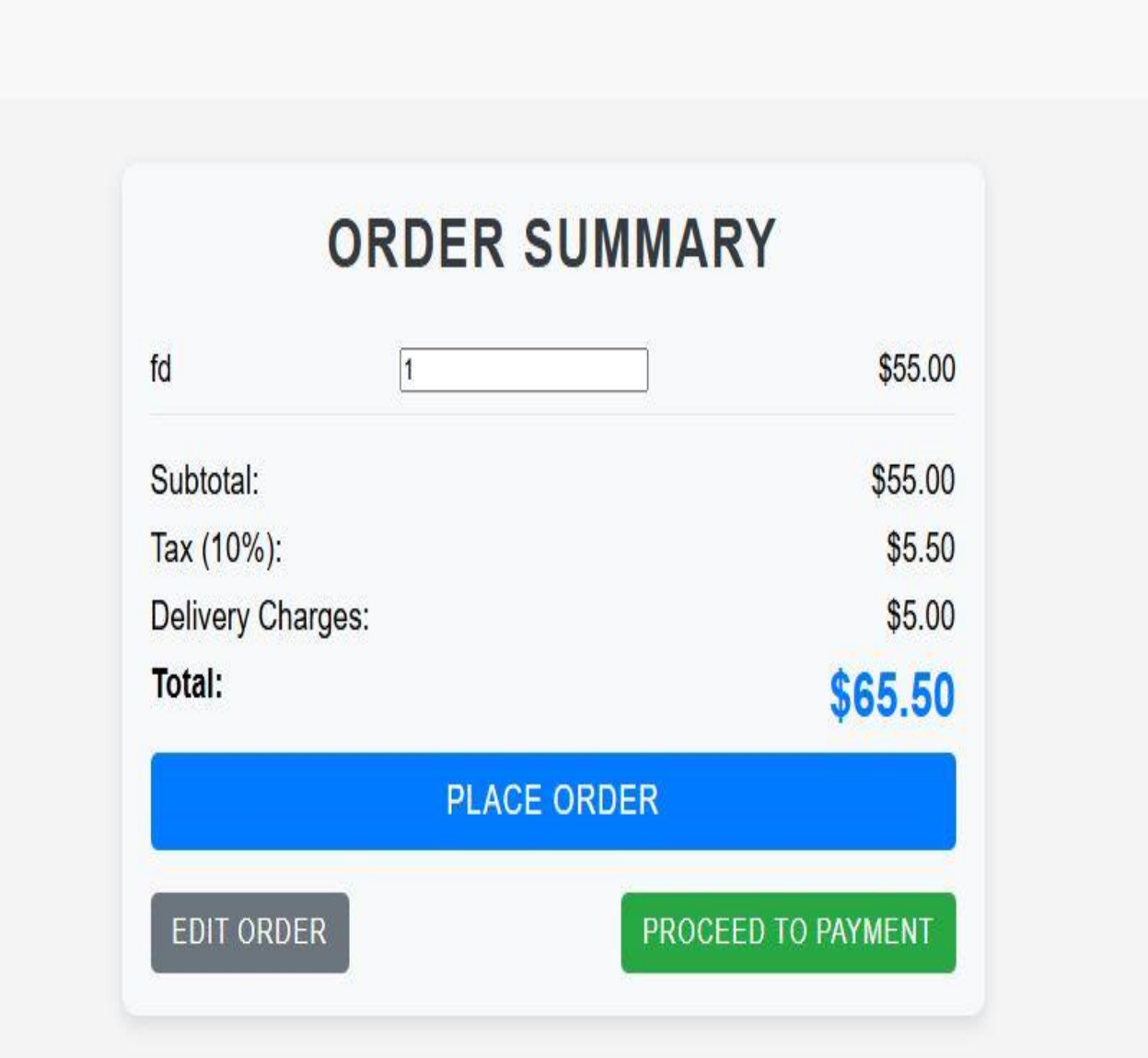


Figure 9. Customer order summary.

The Customer Order Summary is the final step before order confirmation, providing customers with an overview of their selected items and total cost. It includes:

Customer Payment

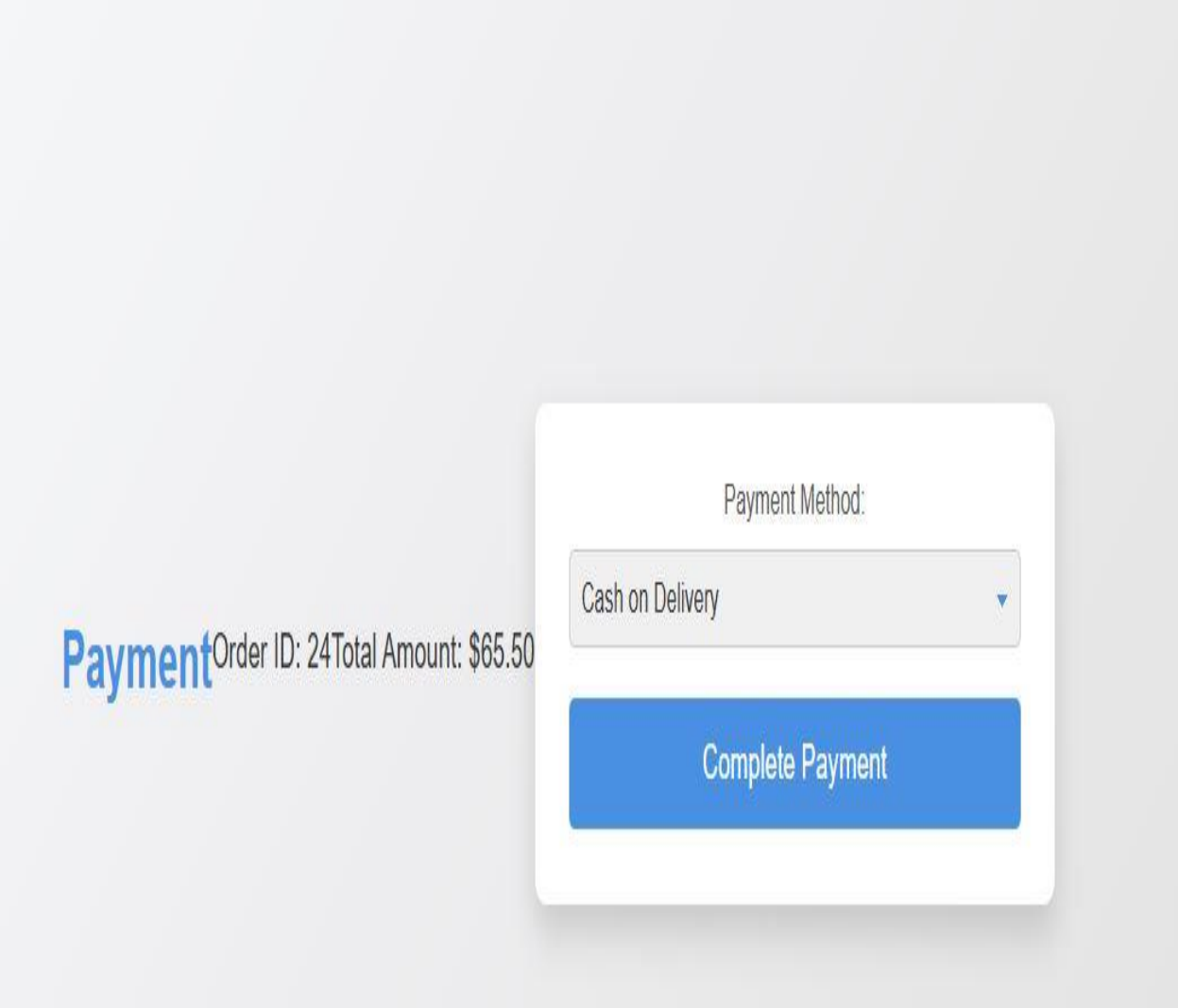


Figure 10. Customer Payment

That’s where customer complete their payment to confirm their order.

4.1.3.2 Admin Interface.

Admin Home.

Figure 11. Admin Home.

This figure illustrates the Admin Home dashboard, featuring key management sections including: Manage Food, for overseeing menu items; Manage Customer, for handling customer information; Manage Orders, for tracking and processing orders; Manage Payment, for processing transactions; and Report, for generating and reviewing reports.

