
ONLINE EVENT REGISTRATION SYSTEM

Dr.I.Shailaja

Associate Professor, Department of Computer Science & Engineering, Anurag Engineering college, kodada, suryapeta, Telangana.

Submitted: 19-04-2025

Accepted: 16-05-2025

Published: 23-05-2025

Abstract :In many educational institutions, the process of event registration remains largely manual, relying on paper-based forms or basic online tools that lack proper validation and functionality. These outdated approaches are inefficient, error-prone, and do not scale well when handling multiple events or a large number of participants. They also make it difficult to track attendance, manage capacity limits, or provide immediate feedback to participants. To address these challenges, this project proposes an **Online Event Registration System**—a secure, scalable, and user-friendly web-based platform developed using PHP, MySQL, HTML, and CSS. The system allows students and staff to view and register for college events from any device. Admins can create, update, and delete events, track real-time registration data, and generate participant reports with just a few clicks. Role-based access ensures that only authorized users can manage event data. The system is deployed on XAMPP and follows a modular, layered architecture. It significantly reduces manual workload, improves data accuracy, and provides a centralized solution for managing institutional events efficiently.

This is an open access article under the creative commons license
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



1.INTRODUCTION:

Events such as workshops, technical fests, guest lectures, and cultural programs are an integral part of academic life. However, managing registrations for these events is often a tedious task. In most colleges, this is still done through physical sign-up sheets or using unstructured digital methods like shared spreadsheets or Google Forms. These methods are inefficient, vulnerable to errors, and provide little to no control over registration limits or duplicate entries.

An effective event registration system is essential to ensure smooth coordination, timely communication, and proper data handling. Without a reliable registration platform, institutions face problems such as overcrowding, poor participant tracking, delayed communication, and lack of historical data for event planning.

The **Online Event Registration System** addresses these issues by offering a complete digital solution. It enables students and faculty to register online, receive instant confirmation, and ensures that event organizers have accurate and up-to-date information. By providing role-based access, automated confirmations, and structured data storage, the system enhances operational efficiency and improves the user experience for both participants and administrators.

2.LITERATURE SURVEY:

Before designing the **Online Event Registration System**, several existing tools and methods were studied, including paper-based registration, Google Forms, mobile registration apps, and basic spreadsheet tracking.

Comparative Table:

S.No	Author / System	Method / Tool Used	Key Findings / Limitations
1	Sharma et al. (2022)	manual registrations (papers)	Prone to human errors, time consuming, no real time updates
2	Google Forms in Colleges	Cloud forms, no login	No authentication, allows duplicates, lacks report features
3	Ravi et al. (2023)	Mobile registration app	Good UX, but needs internet and Android device
4	Excel and shared drives	Spread sheets based forms	Data scattered, no centralized access, no auto confirmation
5	Proposed OSFS (2025)	PHP/MySQL based web systems	Role-based access, limit checks, real time cofirmations,

Analysis:

The study of existing systems reveals that while current tools may suffice for very small-scale events, they fail to meet the demands of larger or more frequent event coordination efforts in colleges and institutions. Manual forms, google forms, mobile apps, spread sheets are not ideal. From these limitations, it became clear that the ideal system must:

- Work offline or on college LAN
 - Enforce one-time feedback per course
 - Allow real time tracking of capacity and participants
 - Store data in a structured format (MySQL) for easy reporting
 - Provide a simple yet secure interface for both users and admin
- These points directly shaped the architecture and goals of OSFS.

3. EXISTING SYSTEM:

The traditional event registration process used in colleges is largely manual or semi-digital. Paper forms are filled out by students and collected by coordinators who then manually compile the data. In slightly modern setups, platforms like Google Forms are used, but they often lack authentication and data validation.

Key Limitations of Existing Systems:

- Lack of Authentication: Anyone with access to the link can register, which leads to spam or fake registrations.
- Duplicate Entries: Students can submit multiple responses, distorting the data.

- Manual compilation: Registrations are compiled manually using Excel, increasing the chances of errors.
- Limited Functionality: Google Forms don't support capacity tracking or automatic confirmation messages.
- Scalability Issues: These systems are not suitable for managing multiple events simultaneously.

Real-World Example:

A common situation in most colleges is the use of paper-based event registrations forms. These forms are later handed to the admin office, where data entry clerks manually type each line into Excel — a process that may take days and introduces human error.

These outdated systems fail to capture the full value of online event registration and affects students from participating as manual registration is tedious task while online registrations can be made eye catching and easier to register . Hence, a need for a secure, digital, structured, and easy-to-manage system arises.

