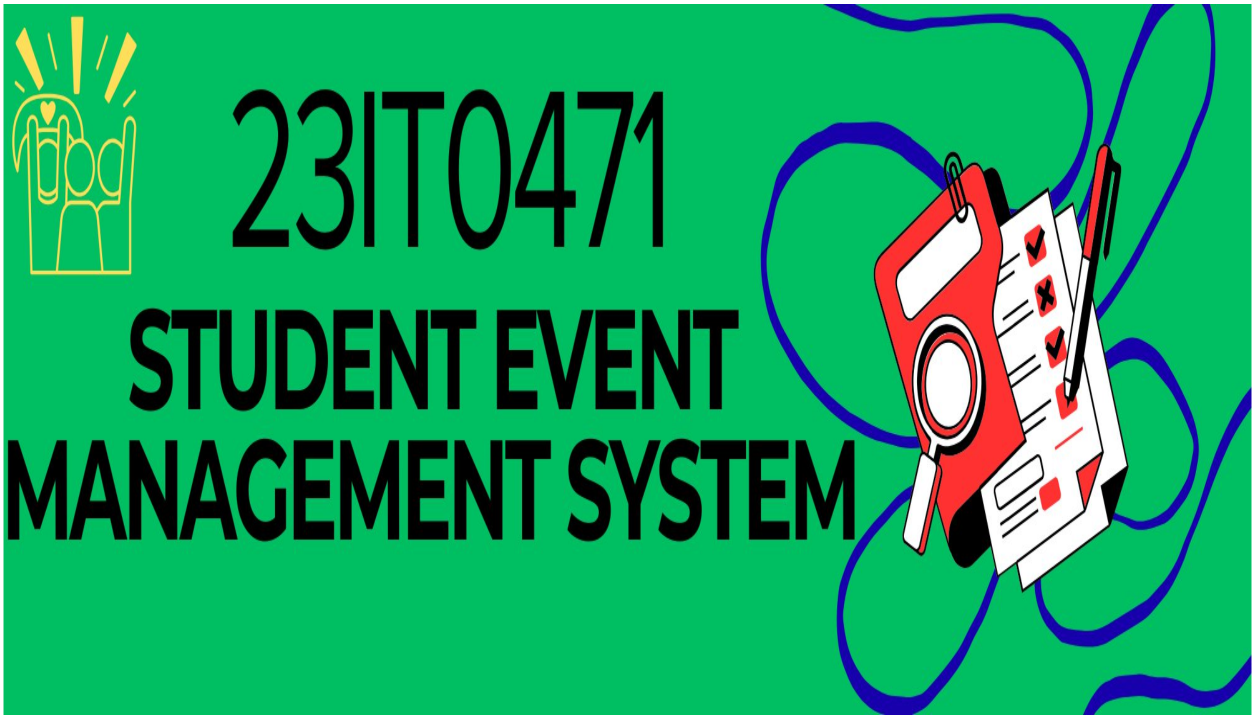
Student Event Management System

IT1208 – Web Technologies

Continuous Assessment 2: Short Report

Name: Sivaranjan Dinath
Student ID: 23IT0471
Database Used: MariaDB
Languages & Tools: HTML, CSS, JavaScript, PHP, MariaDB
Development Environment: Fedora Linux + Visual Studio Code