Admin Login Interface

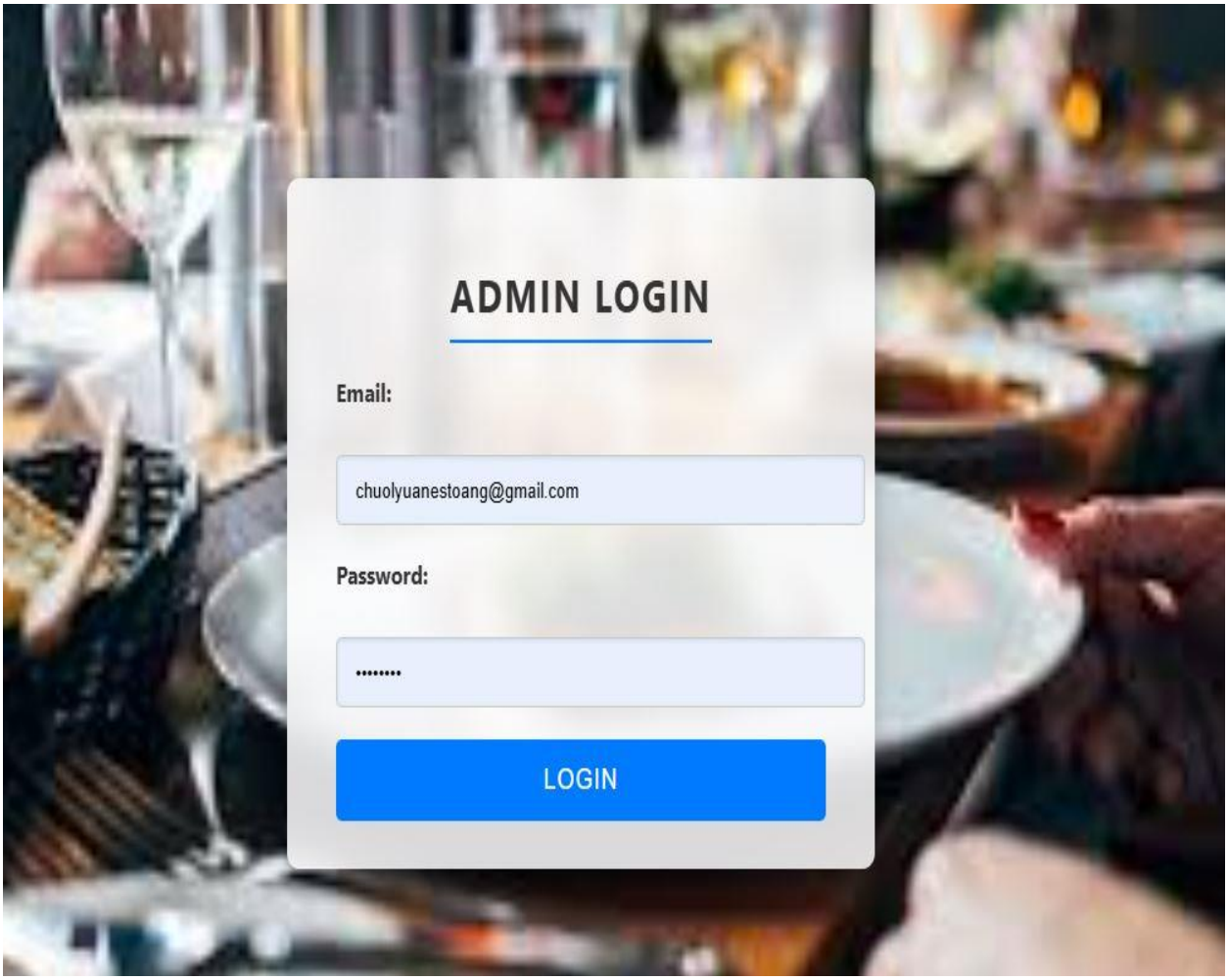


Figure 12. Admin Login Interface.

That's admin login page to access admin dashboard to run restaurant operation.

Manage Order.

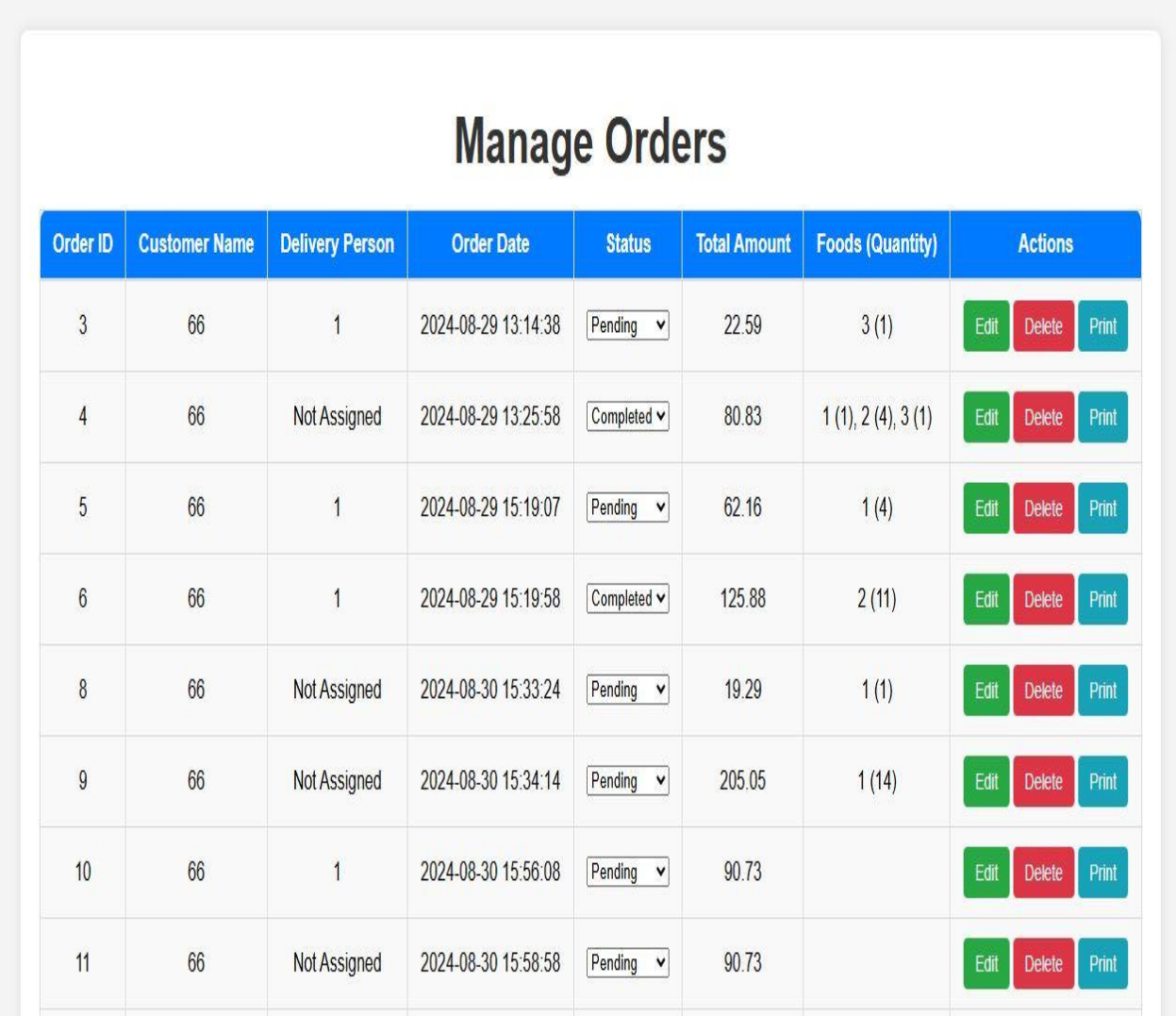


Figure 13. Admin Manage Order.

This figure showcases the Admin Manage Order interface. It displays a detailed view for overseeing and processing orders. The interface includes sections for viewing current and past orders, updating order statuses, managing order details, and addressing any issues or special requests.

Admin Manage customer.

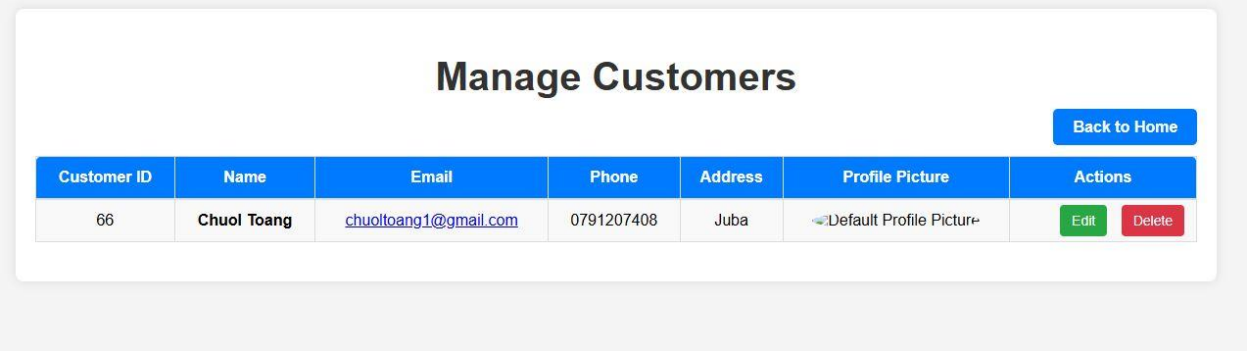


Figure 14. Admin Manage customer.

This figure presents the Admin Manage Customer interface. It provides a comprehensive view for handling customer information.

Admin Manage Payment.

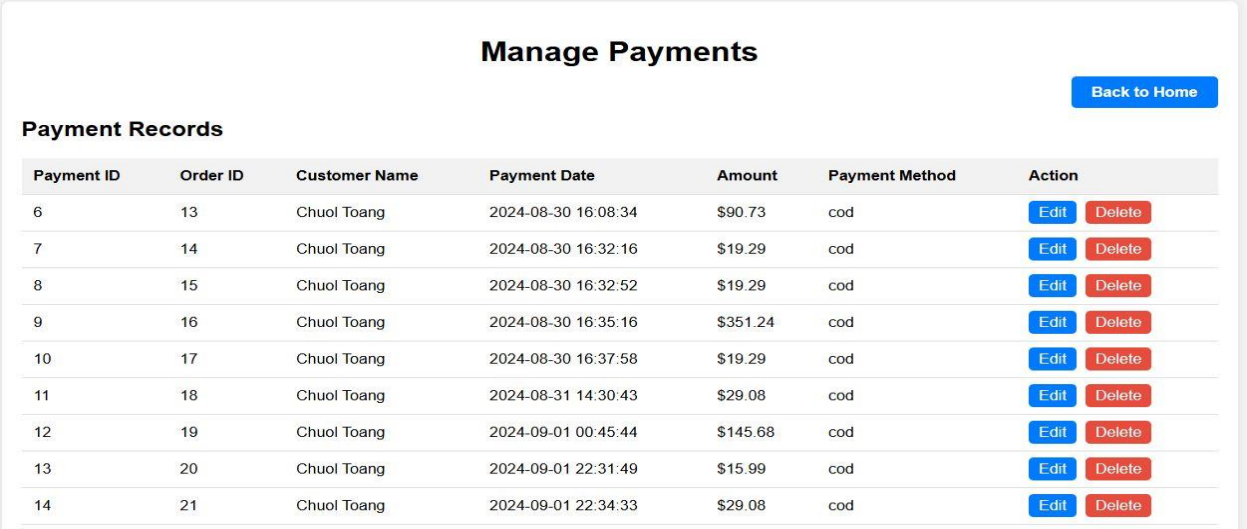


Figure 15. Admin Manage Payment

This figure displays the Admin Manage Payment interface. It highlights tools for overseeing and processing payments.

Admin Manage food current foods.

