Table Football Coding Challenge

Introduction

This assignment is designed to give you a glimpse of some of the challenges you will be facing in this role. Please be aware there are no perfect solutions - for us, it's more important to see how you find solutions, process your ideas, structure your thoughts and how you make your decision paths.

Be creative but realistic about what's possible. We are thrilled to get to know a bit more about the way you solve tasks.

Background

Your friends are table football addicts. Unfortunately, they are also extremely bad at keeping score, which results in endless fights about who is the table football champion. To solve the situation once and for all, you decide to write a simple program that keeps track of the score during the game, stores the final result after the game ends, and displays some statistics.

User stories

US#1: As a user, I want to start a new game between two teams so that I can keep track of score as the game is being played.

US#2: As a user, I want to create a game that has already been played so that I can enter the result of that game.

US#3: As a user, I want to be able to create teams with one or two players so that I can reuse the teams when I create new games.

US#4: As a user, I want to see a dashboard with team and individual player statistics so that I can see who is the ultimate champion.

User Interface

Dashboard

The dashboard shows all teams/players sorted by Win Ratio, GD

`(Numbers shown below are examples)`

Team/Player Name	Games Played	Wins	Losses	Win Ratio	GF	GA	GD
Player 1	10	8	2	0.8	48	12	36
Player 2	12	6	6	0.5	30	12	18
Player 3	10	2	8	0.4	12	30	-18
Player 4	12	4	8	0.33	12	48	-36

```
Columns
- Team/Player Name
- Wins
- Losses
- Ratio (Games Played/Win)
- GF (Goals For)
- GA (Goals Against)
- GD (Goals Difference)
```

Team/Player page

Show the history/log of games played by one player

(Optional) Team n VS Team m page

-Shows the confrontation statistics, for example:

Team N wins in direct confrontations with Team M: **10 - 2**

Shows the history/log of games played by one team vs another

Acceptance Criteria

- The application follows the requirements found in this document
- The user interface is built using a JavaScript framework (React, SolidJS, Preact)
- The user interface talks to a REST API written in (node.js, C#/.NET, PHP, Go)
- The API is self-documenting (e.g. using something like OpenAPI)
- The application connects to and stores the data inside a database (PostgreSQL, MariaDB/MySQL, SQLite)
- The project is properly version tracked and hosted on Github so that it can be cloned
- There are multiple commits that show the progression of your code (i.e. not one big commit once you finish coding the project)
- The project has no external and/or system dependencies **which are not listed in the README.md**
- The project contains a README.md with a `How to run this project` section
- The project runs without errors
- (Optional) The project runs with Docker (see the "how we will run your project" section below)

How we will run your project

In order to share the project with the rest of the team, you should create a simple README.md at the root of the project which lists all required dependencies, and describes what is needed for the project to build and run (e.g. package.json file for node.js, composer.json for php). You can also use Make (Makefile) or Docker to build your project. We should be able to run your project by following the steps in your README.md file. If that is not the case, or if some system dependencies are not listed, we will not run you project.

If you use Docker, you should have Dockerfile and/or a docker-compose.yml which lists all required, as well as optional development dependencies, and that runs your project. And we should be able to execute "docker build ." command from the root of your project to build the image (only if you use Docker).