

# Introduction

A restaurant management system is a software application that helps streamline and automate various processes within a restaurant, such as order management, inventory tracking, table reservations, employee scheduling, and customer management. we are could consider designing and developing a comprehensive system that integrates all these functionalities to improve the overall efficiency and effectiveness of restaurant operations. This project could involve creating user-friendly interfaces for both restaurant staff and customers, implementing secure payment processing systems, and generating detailed reports for performance analysis. By focusing on enhancing the overall dining experience and operational efficiency, your project could make a significant impact on the restaurant industry.

Running a restaurant is hectic enough as it is, so why not make the day- to-day processes easier by having a system that will help ease the workload for you? There are so many day-to-day processes that restaurants have to deal with. These can range from scheduling in employees, managing HR, to preparing for payroll and to keep record of transactions and database.

# Existing System

In The existing restaurant management process is primarily manual or semi-digital, involving separate systems or physical records to handle reservations, menu updates, billing, staff management, and kitchen orders. In many traditional setups, restaurant staff rely on handwritten logs, printed order tickets, and spreadsheets to track daily operations. This often leads to inefficient communication between the kitchen and service staff, delayed updates, misplaced orders, and difficulty in tracking customer preferences or generating accurate reports.

Some restaurants may use basic digital systems, but these are often fragmented, where different software tools handle billing, reservations, and staff accounts independently. This lack of integration results in duplicate data entry, higher chances of error, and poor synchronization of real-time operations. Additionally, security is often overlooked, and there is little to no automation or data analytics to help with business decisions.

The proposed system aims to overcome these limitations by offering a centralized and structured solution that integrates all core restaurant operations into a single platform. This includes managing staff accounts, reservations, menu items, and customer interactions—all from a web-based interface accessible via local server setup.

# Need and Scope

## Need

1. **Order Management:** Managing orders manually can lead to errors, delays, and inefficiencies. A restaurant management system can automate the order-taking process, track orders in real-time, and ensure accurate and timely delivery of food to customers.
2. **Inventory Tracking:** Keeping track of inventory levels, ingredient costs, and supplier information can be a daunting task for restaurant owners. A management system can help automate inventory management, track ingredient usage, and generate alerts for low stock levels.
3. **Table Reservations:** Managing table reservations manually can lead to overbooking, long wait times, and customer dissatisfaction. A restaurant management system can streamline the reservation process, optimize table allocation, and improve the overall dining experience for customers.
4. **Employee Scheduling:** Creating and managing employee schedules manually can be time-consuming and prone to errors. A management system can help automate the scheduling process, track employee availability, and ensure proper staffing levels during peak hours.
5. **Customer Management:** Building and maintaining customer relationships is crucial for the success of a restaurant. A management system can help store customer information, track preferences, and send personalized promotions to enhance customer loyalty.

## **Scope**

1. Order Management: The system should allow for easy and efficient order taking, processing, and tracking, including options for dine-in, takeout, and delivery orders.
2. Inventory Management: The system should enable real-time tracking of inventory levels, ingredient costs, and supplier information to optimize stock levels and reduce waste.
3. Table Reservations: The system should provide a user-friendly interface for managing table reservations, optimizing table allocation, and reducing wait times for customers.
4. Employee Scheduling: The system should facilitate the creation and management of employee schedules, track employee availability, and ensure proper staffing levels during peak hours.
5. Menu Management: The system should allow for easy customization and updating of menus, including pricing, descriptions, and availability of items.
6. Reporting and Analytics: The system should provide comprehensive reporting and analytics capabilities to track key performance indicators, analyze sales trends, and make data-driven decisions.
7. Customer Relationship Management: The system should enable the storage of customer information, track customer preferences, and facilitate personalized promotions to enhance customer loyalty.

# Proposed System

1. **Order Management System:** The system would include a user-friendly interface for taking and processing orders, including options for dine-in, takeout, and delivery orders. It would also allow for order customization, special requests, and order tracking.
2. **Inventory Management System:** The system would enable real-time tracking of inventory levels, ingredient costs, and supplier information. It would provide alerts for low stock levels, automate reordering processes, and help optimize inventory levels to reduce waste and control costs.
3. **Table Reservation System:** The system would include a table reservation feature to manage reservations, optimize table allocation, and reduce wait times for customers. It would provide a visual layout of the restaurant floor and allow for easy reservation management.
4. **Employee Scheduling System:** The system would facilitate the creation and management of employee schedules, track employee availability, and ensure proper staffing levels during peak hours. It would also provide notifications for shift changes and time-off requests.
5. **Menu Management System:** The system would allow for easy customization and updating of menus, including pricing, descriptions, and availability of items. It would enable menu item categorization, specials management, and seasonal menu changes.
6. **Reporting and Analytics System:** The system would provide comprehensive reporting and analytics capabilities to track key performance indicators, analyze sales trends, and make data-driven decisions. It would include customizable reports, dashboards, and insights to help restaurant owners monitor and improve business performance.

# Objective

- To develop a centralized web-based system for managing restaurant operations efficiently.
- To streamline staff management, including account creation, login, and role assignment.
- To simplify table reservations and customer handling through a structured interface.
- To allow administrators to manage menu items, descriptions, and pricing with ease.
- To replace manual processes with a digital system that reduces human error and increases accuracy.
- To improve internal workflow and coordination between admin and staff sections.
- To ensure secure authentication and access control for staff members.
- To provide a user-friendly interface for easy navigation and operation

# Requirement Gathering

**Identify Stakeholders:** Determine who will be involved in the project, including restaurant owners, managers, chefs, waitstaff, customers, and possibly suppliers

Requirement gathering is the process of generating a list of functional, technical and systematic requirements from several project stakeholders, such as clients, product users or vendors. This list may likely include features, activities and tasks for a team to execute in order to achieve the goals of a project. Usually, there are two types of requirements to consider: functional and non-functional. Functional requirements include the information, interactions and processes that a client requests. Non-functional requirements are other technical and operational aspects of a project.

Consider starting the process of requirements gathering at the beginning of a project to help ensure effective planning and management. Here are some aspects to consider as you start requirements gathering:

**Timeline:** This entails estimating the overall length and schedule of a project. This can help you effectively plan for its requirements over the course of an entire project, ensuring that you're Consider who you want to include in a project and what their various roles might be. Try to determine their prepared throughout its duration.

**People:** individual requirements, as well as how they might collaborate with the entire team. **Goals:** Identifying primary goals early can help you determine a project's requirements and ensure that the project specifically focuses on executing these objectives.

# Fact Finding Techniques

## **Fact Finding Techniques**

In Order to gather an analyzed the relevant information following methods or adopted.

### **Observation**

Find hand information about various activities carried out can be studied through observation. This method used to solve how documents were handled, how processes were carried out and whether those steps were actually followed Thus the current scenario was understood using observation.

### **Record Review**

The blood of records maintained to give clear idea as to how actual process carried also management working on whole.

### **Document Searching**

Various documents which are handled these used formats where studied relation with system, information analyzed



# Requirement analysis

Requirement analysis is a software engineering task that bridges the gap between system level requirements engineering & software design. Requirement engineering activities results in the specification of software's operational characteristics include software's interface with other system elements, and establish constraints that software must meet. Requirement analysis provides the software designer with representation of information, function, & behavior that can be translated to data, architectural, interface & component-level designs. Finally the requirement specification provides the developer & the customer with the means to access quality ones software.

**Software requirement analysis may be divided into five area of efforts:**

- Problem recognition
- Evaluation & synthesis
- Modelling
- Specification
- Review

Initially, the analyst studies the systems specification & the software project plan. It is important to understand software in a system context & to review the software scope that was used to generate planning estimates. Next communication for analysis must be established so that the problem recognition is ensured. The goal is recognition of the basic problem elements as perceived by customers or users.

Problem evaluation & solution synthesis is the next major area of effort for analysis. The analyst must all externally observable data objects, evaluate the flow & content of information, define & elaborate all software functions, understand software behavior in the context of events that affects the system, establish system interface characteristics, & uncover additional design constraints. Each of these tasks serves to desirable the problem so that an overall approach or solution may be synthesized

# Feasibility Study

## **Feasibility Study:**

The feasibility study evaluates whether the proposed restaurant management system is practical and beneficial for implementation in a real-world environment. It covers various aspects such as technical, operational, and economic feasibility to ensure the success and sustainability of the system.

## **Technical Feasibility:**

The project is technically feasible as it uses well-supported and widely available technologies like PHP, MySQL, HTML, and CSS. These technologies are easy to deploy on local servers (e.g., XAMPP), and require no specialized hardware. The system design is modular, allowing easy updates and maintenance.

## **Operational Feasibility:**

The system is designed to be user-friendly for both admin and staff users, reducing the learning curve and allowing quick adoption. By replacing manual record-keeping with digital modules for reservations, accounts, and menu management, it significantly improves efficiency and accuracy in daily operations.

## **Economic Feasibility:**

The cost of implementation is minimal as it relies on open-source tools. There is no need for expensive hardware or software licenses. Additionally, the system reduces operational costs in the long term by minimizing manual errors, improving resource management, and enhancing service delivery.

# Platform Selection

## Front End Tool [PHP]:

We have used PHP as a front-end tool. It provides an understanding ability to code. PHP is the fastest and easiest way to create code for web applications. PHP provides us with a complete set of tools to simplify rapid web application development.

PHP is defined as a programming language. With PHP we can design programs, also known to accomplish some specific tasks. Computer programs generally fall into very broad categories packaged & custom.

## PHP offers following facilities

- Dynamic Web Page Generation: PHP enables the creation of dynamic web pages by embedding PHP code within HTML. This allows developers to generate content dynamically based on user input, database queries, or other external factors
- Easy Syntax for coding.
- Fast and Safe Refactoring.

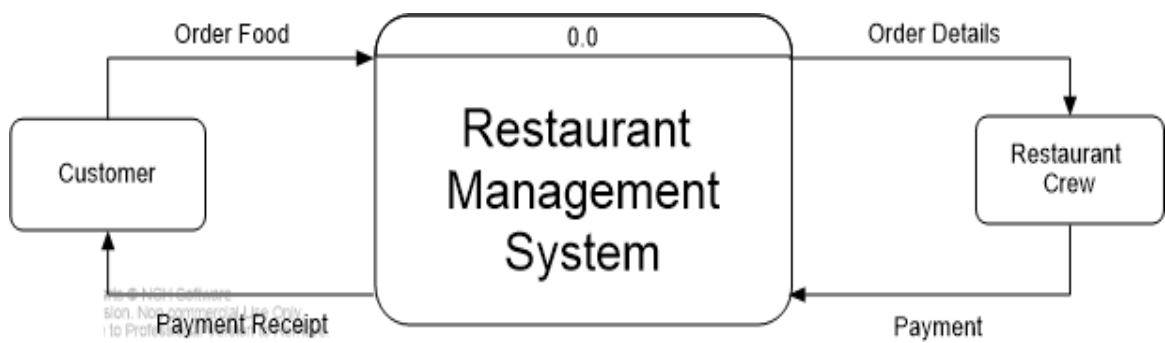
## Back End Tool [MySQL]

Whenever we think of backend, the 'database' is the first word that strikes our mind. It acts as a key component in almost all domains. The main aspects of having a database are storing and retrieving data.

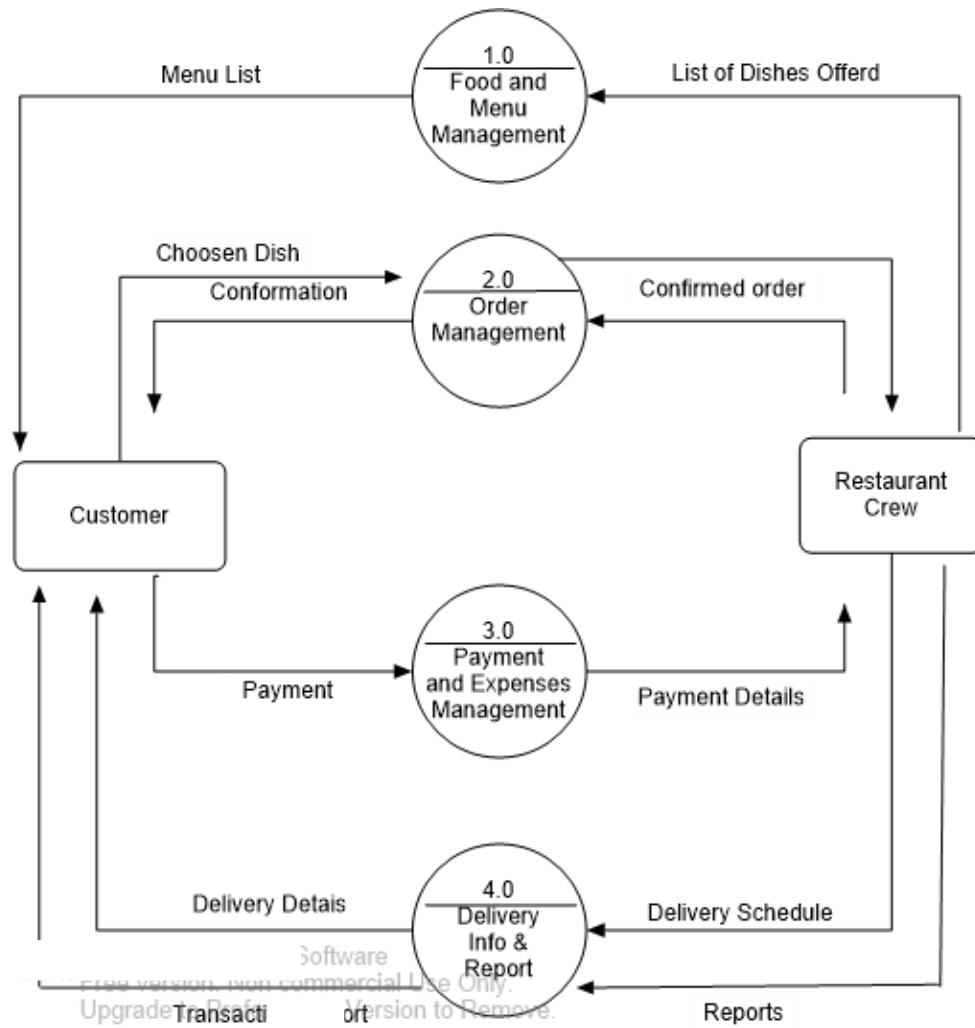
So here we are going to create a MySQL database and then retrieve the necessary data using SQL queries in MySQL

