

# *ONLINE RESTAURANT RESERVATION SYSTEM*



# *PRES*ENTOR:



**Fatima sabir**



**Laiba Rani**



**Malaika Arooj**



**Sania Shafique**

# *Introduction*

- The restaurant industry is increasingly adopting digital solutions
- Customers prefer convenient, fast, and hassle-free reservations
- Online systems streamline bookings, reduce errors, and enhance customer satisfaction



# *PURPOSE OF THE PROJECT*

- To develop an online platform for customers to book tables easily
- Reduce manual effort by restaurant staff
- Manage reservations efficiently and reduce overbooking issues
- Improve customer experience and satisfaction





# *WHY I CHOOSE THIS*

- Identified a common problem in restaurant table bookings
- Offline reservations often lead to mismanagement and long wait times
- Aim to provide a digital, user-friendly solution
- Opportunity to turn theoretical knowledge into a real-world application

# SYSTEM OVERVIEW

- Modules:
- Customer Sign Up/Login
- Table Reservation
- Admin Dashboard
- View Bookings
- Feedback System





## *FEATURE OF THE SYSTEM*

- Real-time table availability checking
- Admin panel for managing reservations
- Clean and responsive UI
- Suggestion box for customer feedback

# TOOL AND TECHNOLOGIES USED



- Frontend: HTML, CSS, JavaScript
- Backend: PHP
- Database: MySQL
- Platform: XAMPP (Apache + MySQL)
- Others: VS Code, phpMyAdmin

# 'DATABASE DESIGN

Tables:

- Users(created\_at ,new, name, email, password)
- Reservations(date,time,email,guest,na me,phone no,time)
- Table\_Info(capacity,status,table\_id)
- (id, user\_id, message)



# PROJECT SCREENSHOT

10

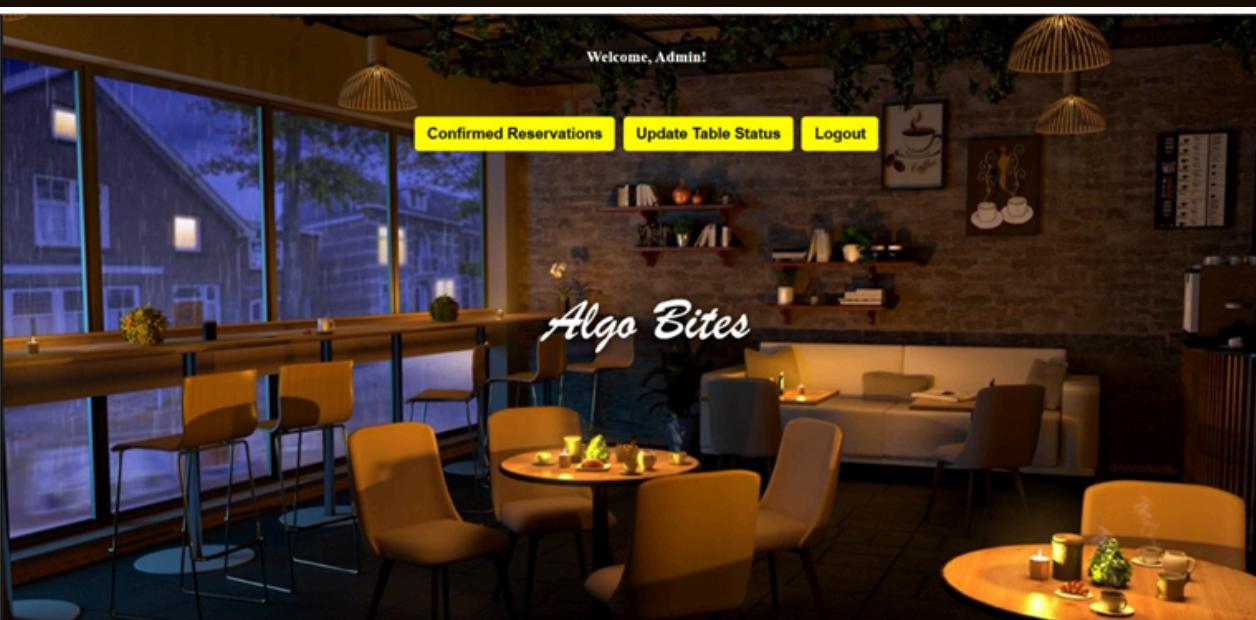
USER PANEL:



RESERVATION FORM:

A screenshot of a reservation form modal. The title is "Make a Reservation". It contains fields for Name (demo), Email (demo@gmail.com), Phone (0312345678), Date (08 / 06 / 2025), Time (84 : 35), Guests (7), and Table No. (1). There are "Submit Reservation" and "Close" buttons at the bottom.

