

## Event Management System

---

## Objectives

- To develop a simple web-based event management system.
  - To implement **Admin and User login** functionality.
  - To allow **Admin to manage events** (Create, Read, Update, Delete).
  - To display a **participant list** for each event.
  - To store and view data using **MySQL** or **JSON**.
  - To gain hands-on experience with **HTML, CSS, JavaScript, Node.js, and MySQL** integration.
- 

## Tools and Technologies Used

Category	Tools/Technologies
Frontend	HTML, CSS, JavaScript
Backend	Node.js, Express.js
Database	MySQL
Hosting	GitHub Pages / Netlify (for frontend), MySQL locally
Code Editor	Visual Studio Code
Version Control	Git & GitHub

---

## System Modules

### Admin Side

- **Login Dashboard**
  - Manage events (**CRUD**: Create, Read, Update, Delete)
  - View participant list per event
  - Role-based access (Admin vs User)
- 

## Code Snippets

### 1 Database Connection (Node.js)

```
const mysql = require('mysql2');
const con = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root',
```

```
        database: 'eventdb'
    });
con.connect(err => {
    if (err) throw err;
    console.log("MySQL Connected!");
});
```

---

## 2 Register User

```
app.post('/register', (req, res) => {
    const { username, password, role } = req.body;
    const sql = "INSERT INTO users (username, password, role) VALUES (?, ?, ?)";
    con.query(sql, [username, password, role], (err) => {
        if (err) throw err;
        res.send('✓ Registered Successfully');
    });
});
```

---

## 3 Admin CRUD (Example)

```
app.post('/add-event', (req, res) => {
    const { name, date, location } = req.body;
    con.query("INSERT INTO events (name, date, location) VALUES (?, ?, ?)", [name, date, location], (err) => {
        if (err) throw err;
        res.send('Event Added!');
    });
});
```

---

# Database Tables

### ◆ Table: `users`

**id username password role**

1	admin	admin123	admin
2	user1	user123	user

### ◆ Table: `events`

**id name date location**

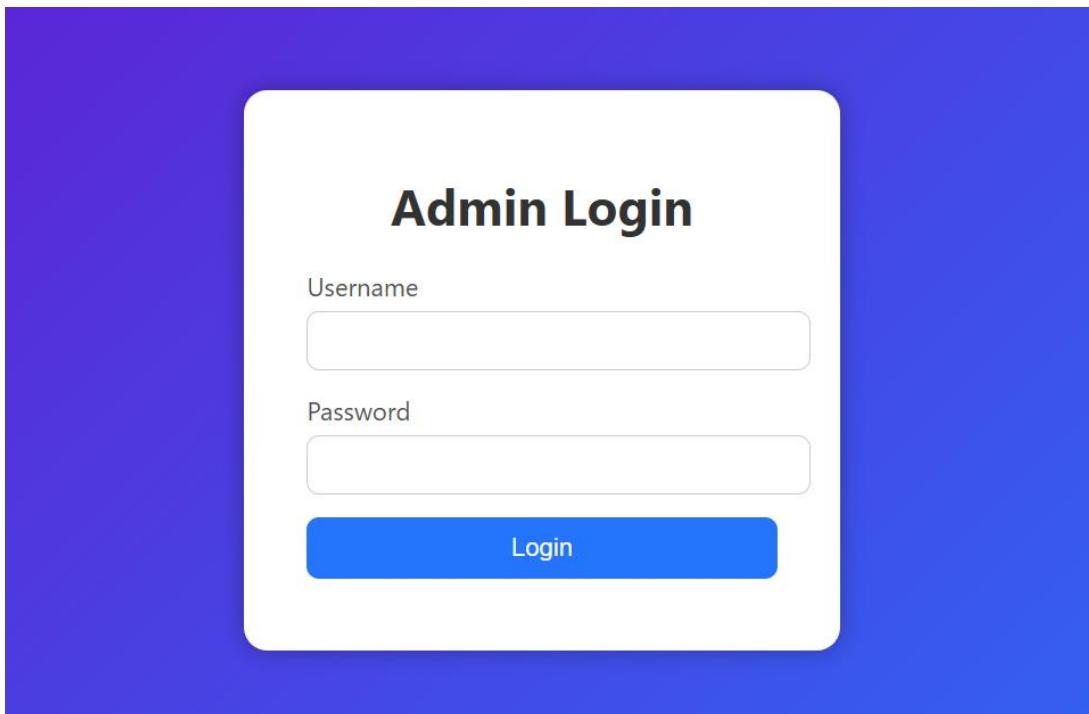
1	Tech Talk	2025-11-15	Seminar Hall
---	-----------	------------	--------------

### ◆ Table: `participants`

**id name event\_id**

1	Neha	1
---	------	---

1. Create a **GitHub repository**
  2. Upload all project files (HTML, CSS, JS, JSON, Documentation.pdf)
  3. Go to **Settings → Pages → Deploy from main branch**
  4. After a few seconds, access your hosted project at:
  5. <https://yourusername.github.io/event-management-system/>
  6. You can open your documentation at:
  7. <https://yourusername.github.io/event-management-system/Documentation.pdf>
- 



A screenshot of an Admin Dashboard. The header is dark grey with the title "Admin Dashboard" in white. Below the header is a navigation bar with "Dashboard" and "Logout" links. The main content area has a light grey background and a white rounded rectangle. Inside this rectangle, the title "Manage Events" is at the top. Below it is a table with a light grey header row and two data rows. The table columns are "ID", "Event Name", "Date", "Category", and "Actions". The first data row shows ID 1, Event Name "Tech Fest", Date "2025-12-01", Category "Technology", and "Update" and "Delete" buttons. The second data row shows ID 2, Event Name "Health Camp", Date "2025-11-20", Category "Medical", and "Update" and "Delete" buttons. At the bottom of the white rectangle is a section titled "Add New Event" with three input fields: "Event Name", "dd-mm-yyyy" (with a date picker icon), and "Category", followed by a "Add Event" button.

# Welcome, Admin

## View Participants by Event

Tech Talk ▾

Tech Talk

AI Workshop

**Neha Sharma**

neha@gmail.com

**Ravi Kumar**

ravi@gmail.com