RESTAURANT MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

K.Sri ranjithkumar [211423104643]

K.Veeramani[211423104723]

S.Shathvaran [211423104618]

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

23CS1312-DATABASE MANAGEMENT SYSTEM LAB

NOVEMBER 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report "RESTAURANT MANAGEMENT SYSTEM" is the bonafide work of " K.Sri ranjithkumar [211423104643], K.Veeramani[211423104723],S.Shathvaran [211423104618]" who carried out the project work under my supervision.

Signature of the Supervisor with date

Mrs. A.KANCHANA M.E.,(Ph.D).,
Assistant professor,
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai-123.

ABSTRACT

The restaurant management system is a database-driven application developed to streamline and automate restaurant operations. This project focuses on creating a web-based solution using technologies such as PHP, MySQL, HTML, and CSS, integrated within a local server environment provided by XAMPP. The primary functionality of the system is to manage customer details, display the restaurant's menu, and allow for order placement and tracking.

Key Points:

1. Purpose: Automates restaurant tasks like menu management, customer details, and order processing.
2. Technologies Used: PHP for backend, MySQL for database, HTML/CSS for frontend, XAMPP for local hosting.
3. Core Features:
    - Menu Management: Add, update, and view menu items with prices.
    - Customer Management: Store and view customer details.
    - Order Processing: Place and view customer orders.
4. Benefits:
    - Efficiency: Reduces manual work and speeds up operations.
    - Data Security: Uses MySQL for secure data storage.
    - Customer Satisfaction: Faster service and reduced errors in order handling.
5. Future Scope: Can be expanded to include online ordering, payment integration, and reporting features.

TABLE OF CONTENTS

INTRODUCTION

The rapid growth of the restaurant industry has led to an increased demand for efficient management solutions. Traditional methods of managing customer orders, tracking inventory, and updating menus can become cumbersome and error-prone, especially for larger restaurants. This project introduces a restaurant management system designed to solve these challenges by digitizing restaurant operations and providing a streamlined process for managing customers, orders, and menus.

The primary motivation behind this project is to create a centralized platform where restaurant staff can manage customer data, view the menu, and place or update orders in real-time. The project incorporates a database to store customer information, menu items, and order details, ensuring data consistency and accuracy. By using PHP and MySQL, the system handles server-side operations and allows seamless communication between the user interface and the database. This project helps restaurant owners and staff manage their operations efficiently, improving the overall customer experience.

OBJECTIVE

The objective of the restaurant management system project is to create a database-driven web application that automates and simplifies the process of managing a restaurant's day-to-day operations. The system is designed with the following specific objectives in mind:

1. Customer Management: The system stores and retrieves customer details efficiently. When a customer places an order, their information is saved in the database, and they are assigned a unique customer ID. This data can later be accessed or modified by restaurant staff.
2. Menu Management: The restaurant's menu is stored in the database, where restaurant staff can add, update, or remove menu items. Customers can view the available items and their prices, allowing for seamless order placement.
3. Order Management: The system allows customers to place orders by selecting items from the menu and specifying the quantity. Each order is recorded in the database and associated with a customer ID, allowing staff to track and manage orders.
4. User-Friendly Interface: The system is designed with simplicity in mind, providing a clean and intuitive interface for both staff and customers.

By meeting these objectives, the system will ensure efficient restaurant operations and a smoother customer experience.

SYSTEM REQUIREMENTS

The restaurant management system is built using web technologies that require specific hardware and software environments to run effectively.

