

Hackathon Assignment: Event Management Dashboard

Problem Statement

Organizations often struggle to organize and manage events efficiently. As part of this hackathon, you are required to develop a web-based Event Management Dashboard to streamline the process of creating events, assigning tasks, and tracking progress.

The application should allow users to:

1. Manage events (CRUD operations).
2. Manage attendees and assign them tasks.
3. Track tasks related to events with progress visualization.

Requirements

1. Frontend Development

You need to create a user-friendly and responsive dashboard with the following pages/features:

- Event Management Page:
 - Display a list of all events with options to add, edit, and delete events.
 - Each event should display details like name, description, location, and date.
- Attendee Management Page:
 - View the list of attendees.
 - Add or remove attendees.
 - Assign attendees to specific events or tasks.
- Task Tracker Page:
 - Display tasks associated with each event.
 - Allow users to update task status (Pending/Completed).

UI Guidelines

- The interface must be intuitive and user-friendly.
 - Ensure the design is responsive (works on both mobile and desktop devices).
 - Provide form validation (e.g., no empty fields when adding an event or attendee).
-

2. Backend Development

Develop RESTful APIs to support the frontend functionality. The APIs must include:

1. Event Management API:

- Create an Event
- Get all Events
- Update an Event
- Delete an Event

2. Attendee Management API:

- Add an Attendee
- Get all Attendees
- Delete an Attendee

3. Task Management API:

- Create a Task
 - Get Tasks for an Event
 - Update Task Status
-

3. Integration

1. Integrate the frontend with the backend APIs to fetch and display real-time data.
 2. Ensure smooth user interactions, such as:
 - Displaying success/error messages for API calls.
 - Loading indicators while waiting for API responses.
-

Constraints

1. Each event should have:
 - Name, description, location, date, and a list of attendees.
 2. Each task should have:
 - Name, deadline, status (Pending/Completed), and the assigned attendee.
 3. Form inputs must be validated (e.g., no empty fields, valid dates).
-

Bonus Points

1. Add authentication (login/logout) for accessing the dashboard.
2. Show progress using a visual indicator (e.g., progress bar) based on task completion (Task Tracker Page).
3. Implement a calendar view for displaying events visually.
4. Integrate real-time updates (e.g., using WebSockets) to show task progress dynamically.

Deliverables

1. Complete source code of the project (GitHub repository link).
2. A short presentation/demo to explain your application.
3. Documentation that includes:
 - Instructions to set up and run the project.
 - Details of the APIs developed.

Instructions : After completing the project, create a repository in your GitHub account and push your code to it. Add the provided email as a collaborator to review your code.

Email Id: **developer@webknot.in**