Assignment 6B

Reflection

One of the challenges that I faced during the implementation process was that I could not figure out which item to delete at first, because when I added the delete buttons to the cart I forgot to put in unique identifiers to the delete button so it only removes items from the first one in the array. I resolved it by passing a unique index for each item and pass it on to the parameter.

Programming Concepts

1. Event listeners

   I learned how to incorporate event listeners inside HTML to trigger events written in javascript. This is helpful for building buttons. Here's an example of an event listener in the HTML file of my website:

   ```
   <button id="order" onClick="order()">Order</button>
   ```

   Which triggers the order() function that is written in javascript.

2. Local Storage

   Local storage is helpful when implementing the cart and transferring that information to the following pages. I use this to parse and pass on information usually in the beginning and end of my functions:

   ```
   function deleteItem(index){
       //remove item from cart
       var cart = JSON.parse(localStorage.getItem("shoppingCart"));
       cart.splice(i-1,1);
       localStorage.setItem("shoppingCart",JSON.stringify(cart));     //repopulate page
       getCart();
   ```

3. HTML DOM nodes

I used DOM nodes to populate the cart page with the information I stored. To successfully add a complex div, I layed out the structure firs to figure out the correct parent-child relationships.

To create the nodes in the DOM:

```
var item = document.createElement("div");
        item.setAttribute("class","product");
        item.setAttribute("id","item"+i);
```

Then, I attached them to the parent div:

```
left.appendChild(item);
```

4. Objects

Object was helpful because I have different buns in the cart with specific attributes to the bun, and object is a great way to organize the content. First, I set up the attibutes to the bun:

```
// bun object
function Bun(flavor,glazing,quantity){
    this.flavor = flavor;
    this.glazing = glazing;
    this.quantity = quantity;
}
```

5. For loops

To access every single bun in added to the cart, I had to use for loops to loop through the cart array:

```
var cart=[];
for (i=0; i<cart.length; i++){
```

This allows my cart to function properly no matter how many items the user choose to add, and most of my DOM node code is built inside the loop.