Campus Event Management System

Technical Specification & Implementation Report

A Cross-Platform Event Management Application
Built Using Flutter (Frontend) & PHP (Backend)

Prepared By

Ajai R – Backend Lead

Chandrama Kumar – Backend Support & Database

Anjali Kumari – Flutter UI Developer

Arsath Barvez M – Flutter Integration & Documentation

Department of Computer Science & Engineering
February 2026

CONTENTS

1. Executive Summary

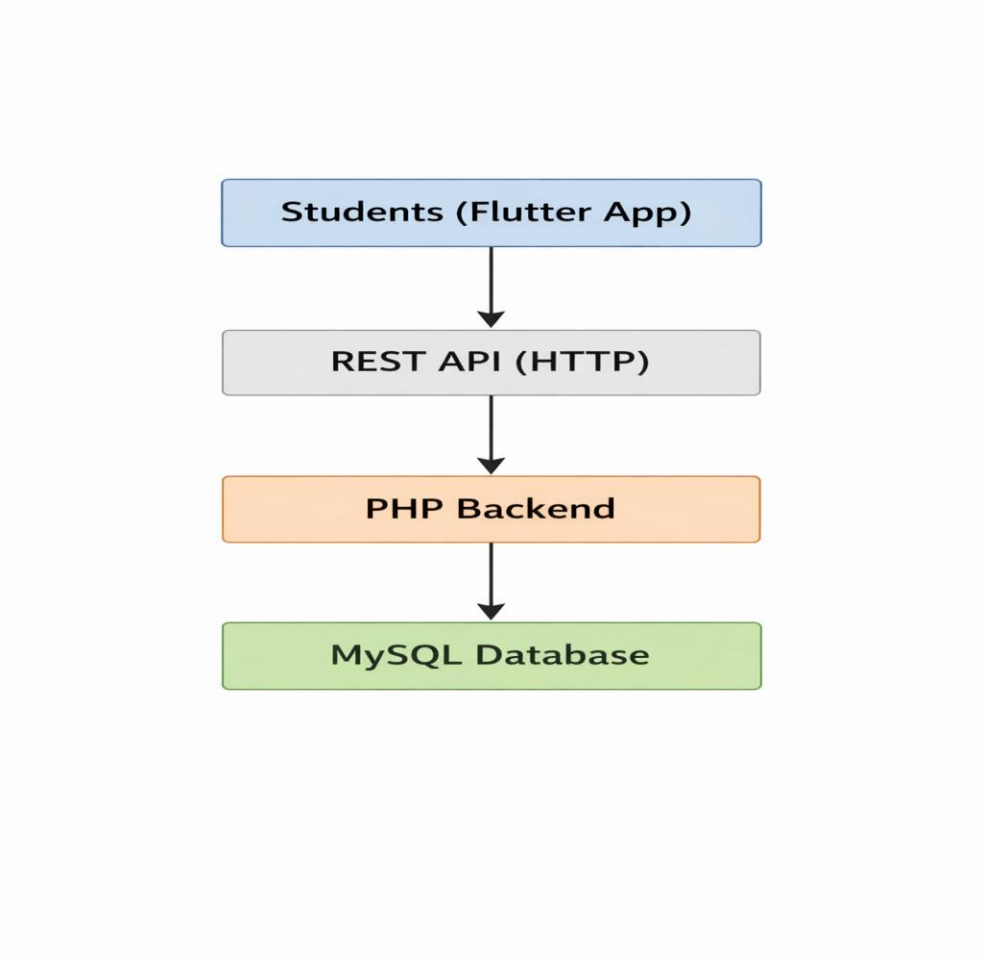
The Campus Event Management System is a digital platform developed to simplify the process of organizing and managing campus events. The system allows administrators to create and manage events, while students can easily register and track their participation using a mobile application.

This project is developed using Flutter for the frontend mobile application and PHP for the backend server. The system ensures smooth communication between the mobile app and the database through REST API integration.

The primary objective of this system is to eliminate manual event registration processes and replace them with a centralized, efficient, and real-time digital solution. The application provides features such as event creation, student registration, attendance tracking, and instant notifications.

