A

MAJOR PROJECT II

(PROJ-CSE403)

ON

3FE RESTAURANT

A major report submitted in the partial

fulfillment of the requirement for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

SESSION (2021-2025)

SUBMITTED TO:	**SUBMITTED BY:**
CSE DEPARTMENT	**RAHUL KUMAR**
	B. TECH CSE 8TH SEM
	(21097116780016)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAN NAYAK CHOUDHARY DEVI LAL COLLEGE OF ENGINEERING

SIRSA, HARYANA

Department of Computer Science and Engineering

JAN NAYAK CH. DEVI LAL VIDYAPEETH SIRSA

CANDIDATE DECLARATION

I, **Rahul Kumar** declare that the work presents in this project entitled as **“3FE Restaurant – Web-based Project”** submitted in the partial fulfilment of the requirement for the award of the degree of Bachelor of Technology (CSE). It is an authentic record of work as part of 8th semester major project part II. It is further certified that this complete project has been checked by plagiarism software, the similarity index as per university norms and it contains 8% plagiarism.

RAHUL KUMAR

Department of Computer Science and Engineering

JAN NAYAK CH. DEVI LAL VIDYAPEETH SIRSA

CERTIFICATE FROM SUPERVISOR

This is to certify that the project entitled **“3FE Restaurant – Web-based Project”** being submitted by **Rahul Kumar**, Roll No: - **21097116780016** for the partial fulfilment of the requirement for award of the degree of the Bachelor of Technology (CSE) has been carried out by his under my supervision satisfactorily.

I wish him all success.

Date:

Place: Sirsa **Supervisor**

PREFACE

This project presents the design and development of a **3FE Restaurant – Web-based Project** for **3FE (Epic Family Eats)**, using core web technologies: **PHP**, **HTML**, **CSS**, and **JavaScript**. The system aims to streamline restaurant operations, enhance efficiency, and provide a modern digital solution for daily management tasks such as order processing, table booking, menu management, and staff coordination.

With the increasing demand for digital transformation in the hospitality industry, this project was conceptualized to demonstrate how technology can simplify the complex workflows within a restaurant environment. The project uses 3fe as a case model due to its reputation for operational excellence and customer focus.

Developing this system has deepened my understanding of full-stack web development, database connectivity, and user interface design within a real-world context.

ACKNOWLEDGEMENT

I would like to thank everyone associated with this project. Many people have supported me in successfully completing it.

I extend my sincere thanks to **Er. Sheenu Sachdeva** (Project Guide), CSE Department, and **Er. Krishan Kumar** (HOD, CSE Department), under whose guidance I learned a great deal. Their suggestions and direction were instrumental in the completion of this project.

Finally, I would like to thank my parents and friends for their valuable suggestions, guidance, and support throughout the various stages of the project.

Rahul Kumar
(21097116780016)

ABSTRACT

This project involves the development of a **web-based restaurant management system** tailored for **3FE Restaurant**, a leading specialty coffee and restaurant brand. Built using **PHP for backend processing**, **HTML and CSS for layout and styling**, and **JavaScript for interactivity**, the system is designed to improve restaurant workflow and customer service.

Key features of the system include:

- A dynamic menu management interface
- Order and billing system
- Table reservation module
- Admin dashboard for staff and inventory control
- Responsive design for accessibility across devices

The project addresses the limitations of traditional manual processes and highlights how a custom software solution can enhance operational efficiency, reduce errors, and improve customer experience. MySQL was used for the database to manage structured data effectively.

This work demonstrates how digital tools can be customized to meet the specific needs of a restaurant business and provides a scalable foundation for future enhancements such as analytics, customer feedback integration, and mobile app support.

TABLE OF CONTENT

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION OF PROJECT

1.1 Introduction

Introduction to 3FE Restaurant Management System

The **3FE Restaurant Management System** is a comprehensive web-based application designed to simplify and optimize the operations of a modern restaurant. It provides a seamless experience for all key stakeholders, including **staff**, **customers**, and **administrators**. The system facilitates various processes such as table management, reservations, orders, billing, and staff coordination, ensuring that both the customer experience and restaurant operations are smooth, efficient, and well-managed.

1.2 Key Components and Features:

1. **Staff:**

   The **staff** of the restaurant play an integral role in ensuring smooth day-to-day operations. The system allows staff members to access the orders, track reservation details, and manage table assignments in real-time. With the 3FE system, staff can:

   - View current orders and manage their processing.
   - Update table statuses (e.g., occupied, free).
   - Manage customer requests and order modifications.

2. **Customer:**

   The **customer** experience is a central focus of the 3FE Restaurant System. Customers can interact with the system via an intuitive, user-friendly interface. Key features for customers include:

   - Menu Browsing: Customers can explore the menu, view dish descriptions, prices, and images.
   - Reservation Management: Customers can make reservations online for a specific date, time, and number of people.

- **Order Placement:** Customers can place their orders online or in the restaurant, adding selected items to their cart.
- **Bill Payment:** After dining, customers can review their bills and make payments online or at the restaurant.

3. Admin (Administrator):

The **admin** or restaurant manager has full control over the restaurant’s operations. Admins can manage the restaurant's resources, employees, menu items, orders, and reservations. Key functions of the admin include:

Menu Management: The admin can add, edit, or remove items from the menu, set prices, and update descriptions.

- **Reservation Management:** Admins can view all customer reservations, manage available time slots, and allocate tables.
- **Order Management:** Admins can track the status of orders placed by customers, assign them to staff, and mark them as complete.
- **Reporting and Analytics:** The admin can generate reports on sales, customer preferences, and operational efficiency.

4. Reservation:

The **reservation** feature allows customers to book a table in advance. Customers can select their desired date, time, and table for the reservation, ensuring they have a guaranteed spot at the restaurant. For the restaurant, this feature helps avoid overbooking and allows efficient table management. Features include:

- **Customer Reservation Form:** A form for customers to submit their reservation details.
- **Reservation Confirmation**: Customers receive a confirmation email or notification with reservation details.
- **Table Allocation:** The system helps assign available tables based on reservation timing and party size.

5. Order:

The **order** system is an essential component that tracks customer orders from placement to fulfillment. Customers can select food and beverages from the menu, add items to their order, and submit it for preparation. Staff can view the current orders and update their status as they are processed. Key features include:

- **Order Placement:** Customers can select items from the digital menu, add them to the cart, and place an order.
- **Order Tracking:** The system allows staff to track the progress of each order, ensuring timely and accurate service**.**
- **Order Modification:** Customers can modify or cancel their orders before the food preparation begins.

6. Bill:

The bill feature generates an itemized summary of the customer’s orders. Once the customer finishes their meal, the system calculates the total cost, including any taxes or service charges. Customers can view their bills and make payments online or directly at the restaurant. The bill system includes:

- **Bill Generation:** An automatic calculation of the total amount based on the customer’s order.
- **Payment Options:** Multiple payment methods, such as credit/debit cards, digital wallets, or cash, are supported for seamless transactions.
- **Bill History:** Customers can view past bills for reference or reordering.

7. Table:

The table management feature helps both customers and staff track table availability. Customers can view table availability when making reservations, while staff can manage and allocate tables as needed. The system enables:

- **Table Availability:** A real-time view of available and occupied tables in the restaurant.
- **Table Assignment:** Staff can assign tables to customers based on reservation data or walk-in status.
- **Occupancy Tracking:** Staff can update the status of a table (e.g., "Occupied," "Available," "Reserved").

CHAPTER 2

REQUIREMENT ANALYSIS

A Requirements Analysis for a restaurant management system (such as "3fe") involves identifying and documenting the key functional and non-functional needs of the system to ensure its effectiveness and smooth operation. For the sake of this analysis, let's break down the requirements into core categories such as:

2.1Functional Requirements

These outline the system's behavior, features, and tasks it needs to perform.

a. Reservation Management

- **Online reservation:** Allow customers to book tables online, select time slots, and choose party sizes.
- **Table management:** Automatically allocate tables based on availability and preferences (e.g., by size or location).
- **Waitlist management:** If no table is available, customers can join a waitlist, with notifications sent when a table becomes free.

b. Order Management

- **Menu management:** Restaurant managers and chefs should be able to update and manage the menu, including prices and descriptions of food/drinks.
- **Customer orders:** Ability to take customer orders, customize them (add special requests), and send them directly to the kitchen or bar.
- **Order status tracking:** Track the progress of each order in real-time (pending, preparing, ready for pickup).

c. Billing and Payments

- **Check management:** Generate customer checks, itemize their orders, and provide itemized totals.
- **Payment processing:** Integrate with various payment methods like credit/debit cards, mobile wallets, or cash.
- **Split checks:** Allow customers to split their bills easily.

d. Inventory Management

- **Stock tracking:** Track the inventory of ingredients and drinks, update in real-time as orders are placed.
- **Automatic alerts:** Notify staff or managers when stock is low and needs to be replenished.
- **Supplier management**: Manage suppliers and order ingredients from them directly through the system**.**

e. Staff Management

- **Employee scheduling**: Allow managers to create and adjust employee work schedules, track hours, and assign roles.
- **Payroll:** Automate payroll calculation based on hours worked or salary agreements.
- **Role-based access control:** Different access levels for staff (e.g., waitstaff, kitchen staff, managers).

f. Customer Management

- **Loyalty program:** Track customer visits and reward loyal customers with discounts, points, or special promotions.
- **Customer feedback:** Collect and analyze customer feedback to improve service.
- **Order history:** Allow customers to view their past orders and reorder easily.

g. Reporting and Analytics

- **Sales reports:** Track revenue, profit, and performance by item, category, or time period.
- **Employee performance reports:** Track staff performance (sales, efficiency, customer satisfaction).
- **Inventory analysis**: Generate reports about stock usage, waste, and future stock needs.

