**Project Brief: *Mini Event Manager***

**Overview**

**Project:** Build a simple “Event Manager” application. The application should allow users to:

1. **Register** as new users (sign-up).
2. **Log in** with existing credentials.
3. **Create, view, update, and delete** events.
4. **View** all events or filter events by date or organizer.
5. **Perform basic error handling** and validations.

**Technology Stack (Suggested):**

* **Frontend:** Angular
* **Backend:** Java Spring Boot
* **Database:** PostgreSQL
* **Authentication:** JSON Web Tokens (JWT) or session-based authentication

**1. Core Functionality**

1. **User Registration & Login**
   * New users can register with **name**, **email**, and **password**.
   * Existing users can log in with **email** and **password**.
   * Maintain sessions or use JWT for authentication.
2. **Create & Manage Events**
   * **Create Event:** Authenticated users can create a new event with:
     + **Title** (string)
     + **Description** (text)
     + **Date** (e.g., “2025-02-10”)
     + **Location** (e.g., “Online” or a physical address)
   * **Edit/Delete Event:** A user can edit/delete **only the events they created**.
   * **List Events:** Show all created events in a simple list or grid.
3. **Register for Events**
   * Any logged-in user can **register** for an event (not necessarily only the creator).
   * Show a list of **upcoming events** with a button to “Register.”
   * The user should also be able to **unregister** (cancel) if they change their mind.
4. **View Registered Events**
   * Each logged-in user can view:
     + **All events** (publicly listed).
     + **My Registered Events** (a simple list of events they’ve signed up for).
     + **Events I Created** (for editing or deleting).

**2. Technical Requirements**

1. **Frontend**
   * A simple single-page application using Angular
   * Pages or sections needed:
     + **Sign Up / Login** forms.
     + **Event List**: Displays all events, with a button to register (or unregister).
     + **Create/Edit Event** form (only visible to the event creator for edit).
     + **My Registrations**: Lists all events a user is registered for.
2. **Backend**
   * + Using Java/ Spring boot, create as many endpoints as necessary
3. **Database**
   * Use the sepcified database
4. **Authentication & Authorization**
   * Protect endpoints so that only **logged-in users** can:
     + Create, edit, delete events.
     + Register/unregister for events.
   * Ensure **only the creator** of an event can edit or delete it.
   * Use JWT (token-based) or express-session (server-side sessions).

**3. Deliverables**

1. **Source Code**
   * A Git repository containing both the frontend and backend code.
   * If using environment variables (.env), provide a sample file.
2. **Demo / Walkthrough**
   * In the next interview show the following:
     + Signing up a new user and logging in.
     + Creating, editing, and deleting an event.
     + Registering for another user’s event.
     + Viewing your registrations in a simple list.

Optional small tests: If time allows, add a **few unit tests** or integration tests for critical routes (e.g., authentication and event creation).

**Evaluation Criteria**

1. **Correctness:**
   * Does the app support user authentication, event creation, registration, and the necessary read/update/delete operations?
2. **Code Quality & Organization:**
   * Is the codebase relatively clean and well-structured (frontend + backend)?
3. **UI/UX:**
   * Minimal but functional UI is fine. Are basic edge cases handled?
4. **Documentation & Setup:**
   * Is it easy to clone the repo, install dependencies, and get the app running?
5. **Communication / Demo:**
   * Clarity in explaining how the solution works and why certain choices were made.

**Final Notes**

* This **Mini Event Manager** is challenging for 1 day dev on purpose (it is Ok not to finalize all the functionalties), so focus on essential functions, avoid loosing time on details: