

CPSC 436I - Assignment 2

DUE DATE: **Monday, June 1 (10PM)**

- Final code must be committed to a Github repo before 10pm
- **Keep your repo private**

Postings Website 2.0!

For the second assignment, you'll be essentially re-creating your assignment 1, but this time with React+Redux! The requirements are slightly different, and you may use as much of your assignment 1 code as you would like, or create something new if you're feeling adventurous.

We're expecting the following:

- 1) You **MUST** use React AND Redux, and start your assignment with the **create-react-app** boilerplate code. Remember, you will be asked to explain sections of your code.
- 2) Some kind of text input or textarea, to write out a message, as well as a button to add the message to a list. (A button to clear the form is optional.)
- 3) Some pre-loaded message list data **stored in your reducer**.
- 4) A list made up of list items that contain the message text (it should be updated with the new message whenever you click the add button)
- 5) Basic styling. (Do not spend too much time on this, and re-use as much as you'd like from Assignment 1!)
- 6) ***NEW*** The ability to click on the list item to go to a "detailed" view. This view may appear next to the list or appear in a pop-up/dialog. The goal here is for this "detail" view to know which item was selected. The detailed view **MUST** be its own React component.
- 7) Something cool and extra! This is wide open for you to explore, and try to push your knowledge and boundaries.
For example:
 - you could have individual buttons for each message list item that will allow you to delete them (e.g. a button with an X)
 - you could have additional form elements that show up in the messages (e.g. a text input for name, a dropdown that includes different options, etc.)
 - A toggle for each list item that does something!

NOTE FOR IMPLEMENTATION: DO NOT try to take your assignment 1 code, and re-shape it into React components. Instead, start with create-react-app, and think about how you would organize your component structure. Begin to create those components, and then copy over

pieces of your assignment 1 code as needed. Remember that handling actions and rendering the view is handled much differently in React+Redux. If you find yourself needing to use HTML DOM functions (e.g. `document.getElementById`), then STOP, and try to think of the “React” way to do the same thing.

As described in the individual assignment rubric, your code will need to meet these requirements and be functional, up to perhaps a few minor glitches in tricky cases. Note that functionality includes both user-visible and console-visible issues.

It’s up to you! We’re hoping that you’ll use the above requirements as a guide, but that you’ll let your imagination take over, and build something unique and interesting!

You should be ready to demo this to a TA during your second week lab, and should be ready to answer questions about it, as well as explaining what you’ve done.

HAVE FUN!!!