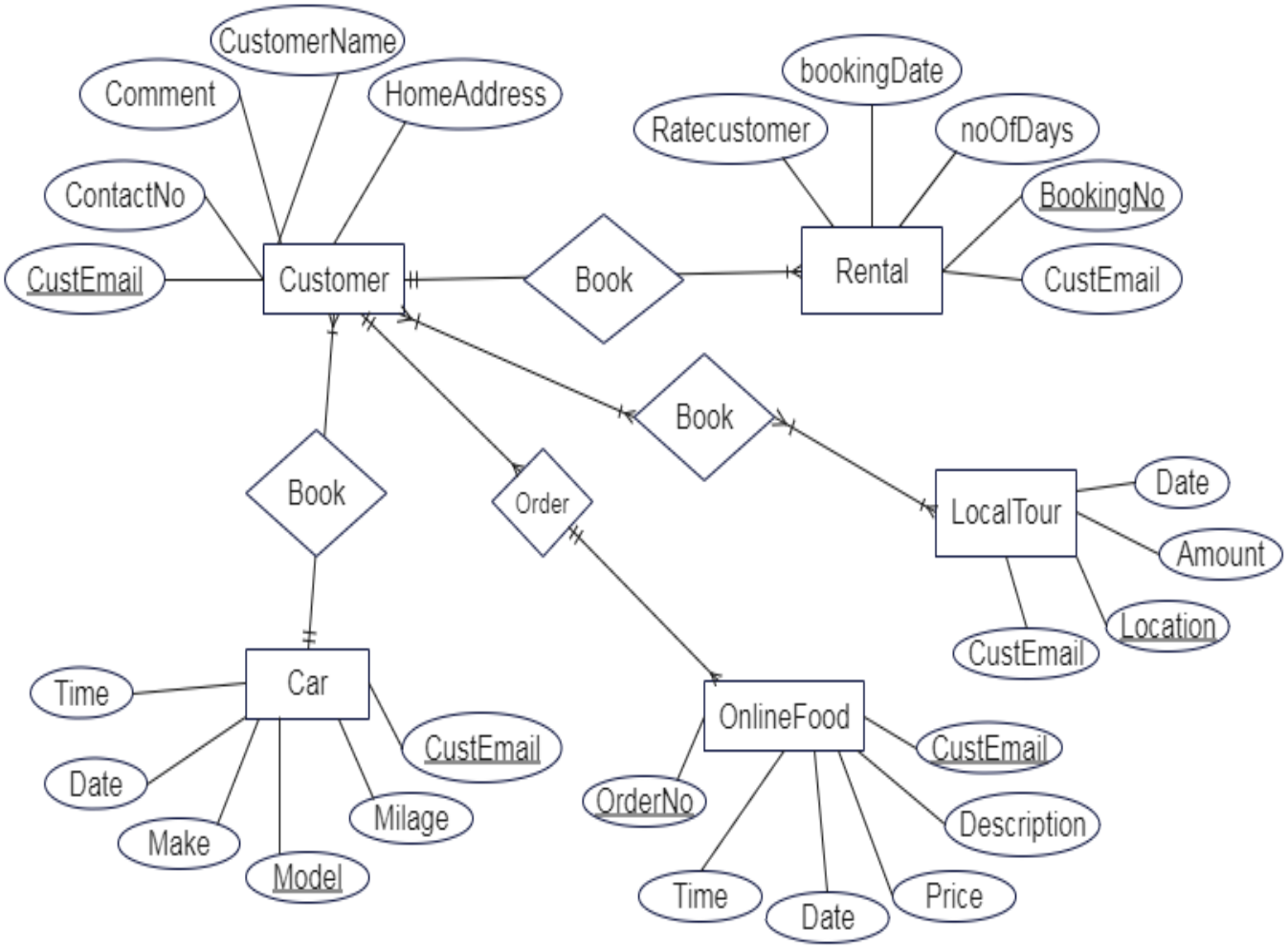
Hardware Requirements:

- Processor: 2 GHz or faster processor.
- RAM: 4 GB minimum (8 GB recommended for better performance).
- Hard Disk Space: At least 1 GB of free space for storing the database and running server-side components.
- Network: Access to a stable internet connection (for remote access) or a local network.

Software Requirements:

- Operating System: Windows 10/11, macOS, or Linux (Ubuntu 18.04+).
- Development Tools: A text editor or integrated development environment (IDE) like Visual Studio Code or Sublime Text for writing code.
- Web Server: XAMPP or WAMP (for running Apache, PHP, and MySQL).
- Database: MySQL (included with XAMPP/WAMP).
- Languages: HTML, CSS, JavaScript for front-end; PHP for back-end logic.
- Browser: Google Chrome, Mozilla Firefox, or any modern web browser for testing the system.

ER-DIAGRAM


CustomerName
Comment
HomeAddress
ContactNo
CustEmail
Customer
Book
Rental
bookingDate
Ratecustomer
noOfDays
BookingNo
CustEmail
Book
Book
Order
LocalTour
Date
Amount
Location
CustEmail
Time
Car
Date
Make
Model
Milage
CustEmail
OnlineFood
OrderNo
Time
Date
Price
Description
CustEmail

IMPLEMENTATION

The implementation of the restaurant management system involves the use of several key technologies to ensure that all objectives are met. The system is built using PHP for the back-end logic, MySQL for the database, and HTML/CSS/JavaScript for the front-end. The entire system is hosted locally using XAMPP, which provides an Apache server for running PHP and a MySQL database for storing data.