2.2Non-Functional Requirements

These requirements are related to system performance, security, and usability.

a. Scalability

- The system should be able to handle growth, both in terms of users (e.g., a larger number of reservations) and the number of restaurants if it is part of a chain.

b. Usability

- The system should have a user-friendly interface for both customers and restaurant staff (easy-to-navigate dashboard, mobile app, etc.).
- Multi-language support: Depending on the restaurant's location, there may be a need for multiple language options**.**

c. Reliability and Availability

- The system should have high uptime and be able to handle peak periods (e.g., lunch and dinner rush).
- Backup and recovery: Ensure that data is regularly backed up and that the system can recover from failures quickly**.**

d. Security

- Data encryption: All sensitive customer and payment data should be encrypted.
- Compliance with regulations: Ensure that the system adheres to privacy laws like GDPR and PCI-DSS for payment security.

e. Integration

- The system should integrate with third-party services like payment gateways, POS (point-of-sale) systems, and online reservation platforms (e.g., OpenTable).
- Social media integration: Integration for marketing purposes, allowing customers to share reviews or promotions.

f. Performance

- The system should provide fast response times (for customer orders, reservations, etc.), even with a high number of concurrent users.

2.3 User Stories

Here are some examples of user stories that might come up in a restaurant management system:

- **As a customer,** I want to be able to make a reservation online, so I don’t have to wait when I arrive at the restaurant.
- As a waiter, I want to be able to take orders quickly and send them directly to the kitchen, so the food can be prepared without delay.

- As a manager, I want to be able to see the restaurant’s sales in real-time, so I can make decisions about inventory and staffing.

2.4 System Architecture

A web-based application is ideal for a restaurant management system, with the following key components:

- **Frontend:** HTML, CSS, JavaScript (with a framework like React or Angular) for dynamic pages and responsiveness.
- **Backend**: PHP (possibly using frameworks like Laravel or Symfony for faster development) to handle the logic and interactions with the database.
- **Database:** MySQL or PostgreSQL to store reservations, orders, inventory, staff details, and customer data.
- **Payment Gateway:** Integration with Stripe, PayPal, or other payment providers for processing transactions.

2.5 Social Media Integration (optional)

For sharing promotions, reviews, or reservations on platforms like Facebook or Instagram.

- **Facebook Graph API**: Can be used to share customer reviews, posts, or promotions to social media.

2.6 Analytics and Reporting

- **Google Analytics:** For tracking visitor traffic on the restaurant’s website.
- **Custom Reporting:** For generating reports on sales, inventory, staff performance, etc., using PHP's built-in libraries or third-party tools like Charts.js for displaying graphs.

2.7 Technology Stack

For a PHP-based system, the typical technology stack would look like:

Frontend:

- HTML5, CSS3, JavaScript, and frameworks like React or Vue.js for a dynamic and responsive user interface**.**

Backend:

- PHP (using frameworks like Laravel or Symfony for rapid development and robust architecture).

Database:

- MySQL or PostgreSQL for relational data storage**.**

Payment Integration:

- Stripe, PayPal, or other payment gateway APIs.

Hosting/Server:

- Apache or Nginx on a Linux-based server (e.g., LAMP stack)**.**

CHAPTER 3

SOFTWARE REQUIREMNTS

Software Requirements for 3FE Restaurant Management System

To develop the 3FE Restaurant Management System in PHP, we need to define the necessary software components that will ensure the system works efficiently, securely, and scalably. Below is a detailed list of the software requirements:

3.1 Operating System Requirements

- **Development Environment**:
  Windows 10/11, macOS, or any Linux distribution (Ubuntu recommended)
  Minimum 4 GB RAM, dual-core processor
  Compatible with XAMPP/LAMP stack
- **Server Deployment (optional/production)**:
  Ubuntu Server 20.04+ or CentOS
  Apache/Nginx with PHP and MySQL support
  Minimum 2 vCPUs, 2–4 GB RAM

3.2Frontend Software Requirements

- HTML5 **– Page structure and content**
- CSS3 **– Styling and responsive layout**
- JavaScript (ES6+) **– Client-side interactivity, DOM manipulation**
- Bootstrap (optional) **– Responsive design framework**
- Font Awesome **– Icons for UI elements**

3.3 Backend Software Requirements

- **PHP 7.4 or later** – Server-side scripting

- **MySQL 5.7 or later** – Relational database system
- **Apache 2.x** – Web server for local or production hosting
- **phpMyAdmin** – Web interface for MySQL database management
- **XAMPP/WAMP/LAMP –** Local development stack (bundled PHP, MySQL, Apache)

3.4 Cloud Integration Software (Optional for Live Hosting)

- **Firebase** – For real-time database (optional future enhancement)
- **AWS EC2 or Digital Ocean** – For cloud-based hosting
- **Amazon S3 / Cloudinary** – For image uploads and static file storage
- **Cloudflare** – For performance optimization and security

3.5 API and HTTP Testing Tools

- **Postman** – API testing and response validation
- **cURL** – Command-line tool for API requests
- **Insomnia** – Alternative REST client for debugging HTTP endpoints
- **Browser DevTools** – Network tab for live request/response monitoring

3.6 Version Control and Collaboration Tools

- **Git** – Source code version control
- **GitHub / GitLab / Bitbucket** – Remote code repositories
- **VS Code + Live Share** – Code editor with collaboration plugins
- **Trello / Jira / Notion** – Task tracking and project management

3.7Optional Tools and Enhancements

- **AJAX** – For asynchronous page updates (order status, reservations)
- **jQuery** – Lightweight library for DOM handling (if needed)
- **Chart.js** – For generating graphical sales reports or analytics

- **SweetAlert** – For enhanced alert messages
- **DataTables.js** – For dynamic tables (reservations, orders)

3.8 Software Licenses

Software/Tool	**License Type**	**Usage**
PHP	Open Source (PHP License)	Backend processing
MySQL	Open Source (GPL)	Database management
Apache	Open Source (Apache License 2.0)	Web server
HTML/CSS/JS	Free/Open standard	Frontend structure
Bootstrap	MIT License	Responsive design
Git	Open Source (GPL)	Version control
Postman	Freemium	API testing
Font Awesome	Free & Pro (paid)	Icons
VS Code	Freeware (Microsoft)	Code editor

CHAPTER 4

TECHNOLOGY ANALYSIS

For the **3FE restaurant project**, using **PHP, CSS, HTML, JavaScript (JS), and MySQL** is a great choice for building a dynamic and responsive restaurant management system. Here's how each of these technologies can be applied:

4.1. Frontend Development

These technologies form the user-facing portion of your website or app.

a. HTML (HyperText Markup Language)

- **Purpose**: HTML will define the structure and content of your website or web app. It's used to create forms, tables, buttons, headers, and other interactive elements.
- **Usage**:
- **Reservation forms**: For customers to book tables.
- **Menu display**: Presenting your menu items in a user-friendly manner.
- **Order forms**: Allowing waitstaff to enter customer orders into the system.

b. CSS (Cascading Style Sheets)

- **Purpose**: CSS is responsible for styling the HTML content. It controls the layout, colors, fonts, and overall visual appearance.
- **Usage**:
- **Styling the reservation system**: Making the booking page visually appealing and easy to navigate.
- **D4signing the order and bill forms:** Ensuring that the customer and staff interfaces are clean and user-friendly.
- **Responsive Design**: Ensuring that the website or app looks good on both mobile and desktop devices.

c. Java Script (JS)

- **Purpose**: JavaScript makes the website interactive by adding dynamic behavior, such as updating content without reloading the page, validating forms, and handling user inputs.

- **Usage:**
- **Interactive reservations:** Handling calendar selections and time slot availability in real-time.
- **Order management:** Dynamically updating the cart, adding or removing items, and displaying prices as the customer selects food.
- **AJAX calls:** Sending and receiving data between the client (frontend) and server (backend) without reloading the page, such as checking table availability or updating the order status.

4.2 Backend Development

This is where the core functionality of your restaurant management system will reside, such as processing reservations, managing orders, and handling billing.

PHP (Hypertext Preprocessor)

- **Purpose**: PHP is a server-side scripting language used to handle the backend logic, interact with the database, and process user requests. It helps generate dynamic web pages based on user inputs or database queries.
- **Usage:**
- **Database Interaction:** PHP will handle queries to the MySQL database, such as fetching menu items, processing reservations, and storing order details.
- **Session Management:** PHP can track the user session, for example, when a customer logs in to the system for their reservation or order.
- **Order Processing:** When an order is placed, PHP handles the logic for sending it to the kitchen, updating the order status, and calculating the total bill.

4.3. Database Management

MySQL is used to manage the restaurant's data, ensuring efficient storage and retrieval of critical information like reservations, menu items, orders, and customer details.

a. MySQL Database

- **Purpose:** MySQL is a relational database management system that stores and manages the data used by the restaurant's management system.
- **Usage**:
- **Tables**: You'll need multiple tables for different aspects of the system:
- **Customers Table:** Stores customer details (e.g., name, contact information).
- **Menu Table:** Stores information about the food items (e.g., name, description, price).
- **Orders Table:** Stores the order details (e.g., customer ID, menu items ordered, quantity).
- **Reservations Table:** Stores reservation details (e.g., customer ID, date, time, table number).
- **Bill Table:** Stores the final bill (e.g., total amount, payment status, tips).
- **Queries:** MySQL will handle the retrieval, insertion, and modification of data:
- Fetch available tables for reservations.
- Calculate the total order cost.
- Update inventory or stock levels as orders are processed.

4.4 Integration of Technologies

- **PHP and MySQL Integration:** PHP scripts will interact with the MySQL database to perform actions such as creating, reading, updating, and deleting records.
- **AJAX and JavaScript with PHP:** JavaScript can send AJAX requests to PHP scripts for background processing, like checking table availability or adding items to an order without reloading the page**.**
- **Form Validation:** JavaScript will handle client-side validation (e.g., making sure required fields are filled in before submitting), while PHP can perform server-side validation for additional security.

Example Workflow in the System:

- **Customer Reservation:**
  Frontend: A customer fills out a reservation form on a webpage built with HTML, CSS, and JavaScript.

Backend: PHP handles the form submission and checks table availability in the MySQL database.

Database: The reservation details are stored in the MySQL database.

- **Order Management**:

  Frontend: Waitstaff enters customer orders via a webpage built with HTML and JavaScript.

  Backend: PHP processes the order, calculating the total cost and adding it to the database.

  Database: The order is stored in the MySQL database, including the customer's chosen items and quantities.

- **Bill Generation:**

  Frontend: After the meal, the server generates the bill using a web page.

