

Design Document

Campus Event Management Reporting System

**PREPARED BY SAKSHI GUPTA
R22EQ090**

Problem Statement

Colleges organize many events (workshops, hackathons, fests, seminars), but managing them across multiple institutions is difficult. Staff require tools to create events, track registrations, and generate reports, while students need a simple way to register and check-in. This project provides a basic event reporting system that addresses these needs with a simple database + API-driven approach.

Assumptions & Decisions

- Event IDs are unique per college, not globally.
- A student can register for multiple events but not twice for the same event.
- Attendance can be marked as Present, Absent, or Late (extra edge case).
- Feedback is optional, rating scale = 1–5, with optional comments.
- Cancelled events will have an “inactive” status so reports remain consistent.
- The system will scale to ~50 colleges, ~500 students/college, ~20 events per semester.

Data to Track

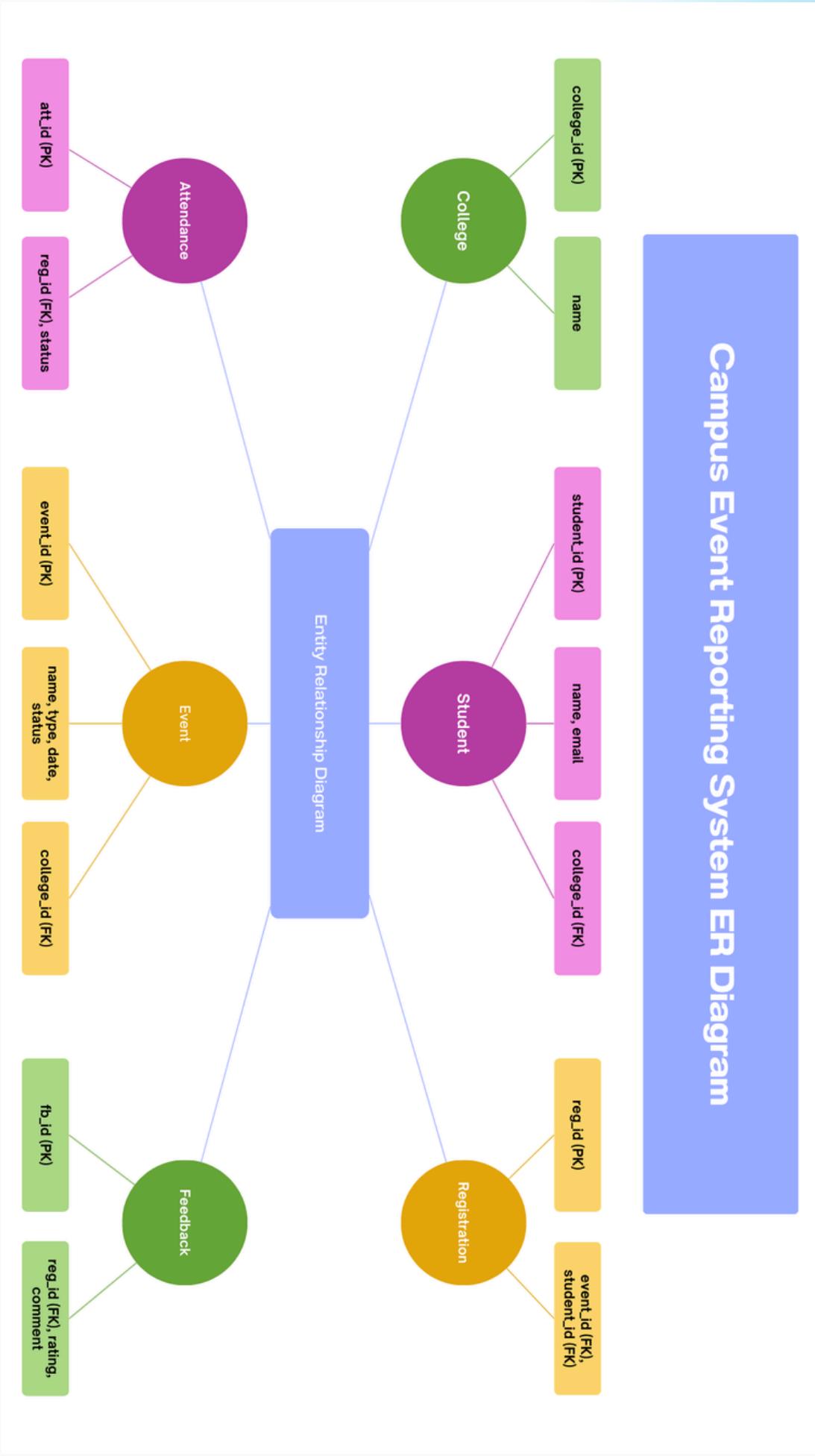
- Events → ID, name, type, date, status, college.
- Students → ID, name, email, college.
- Registrations → links student ↔ event.
- Attendance → status for each registered student.
- Feedback → rating (1–5) and optional comments.

Database Schema

- Colleges
- college_id (PK)
- name
- Students
- student_id (PK)
- name
- email
- college_id (FK)
- Events
- event_id (PK)
- college_id (FK)
- name
- type (Workshop, Fest, Seminar, Hackathon)
- date
- status (Active, Cancelled, Completed)
- Registrations
- reg_id (PK)
- event_id (FK)
- student_id (FK)
- (Unique constraint: event_id + student_id)
- Attendance
- att_id (PK)
- reg_id (FK)
- status (Present, Absent, Late)
- Feedback
- fb_id (PK)
- reg_id (FK)
- rating (1–5)
- comment

ER Diagram

Campus Event Reporting System ER Diagram



API Design

- MethodEndpointDescription
- POST
- /create_event
- Create new event
- POST
- /register_student
- Register student for event
- POST
- /mark_attendance
- Mark student attendance
- POST
- /give_feedback ⁰¹
- Submit feedback
- GET
- /report/event_popularity
- Registrations per event
- GET ⁰²
- /report/student_participation
- Events attended by each student
- GET
- /report/top_students
- Top 3 active students ⁰³
- GET
- /report/filter?type=workshop
- Events filtered by type

Workflows

Registration Workflow

1. Student selects event → system checks for duplicates.
2. If valid → Registrations entry created.
3. Confirmation sent to student.

Attendance Workflow

1. On event day → Admin selects student.
2. Status = Present / Absent / Late → recorded in Attendance.

Reporting Workflow

1. System aggregates data from Registrations + Attendance + Feedback.
2. Generates reports (event popularity, participation, feedback average).

Student → Register API → DB → Confirmation

Admin → Attendance API → DB → Report Generated

Edge Cases

- Duplicate registration → Blocked by unique constraint.
- Cancelled event → Status “Inactive”, reports exclude it.
- Missing feedback → Reports handle null values.
- Attendance not marked → Default “Absent”.

Reports

- Event Popularity: COUNT(*) of registrations per event.
- Attendance %: $(\text{Present} \div \text{Total Registered}) \times 100$.
- Average Feedback: AVG(rating) per event.
- Student Participation: events attended per student.
- Bonus: Top 3 most active students.

Future Enhancements

- QR code scanning for attendance.
- Student dashboard to view registered events.
- Automated event reminders via email/SMS.
- Analytics dashboard for college admins.