CPSC 436I - Assignment 5

DUE DATE: **Monday, July 20 (10PM)**

- You will submit this assignment by sending/posting the link to your deployed project! (We will send out a form later.)
- **Make your repo private again - until after it's due**

<u>Postings Website… Complete!</u>

For the fifth assignment, you'll be deploying the postings website that you've been working on all term! It will build on the previous assignment, so start from your completed assignment 4, and begin to incorporate the requirements below. We expect you'll continue to put less time into assignments and more into the project.

We're expecting the following:

1) Deploy your postings app by using a platform such as heroku! (If you have not deployed anything before, we strongly suggest you use heroku.)

2) Ensure that once your app has been deployed, you can access it on someone else's computer, and that everything still works! (All the way from DB storage and your APIs to React, Redux, and any UI elements.)

Note: At the very minimum you should maintain the functionality based on the requirements from assignment 4. You do NOT need to maintain all of your functionality from the previous assignments (1, 2, 3), however, <u>make sure you still have all the other technologies of the assignment, including React, Redux, Express, Node, and MongoDB</u>! You shouldn't need to make drastic changes to your existing code.

Reminder of requirements from assignment 4:

- The data should now be stored in the database.
- When you add new messages, they should be added to your database. You should be able to refresh the page, or even restart your server, and your data should persist.
- Additional functionality of your choice. Examples could be:
    - Deleting an existing message
    - Editing an existing message
    - Filtering data from the DB
    - Getting details for a message (stored in the same table)
    - Getting details for a message (stored in a different table!)

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.

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 explain what you've done.

HAVE FUN!!!