

Booking table at Restaurant Using Restro_Book

Created Date: - 03-Sep-2022

Last update Date: -

Author: - Yadnesh Budukh

Reviewer: -

➤ **Overview:**

Customer

Reserve the table at any restaurant you want on just a single click. See all your previous and upcoming bookings on the dashboard and update or delete your upcoming bookings. Also, you get a reservation confirmation email along with reservation and restaurant details.

Restaurant

Restaurants will be able to check all the bookings in their restaurant in ascending order of date and time. Also, restaurants will get notified for each successful reservation at their restaurant by email along with customer details and reservation details.

➤ **Context:**

If any customer ever wants to reserve a table at any restaurant, he has to find the contact number of that restaurant and ask them to reserve a table or has to go to restaurant to reserve. Finding contact number may be difficult some time as all the restaurants number is not available online and going physically at their place consumes time.

In these cases, the Restro_Book will work as mediator not only for booking but also help user to modify their reservations.

➤ **Milestones:**

- Create a home page for Restro_Book.
- Create two UI for user and restaurant.
- Responsive UI.
- Password hashing using “password hash(‘ ’,Password Default)” as it will provide latest hashing for password .
- Making pages accessible only after successful login.
- Managing the reservations at dashboard.
- Session end when log out.

➤ **Architecture:**

Frontend: HTML, CSS, Bootstrap

Backend: PHP

Database: MySQL

API: RESTful API for Update and Delete

Mail: phpMailer.

Customer UI:

Pages:

Sr. no.	Name	Path
1	Login	C:\xampp\htdocs\project_sapl2\user\login.php
2	Create New Account	C:\xampp\htdocs\project_sapl2\user\register.php
3	Book Now	C:\xampp\htdocs\project_sapl2\user\book1.php
4	Dashboard	C:\xampp\htdocs\project_sapl2\user\dashboard.php
5	Logout	C:\xampp\htdocs\project_sapl2\user\logou.php

Login: (backend : logintab.php)

// Query to check credentials

```
$sql = "SELECT * FROM userinfo WHERE uname = ?";  
$stmt = $conn->prepare($sql);
```

It checks if the inputted username is in the table or not

// Use password_verify to check the password

```
if (password_verify($upass, $user['upass'])) {  
    write_log('logintab.php', __FILE__, 'SUCCESS', $uname." User Logged In Successfully");  
    $_SESSION['authenticated'] = TRUE;  
    $_SESSION['uname'] = $user['uname'];  
    $_SESSION['username'] = $user['username'];  
    $_SESSION['cont'] = $user['cont'];  
    header("Location: dashboard.php");  
    exit();  
} else {  
    write_log('logintab.php', __FILE__, 'Warning', "Entered Username or Password Does not  
match");  
    echo "<script> alert('Wrong username or password'); window.location.href='login.php';  
</script>";  
    exit();  
}  
} else {  
    write_log('logintab.php', __FILE__, 'Warning', "Entered Username or Password Does not  
match");  
    echo "<script> alert('Wrong username or password'); window.location.href='login.php';  
</script>";  
    exit();  
}
```

As password in database is in hashed format it checks if it matches with or not using password_verify().

If user logged in it starts a session and send uname(i.e. user id), username(i.e name) and cont(i.e Contact) and user get redirected to dashboard.

Register: (backend: registerdata.php)

```
$username = $_POST['username'];
$cont = $_POST['cont'];
$uname = $_POST['uname'];
$upass = $_POST['upass'];
$confirm_password = $_POST['confirm_password']
```

Receives user inputted data.

```
// Hash the password using password_hash()
$hashed_password = password_hash($upass, PASSWORD_DEFAULT);
```

```
if ($result->num_rows > 0) {
// Username already exists
echo "<script>alert('Username already exists');</script>";
echo "<script>window.location.href='register.php';</script>";
} else {
// Insert data into userinfo table
$sql = "INSERT INTO userinfo (username, uname, cont, upass) VALUES (?, ?, ?, ?)";
$stmt = $conn->prepare($sql);
if ($stmt === false) {
    throw new Exception("Prepare failed: " . $conn->error);
}
$stmt->bind_param("ssss", $username, $uname, $cont, $hashed_password);
```

Checks for if username already exists if no then enters newly registered data into the table.

```
if ($stmt->affected_rows > 0) {
    echo "<script>alert('Registration successful');</script>";
    echo "<script>window.location.href='login.php';</script>";
} else {
    throw new Exception("Registration failed.");
}
```

For successful register it will redirect user to login page.

