CHAPTER – 1

INTRODUCTION

In today's fast-paced world, digital solutions are reshaping industries, including the food and beverage sector. The QR code-based food ordering system is an innovative approach that leverages technology to enhance the dining experience. Integrating QR codes with an interactive digital menu minimizes the need for traditional menu cards and enables seamless interaction between customers and restaurants. Below, we outline key aspects of the introduction to this system.

1.1 PURPOSE OF THE SYSTEM

The primary aim of a QR code-based food ordering system is to streamline the ordering process, reduce manual dependencies, and improve customer satisfaction. The system eliminates physical menus, allowing customers to access digital menus on their smartphones by scanning a QR code. This modern solution supports real-time updates and promotes an efficient, eco-friendly approach to menu management.

1.2 ADVANTAGES OF QR CODE-BASED ORDERING

1. **Convenience:** Customers can browse menus, customize their orders, and place requests directly from their devices.
2. **Speed**: Orders are processed quickly without relying on waitstaff, reducing wait times and enhancing service quality.
3. **Hygiene**: A contactless solution minimizes physical touchpoints, ensuring safer dining in line with health and safety standards.
4. **Eco-Friendly**: Digital menus reduce the need for printed materials, contributing to sustainability efforts.

1.3 TECHNOLOGY STACK OVERVIEW

The development of the QR code-based food ordering system relies on a robust technology stack to ensure reliability, functionality, and user-friendliness.

- **HTML/CSS**: Used to design a responsive interface for the digital menu, adaptable to various screen sizes, from smartphones to tablets.
- **JavaScript**: Facilitates backend operations such as dynamic web content, QR code generation, and communication between the client and server.
- **SQL**: Manages and organizes the relational database for storing essential information, including menu items, customer details, and transaction histories.

1.4 PROBLEM STATEMENT AND MOTIVATION

Due to shared physical menus in traditional restaurant setups, customers face challenges such as delayed service, limited menu visibility, and hygiene concerns. This system addresses these issues by leveraging QR codes for digital interaction.

Problem Statement: "How can restaurants enhance efficiency and customer experience using a digital solution?"

Motivation: The widespread use of smartphones and the increasing need for contactless solutions inspired the development of this system. It bridges the gap between traditional service methods and modern customer expectations.

1.5 SCOPE OF THE SYSTEM

The QR code-based food ordering website caters to:

- **Dine-In Customers**: Offers quick and convenient ordering with real-time updates.
- **Restaurants**: Simplifies menu management, reduces operational overhead, and allows seamless customer communication.
- **Delivery/Takeout Services**: Enables customers to place and customize orders remotely.

1.6 METHODOLOGY

The development and implementation of the QR code-based food ordering system follow a systematic methodology to ensure a functional and efficient solution. The key steps involved are:

1. **Requirement Analysis**:
    - Identifying the needs of restaurants and customers.
    - Analyzing existing challenges in traditional ordering systems, such as delays, hygiene concerns, and inefficiencies.
2. **System Design**:
    - Creating a modular architecture with separate components for the user interface, backend logic, and database management.
    - Designing wireframes and prototypes for the user-friendly digital menu.
3. **Technology Selection**:
    - Choosing technologies like HTML/CSS for the front end, JavaScript for backend logic and QR code generation, and SQL for database management.
4. **Development**:
    - Implementing the user interface to ensure responsiveness and cross-device compatibility.
    - Coding the backend functionalities to process orders, manage QR code interactions, and integrate database operations.
5. **Testing and Debugging**:
    - Conducting functional testing for order processing and menu navigation.
    - Ensuring cross-browser and cross-platform compatibility for the digital menu.

1.7 TARGET AUDIENCE

The QR code-based food ordering system is designed to cater to the following groups:

1 **Restaurant Owners/Managers**: To simplify menu management, enhance operational efficiency, and improve customer satisfaction.

2 **Customers**: Individuals seeking a convenient and hygienic way to order food using their smartphones or tablets.

3 **Developers and IT Professionals**: For those interested in implementing or enhancing similar systems in various dining environments.

4 **Food Industry Stakeholders**: Consultants, franchise owners, and investors keen on adopting modern, tech-driven solutions to improve business operations.

1.8 REPORT CONTENTS

The following sections of the report will be explored in detail:

Chapter 2: System Design and Architecture: Overview of the technological framework and tools used for development.

Chapter 3: Front-End Development: Detailed explanation of the user interface design using HTML and CSS for responsiveness.

Chapter 4: Back-End Functionality: Role of JavaScript in server-side processing, QR code generation, and integration.

Chapter 5: Database Management: Implementation of SQL to store and manage menu items, customer orders, and transaction details.

Chapter 6: Features and Benefits: Highlighting the unique aspects and advantages of the system.

Chapter 7: Challenges and Limitations: Discussion of potential issues faced during development and deployment.

Chapter 8: Conclusion and Future Scope: Insights into the impact of the system and potential enhancements for future iterations.

CHAPTER – 2

SYSTEM DESIGN AND ARCHITECTURE

The QR code-based food ordering system relies on a robust and scalable architecture to ensure efficient operation and seamless user experience. This section outlines the system's design components, their interconnections, and the underlying technologies that power the system.

2.1 ARCHITECTURAL OVERVIEW

The system follows a three-tier architecture to separate concerns and enhance scalability:

1. **Presentation Layer:** The user interface (UI) is designed using HTML, CSS, and JavaScript.
2. **Logic Layer:** Server-side processing and backend logic using JavaScript (Node.js/Express).
3. **Data Layer:** A relational database implemented using SQL for data storage and management.

