Recruiting Challenge

For Full-Stack Developer

# Introduction

The tasks in this challenge are inspired by the Dungeons and Dragons role playing game.

The **dragon-data.json** contains data about dragons.

When evaluating your response to the challenge we will look at:

1. how you have solved the tasks
2. how you structure your code
3. which solutions you have chosen to solve the different task
4. We will also appreciate to see examples on how you would ensure that your code works as intended

Be creative and feel free to extend the tasks with additional functions, for example for sorting, filtering or searching in data.

As the cases only contain a limited amount of details, feel free to specify any additional assumptions, which are relevant for your answers.

Your code and written answers will be used for our internal evaluation and the subsequent interview.

# Task 1: Create an “There be dragons” application

Please see the **index.ts** file that contains instructions on the different sub tasks.

All the tasks are about processing some json data and provide some output.

You can use the index.ts file as basis in your project or create another structure.

Structure the code for the application to be used as a backend for a react web frontend.

# Task 2: Create a FE app for “There be dragons”

Create a simple web app in React with features to support these user stories:

1. As a user I want to see a list of dragons sorted by their hitpoints
2. As a user I want to get the attack damage of a specific dragon – this includes a random factor from the roll of a virtual die (done by the backend application).
3. As a user I want to select two dragons and perform a battle. I want to see the result of the battle.

You can use any third-party libraries or frameworks that you find necessary.

The React app should use a state manager to store and manage calculated attack damage and the performed battles.

Bonus task: Use CSS to style the component and make it visually appealing.

Please submit a link to a GitHub repository containing your code. The repository should include a README file with instructions on how to run the application.

# Task 3: Describe Testing Strategy

Please describe the optimal strategy for quality assurance of an application consisting of a backend and a web frontend as in the above example.

Using the app from case 1 and 2, how would you have implemented automatic testing using a TDD approach if you have had more time?