Cyber4All Test Assessment DevLog 2022

**Task 1:**

Install Express

* + **Status**:
  + Completed!
    - **Comments**
    - I also created this DevLog during this task. I never created a DevLog before so I did some research and looked up a few examples to come up with my own style of documenting this process. I am not sure if this is the best way to do it though.
    - I am not very familiar with the terminal but I was able to find a tutorial on JavaTpoint.com to help me complete this step. Using the terminal I also installed the modules:
      * Body-parser which helps with handling JSON, Raw, Text, and URL encoded from the data
      * Cookie-parser used to parse cookie header and populate req.cookies
      * Multer a node.js middleware for handling multipart/form-data

**Task 2:**

Running the API

* **Status:**
* Very challenging, completed to the best of my ability. I was not very familiar with the terminal and I may not have done exactly what I needed to do but I am not more comfortable with the terminal from attempting this task!
  + **Comments**
  + I have never used an API prior to this so a lot of research will be done
  + Command prompt npm install runs but the prompt npm start is not working and throws and error ‘Missing script’.
  + Did some research and StackOverflow.com suggested the problem might be I didn’t have a start script defined in my server.js file?
  + I didn’t know I needed a server.js file so I created one using <https://www.tutorialsteacher.com/nodejs/create-nodejs-web-server> and placed it in my folder on Vscode called backend.
  + Tried to access the new file created using command prompt node server.js and node c:/node/server.js and ran into an error “cannot find module…”
  + Researched the error message and used the command prompt “npm install --save server.js and I believed it installed it..not sure though
  + During my research on trying to figure out how to run server.js, I ran into the website <https://serverjs.io/> which gave me the command npm install server and I assume this worked.
  + Updated my server.js file with the code from this website and ran the command prompt node server.js again and the localhost:3000 is still not working.
  + I don’t want to spend too much time on one task so, I’m going to go ahead and try to start doing some coding
  + Still very confused so I used a youtube video and redownloaded express.
  + Created a file titled nodejs\_restapi that includes node\_modules, package-lock.json, and package.json. Through this, I was able to access the localhost:3000!
  + When I ping the server I get a GET error
  + Fixed the problem and got the localhost to say something
  + I will move on to the actual tasks now

**Task 3:**

Backend task 1

* **Status:**  Completed to the best of my ability!
  + **Comments:**
    - Started with refreshing my memory on what the GET method does exactly. The GET method is used for returning a specific element among all the other elements which are present in the map
    - Having trouble trying to get the HTTP status codes to respond to the server.
    - Nothing is showing up on the server
    - Did some research and realized the HTTP status code methods do not show up on the actual page they show up in the developer tools. The status codes are used to help the browser. The browser will use this information to help load and process the page.
    - My original method to this task using fetch was not working so I decided to do some research to find a different method.
    - I used the XHTML request object to achieve the status code response

**Task 4:**

Backend task 2

* **Status:** Completed to the best of my ability!
  + **Comments:** 
    - Did some research on how to implement a POST request
    - I added the code used for routing while doing backend task 1 to this file
    - I was having trouble using GitHub Desktop to try to push my files but I learned that Vscode has a simple way to push code to GitHub using the source control tab. I will use this for now on.

**Task 5:**

Backend task 3

* **Status:** Completed to the best of my ability!
  + **Comments:** 
    - I never used a PATCH method so I researched what PATCH was.
    - PATCH is similar to GET and PUSH and PUT. The PATCH method modifies existing resources. But this differs from PUT because the request body only needs to contain the specific changes to the resource.
    - Implementing the PATCH using a similar method for backend task 2
    - I realized while doing the PATCH method that there are multiple ways to do a response body for HTTP status codes. I went back to my previous files and fixed them to only using .status(getStatusCode(‘...’)). The other ways seemed redundant to me.

**Task 6:**

Backend task 4

* **Status:** Completed to the best of my ability!
  + **Comments:** 
    - I researched different ways to implement the delete method and it looks pretty simple
    - I decided to fix some syntactical errors from using fetch in this task as well as in the backend task 3 by using XHTML to implement these methods.

**Final thoughts:**

I have never done anything like this assessment before and that made me hesitant to start. So, I decided to treat this assessment like an experiment and do as much research as I could. This challenge ended up being something I enjoyed doing rather than something I dreaded. I appreciated the things I learned along the way. I am now more familiar with the terminal, javascript, and node.js.