2.2 SYSTEM COMPONENTS

a) USER INTERFACE:

- **Responsiveness:** Designed to work seamlessly across various devices such as smartphones, tablets, and desktops.
- **Ease of Navigation**: Includes features like category filters, search functionality, and dynamic updates for user convenience.
- **Digital Menu**: Displays menu items with details such as images, prices, and descriptions.

b) QR CODE INTEGRATION:

- **QR Code Scanning:** Allows users to access the menu instantly through any smartphone camera or QR code scanner app.

c) BACKEND LOGIC:

- **Order Processing:** Handles order submission, modifications, and cancellations.

- **Dynamic Content**: Updates menu items in real-time based on availability or special offers.
- **Automation**: Includes features like automatic notifications for order confirmation and estimated preparation time.

d) DATABASE DESIGN:

- **Menu Management**: Stores menu items with details such as name, category, price, and availability status.
- **Order Management**: Tracks customer orders, including status and table number.
- **Transaction Records**: Logs payment details for analysis and auditing.

e) SYSTEM WORKFLOW:

- **Menu Access:** Customers scan the QR code, which redirects them to the digital menu hosted on the website.
- **Order Placement**: Customers select items, customize orders (if needed), and submit them.
- **Backend Processing**: The system validates the order, communicates with the database, and updates the status.
- **Kitchen Notification**: The kitchen staff receives the order details for preparation.
- **Payment**: Customers pay through integrated online or in-person payment gateways.
- **Completion**: The system marks the order as complete and updates transaction records.

2.3 TOOLS AND TECHNOLOGIES

- **Front-End Development:** HTML5, CSS, JavaScript.
- **Backend Development**: Javascript for server-side logic.
- **Database Management**: SQL
- **QR Code Library:** JavaScript-based libraries for QR code generation and processing.
- **Hosting**: Web server or cloud platforms

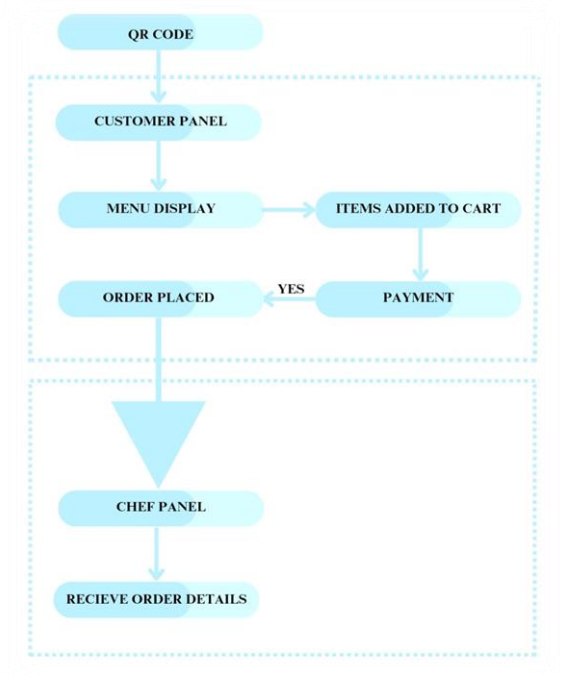

Figure2. 1: System architecture

The image depicts a flowchart illustrating the process of a QR code-based food ordering system. Here's a description of the steps:

- **QR Code**: Customers scan the QR code to initiate the process.
- **Customer Panel**: After scanning, they are directed to a customer interface.
- **Menu Display**: The menu is displayed on the interface, allowing customers to browse items.
- **Items Added to Cart**: Customers select items, which are added to their cart.
- **Order Placed**: Customers confirm their selection and proceed to place the order.
- **Payment**: The system processes the payment if required.
- **Chef Panel**: Once the order is confirmed, details are sent to the chef's panel.
- **Receive Order Details**: The chef receives the order details to prepare the meal.
- The flowchart visually segments the process into two main parts: the **customer's interaction** and the **chef's panel** operations.

CHAPTER – 3

FRONT-END DEVELOPMENT

The front-end development of the QR code-based food ordering system uses HTML and CSS to ensure a responsive, user-friendly, and visually appealing interface. Below is a concise explanation of the design approach.

3.1 OBJECTIVES

- Ensure compatibility across various devices.
- Provide a modern and easy-to-navigate interface.
- Enhance visual appeal while maintaining functionality.

3.2 TECHNOLOGIES USED

1. **HTML**: Defines the structure of the web pages, including forms, buttons, and navigation.
2. **CSS**: Styles and formats the layout, incorporating animations and responsive design.

3.3 KEY DESIGN FEATURES

1. **DIGITAL MENU LAYOUT:**

- Grid-based display with images, item names, prices, and "Add to Cart" buttons.
- Category tabs or dropdowns for easy navigation.

2. **RESPONSIVE DESIGN**:

- Media queries adjust layouts for devices like smartphones and tablets.
- Scalable fonts and buttons for better usability.

3. **INTERACTIVE ELEMENTS**:

- Hover effects on buttons and items for visual feedback.
- Real-time cart updates without page reloads.

4. **NAVIGATION AND ACCESSIBILITY:**

- **Navigation Bar:** Quick access to menu, cart, and support options.
- **Accessibility**: Keyboard navigation, high-contrast colors, and alt text for images to ensure inclusivity.

5. TOOLS AND FRAMEWORKS:

- **Bootstrap:** For rapid and responsive UI development.
- **Font Awesome**: Provides user-friendly icons for buttons and navigation.

