**Campus Event Management Platform**

**1. Introduction**

This document explains the design and prototype of a **Campus Event Management Platform** that I built.  
The platform has two main parts:

* **Admin Portal (Web)** – For college staff to create, manage, and monitor events.
* **Student App (Mobile)** – For students to browse events, register, mark attendance, and give feedback.

The goal is to make event management smooth for staff while keeping it easy and engaging for students.

**2. Assumptions & Decisions**

While designing the system, I made some assumptions and choices to keep things simple yet functional:

* Every student has a unique **Student\_ID** within their college.
* Events have unique IDs per college, combined as CollegeID\_EventID to avoid clashes.
* A student can register for an event only once.
* Attendance can only be marked for registered students and only **on the day of the event**.
* Feedback is optional but tied to attendance.
* Cancelled events are ignored in reports.
* Database: I chose **SQLite** because it’s easy to set up and perfect for a small prototype.

**Deviations from AI suggestions:**

1. I included a College\_ID column in all tables so each college’s data is clearly separated.
2. Attendance is restricted to the actual event day to prevent any fake check-ins.

**3. Data to Track**

To make reporting and management possible, the platform tracks:

* **Events:** Event\_ID, Name, Type (Hackathon, Workshop, Fest), Date, College\_ID, Capacity, Category.
* **Students:** Student\_ID, Name, Email, College\_ID.
* **Registrations:** Student\_ID, Event\_ID.
* **Attendance:** Student\_ID, Event\_ID, Status (Present/Absent).
* **Feedback:** Student\_ID, Event\_ID, Rating (1–5), Comment.

**Deviations from AI suggestions:**

* Added Event Capacity so the system can handle maximum registrations.
* Added Event Category to allow filtering and better reporting.

**4. Database Schema**

Here’s the table structure I used:

* **Colleges**(College\_ID, Name)
* **Students**(Student\_ID, Name, Email, College\_ID)
* **Events**(Event\_ID, Name, Type, Date, College\_ID, Capacity, Category)
* **Registrations**(Reg\_ID, Student\_ID, Event\_ID)
* **Attendance**(Attend\_ID, Student\_ID, Event\_ID, Status)
* **Feedback**(Feedback\_ID, Student\_ID, Event\_ID, Rating, Comment)

**Deviations from AI suggestions:**

* Added Capacity and Category fields in Events to make the platform more practical.

**Action:** Include a simple ER diagram (hand-drawn or using draw.io) here.

**5. API Design**

I built REST APIs to handle all operations:

* **POST /events** → Create a new event.
* **POST /students** → Add a new student.
* **POST /register** → Register a student for an event.
* **POST /attendance** → Mark attendance for a student.
* **POST /feedback** → Submit feedback for an event.
* **GET /reports/popularity** → Get events sorted by registrations.
* **GET /reports/attendance/:event\_id** → Get attendance percentage for an event.
* **GET /reports/feedback/:event\_id** → Get average feedback score.
* **GET /reports/student/:student\_id** → Get all events attended by a student.
* **GET /reports/top-students** → Show top 3 most active students.
* **GET /reports/event-type/:type** → Filter reports by event type.

**Deviation from AI suggestions:**

* I added /reports/top-students and /reports/event-type/:type because these would make reports more meaningful for admins.

**6. Workflows**

Here’s how the main workflow works:

1. Student selects an event → API /register.
2. For events with limited capacity → registration requires **manual approval**.
3. On the event day → API /attendance. Only registered students can mark attendance.
4. After the event → API /feedback.
5. Reports are generated from the database using queries.

**Deviation from AI suggestions:**

* Added the manual approval step for limited-capacity events to make the workflow more realistic.

**Action:** Include a sequence diagram image for one workflow (e.g., registration → attendance → feedback).

**7. Reports**

The platform can generate these reports:

* **Event Popularity Report** → Sorted by number of registrations.
* **Attendance Percentage Report** → Shows % of attendees for each event.
* **Feedback Report** → Average feedback score per event.
* **Student Participation Report** → Shows how many events each student attended.
* **Top 3 Active Students Report** → Lists the most active students.
* **Filter by Event Type Report** → Filter reports by event category.

**Deviations from AI suggestions:**

* Feedback is displayed anonymously in reports to encourage honest responses.
* Added Top 3 Active Students report and the ability to filter by event type.

**8. Edge Cases**

* Duplicate registrations are prevented by a unique (Student\_ID, Event\_ID) constraint.
* Feedback is optional.
* Cancelled events are excluded from reports.
* Students who aren’t registered cannot mark attendance.

**9. Scale Considerations**

* Designed for ~50 colleges × 500 students × ~20 events per semester.
* Event IDs are unique per college to avoid conflicts.
* Data is stored in a single database but filtered by College\_ID in all queries to keep reports accurate.

**Bonus Features**

1. **Top 3 Most Active Students**

SELECT student\_id, COUNT(\*) AS events\_attended

FROM Attendance

WHERE status='Present'

GROUP BY student\_id

ORDER BY events\_attended DESC

LIMIT 3;

1. **Flexible Reports (Filter by Event Type)**

SELECT e.name, COUNT(r.id) AS total\_registrations

FROM Event e

JOIN Registration r ON e.id = r.event\_id

WHERE e.type='Workshop'

GROUP BY e.id;

1. **Simple UI Mockups / Wireframes**

* Event browsing list
* Register button
* Attendance & feedback submission  
  *(Can be drawn in draw.io, Figma, or paper for submission)*

**4. Scale Assumptions**

* ~50 colleges × 500 students × 20 events/semester.
* Event IDs: global auto-increment in prototype.
* Data storage: single dataset; college\_id can be added for production.

Student

+----+-------+-------+

| id | name | email |

+----+-------+-------+

Event

+----+------+-------+------+

| id | name | type | date |

+----+------+-------+------+

Registration

+----+------------+---------+

| id | student\_id | event\_id|

+----+------------+---------+

Attendance

+----+------------+---------+--------+

| id | student\_id | event\_id| status |

+----+------------+---------+--------+

Feedback

+----+------------+---------+--------+---------+

| id | student\_id | event\_id| rating | comment |

+----+------------+---------+--------+---------+