# DFD

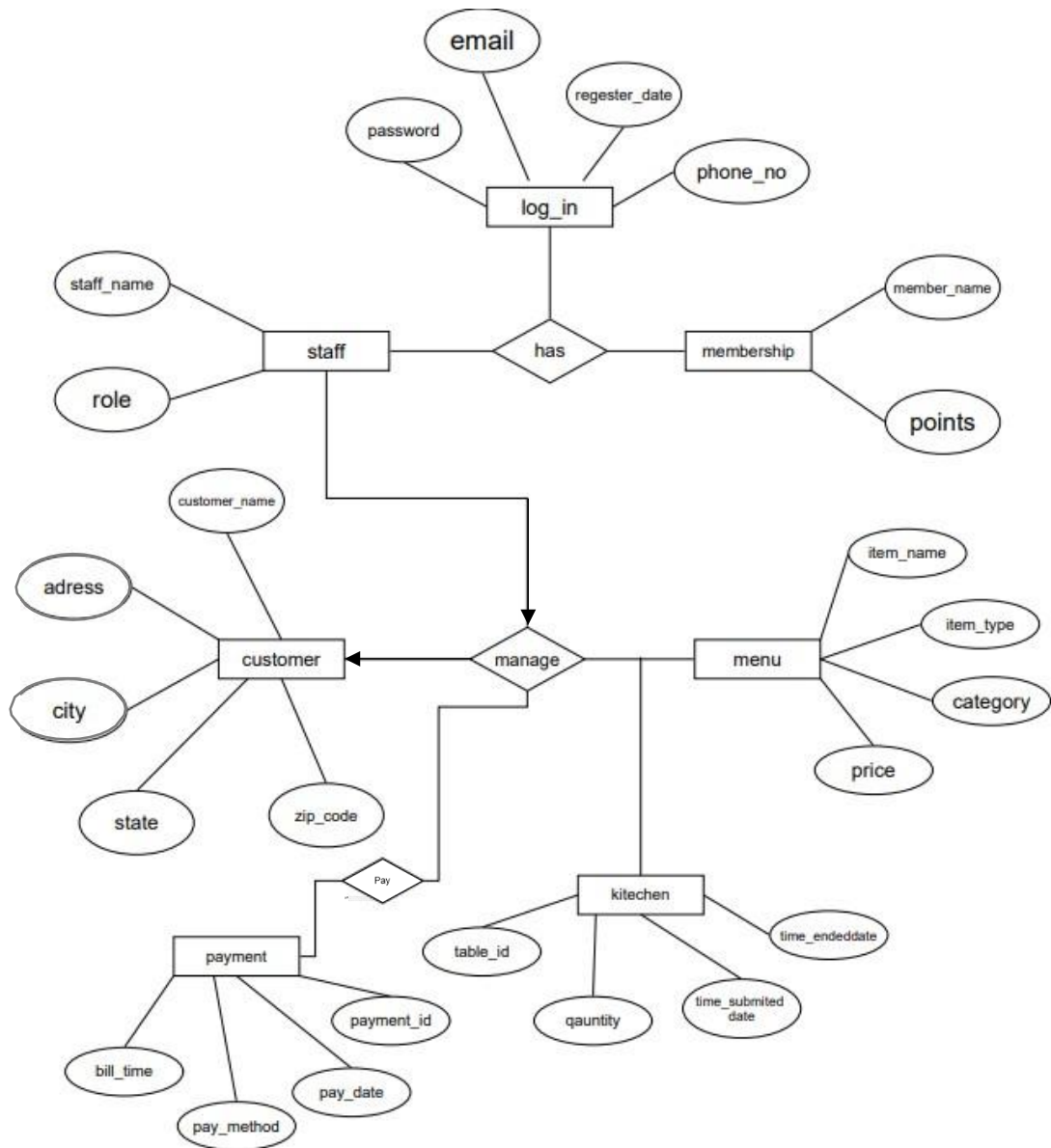
## 0 level



# 1 level



# ERD



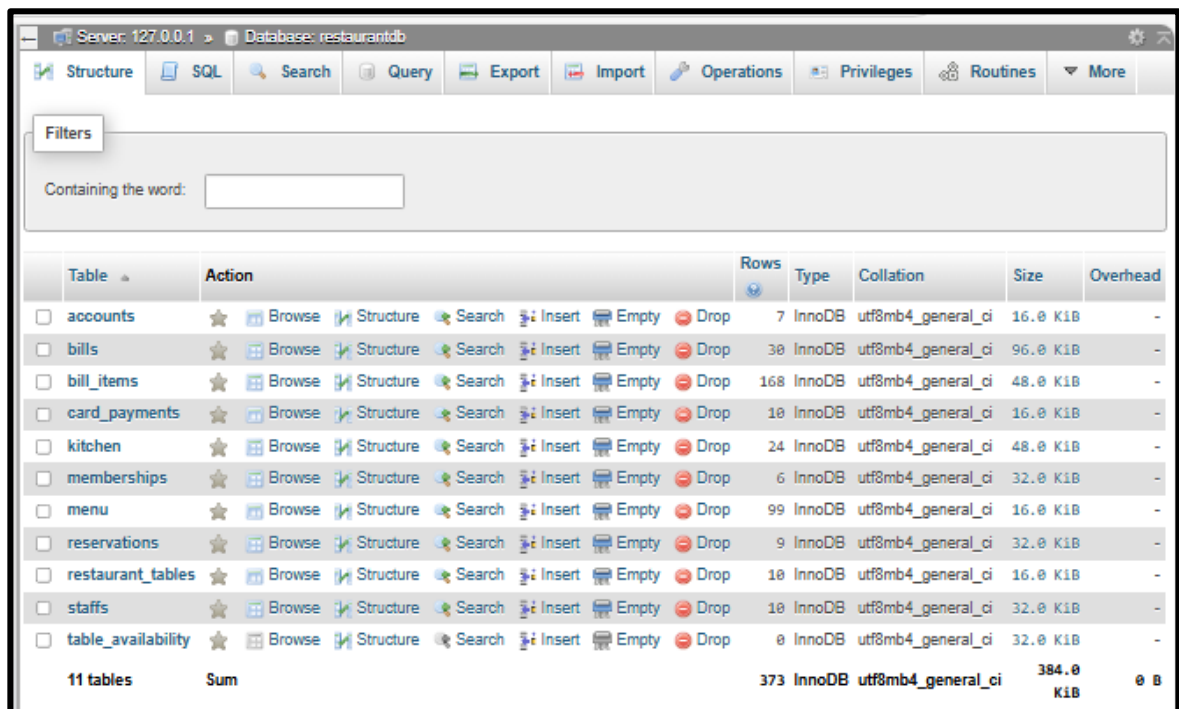
## **Hardware Requirements :**

- Processor : Intel Core i3 or equivalent
- RAM : 4 GB or higher
- Hard Disk: 500 Mb of Free space
- Monitor: 1024x768 resolution or higher
- Keyboard & Mouse: Standard input devices

## **Software Requirements :**

- Operating System : Windows 7/10/11 or Linux
- Web Server : Apache(via XAMPP)
- Database Server: MYSQL(included in XAMPP)
- Programming Language: PHP 7.0 or higher
- Web browser : Google chrome or any
- Code editor : Visual Studio Code

# DATABASE DESIGN



Server: 127.0.0.1 Database: restaurantdb

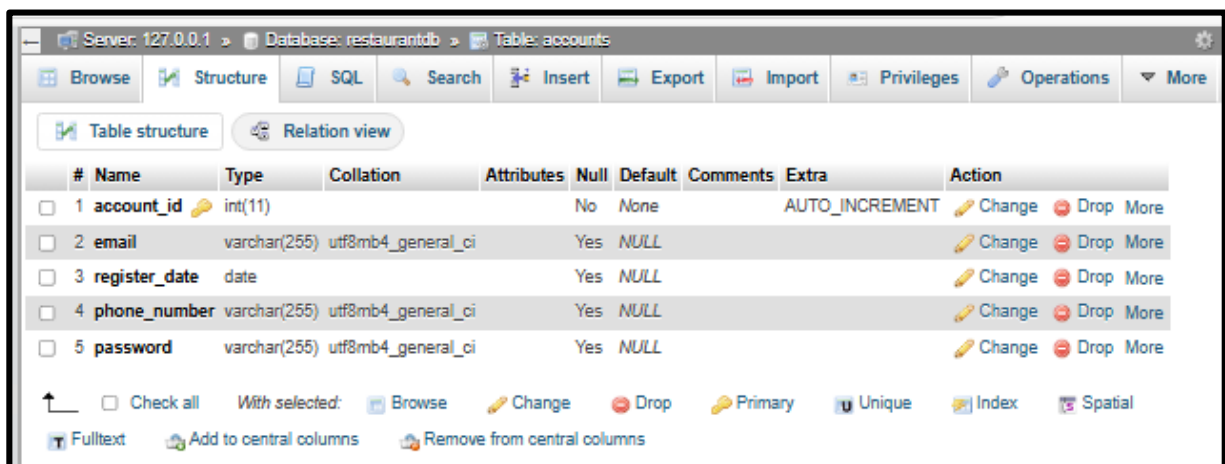
Structure SQL Search Query Export Import Operations Privileges Routines More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> accounts	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> bills	★ Browse Structure Search Insert Empty Drop	30	InnoDB	utf8mb4_general_ci	96.0 KIB	-
<input type="checkbox"/> bill_items	★ Browse Structure Search Insert Empty Drop	168	InnoDB	utf8mb4_general_ci	48.0 KIB	-
<input type="checkbox"/> card_payments	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> kitchen	★ Browse Structure Search Insert Empty Drop	24	InnoDB	utf8mb4_general_ci	48.0 KIB	-
<input type="checkbox"/> memberships	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	32.0 KIB	-
<input type="checkbox"/> menu	★ Browse Structure Search Insert Empty Drop	99	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> reservations	★ Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	32.0 KIB	-
<input type="checkbox"/> restaurant_tables	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KIB	-
<input type="checkbox"/> staffs	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 KIB	-
<input type="checkbox"/> table_availability	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KIB	-
11 tables	Sum	373	InnoDB	utf8mb4_general_ci	384.0 KIB	0 B

## ACCOUNTS



Server: 127.0.0.1 Database: restaurantdb Table: accounts

Browse Structure SQL Search Insert Export Import Privileges Operations More

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	account_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	email	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	register_date	date			Yes	NULL			Change Drop More
<input type="checkbox"/> 4	phone_number	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 5	password	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

↑ ☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial

Fulltext Add to central columns Remove from central columns



# BILLS

Server: 127.0.0.1 > Database: restaurantdb > Table: bills

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	bill_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	staff_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	member_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 4	reservation_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 5	table_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 6	card_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 7	payment_method	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 8	bill_time	datetime			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 9	payment_time	datetime			Yes	NULL			Change  Drop  More

☐ Check all    With selected: Browse Change Drop Primary Unique Index Spatial  
☐ Fulltext Add to central columns Remove from central columns

# BILL ITEM

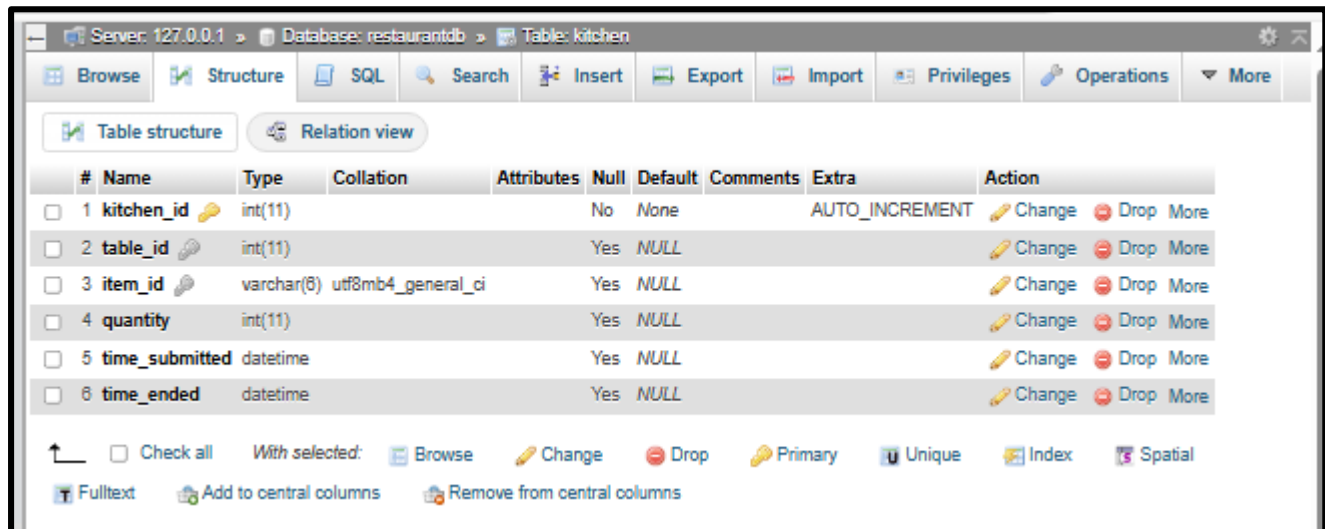
Server: 127.0.0.1 > Database: restaurantdb > Table: bill\_items

