Design Document - Campus Event Management & Reporting System

1. Introduction

This project is focused on building the Event Reporting System for a Campus Event Management platform. The core platform for event creation and publishing is assumed to exist. Our task is only the reporting part: collecting participation data, analyzing it, and presenting it to admins in a clean and simple way.

2. Goals

- Provide admins with reports: event popularity, student participation, top students, event metrics.
- Provide students with minimal flows: browse, register, attend, feedback.
- Keep it simple, easy to run locally, and extensible.

3. Assumptions

- MySQL for database.
- Node.js + Express for backend.
- React + Vite + Tailwind for frontend.
- Simplified authentication (dummy admin login, student = email + ID).
- Event types limited to Workshop, Fest, Talk.

4. System Architecture

Three main components:

1. Frontend (React + Vite + Tailwind)
2. Backend (Express + MySQL)
3. Database (MySQL)

Data flows from students -> backend -> database -> reports for admins.

5. Database Design

Tables:

- colleges(id, name)
- students(id, student_uid, name, email, college_id)
- events(id, title, description, type, date, college_id, is_cancelled)
- registrations(event_id, student_id)

- attendance(event_id, student_id)
- feedback(event_id, student_id, rating, comment)

Relationships: One college -> many students/events, Many-to-many between students and events via registrations, attendance, feedback.

6. API Design

Student APIs: register, login, register for event, attendance, feedback.

Admin APIs: create event, list events.

Reports: event popularity, student participation, top students, event metrics.

7. Workflows

Student:

1. Login/Register
2. Browse events
3. Register
4. Check-in
5. Feedback

Admin:

1. Login
2. Create events
3. View dashboard reports

8. Reporting Logic

- Event popularity = count(registrations).
- Attendance % = attended/registered*100.
- Student participation = count(attendance).
- Top students = order by attendance.
- Feedback = avg(rating).

9. Frontend Design

Admin: dashboard with charts, event management.

Students: event list, register, check-in, feedback.

UI is minimal, clean, responsive with Tailwind.

10. Deployment

Backend: Node.js server.

Frontend: React build (Netlify/Vercel).

Database: MySQL.

Config via .env files.

11. Limitations

- Dummy authentication.
- No CSV export.
- Reports are basic.
- No real-time updates.

12. Future Enhancements

- JWT authentication.
- Role-based access.
- CSV/PDF export.
- QR-based attendance.
- Notifications.
- Analytics dashboard.

Diagrams

[ER Diagram - Simplified]

Colleges --< Students

Colleges --< Events

Students --< Registrations >-- Events

Students --< Attendance >-- Events

Students --< Feedback >-- Events

[Workflow Diagram]

Students -> Register/Login -> Browse Events -> Register -> Attend -> Feedback

Admins -> Login -> Create Events -> View Dashboard (Reports)