# Campus Drive Assignment – Webknot Technologies

Name: Mohmmed Sabeel   USN: 1AY22CD029   Date: 07-09-2025

## Assumptions and Key Decisions

1. The system will be used by multiple colleges, all data will be stored in a single database and separated by college_id.

2. Event identifiers will be unique within each college, not across all colleges.

3. A student can register only once for the same event; duplicate registrations will be blocked at the database level.

4. Attendance can only be marked on the event day, either by QR scan or manual update by admin.

5. Feedback will consist of a numeric rating between 1–5 and one comment per student per event.

6. Cancelled events will not allow further registrations, attendance marking, or feedback submission.

7. The prototype will prioritize simplicity: minimal APIs, SQLite database, and basic validations.

## AI Brainstorming Log

I used an AI assistant to brainstorm ideas for database schema, API design, workflows, and possible edge cases.

The conversation helped me refine:

➢ How to structure the entities and relationships between students, events, registrations, attendance, and feedback.

➢ Which APIs are essential for reporting.

➢ Which SQL queries can generate event popularity, student participation, and feedback averages.

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

Share

New chat
Search chats
Library

Sora
GPTs

New project

Chats

Placement assignment steps
AI-driven chatbots overview
AI in business case study
Python parameter check
AI in Business Case Study
AI case study activity
Push-up tracking code
Sofa movement problem

sabeel
Free

Campus Event Reporting – Design Doc (webknot Assignment)    Copy  Edit  Download

## 6) Workflows (Sequence)

### 6.1 Registration → Attendance → Feedback → Reporting

```
sequenceDiagram
    participant S as Student App
    participant A as Admin API
    participant DB as Database

    S->>A: GET /events (browse)
    A->>DB: Query events
    A-->>S: Event list

    S->>A: POST /registrations (event_id, student_id)
    A->>DB: Insert registration (unique)
    A-->>S: 201 reg_id

    S->>A: POST /attendance (present=true)
    A->>DB: Insert attendance
    A-->>S: 201 att_id

    S->>A: POST /feedback (rating, comment)
```

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

23°C
Mostly cloudy

ENG
IN

02:07
07-09-2025

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

Share

Campus Event Reporting – Design Doc (webknot Assignment)  Copy  Edit  Download

## 5.6 Reports

- **Event Popularity** (registrations per event)
  - `GET /reports/events/popularity?type=Workshop`
  - `200` `[ { "event_id":"...", "title":"...", "registrations":120 } ]`
- **Attendance Summary (per event)**
  - `GET /reports/events/attendance?event_id=...`
  - `200` `{ "event_id":"...", "registered":120, "present":100, "attendance_pct":83.33 }`
- **Average Feedback (per event)**
  - `GET /reports/events/feedback?event_id=...`
  - `200` `{ "event_id":"...", "avg_rating":4.3, "responses":88 }`
- **Student Participation**
  - `GET /reports/students/participation?student_id=...`
  - `200` `{ "student_id":"...", "attended":7 }`
- **Top 3 Most Active Students (Bonus)**
  - `GET /reports/students/top?limit=3`

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

23°C
Mostly cloudy

ENG
IN
02:07
07-09-2025

---

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

Share

**Campus Event Reporting – Design Doc (webknot Assignment)**    Copy   Edit   Download

**Style:** REST, JSON. Base path `/api/v1`. All requests must include `X-College-ID` header (or it can be encoded in auth in real systems).

### 5.1 Events

- **Create Event**
  - `POST /events`
  - Body `{ "title":"Intro to AI", "type":"Workshop", "start_time":"2025-09-15T10:00:00Z", "end_time":"2025-09-15T12:00:00Z", "venue":"Auditorium", "capacity":100 }`
  - `201 { "event_id":"..." }`
- **List Events** (filters)
  - `GET /events?type=Workshop&from=2025-09-01&to=2025-09-30`
  - `200 [ { "event_id":"...", "title":"...", "type":"Workshop", ... } ]`