Steps for Implementation:

1. Database Creation: Using MySQL, create the necessary tables (customers, menu_items, orders, order_items) to store the data. The database is structured to efficiently manage customer details, menu items, and order records.
2. Front-End Design: HTML and CSS are used to create a user-friendly interface. The interface consists of pages for viewing the menu, adding customers, placing orders, and viewing existing orders. The design ensures simplicity for both the customers and restaurant staff.
3. Back-End Logic: PHP scripts are responsible for handling database queries and logic, including adding new customers, inserting new orders, and fetching data from the database for display on the front-end.
4. Testing: After the system is implemented, it is tested for functionality. Each form is tested to ensure that data is properly inserted into and retrieved from the database.
5. Deployment: The system is deployed locally using XAMPP, allowing it to be accessed on any device connected to the same network. Ngrok can be used to expose the local server

to the internet, enabling remote access.

SOURCE CODE

1.Database Schema (MySQL)

```
-- Create the database
CREATE DATABASE restaurant;
-- Use the newly created database
USE restaurant;
-- Create the customers table
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL,
    phone VARCHAR(15)
);
-- Create the menu_items table
CREATE TABLE menu_items (
    item_id INT AUTO_INCREMENT PRIMARY KEY,
    item_name VARCHAR(100) NOT NULL,
    price DECIMAL(5, 2) NOT NULL
);
```

```
-- Create the orders table
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT NOT NULL,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);

-- Create the order_items table
CREATE TABLE order_items (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    item_id INT NOT NULL,
    quantity INT NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (item_id) REFERENCES menu_items(item_id)
);

-- Insert sample data into menu_items table
INSERT INTO menu_items (item_name, price) VALUES
    ('Margherita Pizza', 499.00),
    ('Pepperoni Pizza', 599.00),
    ('Cheeseburger', 299.00),
```

```
('Grilled Chicken Sandwich', 349.00),
('Caesar Salad', 199.00),
('French Fries', 99.00),
('Coke', 50.00),
('Iced Tea', 60.00);
```

2.PHP CODE:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "restaurant";

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

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

MAIN INTERFACE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Restaurant Management</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      color: #333;
    }
    .container {
      width: 80%;
      margin: 0 auto;
    }
    h1 {
      text-align: center;
      color: #333;
      margin-top: 20px;
```

```
    font-size: 2.5rem;
}
form, .links, .menu {
    background-color: #fff;
    padding: 20px;
    margin: 20px 0;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
input, button {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    font-size: 1rem;
}
button {
    background-color: #28a745;
    color: white;
    border: none;
    cursor: pointer;
```

```
}
button:hover {
    background-color: #218838;
}
table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
}
table, th, td {
    border: 1px solid #ddd;
}
th, td {
    padding: 12px;
    text-align: center;
}
th {
    background-color: #f4f4f4;
}
.menu h2 {
```

```
            text-align: center;
            font-size: 1.8rem;
            color: #333;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Restaurant Management System</h1>

        <!-- Display Menu Items -->
        <div class="menu">
            <h2>Menu</h2>
            <table>
                <thead>
                    <tr>
                        <th>Item ID</th>
                        <th>Item Name</th>
                        <th>Price (₹)</th>
                    </tr>
```

```
            </thead>
            <tbody>
                <?php
                include 'db_connect.php';
                $sql = "SELECT * FROM menu_items";
                $result = $conn->query($sql);

                if ($result->num_rows > 0) {
                    while ($row = $result->fetch_assoc()) {
                        echo "<tr>";
                        echo "<td>" . $row['item_id'] . "</td>";
                        echo "<td>" . $row['item_name'] . "</td>";
                        echo "<td>₹" . number_format($row['price'], 2) . "</
td>";
                        echo "</tr>";
                    }
                } else {
                    echo "<tr><td colspan='3'>No items found in the
menu</td></tr>";
                }
```

```
            $conn->close();
            ?>
          </tbody>
        </table>
      </div>

      <!-- Add Customer Form -->
      <h2>Add Customer</h2>
      <form id="customerForm" action="add_customer.php"
method="POST">
        <input type="text" name="name" placeholder="Customer
Name" required>
        <input type="email" name="email" placeholder="Customer
Email" required>
        <input type="text" name="phone" placeholder="Phone
Number">
        <button type="submit">Add Customer</button>
      </form>