The backend system handles user authentication, event data storage, and attendance records using a MySQL database. The Flutter application provides a user-friendly interface for students and administrators to interact with the system seamlessly.

This system improves transparency, reduces paperwork, enhances communication, and ensures proper event management within the campus environment.



2. System Overview

2.1 Introduction

The Campus Event Management System is a centralized digital platform developed to manage and streamline all campus-related events efficiently. The system replaces traditional manual registration methods with a secure and automated mobile-based solution.

The application enables administrators to create and manage events, while students can browse available events, register instantly, and track their attendance digitally.

2.2 Objectives of the System

- To digitalize campus event registration.
- To reduce paperwork and manual data entry errors.
- To maintain accurate attendance records.
- To send real-time notifications to students.
- To provide a transparent and organized event management process.

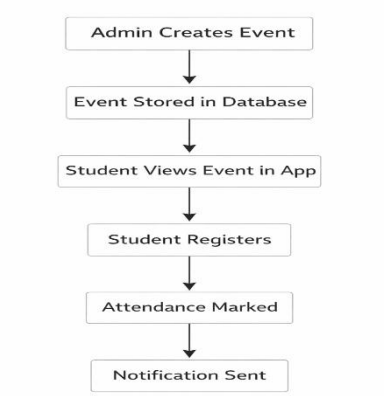
2.3 Working Flow of the System

The system operates using a client-server architecture. The Flutter mobile application acts as the client interface, while the PHP backend handles business logic and database operations. All data transactions are performed through REST API calls.

When a student registers for an event:

1. The Flutter app sends an HTTP request.
2. The PHP server validates and processes the request.
3. The data is stored in the MySQL database.
4. A confirmation response is sent back to the application.

This ensures secure and real-time data processing.



3. Technical Stack Architecture

3.1 Frontend – Flutter Mobile Application

The frontend of the Campus Event Management System is developed using Flutter, a cross-platform UI framework. Flutter enables the development of high-performance mobile applications for both Android and iOS using a single codebase.

The mobile application provides:

- Event listing interface
- Student registration forms
- Attendance status view
- Notification alerts

Flutter ensures a responsive user interface and smooth navigation across different screens.

3.2 Backend – PHP REST API

The backend is developed using PHP, which handles all server-side logic and API processing. The backend acts as a bridge between the Flutter application and the database.

Responsibilities of the PHP backend include:

- User authentication and validation
- Event creation and management
- Registration processing
- Attendance record handling
- Notification triggering

The backend communicates with the Flutter app using RESTful APIs over HTTP protocol.

3.3 Database – MySQL

The system uses MySQL as the relational database to store structured data such as student details, event information, and attendance records.

The database ensures:

- Data consistency
- Secure storage
- Efficient querying
- Relational integrity between tables

4.Core Modules & Features

4.1 Admin Module

The Admin Module is responsible for managing and controlling the overall event system. The administrator has full access to create, update, and delete event information.

Key Features:

- Create new campus events
- Edit event details (date, venue, description)
- Delete or cancel events
- View registered students list
- Monitor attendance records
- Send notifications to students

4.2 Student Module

The Student Module allows students to interact with the system through the Flutter mobile application.

Key Features:

- View available campus events
- Register for events
- View registration confirmation
- Check attendance status
- Receive event notifications

4.3 Attendance & Notification Module

This module manages event participation tracking and communication.

Attendance System:

- Marks student presence during the event
- Stores attendance data in the database
- Prevents duplicate entries

Notification System:

- Sends alerts for new events
- Sends reminders before event start time

5. Database Schema Overview

5.1 Introduction

The Campus Event Management System uses a MySQL relational database to store and manage structured data efficiently. The database is designed to maintain integrity between students, events, registrations, and attendance records.