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	bill_item_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	bill_id	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	item_id	varchar(6)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 4	quantity	int(11)			Yes	NULL			Change  Drop  More

☐ Check all    With selected: Browse Change Drop Primary Unique Index Spatial  
☐ Fulltext Add to central columns Remove from central columns

# KITCHEN



Server: 127.0.0.1 > Database: restaurantdb > Table: kitchen

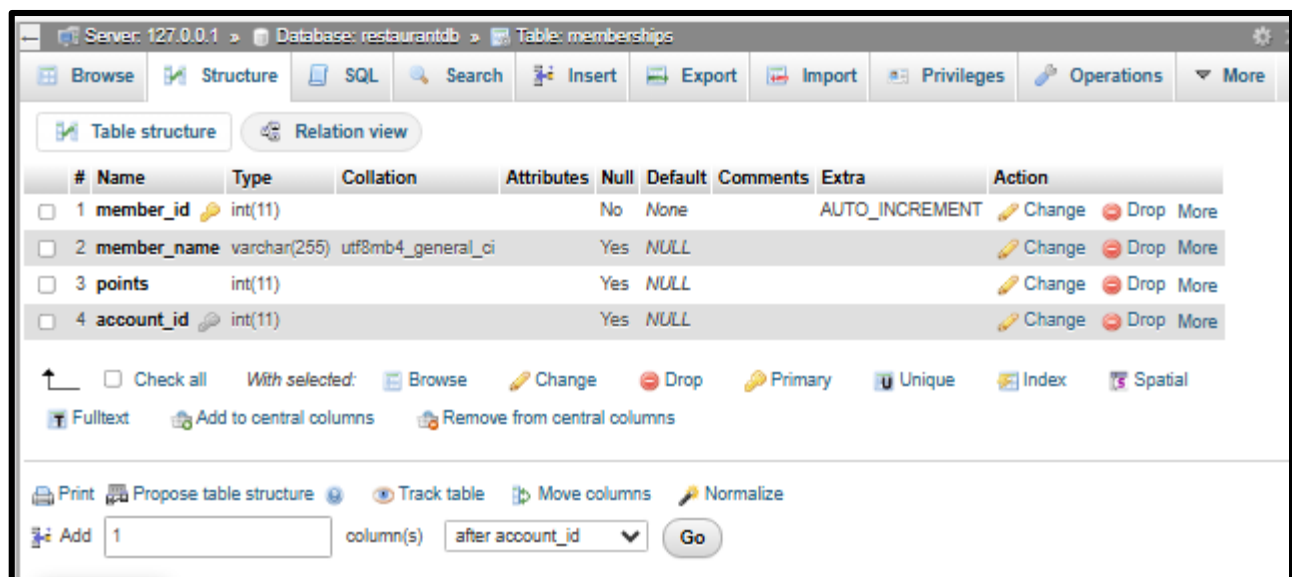
Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 kitchen_id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 table_id	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 item_id	varchar(8)	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 quantity	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	5 time_submitted	datetime			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	6 time_ended	datetime			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#)

[Fulltext](#) [Add to central columns](#) [Remove from central columns](#)

# MEMBERSHIP



Server: 127.0.0.1 > Database: restaurantdb > Table: memberships

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 member_id	int(11)			No	None		AUTO_INCREMENT	<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	2 member_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	3 points	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/>	4 account_id	int(11)			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

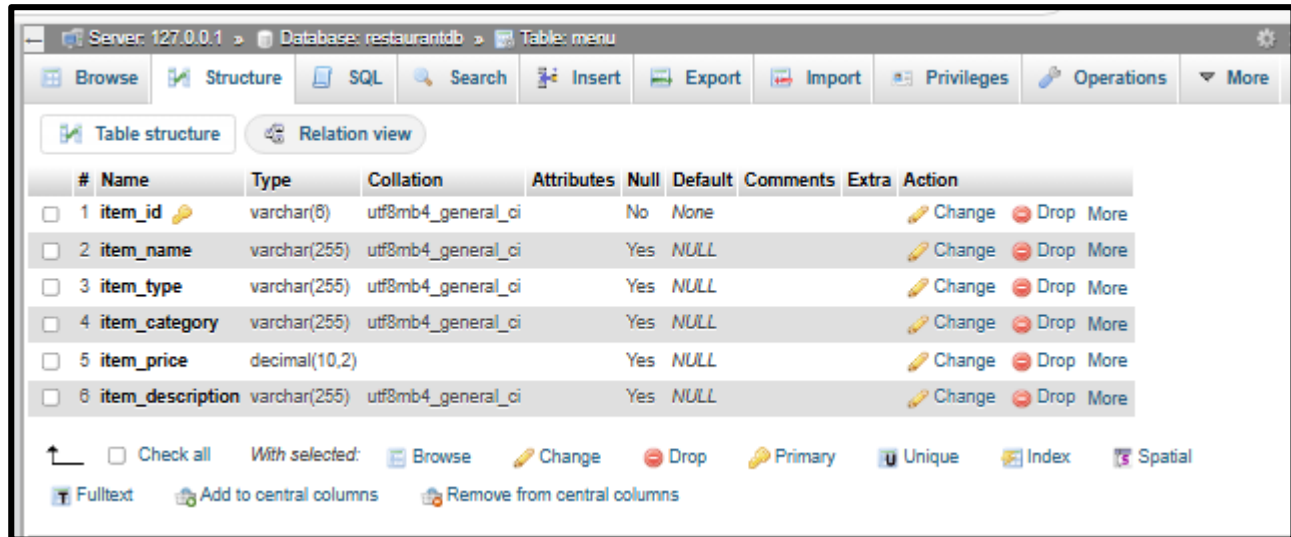
With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Spatial](#)

[Fulltext](#) [Add to central columns](#) [Remove from central columns](#)

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

[Add](#) 1 column(s) after account\_id [Go](#)

# MENU



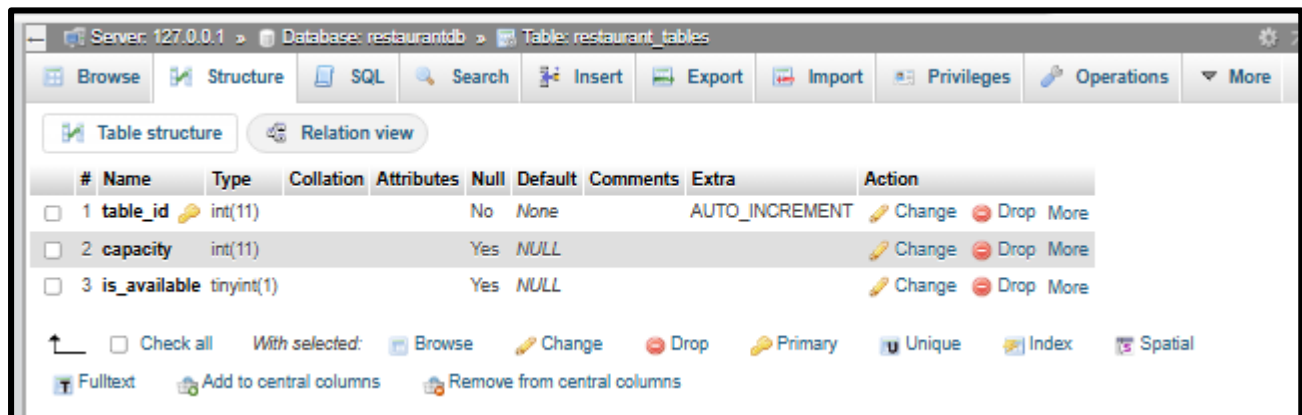
The screenshot shows a database management interface with the following components:

- Server: 127.0.0.1 > Database: restaurantdb > Table: menu
- Navigation tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, More.
- Active tabs: Table structure, Relation view.
- Table structure table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 item_id	varchar(6)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2 item_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	3 item_type	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	4 item_category	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/>	5 item_price	decimal(10,2)			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	6 item_description	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change  Drop  More

Below the table, there are checkboxes for "Check all" and "With selected:", followed by icons for Browse, Change, Drop, Primary, Unique, Index, and Spatial. At the bottom, there are icons for Fulltext, Add to central columns, and Remove from central columns.

# RESTAURANTS\_TABLE



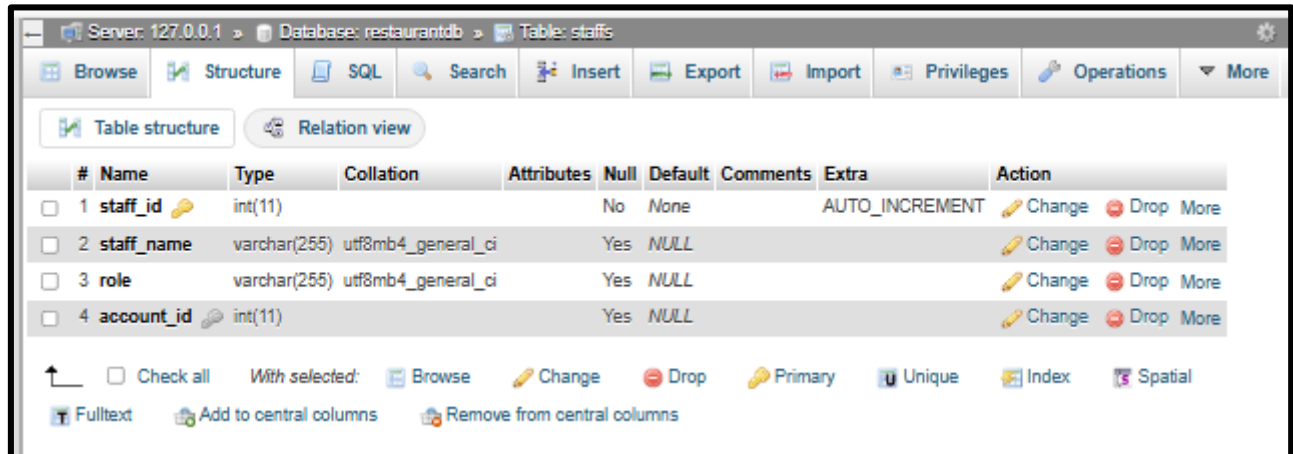
The screenshot shows a database management interface with the following components:

- Server: 127.0.0.1 > Database: restaurantdb > Table: restaurant\_tables
- Navigation tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, More.
- Active tabs: Table structure, Relation view.
- Table structure table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 table_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 capacity	int(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/>	3 is_available	tinyint(1)			Yes	NULL			Change  Drop  More

Below the table, there are checkboxes for "Check all" and "With selected:", followed by icons for Browse, Change, Drop, Primary, Unique, Index, and Spatial. At the bottom, there are icons for Fulltext, Add to central columns, and Remove from central columns.

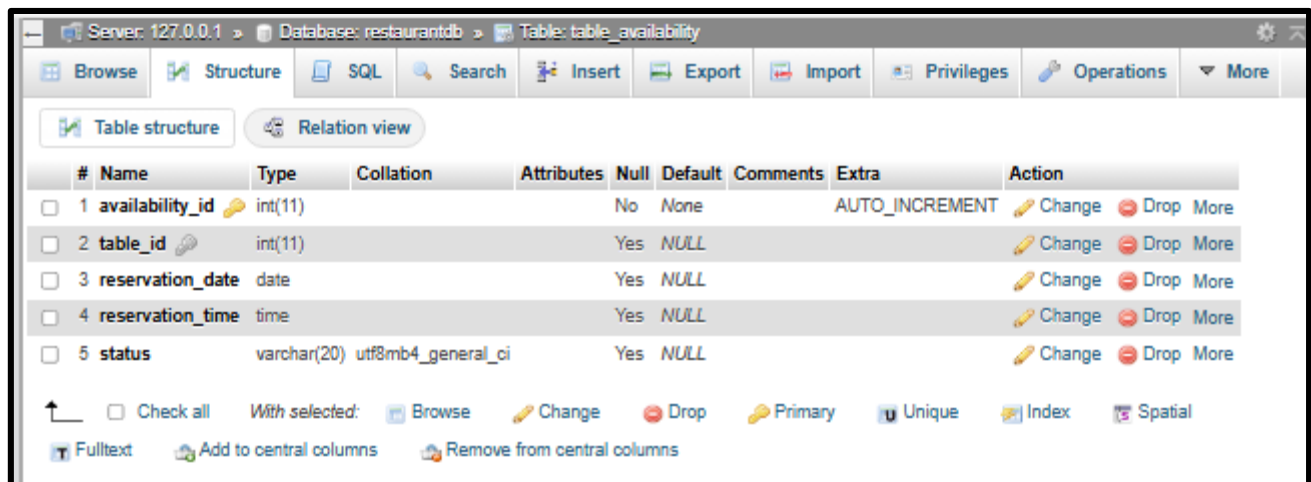
# STAFF



The screenshot shows the 'Table structure' view for the 'staff' table in the 'restaurantdb' database. The table has four columns: 'staff\_id' (int(11), AUTO\_INCREMENT, primary key), 'staff\_name' (varchar(255), utf8mb4\_general\_ci), 'role' (varchar(255), utf8mb4\_general\_ci), and 'account\_id' (int(11)). Each column has a 'Change', 'Drop', and 'More' action button. Below the table structure, there are options to 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', 'Add to central columns', and 'Remove from central columns'.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	staff_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	staff_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	role	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	account_id	int(11)			Yes	NULL			Change Drop More