### 5.2 Students

- **Create/Upsert Student**
  - `POST /students`

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

---

ChatGPT ⌄

Share

**Campus Event Reporting – Design Doc (webknot Assignment)**    Copy   Edit   Download

```
  }
```

### 4.2 Table Sketch (DDL-like)

Use **SQLite** for the prototype. Replace `uuid` with `TEXT` or `INTEGER PRIMARY KEY` as needed.

```sql
-- COLLEGE
CREATE TABLE colleges (
  college_id TEXT PRIMARY KEY,
  name TEXT NOT NULL,
  domain TEXT
);

-- STUDENT
CREATE TABLE students (
  student_id TEXT PRIMARY KEY,
  college_id TEXT NOT NULL REFERENCES colleges(college_id),
  name TEXT NOT NULL,
  email TEXT NOT NULL,
  roll_no TEXT,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  UNIQUE(college_id, email)
);
CREATE INDEX idx_students_college ON students(college_id);
```

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

**OneDrive**
Screenshot saved
The screenshot was added to your OneDrive.

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

**Campus Event Reporting – Design Doc (webknot Assignment)**   Copy  Edit  Download

```
    }
```

## 4.2 Table Sketch (DDL-like)

Use **SQLite** for the prototype. Replace `uuid` with `TEXT` or `INTEGER PRIMARY KEY` as needed.

```sql
-- COLLEGE
CREATE TABLE colleges (
  college_id TEXT PRIMARY KEY,
  name TEXT NOT NULL,
  domain TEXT
);

-- STUDENT
CREATE TABLE students (
  student_id TEXT PRIMARY KEY,
  college_id TEXT NOT NULL REFERENCES colleges(college_id),
  name TEXT NOT NULL,
  email TEXT NOT NULL,
  roll_no TEXT,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  UNIQUE(college_id, email)
);
CREATE INDEX idx_students_college ON students(college_id);
```
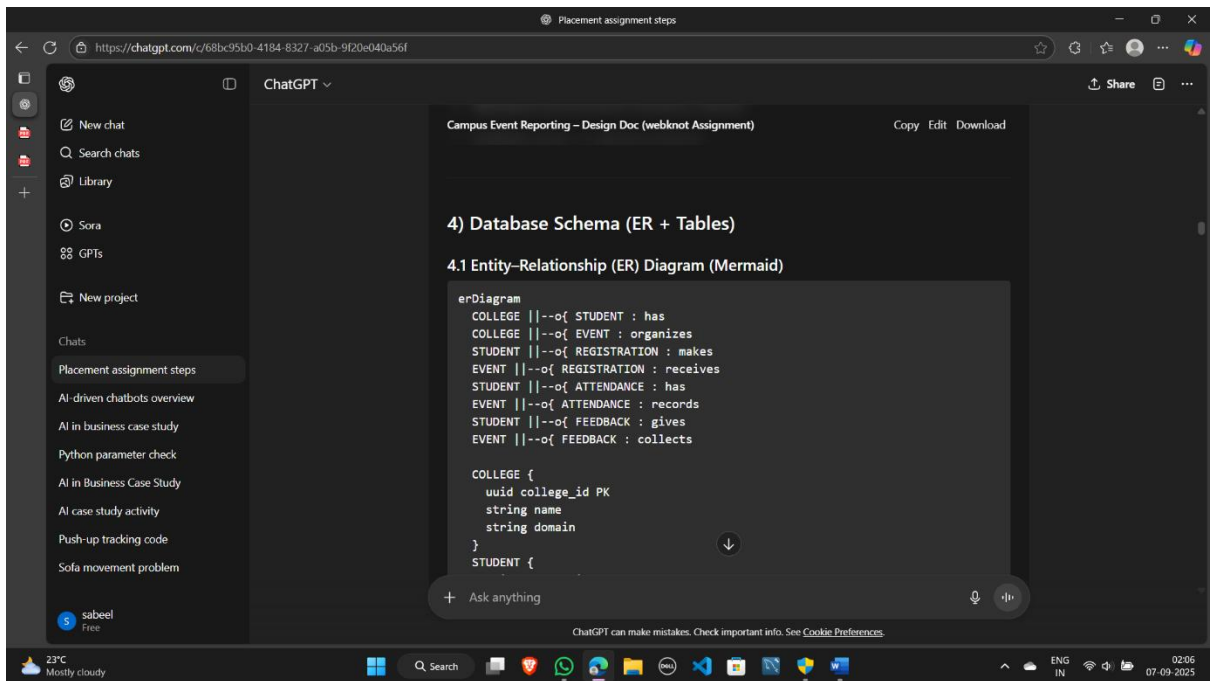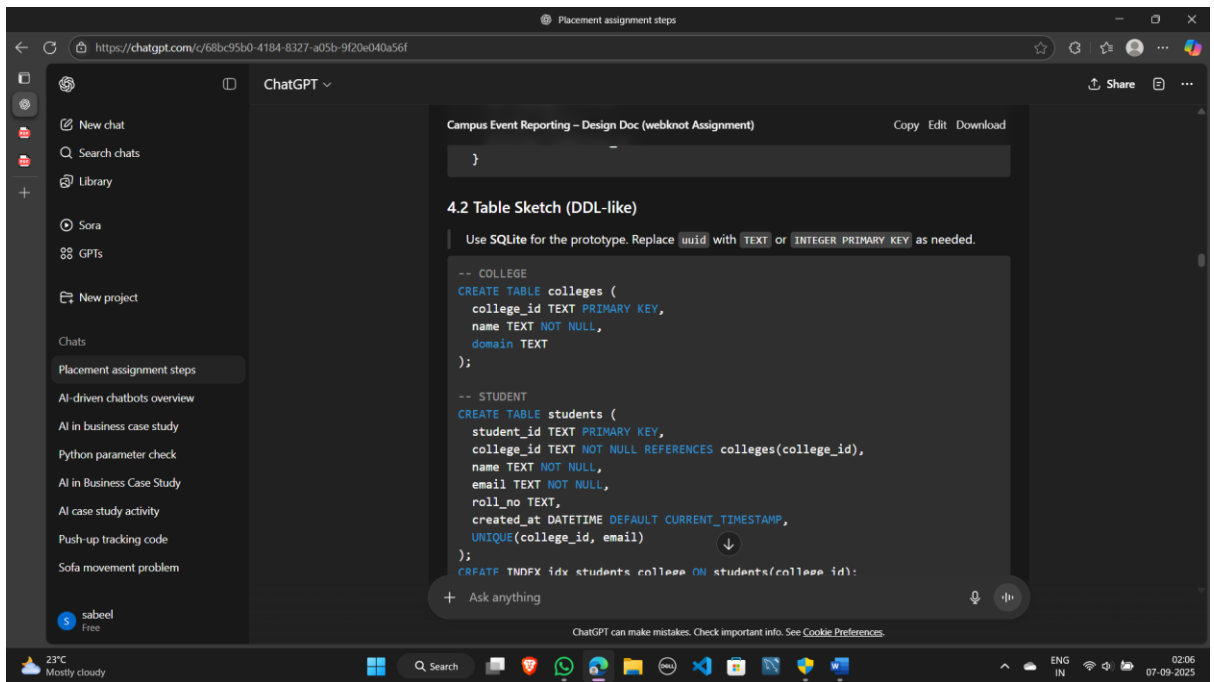
Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