  Backend: PHP calculates the bill, adding taxes and tips as needed.

  Database: The bill details are updated and stored in the MySQL database.

- **Payment Processing:**

  Frontend: Customers pay online or at the restaurant via a payment page (HTML/JS).

  Backend: PHP handles the payment request and updates the database with the payment status.

  Database: The transaction status (paid/unpaid) is recorded in the database.

Benefits of Using These Technologies:

PHP: It's open-source, flexible, and widely used for web development, making it a good fit for handling backend logic and database interactions.

CSS & HTML: These technologies provide a clean, responsive interface, ensuring a seamless user experience across devices.

JavaScript: JS helps make the application interactive and dynamic, providing a more engaging experience for both customers and staff.

MySQL: MySQL is powerful for managing relational data efficiently, ensuring that all restaurant information (reservations, orders, inventory) is well-organized.

System Analysis for the 3FE Restaurant Project

System analysis is a critical phase in understanding the project's needs, defining the structure of the system, and ensuring all components work together efficiently. For the **3FE Restaurant** system, a detailed analysis helps define system requirements, process flows, and the interactions between various modules (reservation, order, billing, and database management).

Below is a breakdown of the system analysis for the **3FE restaurant project**:

4.5 Overview of the System

The 3FE restaurant management system is intended to streamline operations by managing reservations, orders, billing, and table management in an efficient and user-friendly way. It will be a web-based system that provides functionality to both customers (for booking and ordering) and staff (for managing orders, generating bills, and checking availability).

4.6 Functional Requirements

These are the core functionalities that the system must provide:

a. Reservation Management

- **Customer Functionality**:

  Customers can book a table for a specific time and date via an online reservation system.
  The system will show available time slots and update availability in real-time.
  Customers can cancel or modify their reservation.

- **Staff Functionality**:

  Staff can view and manage the reservation list.
  They can modify or cancel reservations if necessary.
  Staff can manually add reservations if customers call to book.

b. Order Management

- **Customer Functionality**:

  Customers (or waitstaff) can place orders from the menu.
  The system will allow customers to select items, modify quantities, and request special instructions (e.g., dietary needs).

- **Staff Functionality**:

  Waitstaff will input orders into the system, which will be relayed to the kitchen in real-time.

Waitstaff can check the status of orders, mark orders as served, and notify the kitchen if any issues arise.

c. Table Management

- **Customer Functionality**:

  The system will automatically assign tables based on reservation time or order size.

- **Staff Functionality**:

  The system will display available tables in real-time, allowing staff to optimize seating arrangements.

  Staff will be able to update table statuses (e.g., reserved, occupied, cleaned).

d. Billing and Payment

- **Customer Functionality:**

  The system will generate an itemized bill, including taxes, tips, and any discounts.

  Customers can choose how to pay (card, cash, digital payment).

- **Staff Functionality:**

  Waitstaff can finalize the bill, apply discounts if necessary, and process payments through integrated POS systems.

4.7 Non-Functional Requirements

a. Performance

The system should be fast and responsive, especially for reservation booking, real-time order updates, and payment processing.

The database should handle concurrent user requests smoothly, as the restaurant may have many active reservations or orders at once.

b. Security

Customer data (including reservation and payment information) must be securely handled and stored.

Secure payment processing should be integrated (e.g., via Stripe or PayPal).

Proper authentication and authorization should be in place for staff to access various parts of the system.

c. Scalability

- The system should be scalable to accommodate future growth (e.g., if the restaurant expands or additional features are added).
- The system must handle increased load during peak hours (e.g., holiday seasons).

d. Usability

The system should be intuitive for both customers and staff.

The user interface should be simple and easy to navigate, with clear instructions and minimal clicks to complete a task.

CHAPTER 5

SYSTEM ANALYSIS

5.1 Introduction of System

System design refers to the process of creating the architecture of an information system. It breaks down the system into smaller, manageable parts and specifies how these components interact with one another. The primary goal of system design is to create a blueprint for the development and implementation of the system that meets both functional and non-functional requirements.

For the **3FE Restaurant Management System**, the design will focus on the following key components:

- **Reservation Management**
- **Order Management**
- **Table Management**
- **Billing and Payment Management**
- **Database Architecture**

The design will ensure that the system is scalable, secure, reliable, and user-friendly.

5.2 System Design

System design involves detailed planning of both the **logical structure** (how the system works) and **physical structure** (hardware, software, and network resources needed for deployment). The design is broken down into several phases:

a. Logical Design

This phase involves planning how the system will function logically, independent of the actual implementation. It defines the flow of data, how modules interact, and the overall behavior of the system.

- **Modules**:
- **Reservation Management**: Handles customer reservations, availability, and confirmation.

- **Order Management**: Processes customer orders, updates the kitchen staff, and tracks order status.
- **Table Management**: Manages table assignments, including availability and seating arrangements.
- **Billing and Payment**: Generates customer bills and handles payment transactions.
- **Reporting and Analytics**: Provides insights into restaurant operations, sales, and customer behavior.
- **Data Flow**:

  Customer reservations and orders flow through the system, interacting with the database for updates on availability and order status.

  Billing data is gathered from orders and passed through payment gateways for processing.

  Tables are managed dynamically as reservations and orders are processed.

b. Architecture Design

This step involves defining the overall architecture of the system, including its components and how they interact. It defines the **system's structure**, including hardware, software, and data management. The architecture ensures the system is efficient and can be scaled easily.

- **Architecture Type**:

  Client-Server Architecture: The system will operate in a client-server setup. The client (user-facing part) will interact with the server, which will handle all the logic and database processing.

  Web-Based System: A web interface will allow customers to make reservations, view the menu, and order. Staff members will use web-based interfaces for order entry, billing, and managing reservations.

  Three-Tier Architecture:

  Presentation Layer: The frontend of the system (HTML, CSS, JavaScript), where customers and staff interact.

  Business Logic Layer: The backend logic (PHP), which processes reservations, orders, and generates bills.

  Data Layer: The database (MySQL), where all data related to reservations, orders, menu items, and billing are stored.

- **Technology Stack**:

  Frontend: HTML, CSS, JavaScript (React or plain JS for interactivity)

  Backend: PHP (for server-side processing, data handling, and communication with the database)

  Database: MySQL (for storing and managing data)

c. Physical Design

Physical design refers to how the system will be implemented in terms of hardware, software, and network resources. It specifies the physical aspects of the system's operation.

- **Deployment**:

  The system will be deployed on a web server (e.g., Apache or Nginx), accessible via web browsers.

  A MySQL server will host the database, either on-premise or through a cloud service.

  The system must support mobile and desktop devices, ensuring responsive design for both customers and staff.

- **Security**:

  SSL certificates to secure data transmission between clients (customers) and servers.

  User authentication and authorization for staff to ensure secure access to different modules (reservations, orders, payments).

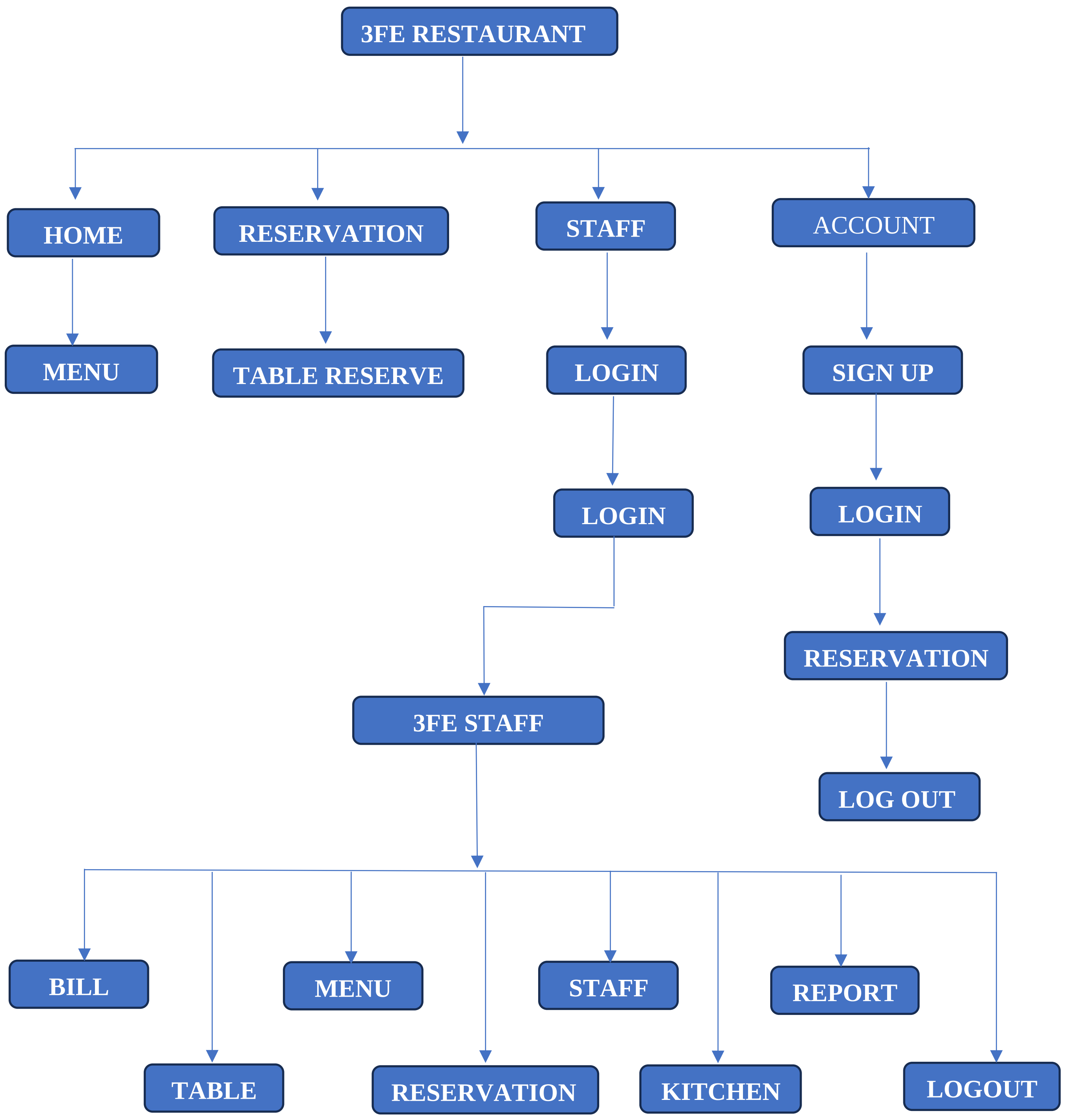
  Secure payment processing via integration with third-party payment gateways (Stripe, PayPal, etc.).