ADMIN PANEL:



CONFIRM RESERVATION:

Confirmed Reservations								
ID	Name	Email	Phone	Date	Time	Guests	Table No.	
10	SANIA SHAFIQUE	zerjan@gmail.com	03195750296	2025-05-16	23:59:00	4	4	
7	laiba	laiba@gmail.com	033254671	2025-05-21	08:26:00	3	5	
2	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-23	17:12:00	2	5	
3	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-23	17:12:00	2	5	
5	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-23	17:40:00	2	1	
6	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-27	10:15:00	2	2	
9	SANIA SHAFIQUE	zerjan@gmail.com	03195750296	2025-05-28	02:00:00	1	4	
8	Zaheer Ahmed	zerjan@gmail.com	03213846436	2025-05-28	03:00:00	4	2	
11	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-30	16:50:00	8	45	
12	SANIA SHAFIQUE	abc@gmail.com	03195750296	2025-05-31	03:55:00	3	2	
13	abc	abc@gmail.com	1234567	2025-05-31	05:00:00	5	2	
4	SANIA SHAFIQUE	saniashafique278@gmail.com	03195750296	2025-05-31	17:28:00	3	5	
14	demo	demo@gmail.com	0312345678	2025-06-08	04:35:00	7	1	

# Sign in:

```

login.php
36 if ($conn->connect_error) {
37     die("Connection failed: " . $conn->connect_error);
38 }
39 // Prepare statement to prevent SQL injection
40 $stmt = $conn->prepare("SELECT email, password FROM users WHERE email = ?");
41 if (!$stmt) {
42     // Prepare failed
43     $conn->close();
44     header(header: "Location: index.html?error=1");
45     exit();
46 }
47
48 $stmt->bind_param(types: "s", var: &$email);
49 $stmt->execute();
50 $stmt->store_result();
51
52 if ($stmt->num_rows === 1) {
53     // Bind result variables
54     $stmt->bind_result(var: &$db_email, vars: &$db_password_hash);
55     $stmt->fetch();
56
57     // Verify password using password_verify()
58     if (password_verify(password: $password, hash: $db_password_hash)) {
59         // Password is correct
60         $_SESSION['user'] = 'user';
61         $_SESSION['email'] = $db_email;
62     }
}

```

# Sign up:

```

signup.php
20 if ($_SERVER["REQUEST_METHOD"] == "POST") {
21     $password = trim(string: $_POST[password]);
22
23     // Validate input
24     if (empty($name) || empty($email) || empty($password)) {
25         echo json_encode(value: ['status' => 'error', 'message' => 'All fields are required.']);
26     } else {
27         // Check for existing email
28         $checkEmail = $conn->prepare(query: "SELECT id FROM users WHERE email = ?");
29         $checkEmail->bind_param(types: "s", var: &$email);
30         $checkEmail->execute();
31         $checkEmail->store_result();
32
33         if ($checkEmail->num_rows > 0) {
34             echo json_encode(value: ['status' => 'error', 'message' => 'Email already exists. Please use a different one.']);
35         } else {
36             // Hash the password
37             $hashedPassword = password_hash(password: $password, algo: PASSWORD_DEFAULT);
38
39             // Insert user
40             $stmt = $conn->prepare(query: "INSERT INTO users (name, email, password) VALUES (?, ?, ?)");
41             $stmt->bind_param(types: "sss", var: &$name, vars: &$email, $hashedPassword);
42
43             if ($stmt->execute()) {
44                 echo json_encode(value: ['status' => 'success', 'message' => 'Signup successful!']);
45             } else {
46
47
}
}

```

## Confirmed reservations:

```

confirmed_reservations.php
1  <?php
2  // Database configuration
3  $host = 'localhost';
4  $dbname = 'algo_bites';
5  $username = 'root';
6  $password = '';
7
8 try {
9     $conn = new PDO(dsni: "mysql:host=$host;dbname=$dbname", username: $username, password: $password);
10    // Set error mode
11    $conn->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
12
13    // Query all confirmed reservations (you can add WHERE confirmed=1 if you have such a field)
14    $stmt = $conn->prepare(query: "SELECT * FROM reservations ORDER BY date, time");
15    $stmt->execute();
16
17    $reservations = $stmt->fetchAll(mode: PDO::FETCH_ASSOC);
18 } catch (PDOException $e) {
19     die("Connection failed: " . $e->getMessage());
20 }
21 ?>
22
23 <!DOCTYPE html>
24 <html lang="en">
25 <head>
26     <meta charset="UTF-8">
27     <title>Confirmed Reservations</title>