4. PROPOSED SYSTEM:

The **Online Event Registration System** is proposed as a secure, role-based, and efficient web application to overcome the shortcomings of manual and semi-digital registration methods.

Key Objectives:

- Provide a single platform to view and register for events.
- Eliminate duplicate entries and enforce registration limits.
- Give real-time access to participant data for administrators.
- Automate communication through confirmation messages

System Roles & Functionalities:

1. Participant Module:

- Receive confirmation on-screen or via email.
- Fill a registration form for selected events.
- Add optional comments
- Receive confirmation on-screen or via email.
- Access is open, no login required (can be added optionally).

2. Admin Module:

- Login securely
- Add, edit, and delete events with registration limits
- View event-wise participant lists.
- Export data as Excel or PDF
- Track event popularity using participant count.

Tech Stack:

- **Frontend:** HTML5, CSS3, JS (optional enhancements later)

- **Backend:** PHP 7+ with secure session handling
- **Database:** MySQL with relational tables
- **Hosting:** Local server via XAMPP (Apache + MySQL)

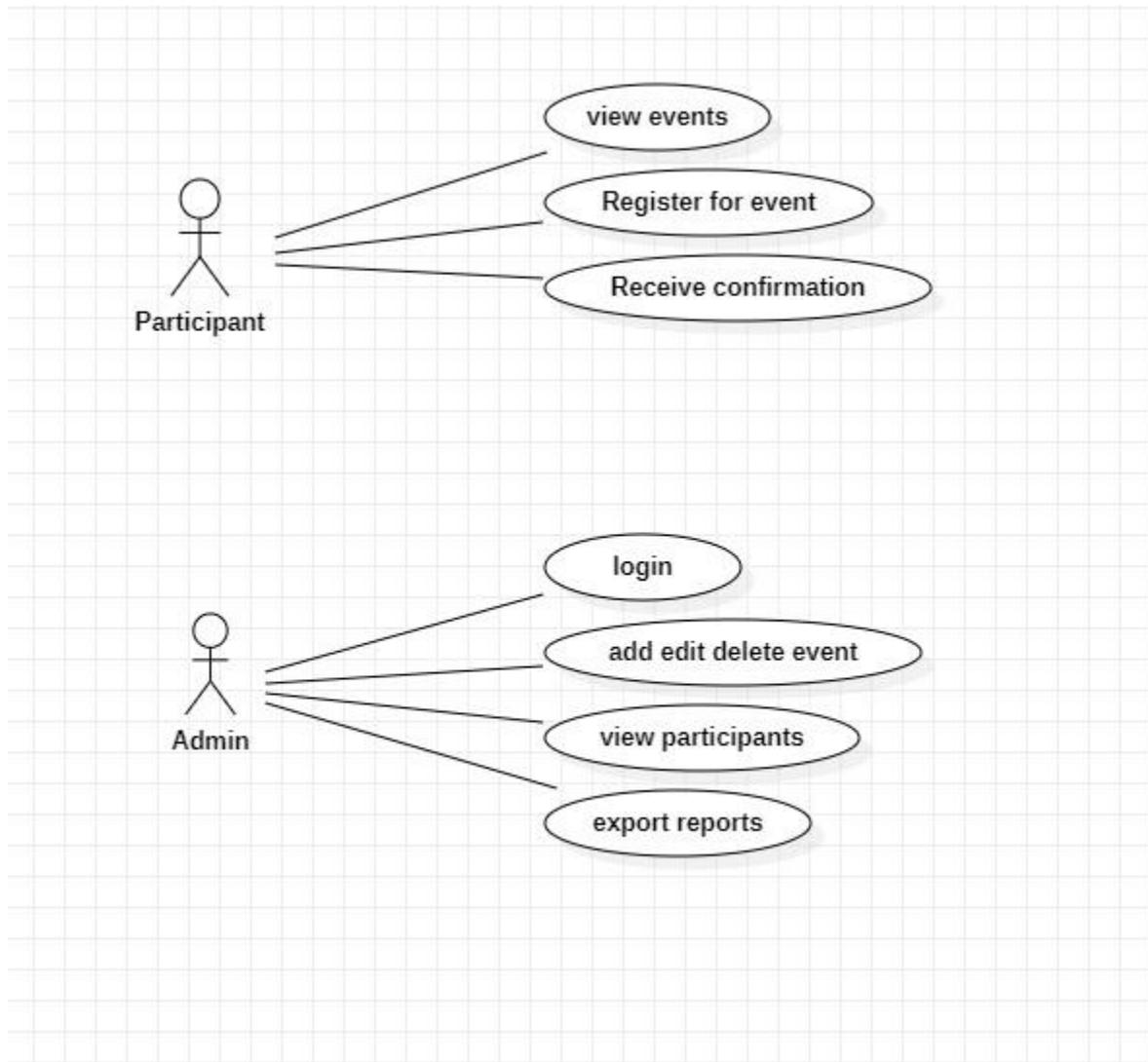


Figure 1: Use Case Diagram

5. SYSTEM DESIGN

The design of OEVS focuses on modularity, security, and role-based functionality. It follows a standard web application architecture with separation of concerns (frontend, backend, and database layers).

a) Logical Architecture

Layers:

1. Presentation Layer: HTML/CSS pages for user interface.

2.Application Layer (Controller): PHP scripts handle form submissions, validation, and business logic.

3.Data Layer: MySQL handles data storage for users, events, and registrations.

