

**Dr. Babasaheb Ambedkar Marathwada University, Chh.
Sambhajinagar**



QUEST FOR EXCELLENCE

G. S. Mandal's

MARATHWADA INSTITUTE OF TECHNOLOGY
CIDCO, Chh. Sambhajinagar

A project report

Restaurant-Table-Booking-System

Submitted by –

Miss. Yashoda Gautam

Mr. Akash Nikalje

MSc.(CS) – 2nd Sem.

Guided by –

Dr. Mahendra Kondekar Sir

In the fulfillment of the degree

Master of Science (Computer Science) 1st Year (2nd Sem.)

Department of CS [PG]

Academic Year :- 2024 - 25



G. S. Mandal's

**MARATHWADA INSTITUTE OF TECHNOLOGY
CIDCO, AURANGABAD**

Certificate

This is to certify that Mr. Rajesh Nikam have successfully completed the project entitled “**Restaurant-Table-Booking-System**” in the fulfillment of the degree ‘Master of Science (Computer Science)’ in the academic year 2024-25 in the Department of Computer Science & Information Technology[PG].

During the project work, he/she has done the work very sincerely.

HOD
(Prof. S.A.Vyavahare)

Project Guide
(Prof. Mahendra
Kondekar)

External Examiner

Principal
(Dr. M.H.Kondekar)

“Restaurant-Table-Booking-System”

Acknowledgement

It gives me proud privilege to complete this project work. This is the only page where I have the opportunity to express my emotions and gratitude from the bottom of my heart.

It is my great pleasure in expressing sincere and deep gratitude towards my **guide Dr. Mahendra Kondekar**, Marathwada Institute of Technology, Cidco, Chh. Sambhajinagar, for his valuable and firm suggestions, guidance and constant support throughout this work. I am thankful to for providing me various resources and infrastructure facilities.

I also offer my most sincere thanks to Principle **Dr. M.H.Kondekar, Principal**, Marathwada Institute of Technology, Cidco, Chh. Sambhajinagar, my colleagues and staff members of Computer Science and Management Department, Marathwada Institute of Technology, Cidco for cooperation provided by them in many ways.

Last but not the least I am thankful to my Company boss/guide Mr. [-----] for his valuable guidance to complete my work.

Mr. Rajesh Nikam
MSc(CS)

INDEX

1. Introduction

- 1.1 Overview of the Project
- 1.3 Scope of the Project
- 1.4 Technologies Used

2. System Analysis

- 2.1 Objective
- 2.2 Functional requirements
- 2.3 Non-functional requirements

3. Feasibility Report

- 3.1 Economic Feasibility
- 3.2 Technical Feasibility
- 3.1 Operational Feasibility

4. System Design

- 4.1 ER Diagram (Entity-Relationship Diagram)
- 4.2 Data Flow Diagram (DFD)
- 4.3 Use Case Diagram

5. Annexure-1

- 5.1 Program Code -php
- 5.2 Program Code -html

6. Testing

6.1 Introduction

6.2 Testing Methodology

6.3 Conclusion

7. Annexure-2 Results & Screenshots

7.1 Table Booking Form

7.2 Booking Details

7.3 Manage Tables

7.4 Tracking ID

7.5 Admin Panel Login

7.6 Admin Dashboard

7.7 Database

8. Drawbacks and Limitations

8.1 Verification

8.2 Validation

8.3 Drawbacks / Limitations

8.4 Conclusion

9. Bibliography

1. Introduction

1.1 Overview

With the rise of digital solutions, traditional restaurant booking methods have become outdated. This project introduces a web-based system where customers can book tables online conveniently. The system provides a user-friendly interface, database storage for bookings, and automated notifications to confirm reservations.

1.2 What is WAMP?

WAMP (Windows, Apache, MySQL, PHP) is a software stack that provides a local web development environment. It includes:

- Windows: The operating system.
- Apache: A web server to handle HTTP requests.
- MySQL: A relational database for storing restaurant bookings.
- PHP: A server-side scripting language for processing bookings.