---

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

**Campus Event Reporting – Design Doc (webknot Assignment)**   Copy  Edit  Download

# 4) Database Schema (ER + Tables)

## 4.1 Entity–Relationship (ER) Diagram (Mermaid)

```
erDiagram
    COLLEGE ||--o{ STUDENT : has
    COLLEGE ||--o{ EVENT : organizes
    STUDENT ||--o{ REGISTRATION : makes
    EVENT ||--o{ REGISTRATION : receives
    STUDENT ||--o{ ATTENDANCE : has
    EVENT ||--o{ ATTENDANCE : records
    STUDENT ||--o{ FEEDBACK : gives
    EVENT ||--o{ FEEDBACK : collects

    COLLEGE {
      uuid college_id PK
      string name
      string domain
    }
    STUDENT {
```

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

- What if event is canceled?
- What if feedback not submitted?

📌 **Deliverable:** PDF/Doc with schema diagram, sequence flow, and explanations.

---

## Step 3: Build a Prototype (Code)

Keep it simple but working.

- **Tech Stack (Choose one):**
    - Backend: Python (Flask/FastAPI) OR Node.js (Express)
    - Database: SQLite (easiest to set up)
- **Basic Features:**
    - Register student to event
    - Mark attendance
    - Collect feedback (1–5 rating)
- **Reports:**
    - Registrations per event → SQL `COUNT(*)`
    - Attendance percentage → `attended/registered * 100`
    - Average feedback → `AVG(rating)`

📌 **Deliverable:** Code + README (write README in your own words, no AI).