# TABLE AVAILABILITY

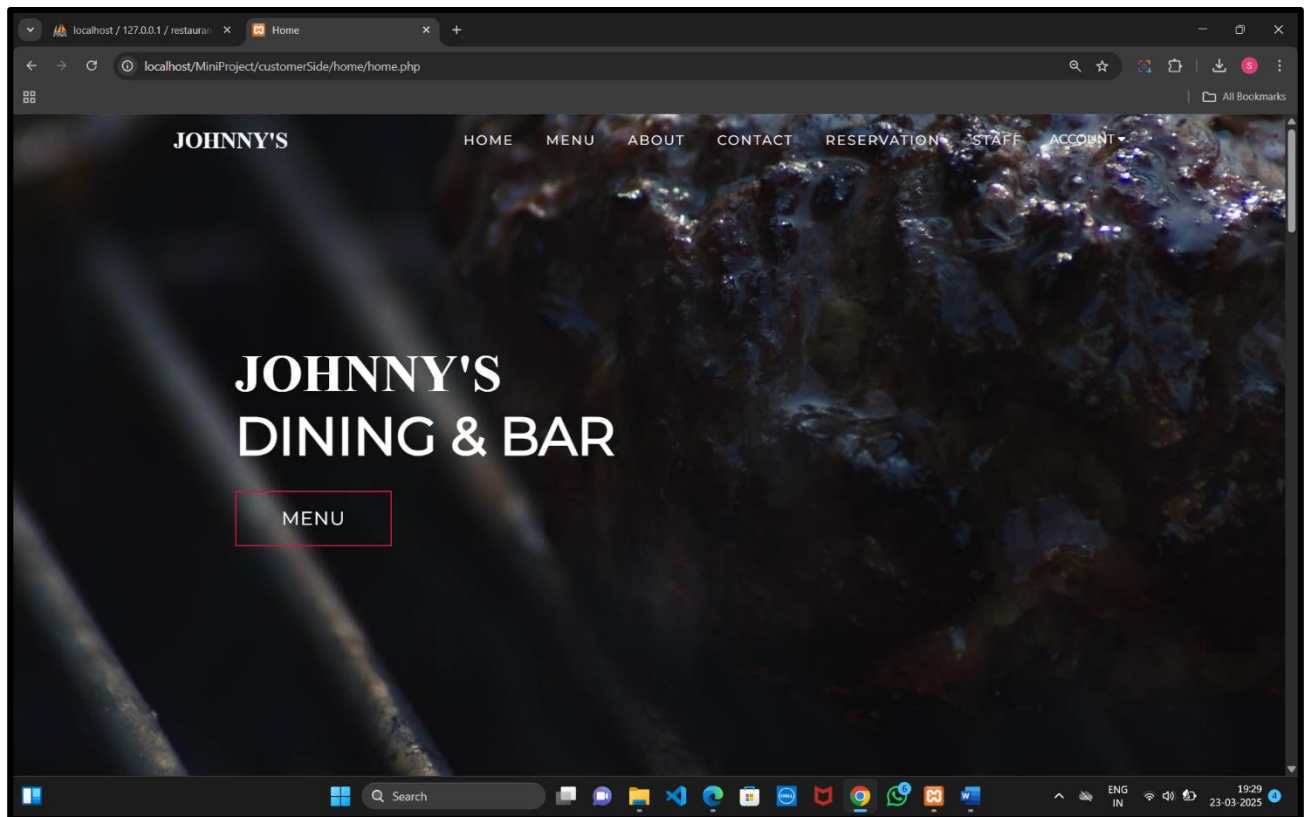


The screenshot shows the 'Table structure' view for the 'table\_availability' table in the 'restaurantdb' database. The table has five columns: 'availability\_id' (int(11), AUTO\_INCREMENT, primary key), 'table\_id' (int(11), foreign key), 'reservation\_date' (date), 'reservation\_time' (time), and 'status' (varchar(20), utf8mb4\_general\_ci). Each column has a 'Change', 'Drop', and 'More' action button. Below the table structure, there are options to 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', 'Fulltext', 'Add to central columns', and 'Remove from central columns'.

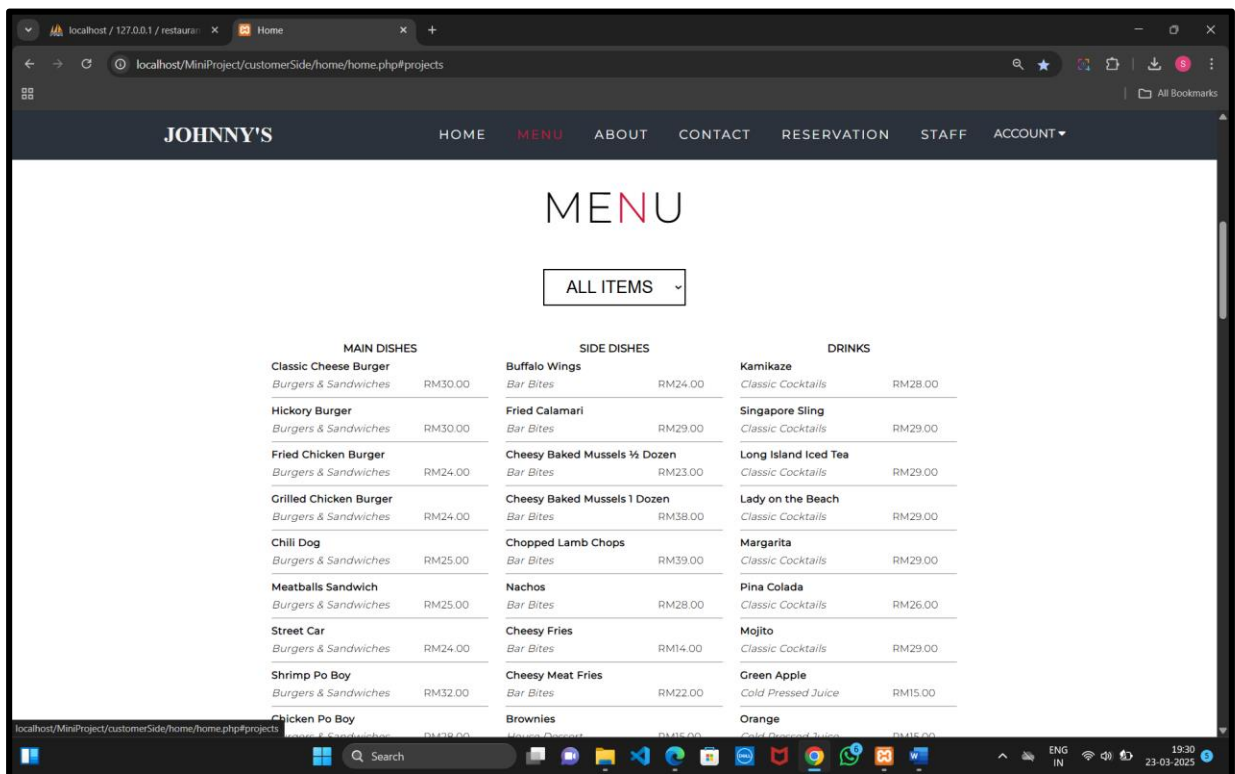
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	availability_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	table_id	int(11)			Yes	NULL			Change Drop More
3	reservation_date	date			Yes	NULL			Change Drop More
4	reservation_time	time			Yes	NULL			Change Drop More
5	status	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More

# Input Design

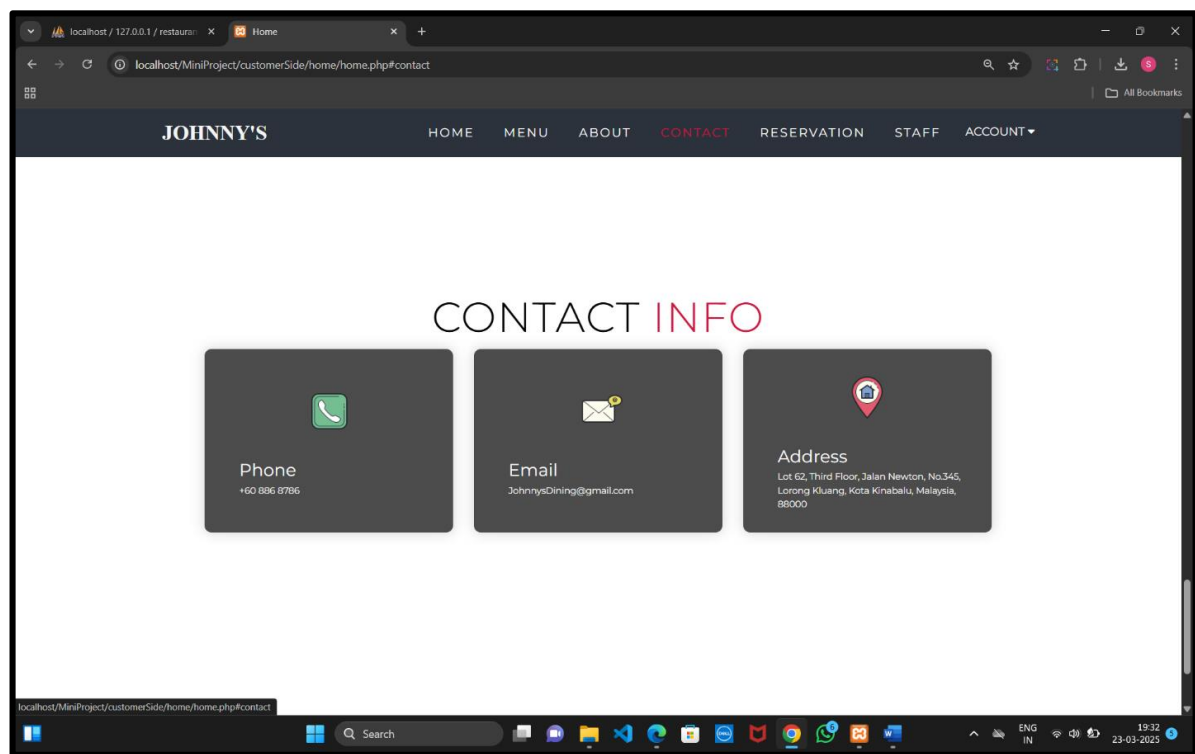
## HOME



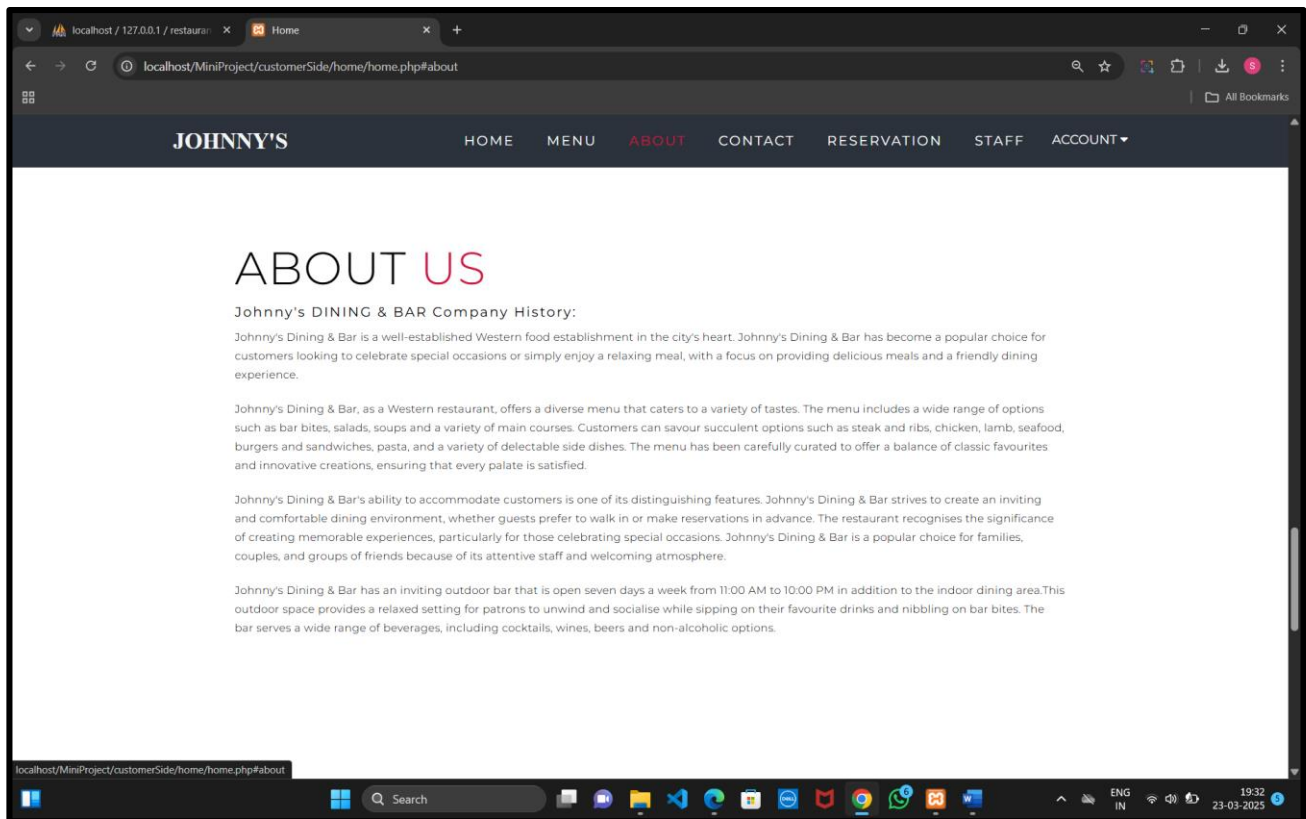
# MENU



# CONTACT



# ABOUT





# RESERVATION

User can make reservation for table

The screenshot shows a web browser window with two tabs: 'localhost / 127.0.0.1 / restaura...' and 'Customer Reservation'. The address bar shows the URL 'localhost/MiniProject/customerSide/CustomerReservation/reservePage.php'. The page content is on a dark blue background and features the restaurant name 'JOHNNY'S' in white. Below the name are two columns of reservation forms. The left column, titled 'Search for Time', includes a 'Select Date' field with a date picker showing 'dd-mm-yyyy', an 'Available Reservation Times' dropdown menu currently set to 'Select a Time', and a 'Search' button. The right column, titled 'Make Reservation', includes a 'Customer Name' text input field, a 'Reservation Date' field with a date picker showing '23-03-2025' and a time field showing '13:00', and an 'Available Tables' dropdown menu currently set to 'Select a Table'. The Windows taskbar is visible at the bottom of the screen.