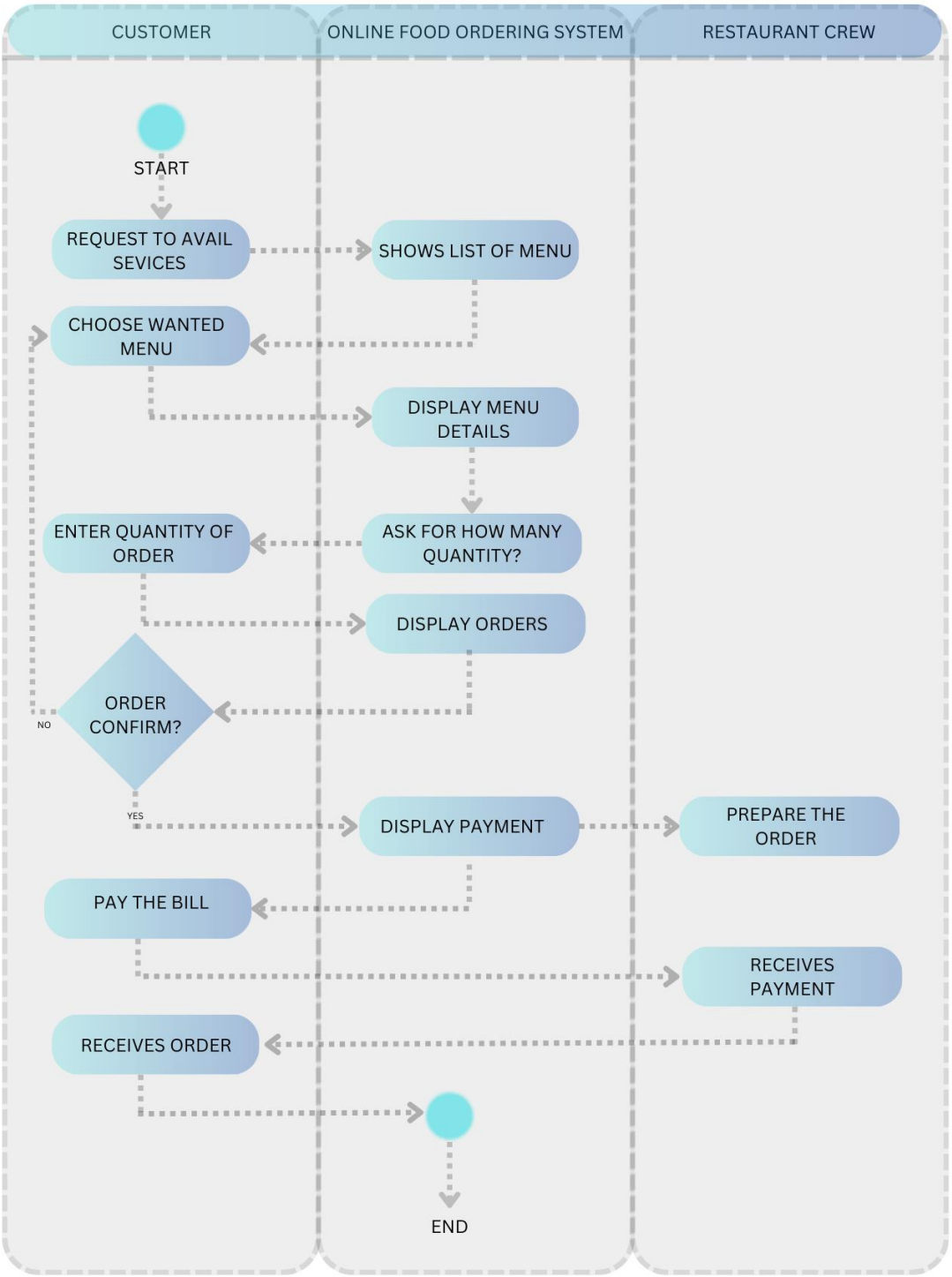

Figure3.3: Activity diagram

CHAPTER – 4

BACKEND IMPLEMENTATION

The backend of the **QR Code-Based Food Ordering System** was built using **PHP** as the server-side scripting language and **XAMPP** as the development environment. It manages customer orders, interacts with the database, and ensures secure and efficient data handling.

TECHNOLOGY STACK:

- **PHP:** Handles server-side logic and database interactions.
- **XAMPP**: Provides a local environment with Apache, PHP, and MySQL for testing and development.
- **MySQL Database**: Stores structured data such as menu items, orders, and customer details.

CORE FUNCTIONALITIES:

1. **ORDER PROCESSING:**
- Processes customer orders by validating inputs and saving order details, such as menu items, quantities, and table numbers, in the database.
2. **MENU MANAGEMENT:**
- Fetches dynamic menu data from the database and updates availability.
3. **DATABASE INTEGRATION**:
- Stores menu data and order details in dedicated table (Orders) using SQL queries.
4. **ERROR HANDLING:**
- Validates input data and provides error feedback for issues like exceeding item availability or invalid entries.

WORKFLOW:

1. **PLACING AN ORDER**:
- Customers select menu items via the HTML interface.
- Order data is sent to the backend, validated, and stored in the Orders table.
2. **CONFIRMATION**:
- Upon successful storage, a confirmation message is sent to the frontend.

CHAPTER – 5

DATABASE MANAGEMENT

Database management is a critical component of the QR code-based food ordering system, ensuring efficient data storage, retrieval, and organization. SQL (Structured Query Language) is utilized to implement a relational database that supports essential functionalities like storing menu items and managing customer orders.

DATABASE DESIGN:

The database is designed with a single table for storing order details, as the front end directly handles menu data. The structure of the table is as follows:

ORDERS TABLE:

COLUMNS:

- order_id: A unique identifier for each order (primary key).
- table_number: The table number provided by the customer.
- order_details: Details of the selected menu items, including names and quantities (stored in JSON or text format).
- total_price: The total cost of the order, calculated by the backend.
- order_time: The timestamp when the order was placed.

