Creating the Store Lab

Imagining that we have a register user page, we'd like to maintain the data using Redux. This means we have to create a store, read state from it, and eventually change state by dispatching an action to it. Let's set that up in this lab.

Adding a minimal redux store

1. In the src directory, create a new file called store.js.
2. In store.js, go ahead and create a function called reducer. It should received two parameters, state and action and return state. Yes, that same state it received.
3. Also create an object called initialState. It can be an empty object.
4. Create a const called store by running this JavaScript:

```
export const store = createStore(reducer, initialState);
```

5. Don't forget to import createStore from redux at the top of the file.

```
import { createStore } from 'redux';
```

6. Run and test again. It should be compiling but doing nothing different.

Using it in a React component

7. Find App.js in your project. In it, import your store object.
8. Add this to the App class:

```
constructor() {
  super();
  this.state = store.getState();
  console.log("App state", this.state);
}
```

9. Run and test. Did you get something in the console?

Dispatching an action

10. Edit Register.js. Import store at the top like you did in App.js.
11. Look for a method called changePerson. In there, call ...

```
store.dispatch({type:"FAKE_ACTION"});
```

12. Run and test. It should run, but nothing different should be happening.
13. Now edit store.js again and add a line to your reducer function. You should console.log("In the reducer", action);
14. This time when you run it and type into any form field you should see something in the console. Give it a try.

Once you've got your store created and a reducer running, you can be finished.

Bonus! Refactoring the Redux structure

15. Make the store easier to manage by moving it into its own subdirectory called "store". This is where we'll put all of our Redux store-related things like reducers, middleware, action creators, and so forth.
16. Don't forget to run and test again to make sure that it still transpiles and runs properly.