JOHNNY'S

**Search for Time**

Select Date  
dd-mm-yyyy

Available Reservation Times  
Select a Time

Search

**Make Reservation**

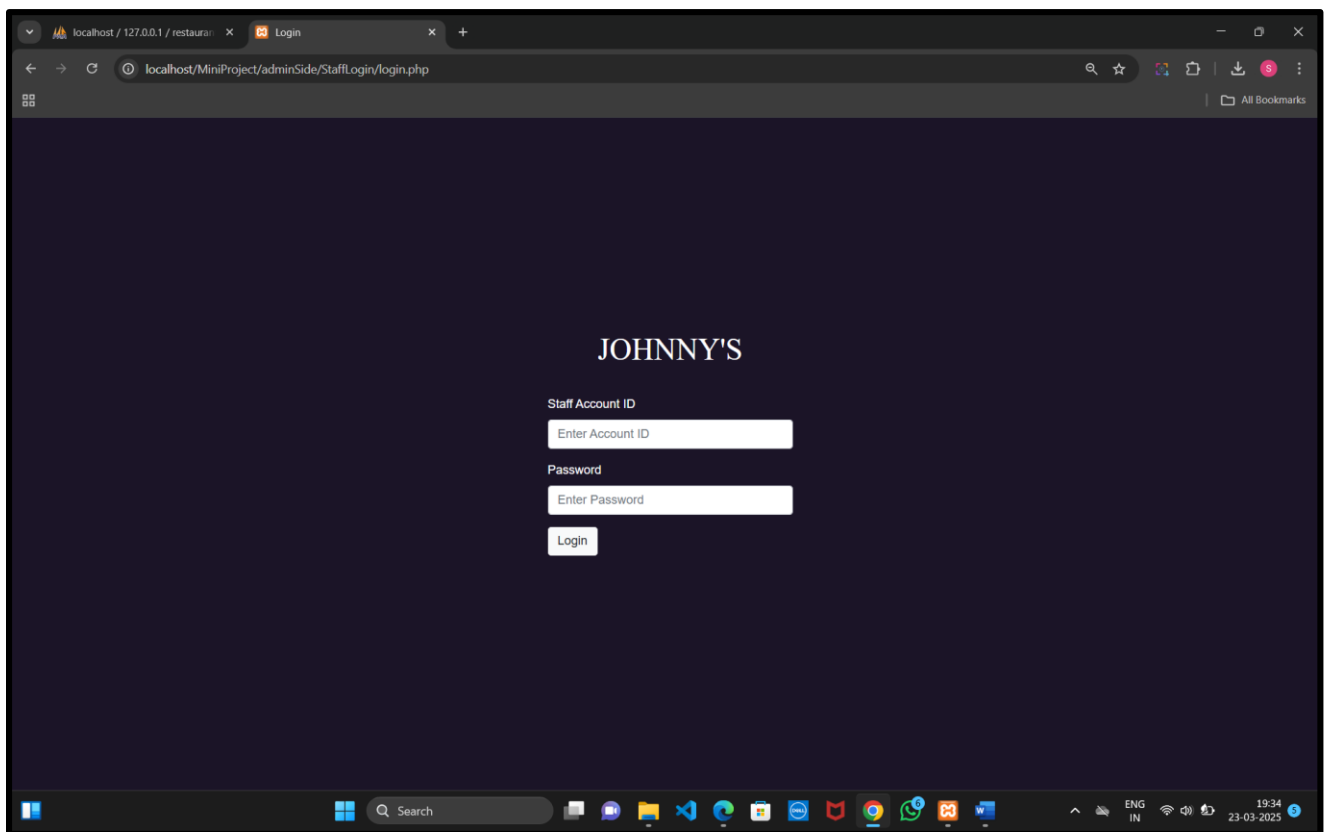
Customer Name

Reservation Date  
23-03-2025 13:00

Available Tables  
Select a Table

# STAFF

## Staff login



The screenshot shows a web browser window with the address bar displaying 'localhost/MiniProject/adminSide/StaffLogin/login.php'. The page has a dark blue background with the text 'JOHNNY'S' in white. Below the text, there are two input fields: 'Enter Account ID' and 'Enter Password'. A 'Login' button is positioned below the password field. The browser's taskbar at the bottom shows various application icons and the system clock indicating 19:34 on 23-03-2025.

JOHNNY'S

Staff Account ID

Enter Account ID

Password

Enter Password

Login

# STAFF

**Admin can create new staff**

### Create New Staff

Please fill in the Staff Information

Staff ID:

Staff Name:

Role:

Account ID:

Email :

Register Date :

Phone Number:

Password :

Create Staff

# Membership

**Admin can create new membership**

### Create New Membership

Please fill in Membership Information

Member ID:

Member Name :

Points :

Account ID:

Email :

Register Date :

Phone Number:

Password :

Create Membership

## Admin can create a new table for reservation

### Create New Table

Please fill in the Table Information

Table ID :

Capacity :

Create table

## Admin can add new menu item

### Create New Item

Please fill Items Information Properly

Item ID :

Item Name :

Item Type:

Item Category:

Item Price :

Item Description :

Create Item

# User Guidelines

- Users must log in using their valid **Account ID** and **password** to access the system.
- After login, **admin users** are directed to the admin dashboard, while **staff users** access limited features based on their roles.

## **Admins can:**

- Create, update, or delete staff and customer accounts.
- Manage table reservations and check table availability.
- Add, edit, or remove menu items and update their details.

## **Staff users can:**

- View assigned reservations and customer-related details.
- Interact with the system based on their assigned roles (e.g., waiter, chef).
- Use the Logout option after every session to ensure account security.
- Keep login credentials confidential to prevent unauthorized access.
- Make sure to use a modern web browser (e.g., Google Chrome, Mozilla Firefox) for best performance.
- The system is designed to be run locally using XAMPP or any other local PHP server.
- For issues such as login errors, missing features, or bugs, users should contact the system administrator.

# Installation Procedure:

## Set up for developer :

- Install **Visual Studio Code** or any preferred source code editor to edit project files.
- Install **XAMPP** which provides an Apache server and MySQL database environment for running PHP applications.
- (Optional) Install **Node.js** if you plan to add JavaScript-based enhancements in the future.
- Extract the project folder (RestaurantProject-main) and place it in the XAMPP htdocs directory:

`C:\xampp\htdocs\`

- Open the XAMPP Control Panel and start both Apache and MySQL services.
- On your browser and navigate to <http://localhost/phpmyadmin>, then:
- Create a new database named `restaurantdb`.
- Import the SQL file (`restaurantdb.sql`) provided with the project to set up the required tables and sample data.

## Set up for end user :

To access and use the system, the following steps should be followed by the end users:

- Ensure a modern **web browser** (such as Google Chrome or Mozilla Firefox) is installed.
- Ensure **local network connectivity** is available, especially if hosted on a local server.
- Open the browser and visit the following URL:

<http://localhost/RestaurantProject-main/MiniProject/>

- Log in with the provided **Account ID** and **Password** to begin using the system.

```

<?php
define('DB_HOST','localhost');
define('DB_USER','root');
define('DB_PASS','');
define('DB_NAME','restaurantdb');

//Create Connection

$link = new mysqli(DB_HOST,DB_USER,DB_PASS,DB_NAME);

//Check Connection
if($link->connect_error){ //if not Connection
die('Connection Failed'. $link->connect_error);//kills the Connection OR terminate
execution
}

$sqlmainDishes = "SELECT * FROM menu WHERE item_category = 'Main Dishes'
ORDER BY item_type; ";
$resultmainDishes = mysqli_query($link, $sqlmainDishes);
$mainDishes = mysqli_fetch_all($resultmainDishes, MYSQLI_ASSOC);

$sqldrinks = "SELECT * FROM menu WHERE item_category = 'Drinks' ORDER BY
item_type; ";
$resultdrinks = mysqli_query($link, $sqldrinks);
$drinks = mysqli_fetch_all($resultdrinks, MYSQLI_ASSOC);

$sqlsides = "SELECT * FROM menu WHERE item_category = 'Side Snacks' ORDER
BY item_type; ";
$resultsides = mysqli_query($link, $sqlsides);

```



```
$sides = mysqli_fetch_all($resultsides, MYSQLI_ASSOC);
```

```
// Check if the user is logged in
```

```
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
```

```
    echo '<div class="user-profile">';
```

```
    echo 'Welcome, ' . $_SESSION["member_name"] . '!';
```

```
    echo '<a href=" ../customerProfile/profile.php">Profile</a>';
```

```
    echo '</div>';
```

```
}
```

```
session_start();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

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

```
    <link rel="stylesheet" href=" ../css/style.css">
```

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/4.7.0/css/font-awesome.min.css">
```

```
    <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></scri
pt>
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

```
<title>Home</title>
```

```
</head>
```

```
<body>
```

```
<!-- Header -->
```

```
<section id="header">
```

```
<div class="header container">
```

```
<div class="nav-bar">
```

```
<div class="brand">
```

```
<a class="nav-link" href="../home/home.php#hero">
```

```
<h1 class="text-center" style="font-family:Copperplate; color:whitesmoke;">
JOHNNY'S</h1><span class="sr-only"></span>
```

```
</a>
```

```
</div>
```

```
<div class="nav-list">
```

```
<div class="hamburger">
```

```
<div class="bar"></div>
```

```
</div>
```

```
<div class="navbar-container">
```

```
<div class="navbar">
```

```

<ul>
  <li><a href="#hero" data-after="Home">Home</a></li>

  <li><a href="#projects" data-after="Projects">Menu</a></li>
  <li><a href="#about" data-after="About">About</a></li>
  <li><a href="#contact" data-after="Contact">Contact</a></li>
  <li><a href="../CustomerReservation/reservePage.php" data-
after="Service">Reservation</a></li>
  <li><a href="../../adminSide/StaffLogin/login.php" data-
after="Staff">Staff</a></li>

</ul>

<div class="dropdown">
  <button class="dropbtn">ACCOUNT <i class="fa fa-caret-down" aria-
hidden="true"></i> </button>
  <div class="dropdown-content">

<?php

// Get the member_id from the query parameters
$account_id = $_SESSION['account_id'] ?? null; // Change this to the way you obtain
the member ID

// Create a query to retrieve the member's information
//$query = "SELECT member_name, points FROM memberships WHERE
account_id = $account_id";

```

```

// Execute the query
// $result = mysqli_query($link, $query);

// Check if the user is logged in
if (isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true &&
    $account_id != null) {
    $query = "SELECT member_name, points FROM memberships WHERE
    account_id = $account_id";

// Execute the query
$result = mysqli_query($link, $query);
    // If logged in, show "Logout" link
    // Check if the query was successful
    if ($result) {
        // Fetch the member's information
        $row = mysqli_fetch_assoc($result);

        if ($row) {
            $member_name = $row['member_name'];
            $points = $row['points'];

            // Calculate VIP status
            $vip_status = ($points >= 1000) ? 'VIP' : 'Regular';

            // Define the VIP tooltip text
            $vip_tooltip = ($vip_status === 'Regular') ? ($points < 1000 ? (1000 - $points)
            . ' points to VIP ' : 'You are eligible for VIP') : "";

```

```

        // Output the member's information

        echo "<p class='logout-link' style='font-size:1.3em; margin-left:15px;
padding:5px; color:white; '>$member_name</p>";

        echo "<p class='logout-link' style='font-size:1.3em; margin-
left:15px;padding:5px;color:white; '>$points Points </p>";

        echo "<p class='logout-link' style='font-size:1.3em; margin-
left:15px;padding:5px; color:white; '>$vip_status";


        // Add the tooltip only for Regular status
        if ($vip_status === 'Regular') {
            echo " <span class='tooltip'>$vip_tooltip</span>";
        }

        echo "</p>";
    } else {
        echo "Member not found.";
    }
} else {
    echo "Error: " . mysqli_error($link);
}

    echo '<a class="logout-link" style="color: white; font-size:1.3em;"
href="./customerLogin/logout.php">Logout</a>';
} else {
    // If not logged in, show "Login" link

    echo '<a class="signin-link" style="color: white; font-size:15px;"
href="./customerLogin/register.php">Sign Up </a> ';
```

```
// Close the database connection
mysqli_close($link);

?>
```

```
<!-- Hero Section with Video Background and Text Overlay -->
<section id="hero" style="position: relative;">
```

```
<video autoplay loop muted playsinline poster="your-poster-image.jpg"
style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; object-fit: cover;">
```

```
<source src="../../image/SteakOnGrillCloseup.mp4" type="video/mp4">
```

```
<!-- Add additional source elements for
```

```
1. SteakOnGrillCloseup
```

```
other video formats if needed -->
```

```
</video>
```

```
<div class="hero container" style="position: relative; z-index: 1;">
```

```
<div>
```

```
<h1><strong><h1 class="text-center" style="font-family:Copperplate;
color:whitesmoke;"> JOHNNY'S</h1><span></span></strong></h1>
```

```
<h1><strong style="color:white;">DINING &
BAR<span></span></strong></h1>
```

```
<a href="#projects" type="button" class="cta">MENU</a>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- End Hero Section -->
```

```
<!-- menu Section -->
```

```
<section id="projects">
```

```
<div class="projects container">
```

```
<div class="projects-header">
```

```
<h1 class="section-title" >Me<span>n</span>u</h1>
```

```
</div>
```

```

<select style="text-align:center;" id="menu-category" class="menu-category">
  <option value="blue">ALL ITEMS</option>
  <option value="yellow">MAIN DISHES</option>
  <option value="red">SIDE DISHES</option>
  <option value="green">DRINKS</option>
</select>

<div class="yellow msg">

  <div></div>
  <div class="mainDish">
    <h1 style="text-align:center;">MAIN DISHES</h1>
    <?php foreach ($mainDishes as $item): ?>
  <p>
    <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
    <span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
    <span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
    <hr>