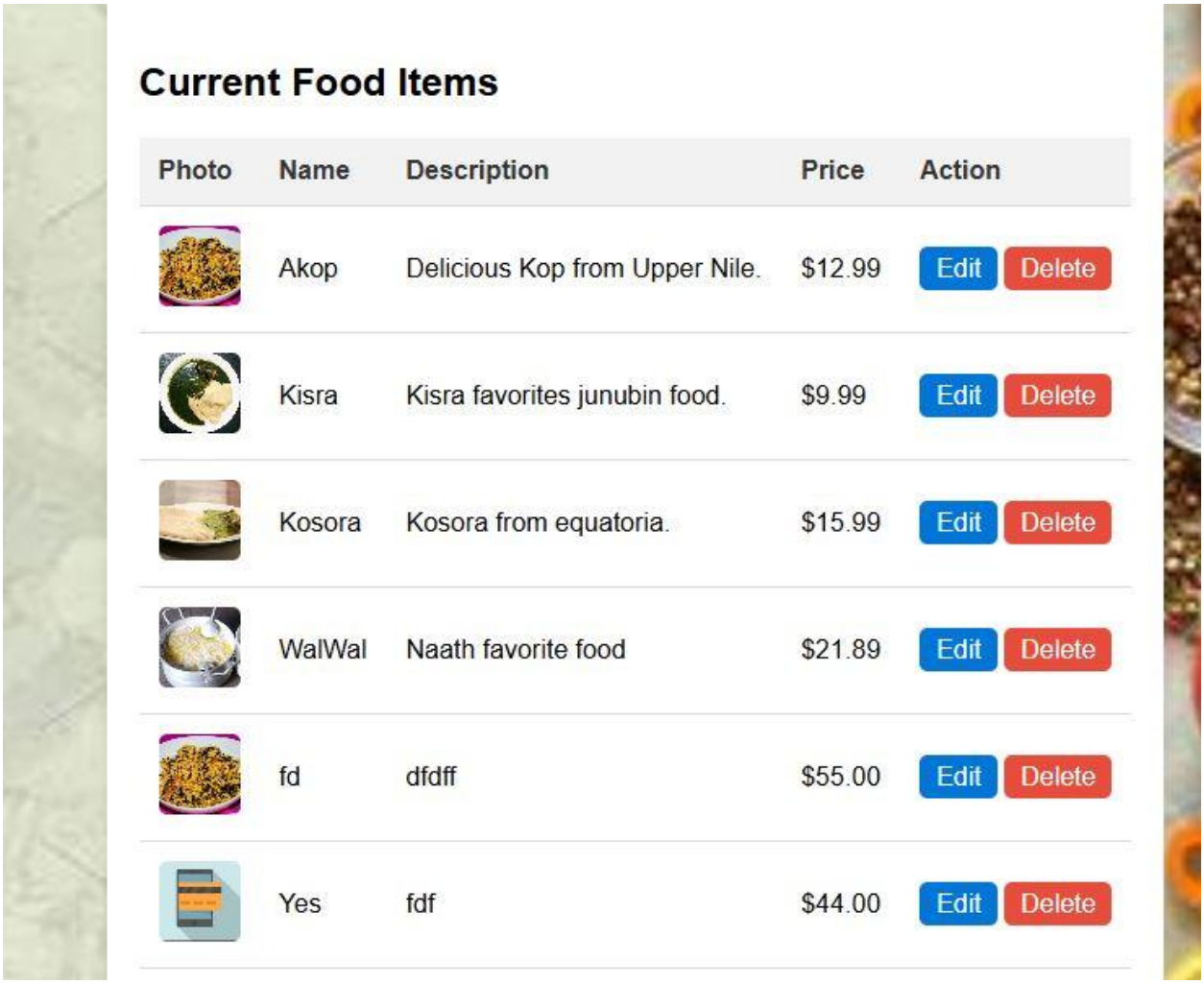


Figure 16. Admin Manage food current foods.

This figure illustrates the Admin Manage Food interface focusing on current food items. It provides a detailed view of the existing menu, including options for viewing, editing, and updating food items. The interface includes information on item names, descriptions, prices

Admin new food to be added.

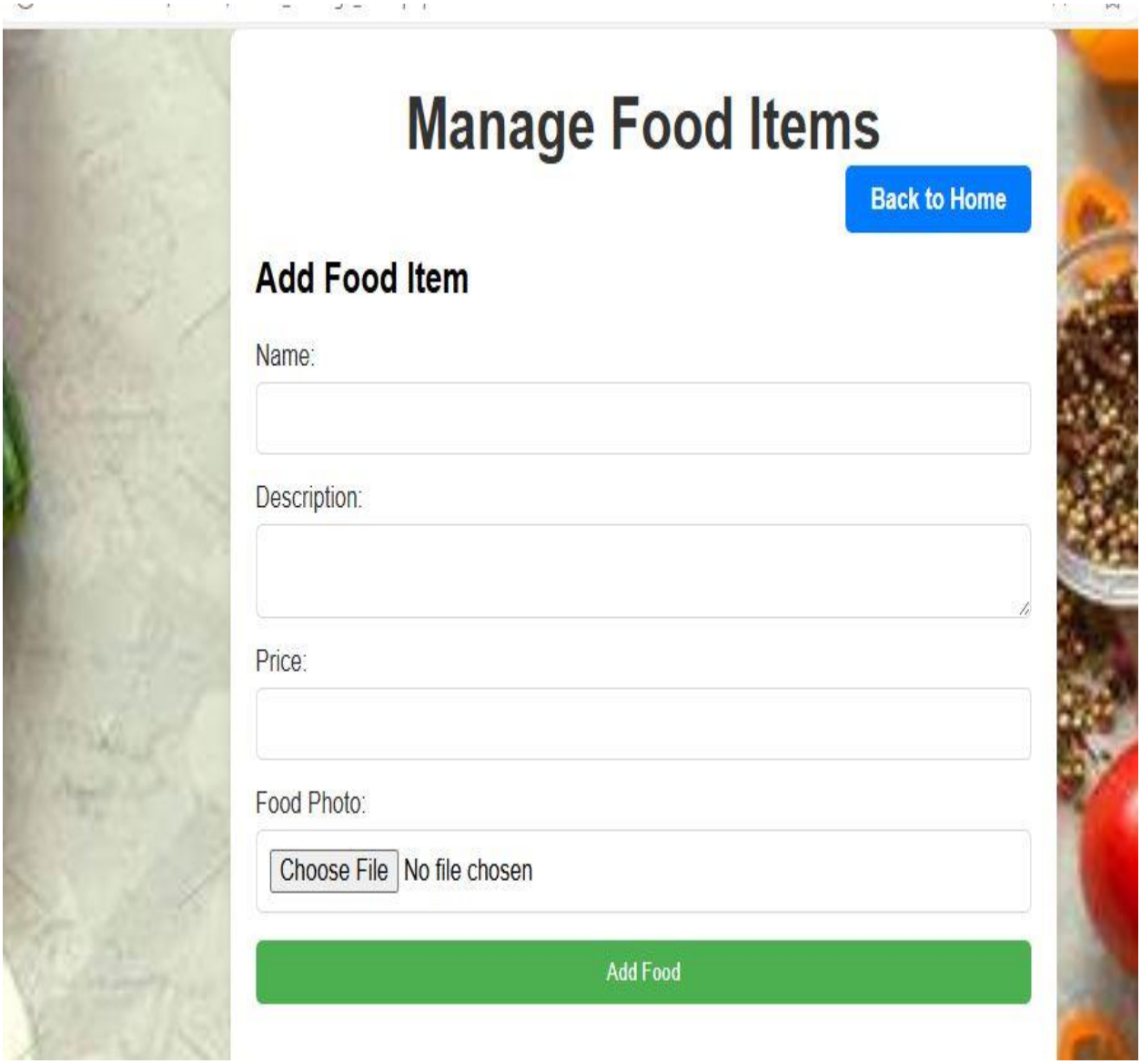


Figure 17. Admin new food to be added.

This figure depicts the interface for adding new food items within the Admin Manage Food section. It showcases the form or input fields required to introduce a new menu item, including options for entering item names, descriptions, prices,

4.2. Testing

Testing is a critical phase in the software development lifecycle, ensuring that the system functions as intended, meets the requirements, and is free of major defects. The primary objective of testing is to identify and fix any issues before the system is deployed for use by real users. In this chapter, the testing strategies, tools, and outcomes of the testing process for the online food ordering system at Maale Restaurant will be discussed.

4.2.1. Introduction

The testing phase involves systematically evaluating the online food ordering system to verify that it operates according to the specified requirements and delivers the expected results. This process includes several types of testing, such as unit testing, integration testing, system testing, and user acceptance testing. Each type of testing serves a specific purpose and helps to ensure that the system is reliable, secure, and user-friendly.

The goal of testing is not only to find and fix bugs but also to validate that the system meets the business needs and provides a seamless experience for both customers and administrators. During this phase, different testing methodologies and tools are employed to thoroughly assess the system's performance, functionality, and usability.

4.2.2. Unit testing outputs

Unit testing is the first level of testing where individual components or modules of the online food ordering system are tested in isolation. The purpose of unit testing is to ensure that each unit of the software performs as expected. Below are the key outputs and results from the unit testing phase:

Functionality Verification: Each function and method in the system was tested individually to ensure they operate as expected.

The functions responsible for handling user inputs, processing orders, calculating totals, and managing database interactions were rigorously tested.

Input Validation: The system's ability to correctly validate various user inputs was tested, including text fields, numeric inputs, email formats, and special characters.

Test cases included scenarios with valid, invalid, and edge-case inputs to ensure robust validation mechanisms.

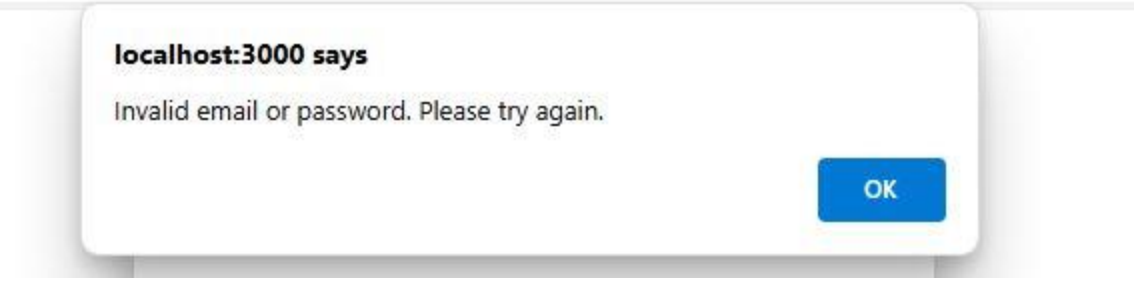


Figure 18. Testing

This figure show validation output when user enter wrong email or password.

