Sportradar Coding Academy
Coding Exercise (FE)

Frontend Coding Exercise

Overview:

The goal of this exercise is to assess your understanding of basic frontend programming concepts, including building user interfaces, handling user interactions, and ensuring responsiveness across devices. You will create a sports event calendar that displays events, allows users to view event details, and lets users add new events during runtime.

Task 1 – Calendar View

- **Objective:** Implement a calendar view that displays sports events.
- **Instructions:**
    - Create a calendar interface for the current month.
    - Display the days of the month in a grid format.
    - Indicate days that have scheduled sports events. This can be as simple as a dot or marker on the day.
    - (Optional) Show brief information about the events on the calendar (e.g., event name on the day cell).
- **Sample Events to Include (you can leverage the json file shared to get events):**
    - **Sat., 18.07.2019, 18:30, Football, Salzburg vs. Sturm**
    - **Sun., 23.10.2019, 09:45, Ice Hockey, KAC vs. Capitals**

Task 2 – Event Detail Page

- **Objective:** Provide a way to view full details of a specific event.
- **Instructions:**
    - Implement functionality so that when a user clicks on an event marker on the calendar, they are taken to a detail page.
    - The detail page should display all relevant information about the event, such as date, time, sport, teams/participants, and any other details you consider important.

Task 3 – Add Event Functionality

- **Objective:** Allow users to add new events during runtime.
- **Instructions:**
    - Create a form where users can input details for a new event (e.g., date, time, sport, teams/participants).
    - Implement the functionality to add the new event to the calendar view.
    - Note: It is not required to save the data persistently (e.g., no need to use a database or local storage). The new event can exist only during the current session (runtime).

Task 4 – Responsiveness

- **Objective:** Ensure the site is fully responsive on mobile and tablet devices.
- **Instructions:**
    - Use responsive design techniques to make sure the calendar view and all pages adapt to different screen sizes.
    - Test your site on various screen widths to ensure usability on mobile phones, tablets, and desktops.
    - Pay attention to layout, readability, and touch interactions on smaller screens.

Task 5 – Navigation

- **Objective:** Create a simple navigation system within your application.
- **Instructions:**

- Implement a navigation bar or menu that allows users to switch between the calendar overview and the "Add Event" page.
- Ensure the navigation is accessible from all pages.
- The design can be minimal; focus on functionality.

Additional Features (Optional)

- **Filters:**
  - Implement filters to allow users to view events based on specific criteria (e.g., by sport, date range).
- **Styling and Enhancements:**
  - Add CSS styling to improve the visual appeal of your application.
  - Use animations or transitions for a better user experience.
- **Persistent Storage:**
  - Store events using local storage or any other method to retain data between sessions.
- **Testing:**
  - Write tests to verify that your code works as expected.
  - Use any testing framework or method you are comfortable with.

Submission Guidelines

- **Code Hosting:**
  - Upload your code to a GitHub repository.
  - Ensure the repository is public or that we have access to it.
- **Documentation:**
  - Include a README.md file with:
    - An overview of your project.
    - Instructions on how to run your application.
    - Any assumptions or decisions you made during development.
- **Version Control:**
  - Commit your code regularly with clear and descriptive commit messages.

- Your commit history should logically reflect the progression of your development.

Evaluation Criteria

- **Understanding of Frontend Concepts:**
  - Ability to create a dynamic and interactive user interface.
  - Proper handling of user interactions and events.
- **Code Quality (nice-to-have):**
  - Clean and readable code structure.
  - Appropriate use of comments and documentation within the code.
- **Functionality:**
  - Correct implementation of the required features.
  - The application runs without errors.
- **Responsiveness:**
  - The site adapts well to different screen sizes.
  - Usability is maintained across devices.
- **Presentation:**
  - User-friendly interface and navigation.
  - Logical organization of content and features.
- **Version Control Usage:**
  - Effective use of GitHub and adherence to version control best practices.
- **Optional Features:**
  - Implementation of additional features like filters or persistent storage.
  - Inclusion of tests to verify functionality.

Notes

- **Technology Choices:**
  - You may use plain HTML, CSS, and JavaScript, or any frontend framework/library you are comfortable with (e.g., React, Vue.js, Angular). You can also do the same for the styling.
  - Ensure that instructions for setup are included in your README.

- **Sports Data JSON (mock database):**
    - **Attached to the same email you should find a .json file that you can use to get the sports data for your calendar.**
- **Time Management:**
    - Focus on completing the core tasks first.
    - Optional features are a bonus but not required.
- **Assistance:**
    - If you have any questions or need clarification, feel free to reach out to us.

We look forward to seeing your solution. Good luck!