WEB222 Assignment 3

Submission Deadline:

Assessment Weight:
5% of your final course Grade

Objective:
Practice writing HTML Containers / Media Elements, Updating the DOM with data

Specification:
Update HTML & JavaScript code for each of the following pages according to the instructions outlined below. To begin, download the assignment3.zip file containing all of the files required for Assignment 3 from blackboard.

Uncompress the zip files somewhere on your local machine. When you're ready to begin editing the files, open the uncompressed folder (assignment3) in Visual Studio Code (https://code.visualstudio.com) using "Open Folder". You may test your html files in any modern browser (Chrome, Firefox, Safari, Internet Explorer, etc).

"Assignment 3 Home" - index.html:

Update the "Assignment 3 Home" page with the following:
1. Add a **professional greeting** to the visitor, i.e.: "Welcome to my website, I will be demonstrating HTML5 principles and techniques, DOM manipulation"… and so on.

2. Add a relevant **header** as a title for the next section (step 3)

3. Add a **short paragraph** that introduces this website, i.e.: "This site contains 8 pages, including: Home, HTML Text, HTML Lists"… and so on. **NOTE:** This paragraph **must include all relevant links** to the **8 pages .**

4. Add a relevant **header** as a title for the next section (step 5)

5. Add a **short paragraph** introducing HTML 5, i.e.: "This site utilizes HTML5: a markup language used for structuring and presenting content on the World Wide Web"… and so on.

"HTML Lists" - list.html:

Update the "HTML Lists" page with the following:
1. Inside list.html, create any **nested list** with meaningful items using countries, provinces and cities.
• The nested list should contain at least **one ordered list** and at least **one unordered list.**

2. Update the list.html page to include two containers (i.e.: <div>) with unique id values.

3. Add a relevant **header** as a title for the next section (step 4)

4. Use the **fruits** array inside the **lists.js** file (js/list.js) to write (using JavaScript) an ordered list containing all the fruit inside the array, (ie:"Apples","Oranges","Pears","Grapes","Pineapples","Mangos") to one of your containers in the list.html page. Be sure to write to your container (using JavaScript) **after** the window has completely loaded. The ordered list should look like the image below when complete:

1. Apples
2. Oranges
3. Pears
4. Grapes
5. Pineaples
6. Mangos

5. Add a relevant **header** as a title for the next section (step 6)

6. Use the **directory** array inside the **list.js** file (js/list.js) to write (using JavaScript) a nested unordered list containing all the files and directories (with their files) inside the array to one of your containers in the list.html page.

You will notice that files have the type "**file**" and consist of a "**name**" property, whereas directories have the type "**directory**" and consist of a "**name**" property in addition to an **array of files**. Use these properties to correctly construct your nested unordered list (using JavaScript).

Be sure to write to your container **after** the window has completely loaded (using JavaScript). The nested unordered list should look like the image below when complete:

- file1.txt
- file2.txt
- HTML Files
    - file1.html
    - file2.html
- file3.txt
- JavaScript Files
    - file1.js
    - file2.js
    - file3.js

"HTML Tables" - table.html:

Update the "HTML Tables" page with the following:
1. Inside table.html, create the following two tables, each with a relevant **header** as a title:

    • The **1st table** should be composed of the elements: **<table style="border:1px solid;">**, **<tr>**, **<th>**, **<td>** and **<caption>** with meaningful table contents, i.e.: a table of your hobbies, description and time spent etc.

    • The **2nd table** should be composed of the elements: **<table style="border:1px solid;">**, **<tr>**, **<th>**, **<td>**, **<thead>**, **<tfoot>** and **<tbody>** with meaningful table contents, i.e.: a table of your favourite personalities, authors, descriptions etc.

2. Update the table.html page to include an additional container (i.e.: <div>) with a unique id value.

3. Add a relevant **header** as a title for the next section (step 4)

4. Use the **users** array inside the **table.js** file (js/table.js) to write a complete **3rd table** containing all the users inside the array (and an appropriate header row), to your container in the table.html page.

You will notice that users consist of the properties: **first_name**, **last_name**, **age**, and **email**. Use these properties to correctly construct your table (using JavaScript) with the following headers: **First Name**, **Last Name**, **Age** and **Email**. You must also ensure

that all email addresses are rendered as a valid "mailto" link and will open as a new message in the user's default mail client when clicked.

