IMY 220
Assignment 7:
React.js

Due: Wednesday 12 October @ 13:00

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- The aim of this practical is to use React.js to create a form that validates a username and password and only allows users to submit if they are valid. Once the user has successfully logged in, they should be presented with a page that displays their name and a logout button.
- All of your code for this assignment must be written inside your own files called *index.html* and *script.js*.
- You must use CDN versions for the react library and babel.

Section 1: UsernamePasswordInput component

Create a file called *index.html* which includes CDN links for all the necessary files for you to use React.js as well as the Bootstrap CSS. Create another file called *script.js* that you will use to write all of your JS code. Inside *script.js* define a React component called UsernamePasswordInput which renders two input boxes (for the sake of testing, define it as a text-input, not a password-input), which must be styled with Bootstrap. This component must encapsulate functionality for validating its content. For a password to be valid it must adhere to the following:

- must be at least 8 characters long
- must contain at least one capital letter
- must contain at least one digit.

For a username to be considered valid it needs the following:

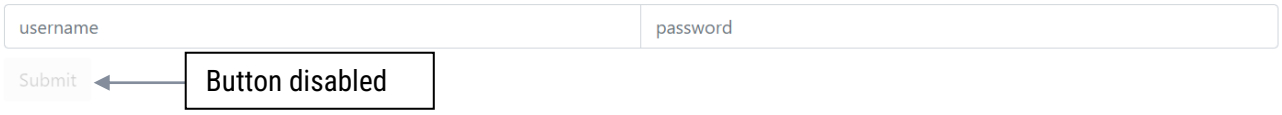
- must be longer than 2 characters
- must start with a capital letter
- must not contain any special characters

While the password or username is being input, i.e. when the contents of the input box are being changed, the component must call a member function called *checkPass()* or *checkUsername()* which validates the password or username respectively according to the above specifications and calls another function called *validatePass()* or *validateUsername()*, which will be defined as its prop, with the result of the validation. In other words, when a UsernamePasswordInput component is rendered, it will be rendered with a prop called *validatePass* and *validateUsername* which will refer to a function, for example:

```
<UsernamePasswordInput validatePass={someFunctionHere}
validateUsername={someFunctionHere}/>
```

Section 2: LoginForm component

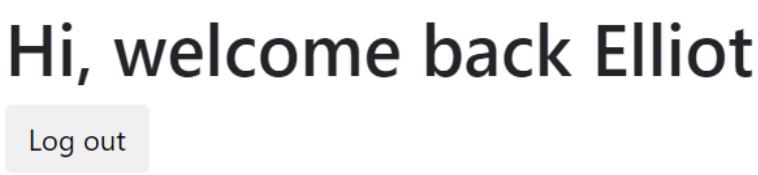
Define a React component called LoginForm which contains a Bootstrap input-group div which contains a UsernamePasswordInput component and a button for submitting the form. The button must be disabled by default. If the username and password entered into the UsernamePasswordInput is valid, according to the specifications above, the button must be enabled, but if it changes back to being invalid, the button must be disabled again. You will need to use **component state** to achieve this, which you must manage through the function passed as the *validateUsername* and *validatePass* prop of the UsernamePasswordInput component. In other words, as the contents of the input changes, a member function of the LoginForm component must be called, which updates the LoginForm state, which reflects in the button being enabled/disabled. For example, if an invalid username and password is entered:



And if a valid password and username is entered:

Section 3: ProfilePage component

If a valid username and password is entered and the user selects the submit button, the user must be taken to a profile page where relevant information should be shown and the LoginForm component should not be displayed, you must also make use of component states for this section. There should also be a log out button that resets the global state. Below is an example of what this should look like when a user successfully logs in as 'Elliot':



Submit ONLY the following file(s) according to the submission instructions.

- *index.html*
- *script.js*