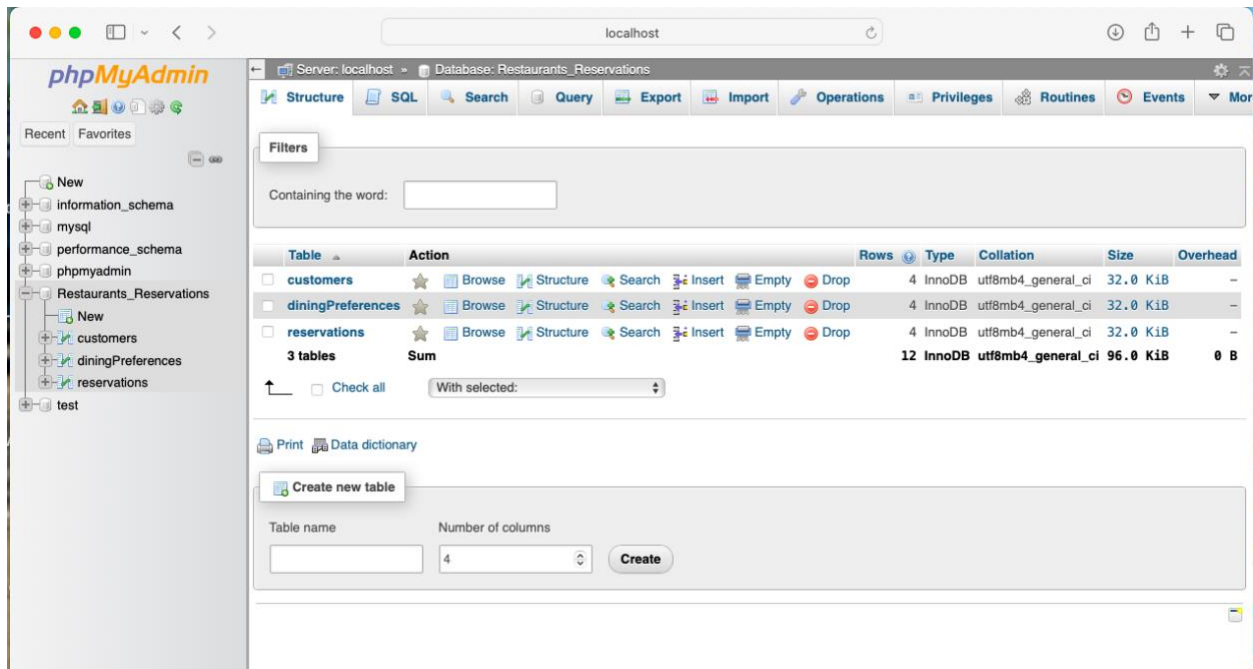


NAME: JAMES OBOUR

DATE: December 10, 2024

CIS 344 FINAL PROJECT REPORT: RESTAURANT RESERVATION MySQL

Starting with the MySQL component of the project, I created the restaurant_reservations database and designed three tables: customers, diningpreferences, and reservations. Each table was structured with relevant attributes and associated foreign keys. Afterward, I populated the tables with sample data to validate their initialization and ensure the entries were correctly inserted and visible, as shown below.



The screenshot displays the phpMyAdmin web interface for a MySQL database named 'Restaurants_Reservations' on a localhost server. The left sidebar shows the database structure, including a 'New' button and a list of databases: information_schema, mysql, performance_schema, phpmyadmin, and Restaurants_Reservations. The main panel shows the 'Structure' tab for the 'Restaurants_Reservations' database. It lists three tables: 'customers', 'diningPreferences', and 'reservations'. Each table has 4 rows, is of type 'InnoDB', and uses the 'utf8mb4_general_ci' collation. The total size of the tables is 96.0 KiB. Below the table list, there is a 'Create new table' button and a form to create a new table with a name and number of columns.

