

# WEB DEVELOPMENT

## (MINI PROJECT)

### Project Report: Event Management System

NAME	ARMAAN SINGH
UID	23BCA10363
SECTION	23BCA-3A
SUBJECT	WEB DEVELOPMENT
SUBJECT CODE	23CAH-254
SUBMITTED TO	MS. PARDEEP KAUR
DATE	25/3/2025

## **CONTENT: -**

- 1) INTRODUCTION**
- 2) FLOWCHART**
- 3) SOURCE CODE**
- 4) OUTPUT**
- 5) FINAL RESULT**
- 6) LEARNING OUTCOME**

# 1. INTRODUCTION

The Event Management System is a web-based application developed using PHP, MySQL, HTML, CSS, and JavaScript. This system provides a comprehensive platform for managing events, allowing users to register, log in, browse various events, and book tickets. Traditional event management methods are time-consuming, prone to errors, and lack scalability. This system addresses these challenges by offering an automated and user-friendly platform for organizing and attending events. The platform is designed for both event organizers and attendees, enabling event creation, registration tracking, and seamless bookings. This report documents the system's development, including PHP and Apache installation, database design, user authentication implementation, UI creation, and profile picture uploads. Additionally, it highlights key learning outcomes related to modern web development practices.

# 2. ALGORITHM AND FLOWCHART

## Algorithm for Event Management System

1. A structured flowchart visually represents the overall process, including user registration, authentication, event browsing, booking, and profile management.
2. SOURCE CODE The source code includes the following major components:
  - **Database Connection:** Secure connection using MySQLi.
  - **User Authentication:** Login, registration, session management.
  - **Event Management:** Event creation, listing, filtering, and booking.
  - **User Dashboard:** Profile updates, bookings management, profile picture upload.
  - **CSS & JavaScript:** Enhancing UI interactivity and responsiveness.
4. OUTPUT Key outputs include:
  - **User Dashboard:** Displays user profile, booked events, and event recommendations.
  - **Event Listings:** Users can view and search for events.
  - **Booking Confirmation:** Users receive confirmation after successfully booking an event.
  - **Admin Panel:** Manages events, categories, and user registrations.

# Source code:(db\_connect.php, login.php, logout.php my-event.php,home.php)

```
<?php
$servername = "localhost:3307";
$username = "root"; // Change as per your MySQL configuration
$password = ""; // Change as per your MySQL configuration
$dbname = "event_management";

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

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

// Create database if it doesn't exist
$sql = "CREATE DATABASE IF NOT EXISTS $dbname";
if ($conn->query($sql) === TRUE) {
    // Database created successfully or already exists
} else {
    echo "Error creating database: " . $conn->error;
}

// Select the database
$conn->select_db($dbname);

// Create users table if it doesn't exist
$sql = "CREATE TABLE IF NOT EXISTS users (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    contact VARCHAR(15) NOT NULL,
    email VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    event_interest VARCHAR(50) NOT NULL,
    profile_picture VARCHAR(255) DEFAULT NULL,
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    // Table created successfully or already exists
} else {
    echo "Error creating table: " . $conn->error;
}

// Create events table if it doesn't exist
$sql = "CREATE TABLE IF NOT EXISTS events (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    event_name VARCHAR(100) NOT NULL,
    description TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    // Table created successfully or already exists
} else {
    echo "Error creating table: " . $conn->error;
}

// Create event_registrations table if it doesn't exist
$sql = "CREATE TABLE IF NOT EXISTS event_registrations (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id INT(6) UNSIGNED NOT NULL,
    event_id INT(6) UNSIGNED NOT NULL,
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE KEY (user_id, event_id),
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (event_id) REFERENCES events(id) ON DELETE CASCADE
)";

if ($conn->query($sql) === TRUE) {
    // Table created successfully or already exists
} else {
    echo "Error creating table: " . $conn->error;
}

// Insert default events if they don't exist
$events = ["Dance", "Music", "Poetry", "Art", "Theater", "Photography", "Film", "Literature", "Fashion", "Technology"];
foreach ($events as $event) {
    $stmt = $conn->prepare("SELECT * FROM events WHERE event_name = ?");
    $stmt->bind_param("s", $event);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows == 0) {
        $stmt = $conn->prepare("INSERT INTO events (event_name, description) VALUES (?, ?)");
        $description = "This is a " . $event . " event.";
        $stmt->bind_param("ss", $event, $description);
        $stmt->execute();
    }
    $stmt->close();
}
?>
```

```

login.php

<?php
session_start();
include 'db_connect.php';

$errors = [];

// Check if user is already logged in
if (isset($_SESSION['user_id'])) {
    header("Location: home.php");
    exit();
}

// If form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get form data
    $email = trim($_POST['email']);
    $password = trim($_POST['password']);

    // Validate email
    if (empty($email)) {
        $errors['email'] = "Email is required";
    } elseif (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $errors['email'] = "Invalid email format";
    }

    // Validate password
    if (empty($password)) {
        $errors['password'] = "Password is required";
    }

    // If no errors, check credentials
    if (empty($errors)) {
        $stmt = $conn->prepare("SELECT id, first_name, last_name, password FROM users WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows == 1) {
            $user = $result->fetch_assoc();

            // Verify password
            if (password_verify($password, $user['password'])) {
                // Password is correct, start a new session
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['first_name'] = $user['first_name'];
                $_SESSION['last_name'] = $user['last_name'];

                // Redirect to home page
                header("Location: home.php");
                exit();
            } else {
                $errors['login'] = "Invalid email or password";
            }
        } else {
            $errors['login'] = "Invalid email or password";
        }
    }

    $stmt->close();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Event Management System</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <div class="form-container">
            <h1>Login</h1>

            <?php if (isset($errors['login'])): ?>
            <div class="error-message">
                <?php echo $errors['login']; ?>
            </div>
            <?php endif; ?>

            <form id="loginForm" method="POST" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
                <div class="form-group">
                    <label for="email">Email</label>
                    <input type="email" id="email" name="email" value="<?php echo isset($_POST['email']) ? htmlspecialchars($_POST['email']) : ''; ?>">
                    <?php if (isset($errors['email'])): ?>
                    <span class="error"><?php echo $errors['email']; ?></span>
                    <?php endif; ?>
                </div>

                <div class="form-group">
                    <label for="password">Password</label>
                    <input type="password" id="password" name="password">
                    <?php if (isset($errors['password'])): ?>
                    <span class="error"><?php echo $errors['password']; ?></span>
                    <?php endif; ?>
                </div>

                <div class="form-group">
                    <button type="submit">Login</button>
                </div>

                <div class="form-footer">
                    Don't have an account? <a href="register.php">Register</a>
                </div>
            </form>
        </div>
    </div>

    <script src="script.js"></script>
</body>
</html>

