COMP 1536 – Assignment 4

Due: The week of November 8th, before the beginning of your scheduled lab class, for marks! This assignment is out of **20** – heavier because you are doing more work!

This assignment must be done individually.

Objective

To use JavaScript (JS) for creating client-side code that retrieves content from the server – the most common use of JS for web development.

Part A

For this lab, you are going to get acquainted with Node.js, or simply known as Node. Node is a server environment that allows you to run JS on the server-side. This can streamline your development process since you are using the same language on both the client as well as the server. As well, Node.js comes with a plethora of modules for various features. Windows users, use this link to install Node.js:

https://nodesource.com/blog/installing-nodejs-tutorial-windows/

Mac OS users, follow this link to install Node.js:

https://treehouse.github.io/installation-guides/mac/node-mac.html

You might be asking why you need to already start using Node this early. After all, you haven't even completed one JS assignment and we are already moving to the server side. The most important reason is that web browsers have this policy of not allowing Cross Site Scripting (XSS). If you write some JS code for the HTML web document and attempt to refer to files (e.g., an image, more JS files, another HTML document, etc.), the browser will block you from doing this. This is good since it protects our computers from malicious scripts that may attempt to scan through our file system and perform actions that we are not aware of or want.

So, in order to successfully retrieve files programmatically with JS, we need to be running from a server environment. This requires us to be running Node – which acts as a web server for us and allows our JS to refer to the files that are on the server.

❗ **Note:** when you load your HTML documents in the web browser, do not refer to them from the file system directly. So for example, this would be the incorrect way to refer to the file:

file:///C:/apps/myapps/node-template/public/html/index.html

While this will load the file for you, this file will be considered locally loaded and therefore will block your JS from accessing files, such as images, from the file system. Refer to your HTML document from the server, such as this:

http://localhost:8000/

❗ **Note:** Code editors such as Microsoft Visual Studio Code, the late Adobe Brackets.io, and the rest do have launcher plugins. It is best to avoid using these and get yourselves acquainted with the console/command line/terminal so as to avoid odd behaviors and tool misconfigurations.

Once you've set up Node, you'll be able to run your apps in the console (Windows) or terminal (Mac OS).

Part B

Acquaint yourself with the Node template that is provided. It has several files:

- index.js: the server-side code that Node runs as an application
- index.html: the HTML document that acts as the main page for your application, and is sent to the web browser by index.js
- style.css: the style for your index.html HTML document
- client.js: the JS file that acts as the programmatic part of the client-side of your application, and that also contacts the server asking for more content (the newsfeed.xml document – for now)
- newsfeed.xml: a snippet of HTML code, in this case an unordered list

This basic template has very little code in it but so much is going on within it. It demonstrates the basics of what happens with a client-server application which is depicted in figure 1.

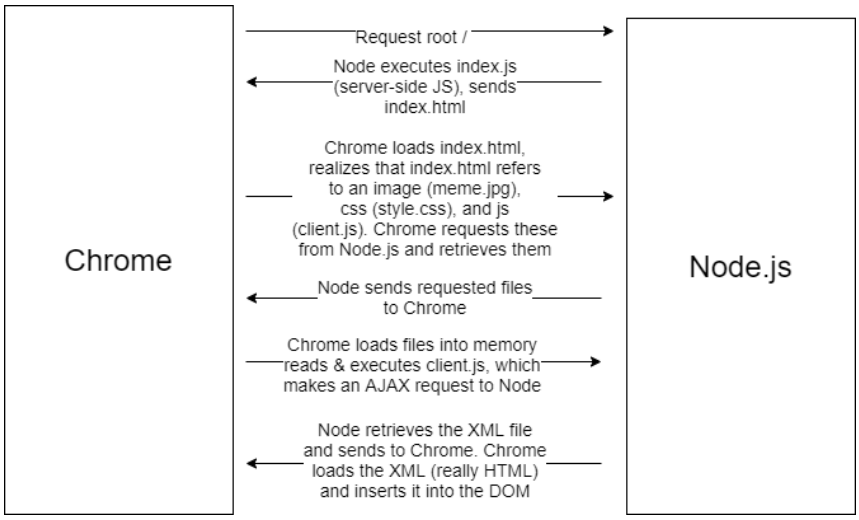

Figure 1 - Basic Node Template, Client-Server Interaction

A lot just happened there! And this is just a basic template! But it demonstrates the goings on within a web app. Study this template and understand this process. Ask questions in lecture and lab because it is imperative for you to know how this process works.

Part C

Your job is to use this template and create a very simple request-response, “hello world” web app. The content that you want to display can be anything you would like, but here are some suggestions:

- Recipes for cooking
- Hockey team scores
- Movie listings for a theatre
- Courses for a program (e.g., CST)

What you pick is up to you, but the following requirements must be met in order to receive full marks:

- Return 10 or more records of data (as an HTML table in an XML file) where each record has at least five attributes (i.e., five columns)
- The data must come from an XML file delivered from the server-side (no using FireBase!)
- Contain CSS styling that creates a fluid/liquid (doesn’t have to be responsive – for this assignment)
- The data must be retrieved using two different ways:
    - On page load and populates one part of your layout
    - When the user clicks on an element the same list populates another part of your layout

Some questions answered:

1. Can I have two lists instead of the same list? Sure. But you’ll need two separate files and a slight change to the index.js file on the server
2. When the user clicks on an element … don’t you mean an input element of type button or a button element? Can be, but doesn’t have to be. JS will allow you to capture a button click on any element you choose
3. Does this have to be responsive? Not for this assignment. For now get used to using JS on the client-side to request content from the server
4. Why do I have to load the same content twice and put in two different spot on the page? Because I want you to get experience loading the content when the page is loaded, but also get the experience handling user interaction which is also vitally important to web development

Requirements

The following expectations are on this assignment:

- You have external CSS, external JS, and at least one image referenced in your layout
- Your JS contacts the server on page load and populates a portion of the page with the records
- Your JS allows the user to click/tap on an element which will also contact the server and retrieve the records
- In both cases you are retrieving 10 or more records in the form of a table and that the table has at least five attributes (i.e., columns)

Once you have completed, ensure that your document is valid. It is suggested that you use the HTML 5 validator online at:

https://html5.validator.nu/

To ensure that you don't lose marks. Also use the CSS validator in Visual Studio Code.

Submission

Create a zip archive of any and all text files that are part of your assignment submission. Your HTML files will have the extension ".html", your CSS files will have the extension ".css", and your JavaScript files will have the extension ".js". If you have images, be sure to include them in this archive that you create.

Once you create your zip archive file, rename it to follow the format SurnameFirstnameCOMP1537Assignment4. My zip file would be FergusonArronCOMP1537Assignment4, for instance.

Finally, submit it to the learning hub in the folder labeled "assignment 4".