Intern Software Engineer Assignment

Task:

Build a robust Event Management System using React (Frontend) and Spring Boot, .NET, or Node.js (Backend), with PostgreSQL or MySQL as the database. The system should showcase good coding skills and maintainability.

Use of templates or AI-generated code is prohibited and will result in rejection.

Backend

1. Database Schema:

- Create tables for:
  - Events (name, description, date, location, created_by, capacity, remaining_capacity, tags).
  - Attendees (name, email, registered_events).
- Ensure referential integrity and data constraints (e.g., capacity limits).

2. API Endpoints (All endpoints must validate input and handle errors):

- Fetch all events (support pagination, filtering by date, location, and tags).
- Add a new event (validate for capacity limits and input length).
- Update event details (disallow past events from being updated).
- Delete an event (cascade delete attendees registered to the event).
- Register an attendee to an event (validate against event capacity).
- Fetch attendees for a specific event.
- Fetch event analytics (total attendees, capacity utilization).

3. Backend Requirements:

- Use Spring Boot, .NET Core, or Node.js for the backend.
- Implement DTOs (Data Transfer Objects) and separate models for database entities.
- Apply Service Layer Design Pattern for business logic.

Frontend

1. Key Components:

- Reusable Components: Create common reusable components such as:
  - Button, Input, Modal, and Table.
- Pages:
  - Event List Page: Displays all events in a paginated table with filters and search.
  - Event Detail Page: Shows detailed event information and a list of registered attendees.

- Event Creation Page: Form for adding new events.
- Event Update Page: Editable form prefilled with existing event data.

- Include modals for confirmation (e.g., before deletion).

2. Frontend Features:

- Fetch and display event data using API endpoints.
- Use React Context API or Redux Toolkit for state management.
- Implement form validation for event creation and attendee registration.

UI/UX and Styling

1. Design:

- Use CSS Modules or SCSS for styling.
- Create a responsive layout supporting desktop, tablet, and mobile views.
- Design an intuitive and professional UI.

General Guidelines

1. Code Quality:

- Follow industry-standard best practices (e.g., SOLID principles, DRY).
- Include meaningful comments and use a consistent code style.
- Avoid hardcoding values; use environment variables where needed.
- Structure the project into modular folders (e.g., services, components, utils).

2. README.md:

- Include detailed instructions to set up and run the application locally.
- Add API documentation links, project structure explanation, and design decisions.

3. Submission:

Push your code to a public GitLab or GitHub repository.