Test Case Results: A series of test cases were executed for each module, with the results documented in a test report.

Most unit tests passed successfully, with any issues identified being documented and resolved before proceeding to the next testing phase.

4.2.3. Validation Testing outputs

Validation testing is performed to ensure that the online food ordering system meets the specified requirements and functions correctly in the intended environment. This phase focuses on validating the entire system against the end-user requirements and expectations. Below are the key outputs and results from the validation testing phase:

Requirements Fulfillment: The system was tested against the original requirements specification to ensure all features and functionalities were implemented as intended.

The outputs confirmed that the system meets the business requirements, including order processing, customer registration, payment processing, and report generation.

User Interface Validation: The user interface (UI) was tested to ensure it is user-friendly, intuitive, and visually appealing.

Validation tests included checking the layout, responsiveness, and accessibility of the system on different devices and browsers.

The outputs showed that the UI is consistent, with all elements properly aligned, and the system provides a smooth user experience.

Integration Testing Results: Validation testing included the integration of various system components, ensuring they work together seamlessly.

The outputs indicated that data flows smoothly between modules (e.g., from order processing to payment) and that no integration issues were present.

Error Handling and Notifications: The system's ability to handle errors and provide appropriate user notifications was validated.

The outputs showed that the system effectively manages exceptions and provides meaningful error messages, guiding users to take corrective actions.

Security Validation: Security features, such as user authentication, password encryption, and data protection, were tested to ensure compliance with security standards.

The outputs confirmed that the system securely handles user data and transactions, protecting against unauthorized access and data breaches.

Compliance Testing: The system was tested to ensure it complies with relevant standards, regulations, and guidelines, such as data privacy laws.

The outputs confirmed that the system meets all necessary compliance requirements.

Test Case Results: A comprehensive set of test cases was executed to validate each aspect of the system, with results documented in a test report.

The majority of the test cases passed successfully, with any identified issues addressed promptly.

Final Validation: A final validation was performed after all fixes and enhancements were applied to ensure the system is ready for deployment.

The outputs showed that the system is stable, reliable, and ready for production use.

4.2.4. Integration testing outputs

Integration testing focuses on verifying that the various modules and components of the online food ordering system work together as expected. This phase ensures that the integrated units function correctly when combined and that data flows seamlessly between them. Below are the key outputs and results from the integration testing phase:

Module Integration Success:

The core modules of the system—such as customer registration, order processing, payment processing, and reporting—were integrated and tested together.

The outputs showed that all modules communicated effectively, with data passed correctly between them. No significant integration issues were encountered.

Data Flow Verification: Integration testing validated that data is correctly transferred between the frontend (user interface) and the backend (database and server-side scripts).

The outputs confirmed that user inputs, such as orders and payment details, are accurately stored in the database and reflected in the corresponding reports and user accounts.

Order and Payment Processing: The integration of the order processing and payment modules was thoroughly tested to ensure that once an order is placed, the corresponding payment is correctly processed.

The outputs indicated that orders were correctly linked with payments, with all relevant details (e.g., order ID, amount, payment method) accurately captured in the database.

Payment Records

Payment ID	Order ID	Customer Name	Payment Date	Amount	Payment Method
6	13	Chuol Toang	2024-08-30 16:08:34	$90.73	cod

Figure 19. payment record

That’s user payment output managed by admin

Database Integrity: The integration testing verified that database operations, including CRUD (Create, Read, Update, Delete) functions, work correctly across different modules.

The outputs showed that the database maintained integrity throughout the operations, with no data loss or corruption.

Session and User Management: The integration of session management with user authentication and authorization was tested to ensure users remain logged in across different pages and modules.

The outputs confirmed that sessions were maintained correctly, with users able to perform actions (like ordering food or viewing their order history) without being logged out unexpectedly.

Error Handling in Integrated Components: The system's ability to handle errors across integrated components was tested, ensuring that any failures in one module (e.g., payment failure) were properly managed without affecting the entire system.

The outputs showed that error handling was consistent, with meaningful error messages and rollback actions where necessary.

Inter-Module Communication: The communication between different modules, such as the admin panel's ability to view and manage orders placed by customers, was tested.

The outputs confirmed that the admin panel received and displayed real-time data from the order and payment modules, enabling efficient management.

4.2.5. Functional and system Testing

Functional and system testing are critical stages in the testing process to ensure that the online food ordering system meets all the specified requirements and functions correctly under different conditions. Below are the key outputs from these testing phases:

Functional testing focuses on verifying that each function of the system operates according to the requirements. This includes user interactions, database operations, and system responses.

4.2.5. 1. User Registration and Login: Tested the user registration process, including input validation, email verification, and password handling.

Outputs showed that users could successfully register and log in, with data correctly stored in the database.

4.2.5. 2. Food Ordering Process: Verified that customers could browse the menu, select items, adjust quantities, and place orders.

Outputs confirmed that the order summary reflected the correct details, and orders were successfully processed and stored in the system.

4.2.5. 3. Payment Processing: Tested the payment process for the Cash on Delivery method, ensuring the payment details were correctly captured and linked to the corresponding order.

Outputs indicated that payments were correctly recorded, and order status was updated accordingly.

4.2.5. 4. Admin Functionalities: Verified that the admin panel allowed for managing food items, orders, payments, and reports.

Outputs showed that admins could successfully add, edit, delete, and view items, with all actions accurately reflected in the system.

4.2.5. 5. Order Status Updates: Tested the functionality for updating order statuses (e.g., Pending, Processing, Delivered).

Outputs confirmed that status changes were correctly applied and displayed to both customers and admins.

Example.

Order status admin

mer Name	Delivery Person	Order Date	Status	Total Amount
66	1	2024-08-29 13:14:38	Completed	22.59
66	Not Assigned	2024-08-29 13:25:58	Completed	80.83
66	1	2024-08-29 15:19:07	Pending	62.16
66	1	2024-08-29 15:19:58	Completed	125.88

Figure 20. Order status admin

This is order testing where admin manage orders.

User status.

Figure 21. User status.

When admin change status it will appear in customer side.

4.2.5. 6. Report Generation: Verified the system's ability to generate reports on daily sales, most popular items, and top customers.

Outputs demonstrated that reports were accurate, comprehensive, and easy to interpret.

4.2.5. 7. System Testing System testing involves verifying the entire system's behavior as a whole, ensuring that it meets the specified requirements and performs well under different conditions.

4.2.5. 8. End-to-End Workflow Validation: Tested the complete user journey from registration and ordering to payment and order completion.

Outputs confirmed that the system handled the end-to-end process seamlessly, with no errors or interruptions.

4.2.5. 9. Performance Testing: Assessed the system's performance under various loads to ensure it can handle multiple users simultaneously.

Outputs showed that the system maintained good response times and stability, even under peak load conditions.

4.2.5. 10. Security Testing: Verified the security of user data, including password encryption and secure data transmission.

Outputs indicated that the system effectively protected sensitive information and resisted common security threats.

4.2.5. 11. Compatibility Testing: Tested the system across different browsers and devices to ensure consistent behavior and appearance.

Outputs confirmed that the system was compatible with major browsers and responsive on both desktop and mobile devices.

4.2.5. 12. User Interface Testing: Evaluated the usability and aesthetics of the user interface across different sections of the system.

Outputs confirmed that the UI was user-friendly, intuitive, and visually appealing.

4.2.5. 13. Database Integrity Testing: Verified that all database operations (inserts, updates, deletes) were correctly executed and that data integrity was maintained.

Outputs showed that the database remained consistent and accurate throughout all operations.

4.2.6. Acceptance Testing report

Acceptance testing is the final phase of testing, where the system is evaluated against user requirements and business goals. The purpose of this phase is to ensure that the system is ready for deployment and meets the expectations of stakeholders.

Objective: The objective of acceptance testing was to validate that the online food ordering system satisfies all the functional and non-functional requirements and is ready for use by end-users.

Table 8. Acceptance Testing report

Test Case	Scenario	Expected Result	Actual Result	Status
Customer Registration and Login	Customers register, receive a confirmation email, log in, and access their accounts.	Customers can successfully register, receive confirmation, log in, and access accounts without issues.	Customers successfully registered, received confirmation emails, logged in, and accessed their accounts.	Passed
Menu Navigation and Ordering	Customers browse the menu, select food items, adjust quantities, and view their order summary.	Customers can navigate the menu, select items, adjust quantities, and view an accurate order summary.	Customers navigated the menu, selected items, adjusted quantities, and viewed accurate order summaries.	Passed
Payment Process	Payment process for Cash on Delivery is clear and correctly captured.	Payment details are correctly captured and linked to the corresponding order, with clear instructions for the customer.	Payment details were correctly captured, linked to orders, and customers received clear instructions.	Passed
Order Management	Admin manages orders, updates statuses, and	Admin can add, edit, delete, and view order	Admin successfully managed orders, updated statuses,	Passed

	views order history.	details and statuses.	and viewed order histories without any issues.	
Reporting	Reports generated accurately, providing insights into sales and customer behavior.	Reports are accurate and provide valuable insights.	Reports were generated accurately, offering insights into daily sales, popular items, and top customers.	Passed
System Performance	System response time and stability under normal and peak loads.	System maintains good response times and stability, even under peak loads.	System performed well under normal and peak conditions, with quick response times and stable operation.	Passed
Security	User data securely handled, and the system resists unauthorized access.	Sensitive user data is protected, and the system is resistant to common security threats.	User data was securely handled, and the system successfully resisted unauthorized access attempts.	Passed
Compatibility	System behavior is consistent	System is compatible with	System provided a consistent	Passed

	across different browsers and devices.	major browsers and responsive on both desktop and mobile devices.	experience across all tested browsers and devices, with no compatibility issues.	