1.3 What is PHP?

PHP (Hypertext Preprocessor) is a widely used scripting language that enables dynamic web applications. It helps process form submissions, interact with databases, and send automated notifications.

1.4 What is MySQL?

MySQL is an open-source relational database management system (RDBMS) that stores customer reservations, user details, and booking history.

1.5 Scope of Work

The system includes the following features:

- User-friendly booking form.
- Database to store reservation details.
- Automated email and SMS notifications.

- Admin panel to manage reservations.
- Responsive design for mobile and desktop compatibility.

1.6 Operating Environment

The operating environment consists of the necessary hardware and software required for the proper functioning of the Online Restaurant Booking System.

Hardware Requirements:

- Processor: Minimum 1GHz processor for handling web requests.
- RAM: Minimum 4GB RAM for smooth application performance.
- Storage: Minimum 100GB HDD/SSD for database and application storage.
- Network Connection: A stable internet connection for online access.

Software Requirements:

- Operating System: Windows
- Web Server: Apache (included in WAMP)
- Database Server: MySQL for managing reservations.
- Scripting Language: PHP for server-side logic.
- Web Browser: Chrome, Firefox, Edge for user interface access.
- Development Tools: Sublime Text, VS Code, or Notepad++ for coding.
- Frameworks/Libraries: JavaScript for UI enhancements.

The application runs on the WAMP server, which acts as a local host for web development and testing before deployment to an actual web server. It is designed to be accessible on various devices, ensuring maximum usability for customers and restaurant managers.

2. Proposed System

2.1 Objectives of the System

- To develop an efficient and user-friendly restaurant booking system.
- To store customer booking details securely in a database.
- To send confirmation messages upon successful reservation.
- To allow restaurant staff to manage and view booking requests.
- To improve the overall customer experience through automation.

2.2 User Requirements

- Customers should be able to book tables online easily.
- Admins should be able to manage reservations.
- Customer can check booking status by tracking id.

1. Functional Requirements

Functional requirements define the core functionalities that the system must perform. These are directly related to user interactions, system processes, and business rules.

1.1 Table Booking Management

- Customers should be able to enter details like name, contact number, number of guests, and booking date.
- The system should check for available tables before confirming a reservation.

1.2 Booking Confirmation and Notification

- Once a booking is successfully made, the system should display a confirmation message.
- An email or SMS notification should be sent to the customer confirming their reservation details.

- Restaurant administrators should also receive notifications for new bookings.

1.3 Admin Panel for Booking Management

- Admins should be able to view, update, or cancel reservations.
- Admins should have access to customer details and booking history.

1.4 Special Requests and Customization

- Customers should be able to mention special requests (e.g., birthday celebration, window seat).
- The system should store these requests and display them to the restaurant staff.

1.5 Reporting and Analytics

- Admins should be able to generate reports on booking trends, peak hours, and customer preferences.
- Monthly or weekly summaries should be available for business analysis.

2. Non-Functional Requirements

Non-functional requirements define the overall system attributes, including performance, security, usability, and reliability.

2.1 Performance Requirements

- The system should handle multiple concurrent bookings without delays.
- The response time for booking confirmation should be less than 3 seconds.
- The system should support at least 1000 simultaneous users.

2.3 Usability Requirements

- The system should be intuitive and easy to navigate for all users.
- The UI should be responsive, ensuring compatibility across different devices (mobile, tablet, desktop).

- Users should be able to complete a booking with minimal clicks (ideally within 3-5 steps).

2.4 Availability and Reliability

- The system should have at least 99.9% uptime to ensure continuous availability.
- Backup mechanisms should be implemented to restore booking data in case of system failure.
- Scheduled maintenance should be minimal and should not disrupt peak booking hours.

