Reflection - Assignment 6B

Ije Okafor

Link to live version of the site hosted on GitHub pages:

- https://okafor3.github.io/homework_6/HomePage.html

Link to the repository where the code is hosted:

- https://github.com/okafor3/homework_6

What challenges or bugs did you encounter? How did you overcome these challenges?

Initially, I struggled with the use of local storage and transferring data from one page to another. For example, having the product page information update based on the menu item that was selected, and having the shopping cart update based on the quantities of that item that were selected. I overcame these challenges by using online resources (such as W3Schools), reviewing past labs, and emailing my TA / attending office hours. It also helped to print statements (via console.log) to gain a better understanding of where my issues / errors lied. The use of local storage was especially helpful in situations where I added multiples quantities of the same item. For example, if I added one order of 3 Blackberry buns and another order of 12 Blackberry buns, local storage saved the initial order and updated the bun total to 15 (instead of overwriting past orders). Local storage is especially important to use when creating a shopping cart, whose purpose is to "hold" items for user until the user is ready to make a purchase. If a user were to exit a site and return later to try to access their shopping cart, local storage ensures that the items they previously selected remain in their cart.

What programming concepts did you learn as a part of the assignment? Illustrate at least 5 concepts with an example.

(1) **If / Else Statement:** this statement is used to perform different actions based on different conditions. It executes a block of code if a condition is true, and a different block of code if the condition is false. In my code, if an item is already inside the cart, then its current quantity is be added on to. If not (else), then the item's initial quantity is added.

```
if (productname in cart) {
   cart[productname] += parseInt(quantity);
 } else {
   cart[productname] = parseInt(quantity);
 }
```

(2) **For Loop:** facilitates the execution of a set of instructions while some condition is true. My for loop says that for all the products that are found in the cart, the cart should display the product's name, quantity, and price.

```
for (var productid in cart) {
   if (cart.hasOwnProperty(productid)) {
```

```
  price = cart[productid] * 6.25;
  cartinfo += productid + " : " +cart[productid] + "<br/>";
```

(3) Local Storage: a type of web storage that allows JavaScript sites and apps to store and access data. This type of storage has no expiration date. I used local storage to ensure that previously selected menu items remained in the cart even as new items were added.

```
function loadCart() {

 var cart = JSON.parse(localStorage.getItem("cart") || "{}");
```

(4) GetElementById(): this method returns the element of a specific id. I used this method to return the quantity and glazing elements that I linked to the id's 'num' and 'glaze'. Specifically, they are returned once the 'Add to Cart' button is clicked.

```
function addToCart(){
 alert("Item added to cart!");
 var quantity = document.getElementById("num").value;
 var glazing = document.getElementById("glaze").value;
```

(5) ParseInt: The parseInt() function parses a string and returns an integer. This function changed my item quantities, which were initially stored in strings, into integers. Once they were converted to integers, I was able to calculate the total price (which equals the different item quantities multiplied by the individual bun price, $6.25).

```
if (productname in cart) {
  cart[productname] += parseInt(quantity);

 } else {
  cart[productname] = parseInt(quantity);
 }
```