Create a form for touring bands!

Overview

For this exercise, you'll create a simple form that allows users to purchase tickets to a concert for a given band.

You'll find included in this repo:

1. Three JSON files (located in `src/band-json/`) that represent the data structure that the form should be able to accommodate
2. A Wireframe ("BandTickets_Anonymous.png") that should help guide you as a loose representation of what the form should be structured like
3. A starter React project

We'd like you to build on the included React project that has been scaffolded using `create-react-app` . Feel free to add any javascript libraries that you typically use or think are a good fit for this project. For styling, you're welcome to write custom CSS, use a component library, or do some combination of both.

Acceptance Criteria

The acceptance criteria for this exercise is that after consuming any of the JSON objects from the band-json directory, the form should include:

1. The band name, description, location, date, and image to learn about the concert
2. A list of ticket types that includes their metadata and the ability to add any number of tickets of each type
3. A total amount section that adds up the ticket prices
4. Some basic inputs for adding credit card and personal information
5. A button to purchase the tickets

Some Things To Consider

When the "Get Tickets" button is clicked, you can add a `console.log()` or an `alert()` with the data that would be sent to a hypothetical backend server. This exercise is specifically geared towards candidates applying for a Front End focused role at ATS, which is why we are asking for folks not to add any data persistence or backend logic.

The tour date cost is listed as cents, so `500` would be `$5` .

The wireframe is a rough estimate of the layout of the form that we expect, but it's not necessary to match it exactly. The most important parts are the general page structure (two columns, with the description area on the left), and an approximate location of where the various page elements should be.

Rough time estimate for this exercise

Doing a technical exercise on top of having all the other stuff you have to do during a week can be a challenge, so first off: thank you for taking the time to do this take-home.

In general, we recommend spending **about 2 hours on this exercise**, the first chunk of which is likely just reading through this document and choosing an approach. If you can't find that amount of time because of work or life or anything else, get through as much as you are able to (and please feel free to reach out to get additional time!). We'll take a look at your submission no matter how far long you've made it in the process.

If you complete the core acceptance criteria, and want to get creative with the exercise, then you might consider 1 or 2 of the following options:

1. Style the forms so they look fabullooousss
2. Write some tests using Jest (or any library you prefer)
3. Add types
4. Deploy this somewhere
5. Make it Web Content Accessibility Guidelines (WCAG) compliant

Questions

If you have questions about this assignment, from the specific to the broad, please feel free to reach out to your recruiter point of contact or to the hiring manager. Good luck!

Submission

To submit your solution, please upload a text file or `README` to Greenhouse that includes the following:

1. A link to your code that we can access, such as a Google Drive folder or a Github repo
2. A description of how to run your code
3. Anything else we should know

If you're having any trouble at all with Greenhouse, feel free to email us the file instead.