- **Backup and Recovery**:

  Regular database backups should be set up to ensure recovery in case of system failure.

  A disaster recovery plan should be in place for system restoration.

5.3DATA FLOW DIAGRAM



5.4. Data Flow Diagrams (DFD)

Data Flow Diagrams illustrate the flow of data within the system, showing how data is input, processed, stored, and output. They are helpful in understanding how different components of the system interact with each other and how information flows through various processes.

5.4.1 Level 0 DFD (Context Diagram): This is the highest level of abstraction and shows the system as a whole, with external entities and data flows.

- **Entities**:

  Customer: Books reservations, orders food, and makes payments.

  Restaurant Staff: Handles order entry, table management, and billing.

  Payment Gateway: Processes payments from customers.

  Database: Stores reservation data, order details, menu items, and billing information.

- **Processes**:

  Reservation Management: Takes customer reservation details, updates availability, and stores it in the database.

  Order Management: Receives customer orders, updates kitchen staff, and processes the payment.

  Billing & Payment: Generates bills for orders and processes payment through integrated gateways.

Context Diagram Example:

Staff Interface

5.4.2 Level 1 DFD (Expanded Processes): This diagram goes deeper into the processes, showing how individual modules interact.

- **Processes**:

  Reservation System: Manages bookings and communicates with the database.

  Order Management: Receives orders from customers and staff, processes them, and sends order information to the kitchen.

  Billing System: Generates the bill based on the order, applies taxes, discounts, and processes payment.

5.4.3 Level 2 DFD (for specific processes like **Order Management** or **Billing System**) would expand the internal operations further (e.g., order creation, table assignment, payment processing).

CHAPTER 6

IMPLEMENTATION AND TESTING

Implementation and Testing for the 3FE Restaurant Management System

The **Implementation and Testing** phases are crucial steps in the software development lifecycle, where the system is built and its functionality is verified to ensure that it meets the requirements and works as expected. Here's an overview of these phases for the **3FE Restaurant Management System**:

6.1Implementation

Implementation is the phase where the actual system is developed based on the design specifications. This step involves coding the system, setting up the infrastructure, and integrating all modules.

A. Development Process

- **Setup Development Environment**:
  Install necessary tools (IDE, text editor, etc.).
  Set up the backend server environment (e.g., **Apache** or **Nginx**) and database server (**MySQL**).
  Set up the **frontend** environment (HTML, CSS, JavaScript) and integrate frameworks (e.g., **React** or **Vue.js** if used).
- **Database Creation**:
  Create the necessary **tables** in the **MySQL** database (e.g., **Customer**, **Reservation**, **Order**, **Menu**, **Bill**, **Payment**).
  Set up **foreign key relationships** to link tables (e.g., customer ID linking to reservations and orders).

B. Backend Development:

PHP scripts are developed to handle server-side logic, such as:
Processing **reservation requests** and checking table availability.

Managing **order details** (adding, updating, deleting).

Generating **bills** and calculating totals, taxes, and tips.

Interfacing with **payment gateways** (e.g., **Stripe**, **PayPal**) for payment processing.

Implement user **authentication** and **authorization** for staff access (using sessions or tokens).

C. Frontend Development:

Design and build the **user interface** using **HTML**, **CSS**, and **JavaScript**.

For customers: Pages for **reservations**, **menu browsing**, and **order placement**.

For staff: Interfaces for **order entry**, **table management**, and **bill generation**.

Use **AJAX** for seamless communication with the backend (e.g., updating order status without refreshing the page).

- **Integration**:

  Integrate the **frontend** and **backend** so that data can flow seamlessly between the user interface and the database.

  Ensure that the system can handle **real-time updates** for reservations, orders, and payments.

- **Security**:

  Implement security measures such as **SSL encryption** for data transmission.

  Use **parameterized queries** to prevent **SQL injection** attacks.

  Implement **user authentication** (login/logout) for restaurant staff and admins.

  Use **session management** to track active users and prevent unauthorized access.

D. Deployment:

Deploy the system on a web server (e.g., **Apache**).

Ensure that the system is hosted on a reliable platform with proper **backup** and **scalability** capabilities.

Set up a **domain name** and ensure the website is accessible to customers and staff.

6.2 Testing

Testing is essential to verify that the system meets the required specifications, is free from bugs, and functions properly in all use cases. This phase involves both manual and automated testing.

Types of Testing

- **Unit Testing**:

  Unit tests focus on testing individual components or functions of the system.

  Test each function (e.g., adding a reservation, processing an order, generating a bill) independently.

  Use testing frameworks such as **PHP Unit** (for PHP) or **Jest** (for JavaScript).

  Example: - Testing the PHP function that checks table availability during reservation.

Integration Testing:

- Test the **interaction** between different modules or systems.
- Ensure that the frontend can successfully communicate with the backend.
- Verify that the payment gateway processes payments correctly and updates the database.
- Example: Test if the order information entered by staff gets correctly stored in the **Order** table and reflects in the bill generation.

Functional Testing:

- Test each feature of the system to ensure it works according to the requirements.
- For example, test the reservation system by simulating a customer booking a table.
- Ensure that orders can be placed, updated, and served.
- Example: Verify if a customer can reserve a table, view the menu, place an order, and receive a bill.

Usability Testing:

- Ensure the system is user-friendly and intuitive.
- Ask real users (restaurant staff and customers) to interact with the system and provide feedback.
- Focus on whether the user interface is easy to navigate and understand.

- Example: Test if staff can easily generate a bill or if customers can find the reservation form easily.

Performance Testing:

- Test how well the system performs under stress or high load.
- Simulate multiple users accessing the system at the same time to see how it handles peak traffic.
- Check how the system handles concurrent reservations and orders during busy hours.
- Example: Test if the system can handle 100+ simultaneous reservation requests or if order updates are reflected in real-time for staff.

Security Testing:

- Ensure the system is secure and can withstand attacks.
- Test for vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).
- Verify that sensitive customer data (e.g., payment information) is securely stored and processed.
- Example: Test if the login page is protected against brute-force attacks and if payment processing is secure.

System Testing:

- Perform end-to-end testing to verify that the complete system works as expected.
- Simulate the entire restaurant flow, from reservation to order placement, bill generation, and payment.
- Test interactions between the system's various modules, such as Order Management and Payment Processing.

Acceptance Testing:

- The final test before the system goes live, ensuring that it meets all business requirements and stakeholder expectations.

- Test based on real-world use cases and verify if the system performs the intended tasks without issues.
- Example: Test if the system can handle real reservations, real orders, and real payments during a trial run.

b. Example Testing Scenarios

Scenario 1: Reservation Management Test

- **Test Case**: A customer reserves a table for 4 people at 7:00 PM.
- **Expected Outcome**: The system should check if the table is available, confirm the reservation, and store the details in the database.

Scenario 2: Order Management Test

- **Test Case**: Waitstaff enters an order for 2 pizzas and a soda for a table.
- **Expected Outcome**: The system should store the order details, update the kitchen staff, and provide an estimated wait time.

Scenario 3: Payment Processing Test

- **Test Case**: A customer pays the bill with a credit card through an integrated payment gateway (e.g., Stripe).
- **Expected Outcome**: The payment should be processed successfully, the bill should be marked as paid, and the database should update the payment status.

Scenario 4: Table Management Test

- **Test Case:** A customer arrives for their reservation and is seated at the appropriate table.
- **Expected Outcome:** The system should mark the table as occupied and ensure that no other reservations are made for that table.

