**ASSIGNMENT 4 NOTES**

* Using React Router, Hooks, AJAX/fetch for SPA
* Style a React Application with a library
* Creating an SPA with a library
* Backend code is given in the code (run **npm install** and then **node index.js** in the backend directory)
  + Allows to search, read, create, update and destroy pets in the pet inventory
  + This will run a server on port 3001 that makes available routes
  + In case you want to “start with a fresh backend/database”, we have to stop the backend and run it again with **node index.js**
* ReactJS app should make AJAX requests to the backend from a frontend React application.
* The **normally** operations that modify state such as update, delete and add operations would not be available via GET requests for security reasons in a real-world application
* Fair amount of starter code is provided in the assignment itself.
  + Run **npm install** in the frontend directory
  + First make sure that the backend code is running, then start on the frontend code
  + After the backend is running, run the frontend with the command **npm start**

**MAIN AGENDA**

* Create a frontend that includes 4 pages: Home, Inventory, Search, and About (Using React Router)
* Home page route “/”; the inventory page should have the route “/inventory”; search page should have the route “/search” and the about page should have the route “/about”.
  + Currently active page and navigation link should be highlighted
  + The Home page should contain an image suitable for a pet store inventory
  + The about page can contain whatever text you like describing the pet store inventory.

**INVENTORY PAGE**

* Present the pet inventory data in a table. (Columns for animal, description, age and price should be present)
* Each row in a table should be for a pet in the pet inventory.
* User should be able to delete a pet from the inventory by clicking a delete link provided a delete link provided in each row of the table.
* User should be able to add a pet by filling out a form with data and clicking submit. The form should allow the user to edit a pet by clicking an edit link provided in each row of the table.
* When the edit link is clicked, the user should be able to edit the animal, description, age, and price information for that pet.

NOTE: You can either provide text input box options in-line in the same table row, or you can re-use the text input boxes used to add a pet.  If you decide to provide text input box options in-line in the same table row whose edit link was clicked, these input boxes should only appear ***after*** the edit link was clicked and they should not appear after the user has completed editing the pet data (e.g. they can revert to text). < -- This part I did not understand 😊

**SEARCH PAGE**

* Pet inventory data should be presented in a nicely formatted table. Refer to point 1. and 2. of the Inventory page
* The user should be able to search the pet inventory by entering a search term into an input textbox that is blank when the page first loads. As each character of the search term is entered into the textbox by the user, the table of pet inventory data should be filtered to present only those pets returned by the server for the search term. If the textbox is made blank again by deleting all entered characters, then all the pets in the inventory should be displayed in the table again. < -- This part I did not understand 😊

**GENERAL POINTERS FOR THE ASSIGNMENT**

* Everything should be implemented using the provided backend and AJAX in the expected ways.
* Only use the function components with hooks and effects as required to implement this functionality, do not use class components for this assignment.
* The application (table, buttons, form inputs) should be styled using Material-UI.
* Use create -react-app for this assignment.
* Assuming that the provided backend is running on the same machine, your solution should work when it is unzipped and the commands **npm install** followed by **npm start** run in the app directory should be enough to run your application.