      <!-- Place Order Form -->
      <h2>Place Order</h2>
      <form id="orderForm" action="add_order.php"
```

```
method="POST">
        <input type="number" name="customer_id"
placeholder="Customer ID" required>

        <h3>Items</h3>
        <input type="text" name="items[0][item_id]"
placeholder="Item ID" required>
        <input type="number" name="items[0][quantity]"
placeholder="Quantity" required>
        <button type="submit">Place Order</button>
    </form>

    <!-- Links to View Data -->
    <div class="links">
        <h2>View Data</h2>
        <a href="get_customers.php" target="_blank">View
Customers</a>
        <a href="get_orders.php" target="_blank">View Orders</a>
    </div>
  </div>

  <script>
```

```
        document.getElementById('customerForm').onsubmit =
function() {
            alert('Customer added successfully!');
            this.reset();
            return false;
        };

        document.getElementById('orderForm').onsubmit = function()
{
            alert('Order placed successfully!');
            this.reset();
            return false;
        };
    </script>
</body>
</html>
```

Add Customer Logic :

```
<?php
include 'db_connect.php';

$name = $_POST['name'];
$email = $_POST['email'];
$phone = $_POST['phone'];

$sql = "INSERT INTO customers (name, email, phone) VALUES ('$name',
'$email', '$phone')";

if ($conn->query($sql) === TRUE) {
   echo "Customer added successfully!";
} else {
   echo "Error: " . $sql . "<br>" . $conn->error;
}

