Tomo Spice Blend Take Home Challenge

FIRST: thank you so much for investing time and energy into this process with us. We're very grateful!

SECOND: The goal for this challenge is for you to highlight your technical strengths on the Front-End. Do your best, and spend the time you need to give us a clear understanding of your technical skills and interests. We'll review your submission, and then spend the majority of the follow up interview discussing your solution.

Background

We're working on a spice blend application. This is important because spices make food delicious. We have a single page React app that we've started and we would like to enhance this application in a number of ways to offer our spice blend clients a better experience.

Getting Started

Run `npm install` and then `npm run dev`. This should install the application in its entirety, open a webpage to localhost:5174, and watch for any of your changes in order to hot reload the page. You'll need node/npm installed (hopefully you have those already...), and it should work. If you run into troubles, consider those a challenge, and document what you needed to do to get the application running.

Tasks

We have a lot to get done, and we want you to feel empowered to solve issues however you think is best and to show off your strengths! Feel free to add any libraries/dependencies/etc (you'll just want to explain your choices), with the exception of kit/component libraries such as "material-ui" or "bootstrap-react". Our home route does some good work of fetching a list of spices and blends and then having a basic details page (data not super factual), for each spice. We need to accomplish the following items:

- DONE -- Blend details page correctly lists the name of all included spices on first load.
- DONE -- State management between pages, so as not to re-request data.
- DONE -- Load a "blend of blends". Implement functionality to display all unique spices contained within a blend. This includes spices directly associated with the blend as well as spices found in any nested child blends. Blends may be nested to an arbitrary depth (n levels).
- Add ability to create and view a new blend.
- DONE -- One unit test.
- DONE -- One new "feature" or "refactor" that shows what you consider to be your FE strengths
- DONE -- One new "architecture pattern", or least be prepared to talk about one you'd like to add.

Technical Background

We're using React and vite to let us get up and running quickly. We haven't done much in the ways of customization here, and we've elected to use typescript by default. Additionally, we have setup Prettier for ease of code formatting, and have imported Tailwind for ease of styling. Currently, the code to use Tailwind is commented out, but you can change that by updating `main.css`. If you're not a fan of (or familiar with) Tailwind feel free to use something else.

An interesting piece is that we're using MSW to mock out the backend and to allow us to have full stack interactions. You'll be able to query against those requests as if there was an actual API service on the backend. If you're so interested, you may even add/alter/research some of those mocked endpoints in the `mocks/handlers.ts` file.

NOTE: Network requests will get logged in the console, and will not show up in your network request tab within dev tools.

You can run prettier over your code at anytime by running the `npm run format` command. This will make your code standarized and more readable for us.

How to submit?

Any way that we can see your work is fine. Having a github repo you can share with us, zip folder, ETC. Again, the key here is to allow us to get a little insight into your development skills and allow us to have a follow up discussion.