```

```

my-events.php

<?php
session_start();
include 'db_connect.php';

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

$user_id = $_SESSION['user_id'];

// Get events the user is registered for
$registered_events = [];
$stmt = $conn->prepare("
    SELECT e.*
    FROM events e
    JOIN event_registrations er ON e.id = er.event_id
    WHERE er.user_id = ?
    ORDER BY er.registration_date DESC
");
$stmt->bind_param("i", $user_id);
$stmt->execute();
$result = $stmt->get_result();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $registered_events[] = $row;
    }
}
$stmt->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta names="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Events - Event Management System</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <header class="main-header">
            <h1>My Events</h1>
            <nav>
                <ul>
                    <li><a href="home.php">Home</a></li>
                    <li><a href="events.php">Events</a></li>
                    <li><a href="my-events.php" class="active">My Events</a></li>
                    <li><a href="about.php">About</a></li>
                    <li><a href="logout.php">Logout</a></li>
                </ul>
            </nav>
        </header>

        <div class="content">
            <div class="my-events-section">
                <h2>Events You're Registered For</h2>

                <?php if (empty($registered_events)): ?>
                    <div class="error-message">
                        You haven't registered for any events yet. <a href="events.php">Browse events</a> to find something you're interested in!
                    </div>
                <?php else: ?>
                    <div class="events-grid">
                        <?php foreach ($registered_events as $event): ?>
                            <div class="event-card">
                                <div class="event-card-image">
                                    ">
                                </div>
                                <div class="event-card-content">
                                    <h3><?php echo htmlspecialchars($event['event_name']); ?></h3>
                                    <p><?php echo htmlspecialchars(substr($event['description'], 0, 100) . '...'); ?></p>
                                </div>
                                <div class="event-card-footer">
                                    <span class="date"><?php echo date('F j, Y', strtotime($event['created_at'])); ?></span>
                                    <a href="event-details.php?id=<?php echo $event['id']; ?>" class="btn btn-small">View Details</a>
                                </div>
                            </div>
                        <?php endforeach; ?>
                    </div>
                <?php endif; ?>
            </div>
        </div>

        <script src="script.js"></script>
    </body>
</html>

