Frontend Developer test

Instructions

This document has the definition for a practical exercise which should be doable by any candidate who applies to the current ***Frontend Developer*** open position. The MediSprout technical team expects the candidate to complete the following test.

Please read carefully and follow these instructions to complete the test.

Web Development Test

Create a GitHub Repository with the following App to share with us.

The stack

Please use a recent if not the latest version of the following technologies:

- React
- TypeScript
- React-router
- Optional, but encouraged:
  - React-query / Tanstack-query (for query and data management)
  - Jest (for testing)
  - MaterialUI (as a design system)

Requirements

Setup

Create the React application. For this you can use any bootstrapping method such as vite, create-react-app, or anything else as long as it fulfills the tech stack requirements (adding additional technologies is OK).

If you use a generator such as vite, make sure you start the project using TypeScript.

The app will contain two distinct routes, which will be explained below.

The app flow

The application will access http://jsonplaceholder.typicode.com/posts/1/comments a single time; pull down all of the data then store it locally. It will then use this store when each page needs data to be shown, so no more HTTP requests should be needed after the initial page load.

Feel free to lay this out how you like but the architecture of the solution is an important part. Assume this will soon be a much larger solution so keeping to best practices that would be used in a larger solution is the key (layers, SPA, frameworks, async, routing, unit tests, etc). UX/Design is not so important, but we will evaluate proper use of HTML, semantics and CSS.

Route/View 1

A page for displaying a list of all comments, clicking on a comment will route the browser to route 2. Each item will display:

- Name (only first 20 characters, append '...' if over 20 characters)
- Email
- Comment (only first 30 characters, append '...' if over 30 characters)

Route/View 2

A page for displaying all info about a specific comment. This should include a back button which takes the user back to the list.

Commented incremental commits on your repository while building the application will be appreciated.

Unit Testing (optional)

Testing is optional but will net you bonus points. If you consider adding tests to the app, we advise you to try a component test, where a component is being rendered and you assert some of the rendered content, and a utility test where you test parts of the application that are not a component.