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

---

https://chatgpt.com/c/68bc95b0-4184-8327-a05b-9f20e040a56f

ChatGPT ⌄

## Step 2: Create a Design Document

This is the most important **non-code part**.

- **Data to Track**
    - Event details, student details, registrations, attendance, feedback.
- **Database Schema (ER Diagram)**
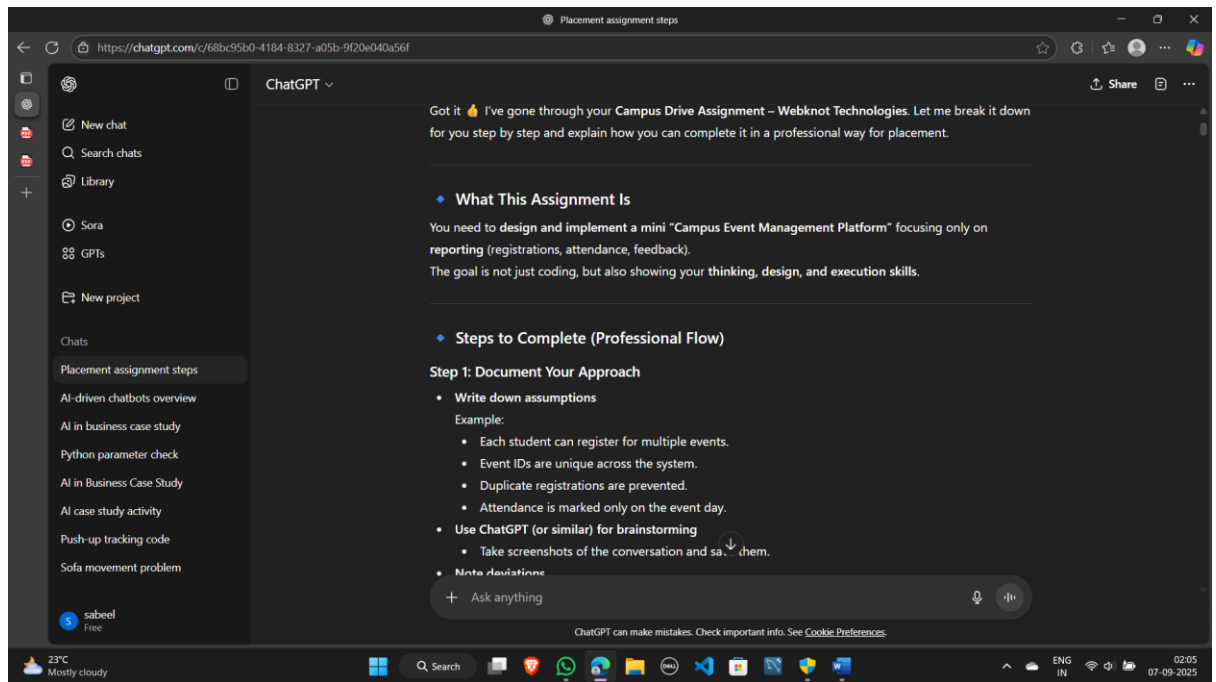    Example tables:
    - `students(student_id, name, email, college_id)`
    - `events(event_id, title, type, date, college_id)`
    - `registrations(reg_id, student_id, event_id)`
    - `attendance(att_id, student_id, event_id, status)`
    - `feedback(fb_id, student_id, event_id, rating)`
- **API Design**
    - `POST /events` → Create event
    - `POST /register` → Register student
    - `POST /attendance` → Mark attendance
    - `POST /feedback` → Submit feedback
    - `GET /reports/events` → Event popularity
    - `GET /reports/students` → Student participation
- **Workflows (Sequence Diagram)**

Ask anything

ChatGPT can make mistakes. Check important info. See Cookie Preferences.

# Following vs. Deviating from AI Suggestions

| AI Suggested Approach | Final Decision | Reason |
|---|---|---|
| • Separate database for each college | • Single database with college_id field | • Easier management and reporting across colleges |
| • Feedback stored only as free text | • Feedback includes numeric rating 1–5 with optional comment | • Supports report generation and analytics |
| • Implement full authentication (JWT/OAuth) | • Simple API key or header check for prototype | • Keeps prototype lightweight and focused on reporting |
| • Strict enforcement of capacity limit at registration | • Capacity check optional in prototype | • Focus on reporting, not full event lifecycle |