XXXLdigital

Thank you very much for your interest and willingness to participate in our recruitment process.
As you already know the next step is the development of a technical challenge.

Before you start with it, we would like to share some points you should keep in mind.

1. You might not find clear instructions for every detail, so please provide information on your solution process and how you tackled the challenge.
2. We care about best practices. Please have them in mind on the presentation of your solution.
3. For any text content (comments, dissertation, documentation, etc.) please add a `README.md` file. We expect from you instructions we may need to get your application working.
4. We try to evaluate our candidates objectively, so please remove personal information (full name, usernames) from your solution so that we can review it anonymously.
5. To share your solution with us, please transfer your files in a ZIP archive using a file sharing service (Firefox Send, WeTransfer or others).

React Todo App

Let's add quality to the react todo app received in the folder. Setup and configure a protected workflow for your repository on github that a developer would follow to contribute code to the project.

Install dependencies. Make sure you already have nodejs & npm installed in your system.

```
$ npm install # or yarn
```

Run it

```
$ npm start # or yarn start
```

Acceptance Criteria

1. Create a document with a brief test strategy for this product list while having the following points in mind:
    - what are your first steps to approach the testing
    - what types of testing would you consider and why
    - propose the technology to be used
2. Ready to get down to the job:
    - setup a testing framework that runs as part of your CI workflow
    - setup an automation framework that can be used by your tests
    - write a couple of examples tests that show your system working and demonstrate how developers would write automate tests using your chosen technologies.

Bonus Round (not required, but nice-to-have)

- Usage of code quality tools such as eslint, prettier, typescript
- Integration of your tests into a CI pipeline
- Tests run against a testing service (SauceLabs, BrowserStack or similar)
- A “beautiful” test report template to reflect test results
- Surprise us…

How we evaluate the result

Prioritized from most important to least important, here are our evaluation criteria:

1. Acceptance Criteria: Have all acceptance criteria been fulfilled correctly?
2. Documentation: Did you documents and how to run your tests well? Is your written communication clear and easy-to-understand?
3. Code Quality: Is the code that you've written clean, well-structured and easy to understand?
4. The extra mile: Everything you did on top of the acceptance criteria.