```

## Update table info:

```

update_table_info.php
1 if ($_SERVER["REQUEST_METHOD"] == "POST") {
2     // Check if all required fields are provided
3     if (isset($_POST['table_id'], $_POST['capacity'], $_POST['status'])) {
4         $table_id = intval(value: $_POST['table_id']);
5         $capacity = intval(value: $_POST['capacity']);
6         $status = $conn->real_escape_string(string: trim(string: $_POST['status']));
7
8         // Check if the table exists
9         $sql_check = "SELECT * FROM table_info WHERE table_id = $table_id";
10        $result = $conn->query(query: $sql_check);
11
12        if ($result && $result->num_rows > 0) {
13            // Update existing table
14            $sql_update = "UPDATE table_info SET capacity = $capacity, status = '$status' WHERE table_id = $table_id";
15            if ($conn->query(query: $sql_update) === TRUE) {
16                echo "Table info updated successfully.";
17            } else {
18                echo "Error updating record: " . $conn->error;
19            }
20        } else {
21            // Table does not exist
22            echo "Error: Table ID $table_id does not exist. Only updates are allowed.";
23        }
24    } else {
25        echo "Missing required form fields.";
26    }
27 }

```

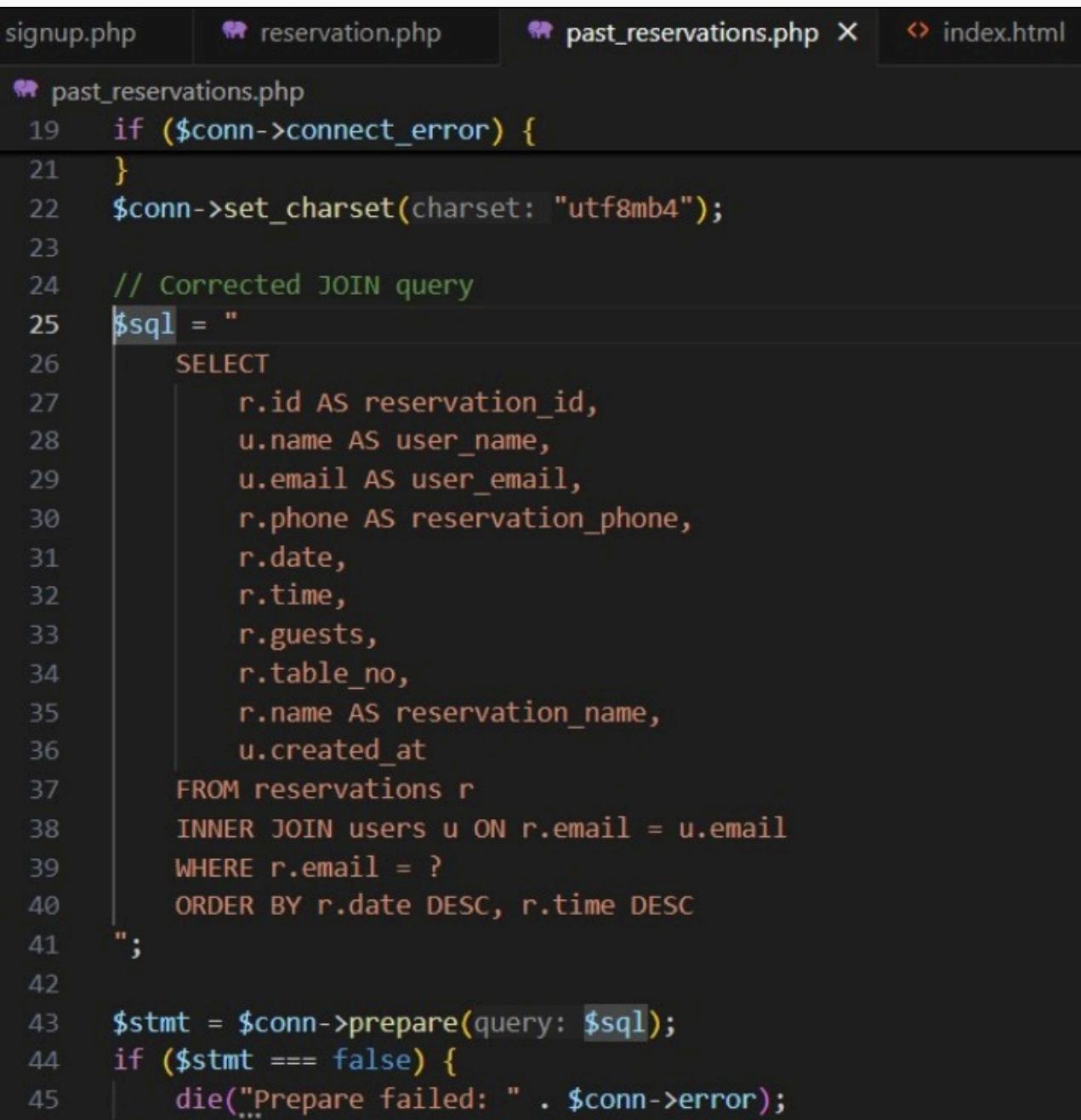
# *Reservations:*

```
reservation.php X past_reservations.php index.html login.php admin.html login.js

reservation.php
57 if ($guests < 1 || $guests > 20) {
58     http_response_code(response_code: 400);
59     exit("X Number of guests must be between 1 and 20.");
60 }
61
62 // Set charset
63 $conn->set_charset(charset: "utf8mb4");
64
65 // Check if preferred table is available
66 $check_stmt = $conn->prepare(query: "SELECT status FROM table_info WHERE table_id = ? LIMIT 1");
67 if (!$check_stmt) {
68     http_response_code(response_code: 500);
69     exit("X Database prepare error: " . $conn->error);
70 }
71
72 $check_stmt->bind_param(types: "i", var: &$table);
73 $check_stmt->execute();
74 $check_stmt->store_result();
75
76 if ($check_stmt->num_rows === 0) [
77     http_response_code(response_code: 400);
78     exit("X The selected table does not exist.");
79 ]
80
81 $check_stmt->bind_result(var: &$status);
82 $check_stmt->fetch();
```

```
reservation.php ✘ post_reservations.php ✘ index.html ✘ login.php ✘ adminhtml ✘ login.p  
update_table_info.php ✘ confirmed.r  
  
reservation.php  
/  
8  
9     $check_start->close();  
0  
1 // Insert reservation  
2 $stmt = $conn->prepare(query: "INSERT INTO reservations (name, email, phone, date, time, guests, table_no) VALUES (?, ?, ?, ?, ?, ?, ?)");  
3 if (!$stmt) {  
4     http_response_code(response code: 500);  
5     exit("✖ Database prepare error: " . $conn->error);  
6 }  
/  
8  
9     $stmt->bind_param(types: "sssssis", var: $name, vars: $email, $phone, $reservation_date, $time, $guests, $table);  
0  
1 if ($stmt->execute()) {  
2     // Update table status to 'reserved' after successful booking  
3     $update_stmt = $conn->prepare(query: "UPDATE table_info SET status = 'booked' WHERE table_id = ?");  
4     if (!$update_stmt) {  
5         http_response_code(response code: 500);  
6         exit("✖ Database prepare error (update): " . $conn->error);  
7     }  
8     $update_stmt->bind_param(types: "i", var: $table);  
9     if ($update_stmt->execute()) []  
0         echo "✓ Reservation submitted successfully! Table status updated.";  
1     } else {  
2         http_response_code(response_code: 500);  
3         echo "✖ Reservation saved but failed to update table status: " . $update_stmt->error;  
4     }  
5 }
```