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

Place Order Logic :

```
<?php
include 'db_connect.php';
```

```
$customer_id = $_POST['customer_id'];
$items = $_POST['items'];

$sql = "INSERT INTO orders (customer_id) VALUES ($customer_id)";
if ($conn->query($sql) === TRUE) {
    $order_id = $conn->insert_id;

    foreach ($items as $item) {
        $item_id = $item['item_id'];
        $quantity = $item['quantity'];
        $sql_item = "INSERT INTO order_items (order_id, item_id, quantity)
VALUES ($order_id, $item_id, $quantity)";
        $conn->query($sql_item);
    }

    echo "Order placed successfully!";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

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

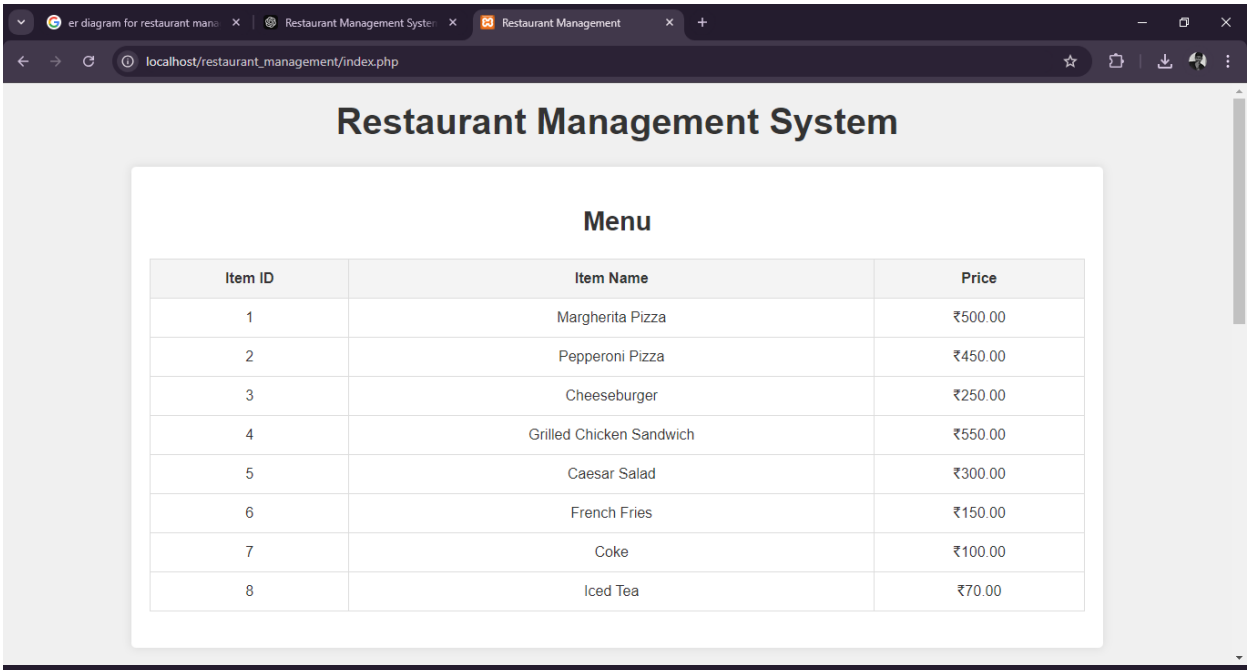
View Customers :

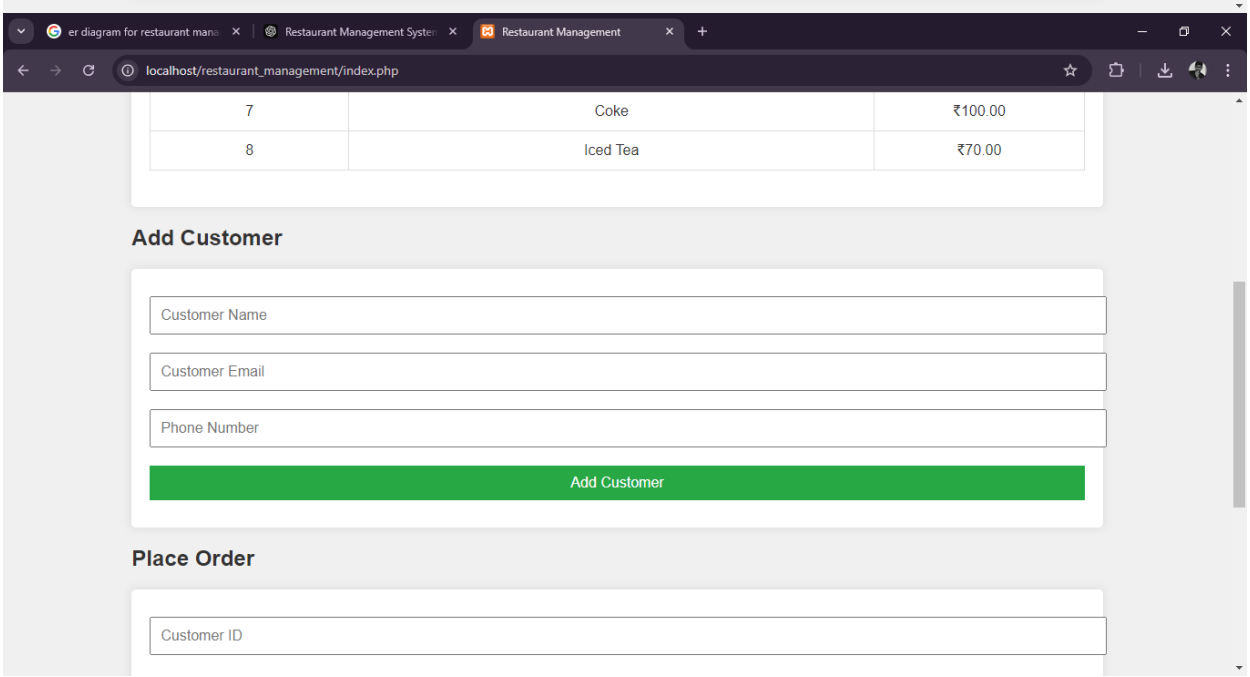
```
<?php
include 'db_connect.php';


$sql = "SELECT * FROM customers";
$result = $conn->query($sql);


if ($result->num_rows > 0) {
  echo "<table><tr><th>Customer ID</th><th>Name</th><th>Email</
th><th>Phone</th></tr>";
  while ($row = $result->fetch_assoc()) {
    echo "<tr><td>" . $row['customer_id'] . "</td><td>" . $row['name'] . "</
td><td>" . $row['email'] . "</td><td>" . $row['phone'] . "</td></tr>";
  }
  echo "</table>";
} else {
  echo "No customers found";
}