2.5 Compatibility Requirements

- The application should be compatible with popular browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- The system should function properly across Windows, macOS, and Linux operating systems.

2.6 Maintainability and Scalability

- The system should be modular, making it easy to add new features (e.g., loyalty programs, AI-based recommendations).
- The database should be optimized to handle increasing users and bookings without performance degradation.

2.7 Data Integrity and Backup

- The system should ensure that all stored booking data remains consistent and accurate.
- Automated daily backups should be implemented to prevent data loss.

3. Feasibility Report

3.1 Economic Feasibility

Economic feasibility determines whether the system is cost-effective. Since this project uses open-source technologies like PHP, MySQL, and WAMP, there is no need for additional software expenses, making it a low-cost solution.

3.2 Technical Feasibility

The technical feasibility study evaluates whether the system can be implemented with existing technology. This project is technically feasible because it utilizes standard web development technologies and relational databases.

3.3 Operational Feasibility

Operational feasibility ensures that the system will be accepted and used effectively. By simplifying restaurant operations and providing a seamless user experience, this system enhances efficiency.

4. System Design

4.1 ER Diagram

The Entity-Relationship (ER) diagram visually represents the database structure and relationships between tables.

4.2 Data Flow Diagram (DFD)

The DFD represents the flow of information within the system, showcasing user interactions and database operations.

4.3 Use Case Diagram

The use case diagram illustrates the different actions performed by users and the system's response.

5. Annexure 1

Program Code:-

// PHP Code for Booking Request Processing

```
<?php
```

```
use PHPMailer\PHPMailer\PHPMailer;
```

```
use PHPMailer\PHPMailer\Exception;
```

```
require 'vendor/autoload.php'; // Include PHPMailer
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $conn = new mysqli("localhost", "abc", "", "restaurant");
```

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

```
// Retrieve data from form
```

```
$name = $_POST['name'];
```

```
$persons = $_POST['persons'];
```

```
$date = $_POST['date'];
```

```
$req = $_POST['req'];
```

```
$contact_no = isset($_POST['contact_no']) ? trim($_POST['contact_no']) : "";
```

```
// Insert into database
```

```
$sql = "INSERT INTO book_req (name, persons, date, req, contact_no) VALUES  
('$name', '$persons', '$date', '$req', '$contact_no')";
```

```
if ($conn->query($sql) === TRUE) {
```

```
    // Data stored successfully, now send email via PHPMailer
```

```

$mail = new PHPMailer(true);

try {
    // SMTP Configuration
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'Mailid.com'; // Your email
    $mail->Password = 'Your Password'; // Your email password or App Password
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // Email Settings
    $mail->setFrom('mail id', 'Restaurant Booking');
    $mail->addAddress('mail id'); // Replace with your notification email
    $mail->Subject = "New Booking Request from $name";
    $mail->Body = "You have received a new booking request:\n\nName:
$name\nPersons: $persons\nDate: $date\nSpecial Request: $req";

    $mail->send();
} catch (Exception $e) {
    // Email error handling (optional)
}

// Styled confirmation message with redirect to booking4.html
echo "<div style='width: 50%; margin: 20px auto; padding: 15px; border-radius:
5px; background-color: #dff0d8; color: #3c763d; text-align: center; font-size: 18px;'>";
    echo "Thank you, $name! Your booking request for $persons person(s) on $date has
been received.<br><br>";
    echo "<a href='booking4.html' style='padding: 10px 20px; border-radius: 5px;
background-color: #007bff; color: white; text-decoration: none;'>Go to Home</a>";
echo "</div>";

```

```

// Auto redirect to booking4.html after 5 seconds
echo "<script>setTimeout(function(){ window.location.href =
'http://localhost/Restaurant_booking/booking4.html'; }, 5000);</script>";

} else {
    echo "<div style='width: 50%; margin: 20px auto; padding: 15px; border-radius:
5px; background-color: #f2dede; color: #a94442; text-align: center; font-size: 18px;'>";
    echo "Error: " . $sql . "<br>" . $conn->error;
    echo "</div>";
}

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

