React test Assignment

Coding skill assignment

Table of contents

1. Introduction

This task is used as a skill assessment assignment which Q uses in its selection process to determine the best candidate for the job. All test assignments should be solvable within a maximum of 8 hours by an experienced developer. These assignments are not a part of any client project and are not a part (and will not be) of any production project. We do however expect you to architect the code to support scaling into a bigger application. Q reserves the right to limit the disclosure of the assignment contents. This assignment is aimed at all developers' seniority and experience levels.

2. Assignment

Overview

- Create a React application that shows a list of Posts and associated post Comments that are fetched via provided API.
- Push the code to github + describe the build/run process in a Readme file
- Write approximate time needed to finish the assignment

Features

- Create 2 routes: '/posts' & 'post/{id}'.
- Posts route should show a list of posts and associated comments. Every post should have a user's name associated.
- Create a search input and filter posts by user name using an input field.
- Clicking a post will redirect to a new page
- **MUST!** EVERY component once rendered must log in the console 'Hello from <insert component name>' and should be overridable per component. The part 'Hello from' must be sent to a component via props and defined only once within the scope of the application. It should look something like console.log('${propsmessage} ${componentName}'). Feel free to name the variables as you see fit.

Conditions

- The UI is up to you. This is a React oriented test but at least a minimally usable layout that does not break and is appealing to the eye is required.
- Do not use any 3rd party state management solution. Again this is a React oriented test. That does not mean state management can't be handled in a well structured way.
- Do not use any 3rd party UI component libraries. The UI of the app can be very minimal and does not require 3rd party component libraries.
- Do not use CSS utility libraries (e.g. Bootstrap, Tailwind).
- When writing components try to find a way to make them **reusable** and resilient, meaning they can easily be integrated into other applications. This part of the test is very important.
- Try to use some of the more advanced concepts like HOC, Render props, Compound components etc.
- You are free to use any React bootstrapping tool (like create-react-app or vite)
- It is also highly recommended to use TypeScript. If not, make sure to use type checking With PropTypes
- You are free to structure the code in any way you like (folder structure) but try to make it as real-world as possible

- You are free to include any tools that you use in general that can help you work on this task like linters, code style checkers, UI component testers etc.
- Unit tests are not mandatory but highly recommended

API

- JSONPlaceholder

3. Have questions?

Feel free to contact us at talent@q.agency, preferably the Recruiter you've been in contact with.