1. System Overview

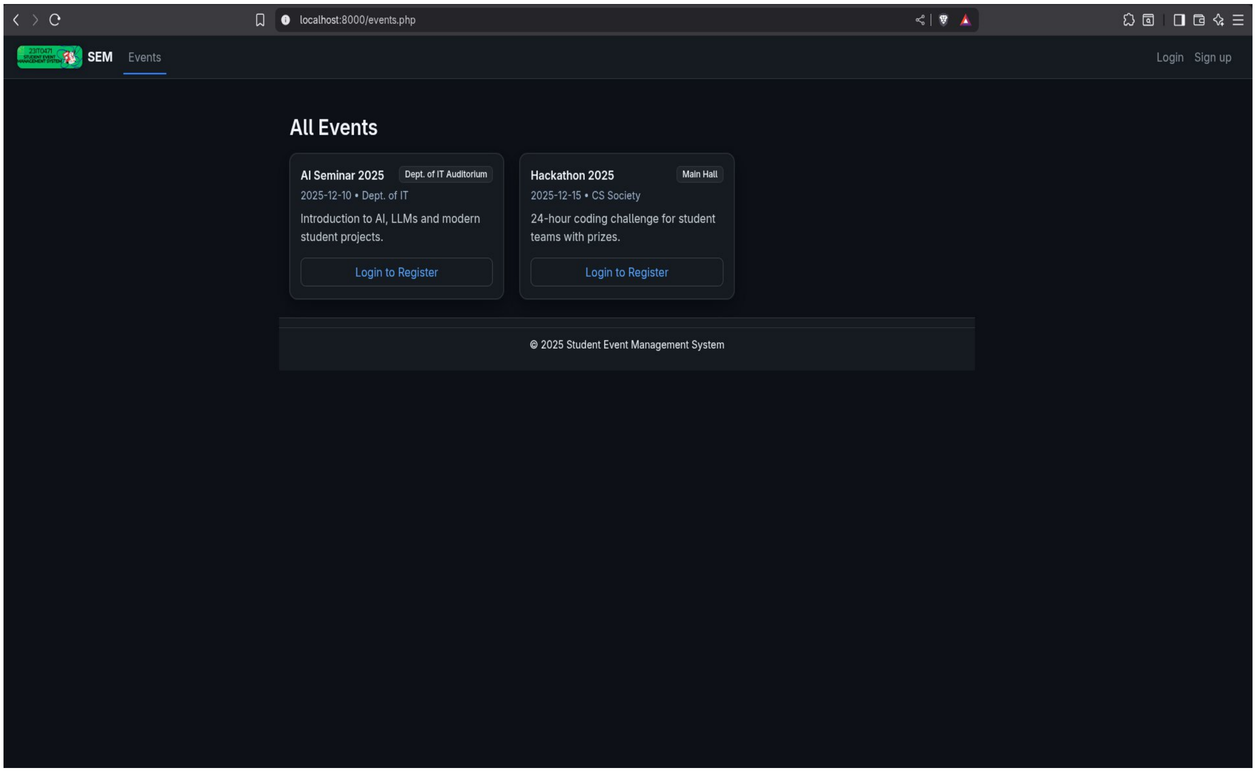
The *Student Event Management System* is a dynamic web application designed to streamline the process of managing university events.
It allows students to browse upcoming workshops, hackathons, and seminars, view details such as event date and venue, and register easily.
Administrators can create, edit, and delete events, while students can register and view participation status.

The system is developed using **HTML, CSS, JavaScript, PHP, and MariaDB**.
It includes both front-end interactivity and back-end database integration to ensure smooth user experiences and secure data management.

Main Features:

- Responsive and modern UI with a dark-themed design.
- Event CRUD operations (Create, Read, Update, Delete).
- Student registration form with client-side JavaScript validation.
- Login and signup system using PHP sessions.
- Data stored securely in a MariaDB database using PDO.
- Event images, dates, and organizers displayed attractively.

Figure 1: *Home Page with Event Cards*



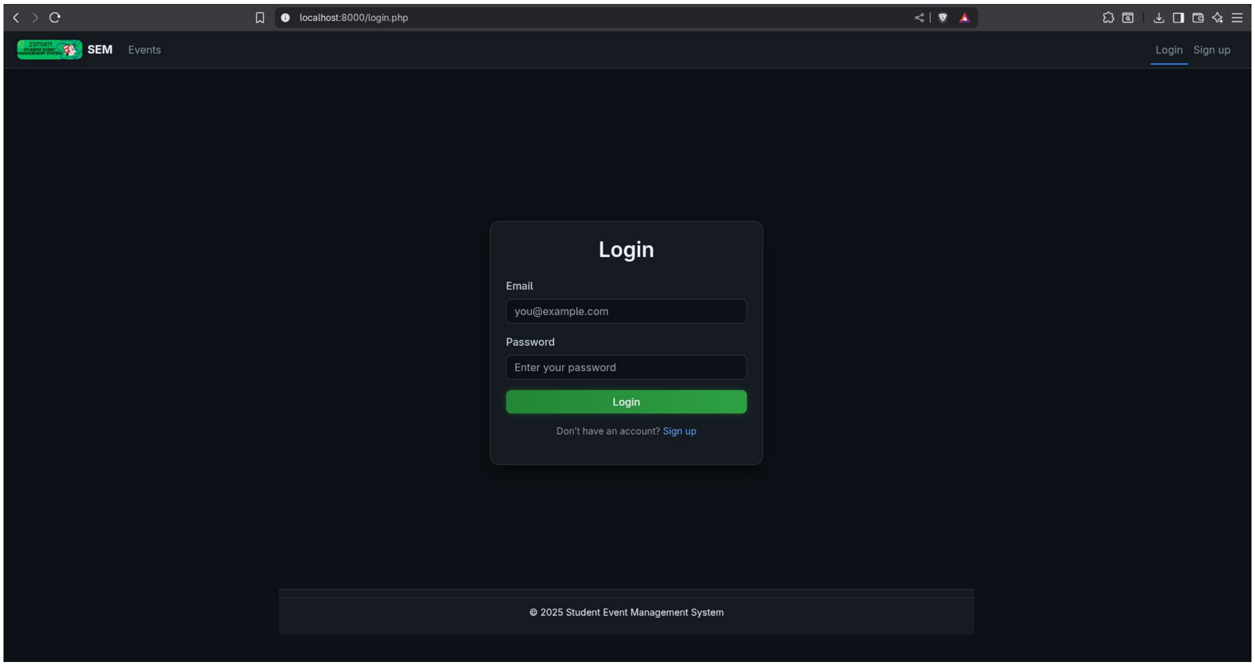
All Events

AI Seminar 2025 — Dept. of IT Auditorium
2025-12-10 • Dept. of IT
Introduction to AI, LLMs and modern student projects.
Login to Register