Table	Action	Rows	Type	Collation	Size	Overhead
customers	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 K	1B
diningPreferences	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 K	1B
reservations	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 K	1B
3 tables	Sum	12	InnoDB	utf8mb4_general_ci	96.0 K	3B

CUSTOMERS

Server: localhost » Database: Restaurants_Reservations » Table: customers

Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

`SELECT * FROM `customers``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	customerid	customerName	contactInfo
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Frank Osei	osei10@gmail.com
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	James Obour	obour00@gmail.com
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Roland Owusu	Roland13@gmail.com
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Salomey Nuamah	salonuaham100@gmail.com

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

DININGPREFERENCES

Server: localhost » Database: Restaurants_Reservations » Table: diningPreferences

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

`SELECT * FROM `diningPreferences``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

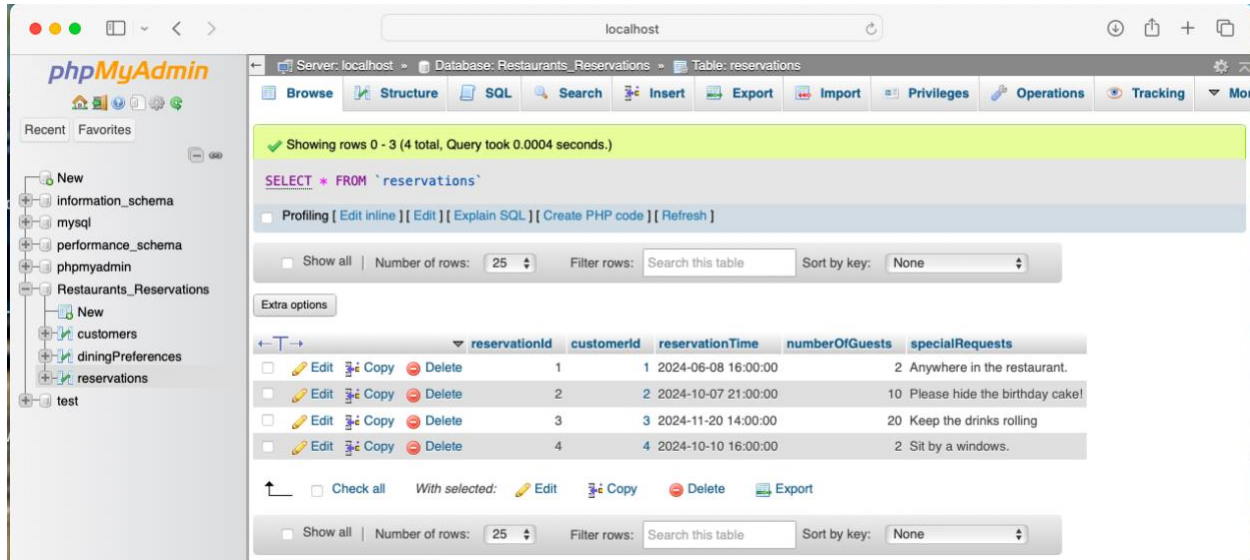
Extra options

	preferenceld	customerid	favoriteTable	dietaryRestrictions
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	Table 3	No alcohol on any food.
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	Table 10	None
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	Table 5	No pork
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	4	Table 7	No sea food

