Date: 01/20/23

Programming Challenge - Backend

As we consider the potential of working together, we'd like to see how you handle a self directed project.

This challenge includes a project brief, requirements, and criteria for presenting your solution. The design brief is just like it sounds; brief—which means you'll be working with a minimal amount of background and context. That's hard, but it's also realistic, and a big part of what we want to see is how well you research, adapt, learn, and innovate - and how you explain technical choices and present your solution to the team.

How You'll Be Evaluated

We'll be evaluating your solution across three criteria.

Code Quality

Did you use clean, consistent, readable code? Is the organization of files and folders within the project logical and consistent?

Execution

Did you meet the project requirements? Did you go above and beyond in any given area or complete any of the optional suggestions?

Clarity

How easy is it to read your code and understand how it is working and what it is doing? How well documented is your code and how accurate is your README?

Requirements Brief

Using the provided JSON dataset, build a GraphQL server to return data about the users, hierarchy, and relationships within.

- Each person has an id, firstName, lastName, jobTitle, departmentId, and managerId.

- The person with jobTitle “CEO” is the root of the hierarchy and has no managerId.
- Each departmentId corresponds to a row in the Departments array.

Requirements

1. Our services are all in TypeScript and we are heavy on Node (and this position would be likewise), so we’d prefer the same for this (if not Typescript, modern standards-compliant Javascript is also acceptable). However, if you feel the best way to present your abilities is in another framework or language, feel free to use that and explain your decision.
2. We should be able to clone it from a git repository and start the server either using a standard `npm i && npm start` command or something else explained in the project README.
3. The API should support retrieving users and departments, either by id or as a total list.
4. The API should support updating user information.
5. Given a user, it should be possible to explore relationships and their hierarchy to pull in other users as well as their department.
6. There should be some automated testing of the provided code. Which can be run via a single command such as “npm test” or something equivalent. Document how to run the tests in the README.

Optional:

1. This application is for a back-end position, but if you want to go above and beyond and create a UI (preferably in React) feel free. If the server is just available to be explored by GraphQL Playground, that’s fine.
2. The effects of the mutation do not need to persist if the server is restarted, but optionally you could store the data in a database rather than local state. If this requires an external database server please provide the service via docker-compose and the corresponding instructions on how to start it in the README.
3. Provide a way to search across users by name or jobTitle.

Effort Expectations

We don't expect you to spend more than 2-4 hours on this. Internally, it's common for us to share unfinished work with each other, so if certain aspects aren't complete, just make a note of that during your presentation and explain what you would have done going forward.

Evaluation

Once you are done please email us a link to your git repository or send all of your code in a zip file. We will evaluate your code using your README and code comments as a guide and we will get back to you with a followup as soon as we can.

Thank you for participating and any work completed on this brief is yours to keep and reuse as you see fit.