```

// html Code for Booking Request Processing

//CSS Code:

```

body, html {
    height: 100%;
    font-family: "Inconsolata", sans-serif;
}

.bgimg {
    background-position: center;
    background-size: cover;
    background-image: url("/Restaurant_booking/images/pexels-reneterp-
1581384.jpg");
    min-height: 75%;
}

```

```
}
```

```
.menu {  
    display: none;  
}
```

```
.scroll-container {  
    display: flex;  
    overflow-x: auto;  
    white-space: nowrap;  
    scroll-behavior: smooth;  
    gap: 10px;  
    padding: 10px;  
    background: #f8f8f8;  
}
```

```
.scroll-container img {  
    height: 200px;  
    border-radius: 10px;  
    transition: transform 0.3s;  
}
```

```
.scroll-container img:hover {  
    transform: scale(1.1);  
}
```

//Java Script

```
<script>
```

```
// Function to switch between menu sections
```

```
function openMenu(evt, menuName) {  
    var i, x, tablinks;
```



```

x = document.getElementsByClassName("menu");
for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
}
tablinks = document.getElementsByClassName("tablink");
for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" w3-white", "");
}
document.getElementById(menuName).style.display = "block";
evt.currentTarget.className += " w3-white";
}

// Add auto slide functionality to images
let images = document.querySelectorAll(".scroll-container img");
let index = 0;
function autoSlide() {
    images[index].style.transform = "scale(1)";
    index = (index + 1) % images.length;
    images[index].style.transform = "scale(1.1)";
}

setInterval(autoSlide, 3000);
</script>

```

// Booking Form

```

<!-- Booking Form
<div class="w3-container w3-padding-64 w3-center">
    <h2>Book a Table</h2>
    <form action="PHP.php" method="POST">
        <label>Name:</label>

```

```
<input type="text" name="name" required>

<label>Email:</label>
<input type="email" name="email" required>

<label>Date:</label>
<input type="date" name="date" required>

<label>Time:</label>
<input type="time" name="time" required>

<button type="submit">Book Table</button>
</form>
```

//User Dashboard for tracking status

```
<h2>Check Your Booking Status</h2>
  <form method="post" action="">
    <input type="text" name="id" placeholder="Enter Booking ID" required>
    <button type="submit" name="track">Track Booking</button>
  </form>
</div>
</body>
</html>
```

// Admin Dashboard

```
<form method="post" action="update_status.php" style="display: flex; gap: 10px;
justify-content: center;">
    <input type="hidden" name="booking_id" value="<?= $row['id'] ?>">
    <button type="submit" name="status" value="Accepted" class="accept-
btn">Accept</button>
    <button type="submit" name="status" value="Rejected" class="reject-
btn">Reject</button>
</form>
```

// Update status in the database

```
$stmt = $conn->prepare("UPDATE bookings SET status = ? WHERE id = ?");
$stmt->bind_param("si", $status, $booking_id);

if ($stmt->execute()) {
    echo "<script>alert('Booking status updated successfully!'); window.location.href =
'admin_dashboard.php';</script>";
} else {
    echo "Error updating record: " . $conn->error;
}

$stmt->close();
}
```

//Admin Login

```
<form action="admin_login.php" method="POST" style="text-align: center;">
    <input type="text" name="username" placeholder="Admin Username" required
        style="display: block; width: 80%; max-width: 300px; padding: 10px; margin:
10px auto; border: 1px solid #ccc; border-radius: 5px;">

    <input type="password" name="password" placeholder="Admin Password" required
        style="display: block; width: 80%; max-width: 300px; padding: 10px; margin:
10px auto; border: 1px solid #ccc; border-radius: 5px;">

    <button type="submit"
        style="background: #007BFF; color: white; border: none; padding: 10px 20px;
font-size: 16px; cursor: pointer; border-radius: 5px; margin-top: 10px;">
        Login
    </button>
</form>
```