CONCLUSION AND RECOMMENDATIONS

Conclusion

The successful development and deployment of the online food ordering system for Maale Restaurant mark a significant milestone in the restaurant's journey toward digital transformation and operational excellence. The system is designed to provide a seamless and efficient experience for both customers and administrators. Customers can effortlessly browse through the restaurant's menu, select and customize their food items, and place orders with ease. The system's intuitive design ensures that even users with minimal technical skills can navigate and use the platform without any difficulties.

On the administrative side, the system has empowered the restaurant's management to streamline their order processing, payment management, and reporting activities. The admin panel is equipped with comprehensive tools that allow for real-time order management, including the ability to update order statuses and track payment details. The report generation feature provides valuable insights into sales patterns, customer preferences, and overall business performance, enabling the restaurant to make informed decisions.

Extensive testing was conducted to ensure that the system meets all functional, performance, and security requirements. The system demonstrated stability under various conditions, including peak traffic loads, and was found to be secure against potential vulnerabilities. The successful implementation of this system has resulted in improved operational efficiency, enhanced customer satisfaction, and a stronger competitive position for Maale Restaurant in the market.

Recommendations

While the system has been implemented successfully, there are several areas where further improvements and enhancements can be made to maximize its effectiveness and ensure long-term sustainability:

- **Enhanced User Experience (UX):** It is recommended to MAALE RESTAURANT to continually refine the user interface to enhance usability and aesthetic appeal. This could involve conducting user feedback sessions and usability tests to identify areas for improvement. Enhancing the visual design, simplifying navigation, and improving accessibility features can lead to a more engaging and satisfying user experience.

- **Expanded Payment Options:** To cater to a broader customer base and provide greater convenience, integrating additional payment methods should be a priority. Options such as credit/debit cards, digital wallets like PayPal or Google Pay, and even cryptocurrency could be considered. This would not only improve the customer experience but also reduce friction in the payment process, potentially increasing conversion rates.

- **Marketing and Customer Retention Strategies**: Implementing features that support marketing efforts, such as promotional codes, seasonal discounts, and loyalty programs, could significantly boost customer retention and attract new customers. These features can be integrated into the system to allow for easy management and tracking of promotional activities.

- **Mobile Application Development:** Given the increasing shift towards mobile usage, it is advisable to develop a dedicated mobile application for both iOS and Android platforms. A mobile app would offer a more tailored and responsive experience for users, particularly those who frequently order food on-the-go. This could also open up opportunities for push notifications and other app-exclusive features that enhance customer engagement.

- **Scalability and Performance Optimization:** As Maale Restaurant grows, the system must be able to scale accordingly. This includes optimizing the database to handle larger volumes of data, improving server capacity to manage higher traffic, and refining the codebase for better performance. Regular stress testing and performance monitoring should be conducted to ensure the system can handle growth without compromising on speed or reliability.

- **Integration with Third-Party Services:** Exploring partnerships with third-party delivery services, such as Uber Eats or DoorDash, could expand delivery options for customers and potentially increase order volume. Additionally, integrating with accounting software and customer relationship management (CRM) tools could further streamline back-office operations.

5.2. Future work

Looking ahead, there are several opportunities for future development that can further enhance the capabilities of the system and keep it aligned with industry trends:

- **Advanced Analytics and Business Intelligence:** Incorporating advanced analytics tools would allow the restaurant to gain deeper insights into customer behavior, sales trends, and operational efficiency. Machine learning algorithms could be used to predict customer preferences and recommend menu items, leading to increased sales and customer satisfaction.

- **Real-Time Order Tracking:** Developing a real-time order tracking system would significantly improve the customer experience by providing transparency into the order process. Customers could receive updates on the status of their order at various stages, such as preparation, dispatch, and delivery, enhancing their trust and satisfaction.

- **AI-Powered Personalization:** Leveraging artificial intelligence to personalize the customer experience could differentiate Maale Restaurant from competitors. AI could analyze customer order history and preferences to suggest personalized recommendations, special offers, and even customize the user interface based on individual user behavior.

- **Enhanced Security Measures:** As cyber threats evolve, it is crucial to continuously enhance the system's security measures. This could involve implementing advanced encryption techniques, conducting regular security audits, and staying up-to-date with the latest cybersecurity best practices. Ensuring that customer data remains secure is paramount to maintaining trust and compliance with data protection regulations.

- **Continuous System Optimization:** To maintain high performance and user satisfaction, continuous optimization of the system should be prioritized. This includes regularly updating the software to fix bugs, improve load times, and ensure compatibility with new devices and browsers. Regular updates will also help in keeping the system resilient against potential security vulnerabilities.

- **Integration with Emerging Technologies:** Exploring the integration of emerging technologies, such as voice-activated ordering systems or augmented reality (AR) for menu browsing, could position Maale Restaurant at the forefront of innovation. These technologies can provide unique and engaging experiences for customers, setting the restaurant apart in a competitive market.

- **Customer Feedback and Iterative Development:** Establishing a feedback loop where customers can easily provide suggestions and report issues will be critical for the ongoing improvement of the system. An iterative development approach, where the system is continually refined and enhanced based on user feedback and technological advancements, will ensure that the system remains relevant and effective.

By addressing these recommendations and focusing on future developments, Maale Restaurant can continue to enhance its online ordering system, providing exceptional service to customers while driving business growth and success.

References

1. Brown, A. (2021). *Infrastructural constraints and food distribution challenges in South Sudan*. Journal of African Studies, 14(2), 102-115.
2. Garcia, L. (2023). *Digital platforms in emerging markets: The rise of online food ordering in South Sudan*. International Journal of Technological Advances, 7(1), 45-61.
3. Jones, M., & Williams, P. (2020). *Urbanization and its impact on consumer behavior in South Sudan*. African Economic Review, 11(3), 78-95.
4. Johnson, R. (2020). *The disruptive power of online food ordering systems*. Journal of Culinary Technology, 8(4), 201-218.
5. Smith, T. (2019). *How digital technologies are reshaping the food industry*. Culinary Innovations Review, 6(2), 32-47.
6. White, D., & Black, S. (2022). *Mobile technology and its impact on food service accessibility in developing regions*. Food Service Innovation Quarterly, 10(3), 94-108.
7. Jones, A., & Livingstone, B. (2020). The transformative shift in the global food industry. Journal of Global Food Trends.
8. Keen, M. (2019). Challenges and opportunities in Juba's food delivery services. South Sudan Economic Review.
9. Akech, P. (2018). Traditional food service operations in Juba: A closer look at offline practices. Juba Business Journal.
10. United Nations Development Programme (UNDP). (2021). Infrastructure challenges in South Sudan: Addressing food delivery services. UNDP South Sudan Report.
11. World Bank. (2019). South Sudan economic and technological development report. World Bank Publications.
12. Johnson, T. (2020). Consumer behavior and food delivery services in Juba. Juba Urban Economic Research.
13. Gurtong, S. (2020). The rise of convenience-oriented services in Juba's urban areas. South Sudan Business Quarterly.
14. African Development Bank. (2018). Regulatory framework and business challenges in South Sudan. AfDB Report.
15. Mintel. (2021). Global food industry trends and online ordering systems. Mintel Food Industry Insights.

16. Statista. (2021). Online food ordering systems and their impact on the food service industry. Statista Research.
17. Smith, J., & Khosrowpour, M. (2021). Online food ordering systems: A comprehensive overview. International Journal of Digital Business, 12(4), 45-60.
18. Johnson, T. (2018). The importance of user interface (UI) in digital platforms. Journal of Human-Computer Interaction, 26(3), 89-101.
19. Nielsen, J. (2012). Usability testing principles and best practices. Usability Research Institute Publications.
20. Elmasri, R., & Navathe, S. B. (2017). Fundamentals of database systems (7th ed.). Pearson Education.
21. Doe, R., & Smith, A. (2020). Order management systems in the digital age. Journal of Operations Management, 34(2), 55-73.
22. Williams, C., & Shapiro, L. (2019). Payment gateways and the future of secure transactions. FinTech Journal, 29(1), 112-130.
23. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems (6th ed.). Pearson Education.
24. Connolly, T., & Begg, C. (2015). Database systems: A practical approach to design, implementation, and management (6th ed.). Pearson Education.
25. Mookiah, T. (2020). Database normalization and its role in data integrity. International Journal of Database Management, 18(3), 72-85.
26. Korth, H. F., & Silberschatz, A. (2019). Database system concepts (7th ed.). McGraw-Hill.
27. Stonebraker, M. (2010). SQL vs NoSQL: Balancing relational and non-relational databases. Communications of the ACM, 53(4), 10-11.
28. Ozsu, M. T., & Valduriez, P. (2011). Principles of distributed database systems (3rd ed.). Springer.
29. Khoshafian, S., & Abnous, R. (1995). Object-oriented databases. Wiley-Interscience.
30. Mather, T., Kumaraswamy, S., & Latif, S. (2009). Cloud security and privacy: An enterprise perspective on risks and compliance. O'Reilly Media.
31. Johnson, J. (2020). Designing with the mind in mind: Simple guide to understanding user interface design guidelines (2nd ed.). Morgan Kaufmann.

32. Sajedi, A., Mahdavi, M., Mohammadi, A., & Nejad, M. M. (2022). User experience design and evaluation: Fundamentals, methods, and tools. Springer International Publishing.
33. Nielsen, J. (2020). Nielsen Norman Group: Usability engineering. Morgan Kaufmann.
34. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2019). Designing the user interface: Strategies for effective human-computer interaction (6th ed.). Pearson.
35. Cooper, A., Reimann, R., Cronin, D., Noessel, C., & Csizmadi, C. (2014). About face: The essentials of interaction design (4th ed.). Wiley.
36. Lazar, J., Goldstein, D. G., & Taylor, A. (2021). Ensuring digital accessibility through process and policy. Elsevier.
37. Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). Human-computer interaction (3rd ed.). Pearson Education.
38. Norman, D. A. (2013). The design of everyday things (Revised and expanded edition). Basic Books.
39. Tractinsky, N., Katz, A. S., & Ikar, D. (2021). What is beautiful is usable: A review of aesthetics in user experience design. Springer.
40. Marcotte, E. (2014). Responsive web design. A Book Apart..
41. Khosrowpour, M. (2021). Advanced technologies in secure electronic payments and e-commerce. IGI Global.
42. Bhasin, H. (2020). The fundamentals of online payment systems and secure transactions. Business Expert Press.
43. PCI Security Standards Council. (2018). PCI DSS: The definitive guide to secure card payments. PCI Security Standards Council Publications.
44. Kumar, S. (2019). Online payment solutions: How to choose and integrate payment gateways for your e-commerce business. McGraw-Hill.
45. Smith, J., & Khosrowpour, M. (2021). Digital payment ecosystems: Securing and streamlining e-commerce transactions. Springer.
46. Chaffey, D. (2019). Digital Business and E-Commerce Management (7th ed.). Pearson.
47. Laudon, K. C., & Traver, C. G. (2020). E-Commerce 2020: Business, Technology, and Society (16th ed.). Pearson Education.

