

CPSC455 - Assignment 4

DUE DATE: **Monday, July 11 (10PM)**

- You will most likely be using the same repo as your assignment 3, so submit by creating a branch called "assignment_4", and do not push any more code after the due date
- **Keep your repo private**

Recipes Website... Continued Again!

For the fourth assignment, you'll be adding more functionality to your recipes website, now with MongoDB! It will build on the previous assignment, so start from your completed assignment 3, and begin to incorporate the requirements below. We hope you'll find this assignment somewhat smaller in scope than the previous one. We expect you'll start putting less time into assignments and more into the project starting at this point in the term.

We're expecting the following:

- 1) You **MUST** use MongoDB, and start your assignment from your completed assignment 3. Remember, you will be asked to explain sections of your code. (Mongoose is not required, but it can be useful.)
- 2) Think about how you would like to organize the data in your DB
 - o You must have the recipe title, ingredients, instructions.
 - o You must have at **LEAST** one other piece of data. Ex.
 - i) Recipe related info (e.g. completion time)
 - ii) User data (e.g. recipe creator (username, id, etc.), recipe reviewer (similar data as creator), etc.)
 - iii) Other data?
- 3) Set up your DB Collection(s) to store this data
- 4) When you add new recipes, 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!
- 5) Ability to delete an individual recipe (and it should be removed from the database)..
- 6) Additional functionality (directly related to MongoDB) of your choice. Examples could be:
 - o Editing an existing recipe
 - o Filtering data from the DB
 - o Getting details for a recipe (stored in the **same** collection)
 - o Getting details for a recipe (stored in a **different** collection!)

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 & Redux, Express, and Node! 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!!!