PURPOSE: Captures and stores all the essential details for each order.

DATA STORAGE AND RETRIEVAL:

1. **STORAGE**:

- When an order is placed, the backend PHP script inserts the validated data into the Orders table using SQL queries.
- All order details are securely stored to ensure accurate tracking and retrieval.

2. **RETRIEVAL**:

- Data retrieval is implemented for viewing or analyzing past orders if needed.
- SQL queries are optimized to fetch specific order details based on criteria such as order_id or table_number.

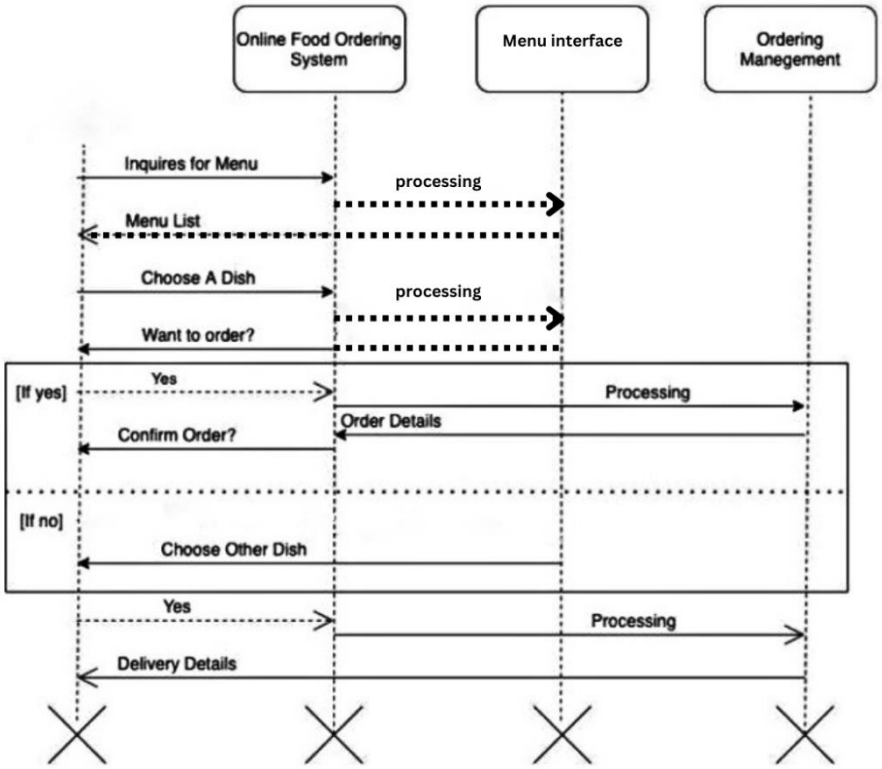

Figure5.1: Sequence diagram

CHAPTER – 6

FEATURES AND BENEFITS

The QR code-based food ordering system offers a range of features designed to enhance user convenience, improve operational efficiency, and provide a modern dining experience. Below is a detailed explanation of the system's unique aspects and advantages, categorized into features and benefits.

6.1 KEY FEATURES

a. QR CODE INTEGRATION:

- **Universal QR Code**: A single QR code provides access to the digital menu for all customers.
- **Customer Input**: Customers enter their table number during the ordering process for proper service.

b. DIGITAL MENU:

- **Customizable Content:** Real-time updates for items, prices, and availability.
- **Visual Appeal**: High-quality images and descriptions of menu items.

c. RESPONSIVE DESIGN:

- **Device Compatibility:** Optimized for smartphones, tablets, and desktops.
- **Touch-Friendly Interface**: Designed for ease of use on mobile devices.

d. ORDER PROCESSING:

- **Customer-Specific Input:** Collects table numbers, special requests, and food preferences.

e. PAYMENT INTEGRATION:

- **Secure Transactions:** Utilizes encryption for data security.
- **Multiple Options:** Customers can choose from digital wallets, UPI, or cash on delivery.

6.2 BENEFITS:

a. FOR CUSTOMERS:

- **CONVENIENCE:** Quickly access the menu and place orders from their devices.
- **PERSONALIZED EXPERIENCE:** Enter table numbers and special requests for precise service.

- **TIME EFFICIENCY:** Reduces waiting times for both ordering and payment processes.

b. FOR RESTAURANTS:

- **STREAMLINED WORKFLOW:** Orders are tagged with table numbers, reducing errors.
- **REDUCED COSTS:** Eliminates dependency on printed menus and minimizes staffing requirements.

c. ENVIRONMENTAL IMPACT:

- **PAPERLESS OPERATIONS:** Encourages eco-friendly practices by reducing printed menus and receipts.

6.3 UNIQUE SELLING POINTS:

a. SCALABILITY:

- Can handle increasing customer numbers and expanded menu options seamlessly.

b. ACCESSIBILITY:

- Easy-to-use interface suitable for all customer demographics.

6.4COMPARATIVE ADVANTAGES:

- **Over Traditional Systems**: Simplifies order-taking with reduced dependency on staff.
- **Over Table-Specific QR Systems**: Offers more flexibility as customers provide their table information dynamically.

CHAPTER – 7

CHALLENGES AND LIMITATIONS

Testing is a crucial phase in the development lifecycle of the QR code-based food ordering system. While testing ensures reliability, usability, and functionality, it also reveals potential challenges and limitations. This chapter focuses on the issues encountered during testing, categorized by different testing methodologies and stages.

