

CPSC 436I - Assignment 3

DUE DATE: **Monday, June 15th (10PM)**

- You will most likely be using the same repo as your assignment 2, so submit by creating a branch called "assignment_3", and do not push any more code after the due date
- **Make sure your assignment repo continues to be private**

Postings Website... Continued!

For the third assignment, you'll be adding more functionality to your postings website, now with MongoDB! It will build on the previous assignment, so start from your completed assignment 2, and begin to incorporate the requirements below. We hope you'll find this assignment somewhat smaller in scope than the previous one, so you should be able to spend more time on your project.

Note about functionality: This will be a strange step for your assignment. You will have the front end (with React + Redux) still working independently from the data you set up in your DB. Your goal here is to get everything ready for when you add the last piece (APIs) in the next assignment, so that it goes as smoothly as possible. For now, your data will STILL NOT PERSIST.

We're expecting the following:

- 1) You MUST use MongoDB, and start your assignment from your completed assignment 2. Remember, you will be asked to explain sections of your code. (Mongoose is not required, but if you plan to use it, you may want to set it up. It may not make sense to have it until the next assignment.)
- 2) Think about how you would like to organize the data in your DB
 - o You must have a list of messages
 - o You must have at LEAST one other piece of data. Ex.
 - i) Message details such as date, length
 - ii) User data (name, id, ??)
 - iii) Other data?
- 3) Set up your DB Collection(s) to store this data
- 4) Make any changes to your front end (React + Redux) to accommodate this data. For example, if you've chosen to include the date posted, make sure that it is being handled with Redux.

Note: You do NOT need to maintain all of your functionality from the previous assignments, as long as you are meeting the requirements above. However, make sure you still have all the other technologies of the assignment, including React and Redux! You shouldn't need to make drastic changes to your existing code.

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.

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!

HAVE FUN!!!