Booking: (backend: book.php)

```
require 'PHPMailer-master/src/PHPMailer.php';
require 'PHPMailer-master/src/SMTP.php';
require 'PHPMailer-master/src/Exception.php';
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
```

Files required for sending a mail to user and restaurant after successful booking.

```
$uname = $_SESSION['uname'];
$cont = $_SESSION['cont'];
$rname = $_POST['rname'];
$res_date = $_POST['date'];
$res_time = $_POST['time'];
$tab_no = $_POST['tab_no'];
```

Receives uname and cont from session and other details from user input.

```
$sql_insert = "INSERT INTO reservations (username, cont, rname, res_date, res_time, tab_no)
VALUES (?, ?, ?, ?, ?, ?)";

$stmt_insert = $conn->prepare($sql_insert);
if ($stmt_insert === false) {
    write_log('book.php', __FILE__, 'ERROR', "Error preparing statement: " . $conn->error);
    die("Error preparing statement: " . $conn->error);
}

$stmt_insert->bind_param("sssssi", $uname, $cont, $rname, $res_date, $res_time, $tab_no);
```

Data inserted in reservations table for after successful reservation.

This is the mail generator which send the reservation confirming mail to the user and reservation update email to restaurant.

```
if ($stmt_insert->execute()) {  
    $reservation_id = $conn->insert_id;  
    write_log('book.php', __FILE__, 'SUCCESS', 'Reservation successful. ID: ' . $reservation_id);  
    echo "Reservation successful! Your reservation ID is: " . $reservation_id;
```

Fetches restaurants email id

```
$query = "SELECT email FROM restinfo WHERE rname = ?";  
$stmt = $conn->prepare($query);  
$stmt->bind_param('s', $rname);  
$stmt->execute();  
$stmt->bind_result($restinfo_email);  
$stmt->fetch();
```

Send email to restaurant

```
$mail = new PHPMailer(true);  
  
try {  
    $mail->isSMTP();  
    $mail->SMTPDebug = 0;  
    $mail->SMTPSecure = 'ssl';  
    $mail->SMTPAuth = true;  
    $mail->Host = 'smtp.gmail.com';  
    $mail->Port = 465;  
    $mail->Username = 'yadneshsbudukh01@gmail.com';  
    $mail->Password = 'zsro mwgmg urjt eafo'; //App Password allows to send mail  
    $mail->setFrom('yadneshsbudukh01@gmail.com', 'Reservation System');  
    $mail->addAddress($restinfo_email);  
    $mail->Subject = 'New Reservation';  
    $mail->Body = "Hello {$rname}! You have a new reservation:\n\n"  
        . "Customer Name: {$booking['username']}\n"  
        . "Contact Number: {$cont}\n"  
        . "Reservation Date: {$res_date}\n"  
        . "Number of People: {$tab_no}";  
    $mail->send();
```

Send email to user

```
$mail->clearAddresses();
```

```
$mail->addAddress($booking['email']);
```

Clears all the address stored at time of sending email to restaurant and add new recipient i.e. user's email address

```
try {
```

```
    $mail->Subject = 'Reservation Confirmation';
```

```
    $mail->Body = "Hello {$booking['username']}! Your reservation is successful:\n\n"
```

```
        . "Restaurant Name: {$rname}\n"
```

```
        . "Reservation Date: {$res_date}\n"
```

```
        . "Number of People: {$stab_no}\n"
```

```
        . "Thank you for choosing us!";
```

```
$mail->send();
```