CHAPTER 7

SNAPSHOTS OF THE PROJECT

7.1 STARTING INTERFACE

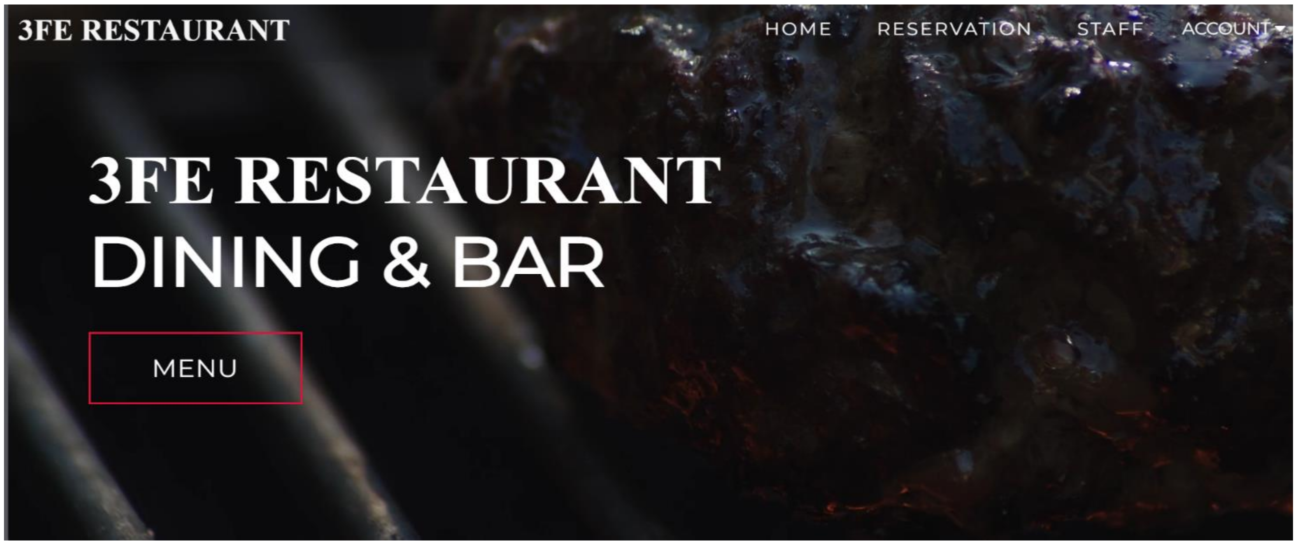


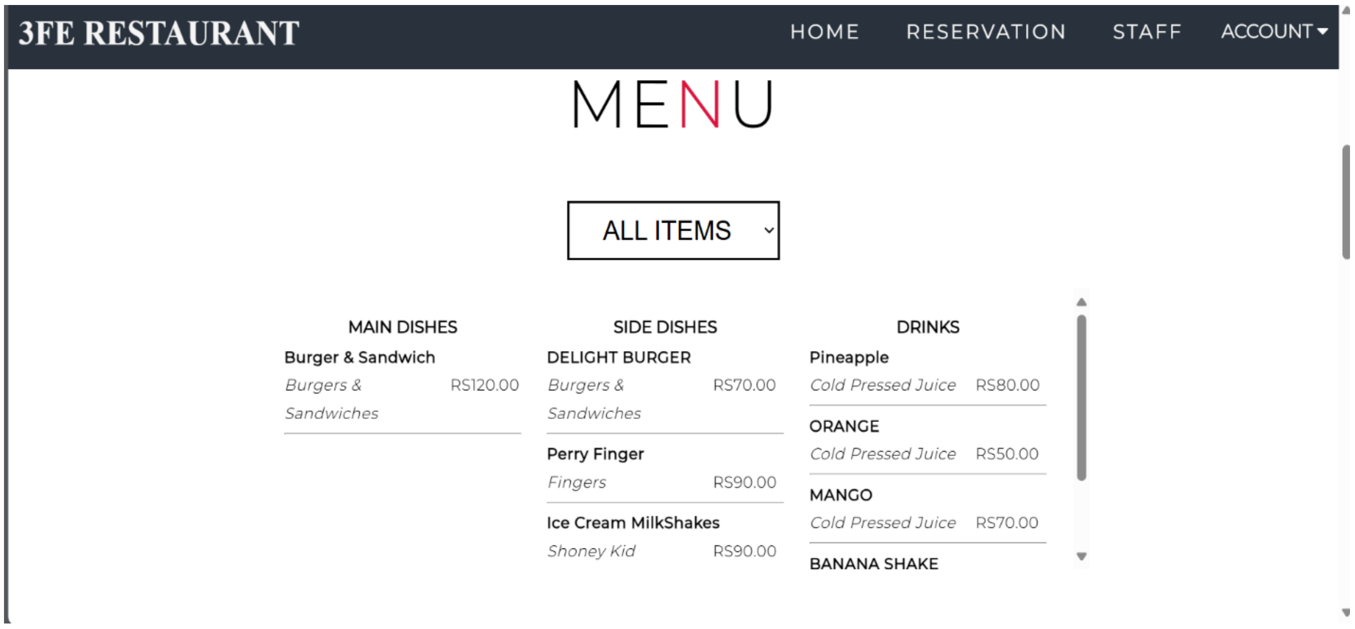
Fig 7.1: - This is the starting interface of our project and we can select the options that are shown in navbar

Homepage Design Features

- Brand Title**: “3FE RESTAURANT” (top-left)**
- Navigation Menu**: Home, Reservation, Staff, Account (top-right)**
- Hero Section**:**

  - ❖ **Bold title: “3FE RESTAURANT DINING & BAR”**
  - ❖ **Prominent** "Menu" **button (outlined with red border**
  - ❖ **Background image (appears to be grilled meat or food item)**

7.2MENU BAR



MAIN DISHES

Burger & Sandwich
Burgers & Sandwiches RS120.00

SIDE DISHES

DELIGHT BURGER
Burgers & Sandwiches RS70.00

Perry Finger
Fingers RS90.00

Ice Cream MilkShakes
Shoney Kid RS90.00

DRINKS

Pineapple
Cold Pressed Juice RS80.00

ORANGE
Cold Pressed Juice RS50.00

MANGO
Cold Pressed Juice RS70.00

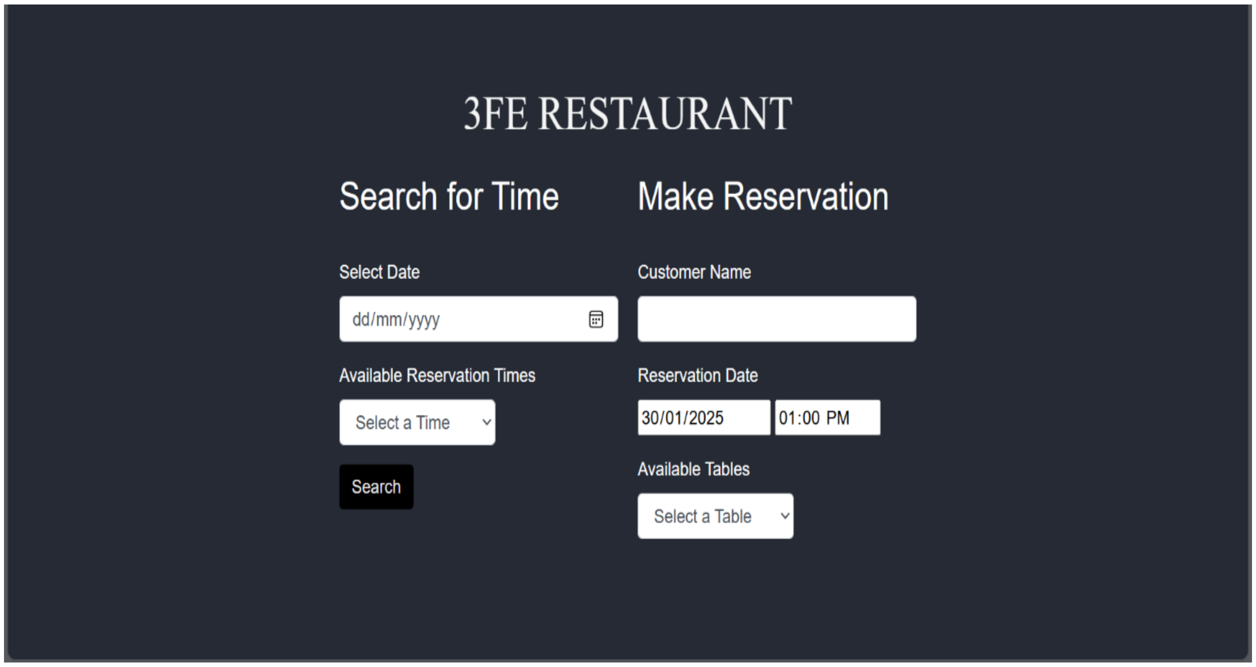
BANANA SHAKE

Fig 7.2: - This is the menu bar of our project and we can select the options that are shown in all items

MENU BAR Overview (Visual Explanation)

- **Top Bar Color:** A dark navy or charcoal background gives it a premium and elegant feel.
- **Logo:** On the far left, you have the **"3FE RESTAURANT"** branding in a bold serif font, emphasizing the restaurant's identity.
- **Navigation Links:** On the right side, links like:
  - **HOME**
  - **RESERVATION**
  - **STAFF**
  - **ACCOUNT** (with a dropdown indicated by an arrow)

7.3 Reservation


3FE RESTAURANT

Search for Time

Select Date

dd/mm/yyyy

Available Reservation Times

Select a Time

Search

Make Reservation

Customer Name

Reservation Date

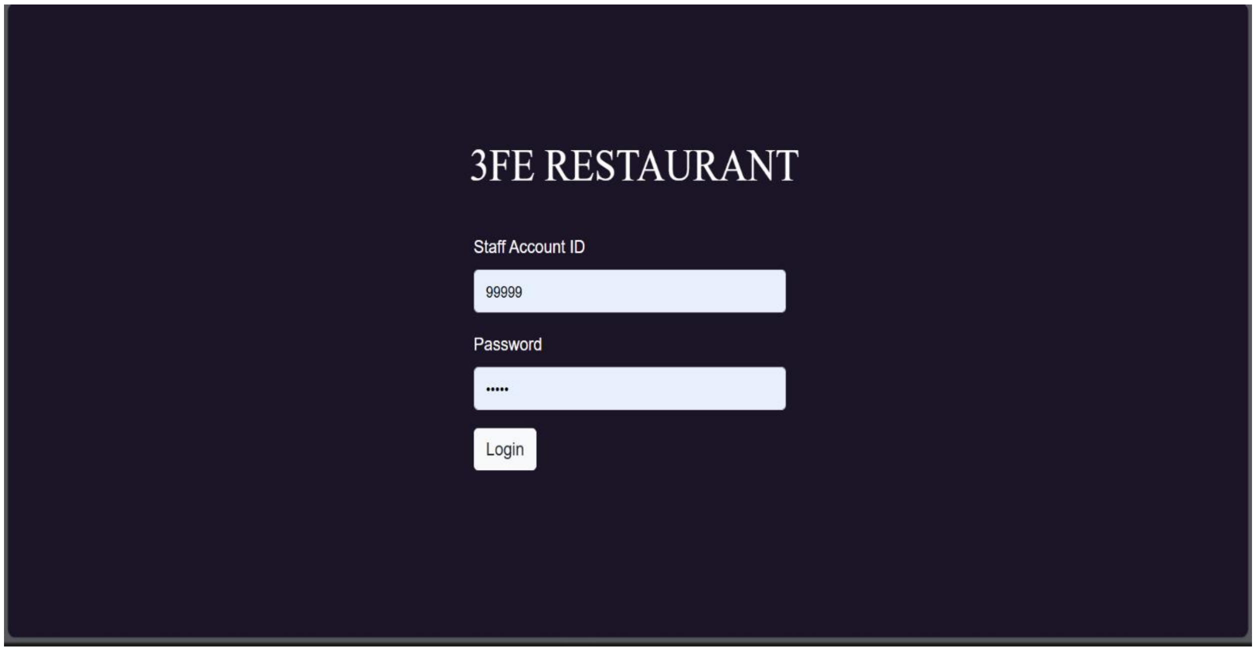
30/01/2025 01:00 PM

Available Tables

Select a Table

Fig 7.3: - This is the reservation interface of our project and we can select the options that select the table, according to time and date.

