**Learning Diaries**

**Unit 1:**

In this unit, I created documentation outlining the website that I will be creating. This documentation included the theme for the site, a site map showing how each of the pages will be linked together, mock ups for each page, as well as personas and scenarios of people using the website. As the company sells only business to business, I felt a simple easy to use and navigate website would be much more effective than a flashy one as many of the customers already know what they want, they just need to find it. For the site map I made the decision of linking every page to one another, other than the registering a vendor page which can only be accessed from the vendor page. I made this decision as most people accessing the website are not there to use that functionality and I didn’t want it to distract customers from what they are looking for.

**Unit 2:**

In this unit, I created all the base HTML code for my website. I have worked with HTML before, and my main goal was to ensure the site would be easy to navigate and be setup with the proper classes and divs such that it would be much easier to implement the css and JavaScript in upcoming units. For this unit I only created 3 out of 5 of the pages that will be on the final website, and I intend to use JavaScript to fill out the other 2 pages so they would be relatively empty outside of the header and footers which are displayed on the other pages. But I feel the other 3 that I created setup the templates that are in my page mock-ups in my site design which I created for Unit 1. It was a small learning curve for me to work with html without css as usually when I am creating something I do them both at the same time. Overall, I feel I met the learning goals for this section implementing many different types of tags into my pages.

**Unit 3:**

In this unit I improved on all the HTML code from the website by styling it with css. Similar to Unit 2 I have had lots of experience with css in the past, so it was relatively straight forward, and I just tried my best to stick within the theme for the site. I tried to stick to a more neutral, yet classy color scheme. This was to keep within the goal of not making the website too flash, as the main goal is ease of access for anyone trying to use it, and I don’t want them to get distracted by other things while navigating through the website. The homepage was made to flow between the different products that are offered with each section guiding you onto the next. I broke apart the css into different files to help improve the maintainability of the code but ended up linking every page to index.css anyway as it had base styling for every page. If I were to do it again, I would split that file into 2 to have styling specific to the homepage and styles for every page such that there is less unused code being loaded into each page.

**Unit 4:**

In this unit I added the first bit of JavaScript code. The code that I used was for validating an email and phone number. I implemented it as part of my registering a vendor form to ensure that the information that they inputted is correct. This functionality related back to the scenario of a contractor trying to display their services. While simultaneously helping to reduce fake entries by sending an alert to the screen if it is either an invalid phone number or an invalid email. I found this section to relate very closely to what I experience as a developer as you rarely every write completely new code. Finding something that matches what you are trying to accomplish then modifying it to my specific needs help prepare much more for a professional work environment. The code which I took is stored in the takenCode.js and was originally found at //https://stackoverflow.com/questions/19646276/javascript-email-and-phone-validation.

**Unit 5:**

 In this unit, I started on coding JavaScript functions without the use of any online code, or libraries. This is a little different than what I am used to as most of the code in which I write utilizes some libraries or frameworks. I started off by writing the pseudo-code for all my functions I intended to create so that I would have a good understanding of what to implement next as I developed the code. One of the challenges I ran into was around the form validation and storing the information. I wanted to use a textarea field instead of a simple input for the services field in the form, as I wanted to allow for vendors to write all of the services that they provide and not be limited to a small input field. Unfortunately, that means the textarea was not actually part of the form being submitted, so when I got the information from the form it was missing. But I was able to work around this by getting the field by its element then individually reading the value of its field when the submit event is triggered. Many of the other changes I did in this unit such as the loading the header and looping through the fabrics to render them where to improve the maintainability of the code and allow for easy changes should they need to be made in the future.

**Unit 6:**

In this unit I implemented 2 different libraries in order to increase the ease of access of my website. The first library (PushbarJS) was used to create a better navigation bar for the website. By blurring the rest of the page, it helps to draw attention to the open menu, as well as being easier to close than the original navigation bar which was implemented in Unit 5. The second library (JQuery) was used to implement a tabs menu in the locations page. It adds more dynamic and liveliness to the website allowing users to easily switch between the different locations. I have never worked with either of these two libraries before so there was a learning curve in order to add them both to my website. It is great to expand my portfolio of libraries that I have worked with and can easily bring out if I ever need them in the future.

**Unit 7:**

In this unit I added 2 different API’s into my site. The first one I added is an email service which allows you to send emails from my site. I create a template on the email.js website and define variables to be passed in. Then call the API from my site passing through the required variables. This is a key feature of the site as it allows users to register themselves as vendors like in the defined scenarios. Users cannot be allowed to instantly add themselves as vendors as anyone could go in putting in whatever they want including profanity or competitors which should not be posted on a professional website. For this reason, it sends an email to the website administrator with all the information requested to be added so it can be reviewed then added if accepted. This was an API I have never worked with before so there was a learning curve in setting up the implementation and debugging it to ensure emails were being sent through properly, and it was nice to learn. I now have an available way to send emails from any website. The second API I used was google maps which I have worked with before and added it to the locations page so that any user/persona wishing to find a warehouse can easily view a map without having to use a separate location. If I were to do this again, I wish I could add additional API’s to the site, but I would need to define more functionality that would be required for the site.