Dashboard: Displays user booking and gives update and delete button.

```
<tbody> <?php
    while ($booking = $result->fetch_assoc()) {
$reservationDate = new DateTime($booking['res_date']);
$currentDate = new DateTime();
?>
<tr>

    <td><?php echo htmlspecialchars($booking['id']); ?></td>
    <td><?php echo htmlspecialchars($booking['rname']); ?></td>
    <td><?php echo htmlspecialchars($booking['rcontact']); ?></td>
    <td><?php echo htmlspecialchars($booking['radd']); ?></td>
    <td><?php echo htmlspecialchars($booking['res_date']); ?></td>
    <td><?php echo htmlspecialchars($booking['res_time']); ?></td>
    <td><?php echo htmlspecialchars($booking['tab_no']); ?></td>
    <td>
        <?php if ($reservationDate >= $currentDate->setTime(0, 0, 0)) { ?>
        <a href="update_reservation.php?id=<?php echo $booking['id']; ?>" class="btn btn-
        primary">Update</a>
        <form action="reservation_api.php" method="post" style="display:inline;">
        <input type="hidden" name="id" value="<?php echo $booking['id']; ?>">
        <input type="hidden" name="_method" value="DELETE">
        <button type="submit" class="btn btn-danger" onclick="return confirm('Are you sure you want to
        delete this reservation?');">Delete</button>
        </form>
        <?php } else { ?> <p>No actions available</p> <?php } ?>
    </td> </tr>

<?php } write_log('dashboard.php',__FILE__, 'INFO', "Reservation Displayed Successfully");?>
        </tbody>
        </table>
    </div> <?php } else { ?>
        <p>You have no bookings yet.</p>
        <?php write_log('dashboard.php',__FILE__, 'INFO', $uname." Not have any reservation");} ?>
```

Query to display data:

```
SELECT r.id, ri.rname, ri.rcontact, ri.radd, r.res_date, r.res_time, r.tab_no
FROM restinfo ri
JOIN reservations r ON ri.rname = r.rname
WHERE r.username = ?
ORDER BY r.res_date ASC, r.res_time ASC"
```

Log Writing:

```
function write_log($fileName, $filePath, $activityType, $remark) {
    date_default_timezone_set('Asia/Kolkata'); //Setting Up Indian time zone
    // Specify the log file path
    $logFilePath = '../logs/application.log';
    // Create logs directory if it doesn't exist
    if (!file_exists('../logs')) {
        mkdir('logs', 0755, true); //0755 gives permission to read, write and run the files in
        directory
    }
    // Get current date and time
    $dateTime = date('Y-m-d H:i:s');
    // Format the log entry
    $logEntry = sprintf(
        "[%s] %s | File: %s | Path: %s | %s: %s\n",
        $dateTime,
        $activityType,
        $fileName,
        $filePath,
        $activityType,
        $remark
    );
};
```



```
// Write log entry to the file  
error_log($logEntry, 3, $logFilePath);  
}
```

Statement to write log:

```
write_log('page',__FILE__,'INFO',$uname." Not have any reservation")
```

contains:

date_time: automatic on at time of writing log

page: name of page

__FILE__: path of the file

Activity_type: info, error, success

Remark: Information about activity

Database:

Tables:

1. Userinfo:

Sr.	Name	Type	Data
1	Id	Int(10)	Id number
2	Username	Varchar(25)	Name of user
3	uname	Varchar(20)	User id(credential)
4	Upass	Varchar(255)	Hashed Password
5	Cont	Varchar(10)	Contact Number
6	Email	Varchar(255)	Email address

2. Restinfo:

Sr.	Name	Type	Data
1	Ruid	Varchar(15)	User id
2	Upass	Varchar(255)	Password
3	rname	Varchar(25)	Restaurant Name
4	Radd	Varchar(100)	Address
5	rcontact	Varchar(10)	Contact Number
1	email	Varchar(255)	Email

3. Reservations:

Sr.	Name	Type	Data
1	Id	Int(10)	Id number
2	uname	Varchar(20)	User name
3	ucontact	Varchar(10)	User Contact Number
4	rname	Varchar(25)	Restaurant Name
5	res_date	Date	Reservation Date
6	res_time	Time	Reservation Time
7	tab_no	Int(11)	Table number