Hackathon 2025 — Main Hall
2025-12-15 • CS Society
24-hour coding challenge for student teams with prizes.
Login to Register

© 2025 Student Event Management System

Figure 2: *Login and Registration Interface*



Login

Email

you@example.com

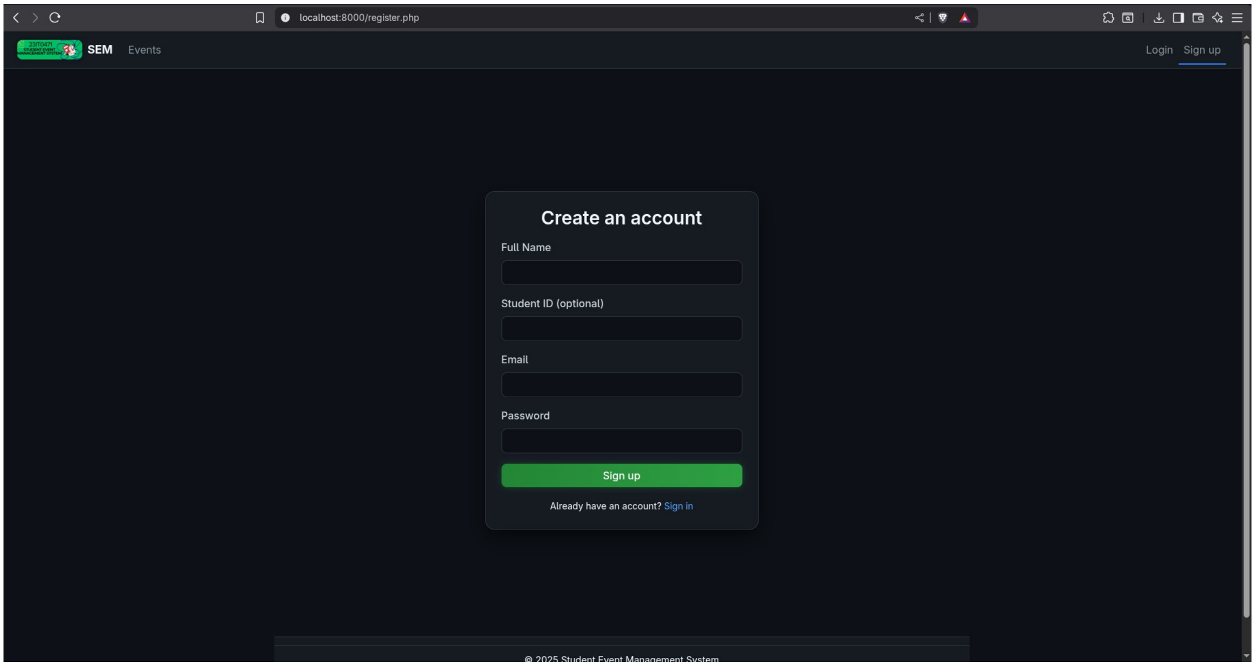
Password

Enter your password

Don't have an account? Sign up

© 2025 Student Event Management System

Figure 3: *Add Event Form with Image Upload Option*



Create an account

Full Name

Student ID (optional)