```

```

JS script.js

document.addEventListener("DOMContentLoaded", () => {
  const registrationForm = document.getElementById("registrationForm")

  if (registrationForm) {
    registrationForm.addEventListener("submit", (event) => {
      let hasError = false

      // Get form fields
      const firstName = document.getElementById("first_name")
      const lastName = document.getElementById("last_name")
      const contact = document.getElementById("contact")
      const email = document.getElementById("email")
      const password = document.getElementById("password")
      const confirmPassword = document.getElementById("confirm_password")
      const eventInterest = document.getElementById("event_interest")

      // Clear previous error messages
      const errorElements = document.querySelectorAll(".error")
      errorElements.forEach((element) => {
        element.textContent = ""
      })

      // Validate first name
      if (firstName.value.trim() === "") {
        showError(firstName, "First name is required")
        hasError = true
      } else if (!/^[a-zA-Z ]+$/.test(firstName.value.trim())) {
        showError(firstName, "Only letters and white space allowed")
        hasError = true
      }

      // Validate last name
      if (lastName.value.trim() === "") {
        showError(lastName, "Last name is required")
        hasError = true
      } else if (!/^[a-zA-Z ]+$/.test(lastName.value.trim())) {
        showError(lastName, "Only letters and white space allowed")
        hasError = true
      }

      // Validate contact
      if (contact.value.trim() === "") {
        showError(contact, "Contact number is required")
        hasError = true
      } else if (!/^[0-9]{10}$/.test(contact.value.trim())) {
        showError(contact, "Contact must be 10 digits")
        hasError = true
      }

      // Validate email
      if (email.value.trim() === "") {
        showError(email, "Email is required")
        hasError = true
      } else if (!/^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/.test(email.value.trim())) {
        showError(email, "Invalid email format")
        hasError = true
      }

      // Validate password
      if (password.value === "") {
        showError(password, "Password is required")
        hasError = true
      } else if (password.value.length < 8) {
        showError(password, "Password must be at least 8 characters")
        hasError = true
      }

      // Validate confirm password
      if (confirmPassword.value === "") {
        showError(confirmPassword, "Please confirm your password")
        hasError = true
      } else if (password.value !== confirmPassword.value) {
        showError(confirmPassword, "Passwords do not match")
        hasError = true
      }

      // Validate event interest
      if (eventInterest.value === "") {
        showError(eventInterest, "Please select an event")
        hasError = true
      }

      // Prevent form submission if there are errors
      if (hasError) {
        event.preventDefault()
      }
    })
  }

  // Client-side validation for login form
  const loginForm = document.getElementById("loginForm")

  if (loginForm) {
    loginForm.addEventListener("submit", (event) => {
      let hasError = false

      // Get form fields
      const email = document.getElementById("email")
      const password = document.getElementById("password")

      // Clear previous error messages
      const errorElements = document.querySelectorAll(".error")
      errorElements.forEach((element) => {
        element.textContent = ""
      })

      // Validate email
      if (email.value.trim() === "") {
        showError(email, "Email is required")
        hasError = true
      } else if (!/^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/.test(email.value.trim())) {
        showError(email, "Invalid email format")
        hasError = true
      }
    })
  }
})

```



```
logout.php

<?php
session_start();

// Unset all session variables
$_SESSION = array();

// Destroy the session
session_destroy();

// Redirect to login page
header("Location: login.php");
exit();
?>
```





Output:

## Login

Email

Password

Login

Don't have an account? [Register](#)

Welcome, Arman singh!

[Home](#)

[Events](#)

[My Events](#)

[About](#)

[Logout](#)

### Create Your Own Events

Share your passion with others by hosting your own events

[Create Event](#)

### Your Profile

Profile picture updated successfully!



[Update Profile Picture](#)

#### User Information

Name: Arman singh

Email: maxray66999@gmail.com

Contact: 9861451517

Event Interest: Film

Registered On: April 3, 2025



## Register

First Name

Last Name

Contact Number

Email

Password

Confirm Password

Event Interest

Select an event



Register

Already have an account? [Login](#)

### 3. FINAL OUTCOME

#### Key Features and Outputs

- **User Interface:**
  - Responsive, visually appealing layout.
  - Mobile-friendly design.
  - Interactive UI components.
- **Authentication System:**
  - Secure registration and login.
  - Password hashing for security.
  - Session-based authentication.
- **Event Management:**
  - Dynamic event listings with filters.
  - Event pages with detailed descriptions and booking options.
- **Database Implementation:**
  - Optimized MySQL schema with foreign key relationships.
  - Secure queries using prepared statements.
- **Booking System:**
  - Real-time ticket availability updates.
  - User dashboard for managing bookings.

#### 6. LEARNING OUTCOME **PHP Development Skills:**

- Setting up a PHP development environment.
- Implementing form validation and session management.
- Secure authentication using password hashing.
- Applying Object-Oriented Programming (OOP) principles in PHP.

#### **Database Design and Management:**

- Designing a normalized database schema.
- Using foreign keys for relational integrity.
- Implementing secure database queries.

**Frontend Development:**

- Creating responsive layouts using CSS and JavaScript.
- Integrating UI libraries like Bootstrap and Font Awesome.

**Security Implementation:**

- Input sanitization to prevent XSS attacks.
- Role-based access control for user authentication.

**Project Management & Problem-Solving:**

- Structuring code for maintainability and scalability.
- Debugging and resolving integration issues.

**Conclusion:** The Event Management System successfully integrates multiple technologies into a functional web application. It streamlines event management processes, enhances user engagement, and serves as a practical demonstration of full-stack web development techniques.