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

RESERVATIONS



Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

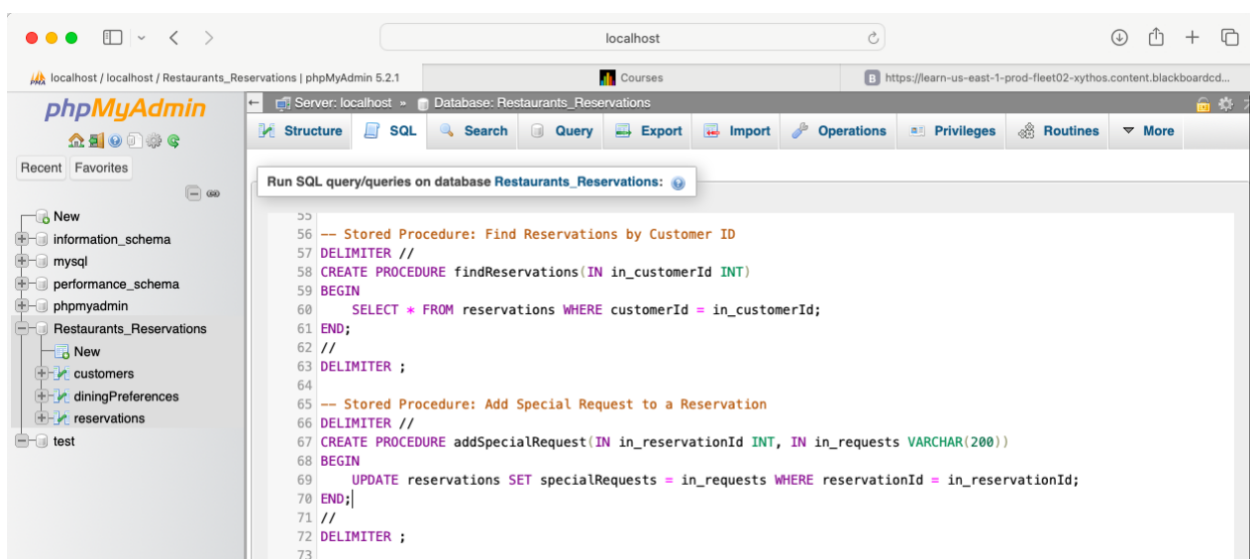
```
SELECT * FROM `reservations`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	reservationId	customerId	reservationTime	numberOfGuests	specialRequests
<input type="checkbox"/> Edit Copy Delete	1	1	2024-06-08 16:00:00	2	Anywhere in the restaurant.
<input type="checkbox"/> Edit Copy Delete	2	2	2024-10-07 21:00:00	10	Please hide the birthday cake!
<input type="checkbox"/> Edit Copy Delete	3	3	2024-11-20 14:00:00	20	Keep the drinks rolling
<input type="checkbox"/> Edit Copy Delete	4	4	2024-10-10 16:00:00	2	Sit by a windows.

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

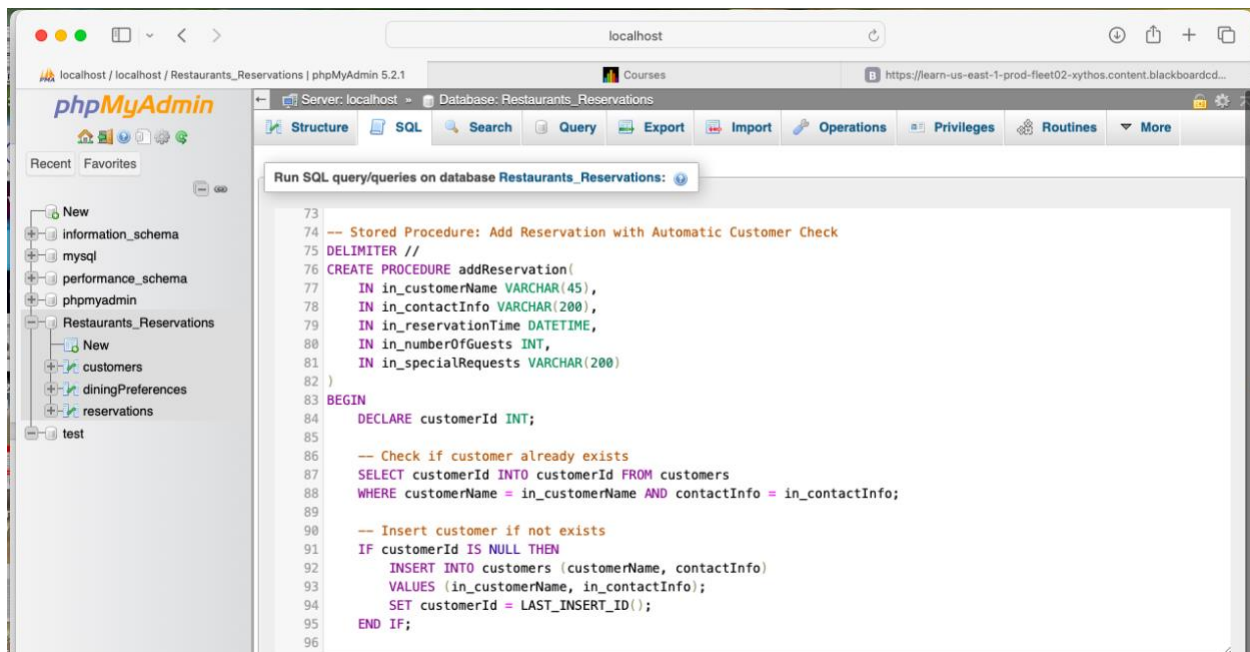
Next, I created two stored procedures: findReservations and addSpecialRequest. The findReservations procedure uses an existing customer ID to retrieve the corresponding reservation, while the addSpecialRequest procedure utilizes an existing reservation to add a special request.



