Sportradar Coding Academy
Coding Exercise (BE)

Overview:

This exercise aims to assess your understanding of basic programming concepts, including database modeling, data handling, backend functionality, and simple frontend development. You will create a sports event calendar that allows events to be added and categorized based on sports.

Task 1 – Database Modeling

- **Objective:** Design a database schema for a sports event calendar.
- **Instructions:**
    - Identify the necessary entities (tables) required to store sports events.
    - Create an Entity-Relationship Diagram (ERD) that includes all tables, their fields, and relationships.
        - (Optional) Follow the third normal form of database normalization.
    - Include additional relevant information that would enhance the sports calendar, such as venue details, team information, or event descriptions.

Examples of Events:

- **Sat., 18.07.2025, 18:30, Football, Salzburg vs. Sturm**
- **Sun., 23.10.2025, 09:45, Ice Hockey, KAC vs. Capitals**

Task 2 – Database Structure and Data

- **Objective:** Implement the database based on your ERD
- **Instructions:**
    - Choose a relational database system (e.g., MySQL, PostgreSQL, SQLite).
    - Create the database and all tables with their respective fields and data types.
    - Define primary keys and foreign keys. Name foreign keys with a prefix underscore (e.g., _foreignkey).

Task 3 – Implementation

- **Objective:** Develop a simple web application to display sports events and handle backend data manipulation.
- **Instructions:**
    - **Backend:**
        - Use a programming language you are comfortable with (e.g., Python, JavaScript, PHP).
        - Establish a connection to your database.
        - Implement backend functionality that allows adding new events to the database.
        - Implement functionality to get events. Write an efficient SQL query to retrieve event data. Avoid executing SQL queries inside loops.
        - Implement functionality to get one event.
    - **Frontend:**
        - Create an HTML page to display the events in a user-friendly format.
        - Display event details such as date, time, sport, teams/participants.
        - Include navigation elements (e.g., a navigation bar with placeholder links). Functionality for these links is not required.
        - (Optional) Add basic styling to enhance readability.
        - (Optional) Implement frontend functionality to add new events (e.g., a form).
    - **Additional Features (Optional):**
        - Implement filters to view events by sport or date.
        - Write tests to verify your code works as expected.
        - Include any other features you believe would add value to the calendar.

Submission Guidelines

- **Code Hosting:**
  - Upload your code to a GitHub repository.
  - Make sure the repository is public or accessible to us.
- **Documentation:**
  - Include a README.md file with:
    - An overview of your project.
    - Instructions on how to set up and run your application.
    - Any assumptions or decisions you made during development.
  - Include an AI_Reflection.txt file to explain how you have used AI (see Guidelines on AI Usage below)
- **Version Control:**
  - Commit your code regularly with meaningful commit messages.
  - The commit history should reflect the development progress logically.

Guidelines on AI usage

You are welcome to use AI tools (e.g., ChatGPT, GitHub Copilot, etc.) during the take-home assignment — but only under the following conditions:

- **You must understand your code**
- **You must be transparent about your use of AI**
  Along with your GitHub submission, you will provide a short reflection (1–2 pages) that explains:
  - Which parts of your solution were written by you
  - Which parts were generated or influenced by AI and how have you evaluated them
  - Why you made certain technical decisions
  - One improvement you would make if you had more time

  This is not about penalizing AI use — it helps us understand your thinking and learning process.
- **AI cannot replace your own contribution**
  Copy-pasting a full solution from AI without understanding it will work against you in later stages. The interview will make it clear whether you can explain and modify your own code.
- **We value how you learn — not perfection**
  We know AI can speed up learning when used thoughtfully. We are not testing whether you can work without AI — we are testing whether you can work with it responsibly.
  Use AI as a support tool, not as a substitute for understanding
- **Honor code**
  By submitting your assignment, you confirm that:
  - The work reflects your own understanding
  - The solution was not implemented by someone else
  - Any use of AI is documented honestly in your reflection

Evaluation Criteria

- **Understanding of Basic Programming Concepts:**
  - Proper database design and normalization (third normal form).
  - Ability to implement backend functionality to add data.
  - Efficient data retrieval without unnecessary queries.
- **Code Quality (nice-to-have):**
  - Clear and readable code.
  - Appropriate use of comments and documentation.
- **Functionality:**
  - Correct implementation of the required features.
  - Ability to display data dynamically from the database.
- **Presentation:**
  - A user-friendly interface.
  - Logical organization of information on the frontend.
- **Version Control Usage:**
  - Effective use of GitHub and version control best practices.
- **Optional Features:**
  - Implementation of additional features such as filters or frontend forms.
  - Inclusion of tests to verify functionality.

Notes

- **Technology Choices:**
  - **You can use any programming language and framework.**
- **Time Management:**
  - We understand that time may be limited. Focus on completing the core tasks first.
- **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!