Reflection: Challenges

- One of the big challenges I encountered was to figure out how to delete items from my product array. Using delete[i] on the array would not work properly, leaving an empty space in the array instead of removing items from that specific index and readjusting the array's length. After further research, I realized that I could use Splice, a method that takes in the index of the item to be removed and the number of items to be removed from the array.

- Another issue that I encountered was that the Item page(Pumpkin Spice), would show errors and not load properly after I added a link to the Javascript file I made. When I looked up if there was a way to avoid this while still using Javascript functionalities, I realized that adding the Javascript link in the body section instead of the head section would solve this problem. This is because the Javascript code alters the HTML code, so when it is loaded before any of the HTML elements are, it gives errors. Thus, adding at the end of the body section after all the HTML elements avoids this.

- Lastly, I was having trouble retrieving the value of the type or name of the roll in Javascript(ex: Pumpkin Spice) from HTML because I was using getElementById and .value to get this. But with more exploration, I figured out that .value property only applies to form elements, or inputs, and not divs. The simplest way to get the contents of any div element is to use .innerHTML.

Programming Concepts

- **localStorage:** The localStorage property was what I used to store the items in the order created by the customer. This property uses a name/ value pair to store values in the web browser. So, for example, before going to the cart page, the code runs the function goToCartPage, where it uses ‘localStorage.setItem('order', JSON.stringify(productArr))’ to store the value of productArr(the customer’s order) locally.

- **Conditional statements:** Conditional statements allow me to control my code based on certain different circumstances and variables. For example, to be able to control the pricing of the bun orders, I used an if/else statement in my Javascript function. These statements use the quantity of rolls chosen by the customer and depending on them, return a total price value by multiplying the quantity with the base price.

  ```
  example : if (amountValue == 1) {
                  productSubtotal == 4.99 ;
  }
  ```

- **Loops:** Loops allow me to iterate through data structures easily and efficiently. In this case, for example, I used a for loop to loop through the different elements of the productArr, the array with all the properties(like quantity, flavor etc) of my product and access each of them.

  ```
  Example: for(var i = 0; i < productArr2.length; i++) {
               }
  ```

  (The contents inside the for loop aren’t shown here)

- **Onload**: The onload event executes Javascript once a page loads all the content. Using this event on my Cart page allowed me to write, style and display HTML elements that are meant to change dynamically depending on the user’s selection. I used it in the body of the Cart page so as to be able to display the user’s order on the page dynamically.
  Example: <body onload="cartLoaded();">

- **On change:** The onchange event executes Javascript when a user changes the selected option of a <select> element. Using this allowed me to dynamically update the total price of the user’s order depending on the quantity of rolls they selected from the <select> options for quantity.
  Example: <input type="radio" value="1" name="quant" onchange="updateProductSubtotal()" checked="">