  </p>
<?php endforeach; ?>
  </div>
</div>

```



```
<div class="red msg">
  <div></div>
  <div class="sideDish">
    <h1 style="text-align:center">SIDE DISHES</h1>
    <?php foreach ($sides as $item): ?>
      <p>
        <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
        <span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
        <span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
      <hr>
    </p>
  <?php endforeach; ?>
</div>
</div>
```

```
<div class="green msg">
  <div></div>
  <div class="drinks">
    <h1 style="text-align:center">DRINKS</h1>
    <?php foreach ($drinks as $item): ?>
      <p>
        <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
```

```
<span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
<span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
<hr>
</p>
<?php endforeach; ?>
</div>
</div>
```

```
<div class="blue msg">
```

```
<div class="mainDish">
  <h1 style="text-align:center">MAIN DISHES</h1>
  <?php foreach ($mainDishes as $item): ?>
    <p>
      <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
      <span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
      <span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
      <hr>
    </p>
  <?php endforeach; ?>
</div>
```

```
<div class="sideDish">
```

```

        <h1 style="text-align:center">SIDE DISHES</h1>
        <?php foreach ($sides as $item): ?>
        <p>
            <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
            <span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
            <span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
            <hr>
        </p>
        <?php endforeach; ?>
    </div>

    <div class="drinks">
        <h1 style="text-align:center">DRINKS</h1>
        <?php foreach ($drinks as $item): ?>
        <p>
            <span class="item-name"> <strong><?php echo $item['item_name'];
?></strong></span>
            <span class="item-price">RM<?php echo $item['item_price']; ?></span><br>
            <span class="item_type"><i><?php echo $item['item_type']; ?></i></span>
            <hr>
        </p>
        <?php endforeach; ?>
    </div>

</div>

```

</div>

</section>

<!-- End menu Section -->

<!-- About Section -->

<section id="about" ">

<div class="about container">

<div class="col-right">

<h1 class="section-title" >About <span>Us</span></h1>

<h2>Johnny's DINING & BAR Company History:</h2>

<p>Johnny's Dining & Bar is a well-established Western food establishment in the city's heart. Johnny's Dining & Bar has become a popular choice for customers looking to celebrate special occasions or simply enjoy a relaxing meal, with a focus on providing delicious meals and a friendly dining experience.

</p>

<p>Johnny's Dining & Bar, as a Western restaurant, offers a diverse menu that caters to a variety of tastes. The menu includes a wide range of options such as bar bites, salads, soups and a variety of main courses. Customers can savour succulent options such as steak and ribs, chicken, lamb, seafood, burgers and sandwiches, pasta, and a variety of delectable side dishes. The menu has been carefully curated to offer a balance of classic favourites and innovative creations, ensuring that every palate is satisfied.

</p>

<p>Johnny's Dining & Bar's ability to accommodate customers is one of its distinguishing features. Johnny's Dining & Bar strives to create an inviting and comfortable dining environment, whether guests prefer to walk in or make reservations in advance. The restaurant recognises the significance of creating memorable experiences, particularly for those celebrating special occasions. Johnny's Dining & Bar is a popular choice for families, couples, and groups of friends because of its attentive staff and welcoming atmosphere.

</p>

<p>Johnny's Dining & Bar has an inviting outdoor bar that is open seven days a week from 11:00 AM to 10:00 PM in addition to the indoor dining area. This outdoor space provides a relaxed setting for patrons to unwind and socialise while sipping on their favourite drinks and nibbling on bar bites. The bar serves a wide range of beverages, including cocktails, wines, beers and non-alcoholic options.

</p>

</div>

</div>

</section>

<!-- End About Section -->

<!-- Contact Section -->

<section id="contact" ">

<div class="contact container">

<div>

<h1 class="section-title">Contact <span>info</span></h1>

</div>

<div class="contact-items">

<div class="contact-item contact-item-bg">

<div class="contact-info">

<div class='icon'></div>

<h1>Phone</h1>

```

        <h2>+60 886 8786</h2>
    </div>
</div>

<div class="contact-item contact-item-bg">
    <div class="contact-info">
        <div class='icon'></div>
        <h1>Email</h1>
        <h2>JohnnysDining@gmail.com</h2>
    </div>
</div>

<div class="contact-item contact-item-bg">
    <div class="contact-info">
        <div class='icon'> </div>
        <h1>Address</h1>
        <h2>Lot 62, Third Floor, Jalan Newton, No.345, Lorong Kluang, Kota Kinabalu,
Malaysia, 88000</h2>
    </div>
</div>
</div>
</div>
</section>
<!-- End Contact Section -->

```

```
<!-- Footer -->
<section id="footer">
  <div class="footer container">
    <div class="brand">
      <h1>Drinks on the House</h1>
    </div>
    <h2>Follow our Socials</h2>
    <div class="social-icon">
      <div class="social-item">
        <a href="https://www.facebook.com"></a>
      </div>
      <div class="social-item">
        <a href="https://www.instagram.com"></a>
      </div>
      <div class="social-item">
        <a href="https://www.pinterest.com"></a>
      </div>
      <div class="social-item">
```

```
        <a href="https://www.tiktok.com"></a>
    </div>

    <div class="social-item">

        <a href="https://www.youtube.com"></a>
    </div>

</div>

<p>© 2023 Johnny's Dining & Bar</p>
```

```
</div>
</section>
<!-- End Footer -->
<script src="../js/app.js"></script>
<style type="text/css">
```

```
.navbar-container {
width: 100%;
padding: 0;
margin: 0;
}

.msg {
font-family: 'Montserrat', sans-serif;
margin-top: 25px;
```



```
padding: 25px;
display: none;
color: black;
}
.yellow {
background: #fff;
}
.green {
background: #fff;
}
.red {
background: #fff;
}

/* Styling the select button */
.menu-category {
font-size: 24px;
padding: 10px;
border: 2px solid black; /* Red border */
outline: none;
cursor: pointer;
transition: border-color 0.3s ease, background-color 0.3s ease, color 0.3s ease;
color: #000; /* Black text */
background-color: #fff; /* White background */
border-radius: 0; /* No border radius (sharp corners) */
}
```

```
/* Style the option text in the select dropdown */
.menu-category option {
  font-size: 20px;
}

/* Hover effect */
.menu-category:hover {
  background-color: black; /* Red background on hover */
  color: white; /* Black text on hover */
}

/* Use CSS Grid to create three columns */
.msg {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* Three columns with equal width */
  grid-gap: 24px; /* Adjust the gap between items */
}

/* Style the menu item content */
.msg p {
  margin: 5px 0;
}

.item-name {
  display: inline-block; /* Ensure items are displayed on separate lines */
  width: 100%; /* Adjust the width as needed */
  float: left;
```

```
}
```

```
.item-price {  
  display: inline-block; /* Ensure prices are displayed on separate lines */  
  width: 30%; /* Adjust the width as needed */  
  float: right;  
}
```

```
.user-profile {  
  display: flex;  
  align-items: center;  
  color: white;  
  margin-right: 20px;  
}
```

```
.user-profile a {  
  margin-left: 10px;  
  color: white;  
  text-decoration: none;  
}
```

```
/* Style for the profile link */  
.profile-link {  
  border: 1px solid #fff; /* Smaller border style and color */  
  padding: 3px 8px; /* Smaller padding inside the border */  
  border-radius: 3px; /* Rounded corners for the border */  
  text-decoration: none; /* Remove the default underline */
```

```
color: #fff; /* Text color */
margin-left: auto; /* Automatically push the link to the right */
margin-right: 10px; /* Add a small right margin for spacing */
}
```

```
#contact .col-right h2 {
    font-size: 24px; /* Adjust the font size */
    color: white; /* Text color for the right column */
}
```

```
#contact .col-right p {
    font-size: 18px; /* Adjust the font size */
    color: white; /* Text color for the right column */
}
```

```
/* Style for the contact-item containers */
.contact-item-bg {
    background-color: rgba(0, 0, 0, 0.7); /* Semi-transparent black background */
    padding: 20px;
    border-radius: 5px;
    margin-bottom: 20px; /* Add margin between contact items */
}
```

```
.contact-item-bg h1,
.contact-item-bg h2 {
    color: white; /* Text color for the contact items */
}
```

```
}
```

```
.contact-item-bg i {  
  color: #fff; /* Icon color */  
}
```

```
.contact-item-bg .icon img {  
  width: 80px; /* Adjust the width of the icon images */  
  height: 80px; /* Adjust the height of the icon images */  
}
```

```
.navbar {  
  overflow: hidden;  
  
}
```

```
.navbar a {  
  float: left;  
  font-size: 16px;  
  color: white;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}
```

```
.dropdown {  
  float: right;  
  overflow: hidden;  
}
```

```
.dropdown .dropbtn {  
  font-size: 17px;  
  border: none;  
  outline: none;  
  color: white;  
  padding: 13.9px 16px;  
  background-color: inherit;  
  font-family: inherit;  
  margin: 0;  
  margin-top: 6px;  
}
```

```
.dropdown:hover .dropbtn {  
  color: crimson;  
  
}
```

```
.dropdown-content {  
  display: none;  
  position: absolute;  
  background-color: rgba(0, 0, 0, 0.5);  
  min-width: 160px;
```

```
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
z-index: 1;
color: black;
}
```

```
.dropdown-content a {
float: none;
color: white;
padding: 12px 16px;
text-decoration: none;
display: block;
text-align: left;
}
```

```
.dropdown-content a:hover {
background-color: #ddd;
}
```

```
/* Style for the dropdown content text */
```

```
.dropdown-content a {
float: none;
color: black;
padding: 12px 16px;
text-decoration: none;
display: block;
text-align: left;
}
```

```

/* Hover effect for dropdown content text */
.dropdown-content a:hover {
    background-color: #ddd;
    color: black; /* Set the text color to black on hover if needed */
}

.dropdown:hover .dropdown-content {
    display: block;
}

.tooltip {
    display: none;
    position: absolute;
    background-color: rgba(0, 0, 0, 0.8);
    color: white;
    padding: 5px;
    border-radius: 3px;
    font-size: 0.9em;
    margin-top: 50px; /* Add margin to move the tooltip below the element */
    left: 0; /* Set left to 0 to align with the element */
    width: 100%; /* Make the tooltip span the width of the element */
    text-align: center; /* Center the text within the tooltip */
}

</style>
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
<script type="text/javascript">

```



```

$(document).ready(function(){
    $("select").change(function(){
        $(this).find("option:selected").each(function(){
            var val = $(this).attr("value");
            if(val){
                $(".msg").not("." + val).hide();
                $("." + val).show();
            } else{
                $(".msg").hide();
            }
        });
    }).change();
});

```

```

$(document).ready(function(){
    // Function to filter menu items based on search input
    function filterMenuItems(searchTerm) {
        $(".item-name").each(function() {
            var itemName = $(this).text().toLowerCase();
            if (itemName.includes(searchTerm)) {
                $(this).closest(".msg").show();
            }
        });
    }

```

```

    } else {
        $(this).closest(".msg").hide();
    }
});
}

// Search button click event
$("#search-button").click(function() {
    var searchTerm = $("#search-input").val().toLowerCase();
    filterMenuItems(searchTerm);
});

// Search input keyup event
$("#search-input").keyup(function() {
    var searchTerm = $(this).val().toLowerCase();
    filterMenuItems(searchTerm);
});
});

$(document).ready(function() {
    $('.dropdown-toggle').dropdown();
});

</script>
<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

```

```
<!-- Bootstrap JS (including Popper.js) -->

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

$(document).ready(function () {
    $('.logout-link').hover(function () {
        var $tooltip = $(this).find('.tooltip');
        var elementHeight = $(this).height();

        $tooltip.css('top', elementHeight + 10 + 'px'); // Position the tooltip below the
element

        $tooltip.css('display', 'block');
    }, function () {
        $(this).find('.tooltip').css('display', 'none');
    });
});

</script>

</body>

</html>
```

```

<?php

// Include your database connection code here

require_once "config.php"; // Make sure to replace "config.php" with your actual
database connection file.

session_start();


// Define variables for email and password
$email = $password = "";
$email_err = $password_err = "";


// Check if the form was submitted.
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Validate email
    if (empty(trim($_POST["email"]))) {
        $email_err = "Please enter your email.";
    } else {
        $email = trim($_POST["email"]);
    }


    // Validate password
    if (empty(trim($_POST["password"]))) {
        $password_err = "Please enter your password.";
    } else {
        $password = trim($_POST["password"]);
    }
}

```

```

}

// Check input errors before checking authentication
if (empty($email_err) && empty($password_err)) {
    // Prepare a select statement
    $sql = "SELECT * FROM Accounts WHERE email = ?";

    if ($stmt = mysqli_prepare($link, $sql)) {
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "s", $param_email);

        // Set parameters
        $param_email = $email;

        // Attempt to execute the prepared statement
        if (mysqli_stmt_execute($stmt)) {
            // Get the result
            $result = mysqli_stmt_get_result($stmt);

            // Check if a matching record was found.
            if (mysqli_num_rows($result) == 1) {
                // Fetch the result row
                $row = mysqli_fetch_assoc($result);

                // Verify the password
                if ($password === $row["password"]) {

```

```

        // Password is correct, start a new session and redirect the user to a
        dashboard or home page.