# Past Reservations:



A screenshot of a code editor showing a file named `past_reservations.php`. The code is a PHP script that connects to a database and executes a SELECT query to retrieve past reservations. The code includes error handling for database connection errors and prepares the SQL query with parameters.

```
signup.php | reservation.php | past_reservations.php X | index.html  
past_reservations.php  
19 if ($conn->connect_error) {  
21 }  
22 $conn->set_charset(charset: "utf8mb4");  
23  
24 // Corrected JOIN query  
25 $sql = "  
26     SELECT  
27         r.id AS reservation_id,  
28         u.name AS user_name,  
29         u.email AS user_email,  
30         r.phone AS reservation_phone,  
31         r.date,  
32         r.time,  
33         r.guests,  
34         r.table_no,  
35         r.name AS reservation_name,  
36         u.created_at  
37     FROM reservations r  
38     INNER JOIN users u ON r.email = u.email  
39     WHERE r.email = ?  
40     ORDER BY r.date DESC, r.time DESC  
41 ";  
42  
43 $stmt = $conn->prepare(query: $sql);  
44 if ($stmt === false) {  
45     die("Prepare failed: " . $conn->error);
```

# CONCLUSION

- Successfully created a functional and user-friendly reservation system
- Learned about web development, database integration, and UI/UX
- This system can improve restaurant efficiency and customer convenience
- Ready for further development and deployment

