CAPSTONE PROJECT REPORT

CampusConnect – College Event Management System

1. Introduction

CampusConnect is a full-stack web application developed as part of my capstone project (Task-5). The goal of the system is to provide a centralized digital platform for managing college events efficiently. It allows administrators to create and manage events while students can view and register for them online.

This project demonstrates practical implementation of full-stack development concepts including database design, authentication, backend logic, AJAX integration, and dashboard analytics.

2. Problem Statement

In many institutions, event registrations are still handled manually through forms or spreadsheets. This often leads to data duplication, lack of transparency, and difficulty tracking participation. There is also no easy way to generate insights or manage events in a structured manner.

CampusConnect addresses these issues by providing a secure and automated system where event management, registration tracking, and reporting are handled digitally.

3. Objectives of the Project

- Develop a real-world full-stack web application
- Implement secure login and role-based access
- Design a relational database with proper relationships
- Enable students to register for events online
- Provide administrators with analytics and control tools
- Prepare the system for live deployment

4. Technology Stack

Frontend: HTML, CSS, JavaScript
Backend: PHP
Database: MySQL
Tools & Libraries: AJAX for real-time search, Chart.js for analytics, XAMPP for development

This combination provides a lightweight yet powerful environment for building scalable web applications.

5. System Architecture

CampusConnect follows a three-tier architecture:

Presentation Layer → User interface built with HTML/CSS/JS
Application Layer → PHP processes requests and business logic
Data Layer → MySQL stores and manages structured data

This separation ensures maintainability, scalability, and modular development.

6. Database Design

The system uses three primary tables:

Users Table: Stores login credentials and user roles (Admin/Student).
Events Table: Stores event details including title, description, date, and location.
Registrations Table: Links users to events, enabling a many-to-many relationship.

This structure ensures data integrity while allowing flexible expansion of the system.

7. Key Features Implemented

Authentication System
- Secure login and registration
- Password hashing
- Role-based access control

Admin Functionalities
- Create and delete events
- View student registrations
- Access analytics dashboard

Student Functionalities
- View available events
- Register for events
- Prevent duplicate registrations
- View personal registered events

Advanced Features
- Real-time event search using AJAX

- Data visualization using Chart.js
- Structured modular project architecture

8. Testing and Results

The application was tested under multiple scenarios including user authentication, event creation, registration workflow, and database validation. All major modules performed correctly and the system handled duplicate registration prevention and role-based routing successfully.

The interface remained responsive and functional across different browsers.

9. Learning Outcomes

Through this project, I gained hands-on experience in:

- Full-stack application development
- Database relationships and normalization
- Secure authentication handling
- AJAX integration for dynamic interfaces
- Debugging backend errors and session conflicts
- Structuring projects for deployment readiness

10. Conclusion

CampusConnect demonstrates the development of a professional full-stack web application capable of solving real-world event management challenges. The project highlights backend development, database design, UI interaction, and analytics integration.

This capstone project significantly strengthened my practical understanding of web technologies and prepared me for industry-level software development roles.

By: Uma Ponugoti