IMY 220
Assignment 9:
MongoDB

Due: Wednesday 26 October at 13:00.

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- Use MongoDB and React to create a list of classes.
- Download *classes.json* and *users.json* from ClickUP. These files contain the data you will need to upload to a MongoDB database and display on your webpage.
- Your package.json file needs to be correct according to the lecture notes and must list all the packages your code refers to. In other words, whoever marks your assignment should be able to run **npm install** and **npm start** and navigate to localhost:3000 and see the required webpage.
- You must use .on() for event handling and you must use the recommended syntax from the slides for adding new elements. Use the most appropriate keywords for declaring variables.

Note: for this assignment you will need to include the connection details, including the password, for your MongoDB database. These details are the ones you created the database with and **not** your login details for *cloud.mongodb.com*. Regardless, you should probably delete this database when you are finished with this module.

Section 1 – Create MongoDB cluster, database, and collections

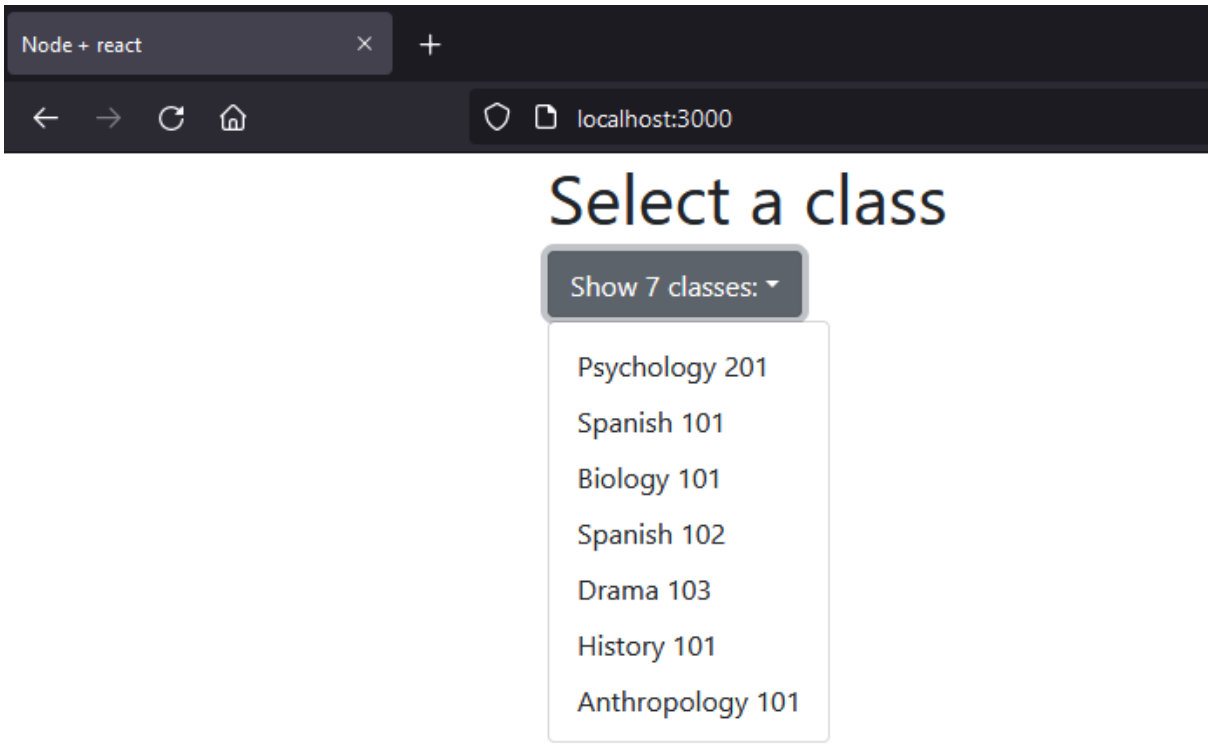
Create a MongoDB cluster, named whatever you want, and database, named DBExample, as per the notes. Create two collections named classes and users and populate them with the contents of *classes.json* and *users.json* respectively. Create a text file (*details.txt*) with the code for connecting to your MongoDB database within the MongoDB shell. The result should be that the marker is able to use the connection string inside *details.txt* to connect to your database and run **show dbs** to see the list of databases, i.e., *DBExample*, *admin*, and *local*.

Section 2 – Load questions

Using the structure for serving a static HTML page (*index.html*) and rendering a React component using Babel and Webpack, define a component named EnrolmentList in its own appropriately named file and render it inside *index.html*. Inside your *server.js* file, connect to your MongoDB database and query the classes collection to fetch only the name and code of all classes and log these in the console when the server starts up. The result should be that when the server is started the following is retrieved from your MongoDB database and listed in the console:

```
{ name: 'Psychology 201', code: 'PSY201' }
{ name: 'Spanish 101', code: 'ESP101' }
{ name: 'Biology 101', code: 'BIO101' }
{ name: 'Spanish 102', code: 'ESP102' }
{ name: 'Drama 103', code: 'DRM103' }
{ name: 'History 101', code: 'HST101' }
{ name: 'Anthropology 101', code: 'ANT101' }
```

Use socket.io to emit the data from the classes and provide it as the props for the *EnrolmentList* component inside *index.js*. The *EnrolmentList* component should render a heading and Bootstrap dropdown list with the names of the classes, which should look like this:



HINT - Use the following webpages as a guide for using MongoDB with NodeJS and socket.io:

- https://docs.mongodb.com/drivers/node/current/quick-start/
- https://docs.mongodb.com/drivers/node/current/usage-examples/find/
- https://docs.mongodb.com/drivers/node/current/fundamentals/crud/read-operations/cursor/#std-label-cursor-methods
- https://stackoverflow.com/questions/63001989/referenceerror-cant-find-variable-regeneratorruntime-on-react

Bonus

Add functionality to list the details of whoever is currently enrolled for a class when that class is clicked on. This will require you to add the code for the class as data attributes when rendering the dropdown items, add onClick functionality to each dropdown link, and define extra functionality inside *server.js*

and *index.js* to send the code for the class back to the server and retrieve all users who have said code in their "enrolled" array. For example, when clicking on Drama 103:

```
{
  _id: new ObjectId("6162c83b13dadb8fa421f48f"),
  id: 39485881,
  name: 'Britta',
  surname: 'Perry',
  age: 41,
  position: 'student',
  enrolled: [ 'ESP102', 'DRM103', 'ANT101', 'BIO101' ],
  passed: [ 'ESP101', 'PSY201', 'HST101' ]
}
{
  _id: new ObjectId("6162c83b13dadb8fa421f491"),
  id: 87458431,
  name: 'Troy',
  surname: 'Barnes',
  age: 32,
  position: 'student',
  enrolled: [ 'ESP102', 'DRM103', 'ANT101', 'BIO101' ],
  passed: [ 'ESP101', 'HST101' ]
}
```

Additional Information

- Refer to the slides and online resources for help

Submit only the following file(s) according to the submission instructions.

All files (including details.txt) ***except*** *the node_modules folder*

(Including node_modules will result in a 10% penalty).