7.1 TESTING CHALLENGES:

a. FUNCTIONAL TESTING:

- **Incomplete Test Cases:**

  Certain edge cases, like incorrect table number entries, may be overlooked.

  Solution: Develop comprehensive test cases covering all possible user interactions

- **Dynamic QR Code Testing**::

  Verifying that the QR code correctly directs users to the intended digital menu and supports all device types.

  Solution: Simulate multiple scenarios across various devices and browsers.

b. PERFORMANCE TESTING:

- **High Traffic Simulation:**

  Difficulty in replicating peak traffic conditions to test the system's stability.

  Solution: Use modern performance testing tools like K6 or Gatling to simulate heavy loads and evaluate system stability.

- **Server Latency**:

  Slow response times for order submissions under high server load.

  Solution: Optimize server queries and implement caching mechanisms.

c. USABILITY TESTING:

- **Customer Accessibility Issues**

  Ensuring that customers with minimal technical skills can navigate the system.

  Solution: Perform usability tests with diverse customer groups.

- **Device compatibility:**

  Testing responsiveness on various screen sizes and operating systems.

Solution: Conduct cross-platform testing using emulators and physical devices.

d. SECURITY TESTING:

- **Data Privacy Concerns:**

  Testing for vulnerabilities in customer data storage and payment systems.

  Solution: Conduct penetration testing and enforce encryption standards.

- **Injection Attacks:**

  Protecting the system from SQL injection.

  Solution: Use automated tools to identify vulnerabilities and patch them.

e. INTEGRATION TESTING:

- **Module Interoperability:**

  Errors arising from the interaction between the front-end, back-end, and database.

  Solution: Use test automation tools like Selenium for continuous integration testing.

- **Real-Time Updates:**

  Testing synchronization between customer orders and kitchen interfaces.

  Solution: Test with dummy data in real-time environments.

7.2 LIMITATIONS IDENTIFIED DURING TESTING:

a. DEPENDENCY ON EXTERNAL FACTORS:

- **Device Variability:**

  Older devices or browsers may not fully support the system's features.

- **Internet Connection:**

  The system is heavily reliant on stable internet connectivity.

b. LACK OF REAL-WORLD TESTING SCENARIOS:

- **Customer Behavior:**

  Simulating real-world scenarios, such as large groups using the system simultaneously, is challenging.

- **Peak-Time Load:**

  Difficult to replicate the exact conditions of a fully operational restaurant during busy hours.

c. MAINTENANCE OVERHEADS:

- **Test Updates:**
  New features or changes require re-evaluating and updating test cases.
- **Bug Fixing**:
  Fixing one bug may introduce others, leading to extended testing cycles.

7.3 STRATEGIES TO ENHANCE TESTING:

a. TEST ENVIRONMENT OPTIMIZATION:

- Create test environments that closely mimic live deployments, including server configurations and customer interactions.

b. ITERATIVE TESTING:

- Continuously test and refine the system after each major update or feedback cycle.

7.4 BENEFITS OF RIGOROUS TESTING:

- **Reliability:** Ensures that the system performs as expected under all conditions.
- **Customer Satisfaction**: Identifies usability issues and resolves them before deployment.
- **Security Assurance**: Protects sensitive customer and transactional data from breaches.
- **Operational Efficiency**: Ensures smooth integration of all system components, reducing downtime during deployment.

Testing is integral to identifying and addressing the challenges and limitations of the QR code-based food ordering system. With a systematic and thorough testing approach, the system can be refined into a robust and user-friendly platform, minimizing deployment issues and enhancing its overall effectiveness.

7.5 TEST CASE:

Below are sample test cases for various components:

Table 7.1: Test case

Test Case ID	Description	Test Steps	Expected Result	Status
TC-001	Scan QR Code	Scan QR code using a supported device	User is redirected to the corresponding menu page.	Pass
TC-002	Place an Order	Confirm selected item and submit the order	Order is submitted successfully, and a confirmation message displays.	Pass
TC-003	Menu Display and Availability	Access the menu page after scanning the QR code	Menu items are displayed accurately, with availability status.	Pass
TC-004	Handle Missing Data During Order Placement	Try to place an order with missing customer details (e.g., table number)	Error message prompts user to fill in missing details.	Pass
TC-005	Database Storage Verification	Place an order and verify data is stored in the database	Order details are saved correctly in the NoSQL database.	Pass
TC-006	Add Items to Cart	Select menu items and add them to the cart	Items are added to the cart, and the total price updates correctly.	Pass
TC-007	Responsive website testing	Open the website on various devices	Website layout adjusts and functions correctly across all the devices.	Pass
TC-008	Limited Quantity Validation	Attempts to add more items than are available in the cart.	Error message is displayed limiting the quantity to the available stock	

CHAPTER – 8

RESULTS AND DISCUSSION

RESULTS:

The QR Code-Based Food Ordering System was successfully implemented with the following outcomes:

1. **Efficient Order Placement**: Customers could seamlessly scan QR codes to access the menu, select items, and place orders without needing any external assistance.
2. **User-Friendly Interface**: The web interface, designed using HTML, CSS, and JavaScript, provided an intuitive experience for users, ensuring smooth navigation and interaction.
3. **Accurate Data Storage**: Order details, including menu items, customer inputs (e.g., table number), and total cost, were successfully stored and managed using a NoSQL database, ensuring reliability and scalability.
4. **Simplified Management**: The system minimized manual errors and streamlined the ordering process, enhancing efficiency and user satisfaction.

