Exercise: Setting up the HTML document

Overview

Using semantic HTML and understanding how it works in React is the next fundamental step in the process of creating the reserve-a-table web functionality for Little Lemon. In this exercise, you will organize your JSX code so that it follows HTML5 best practices.

To revisit using semantic HTML in webpages, you may find it helpful to refer to the following lesson items in the HTML and CSS in-depth course now or as you progress through the exercise:

- Semantic tags
- Semantic HTML cheat sheet

Scenario

By completing this exercise, your React app will use a semantic HTML structure to better describe the intention and semantics of the document. As you have learned, semantic HTML is important for describing the meaning, or semantics, of your HTML document. It also assists accessibility software, such as screen readers, in describing your content to users with disabilities.

Instructions

Step 1: Set up the initial semantic structure

- Open your project in Visual Studio Code.
- Open the App.js file.
- The App component's root element uses a `div` element. As this is not a semantic tag, replace it with a React fragment.

Tip: Remember, a fragment starts with a `<>` tag and ends with a `</>` tag.

- Review your UI/UX and decide how each of the sections of the design will be represented using semantic tags.

Tip: Use the `header`, `nav`, `main` and `footer` semantic tags.

- Inside the fragment, add the semantic tags.

Step 2: Break the sections down into React components

Now that you have planned your semantic structure, it is important to decide which sections will be dynamically updated and if those sections should be child components of the App component.

Tip: For simplicity, you can create a component for each semantic element you added to App.js.

- Create a JavaScript file for each semantic element that will be a React component, for example, Header.js, Nav.js, Main.js and Footer.js.
- In each component, return the semantic HTML element.

- In App.js, replace the semantic tags with the corresponding child components, for example `<header> </header>` will be replaced with `<Header/>`.

Step 3: Add the logo and navigation elements

The initial components are now set up. You will now add the details for the logo and navigation.

- Add the Little Lemon logo to the appropriate React component using the `<img>` tag.
- Add the website navigation to the `Nav` component. It is important to add a hyperlink for each key page of the website based on your mockup.

Tip: Use a `<ul>` element and add an `<li>` element for each `<a>` hyperlink.

Step 4: Add the footer content

Review your design and add the necessary HTML elements to the footer component.

Conclusion

By completing this exercise, you have now set up the semantic foundations to build the rest of your application. As you progress through the course, ensure that you apply the appropriate semantic HTML elements to the content that you are building.