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

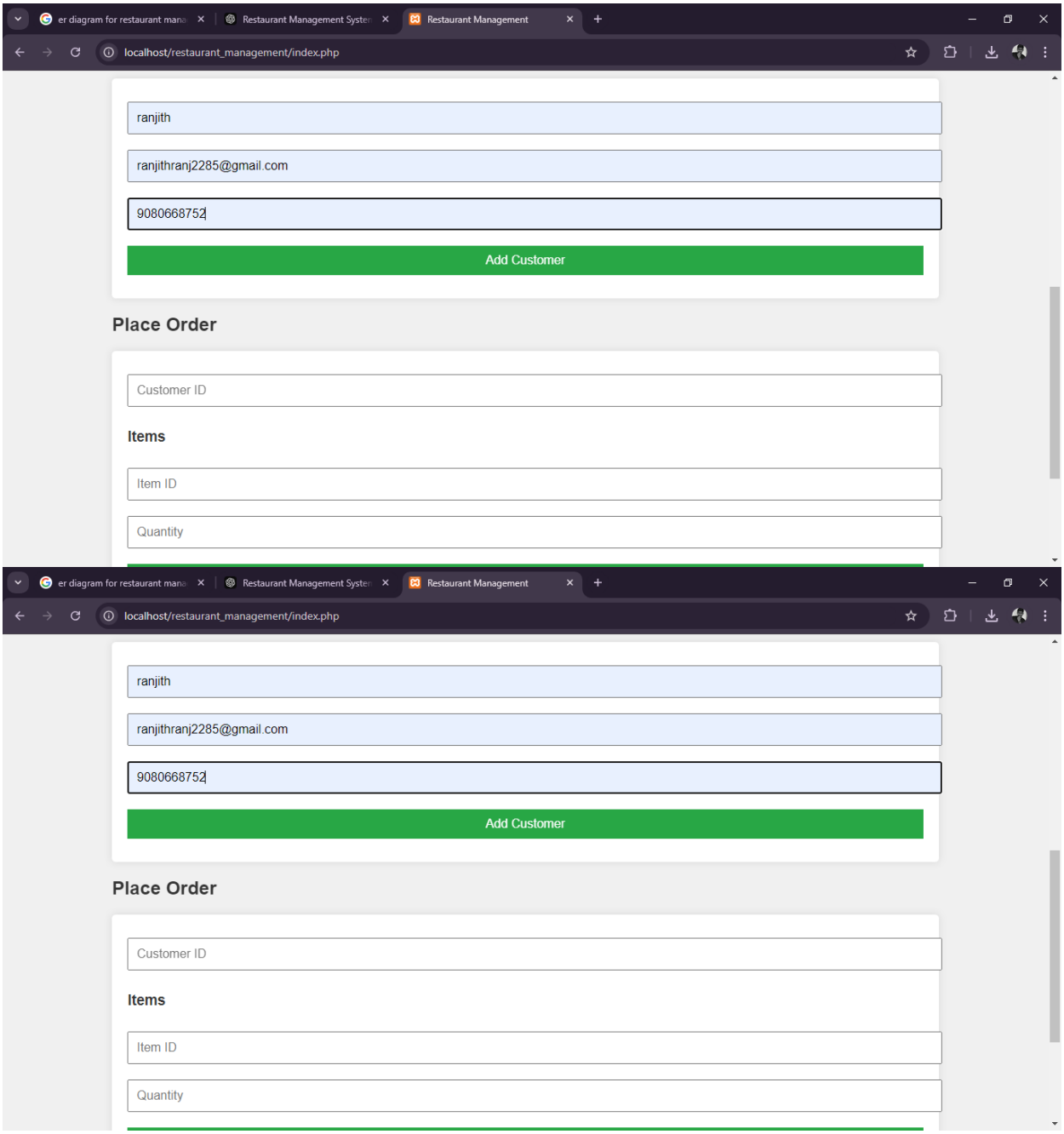
View Orders :