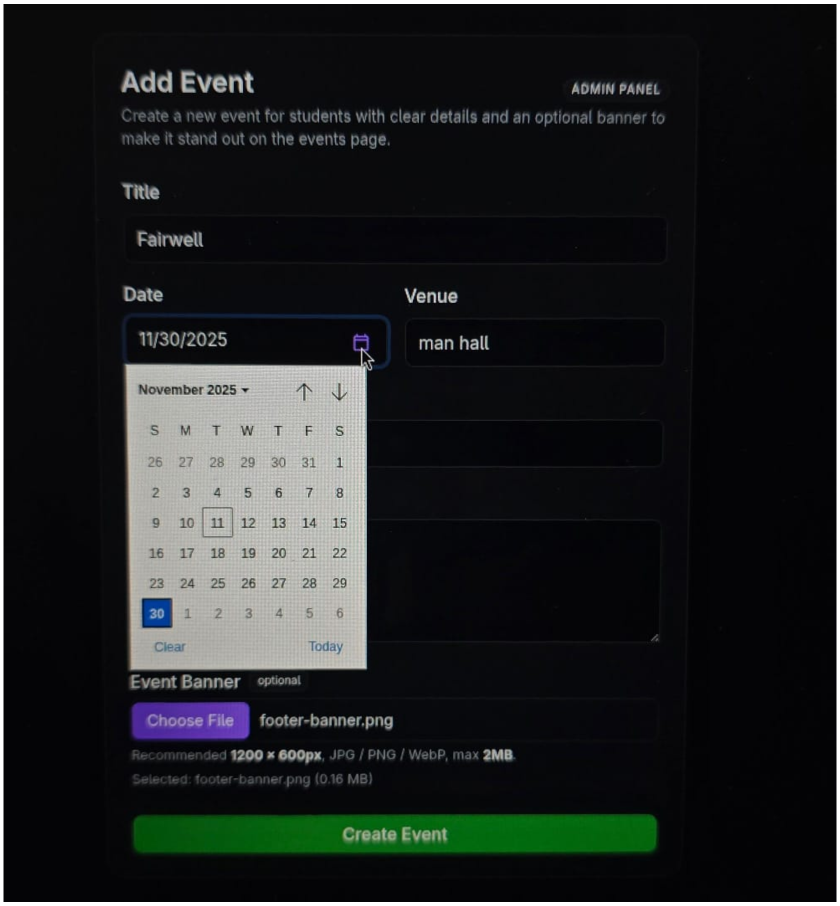
Email

Password

Already have an account? Sign in

© 2025 Student Event Management System

Figure 4: *Event List Display with Details and Calendar Icon*



Add Event

Create a new event for students with clear details and an optional banner to make it stand out on the events page.

Title

Fairwell

Date

11/30/2025

Venue

man hall

November 2025

S	M	T	W	T	F	S
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Event Banner optional

footer-banner.png

Recommended **1200 × 600px**, JPG / PNG / WebP, max **2MB**.

Selected: footer-banner.png (0.16 MB)

2. Database Schema Diagram

The database consists of **three core relational tables** linked through foreign keys.

Tables and Relationships:

- **users** (`user_id`, `name`, `email`, `password`)
- **events** (`event_id`, `title`, `date`, `venue`, `description`, `image`)
- **registrations** (`reg_id`, `user_id`, `event_id`, `registered_at`)

Relationships:

- One user can register for many events → **1 : N** (users → registrations).
- One event can have many registrations → **1 : N** (events → registrations).

This structure ensures data normalization, efficient queries, and secure CRUD operations using PDO prepared statements.

3. Core Functionalities

A. Front-End (HTML, CSS, JavaScript)

- Responsive design built with custom CSS.
- Client-side validation for forms (name, email, ID, contact).
- Calendar date picker with purple icon for event creation.
- Dynamic event listing with image banners and details.
- Smooth hover animations and fade-in transitions.

B. Back-End (PHP & MariaDB)

- PHP handles authentication, sessions, and database CRUD.
- PDO ensures secure queries to prevent SQL injection.
- Admin users can create, update, or delete events.
- Student registrations are inserted into the `registrations` table.
- Data retrieved dynamically for event listings and dashboard pages.

C. Optional Enhancements

- Search and filter functionality using PHP and JavaScript.
- Automatic event date highlight and sorting.
- Prepared for future features like email confirmation.

4. Reflection on Learning Outcomes

Through this project, I learned how to integrate multiple web technologies to create a complete, interactive application.
I gained hands-on experience in designing responsive interfaces, handling form validation with JavaScript, and using PHP to manage server-side logic.
Working with MariaDB helped me understand relational database design and query optimization.

I also learned:

- How to structure and connect PHP with a MySQL/MariaDB database using PDO.
- How to create secure login systems using sessions.
- How to use HTML5 and CSS3 to create accessible, user-friendly interfaces.
- How to manage GitHub repositories and run projects on Fedora using the PHP development server.

The project improved my full-stack development skills and understanding of real-world web application workflows.

5. Challenges Faced

Some key challenges I faced during development include:

- Resolving MariaDB permission errors (`Access denied for 'root'@'localhost'`).
- Designing a consistent dark UI that looks clean on all screens.
- Managing database connection errors and ensuring proper PDO error handling.
- Testing responsiveness on different browsers and devices.

Each challenge helped me learn valuable debugging and troubleshooting skills, making this project a strong foundation for future web development work.

6. Conclusion

The **Student Event Management System** successfully meets all assignment requirements.
It integrates front-end design, back-end logic, and database interaction into a fully functional, responsive web application.
The system demonstrates a solid understanding of web technologies and can be extended with features such as email confirmation and dashboards in future versions.