Stateless Functional Components Lab

Now that we have a project created, let's create a few components. Please create them as stateless functional components. Do not be overly concerned with how they lay out or how they are styled for now. If you are ahead of other teams at the end of a future lab, you can go back and style them. For now, just get them working.

For each component, you will:
1. Create the component using your own handwritten, hardcoded data.
2. Include its tag in App.js so you can see it in the browser
3. Do a little adjustments on each component
4. Remove this component from App.js
5. Move on to the next component

Create components for each of the following things.
- LandingPage
  - Heading that says "What do you want to see?".
  - A date input to pick a day to see a show
- FilmDetails
  - Heading with the title of a movie
  - <p> with a movie tagline
  - Showing times list
  - Poster image (feel free to set the src property to one of the posters we copied in the create-react-app lab)
  - <div> with a description of the movie.
- PickSeats
  - A heading that says "Where would you like to sit?".
  - Movie title
  - Showing date/time
  - Theater name
  - Listing of tables in the theater.
  - Each table has a listing of up to four seats.
- Checkout
  - A heading that says "Checkout"
  - An HTML table with one item (a seat or food item) per row
    - Column for Item name
    - Column for item price
    - Column for item quantity
    - Column for line total (item price * quantity)
  - A row for subtotal.
  - A row for tax
  - A row for Total
  - A button marked "Purchase"
  - A link for "Keep shopping"
- Login
  - A heading that says "Log in"
  - Email textbox
  - Password box
  - Login button
  - Labels for each

Reading data from a local file

In preparation to read some data in the components, let's pull some good JSON data into our components.[1]

1. Look in the server/json folder. You'll see two files called films.json and theaters.json. Copy those into your client/src folder.
2. Open LandingPage.js and import films into it like this:

```
import films from './films.json';
```

3. Inside your LandingPage.js, go ahead and do this:

```
console.log("films are",films);
```

4. Run and test. Make sure you can see films in the console.
5. Add a <section class="filmsList"> to the bottom of your component's JSX.
6. Add a <section class="film"> inside that.
7. Inside that section, create a panel/card/whatever you want to call it to advertise a movie to the user. Load it with an <img> pointing to a movie poster (/img/posters/1.jpg for example), an <h3> with a fake movie title, and a <p> with a fake movie tagline.
8. Run and test. Make sure you can see your one fake movie.

Simple one-way binding

We're going to learn all about React expressions later, but for now let's use them to do simple binding from our business logic to the JSX.

9. In the film section in your list, find your fake title and make it read like this:

```
<h3>{ films[0].title }</h3>
```

10. Do the same with the tagline and poster_path.
11. Run and test. Do you see good data and a path?
12. If so, copy and paste that four times and change the "0" to a "1" for the second, to a "2" for the third, and "3" for the fourth.
13. Run and test again. You should see four good movies that our users can choose from.

---

[1] It's actually webpack that allows us to do this. create-react-app auto-includes the webpack-json-loader so we can read JSON data just like this.