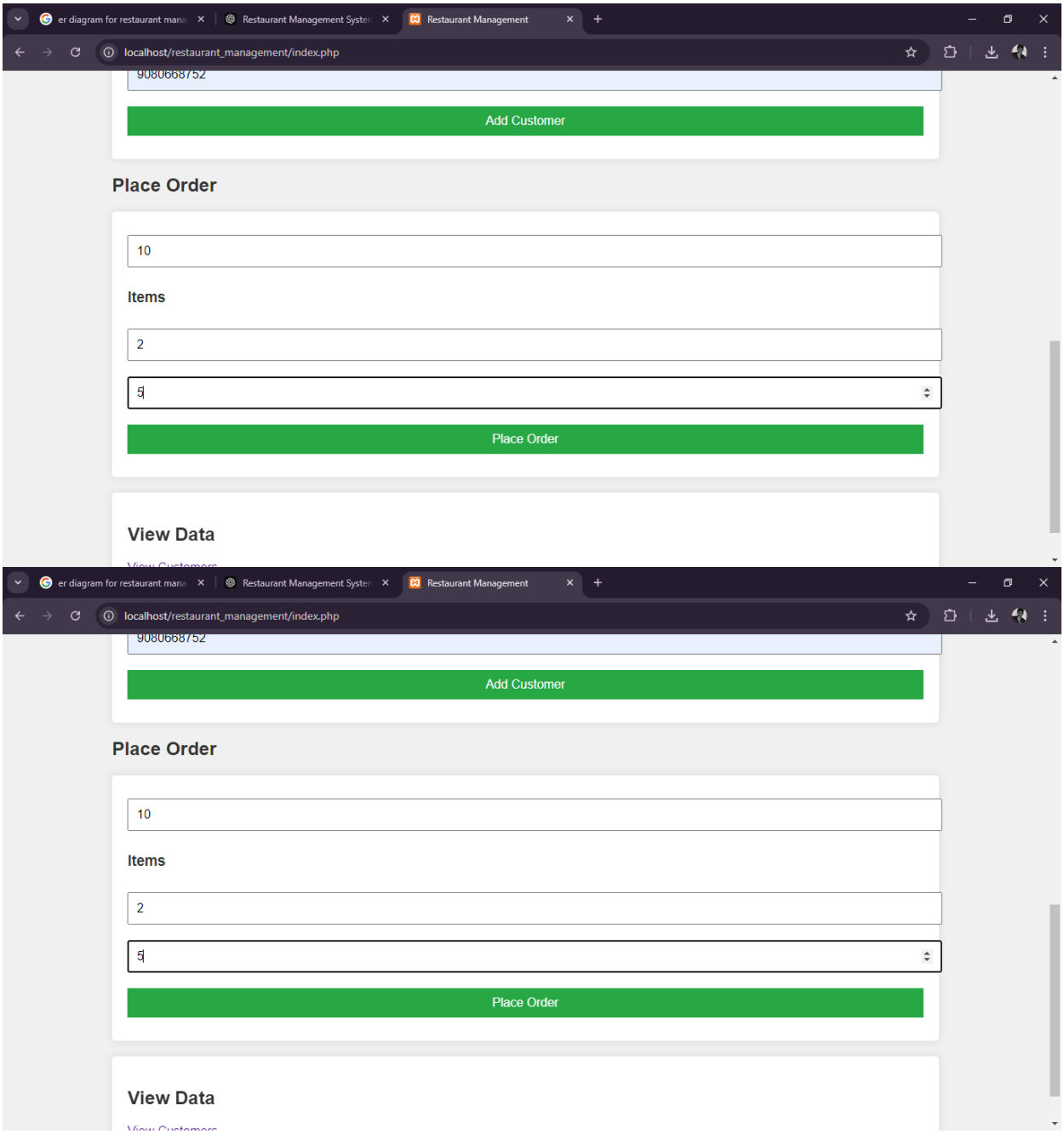
```
<?php
include 'db_connect.php';

$sql = "SELECT orders.order_id, customers.name, orders.order_date FROM
orders JOIN customers ON orders.customer_id = customers.customer_id";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
   echo "<table><tr><th>Order ID</th><th>Customer Name</th><th>Order
Date</th></tr>";
   while ($row = $result->fetch_assoc()) {
      echo "<tr><td>" . $row['order_id'] . "</td><td>" . $row['name'] . "</td><td>" .
$row['order_date'] . "</td></tr>";
   }
   echo "</table>";
} else {
   echo "No orders found";
}

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

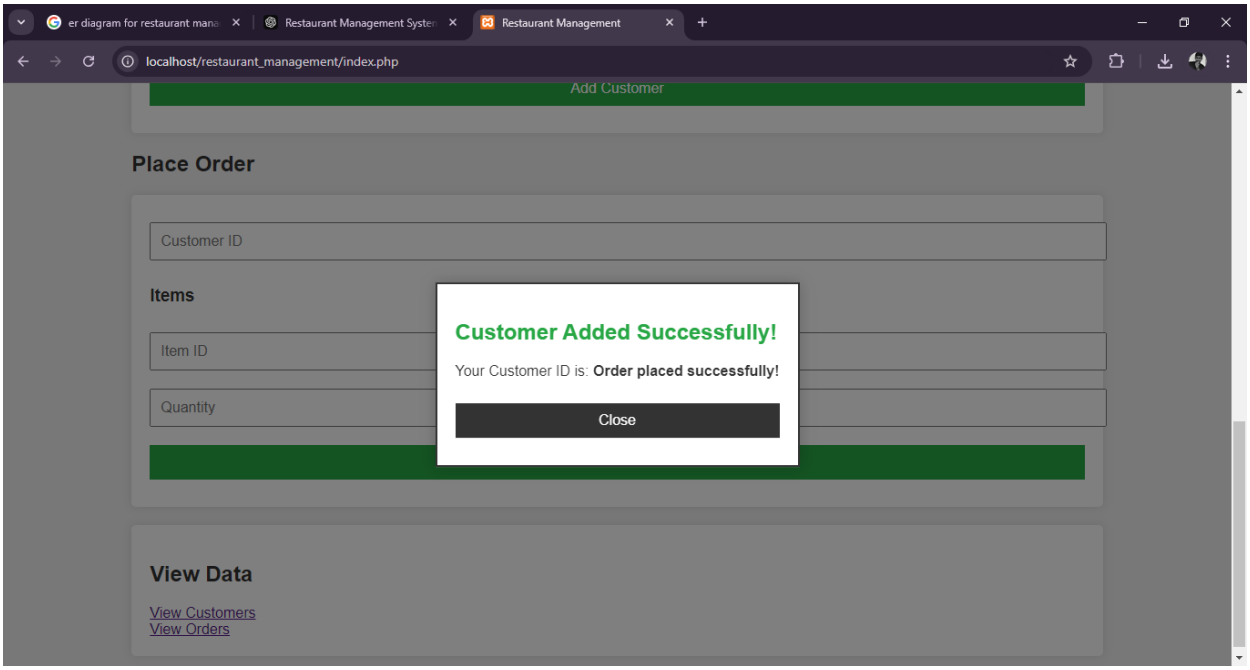
SCREENSHOTS

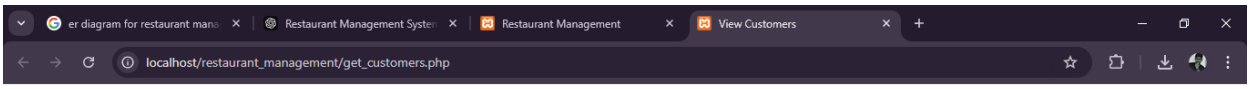





er diagram for restaurant mana
Restaurant Management Syster
Restaurant Management
localhost/restaurant_management/index.php
ranjith
ranjithranj2285@gmail.com
9080668752
Add Customer
Place Order
Customer ID
Items
Item ID
Quantity

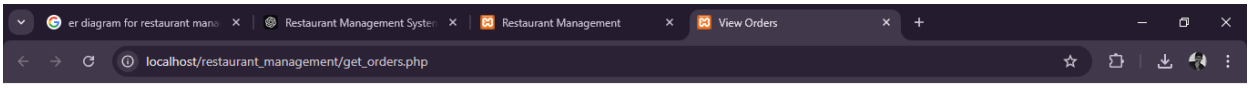

er diagram for restaurant mana
Restaurant Management Syster
Restaurant Management
localhost/restaurant_management/index.php
9080668752
Add Customer
Place Order
10
Items
2
5
Place Order
View Data
View Customers
er diagram for restaurant mana
Restaurant Management Syster
Restaurant Management
localhost/restaurant_management/index.php
9080668752
Add Customer
Place Order
10
Items
2
5
Place Order
View Data
View Customers





Customers List

Customer ID	Name	Email	Phone
9	ranjith	ranjithranj2285@gmail.com	9080668752
10	ranjith	ranjithranj2285@gmail.com	9080668752



Orders List

Order ID	Customer ID	Item ID	Quantity
6	9	2	1
7	10	2	5

CONCLUSION

The restaurant management system project demonstrates the use of a database management system to automate and simplify restaurant operations. By implementing a relational database in MySQL and integrating it with a web interface using PHP and HTML, we successfully created a functional system that manages customer details, menu items, and orders. The system improves the efficiency of restaurant management by providing a centralized platform for handling orders and customer data.

This project highlights the importance of a well-structured database in managing large volumes of data and ensuring data consistency. Through this project, we gained valuable experience in designing and implementing database-driven web applications, reinforcing the importance of DBMS in real-world applications. Going forward, this system can be expanded to include features like online payments, inventory management, and analytics to further enhance restaurant operations.