**FINAL ASSESSMENT EXERCISE**

Your repository: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OVERVIEW

Build a React & Redux application based on the following requirements. You will be graded on functionality and CSS. We’ve given you a starter repository with React, Redux, and Redux-Thunk dependencies and a few other files included. Clone your repository, and make sure to run npm install before you start. When finished, simply commit and push your work.

If you are on Windows, be sure to run ./run-me-on-windows.bat after you clone.

REQUIREMENTS

**FUNCTIONAL *[10 pts]***

1. The Redux state initially contains a loggedInUser property, set to null.
2. There is a SignIn component as shown below. It contains a form with username and password inputs.
3. There is a Profile component that displays the details and the image found in the loggedInUser object (from the Redux state), as shown below. (Note: the images are provided in the public folder, so the given urls should work.) (You can find examples of this object in users.js.)
4. The SignIn form should use two-way sync to keep name and password in its state up to date with the form inputs.
5. Submitting the SignIn form triggers a login action in Redux. The username and password from the form should be passed to the action. (This action is already built for you. You just need to make sure it get’s triggered from the SignIn form.)
6. If username and password provide a valid match, trigger a LOGIN\_SUCCESS action, passing it the user object that was returned from findMatchingUser. (You will need to create this action and trigger it. See TODO in actions/index.js. This is similar to the AJAX setup for dispatching RECEIVE\_… .)
7. When the LOGIN\_SUCCESS action is triggered, update the state by setting the loggedInUser to the provided user object.
8. When updating loggedInUser, the change to state must follow immutability, as always.
9. The SignIn component should only be shown when loggedInUser is null. Whenever loggedInUser is NOT null, the SignIn component should not be shown.
10. The Profile component should only be shown when loggedInUser is NOT null. This component should not be shown when loggedInUser is null.

**VISUAL *[10 pts]***

1. The body’s font needs to be ‘Open Sans’.
2. Use the following colors for appropriate elements: #63CEC2, rgba(67, 75, 89, 0.7), #F7F8F9, and #CCC.
3. Use border-radius on the following elements: form container, button, inputs, profile container, and profile image.
4. Buttons and anchors should have hover states to change the text color to #63CEC2.
5. Image and profile information must be floated to the appropriate sides.
6. The Edit Profile link must be at the bottom and centered inside the profile container.
7. Form and profile container must be centered on page.
8. Use margin on form elements to add spacing between them.
9. Profile information should be displayed as listed elements without bullets.
10. Transform the headers of the form and profile container to uppercase (using CSS).



