WEBD6201 – Client-Side Scripting

Lab 3

Fun with Ajax

Due: Week #9 (Monday March 16, 2020) @ midnight.

Value: 5%

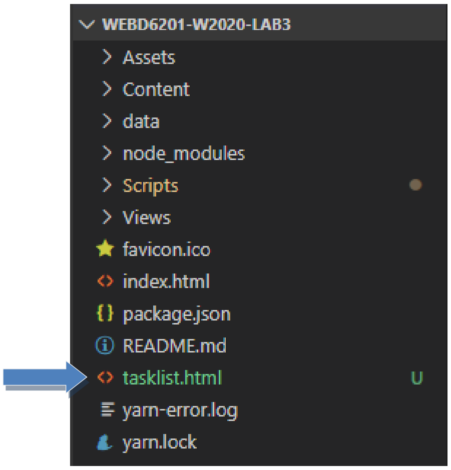
Fun with Ajax **Maximum Mark: 22**

Overview: This lab was made to practice and reinforce the use of Ajax JavaScript application programming interface (API).

Instructions :

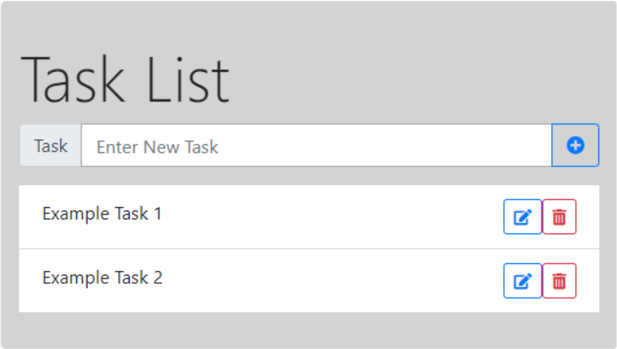
(2 Marks: Site Structure, 12 Marks: Functionality, 4 Marks: Internal Documentation, 4 Marks: Version Control)

1. **Project Setup (2 Marks Site Structure)**
    a. For the most part, your folder structure should follow the same structure shown in class. You may use WEBD6201-W2020-Lab3 GitHub repo as a starter. This will include a new page: tasklist.html page (1 Marks: Site Structure)



```
WEBD6201-W2020-LAB3
  Assets
  Content
  data
  node_modules
  Scripts
  Views
  favicon.ico
  index.html
  package.json
  README.md
  tasklist.html    U
  yarn-error.log
  yarn.lock
```

b. The **tasklist.html** has been created for you. It was the same page from Test 2. Use the following image as a guide to create the page (if needed): (1 Marks: Site Structure)



2. **Single Page Application with AJAX (12 Marks: Functionality)**
    a. **Tasklist page.** Convert the tasklist.html page to a "content" page. This means you will only keep the elements in the main element with an id of "`mainContent`" (Hint: see other examples in the Views/content folder of other content pages). There is no JavaScript required for this step. (2 Marks: Functionality)
    b. **Tasklist page.** Using **only JavaScript and jQuery** ensure to inject the tasklist content into the main element of the index.html template page when the user clicks the Task List button in the home content page (4 Marks: Functionality)
    c. **Tasklist page.** Using **only JavaScript and jQuery** make the page function properly in the Single Page Application (SPA) structure. The `DisplayTaskList` function has been included for your use. (Hint: see the `activateNavBar` function for examples on how to attach a callback to the `LoadPageContent` function) (6 Marks: Functionality).
3. Include **Internal Documentation** for your program **(4 Marks: Internal Documentation):**
    a. Ensure you include a comment header at the top of your app.js file that indicates: Your **Full Name**, **StudentID** and **Date Completed** (2 Marks: Internal Documentation).
    b. Ensure your program uses contextual variable names that help make your code human-readable (1 Marks: Internal Documentation).
    c. Ensure you include inline comments as required. As a rule the code should be descriptive but sometimes some information is required, especially before any function you include (1 Marks: Internal Documentation)

4. Share your files on **GitHub** to demonstrate Version Control Best Practices **(4 Marks: Version Control).**
    a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
    b. Your repository must include **commits** that demonstrates the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

Evaluation Criteria

Feature	Description	Marks
Site Structure	Your Project adheres to the site structure described (including Assets, Content and Scripts folders)	2
Functionality	The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program.	12
Internal Documentation	A comment header is present and includes the name of the student, studentID, and date completed.	4
Version Control	GitHub commit history demonstrating regular updates.	4
Total		**22**

SUBMITTING YOUR WORK

Your submission should include:

1. A zip archive of your project uploaded to DCConnect
2. A link to your complete project files on GitHub

Please zip all files in to a single project archive.

This assignment is weighted **5%** of your total mark for this course.

Late submissions:

- 20% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.