Homework 6 Reflection

Github Pages Link: https://radibi3000.github.io/radibi.github.io/

Github: https://github.com/radibi3000/radibi.github.io

General Reflection:

When I started writing this assignment, I felt strongly that I did not have the necessary tools to complete it. Although I felt that I was up to date with mark up and style sheet languages (HTML and CSS) I didn't think I understood the logic necessary to complete the javascript portion of the homework. I was able to complete assignment 6A but 6B required a level of knowledge that I was unsure if I truly had. As I completed the assignment I felt more confident with my ability and through the use of internet searches and console activity I was able to accomplish much more than I thought I would. There are a few specific bugs I ran into while working on the assignment and working through them helped me better understand Javascript as a whole.

One major bug was mistaking ***getElementbyId*** and ***getElementsbyclassname*** as functions to get ID and classes. I tried many times to call an entire class with getElementsbyclassname and it would never work, upon research it became clear that getElementsbyclassname cannot be used to get a specific class element even if there is only one existing class. Because getElementsbyclassname creates an array with all the class names whereas getElementbyId gets the specific ID. When using javascript this is important especially in regards to creating and using IDs is the easiest way to do this. This was a major bug I came across.

Another issue in my programming experience was at a high level figuring out what my bugs were. There were many times specifically when trying to create a counter that augmented my pillow price, where there were errors that I couldn't see. By using the console and exploring the code in the inspection windo, along with using console.log I was able to find out where I made my errors. For example when trying to call the quantity counter value, I forgot to add the innerHTML property to my line which would provide the HTML content to my javascript. Through my activity with the Console I was able to figure this out.

Overall, it was through research and using the console where I was able to work through the various bugs and issues I had, making this assignment not only manageable but fun.
Throughout this program I learned various different programming concepts that not only helped me better understand programming in general but allowed me to complete this assignment thoroughly and effectively.

1. **Local Storage:** The idea of local storage was unclear to me at first but as I began designing my cart it became apparent that I needed to hold the data from the product object that the user inputed when selecting the cart to then present it on the shopping cart page. If I were to do this without localstorage upon refreshing the page the shopping cart information would be lost. For this reason I utilized local storage to rechieve my product object along with removing it to provide the user with options to add and delete items on their cart.

```
console.log("selecting item", size, color)
let pillows = new cartItem(name, quantity, color, size, price);
cart.push(pillows);
var jsonDetails = JSON.stringify(cart);
localStorage.setItem("MyDetails", jsonDetails);
```

Example of setting my cart to local storage. A necessary step in setting my object to local storage, after this step it was clear in my console that the object was being retrieved.

2. Getting Elements through GetElementbyID: This was arguably thee most important aspect of the entire programming experience, as someone with experience in Python the most unique aspect of Javascript was its ability to directly connect with an HTML file which it does so seemlessly. By mastering getElementbyID, I felt I was able to easily understand the relationship between my HTML code, specifically the IDs I laid out and directly manipulated in my javascript file. For example, to simplify my code instead of calling each ID each time I used it, I set global color and size variables through getElementbyID and from this was able to easily retrieve those IDs and manipulate them, this thoroughly simplifed my program and is something I will utilize through my programming experience to come.

```
// Global Variables.
let quantity = 1;
let color = 0;
let red = document.getElementById('red5')
let blue = document.getElementById('blue5')
let green = document.getElementById('green5')
let small = document.getElementById('small5')
let medium = document.getElementById('medium5')
let large = document.getElementById('large5')
```

An example from my programming of setting my global color and size variables to simplify creating a click feature

3. Connecting HTML to Javascript using onClick(): Another essential topic, I gained from my personal research along with discussion and lessons in the lab was the idea of an onClick() in essence this is a function that executes a function in javascript when the button is clicked in HTML. This is an incredibly powerful tool, one that I used to help create many of my javascript features on the product page. In fact without this feature, I would have been unable to adjust the *color* of the color and size boxes on the product page to provide feedback to the user when they click on a specific button.

```
<div class="sizeandcolor">
  <p id="selectasize5"><strong>Select a size</strong></p>
  <div class="newColor">
    <p onclick="toNewColor(this.id)" id="red5">Red</p>
    <p onclick="toNewColor(this.id)" id="blue5">Blue</p>
    <p onclick="toNewColor(this.id)" id="green5">Green</p>
  </div>
  <p id="selectacolor5"><strong>Select a color</strong></p>
  <div class="newSize">
    <p onclick="toNewSize(this.id)" id="small5">Small</p>
    <p onclick="toNewSize(this.id)" id="medium5">Medium</p>
    <p onclick="toNewSize(this.id)" id="large5">Large</p>
  </div>
</div>
```

```
// Below functions, hardcode help with selecting items, on the purchase page. Ch
💡 that only one color/size can be selected at once.
function toNewColor(goalColorID) {
    red.style.backgroundColor = "#FFFFFF";
    blue.style.backgroundColor = "#FFFFFF";
    green.style.backgroundColor = "#FFFFFF";
    document.getElementById(goalColorID).style.backgroundColor = "#fad980";
}

function toNewSize(goalSizeID) {
    small.style.backgroundColor = "#FFFFFF";
    medium.style.backgroundColor = "#FFFFFF";
    large.style.backgroundColor = "#FFFFFF";
    document.getElementById(goalSizeID).style.backgroundColor = "#fad980"
}
```

A screenshot from my HTML file an example of onClick functions (Left Image) which are then referenced in my javascript file (Right Image)

4. Augmenting HTML IDs through Javascript functions with addEventListener(): Is another Javascript method I learned from this program, whats interesting about this method is how it acts similarly to the onClick() function however it does not require setting a function into an element of the HTML Code, all that is required with an addEventListener is to specify the ID and then add the event listener with a click function and it will do the same thing the onClick function would without having to augment the HTML code in anyway.

```
Populated your cart

st addToCart = document.getElementById('AddtoCart');

ToCart.addEventListener('click', () => {
 document.getElementById("Notification").style.opacity = "1";

 if (quantity >= 1) {
     document.getElementById('Notification').innerHTML = `${quantity}`;
 }
```

In this function I use an eventlistener to notify my HTML that when I click on the notificaiton ID a hidden 1 value
appears. This is done in my JS code without editing my HTML file, much more efficient.

5. Specifically retrieving the HTML value within Javascript using innerHTML: Finally the last programming concept I felt was worth discussing was utilizing innerHTML, as mentioned in my reflection on the bugs I ran into, when I woud call certain IDs from my HTML I was originally unclear on how I could retrieve the specific string value of the ID. With the innerHTML tool, I was able to take various IDs and parse the string value which I tehn augmented.

The example here demonstrates how I was able to gain the quantity element using innerHTML, in essense I found the ID and used innerHTML to get the string value, I then removed the aspects of the string I didn't want and made it an integer, I was then able to use this quantity in my product object, an essential part of my program.

Conclusion: Overall, these higher level design skills broken up into important concepts were essential in improving my javascript skillset and I am excited to engage further with this javascript in my future career.