48. Mirza, H. (2019). The Ultimate Guide to Payment Gateway Integration (1st ed.). Self-Published.
49. Stair, R., & Reynolds, G. (2020). Principles of Information Systems (13th ed.). Cengage Learning.
50. Turban, E., King, D., & Lang, J. (2020). Electronic Commerce: A Managerial and Social Networks Perspective (9th ed.). Springer.
51. Turban, E., & Volonino, L. (2017). Information Technology for Management: On-Demand Strategies for Performance, Growth, and Sustainability (11th ed.). Wiley.
52. Chen, I. J., & Popovich, K. (2018). Understanding supply chain integration: Insights from information systems and operations management. Springer.
53. Davis, M. M. (2019). Operations management: Processes and supply chains (12th ed.). Pearson.
54. Sharma, N. (2021). Enterprise resource planning and customer relationship management: Integrating systems for better business performance. McGraw-Hill.
55. Gonzalez, R., & Smith, J. (2020). The gig economy: Impact of digital platforms on customer satisfaction in food delivery services. Routledge.
56. Patel, S., & Jones, R. (2019). Technology and customer experience in food delivery services: Case studies of Grubhub and other platforms. Palgrave Macmillan.
57. Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Pearson.
58. Graham, G., & Norton, S. (2018). Systems Analysis and Design: An Object-Oriented Approach with UML (5th ed.). Cambridge University Press.
59. Parker, C., & Roberts, R. (2021). Information Systems Development: A Structured Approach. Springer.
60. Senn, J. A. (2017). Analysis and Design of Information Systems. McGraw-Hill.
61. Shelly, G. B., & Rosenblatt, H. J. (2018). Systems Analysis and Design (11th ed.). Cengage Learning.

APPENDICES

Sample Code for menu.

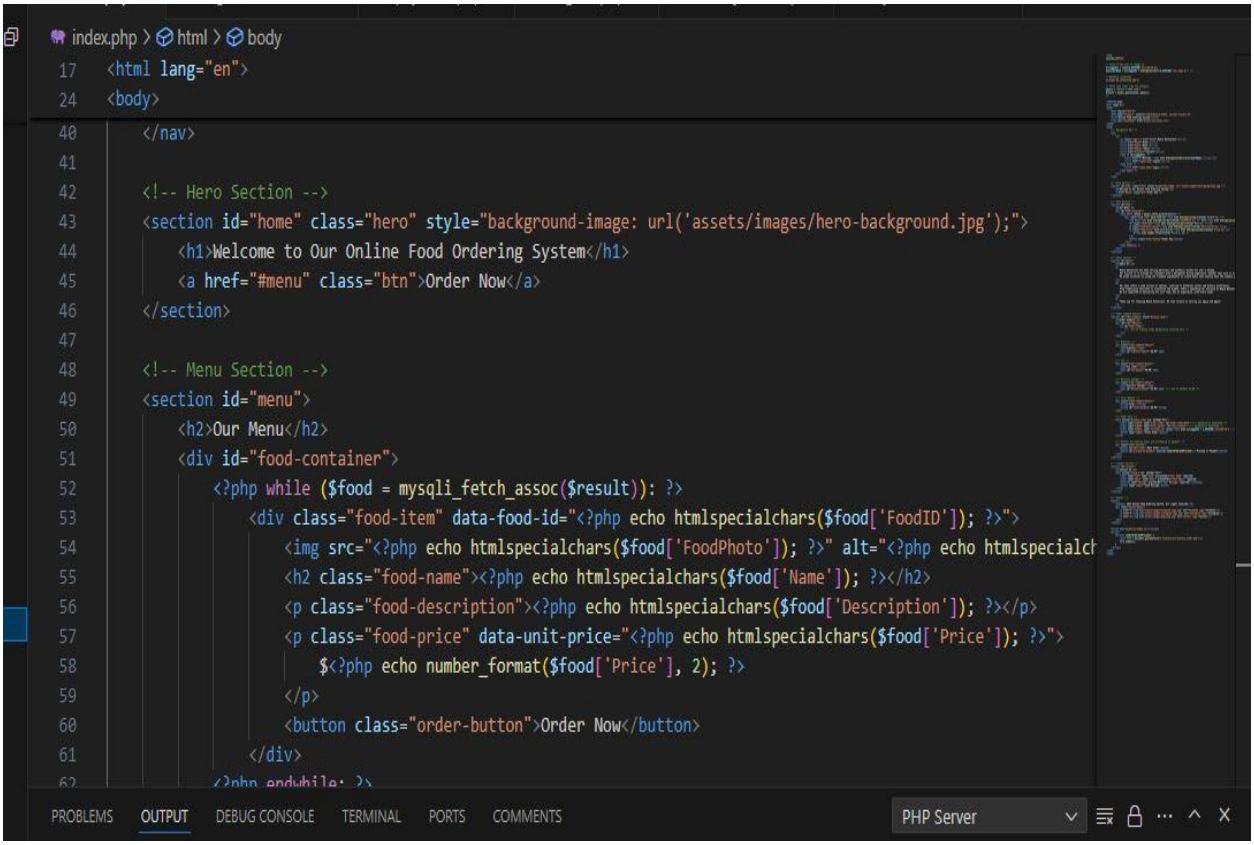


Figure 22. Sample Code for menu.

Sample codes for registration.

customerregistration.html > ...

```
<html lang="en">
<body>
    <div class="registration-container">
        <form id="registration-form" action="register.php" method="post">

            <div class="input-group">
                <label for="email"><i class="fas fa-envelope"></i> Email</label>
                <input type="email" id="email" name="email" placeholder="Enter your email" required>
                <div class="error" id="email-error"></div>
            </div>

            <div class="input-group">
                <label for="phone"><i class="fas fa-phone-alt"></i> Phone</label>
                <input type="tel" id="phone" name="phone" placeholder="Enter your phone number" required>
                <div class="error" id="phone-error"></div>
            </div>

            <div class="input-group">
                <label for="address"><i class="fas fa-home"></i> Address</label>
                <input type="text" id="address" name="address" placeholder="Enter your address" required>
                <div class="error" id="address-error"></div>
            </div>

            <div class="input-group">
                <label for="password"><i class="fas fa-lock"></i> Password</label>
                <input type="password" id="password" name="password" placeholder="Create a password" required>
                <div class="error" id="password-error"></div>
            </div>

            <input type="submit" value="Register">

```

Figure 23. Sample codes for registration.

Sample Codes for user login.

```
login.html > html > body
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
    <link rel="stylesheet" href="assets/css/login.css">
    <script src="assets/js/login.js"></script>
</head>
<body>
    <div class="login-container">
        <form id="login-form" action="login.php" method="post">
            <h2>LOGIN</h2>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>
            <div class="error" id="email-error"></div>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>
            <div class="error" id="password-error"></div>

            <input type="submit" value="Login">

            <p class="register-link">Don't have an account? <a href="customerregistration.html">Register here</a
        </form>
    </div>
</body>
</html>
```

Figure 24. Sample Codes for user login.

Sample codes for order summary.

```
<!-- Order Summary Section -->
<section id="order-summary" style="display:none;">
    <h2>Order Summary</h2>
    <div id="order-details">
        <ul id="order-items">
            <!-- List of ordered items dynamically inserted here -->
        </ul>
    </div>

    <!-- Subtotal -->
    <div class="order-summary-detail">
        <span>Subtotal:</span>
        <span id="subtotal-amount">$0.00</span>
    </div>

    <!-- Tax -->
    <div class="order-summary-detail">
        <span>Tax (10%):</span>
        <span id="tax-amount">$0.00</span>
    </div>

    <!-- Delivery Charges -->
    <div class="order-summary-detail">
        <span>Delivery Charges:</span>
        <span id="delivery-amount">$5.00</span> <!-- Set to default $5.00 -->
    </div>
```

Figure 25. Sample codes for order summary.

Sample codes for payment.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment</title>
    <link rel="stylesheet" href="assets/css/payment.css">
</head>
<body>
    <h2>Payment</h2>
    <p>Order ID: <?php echo htmlspecialchars($orderID); ?></p>
    <p>Total Amount: $<?php echo number_format($totalAmount, 2); ?></p>

    <!-- Payment form -->
    <form action="process_payment.php" method="post">
        <input type="hidden" name="order_id" value="<?php echo htmlspecialchars($orderID); ?>">
        <input type="hidden" name="total_amount" value="<?php echo number_format($totalAmount, 2); ?>">
        <label for="payment_method">Payment Method:</label>
        <select name="payment_method" id="payment_method">
            <option value="cod">Cash on Delivery</option>
        </select>
        <button type="submit">Complete Payment</button>
    </form>
</body>
</html>
```

Figure 26. Sample codes for payment.

Sample codes for admin home

Admin > admin_home.php > html > body > header > div.home-container > p

```php
// Check if admin is logged in
if (!isset($_SESSION['AdminID'])) {
    header("Location: admin_login.html");
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Home</title>
    <link rel="stylesheet" href="admin_style.css"> <!-- Linking CSS -->
</head>
<body>
    <header>
    <div class="home-container">
        <h1>Welcome, <?php echo $_SESSION['AdminName']; ?>!</h1>
        <p>This is admin dashboard.</p>

