Styling Components Lab

In this lab we'll give our ugly components a little style. And we'll do that in a variety of ways. We'll start by applying a popular library that implements Google's Material Design look and feel. Then we'll apply some low-level specific styles on a couple of components using CSS embedded in JS.

Using a global stylesheet

1. Install Material Design Lite (MDL).

```
npm install material-design-lite
```

Remember that importing the css file in an outer JavaScript file will cause it to take effect in nested React components. So ...

2. Import material.min.css into your index.js file so that it will effect every component.
3. Go back to your navigation menu that is in App.js. Add some material classes to style it like a Material Design menu. Here's a link to the documentation for the MDL menu: https://getmdl.io/components/index.html#menus-section
4. As we go forward, feel free to style any of your components using the things you find in MDL.

Bonus! Making the MDL actions work

This lab is really only focused on applying the styles of MDL. But it might be fun to get the actions of MDL to work as well. So here's what to do if you want all those actions to work also.

5. Edit index.js and add this line to the top:

```
import "material-design-lite/material.js";
```

6. Run and test. Clicking on your menu should toggle the dropdown as expected.

Creating local JavaScript styles

7. Host your <Login> component in App and take a look at it in the browser. It probably needs some styling help.
8. Open your Login component in your IDE.
9. Apply styles attributes to the inputs, labels, and button. Something like this:

```
style={inputStyle}
```

10. Now add some objects to the file to hold those styles. Don't forget that they'll be POJSOs with keys that look like CSS properties and values that look like CSS values.
11. If you've written this right, you should be seeing some new looks in your component. Go ahead and take a look. Adjust until you like the look of your login component.

Importing CSS/JS from an external file

Let's do the same for the <LandingPage>. The poster in LandingPage may be too big or too small. The text sizes and decorations may be unattractive. Let's fix all of that.

12. Create a new file called LandingPage.css.js.
13. In it, make a couple of JavaScript objects for the posterStyle, titleStyle, and taglineStyle.
14. Properly export those objects.
15. Open LandingPage.js in your IDE.
16. Import your new CSS/JS file
17. Apply a posterStyle style to the <img> tag that is holding your poster image. (Hint: don't forget the curly braces).
18. Run and adjust until you like the size.

19. In the same file, adjust the title style and the tagline style.
20. Again, adjust everything until it looks good to you.

21. Bonus! If you have extra time, feel free to adjust any other styles in your components.