        $_SESSION["loggedin"] = true;
        $_SESSION["email"] = $email;

        // Query to get membership details

        $sql_member = "SELECT * FROM Memberships WHERE account_id
= " . $row['account_id'];

        $result_member = mysqli_query($link, $sql_member);

        if ($result_member) {

            $membership_row = mysqli_fetch_assoc($result_member);

            if ($membership_row) {

                $_SESSION["account_id"] = $membership_row["account_id"];
                header("location: ../home/home.php"); // Redirect to the home
page

                exit;
            } else {

                // No membership details found

                $password_err = "No membership details found for this
account.";

            }
        } else {

            // Error in membership query

            $password_err = "Error fetching membership details: " .
mysqli_error($link);

        }

```

```

        } else {
            // Password is incorrect
            $password_err = "Invalid password. Please try again.";
        }

    } else {
        // No matching records found
        $email_err = "No account found with this email.";
    }
} else {
    echo "Oops! Something went wrong. Please try again later.";
}

// Close the statement
mysqli_stmt_close($stmt);
}
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>

```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
```

```
<style>
```

```
/* Style for the container within login.php */
```

```
.login-container {
padding: 50px; /* Adjust the padding as needed */
border-radius: 10px; /* Add rounded corners */
margin: 100px auto; /* Center the container horizontally */
max-width: 500px; /* Set a maximum width for the container */
}
```

```
body {
font-family: 'Montserrat', sans-serif;
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
margin: 0; /* Remove default margin */
background-color: black;
background-image: url('../image/loginBackground.jpg'); /* Set the background
image path */
```



```
background-size: cover;
background-position: center;
background-repeat: no-repeat;
background-attachment: fixed;
color: white;
}

.login_wrapper {
width: 400px; /* Adjust the container width as needed */
padding: 20px;
}

h2 {
text-align: center;
font-family: 'Montserrat', serif;
}

p {
font-family: 'Montserrat', serif;
}

.form-group {
margin-bottom: 15px; /* Add space between form elements */
}

::placeholder {
font-size: 12px; /* Adjust the font size as needed */
```

```

    }

    .text-danger{
        font-size: 13px;
    }
</style>
</head>
<body>
    <div class="login-container">
        <div class="login_wrapper">
            <a class="nav-link" href="../home/home.php#hero"> <h1 class="text-center"
style="font-family:Copperplate; color:white;"> JOHNNY'S</h1><span class="sr-
only"></span></a>

            <div class="wrapper">

                <form action="login.php" method="post">
                    <div class="form-group">
                        <label>Email</label>

                        <input type="email" name="email" class="form-control" placeholder="Enter
User Email" required>

                        <span class="text-danger"><?php echo $email_err; ?></span>
                    </div>

                    <div class="form-group">
                        <label>Password</label>

                        <input type="password" name="password" class="form-control"
placeholder="Enter User Password" required>

```

```
<span class="text-danger"><?php echo $password_err; ?></span>
</div>

    <button class="btn btn-dark" style="background-color:black;" type="submit"
name="submit" value="Login">Login</button>

</form>

    <p style="margin-top:1em; color:white;">Don't have an account? <a
href="register.php" style="">Proceed to Register</a></p>
</div>
</div>
</div>
</body>
</html>
```

```

<?php
session_start(); // Ensure session is started
?>

<?php
// Database connection parameters
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "restaurantdb";

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

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

//success login pattern
$message = "Register successful.<br>Welcome to Johnny's Bar &
Dining.<br>Please Login with your Account.";
$iconClass = "fa-check-circle";
$cardClass = "alert-success";
$bgColor = "#D4F4DD";
$direction = "login.php"; // Success, go to staff panel

```

```
// Close the database connection
```

```
$conn->close();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <link  
href="https://fonts.googleapis.com/css?family=Nunito+Sans:400,400i,700,900&displa  
y=swap" rel="stylesheet">
```

```
  <style>
```

```
    /* Your custom CSS styles for the success message card here */
```

```
    body {
```

```
      text-align: center;
```

```
      padding: 40px 0;
```

```
      background: #EBF0F5;
```

```
    }
```

```
    h1 {
```

```
      color: #88B04B;
```

```
      font-family: "Nunito Sans", "Helvetica Neue", sans-serif;
```

```
      font-weight: 900;
```

```
      font-size: 40px;
```

```
      margin-bottom: 10px;
```

```
    }
```

```
    p {
```

```

    color: #404F5E;
    font-family: "Nunito Sans", "Helvetica Neue", sans-serif;
    font-size: 20px;
    margin: 0;
}
i.checkmark {
    color: #9ABC66;
    font-size: 100px;
    line-height: 200px;
    margin-left: -15px;
}
.card {
    background: white;
    padding: 60px;
    border-radius: 4px;
    box-shadow: 0 2px 3px #C8D0D8;
    display: inline-block;
    margin: 0 auto;
}
/* Additional CSS styles based on success/error message */
.alert-success {
    /* Customize the styles for the success message card */
    background-color: <?php echo $bgColor; ?>;
}
.alert-success i {
    color: #5DBE6F; /* Customize the checkmark icon color for success */
}

```

```

.alert-danger {
    /* Customize the styles for the error message card */
    background-color: #FFA7A7; /* Custom background color for error */
}

.alert-danger i {
    color: #F25454; /* Customize the checkmark icon color for error */
}

.custom-x {
    color: #F25454; /* Customize the "X" symbol color for error */
    font-size: 100px;
    line-height: 200px;
}

.alert-box {
    max-width: 300px;
    margin: 0 auto;
}

.alert-icon {
    padding-bottom: 20px;
}

</style>
</head>
<body>
    <div class="card <?php echo $cardClass; ?>" style="display: none;">
        <div style="border-radius: 200px; height: 200px; width: 200px; background:
        #F8FAF5; margin: 0 auto;">

```

```

    <?php if ($iconClass === 'fa-check-circle'): ?>
        <i class="checkmark">✓</i>
    <?php else: ?>
        <i class="custom-x" style="font-size: 100px; line-height: 200px;">X</i>
    <?php endif; ?>
</div>

<h1><?php echo ($cardClass === 'alert-success') ? 'Success' : 'Error'; ?></h1>
<p><?php echo $message; ?></p>
</div>

```

```

<div style="text-align: center; margin-top: 20px;">Redirecting back in <span
id="countdown">3</span></div>

```

```

<script>

//Declare the direction of login success and fail
var direction = "<?php echo $direction; ?>";

// Function to show the message card as a pop-up and start the countdown
function showPopup() {
    var messageCard = document.querySelector(".card");
    messageCard.style.display = "block";

    var i = 3;
    var countdownElement = document.getElementById("countdown");
    var countdownInterval = setInterval(function() {
        i--;
        countdownElement.textContent = i;
    }, 1000);
}

```



```

        if (i <= 0) {
            clearInterval(countdownInterval);
            window.location.href = direction;
        }
    }, 1000); // 1000 milliseconds = 1 second
}

// Show the message card and start the countdown when the page is loaded
window.onload = showPopup;

// Function to hide the message card after a delay
function hidePopup() {
    var messageCard = document.querySelector(".card");
    messageCard.style.display = "none";

    // Redirect to another page after hiding the pop-up (adjust the delay as
    needed)
    setTimeout(function () {
        window.location.href = direction; // Replace with your desired URL
    }, 3000); // 3000 milliseconds = 3 seconds
}

// Hide the message card after 3 seconds (adjust the delay as needed)
setTimeout(hidePopup, 3000);
</script>
</body>
</html>

```

```

<?php
// Include your database connection code here (not shown in this example).

require_once "config.php"; // Make sure to replace "config.php" with your actual
database connection file.

session_start();


// Define variables and initialize them to empty values
$email = $member_name = $password = $phone_number = "";
$email_err = $member_name_err = $password_err = $phone_number_err = "";


// Check if the form was submitted.
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Validate email
    if (empty(trim($_POST["email"]))) {
        $email_err = "Please enter your email.";
    } else {
        $email = trim($_POST["email"]);
    }


    // Validate member name
    if (empty(trim($_POST["member_name"]))) {
        $member_name_err = "Please enter your member name.";
    } else {
        $member_name = trim($_POST["member_name"]);
    }
}

```

```

}

// Validate password
if (empty(trim($_POST["password"]))) {
    $password_err = "Please enter a password.";
} elseif (strlen(trim($_POST["password"])) < 6) {
    $password_err = "Password must have at least 6 characters.";
} else {
    $password = trim($_POST["password"]);
}

// Validate phone number
if (empty(trim($_POST["phone_number"]))) {
    $phone_number_err = "Please enter your phone number.";
} else {
    $phone_number = trim($_POST["phone_number"]);
}

// Check input errors before inserting into the database
if (empty($email_err) && empty($member_name_err) && empty($password_err)
&& empty($phone_number_err)) {
    // Start a transaction
    mysqli_begin_transaction($link);

    // Prepare an insert statement for Accounts table
    // Prepare an insert statement for Accounts table

```

```

$sql_accounts = "INSERT INTO Accounts (email, password, phone_number,
register_date) VALUES (?, ?, ?, NOW())";

if ($stmt_accounts = mysqli_prepare($link, $sql_accounts)) {
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt_accounts, "sss", $param_email,
    $param_password, $param_phone_number);

    // Set parameters
    $param_email = $email;
    // Store the password as plain text (not recommended for production)
    $param_password = $password;
    $param_phone_number = $phone_number;

    // ...
}

// Attempt to execute the prepared statement for Accounts table
if (mysqli_stmt_execute($stmt_accounts)) {
    // Get the last inserted account_id
    $last_account_id = mysqli_insert_id($link);

    // Prepare an insert statement for Memberships table
    $sql_memberships = "INSERT INTO Memberships (member_name, points,
account_id) VALUES (?, ?, ?)";
    if ($stmt_memberships = mysqli_prepare($link, $sql_memberships)) {
        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt_memberships, "sii",
        $param_member_name, $param_points, $last_account_id);
    }
}

```

```

        // Set parameters for Memberships table
        $param_member_name = $member_name;
        $param_points = 0; // You can set an initial value for points

        // Attempt to execute the prepared statement for Memberships table
        if (mysqli_stmt_execute($stmt_memberships)) {
            // Commit the transaction
            mysqli_commit($link);

            // Registration successful, redirect to the login page
            header("location: register_process.php");
            exit;
        } else {
            // Rollback the transaction if there was an error
            mysqli_rollback($link);
            echo "Oops! Something went wrong. Please try again later.";
        }

        // Close the statement for Memberships table
        mysqli_stmt_close($stmt_memberships);
    }
} else {
    // Rollback the transaction if there was an error
    mysqli_rollback($link);
    echo "Oops! Something went wrong. Please try again later.";
}

```

```

        // Close the statement for Accounts table
        mysqli_stmt_close($stmt_accounts);
    }
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Registration Form</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

    <style>
        body {
            font-family: 'Montserrat', sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0; /* Remove default margin */
            background-color: black;

```

```
        background-image: url('../image/loginBackground.jpg'); /* Set the background
image path */
        background-size: cover;
        background-position: center;
        background-repeat: no-repeat;
        background-attachment: fixed;
        color: white;
    }
```

/\* Style for the container within login.php \*/

```
.register-container {
    padding: 50px; /* Adjust the padding as needed */
    border-radius: 10px; /* Add rounded corners */
    margin: 100px auto; /* Center the container horizontally */
    max-width: 500px; /* Set a maximum width for the container */
}
```

```
    .register_wrapper {
        width: 400px; /* Increase the container width */
        padding: 20px;
    }
```

```
h2 {
    text-align: center;
    font-family: 'Montserrat', serif;
}
```

```
p {  
    font-family: 'Montserrat', serif;  
}  
  
.form-group {  
    margin-bottom: 15px; /* Add space between form elements */  
}  
  
::placeholder {  
    font-size: 12px; /* Adjust the font size as needed */  
}  
  
/* Add flip animation class to all Font Awesome icons */  
.fa-flip {  
    animation: fa-flip 3s infinite;  
}  
  
/* Keyframes for the flip animation */  
@keyframes fa-flip {  
    0% {  
        transform: scale(1) rotateY(0);  
    }  
    50% {  
        transform: scale(1.2) rotateY(180deg);  
    }  
    100% {
```



```

        transform: scale(1) rotateY(360deg);
    }
}

</style>
</head>
<body>
    <div class="register-container">
        <div class="register_wrapper"> <!-- Updated class name -->
            <a class="nav-link" href="../home/home.php#hero"> <h1 class="text-center"
style="font-family:Copperplate; color:white;"> JOHNNY'S</h1><span class="sr-
only"></span></a><br>

            <form action="register.php" method="post">
                <div class="form-group">
                    <label>Email</label>

                    <input type="email" name="email" class="form-control" placeholder="Enter
Email" required>

                    <span class="text-danger"><?php echo $email_err; ?></span>
                </div>

                <div class="form-group">
                    <label>Member Name</label>

                    <input type="text" name="member_name" class="form-control"
placeholder="Enter Member Name" required>

                    <span class="text-danger"><?php echo $member_name_err;
?></span>
                </div>
            </form>
        </div>
    </div>

```

```

        <div class="form-group">
            <label>Password</label>

            <input type="password" name="password" class="form-control"
placeholder="Enter Password" required>

                <span class="text-danger"><?php echo $password_err;
?></span>
            </div>

        <div class="form-group">
            <label>Phone Number</label>

            <input type="text" name="phone_number" class="form-control"
placeholder="Enter Phone Number" required>

                <span class="text-danger"><?php echo $phone_number_err;
?></span>
            </div>

        <button style="background-color:black;" class="btn btn-dark" type="submit"
name="register" value="Register">Register</button>

    </form>

    <p style="margin-top:1em; color:white;">Already have an account? <a
href="../customerLogin/login.php" >Proceed to Login</a></p>

</div>

</div>

</body>

</html>

```

```

<?php
session_start(); // Ensure session is started
?>
<?php include '../inc/dashHeader.php'; ?>
<style>
.wrapper{ width: 1300px; padding-left: 200px; padding-top: 20px }
</style>

<div class="wrapper">
<div class="container-fluid pt-5 pl-600">
<div class="row">
<div class="m-50">
<div class="mt-5 mb-3">
<h2 class="pull-left">Account Details</h2>
<a href="../staffCrud/createStaff.php" class="btn btn-outline-dark"><i
class="fa fa-plus"></i> Add Staff</a>
<a href="../customerCrud/createCust.php" class="btn btn-outline-
dark"><i class="fa fa-plus"></i> Add Memberships</a>
</div>

<div class="mb-3">
<form method="POST" action="#">
<div class="row">
<div class="col-md-6">
<input required type="text" id="search" name="search"
class="form-control" placeholder="Enter Account ID, Email">