DISCUSSION:

- **Technology Selection**: NoSQL databases proved advantageous for handling semi-structured data efficiently, particularly in managing dynamic menu items and orders.
- **Challenges Faced**:
    - Initial difficulties were encountered in designing the QR code functionality and ensuring compatibility across various devices.
    - Integration of the database required precise handling of schemas to avoid data inconsistencies.
- **Solutions Adopted**:
    - Thorough testing of the QR code scanning process ensured reliability across multiple devices.
    - Implementation of robust error-handling mechanisms addressed potential issues with missing or invalid data during order placement.
- **Future Scope**:
    - Expanding the system to include advanced analytics for restaurant management.

- Introducing a feedback module to gather customer reviews post-order completion.

The project met its primary objectives, delivering a streamlined, digital ordering experience while maintaining simplicity and ease of use. Feedback from the team and initial testers highlighted the system's efficiency and potential for scalability.

- This QR code allows customers to access the menu by simply scanning it with their smartphone. It provides a seamless and contactless way to browse, select, and place orders directly from their devices.

Figure 8.1: QR Code for accessing the menu

- Upon scanning the QR code, customers are directed to the restaurant's responsive website. The homepage features a clear 'View Menu' option, allowing users to navigate various food categories and place their orders effortlessly.

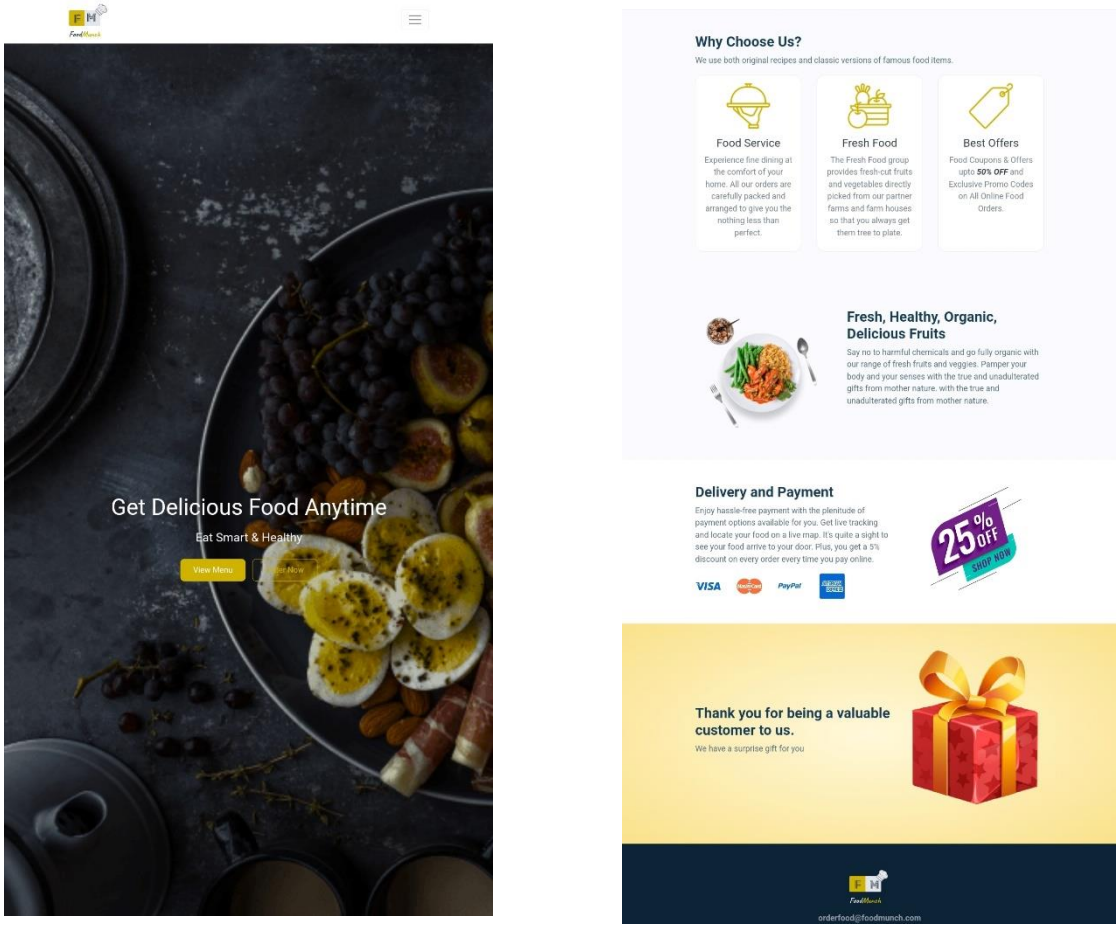

Figure 8.2: Website homepage displayed after scanning the QR code, featuring the 'View Menu' navigation option.

- The 'Explore Menu' section categorizes the menu into Veg Starters, Non-Veg Starters, Drinks, Desserts, Soups, Curries, and Noodles, providing customers with a well-organized interface to browse and select their preferred dishes easily.

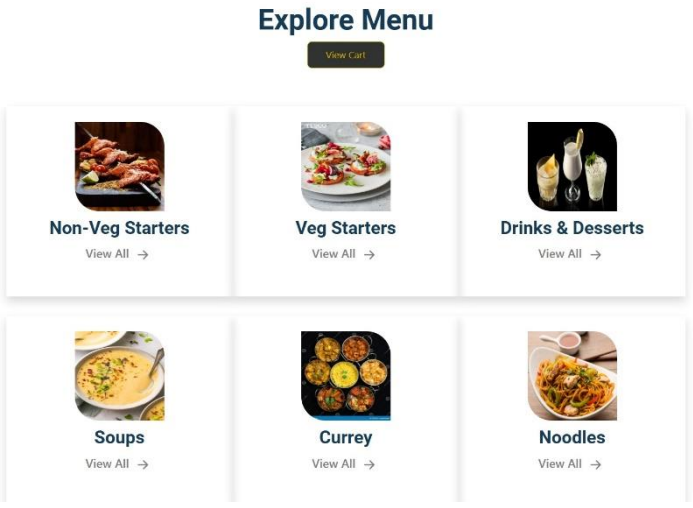

Figure 8.3: 'Explore Menu' section

- When customers navigate to 'View More,' all the items under the selected category are displayed in a detailed format, showcasing item names, descriptions, prices, and an option to add them to the cart. This ensures a user-friendly browsing and ordering experience.

