**Campus Event Management Platform - Design Document**

**1. Data to Track**

* **Events**: ID, name, type, date, time, location, created\_by.
* **Students**: ID, name, email, college, department.
* **Registrations**: student\_id, event\_id, registration\_date.
* **Attendance**: student\_id, event\_id, attendance\_status.
* **Feedback**: student\_id, event\_id, rating (1-5), comments.

**2. Database Schema**

**Tables**:

* **events** (event\_id, name, type, date, time, location, created\_by)
* **students** (student\_id, name, email, college, department)
* **registrations** (id, student\_id, event\_id, registration\_date)
* **attendance** (id, student\_id, event\_id, status)
* **feedback** (id, student\_id, event\_id, rating, comments)

**Relationships**:

* One event → many registrations, attendance, feedback.
* One student → many registrations, attendance, feedback.

**3. API Design**

**Events Router**

* POST /events/ → Create event
* GET /events/ → List events

**Registrations Router**

* POST /registrations/ → Register student to event
* GET /registrations/{event\_id} → View event registrations

**Attendance Router**

* POST /attendance/ → Mark attendance
* GET /attendance/{event\_id} → Get attendance list

**Feedback Router**

* POST /feedback/ → Submit feedback
* GET /feedback/{event\_id} → Get feedback for event

**Reports Router**

* GET /reports/event-popularity → Registrations per event
* GET /reports/student-participation → Events attended by student
* GET /reports/attendance-percentage/{event\_id} → Attendance percentage
* GET /reports/average-feedback/{event\_id} → Average feedback score
* **Bonus**: GET /reports/top-students → Top 3 most active students

**4. Workflows**

**Registration → Attendance → Reporting**

1. **Registration**
   * Student selects event
   * API call to /registrations/
   * Entry created in registrations
2. **Attendance**
   * On event day, student checks in
   * API call to /attendance/
   * Status updated in attendance
3. **Feedback**
   * Student submits rating/comment
   * API call to /feedback/
4. **Reporting**
   * Admin queries reports via endpoints
   * Event popularity, student participation, attendance %, feedback

**5. Assumptions & Edge Cases**

* Each student can register for an event only once (unique student\_id + event\_id in registrations).
* Duplicate feedback not allowed; only one feedback per student per event.
* Cancelled events should not appear in reports.
* Attendance must be marked only for registered students.
* If feedback is missing, exclude from average score calculations.