**Demo**

Welcome to the **Trainer Dashboard**! This web app is designed to offer an intuitive platform for managing trainers, tracking training opportunities, and keeping up with various assigned tasks.

**1. Functionalities**

**Trainer Management**

* **Adding and Editing Trainer Profiles:**
  + **Functionality**: Trainers can be added by clicking the "Add Trainer" button. When adding or editing trainers, we provide an interactive form where key information like **Name**, **Expertise**, **Availability**, and **Contact Information** are captured.
  + **Design Choice**: A modal dialog opens for adding or editing the trainer details. This design ensures the page doesn’t need to reload to add or edit trainers, providing a seamless user experience.
  + **Backend Integration**: When a trainer is added, edited, or removed, an API call (using Axios) is made to the backend to store or update the information in the database.
* **Trainer Listing**:
  + **Functionality**: Trainers are listed with key information such as their **name**, **expertise**, **availability**, and **contact info**.
  + **Design Choice**: Trainers are displayed in cards with buttons for editing or removing them. The design is straightforward with clear labels for readability.
  + **User Interaction**: Buttons allow for editing or deleting the trainers without needing to refresh the page, making it user-friendly.

**Opportunities Management**

* **Viewing and Filtering Opportunities:**
  + **Functionality**: Training opportunities are listed with details like **title**, **type**, **location**, and **status**. Filters for **Type** (Technical, Soft Skills, Leadership) and **Location** (Online, In-Person) can be used to narrow down results.
  + **Design Choice**: This section features drop-down filters at the top to refine the opportunity search. It encourages users to find the most relevant opportunities while minimizing clutter.
  + **Express Interest**: Trainers can express interest in an opportunity by clicking the “Express Interest” button. When clicked, the status of the opportunity changes (on the frontend), and the backend is updated via an API call. This functionality mimics the real-world interaction of trainers indicating their interest in a course.
* **Dynamic Card Rendering**:
  + Each opportunity is dynamically rendered as a card to make the interface visually appealing. By using cards instead of simple lists, users can easily scan the information, and opportunities feel more like a "collection" that users can pick through.

**2. Design Choices**

**User Interface (UI)**

* **Material-UI Components**:
  + The app is built using **Material-UI** for components like **Cards**, **Buttons**, **Select Dropdowns**, **Dialog Modals**, and more. Material-UI is a React component library that allows us to easily build functional, responsive, and attractive interfaces.
  + By utilizing Material-UI, we've ensured that all UI components are mobile-responsive and work on different screen sizes. The usage of Material-UI also helps in creating consistency across elements with minimal effort, so users experience a cohesive UI.

**Grid Layout and Spacing:**

* The layout uses a **Grid system** to ensure a well-structured, flexible interface that adapts to different screen sizes (especially for responsiveness). This approach allows:
  + Dividing content into clear sections.
  + Ensuring that the page doesn't look too cluttered.

All of this is achieved while keeping the design simple and visually appealing, with sections that are easy to understand.

**Color Scheme:**

* We’ve opted for a calm and professional color scheme:
  + **Primary Blue**: For action buttons and highlights (such as the “Add Trainer” button) to guide users' attention.
  + **Neutral Backgrounds**: Light shades to reduce visual clutter, helping users stay focused on core tasks.
  + **Contrast in Text**: Dark text on a light background provides good readability.

**Form UX/UI:**

* When adding/editing trainer details, we use **Text Fields** for input, each well-labeled. A dialog opens, providing a clean, isolated experience focused solely on input collection, allowing the user to quickly edit details without distractions.
* .

**4. Final Thoughts**

* **UI/UX**: The design prioritizes user simplicity and intuitiveness, following clear user flows with responsive components.
* **Maintainability**: The code structure and React component-based design allow easy maintenance and scalability, perfect for adding new features.