```
-- Stored Procedure: Find Reservations by Customer ID
DELIMITER //
CREATE PROCEDURE findReservations(IN in_customerId INT)
BEGIN
    SELECT * FROM reservations WHERE customerId = in_customerId;
END;
//
DELIMITER ;

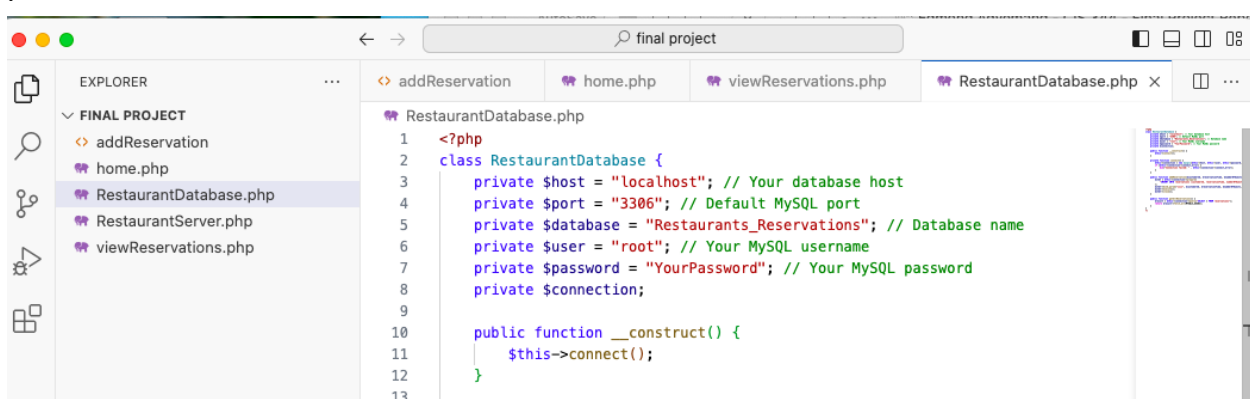
-- Stored Procedure: Add Special Request to a Reservation
DELIMITER //
CREATE PROCEDURE addSpecialRequest(IN in_reservationId INT, IN in_requests VARCHAR(200))
BEGIN
    UPDATE reservations SET specialRequests = in_requests WHERE reservationId = in_reservationId;
END;
//
DELIMITER ;
```

The next stored procedure is the addReservation procedure. This procedure first checks whether the customer ID exists. If the customer ID is not found, a new customer ID is created before proceeding to make the reservation.



