Ginkgo Backend Coding Challenge Grading Rubric

This provides an explicit grading guide for the Backend Coding Take-Home Challenge. It's split into three sections:

1. **Basic Requirements** (70 points), which ensures that the submission meets the basic criteria
2. **Project Breakdown** (30 points), which rewards a submission that went above and beyond the bare minimum on a more detailed level
3. **Bonus Points** (20 extra points), which rewards a submission for including things that weren't required but were greatly appreciated. In short, these points allow us to be more qualitative with our approach

To keep things simple and round, the max total is 100 points (Bonus Points can take a submission up to, not past, 100).

Basic Requirements (Each is worth 7 points, 70 points total)

These requirements cover the bare minimum. A submission either earns the full 10 points or none at all for each requirement.

- ☐ A live version of the solution runs behind a URL that we can access and test against
- ☐ Web client accepts DNA sequence as a form of input
- ☐ Alignment randomly searches through each of the provided proteins and **can handle the test inputs below**
- ☐ Can issue multiple alignment requests before the first submission completes (DNA alignment happens asynchronously)
- ☐ Web client automatically updates with the result. (7 points) The result must include:
    - ☐ The name of the protein found (3 points)

    - [ ] The start and end position of where the (sub)sequence was found in the protein's sequence (4 points)
- [ ] The application persists results on a refresh (through user and/or session tracking)
- [ ] The application is written in Python and Javascript
- [ ] The application is a service. It is not triggered by an IPC library
- [ ] There's a README with instructions on how to set up the project
- [ ] Running the project should happen in one command

User Input Tests

1. **Basic Input**: ATCG (should return any of the 10 proteins)
2. **Targeted Input**: CCTTTTCTCTCGAGCGGAGGGAAAACGGAA should match the genome NC_000852 (NP_048806.1 is one example of a matching a protein, not sure if there are others)

Project Breakdown (Up to 30 points total)

A submission can earn an additional 30 points depending on how much effort went into it.

- Alignment Algorithm
- Session Management
- Scale / Asynchronous Execution
- Style
- Testing
- User Interface

Bonus Points (20 Extra Points)

The distinction between this section and the Breakdown section is that the Breakdown rewards a submission for excelling in the things that were asked for. This section rewards submissions that include unexpected, effective solutions/approaches and additional goodies: For example, one candidate added a performance test framework, while others flexed their full-stack abilities and demonstrated a deep understanding of front-end JavaScript frameworks. While these are not required, they should be highly valued.

There are also instances where a candidate may go against the grain and does not follow requirements but still hands in something worthwhile. For example, one candidate went against the Python and JS requirements and turned in a fully working submission written in Clojure. This ought to lose the full 7 points in the Basic Requirements section, but surely a working submission should count for something, right?

Lastly, the take-home challenge is a time-consuming project and not everyone has the time to submit a perfect attempt. If you felt like the candidate came up short on the actual score but deserves more, this is the place to reward them for their efforts outside of a strict, by-the-numbers evaluation.

In short, this section can reward submissions for qualitative features.

Use your discretion to determine the number of bonus points that should be rewarded.