Login Flow (for admin):

- Admin enters credentials.
- PHP session is initiated
- Admin is redirected to the dashboard.

Feedback Flow:

- User selects an event.
- Enters personal details.
- Form is submitted to the backend.
- System checks event capacity and duplicate entry.
- Data is saved, and confirmation is displayed.

b) Physical Architecture

User (web browser) → Apache Server (PHP) → MySQL Database (events + users + registration)

- Hosted locally using XAMPP or on a LAN.
- Can be accessed via http://localhost/event_register/.

Security & Access Control

- Admin login is password-protected (session-based).
- SQL injection is prevented through sanitization.
- Duplicate entries blocked using unique constraints.
- Event limits enforced via server-side validation.

The architecture diagram below was created using draw.io, a free and open-source diagram tool.

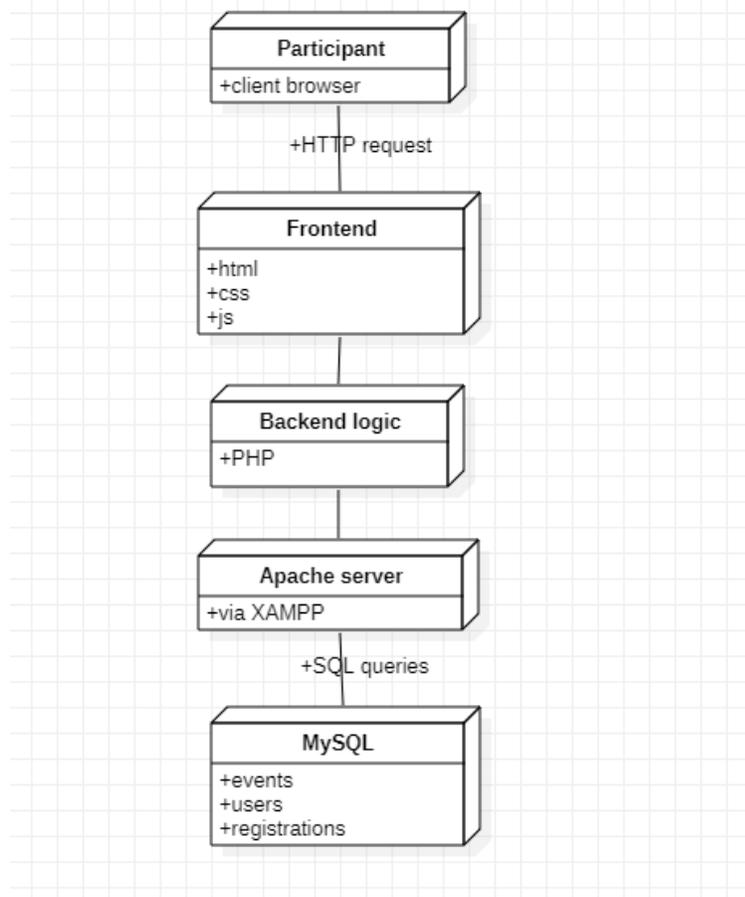


Figure 3: System Architecture Diagram

6.

7. IMPLEMENTATION:

Frontend (UI):

- Participant view:
 - Simple, mobile-friendly event list with registration form.
- Admin view:
 - Login panel
 - Dashboard with buttons to create, edit, delete events
 - Table view of participants

Backend (PHP):

- Login sessions, Prevent duplicate registration

Database Tables:

- users (optional for login)
- events (event_id, name, date, time, limit, description)

- registrations (reg_id, name, email, event_id, timestamp)
- admin (admin_id, username, password)

Functionality Summary:

Module	Feature
User	View events, Register, Receive confirmation
Admin	Login, Add, Edit, Delete, view, Track
Data Storage	Manage data, Structured schema, Timestamps

The diagram below was created using StarUML, a modern UML modeling tool.