```
73
74 -- Stored Procedure: Add Reservation with Automatic Customer Check
75 DELIMITER //
76 CREATE PROCEDURE addReservation(
77     IN in_customerName VARCHAR(45),
78     IN in_contactInfo VARCHAR(200),
79     IN in_reservationTime DATETIME,
80     IN in_numberOfGuests INT,
81     IN in_specialRequests VARCHAR(200)
82 )
83 BEGIN
84     DECLARE customerId INT;
85
86     -- Check if customer already exists
87     SELECT customerId INTO customerId FROM customers
88     WHERE customerName = in_customerName AND contactInfo = in_contactInfo;
89
90     -- Insert customer if not exists
91     IF customerId IS NULL THEN
92         INSERT INTO customers (customerName, contactInfo)
93         VALUES (in_customerName, in_contactInfo);
94         SET customerId = LAST_INSERT_ID();
95     END IF;
96
```

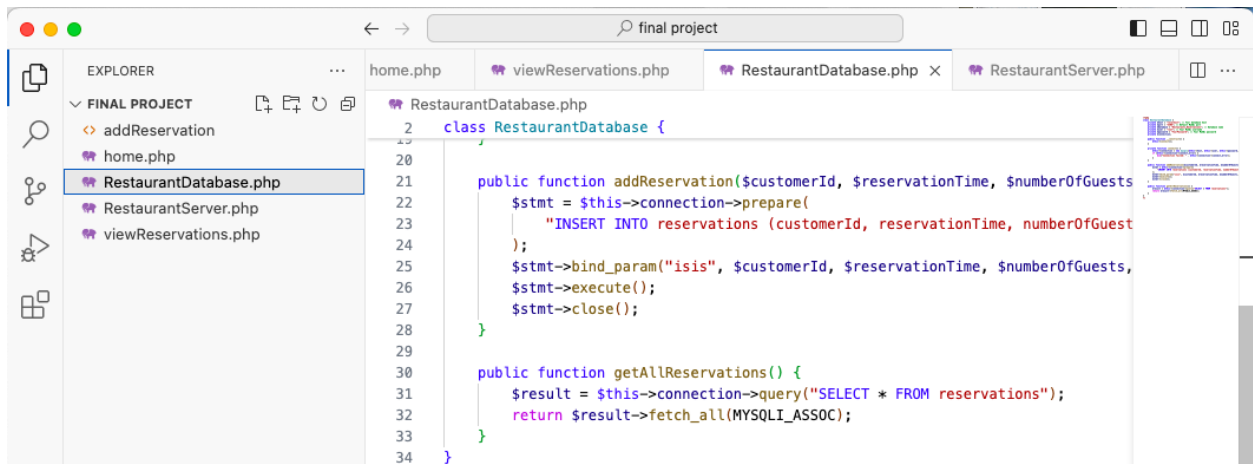
PHP: I started off by modifying the restaurantDatabase.php file with the appropriate MySQL Credentials:



```
1 <?php
2 class RestaurantDatabase {
3     private $host = "localhost"; // Your database host
4     private $port = "3306"; // Default MySQL port
5     private $database = "Restaurants_Reservations"; // Database name
6     private $user = "root"; // Your MySQL username
7     private $password = "YourPassword"; // Your MySQL password
8     private $connection;
9
10    public function __construct() {
11        $this->connect();
12    }
13
```

To first communicate with MySQL, I used the same port number database that's in MySQL so php can retrieve the information.

Next, I added methods in php to do various tasks such as adding and viewing reservations, Adding customer. Each method in this portion of the php code is similar except for changing the “query” portion to each method's respective statements that correspond to the statements in MySQL.

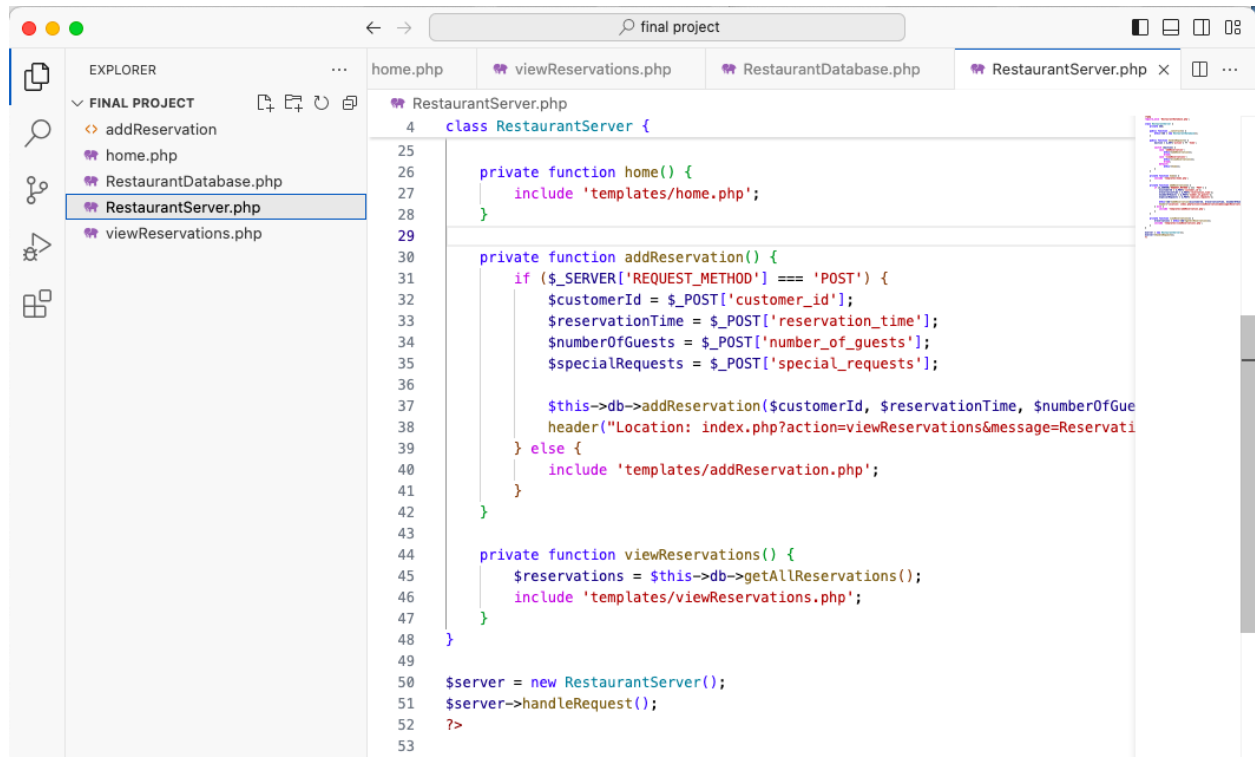


