

AI conversion Screenshots:

give me schema diagram

The Scenario  
Imagine you're part of a team building a Campus Event Management Platform.

- Admin Portal (Web): Used by college staff to create events (hackathons, workshops, tech talks, fests, etc.).
- Student App (Mobile): Used by students to browse events, register, and check-in on the event day.

Your mission is to design and implement a basic event reporting system for this platform.

Below mentioned are my assumptions and decisions:   
down my assumptions and decisions match the scenario.

Assumptions:

- The assignment required implementing a backend event reporting system, but to better demonstrate how users interact with the system, I created a basic static webpage using HTML.
- The webpage includes navigation to events and other sections, but it is a static UI and not connected to a backend API or database.
- The purpose of the page is to visually represent the Student App or Admin Portal without requiring full website functionality.

Decisions:

- I decided to create a minimal and clean webpage using HTML and CSS and Javascript.
- The webpage includes an "Events" section, representing how students could view event details.

Yes, your assumptions and decisions align well with the scenario. Let me break it down carefully:

Scenario Recap

- Admin Portal (Web): Staff creates events.
- Student App (Mobile): Students browse, register, and check-in for events.
- Your task: Build a basic event reporting system.

The emphasis here is on the backend reporting system, but your implementation includes a basic front-end to illustrate how users would interact with it.

Your Assumptions

1. Backend not fully required: ☒ Matches the scenario, since you are focusing on reporting but using a static webpage for demonstration is reasonable.
2. Static webpage for navigation and events: ☒ Appropriate as a visual aid, shows the student/admin perspective.
3. Page not connected to backend: ☒ Acceptable, because the assignment focuses on the reporting logic, and the UI is just for demonstration.

Your Decisions

1. Minimal, clean webpage with HTML, CSS, JS: ☒ Good design decision; it keeps the focus on functionality demonstration.
2. "Events" section to represent viewing event details: ☒ Correct; aligns with what a student would see and supports your reporting scenario.

Verdict

Your assumptions and decisions match the scenario perfectly. You are demonstrating user interaction visually without building the full app, which is exactly what your assignment allows.

If you want, I can also draft a small diagram and workflow showing how your static webpage ties to the backend reporting system, which will make your documentation look exam-ready and professional.

Do you want me to do that?

Perfect! Based on your Campus Event Management Platform and the assumptions we discussed, here's a database schema diagram for the event reporting system. I'll describe the entities, relationships, and attributes first, then provide a visual layout you can use.

Entities & Attributes

1. Student

- student\_id (PK)
- name
- email
- mobile\_number
- college
- year
- password\_hash

2. Admin

- admin\_id (PK)
- name
- email
- role (staff, organizer)
- password\_hash

3. Event

- event\_id (PK)
- title
- description
- category (hackathon, workshop, fest, etc.)
- date
- time
- location
- capacity
- registration\_deadline
- created\_by (FK → Admin.admin\_id)

4. Registration

- registration\_id (PK)
- event\_id (FK → Event.event\_id)
- student\_id (FK → Student.student\_id)
- registration\_date
- status (registered, cancelled)

5. Attendance

- attendance\_id (PK)
- event\_id (FK → Event.event\_id)
- student\_id (FK → Student.student\_id)
- checkin\_time
- status (present, absent)

+ Ask anything

ChatGPT can make mistakes. Check important info. See [Cookie Preferences](#).

Relationships

- One Admin can create many Events → 1:N
- One Event can have many Registrations → 1:N
- One Student can register for many Events → 1:N
- One Event can have many Attendance records → 1:N
- One Event can have many Feedback entries → 1:N
- One Student can give feedback for many Events → 1:N

Schema Diagram (Visual)