// Database Connection

```
<?php
$servername = "localhost";
$username = "abc"; // Default for WAMP
$password = ""; // Default for WAMP
$dbname = "login"; // Change to your actual

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

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

```
}  
?>
```

// Track Booking

```
<form method="post">  
    <input type="text" name="tracking_id" placeholder="Enter Tracking ID" required>  
    <button type="submit">Check Status</button>  
</form>
```

// Admin Logout

```
<?php  
session_start();  
session_unset();  
session_destroy();  
  
echo "Logging out..."; // Debugging line  
header("Location: http://localhost/Restaurant_booking/admin_login.html");  
exit();  
?>
```

6. Testing

1. Introduction

This document outlines the testing process conducted for the Restaurant Booking Website to ensure that all features work correctly, securely, and efficiently. The testing was performed based on functional testing, UI testing, and security testing.

2. Testing Methodology

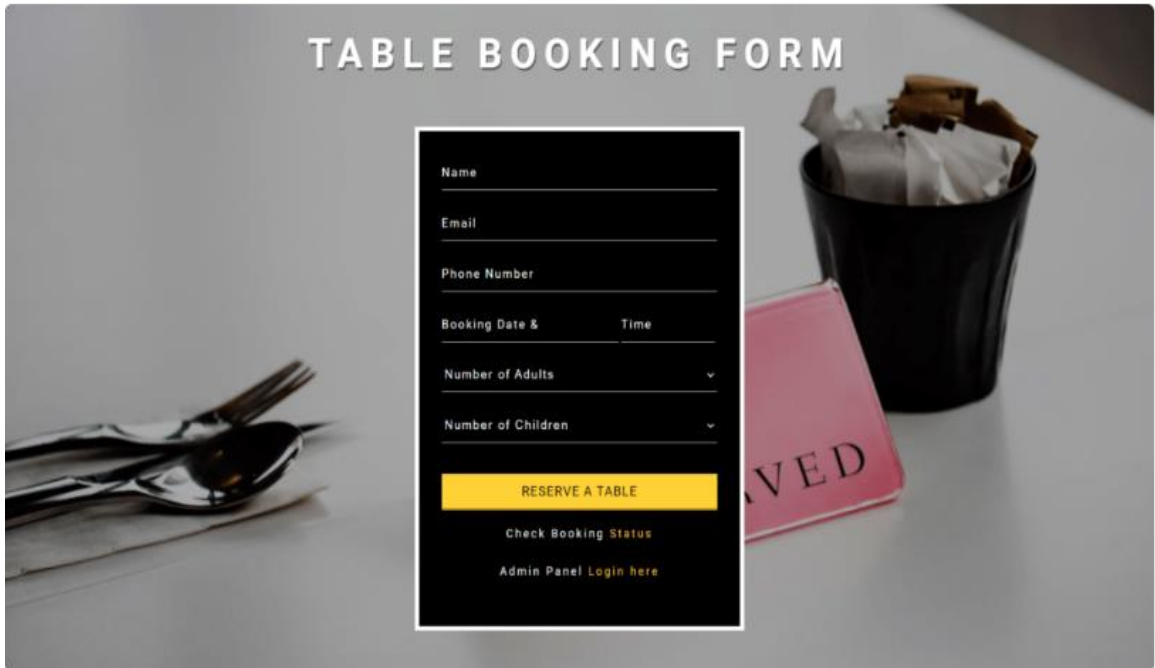
- **Manual Testing:** The website was tested manually by navigating through different functionalities.
- **Functional Testing:** Each feature was tested to verify correct output.
- **Compatibility Testing:** The website was checked on different browsers and devices.

3. Conclusion

The Restaurant Booking Website has been tested thoroughly. All major features are working as expected. Minor UI issues were found and fixed. The website is now ready for deployment.

7. Annexure-2 Results & Screenshots

7.1 Table Booking Form