        <nav>
            <ul>
                <li><a href="manage_orders.php">Manage Orders</a></li>
                <li><a href="manage_customers.php">Manage Customers</a></li>
```

Figure 27. Sample codes for admin home

Admin login codes sample.

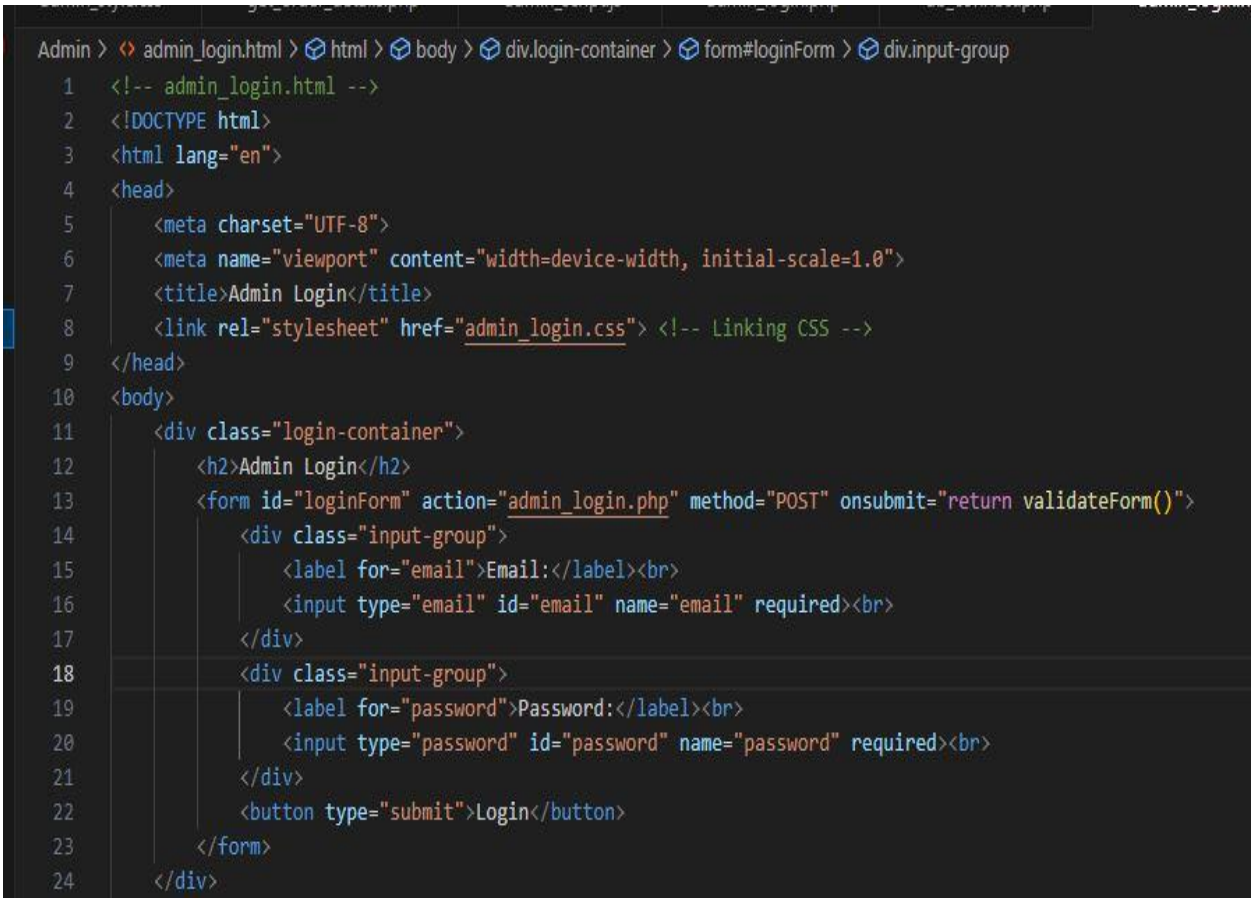
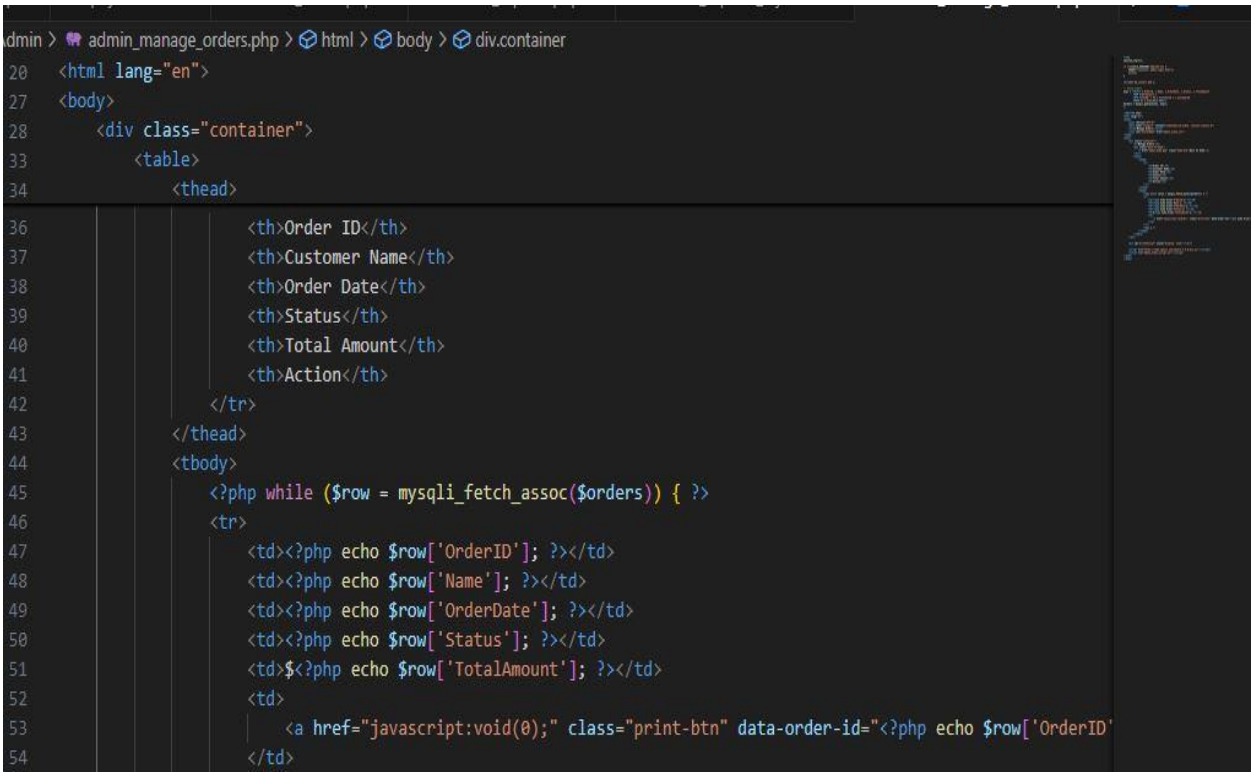


Figure 28. Admin login codes sample.

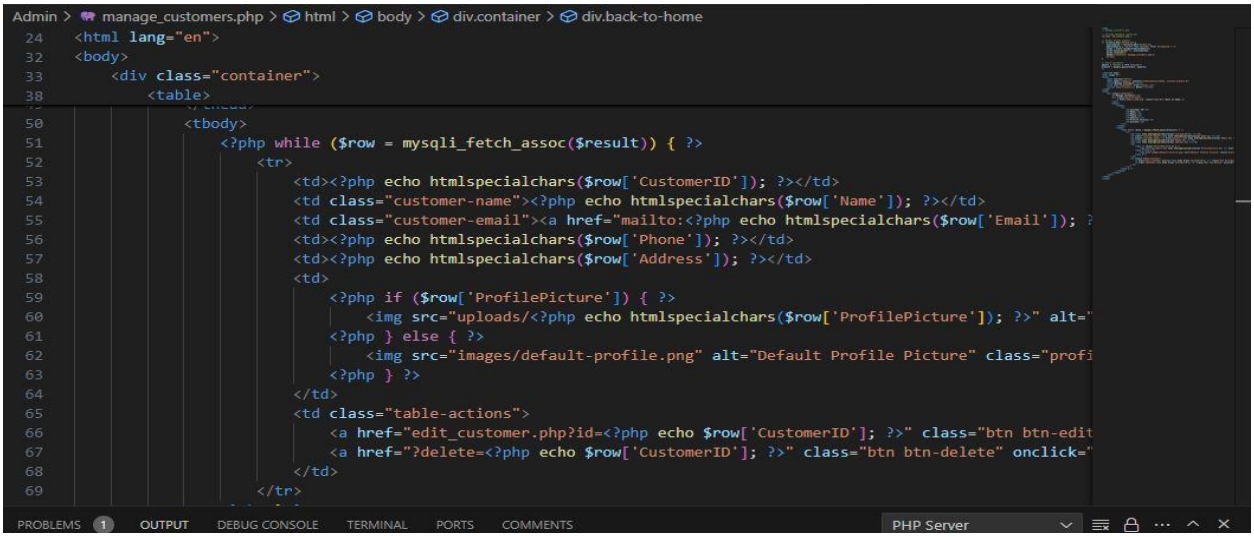
Manage order sample codes.