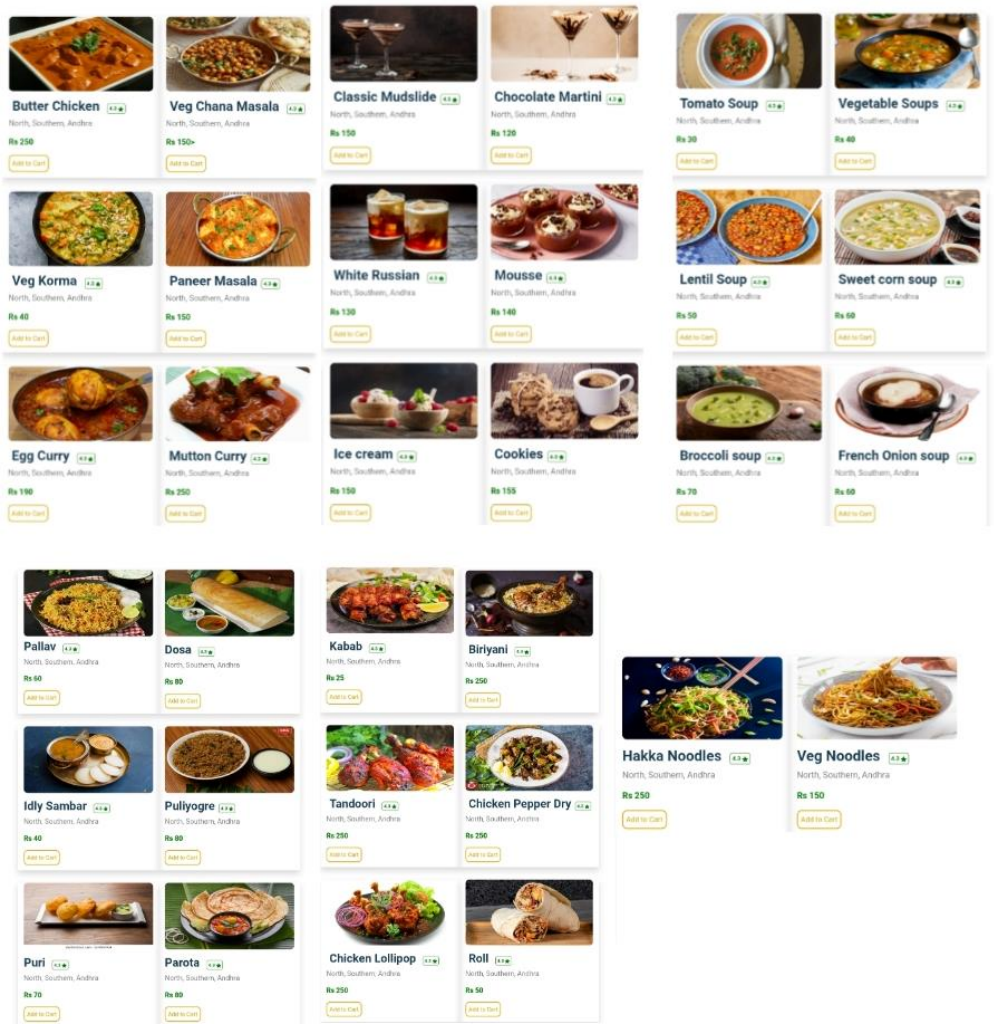

Figure 8.4: Detailed view of menu items under a selected category

- The cart section displays the selected items, their quantities, and the total amount calculated based on the quantity and price of each item. It also includes a 'Remove' option to delete items from the cart. Two buttons, 'Close' and 'Order,' are provided. The 'Close' button exits the cart, while the 'Order' button redirects users to the order form to complete their purchase.

Figure 8.5: Cart section showing selected items, total amount calculation, and options to remove items or proceed to order.

The order form allows customers to finalize their food choices by reviewing the items in their cart, entering necessary details like table number, and confirming their order. It ensures smooth communication between the customer and the restaurant, allowing for a quick and efficient order processing.

