Campus Event Management System - Design Document

Problem Statement

The system aims to simplify the management of campus events across colleges. It allows administrators to create and manage events while providing students with an easy way to register, check in, and give feedback. The system also supports reporting on participation, attendance, and event feedback, helping colleges analyze engagement and plan future events more effectively.

Data to Track

The system must track: - Event details (name, type, date, college association) - Student details (name, email, college) - Registrations (student-event mapping) - Attendance (per registration) - Feedback (ratings and comments) - Colleges (institution-level data)

Database Schema (Tables/Entities)

Entities: - Colleges: id, name - Events: id, college_id, name, type, date, created_by - Students: id, name, email, college_id - Registrations: id, event_id, student_id, registered_at - Attendance: id, registration_id, attended - Feedback: id, registration_id, rating, comment Relationships: - Each event belongs to a college - Each student belongs to a college - Registrations link students to events - Attendance links to registrations - Feedback links to registrations

API Design (Endpoints & Purpose)

Key Endpoints: - POST /events → Create a new event - GET /events → List all events - POST /students → Register a student profile - POST /register → Register a student for an event - POST /attendance → Mark attendance for a student in an event - POST /feedback → Submit feedback (rating & comment) - GET /reports/registrations → Total registrations per event - GET /reports/attendance → Attendance percentage per event - GET /reports/feedback → Average feedback per event - GET /reports/event-popularity → Events sorted by registrations - GET /reports/student-participation → Events attended per student - GET /reports/top-students → Top 3 most active students

Workflows

1. Registration Workflow: - Admin creates an event → Students browse → Student registers → System saves registration. 2. Attendance Workflow: - On event day → Staff marks student attendance → Attendance stored against registration. 3. Reporting Workflow: - Queries run on registrations, attendance, and feedback → Reports generated for popularity, participation, and feedback analysis.

Assumptions & Edge Cases

Assumptions: - Event IDs are unique within a college. - Each student belongs to one college. - Feedback is optional, but rating must be 1–5 if given. Edge Cases: - Duplicate student registrations for the same event. - Students missing feedback submissions. - Events being canceled or rescheduled. - Handling attendance for students not registered.

Reports Required

1. Event Popularity Report → Sorted by number of registrations 2. Student Participation Report → Events attended per student 3. Attendance Report → Percentage of students attended per event 4. Feedback Report → Average feedback score per event Bonus: - Top 3 most active students - Reports filtered by event type (Workshop, Fest, Seminar)