The screenshot shows a 'TABLE BOOKING FORM' overlaid on a background image of a restaurant table. The table has a white tablecloth, a black cup with a napkin, a pink folder with the word 'VED' visible, and a set of silverware. The form itself is a dark rectangle with white text and input fields. It includes fields for Name, Email, Phone Number, Booking Date & Time, Number of Adults, and Number of Children. There is a prominent yellow button labeled 'RESERVE A TABLE' and two smaller links at the bottom: 'Check Booking Status' and 'Admin Panel Login here'.

TABLE BOOKING FORM

Name

Email

Phone Number

Booking Date & Time

Number of Adults

Number of Children

RESERVE A TABLE

[Check Booking Status](#)

[Admin Panel Login here](#)

7.2 Booking Details

RTBS | Admin

admin

Dashboard

Sub-Admins

Tables

Bookings

B/w Dates Report

Account Settings

Booking Details

Home / Booking Details

Booking Details

Booking Number	323168839		
Name	Shivani	Email Id	shivani@gmail.com
Mobile No	4563214520	No of Adults	4
No of Childs	1	Booking Date / Time	2023-06-10/14:10:00
Posting Date	2023-05-27 10:05:04		
Booking Status	Accepted	Updation Date	2023-06-04 22:38:10
Remark	Table booked.		

Version 1.0

7.3 Manage Tables

RTBS | Admin

admin

Dashboard

Sub-Admins

Tables

Bookings

B/w Dates Report

Account Settings

Manage Tables

Home / Manage Tables

Table Details

Copy CSV Excel PDF Print Column visibility

Search:

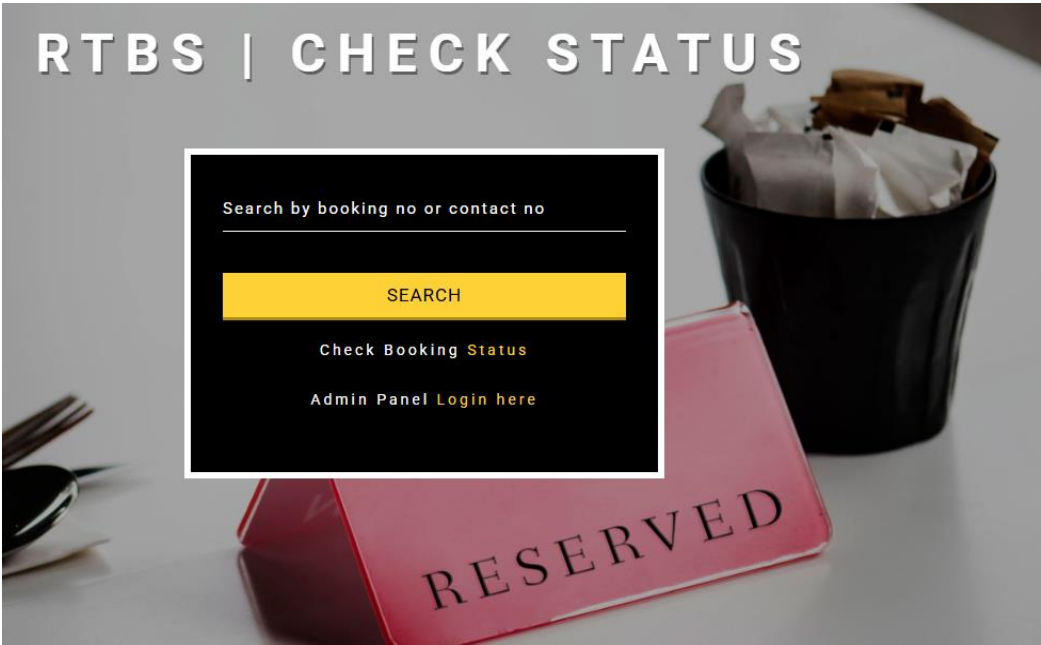
#	Table No	Added By	Creation Date	Action
1	1	Admin	2023-05-27 09:20:35	
2	2	Admin	2023-05-27 09:20:55	
3	1A	Admin	2023-05-27 09:21:01	
4	3	Admin	2023-05-27 09:21:07	
5	3A	Admin	2023-05-27 09:21:11	
6	3B	Admin	2023-05-27 09:21:15	
#	Table No	Added By	Creation Date	Action

Showing 1 to 6 of 6 entries

Previous 1 Next

Version 1.0

- Tracking ID



7.4 Booking Status

search Results against '5611079514'

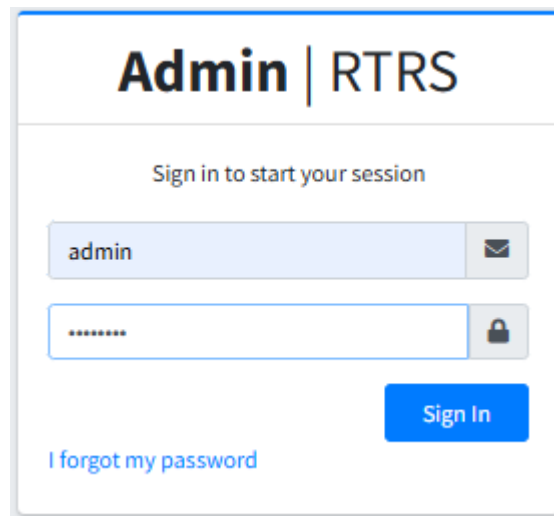
Search Details

Copy CSV Excel PDF Print Column visibility ▾										Search: <input type="text"/>	
#	Bookings No	Name	Email Id	Mobile No	No. Adults	No of Childrens	Boking Date/Time	Posting Date	Action		
1	5611079514	Yashoda Gautam	gautamanu7411@gmail.com	8421494459	1	1	0000-00-00/9:10 PM	2025-04-09 21:11:58	View Details		