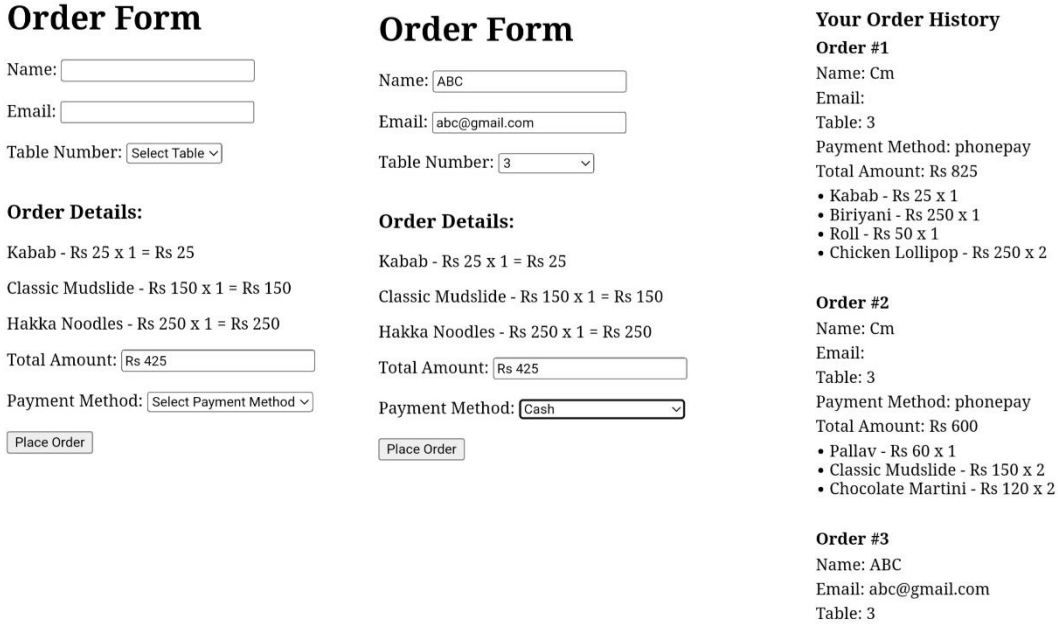

Figure 8.6: Order form displaying item details, table number input, and a final 'Submit Order' button to confirm the order.

In the backend, the XAMPP server is utilized to store order details in the database. The order data, including item names, quantities, total price, and table numbers, is stored in structured tables. This ensures efficient data retrieval and management for processing orders seamlessly.

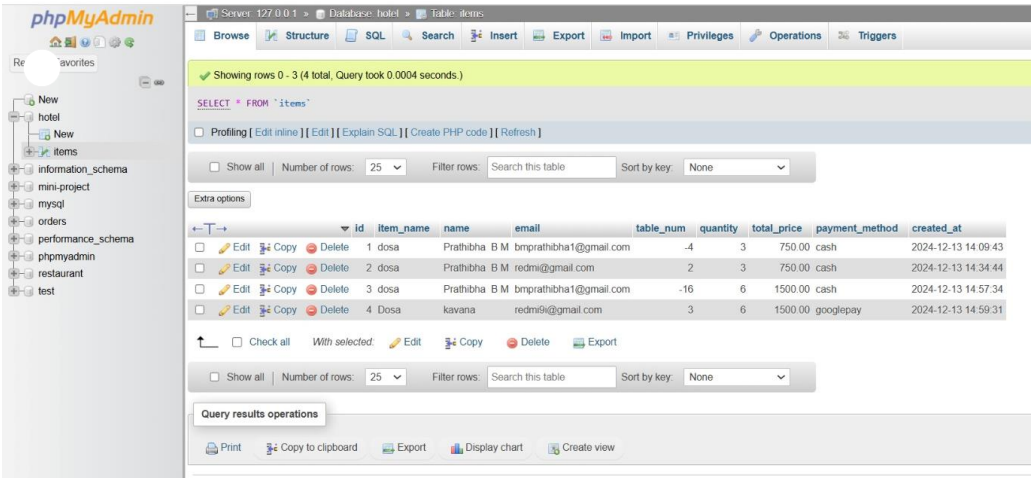

Figure 8.6: XAMPP database displaying stored order details, including item names, quantities, table numbers, and total prices.

CHAPTER – 9

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION:

The QR Code-Based Food Ordering System successfully simplified the food ordering process, enhanced customer experience, and minimized manual errors. By allowing customers to scan a QR code and place orders directly through a user-friendly web interface, the system improved operational efficiency and reduced the reliance on traditional ordering methods. The use of NoSQL databases ensured reliable storage and management of dynamic order and menu data, making the system scalable and robust.

The implementation process demonstrated the importance of integrating diverse technologies like HTML, CSS, JavaScript, and NoSQL databases to create a cohesive solution. The project highlighted the potential for technology-driven improvements in the food and hospitality industry, providing a foundation for further innovations.

FUTURE ENHANCEMENTS:

1. **Integration with Payment Gateways**: Adding online payment options for customers to pay directly through the system, offering convenience and reducing transaction times.
2. **Dynamic Menu Updates**: Enabling real-time updates to the menu to reflect changes in item availability, special offers, or pricing.
3. **Multi-Language Support**: Expanding the system's accessibility by incorporating support for multiple languages to cater to a diverse customer base.
4. **Feedback Module**: Introducing a feature to collect customer feedback after their dining experience for continuous improvement.
5. **Data Analytics**: Implementing analytics tools to generate reports on customer preferences, peak ordering times, and popular menu items for restaurant management.
6. **Mobile Application**: Developing a mobile app version of the system for increased accessibility and user convenience.

7. **Table-Specific QR Codes**: Upgrading the system to include table-specific QR codes to automatically record table numbers, simplifying the ordering process further.

By addressing these enhancements, the system can evolve into a more comprehensive and versatile solution, catering to broader customer and business needs.

REFERENCES

NxtWave CCBP 4.0:

Website: https://www.ccbp.in/

Description: A comprehensive platform for learning full-stack development and enhancing programming skills, providing resources and practical exposure in technologies like HTML, CSS, JavaScript, and more.

ChatGPT

Organization: OpenAI

Description: Used for generating project ideas, writing pseudo-codes, creating documentation, and providing implementation guidance.

Website: https://chat.openai.com/

GitHub Links

QR Code Generation Example: https://github.com/davidshimjs/qrcodejs

Responsive Web Design Template: https://github.com/tailwindlabs/tailwindcss

PHP and Database Example: https://github.com/fzaninotto/Faker