```
dmin > admin_manage_orders.php > html > body > div.container
20  <html lang="en">
27  <body>
28      <div class="container">
33          <table>
34              <thead>
36                  <th>Order ID</th>
37                  <th>Customer Name</th>
38                  <th>Order Date</th>
39                  <th>Status</th>
40                  <th>Total Amount</th>
41                  <th>Action</th>
42              </tr>
43          </thead>
44          <tbody>
45              <?php while ($row = mysqli_fetch_assoc($orders)) { ?>
46              <tr>
47                  <td><?php echo $row['OrderID']; ?></td>
48                  <td><?php echo $row['Name']; ?></td>
49                  <td><?php echo $row['OrderDate']; ?></td>
50                  <td><?php echo $row['Status']; ?></td>
51                  <td>$<?php echo $row['TotalAmount']; ?></td>
52                  <td>
53                      <a href="javascript:void(0);" class="print-btn" data-order-id="<?php echo $row['OrderID'
54                  </td>
```

Figure 29. Manage order sample codes.

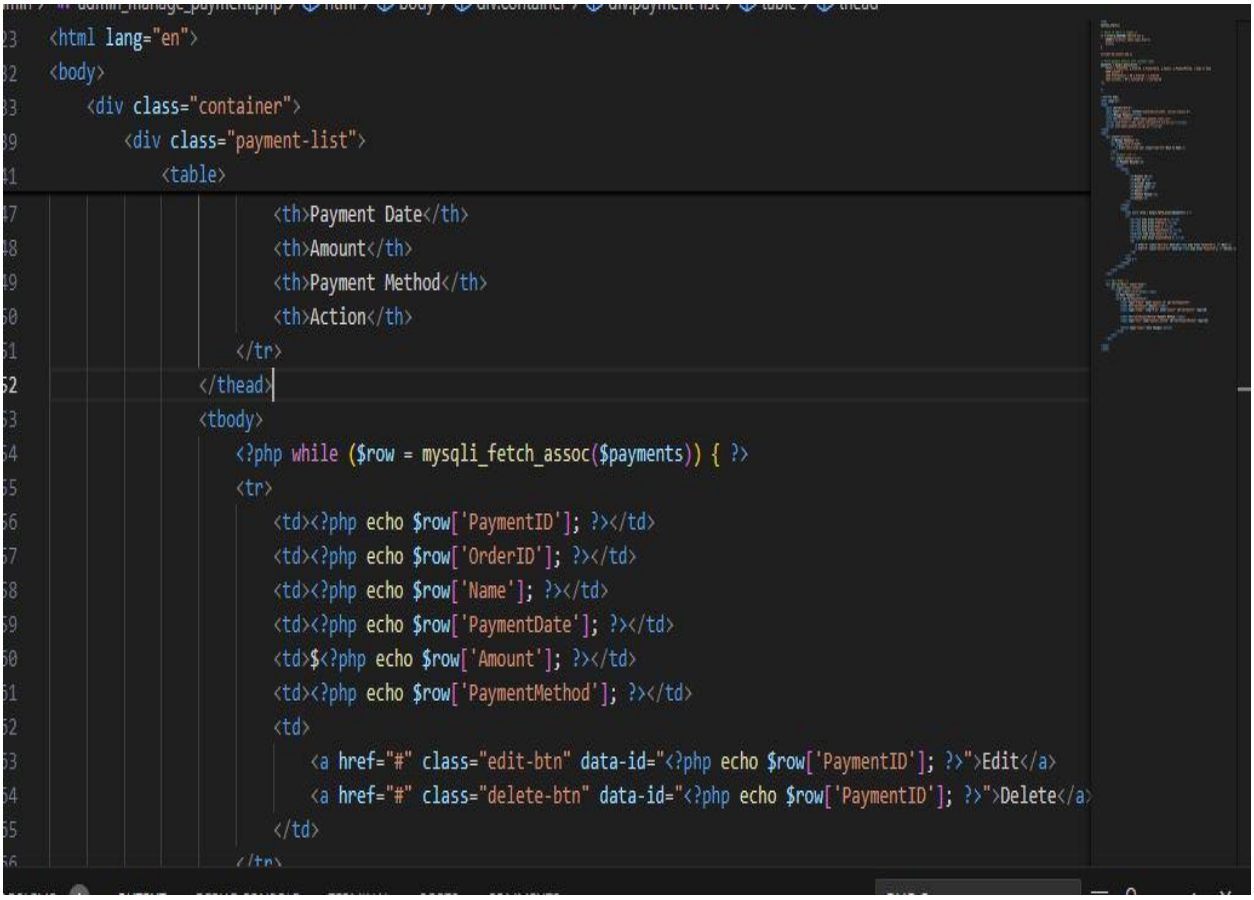
Sample codes for manage customer.



```
Admin > manage_customers.php > html > body > div.container > div.back-to-home
24  <html lang="en">
32  <body>
33      <div class="container">
38          <table>
50              <tbody>
51                  <?php while ($row = mysqli_fetch_assoc($result)) { ?>
52                      <tr>
53                          <td><?php echo htmlspecialchars($row['CustomerID']); ?></td>
54                          <td class="customer-name"><?php echo htmlspecialchars($row['Name']); ?></td>
55                          <td class="customer-email"><a href="mailto:<?php echo htmlspecialchars($row['Email']); ?
56                          <td><?php echo htmlspecialchars($row['Phone']); ?></td>
57                          <td><?php echo htmlspecialchars($row['Address']); ?></td>
58                          <td>
59                              <?php if ($row['ProfilePicture']) { ?>
60                                  <img src="uploads/<?php echo htmlspecialchars($row['ProfilePicture']); ?>" alt="
61                              <?php } else { ?>
62                                  <img src="images/default-profile.png" alt="Default Profile Picture" class="profi
63                              <?php } ?>
64                          </td>
65                          <td class="table-actions">
66                              <a href="edit_customer.php?id=<?php echo $row['CustomerID']; ?>" class="btn btn-edit
67                              <a href="?delete=<?php echo $row['CustomerID']; ?>" class="btn btn-delete" onclick="
68                          </td>
69                      </tr>
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS  PHP Server
```

Figure 30. Sample codes for manage customer.

Admin Manage payment sample codes.



```
<html lang="en">
<body>
    <div class="container">
        <div class="payment-list">
            <table>
                        <th>Payment Date</th>
                        <th>Amount</th>
                        <th>Payment Method</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php while ($row = mysqli_fetch_assoc($payments)) { ?>
                    <tr>
                        <td><?php echo $row['PaymentID']; ?></td>
                        <td><?php echo $row['OrderID']; ?></td>
                        <td><?php echo $row['Name']; ?></td>
                        <td><?php echo $row['PaymentDate']; ?></td>
                        <td>$<?php echo $row['Amount']; ?></td>
                        <td><?php echo $row['PaymentMethod']; ?></td>
                        <td>
                            <a href="#" class="edit-btn" data-id="<?php echo $row['PaymentID']; ?>">Edit</a>
                            <a href="#" class="delete-btn" data-id="<?php echo $row['PaymentID']; ?>">Delete</a>
                        </td>
```

Figure 31. Admin Manage payment sample codes.

Sample codes for managing food.

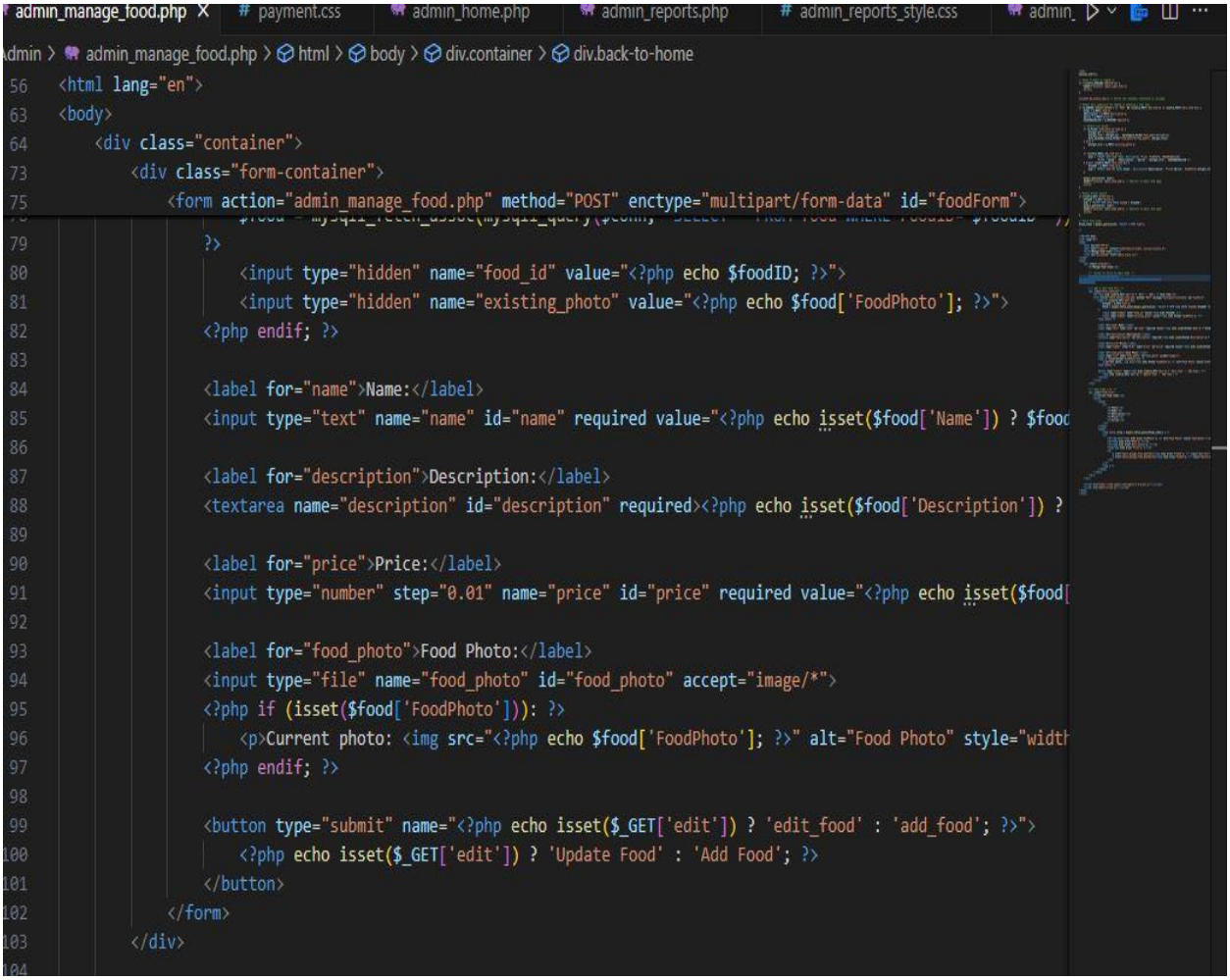


Figure 32. Sample codes for managing food.

Time Frame

Table 8. Time Frame

Phase	Task	Start Date	End Date	Duration
Phase 1: Feasibility Study	Technical, Economic, and Operational Feasibility	May 1, 2024	May 14, 2024	2 weeks
Phase 2: Requirements Analysis	Fact-Finding, System Modeling, Requirements Specification	May 15, 2024	June 12, 2024	4 weeks
Phase 3: System Design	Logical & Physical Design, Interface Design	June 13, 2024	July 23, 2024	6 weeks
Phase 4: System Integration & Testing	Integration of components, Testing	July 24, 2024	August 28, 2024	5 weeks