```
2 class RestaurantDatabase {
20
21 public function addReservation($customerId, $reservationTime, $numberOfGuests
22 {
23     $stmt = $this->connection->prepare(
24         "INSERT INTO reservations (customerId, reservationTime, numberOfGuests
25     );
26     $stmt->bind_param("isis", $customerId, $reservationTime, $numberOfGuests,
27     $stmt->execute();
28     $stmt->close();
29 }
30
31 public function getAllReservations() {
32     $result = $this->connection->query("SELECT * FROM reservations");
33     return $result->fetch_all(MYSQLI_ASSOC);
34 }
```

Some of the additional functions I implemented are `getCustomerPreference`, `addSpecialRequest`, `findReservation`, and `deleteReservation`.

Now in the `restuarantServer.php` file

Mainly I called the methods from the `restuarantDatabase.php` and made sure it was visible on the local host web page. For adding reservation, in `do_POST` I initialize the variables from the reservation table and then call the method to add a new reservation from the database file



```
25 class RestaurantServer {
26
27     private function home() {
28         include 'templates/home.php';
29     }
30
31     private function addReservation() {
32         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
33             $customerId = $_POST['customer_id'];
34             $reservationTime = $_POST['reservation_time'];
35             $numberOfGuests = $_POST['number_of_guests'];
36             $specialRequests = $_POST['special_requests'];
37
38             $this->db->addReservation($customerId, $reservationTime, $numberOfGuests, $specialRequests);
39             header("Location: index.php?action=viewReservations&message=Reservation added successfully");
40             include 'templates/addReservation.php';
41         }
42     }
43
44     private function viewReservations() {
45         $reservations = $this->db->getAllReservations();
46         include 'templates/viewReservations.php';
47     }
48 }
49
50 $server = new RestaurantServer();
51 $server->handleRequest();
52 ?>
53
```

PLATFORM

localhost/final_project/index.php?action=viewReservations

All Reservations

Reservation ID	Customer ID	Reservation Time	Number of Guests	Special Requests
1	1	2024-06-08 16:00:00	2	Anywhere in the restaurant.
2	2	2024-10-07 21:00:00	10	Please hide the birthday cake!
3	3	2024-11-20 14:00:00	20	Keep the drinks rolling
4	4	2024-10-10 16:00:00	2	Sit by a windows.

[Back to Home](#)

localhost/final_project/index.php?action=addReservation

Add Reservation

Customer ID:

12/12/2024, 08:57 PM

5

none

Submit

[Back to Home](#)

Restaurant Reservation

localhost/final%20project/home.php

Welcome to the Restaurant Revervation

Manage reservations, customers, and preferences easily.

[Add a Reservation](#)

[View All Reservations](#)

[Add a New Customer](#)

[Manage Customer Preferences](#)