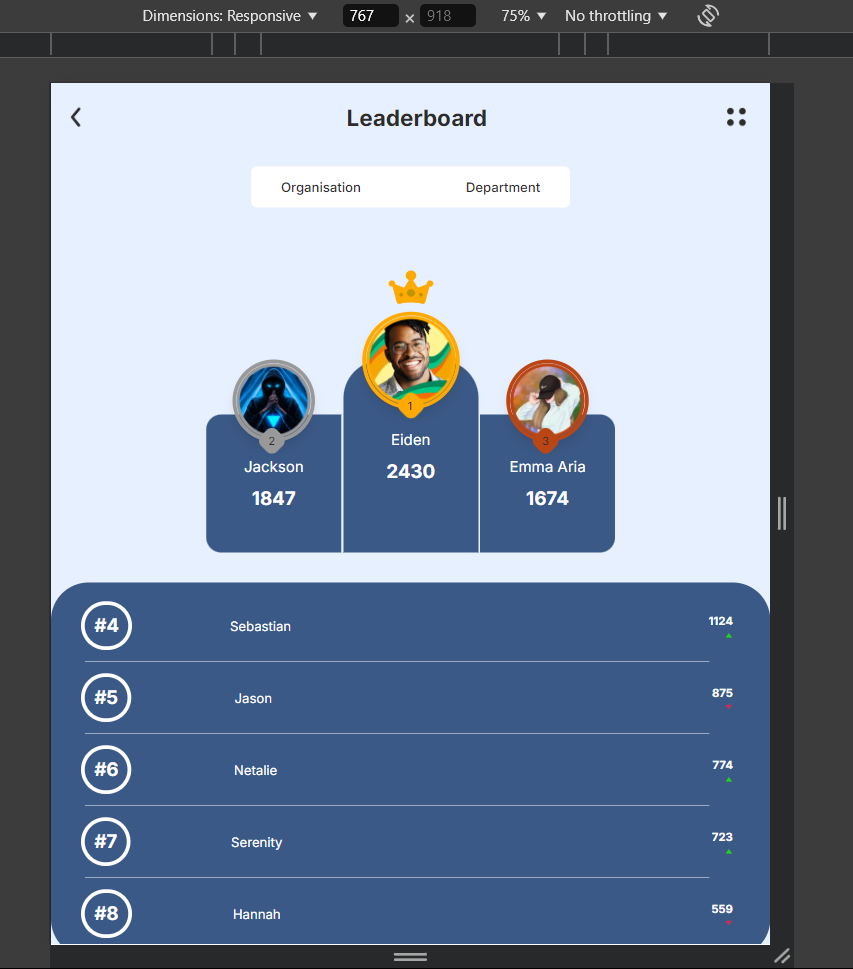
**Website on various resolutions:**



**Key Features and Functionalities**

1. **Responsive Design with Tailwind CSS**:
   * The entire project uses **Tailwind CSS** for styling, which allows for a responsive and utility-first design approach.
   * Components adapt to various screen sizes, ensuring a consistent user experience across all devices. For example, the Body component's layout changes between row and column flex directions based on screen size.
2. **Component-Based Architecture**:
   * The project follows a component-based architecture, which is a core concept of React. Each UI section, such as the header, navbar, stage, and table, is broken down into its own reusable component.
3. **Data Sorting and Display**:
   * The leaderboard data is dynamically sorted based on scores. This sorting logic is applied in the Body.js file using JavaScript's sort() method, which ensures that the leaderboard is always displayed in descending order of scores.
4. **Image and Media Management**:
   * Media assets (images) like icons and leaderboard images are imported and used directly within components. This improves maintainability and keeps the project organized.
5. **Navigation with Clickable Buttons**:
   * The Header component includes buttons with images (Back and Menu) that are clickable, allowing for potential navigation actions.
   * The Navbar component features clickable navigation items (Organisation and Department) with underline animations, which indicate the selected state.
6. **Conditional Rendering and State Management**:
   * The Navbar component uses React's useState hook to manage the selected state of navigation items. Conditional rendering is used to dynamically add underline effects when a nav item is selected.
7. **Scroll Handling and Overflow Management**:
   * The Body component handles overflow with the overflow-y-auto and overflow-x-hidden utilities from Tailwind, ensuring a clean UI without unnecessary scrollbars. This enhances the user experience, especially on smaller screens.
8. **Custom Styling without Global CSS Modifications**:
   * The project avoids modifying global styles such as index.css and instead relies on component-level styling using Tailwind utilities. This approach makes the codebase cleaner and more modular.

**Component Breakdown**

* **App.js**:
  + The main file that sets up the layout of the application. It includes the Header, Navbar, and Body components.
* **Header.js**:
  + Displays the app's header with a title ("Leaderboard") and clickable icons for navigation (Back and Menu).
* **Navbar.js**:
  + A horizontal navigation bar with selectable items. Clicking an item visually indicates the active section with an underline animation.
* **Body.js**:
  + The main content area that contains the Stage and Table components. It dynamically sorts and passes down leaderboard data to these components.
* **Stage.js**:
  + Displays the top three performers in a visually prominent section. It uses the sorted data passed from Body.js to highlight the top ranks.
* **Table.js**:
  + Renders a table with the remaining leaderboard data (ranks 4-8). It shows the name, score, and other relevant details for each participant.

Preview Link:   
Github Link: www.github.com/ak-1344/Leaderboard