Be sure to write to your container **after** the window has completely loaded. The table should look like the image below when complete:

First Name	Last Name	Age	Email
Kaitlin	Burns	23	kburns99753@usermail.com
Joshua	Feir	31	josh319726@usermail.com
Stephen	Shaw	28	steve.shaw47628@usermail.com
Timothy	McAlpine	37	Timbo72469@usermail.com
Sarah	Connor	19	SarahC6320@usermail.com

"HTML Images" - image.html:

Update the "HTML Images" page with the following:

1. Find the given image with the file named **ict.png** under the "**img**" sub-folder included in the zip file. Show the image in your web page using a relative path. Ensure that when the image is not available, the text "**ICT School, Seneca College**" is visible on the page and make the image a **hyperlink to the ICT School website**.

2. Update the image.html page to include an additional container (i.e.: <div>) with a unique id value.

3. Add a relevant **header** as a title for the next section (step 4)

4. Use the **images** array inside the **image.js** file (js/image.js) to write 5 new **<figure>** elements to your container in the image.html page (using JavaScript). You will notice that each image in the array has the following properties: **caption**, **alt** and **url**. Use these properties to correctly construct (using JavaScript) your **<figure>** elements such that each element contains a valid **image** (using **url** and **alt**) as well as a **caption** underneath the image with the message contained within the **caption** property

Be sure to write to your container **after** the window has completely loaded (using JavaScript). The **<figure>** elements should look like the image below when complete (only the first 2 shown):

Red Slate Mountain

Indonesian Jungle

"HTML5 Audio" - audio.html:

Update the "HTML5 Audio" page with the following:

1. Update the audio.html page to include an additional container (i.e.: <div>) with a unique id value.

2. Use the **audio** object inside the **audio.js** file (js/audio.js) to render an audio player in your container within the audio.html page.

You will notice that the audio object consists of the properties: **controls** and **source**. Use these properties to correctly construct (using JavaScript) your audio player with (or without) controls, and the correct source options.

Be sure to write to your container **after** the window has completely loaded.

"HTML5 Video" - video.html:

Update the "HTML5 Video" page with the following:

1. Update the video.html page to include an additional container (i.e.: <div>) with a unique id value.

2. Use the **video** object inside the **video.js** file (js/video.js) to render a video player in your container within the video.html page.

You will notice that the video object consists of the properties: **controls**, **width**, **height** and **source**. Use these properties to correctly construct (using JavaScript) your video player with (or without) controls, in the correct dimensions and using the correct source options.

Be sure to write to your container **after** the window has completely loaded.

"Seneca College" - seneca.html:

Update the "Seneca College" page with the following:

Find an image representing Seneca from the **http://www.senecacollege.ca** website. Download and include in the **img** folder. Use relative path to display the image in this web page.

In case, you are not downloading the image, you may display without downloading this image file (i.e.: use an absolute link to the file) – **HINT:** Right-click on the image and choose "**inspect**" – this will take you directly to the html responsible for rendering this image and you can see the relative path to its location on the server. Generate your **absolute path** using this as a starting point if are not downloading the image.

Ensure that when the image is not available, the text "**Seneca College**" is visible on the page.

Lastly, define three clickable areas on the image (three equal rectangle shapes from left to right is preferred) and make/map the areas hyperlinked to Seneca's home website, ICT website and library website respectively.

"Honesty Statement" - honesty.html:

Update the "Honesty Statement" page with the following:

1. Add date and your name in the provided places to complete the academic honesty declaration.

Other Requirements

• All tags/attributes must be in **lower case.**

• Make sure to **update the title** with relevant text on **each of the pages. Each page footer must have your full name as Student name (zero will be assigned if you do not write your name)**

• All of your html files **MUST not contain any errors** when tested using the W3C Markup Validation Service: https://validator.w3.org/

Assignment Submission:

• Zip **all of your files** (i.e., your **assignment3** folder) as **assignment3.zip**

• Upload the zip file to **My.Seneca** under **Assignments -> Assignment 3** (same submission procedure as Assignments 1 & 2)

• **NOTE**: Your **HTML must not contain any errors** when validated (https://validator.w3.org/)

Important Note:

•   **NO LATE SUBMISSIONS** for assignments. Late assignment submissions will not be accepted and will receive a **grade of zero (0)**.

• After the end (11:00PM) of the due date, the assignment submission link on My.Seneca will no longer be available.