7.4 Staff Login


3FE RESTAURANT

Staff Account ID

99999

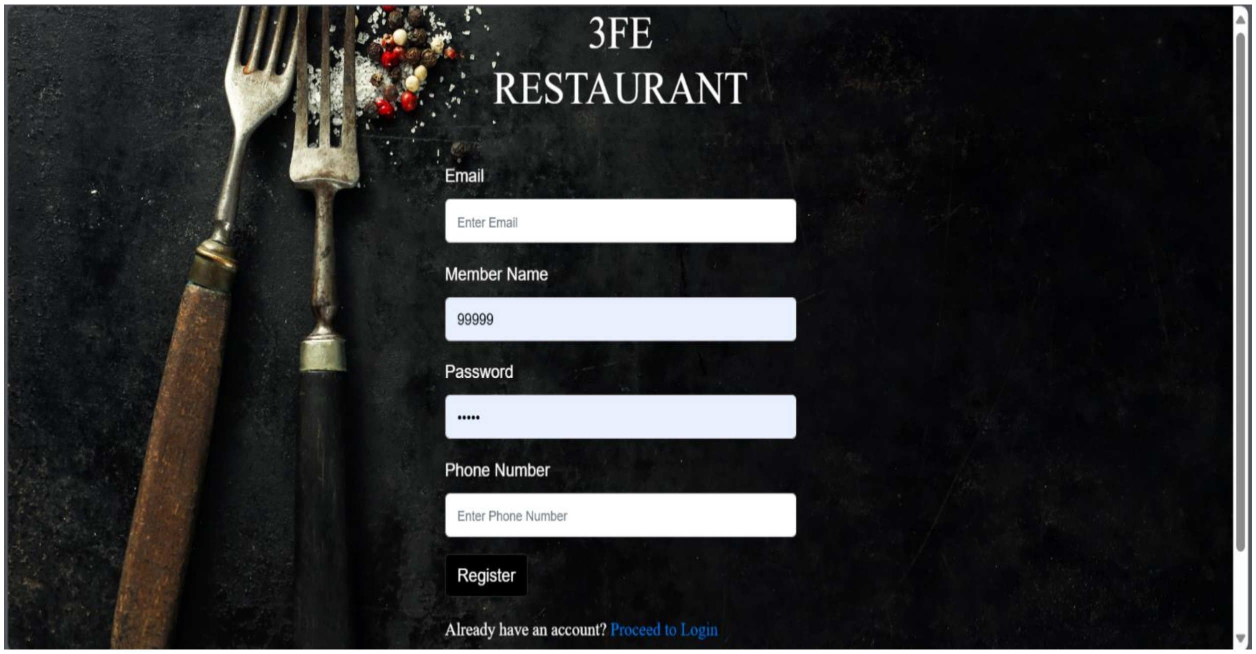
Password

•••••

Login

Fig 7.4: - This fig shown the staff login interface

7.5 Customer Registration


3FE RESTAURANT

Email: Enter Email

Member Name: 99999

Password: •••••

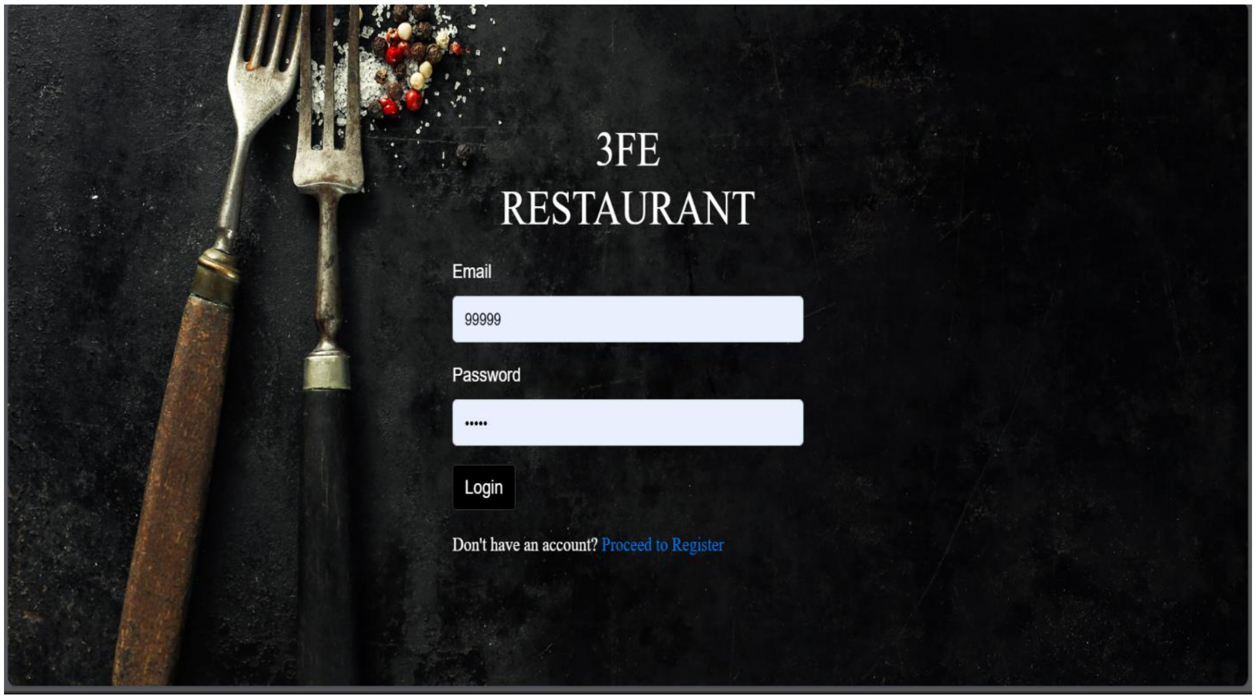
Phone Number: Enter Phone Number

Register

Already have an account? Proceed to Login

Fig 7.5: - This fig shown the registartion of customer with gmail, Name, Password & Contact

7.6 Customer login


3FE RESTAURANT

Email: 99999

Password: •••••

Login

Don't have an account? Proceed to Register

Fig 7.6: - This fig shown the login interface of customer with gmail and Password

7.7 3FE STAFF PANEL

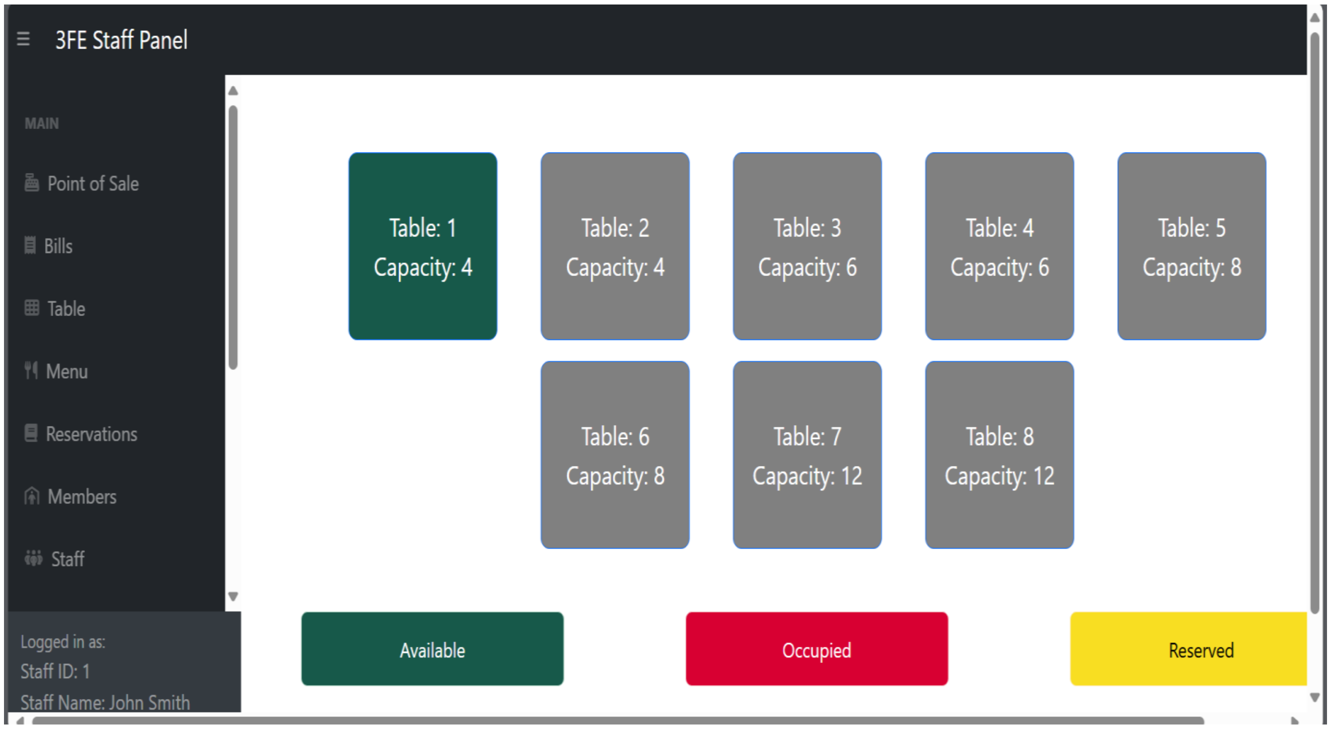


Fig 7.7: - This Fig Shown the 3FE STAFF PANEL, shown the table bills, menu, etc.

