**CPAN 212 – Modern Web Technologies – Lab 1**

For today’s Lab, we will be going over the basics of Node and things we covered in class. I do try to make labs longer so we can cover more material, my goal is to cover some practice while trying to build out an application.

Here is a list of what we will be doing:

* Part 1: Modules, NPM and the package.json
  + Installing a node module
  + Using that module
* Part 2: Creating a Node Server
  + We will talk about building out endpoints for our server
  + We will create webpages to send to a client
    - We will be using the built in modules from node to access and read html files
  + We will send the client json data
  + We will setup a page error for any endpoints that do not exist
  + We will use PostMan to handle different METHODS (GET, POST, PUT, DELETE)
    - Install postman onto our machines
    - Look into the status codes (200, 400, 500)
    - See how we receive results.
* Rubic
* References

**Part 1: Let’s get started with Modules:**

How to get started:

1. Create a new app folder: lab01
2. Open a terminal to that file
3. Enter this command: npm init -y
   1. This command will initialize a package.json file with default settings for our project
4. Next, we will install some packages
   1. Nodemon [nodemon - npm (npmjs.com)](https://www.npmjs.com/package/nodemon)
   2. Lodash [lodash - npm (npmjs.com)](https://www.npmjs.com/package/lodash)
5. We will write some code for it now
6. **Submission: The code file with lodash module**

|  |
| --- |
| **Task:**   1. **Create list of holiday objects with name and date [i.e: Christmas, Canada Day, New Years etc]** 2. **Iterate over array of holidays and print number of days until each holiday from today** 3. **Use Lodash library to output name and date of a random holiday** 4. **Use Lodash library to output indexes of “Christmas” and “Canada Day” holidays** |

**Part 2: Making a Node Server (For the Bookstore)**

1. Create a new Folder: bookstore
2. Create a server file called server.js and initialize a node server
3. For now, we will be working on making endpoints for our server
   1. In this, what webpages would we make?
   2. In class I showcases sending out webpages based on our endpoints, so what do we need
4. We will create webpages to send to a client
   1. For now, I want you to make some placeholder webpages. You may use ChatGPT for this. **Again, our focus for this lab is to do work on the server.**
   2. **Create a HTML folder called pages** and put each of the files we discussed about in class, made by you or online help into that folder
   3. Now, using the FS and Path modules, find, read, and send each of those files.
5. When someone enters an endpoint that doesn’t exist on our server what happens. Let’s make a catch all for all other routes.
6. **I want you to set up an HTML page to showcase “page not found”.**
7. Finally, we will write additional checks for the different methods (GET, POST)
   1. How do you run the POST requests?
   2. We will do this in class, it is something called: PostMan
   3. I will showcase it to you in class, but we will use it next week

You can use the same GitHub you should have set up last week.

**Rubric:**

|  |  |
| --- | --- |
| Weightage | Rubric |
| 40 Marks:  10 marks 10 marks  10 marks  10 marks | Part 1   * Creating dictionary of holidays * Iterating over holidays array * Outputting holiday name and number of days when iterating * Using lodash functions |
| 60 Marks  2.5 marks  2.5 marks  2.5 marks  5 marks  10 marks  5 marks  10 marks  2.5 marks  20 marks | Part 2   * Created your app folder * Created your pages subfolder for HTML code * Created your server file * Setting up the http server * Creating and using your routes * Using methods * Using fs and path * 404 error page handler * Writing your unique routes |
| 100 Marks | Total |

Submission:

File structure:

* lab01 [Folder]
  + part1\_modules [Folder]
    - lab1.js
  + part2\_bookstore [Folder]
    - pages [Folder]
      * html files in here
    - server.js

**References**

* Additional information or References:  
  Status Codes: [List of HTTP status codes - Wikipedia](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)
* Lodash: [Lodash Tutorial - GeeksforGeeks](https://www.geeksforgeeks.org/lodash/)
* Nodemon: [nodemon - npm (npmjs.com)](https://www.npmjs.com/package/nodemon)
* Fs module: [Node.js File System Module (w3schools.com)](https://www.w3schools.com/nodejs/nodejs_filesystem.asp)
* Making a server: [Node.js HTTP Module (w3schools.com)](https://www.w3schools.com/nodejs/nodejs_http.asp)