**Unit 5: JavaScript Use and Workflow Design**

**Header Loading**

This will be a way of loading the header and the footer into every page without having to duplicate the code within every html page. To achieve this, we just need an element in every page with a header and footer id. Then use JavaScript to find the element and add the html code into it.

Pseudo Code:

* Find the Header element using the id for the div
* Create a string containing all the tags to be added into the header
* Add the string into the inner HTML of the div
* Repeat the above 3 steps for the Footer

**Navigation Bar**

All personas of the website wish for it to be easy to navigate. For this reason, I want to add a navbar which will be fixed on the page and can always be accessed no matter how far down you scroll. It also removes clutter from the page so everything on it is easier to find.

Pseudo Code:

* (HTML) Create a div for the nav bar with a width of 0px so it is invisible
* Find the element using the id for the navigation bar
* Append all the links into unordered list which will be added to the html of the nav bar div
* Create a function which opens the div by setting the increasing the width of the div
* Create a function to close the div by setting the width back to 0
* Go back and add the functions utilizing onclick functionality

**Fabrics Load in with Loop**

            Due to the constant changing of the fabrics, an important feature of the website is to easily be able to swap out what is being displayed. The best way to do this instead of always having to update all the html code is to just store a list of every item to be displayed. It can then iterate over the list appending each item into the html. This way if any item is to be added or removed it only has to be added or removed from the list and the function will handle all of rest.

Pseudo Code:

* Get the element for the body of the page using the id
* Create an array for the items. Such that each entry contains the name, display image, and pdf in that order and every 3rd entry in the array is a new item
* Add the table opener to the inner HTML and start the first row
* Loop through the length of the items, increasing the loop by 3 for each iteration
  + If the iteration is a multiple of 3:
    - end the current row and start a new one
  + Add the item as its own cell in the table
* End off the last row and table

**Full Validation of New Vendor**

In the last unit I added a validation to the email and phone number fields in the new vendor form. For the form to perform its function in further units it must validate all required field to ensure that they are not empty. As well at store the information from each field being inputted. A regular expression will be used on the email to ensure it is in proper format.

Pseudo Code:

* Get the element of the form and the textarea box for the service entry
* Create an event listener for the form
* Prevent the page from reloading on a submit
* Store the entries of the form in a dictionary and get the value of the textarea
* Check for any blank entries that were required (not all vendors have a website so it can be left blank)
* Run the phone number and email validation from the last unit
* Send off an alert if anything does not pass, else send a successful message