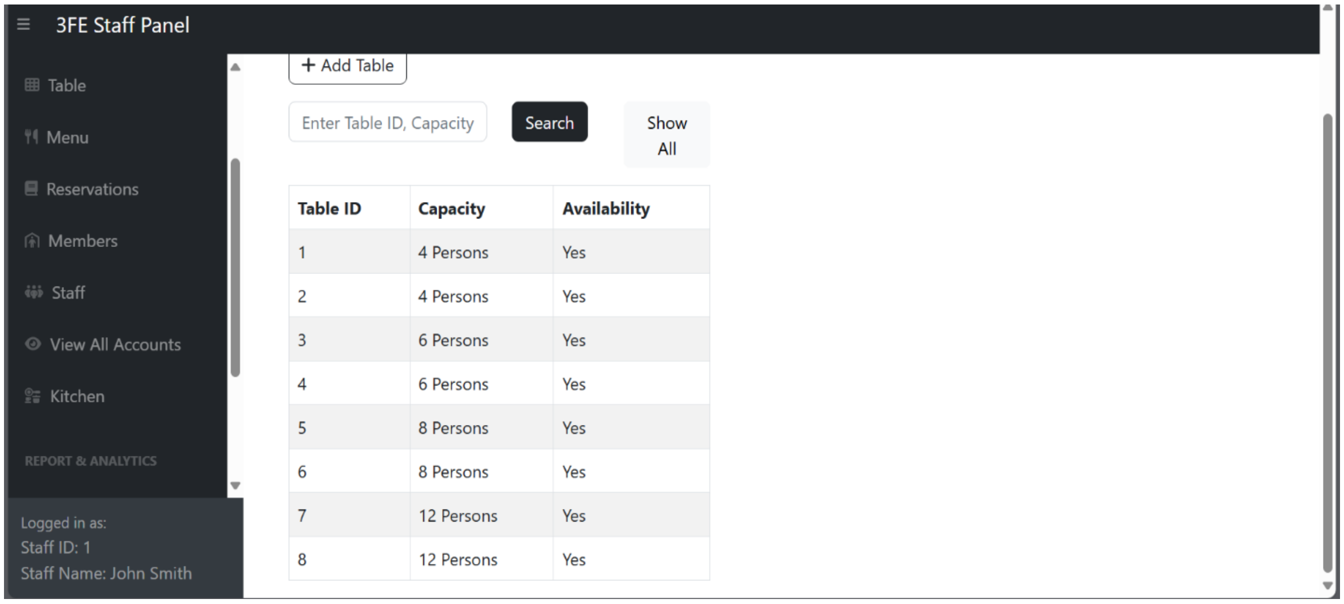
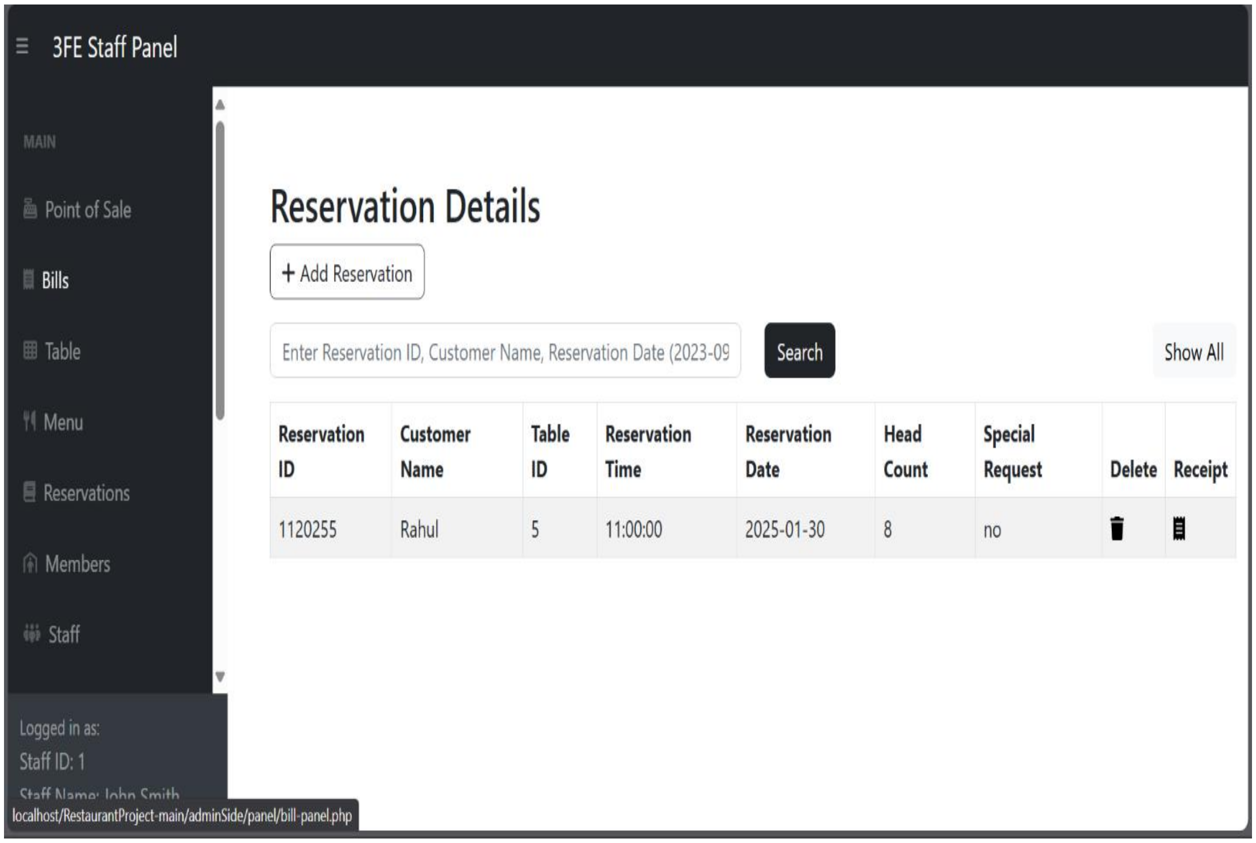


Table ID	Capacity	Availability
1	4 Persons	Yes
2	4 Persons	Yes
3	6 Persons	Yes
4	6 Persons	Yes
5	8 Persons	Yes
6	8 Persons	Yes
7	12 Persons	Yes
8	12 Persons	Yes

Fig 7.8: - This Fig Shown the Availablity of the Table

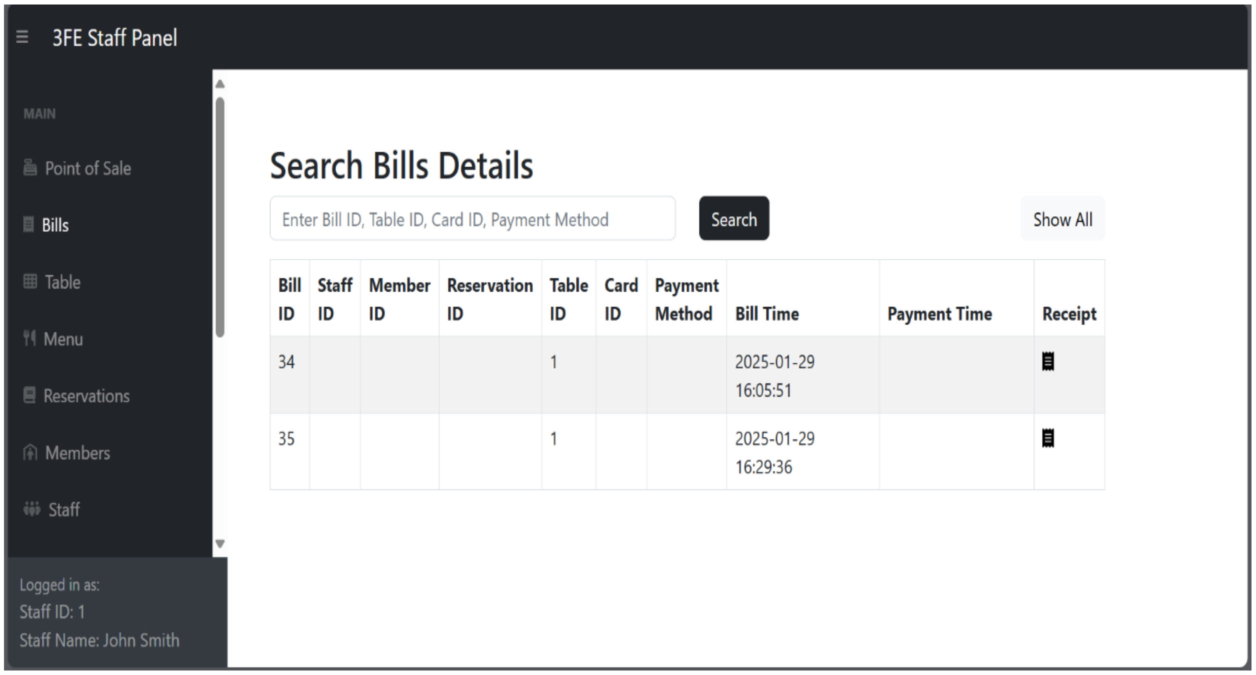


Logged in as:
Staff ID: 1
Staff Name: John Smith
localhost/RestaurantProject-main/adminSide/panel/bill-panel.php

Reservation Details

Reservation ID	Customer Name	Table ID	Reservation Time	Reservation Date	Head Count	Special Request	Delete	Receipt
1120255	Rahul	5	11:00:00	2025-01-30	8	no	🗑	🧾

Fig 7.9:- This fig shown the Reservation details of the Restaurant Table.

