

**WEB PROGRAMMING**

**(SE 513)**

**Winter Semester (2024-2025)**

|  |  |
| --- | --- |
| **Submitted to:** | **SIR QAMAR NAWAZ** |
| **Submitted by:** | **Hamna Nazar** |
| **Registration No.** | **2022-ag-8054** |
| **Class:** | **BS SE (B)** |
| **Semester:** | **5th** |

**UNIVERSITY OF AGRICULTURE FAISALABAD**

Documentation

EVENT BOOKING WEBSITE

**Introduction**

The Event Booking System is a web-based platform designed to facilitate event reservations for clients. It provides an easy-to-use interface for users to book events, with the backend handling user authentication, form submissions, and database storage. The system ensures smooth operation from viewing prices to confirming event bookings.

**Technologies Used**

1. **Frontend:**
   * HTML: For creating the structure of web pages (e.g., price section, login page).
   * CSS: For styling the interface and ensuring a visually appealing design.
   * JavaScript: For interactive elements like button clicks and form validation.
2. **Backend:**
   * PHP: For server-side scripting to handle login, booking, and database interactions.
3. **Database:**
   * MySQL: For storing and retrieving user, event, and booking details.
4. **Environment:**
   * Web server: To host and run the application.
   * Browser: For accessing the website.

**System Workflow**

1. **Navbar:**

* Home
* Services
* About
* Gallery
* Price
* Review
* Feedback

1. **Price Section:**
   * Displays event pricing information.
   * Includes a "Checkout" button that redirects to the login page.
2. **Login Page:**
   * Contains fields for user credentials (e.g., username and event date and time).
   * Authenticates users to proceed with booking.
3. **Event Booking:**
   * After successful login, users are directed to a booking confirmation page.
   * Clicking the "Book Event" button triggers a PHP script to save the booking information in the database.
   * A confirmation message is displayed: "Client has to pay cash, and the event is booked."
4. **Send feedback:**

Send the feedback in the **messages** table in mysql db along with columns

* name
* email
* phone number
* message

**Database Design**

**Tables:**

1. **Messages**
   * id (Primary Key)
   * name
   * subject
   * email
2. **Bookings**
   * id (Primary Key)
   * user\_id (Foreign Key referencing Users table)
   * event\_id (Foreign Key referencing Events table)
   * booking\_date

**FOOTER**

Special features like:

1. **Email:** direct go to gmail in chrome

2. **Phone number**: go to contacts to call

3**. Location**: Show the location in chrome

**Features**

1. **User Authentication:**
   * Validates user credentials to ensure secure access.
2. **Event Pricing:**
   * Displays pricing details for available events.
3. **Booking System:**
   * Saves booking details in the database.
   * Displays a confirmation message upon successful booking.
4. **Payment Method:**
   * Cash payment is required to finalize the booking.

**Key PHP Functions**

**Login Validation:**

<?php

// Database connection details

$servername = "localhost";

$username = "root"; //username: default is 'root'

$password = ""; //default is an empty string

$dbname = "eventbuzz";

// Create a connection to the database

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

// Check if the connection was successful

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Retrieve form data

$username = $\_POST['username']; // Username from form

$phone = $\_POST['phone']; // Phone number from form

$city = $\_POST['city']; // Selected city from form

$event\_date = $\_POST['event\_date']; // Selected date from form

$packageName = $\_POST['package\_name']; //package name from price

// Prepare SQL query to insert data into the booking table

$sql = "INSERT INTO bookings (username, phone, city, event\_date,package\_name)

VALUES ('$username', '$phone', '$city', '$event\_date', '$packageName')";

// Execute the query

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

// Successful insertion message

echo "

**Booking Confirmation:**

<?php

include 'db\_connection.php';

// Retrieve form data

$username = $\_POST['username'];

$phone = $\_POST['phone'];

$city = $\_POST['city'];

$event\_date = $\_POST['event\_date'];

$package\_Name = $\_POST['package\_name']; // Retrieve package\_name

// Debugging output

echo "Username: $username<br>";

echo "Phone: $phone<br>";

echo "City: $city<br>";

echo "Event Date: $event\_date<br>";

echo "Package Name: $package\_Name<br>";

// Check if the package\_name is empty

if (empty($package\_Name)) {

echo "Error: package\_name is empty!";

exit();

}

// Insert into the database

$sql = "INSERT INTO bookings (username, phone, city, event\_date, package\_name)

VALUES ('$username', '$phone', '$city', '$event\_date', '$package\_Name')";

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

echo "Booking successful!";

header("Location: confirmation.php?package\_name=" . urlencode($package\_Name));

exit();

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

**Future Enhancements**

1. Integrate an online payment gateway.
2. Send email or SMS notifications upon successful booking.
3. Implement an admin panel for managing users and bookings.
4. Add a "Cancel Booking" feature for users.

**Conclusion**

This Event Booking System demonstrates the integration of front-end and back-end technologies to create a functional web application. It offers a user-friendly interface, secure booking process, and efficient data storage. Future enhancements can expand the system's functionality and improve user satisfaction.