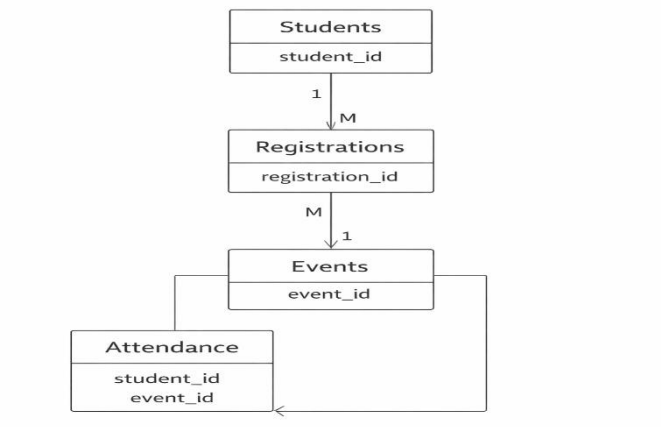
5.2 Main Tables and Relationships

Table Name	Primary Key	Description / Relationships
students	student_id	Stores student details (Name, Email, Dept, Year)
events	event_id	Stores event details (Title, Date, Venue, Description)
registrations	registration_id	Links students and events
attendance	attendance_id	Records student presence for events
notifications	notification_id	Stores event alerts and announcements

5.3 Relationship Explanation

- One student can register for multiple events.
- One event can have multiple registered students.
- Attendance records are linked to both student and event tables.
- Notifications are generated based on event updates.

This relational structure ensures data consistency and avoids redundancy.



6.Development Timeline

6.1 Project Development Plan (7–10 Days)

The Campus Event Management System was developed following a structured short-term development cycle. Each phase was divided based on module implementation and integration priorities.

Phase	Duration	Activities
Phase 1 – Requirement Analysis	Day 1	Requirement gathering, system planning, database design
Phase 2 – Backend Development	Day 2–4	PHP API creation, database integration, authentication setup
Phase 3 – Frontend Development	Day 5–7	Flutter UI design, registration screens, event listing screens
Phase 4 – Integration & Testing	Day 8–9	API integration, attendance testing, bug fixing
Phase 5 – Final Deployment	Day 10	Final testing, documentation, project submission

6.2 Role Distribution Among Team Members

Team Member	Role	Responsibilities
Ajai R	Backend Lead	API development, authentication, server logic
Chandrama Kumar	Backend Support & DB	Database design, query optimization
Anjali Kumari	Flutter UI Developer	UI design, screen development
Arsath Barvez M	Flutter Integration & Docs	API integration, documentation, report preparation

7.Testing & Validation

7.1 Testing Strategy

To ensure system reliability and performance, multiple testing methods were conducted during development. Both frontend and backend components were tested individually and collectively after integration.

The objective of testing was to:

- Verify system functionality
- Identify and fix bugs
- Ensure data accuracy
- Validate API communication
- Check system performance

7.2 Types of Testing Performed

- Unit Testing – Testing individual functions and API endpoints
- Integration Testing – Verifying Flutter and PHP communication
- Database Testing – Checking data storage and retrieval accuracy
- User Acceptance Testing (UAT) – Testing system usability with sample users

Test Case ID	Module	Test Scenario	Expected Result	Status
TC01	Admin Module	Create new event	Event stored successfully	Passed
TC02	Student Module	Register for event	Registration confirmation shown	Passed
TC03	Attendance	Mark attendance	Attendance updated in DB	Passed
TC04	Notification	Send event alert	Notification delivered	Passed

7.3 Validation Results

The system successfully met all functional requirements. API requests were processed correctly, database operations were accurate, and the mobile interface functioned without major issues. Minor UI bugs identified during testing were corrected before final deployment.

8.Conclusion

The Campus Event Management System successfully provides a digital solution for managing campus events efficiently and systematically. By integrating Flutter for the mobile frontend and PHP with MySQL for the backend, the system ensures smooth communication, secure data handling, and real-time processing.

The application eliminates manual registration processes and significantly reduces paperwork, errors, and administrative workload. Students can conveniently register for events, receive notifications, and track attendance through a user-friendly mobile interface.

The backend architecture ensures secure authentication, proper database relationships, and structured data management. The modular design also allows future enhancements without affecting the core functionality.

Future Enhancements

- QR Code-based attendance scanning
- Role-based login system (Admin / Student)
- Event analytics dashboard
- Push notification integration
- Cloud deployment for scalability

This project demonstrates the effective implementation of a client-server architecture using modern mobile and web technologies. The system is scalable, reliable, and suitable for real-time campus event management.