#	Bookings No	Name	Email Id	Mobile No	No. Adults	No of Childrens	Boking Date/Time	Posting Date	Action		

Showing 1 to 1 of 1 entries

Previous 1 Next

7.5 Admin Panel Login



The login form is titled "Admin | RTRS" and prompts the user to "Sign in to start your session". It features two input fields: one for the username "admin" and another for the password, which is masked with dots. Each field has an icon (envelope for email, padlock for password) to its right. A blue "Sign In" button is positioned below the password field. A link "I forgot my password" is located at the bottom left of the form.

Admin | RTRS

Sign in to start your session

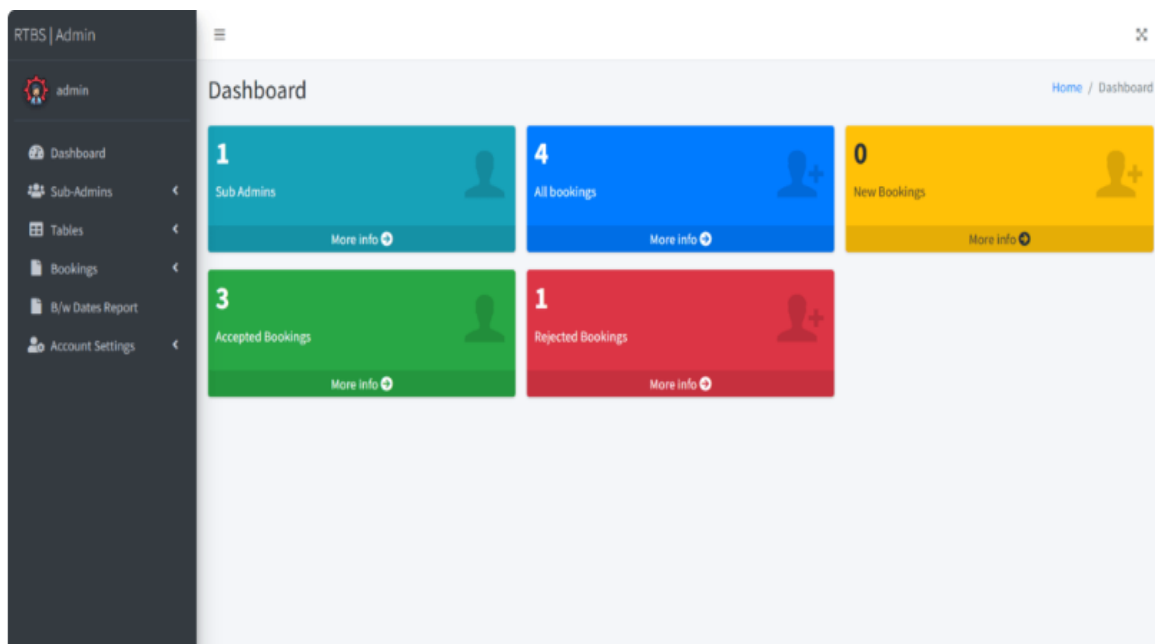
admin

.....

[Sign In](#)

[I forgot my password](#)

7.6 Admin Dashboard



7.7 Database

The screenshot displays the phpMyAdmin web interface. On the left is a sidebar with a tree view of databases and tables. The main panel shows the 'tbladmin' table selected within the 'rtbsdb' database. The table structure is shown as a single column 'AdminName'. Below this, the table data is displayed in a grid. The data includes two rows: one for 'Admin' and one for 'Anuj kumar'. The interface includes various navigation tabs at the top, a status bar indicating the number of rows shown, and a section for query results operations at the bottom.

Server: 127.0.0.1 » Database: rtbsdb » Table: tbladmin

Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

`SELECT * FROM `tbladmin``

☐ 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

	ID	AdminName	AdminuserName	MobileNumber	Email	Password	AdminRegdate	UserType
<input type="checkbox"/>	2	Admin	admin	123456789	admin@gmail.com	f925916e2754e5e03775dd58a5733251	2025-04-04 00:00:00	1
<input type="checkbox"/>	3	Anuj kumar	akr305	1234567891	ak@gmail.com	f925916e2754e5e03775dd58a5733251	2025-04-08 00:00:00	0