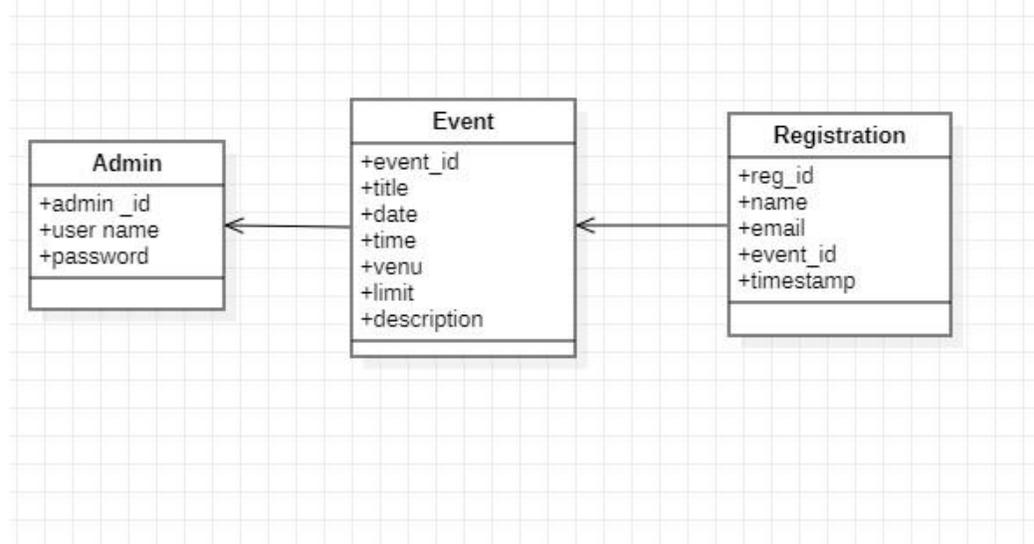


Figure 2: ER Diagram (Class Based)

8. RESULTS & DISCUSSION

The Online Event Registration System was tested locally using XAMPP on a Windows-based environment and also on a local LAN in a lab setting. A sample database was populated with mock participants, events, and admin to test core functionalities.

Functional Results:

- Login: Admin login works using session-based access.
- Duplicate Prevention: Repeated submissions from the same user for one event are blocked.
- Capacity Enforcement: Registration closes automatically when the limit is reached.
- Admin Reports: Participant data can be viewed, sorted, and exported.
- Stability: Tested successfully on localhost and college LAN.

Discussion:

The system performs well in real-world test environments. Event coordinators save time, and participants experience smooth registration. Data is consistent and secure. The interface is simple enough for non-technical users to operate.

Future testing with larger participant groups can help validate load performance and optimize database queries for scale.

9. CONCLUSION AND FUTURE SCOPE

Conclusion:

The **Online Event Registration System** effectively replaces the outdated manual method of managing event registrations. It offers:

- A clean and responsive interface
- Real-time data access
- Role based security
- Reduction in human error and time consumption
- Better control for event organizers

It also serves as a practical example of full-stack web development using open-source tools.

Future Scope:

1. QR Code Check-In: Generate codes for participants and scan them during entry.
2. SMS/Email Notifications: Send reminders or updates to registrants.
3. Calendar Integration: Add events to Google Calendar.
4. Mobile App: Create an Android version using Flutter or React Native.
5. Online Payment Gateway: For events with paid registrations.
6. Multilingual Support: Add language options for broader accessibility

10. REFERENCES

1. Sharma, P. et al. (2022). "Modernizing Student Feedback with Digital Platforms." *International Journal of Advanced Research in Computer Science (IJARCS)*.
2. Mozilla Developer Network. (2023). "HTML, CSS, and JavaScript Documentation." Available at: <https://developer.mozilla.org>
3. MySQL. (2023). "MySQL 8.0 Reference Manual." Available at: <https://dev.mysql.com/doc>
4. PHP Group. (2023). "PHP Manual." Available at: <https://www.php.net/manual>
5. Diagrams.net. (2023). "draw.io – Online Diagram Tool." Available at: <https://draw.io>
6. StarUML. (2023). "StarUML – Modern UML Tool." Available at: <https://staruml.io>