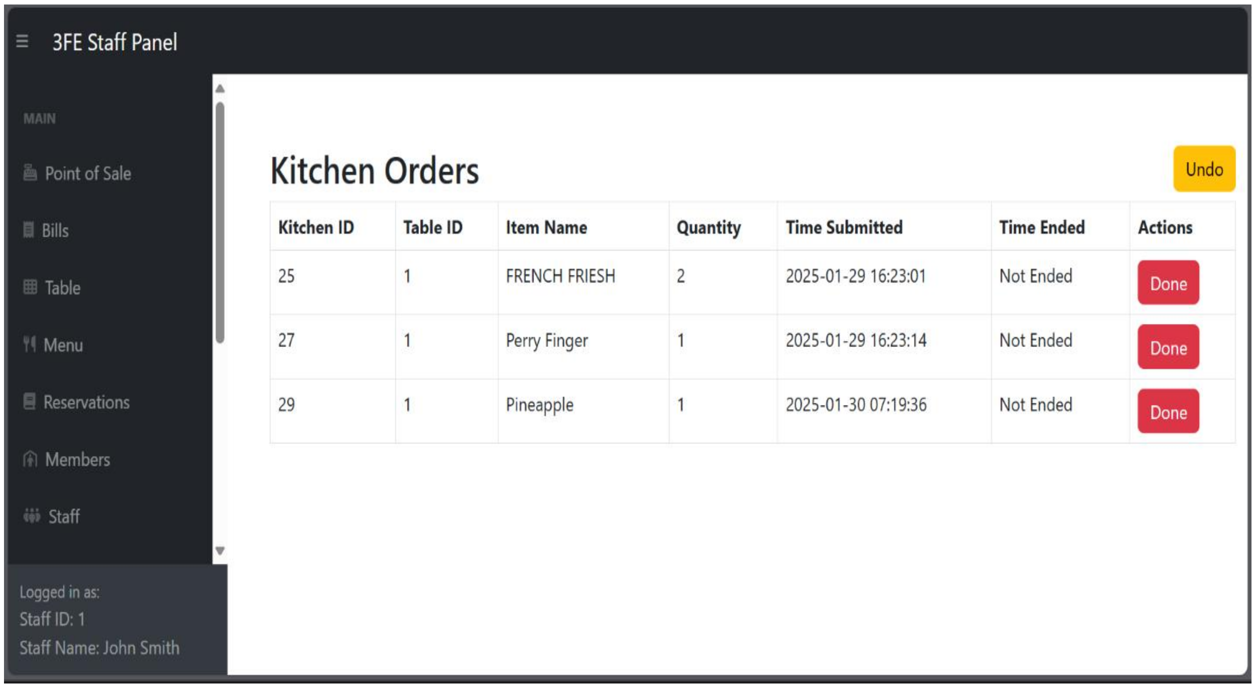


Logged in as:
Staff ID: 1
Staff Name: John Smith

Search Bills Details

Bill ID	Staff ID	Member ID	Reservation ID	Table ID	Card ID	Payment Method	Bill Time	Payment Time	Receipt
34				1			2025-01-29 16:05:51		🧾
35				1			2025-01-29 16:29:36		🧾

Fig 7.10: - This fig shown the Bills detail of customer. Show the transaction card or cash.

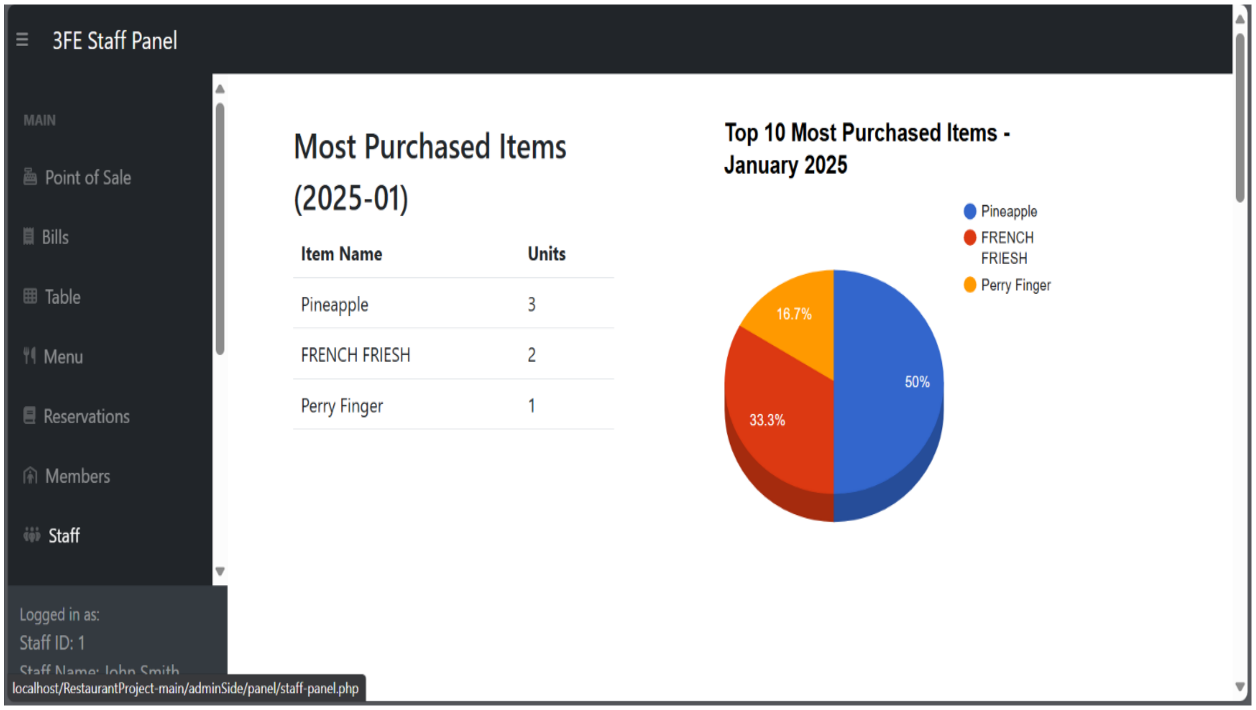


Logged in as:
Staff ID: 1
Staff Name: John Smith

Kitchen Orders

Kitchen ID	Table ID	Item Name	Quantity	Time Submitted	Time Ended	Actions
25	1	FRENCH FRIESH	2	2025-01-29 16:23:01	Not Ended	Done
27	1	Perry Finger	1	2025-01-29 16:23:14	Not Ended	Done
29	1	Pineapple	1	2025-01-30 07:19:36	Not Ended	Done

Fig 7.11 This fig shown the detail of order in Kitchen Chef.



Logged in as:
Staff ID: 1
Staff Name: John Smith

Most Purchased Items (2025-01)

Item Name	Units
Pineapple	3
FRENCH FRIESH	2
Perry Finger	1

Fig 7.12 This fig shown the Annual or Static details of the Restaurant.

CHAPTER 8
Security Considerations

Security is a critical component of any modern web application, especially those handling sensitive user information such as login credentials, contact details, and online reservations. For the 3FE restaurant website—built using PHP, HTML, CSS, JavaScript, and MySQL—ensuring robust security is essential to protect customer data, maintain system integrity, and uphold the restaurant's reputation.

This section outlines the key security practices and considerations implemented in the development of the site. These measures are designed to defend against common web threats such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), brute force attacks, and unauthorized access. By following industry-standard best practices and applying a defense-in-depth approach, the website aims to provide a safe and trustworthy experience for users, staff, and administrators alike.

The subsections that follow address specific areas of concern, including authentication, password protection, input validation, media handling, cookie management, dependency security, and ongoing monitoring.

8.1Authentication and Authorization

- Implement a **secure login system** using sessions and hashed credentials.
- Use **role-based access control (RBAC)** to restrict areas like admin dashboards, menu management, and reservation systems.
- Ensure **session tokens** are securely generated (e.g., session_regenerate_id ()) and stored with proper expiration.
- Logout functionality should invalidate sessions properly.

8.2Password Protection

- Hash all passwords with strong algorithms like **bcrypt** (e.g., password_hash () in PHP).

- Use **password policies**: enforce minimum length, complexity, and expiration if needed.
- Store **salts** implicitly (handled by bcrypt in PHP).
- Implement a **password reset** system using email with time-bound tokens.

8.3 Input Validation and Sanitization

- Validate all user inputs using **server-side** checks in PHP and **client-side** with JavaScript for UX.
- Use **prepared statements** with PDO or MySQLi to avoid SQL injection:
- Sanitize outputs using htmlspecialchars() to prevent **XSS**.
- Restrict file uploads to specific types and validate file contents (e.g., MIME type checking).

8.4 Rate Limiting and Brute Force Protection

- Implement **rate limiting** on login attempts (e.g., max 5 per IP per 10 minutes).
- Use **CAPTCHA** after several failed login attempts.
- Temporarily block or delay response after repeated failures.
- Store and monitor IP addresses for suspicious activity.

8.5 Secure Media Handling

- Store uploaded files **outside the web root** and serve them via PHP after authentication.
- Validate media file type, size, and use unique filenames to avoid overwriting.
- Sanitize file names and use **Content-Disposition** headers to force download if needed.
- Prevent execution of uploaded files using .htaccess or server configuration.

8.6 CORS and Cookie Management

- Set Same Site=Strict or Lax and HTTP Only and Secure flags on cookies.
- Limit CORS to trusted domains using the Access-Control-Allow-Origin header.

- Avoid exposing sensitive data through client-side accessible cookies or local Storage.
- Use CSRF tokens for state-changing actions (form submissions, deletes, etc.).

8.7 Dependency and Environment Security

- Keep PHP, MySQL, and all libraries up to date.
- Use tools like **Composer** for dependency management and avoid outdated or untrusted packages.
- Secure.env or config files (e.g., database credentials) and never expose them via version control.
- Disable directory listing and unnecessary modules in the server (e.g., Apache/Nginx).

8.8 Logging and Monitoring

- Log important actions (logins, failed attempts, admin changes) with timestamps and IPs.
- Do **not log sensitive data** like passwords or full card details.
- Store logs securely and set access controls.
- Use monitoring tools or scripts to detect anomalies or abuse (e.g., sudden spike in traffic or login attempts).

CHAPTER 9

CONCULATION

The management of 3FE Restaurant places a strong emphasis on both security and customer satisfaction, ensuring a safe and enjoyable experience for all guests. By implementing rigorous security protocols, including staff training, surveillance, and adherence to health and safety regulations, 3FE effectively safeguards both patrons and employees. This commitment to security provides customers with peace of mind while dining.

On the satisfaction front, the restaurant's management consistently prioritizes customer experience. Attention to detail in service quality, staff responsiveness, and food excellence ensures that each guest feels valued and well-cared for. Customer feedback is actively sought and integrated into the restaurant’s operations, allowing 3FE to continuously improve and tailor its offerings to meet evolving customer expectations.

In conclusion, the strong management of 3FE Restaurant not only ensures security through comprehensive safety measures but also creates a welcoming, customer-centered environment that fosters satisfaction and loyalty. This balanced approach contributes to the restaurant’s ongoing success and reputation.

REFERENCES

- www.google.com
- www.udemy.com
- www.internshala.com
- www.python.org
- www.geeksforgeeks.org
- www.github.com
- www.chatgpt.com
- www.javatpoint.com
- www.wikipedia.com