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

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

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

☐ Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

8. Verification and Validation

8.1 Verification (Are we building the product right?)

Verification ensures that the system meets its design requirements. This involves code reviews, walkthroughs, and inspections.

- The website meets the specified requirements (e.g., customers can book a table online).
- The UI is responsive across devices (mobile, tablet, desktop).
Navigation is user-friendly and accessible.

- Code follows proper coding standards (PHP, HTML, CSS, JS).
- No syntax errors or broken links.
- Database connections and queries are optimized

8.2 Validation (Are we building the right product?)

- Customers can book tables successfully.
- Admin can view and manage bookings.
- Tracking ID's are sent to users after booking.
- Booking cannot be submitted with empty or invalid fields.

8.3 Drawbacks and Limitations

- No online payment integration.
- No table view given.

8.4 Conclusion

The Online Restaurant Booking System offers an efficient solution for managing reservations. By automating the booking process, it enhances customer satisfaction and streamlines restaurant operations. Future enhancements could include table availability checks, online payment integration, and customer feedback options.

9. Bibliography

- ❖ PHP Documentation
- ❖ MySQL DB Documentation
- ❖ PHP Mailer Documentation
- ❖ Computer Concepts and Web Technology by Backend Developer.....
- ❖ E-Commerce Online Restaurant Table Booking by Ms. Yashoda Gautam.
- ❖ **GitHub Profile Link :-** <https://github.com/Yashoda84>
- ❖ **LinkedIn Profile :-** <https://www.linkedin.com/in/yashoda-gautam-7a627b235/>