```

```

        </div>

        <div class="col-md-3">
            <button type="submit" class="btn btn-dark">Search</button>
        </div>

        <div class="col" style="text-align: right;" >
            <a href="account-panel.php" class="btn btn-light">Show All</a>
        </div>
    </div>
</form>
</div>

```

```

<?php
// Include config file
require_once "../config.php";

if (isset($_POST['search'])) {
    if (!empty($_POST['search'])) {
        $search = $_POST['search'];

        $sql = "SELECT *
                FROM Accounts
                WHERE email LIKE '%$search%' OR account_id LIKE
'%$search%'
                ORDER BY account_id;";
    } else {
        // Default query to fetch all accounts
        $sql = "SELECT *

```

```

        FROM Accounts
        ORDER BY account_id;";
    }
} else {
    // Default query to fetch all accounts
    $sql = "SELECT *
        FROM Accounts
        ORDER BY account_id;";
}

if ($result = mysqli_query($link, $sql)) {
    if (mysqli_num_rows($result) > 0) {
        echo '<table class="table table-bordered table-striped">';
        echo "<thead>";
        echo "<tr>";
        echo "<th>Account ID</th>";
        echo "<th>Email</th>";
        echo "<th>Register Date</th>";
        echo "<th>Phone Number</th>";
        echo "<th>Password</th>";
        //echo "<th>Account Type</th>"; // Display account type
        // echo "<th>Delete</th>";
        echo "</tr>";
        echo "</thead>";
        echo "<tbody>";
        while ($row = mysqli_fetch_array($result)) {
            echo "<tr>";

```

```

        echo "<td>" . $row['account_id'] . "</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "<td>" . $row['register_date'] . "</td>";
        echo "<td>" . $row['phone_number'] . "</td>";
        echo "<td>" . $row['password'] . "</td>";
        //echo "<td>" . ucfirst($row['account_type']) . "</td>"; // Display
account type

        // echo "<td>";

        // $deleteSQL = "DELETE FROM Accounts WHERE account_id = "
. $row['account_id'] . " ";

        // echo '<a href="..accountCrud/deleteAccountVerify.php?id=' .
$row['account_id'] . '" title="Delete Record" data-toggle="tooltip" '
        //      . 'onclick="return confirm(\'Admin permission
Required!\n\nAre you sure you want to delete this Account?\n\nThis will alter other
modules related to this Account!\n\')"><span class="fa fa-trash text-
black"></span></a>';

        // echo "</td>";

        echo "</tr>";
    }
    echo "</tbody>";
    echo "</table>";

    // Free result set
    mysqli_free_result($result);

} else {
    echo '<div class="alert alert-danger"><em>No records were
found.</em></div>';
}

} else {
    echo "Oops! Something went wrong. Please try again later.";
}

```

```
    }

    // Close connection
    mysqli_close($link);
    ?>
</div>
</div>
</div>
</div>

<?php include '../inc/dashFooter.php'; ?>
```

```
<?php
session_start(); // Ensure session is started
?>
<?php
include '../inc/dashHeader.php';
require_once '../config.php';

?>
```

```
<div class="container-fluid">
```

```
    <div class="row">
```

```
        <div class="col-md-6 order-md-1 " style="margin-top: 6rem; margin-left:
15rem;">
```

```
            <h2 class="pull-left">Search Member</h2>
```

```
            <form method="get" action="#">
```

```
                <div class="row">
```

```
                    <div class="col-md-6">
```



```
        <input required type="text" id="member_id" style="width:150px"
name="member_id" class="form-control" placeholder="Enter Member ID">
```

```
    </div>
```

```
    <div class="col-md-6">
```

```
        <button type="submit" class="btn btn-dark">Search</button>
```

```
    </div>
```

```
</div>
```

```
</form><br>
```

```
<?php
```

```
require_once '../config.php';
```

```
$currentMonthStart = date('Y-m-01');
```

```
$currentMonthEnd = date('Y-m-t');
```

```
// Get the current month and year in the format 'YYYY-MM'
```

```
$currentMonth = date('Y-m');
```

```
$memberId = isset($_GET['member_id']) ? $_GET['member_id'] : 1;
```

```
// Get member's most ordered items
```

```
$mostOrderedItemsQuery = "SELECT Menu.item_name,
SUM(Bill_Items.quantity) AS order_count
```

```
    FROM Bill_Items
```

```
    INNER JOIN Menu ON Bill_Items.item_id = Menu.item_id
```

```
    INNER JOIN Bills ON Bill_Items.bill_id = Bills.bill_id
```

```
    WHERE Bills.member_id = $memberId
```

```
    GROUP BY Bill_Items.item_id
```

```
    ORDER BY order_count DESC";
```

```

$mostOrderedItemsResult = mysqli_query($link, $mostOrderedItemsQuery);
// Check if any results were returned
if(mysqli_num_rows($mostOrderedItemsResult) == 0) {
    echo "Member ID not found.";
}
else {
    ?>
    <h3>Showing Member ID - <?php echo $memberId; ?></h3>
    <h3>Most Ordered Items - (All Time)</h3>
    <table class="table ">
        <thead>
            <tr>
                <th>Item Name</th>
                <th>Quantity</th>
            </tr>
        </thead>
        <tbody>
            <?php while ($row = mysqli_fetch_assoc($mostOrderedItemsResult)) :
?>
                <tr>
                    <td><?php echo $row['item_name']; ?></td>
                    <td><?php echo $row['order_count']; ?></td>
                </tr>
            <?php endwhile; ?>
        </tbody>
    </table>
    <?php } ?>

```

```

        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-10 order-md-2" style="margin-top: 3rem; margin-left: -10rem;">
        <div class="container-fluid pt-5 row">
            <h3>Top 5 Favourites - (All Time)</h3>
            <div style="width: 800px; height: 800px;">
                <canvas id="mostOrderedItemsChart"></canvas>
            </div>
        </div>
    </div>
</div>
</div>

```

```

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

```

```

<script>

```

```

    // Get data for the donut chart

```

```

    <?php

```

```

    $chartLabels = [];

```

```

    $chartData = [];

```

```

    $chartItemsResult = mysqli_query($link, $mostOrderedItemsQuery);

```

```

    $itemCount = 0;

```

```

    while ($row = mysqli_fetch_assoc($chartItemsResult)) {

```

```

    if ($itemCount >= 5) {
        break;
    }
    array_push($chartLabels, $row['item_name']);
    array_push($chartData, $row['order_count']);
    $itemCount++;
}
?>

// Create the donut chart
var ctx = document.getElementById('mostOrderedItemsChart');

var mostOrderedItemsChart = new Chart(ctx, {
    type: 'doughnut',
    data: {
        labels: <?php echo json_encode($chartLabels); ?>,
        datasets: [{
            data: <?php echo json_encode($chartData); ?>,
            backgroundColor: [
                'rgb(8, 32, 50)',
                'rgb(255, 76, 41)',
                'rgb(13, 18, 130)',
                'rgb(143, 67, 238)',
                'rgb(179, 19, 18)'
            ],
            borderWidth: 1
        }]
    }
});

```

```
    },  
    options: {  
        responsive: true,  
        maintainAspectRatio: false,  
        legend: {  
            display: true,  
            position: 'right'  
        },  
        is3D: true  
    }  
});  
</script>
```

```
<?php include '../inc/dashFooter.php'; // Include your footer file here ?>
```

# REPORTS

**Accounts :** This report shows all account details.

### Account Details

+ Add Staff

+ Add Memberships

Search

Show All

Account ID	Email	Register Date	Phone Number	Password
1	john@gmail.com	2025-02-28	+1234567890	password123
31	joeljohn0@gmail.com	2025-03-23	9987564512	joeljohn
32	siddhant@gmail.com	2025-03-23	8855467890	sidkamble
33	shripatil9082@gmail.com	2025-03-23	7875499082	shreepatil
34	manasikumbhar9@gmail.com	2025-03-23	9987564512	manasikumbhar
35	tejasdesai18@gmail.com	2025-03-23	8855467890	tejasdesai

**Staffs :** This report shows all staff details.

### Staff Details

+ Add Staff

Search

Show All

Staff ID	Staff Name	Role	Account ID
1	John Smith	Waiter	1
2	Susan Johnson	Waiter	2
3	James Brown	Waiter	3
4	Alice Davis	Waiter	4
5	Mike Wilson	Waiter	5
6	Lisa Martinez	Chef	6
7	Robert Miller	Manager	7
8	Emily Moore	Manager	8
9	David Taylor	Chef	9
10	Olivia Anderson	Chef	10

**Memberships : This report shows all memberships details.**

### Membership Details

+ Add Membership

Enter Member ID, Name

Search

Show All

Member Id	Member Name	Points	Account ID
21	joel john	0	31
22	siddhant kamble	0	32
23	shree patil	0	33
24	manasi kumbhar	0	34
25	tejas desai	0	35

**Menu : This report shows all item details in menu.**















### Items Details

+ Add Item

Select Item Type or Item Category

Search

Show All

Item ID	Name	Type	Category	Price	Description	Edit
C1	Kamikaze	Classic Cocktails	Drinks	28.00	cocktails	
C2	Singapore Sling	Classic Cocktails	Drinks	29.00	cocktails	
C3	Long Island Iced Tea	Classic Cocktails	Drinks	29.00	cocktails	
C4	Lady on the Beach	Classic Cocktails	Drinks	29.00	cocktails	
C5	Margarita	Classic Cocktails	Drinks	29.00	cocktails	
C6	Pina Colada	Classic Cocktails	Drinks	26.00	cocktails	
C7	Mojito	Classic Cocktails	Drinks	29.00	cocktails	
CP1	Green Apple	Cold Pressed Juice	Drinks	15.00	fresh pressed juice	
CP2	Orange	Cold Pressed Juice	Drinks	15.00	fresh pressed juice	
CP3	Carrot	Cold Pressed Juice	Drinks	15.00	fresh pressed juice	
CP4	Watermelon	Cold Pressed Juice	Drinks	13.00	fresh pressed juice	
CP5	Pineapple	Cold Pressed Juice	Drinks	13.00	fresh pressed juice	
CP6	Lime	Cold Pressed Juice	Drinks	10.00	fresh pressed juice	
HC1	Shoney Iced Tea	House Cocktails	Drinks	32.00	Gin_Rum	

**Restaurant Tables : This report shows all table details.**

Table Details

+ Add Table

Enter Table ID, Capacity

Search

Show All

Table ID	Capacity	Availability
1	4 Persons	Yes
2	4 Persons	Yes
3	4 Persons	Yes
4	6 Persons	Yes
5	6 Persons	Yes
6	6 Persons	Yes
7	6 Persons	Yes
8	8 Persons	Yes
9	8 Persons	Yes
10	8 Persons	Yes

**Reservation : This report shows all Reservation details.**







Reservation Details

+ Add Reservation

Enter Reservation ID, Customer Name, Reservation Date (2023-09)

Search

Show All

Reservation ID	Customer Name	Table ID	Reservation Time	Reservation Date	Head Count	Special Request	Receipt
1520256	tejas desai	6	15:00:00	2025-03-24	6	make it special	
1820253	shree patil	3	18:00:00	2025-03-22	4	birthday celebration	
1020251	joel john	1	10:00:00	2025-03-22	4	make it special	
1320254	manasi kumbhar	4	13:00:00	2025-03-20	6	family gathering	
1120252	siddhant kamble	2	11:00:00	2025-03-19	4	make my day good	
1620253	joel john	3	16:00:00	2025-03-11	4	sdasd	



# CONCLUSION

The Restaurant Management System developed in this project offers an efficient and structured solution to manage various operations of a restaurant, including staff account management, table reservations, and menu control. By automating these essential functions, the system helps eliminate manual paperwork, reduce human error, and improve overall operational efficiency.

This system is user-friendly, easy to deploy, and relies on open-source technologies such as PHP and MySQL, making it both cost-effective and scalable for small to medium-sized restaurants. Through the use of role-based access, it ensures secure handling of sensitive operations and allows different users (admin and staff) to interact with the system based on their responsibilities. The project successfully demonstrates the potential of web-based applications in transforming traditional restaurant workflows into smart and manageable digital systems.

## **Suggestions**

### **1. Improve Security**

- Implement **password hashing** (e.g., `password_hash()` and `password_verify()`).
- Prevent SQL injection using **prepared statements**.
- Add server-side validation using regular expressions or PHP validation libraries.

### **2. User Experience**

- Enhance UI using Bootstrap or another CSS framework for responsiveness.
- Provide better error/success messages after actions like login, insert, or delete.

### **3. Code Organization**

- Use a proper MVC structure (Model-View-Controller) for better separation of logic.
- Move SQL queries to a separate database handler or ORM.

### **4. Documentation**

- Add comments and a user manual to help new developers or admins understand the flow.
- Include ER diagrams and flowcharts in documentation for clarity.

## **Future Enhancements**

### **1. Online Ordering & Delivery**

- Allow customers to order food online and choose delivery or pickup.
- Integrate Google Maps for delivery address tracking.

### **2. Payment Gateway Integration**

- Support online payments using Stripe, Razorpay, or PayPal.

### **3. Order Tracking**

- Add real-time kitchen order tracking for staff and customers.

### **4. Analytics Dashboard**

- Include visualizations for sales, popular dishes, peak hours, and customer feedback.

### **5. Mobile Compatibility**

- Convert the system into a responsive web app or build a companion mobile app.

### **6. Role-Based Access Control**

- Different levels of access for Admin, Manager, Waiter, and Chef with restricted views and actions.

# Bibliography

1. [www.codewithharry.com](http://www.codewithharry.com)
2. [www.codeproject.com](http://www.codeproject.com)
3. [www.W3school.com](http://www.W3school.com)
4. [www.youtube.com](http://www.youtube.com)