Web Javascript Prototypes

Website

Link
https://kellikuramoto.github.io/pui-22/homework_6b/

Source Code
https://github.com/kellikuramoto/pui-22/tree/main/homework_6b

Feedback Implemented

Many of my peers were confused when the cart icon disappeared and changed to the feedback of how many items are in the cart. To combat this confusion, I decided to position the cart 'absolute' and relatively positive the number icons on the top right to still provide feedback for the user about how many items they have in their cart. Additionally, many users had a hard time finding the individual product pages because you had to specifically click on "Shop now". To resolve this usability issue I wrapped the entire div in a link to make the entire div clickable while still allowing users to click "Shop now".

Reflection (Bugs & Mitigation)

While coding for this assignment I ran into a couple of bugs in my javascript file: I had to figure out how to store objects in localStorage because the original functionality is for holding strings. Through some research of documentation on the web, I found that you can store objects using a 'JSON.parse()' function. Then the largest bug that I ran into during this project was trying to figure out how to update the object values according to user input. At first, I was trying to update the specific value in the object using localStorage; after speaking with my TA Vikram, he said a better method would be to change the object directly and then resave the updated object into localStorage each time, this worked significantly better. A smaller bug I ran into was trying to figure out when to reset localStorage. At first, I had cleared localStorage every time the user clicked on the 'products.html' page but quickly realized that this did not make any sense if the user wanted to add multiple items to their cart. I eventually landed on clearing it when they clicked the checkout button or the 'index.html' page. Lastly, my other large bug was trying to figure out how to add up the prices for the items in the cart. I found the parseFloat() function and the toString() functions which were very helpful. At first I was just using the parseInt() function so it was cutting off when it got to the decimal point but this was fixed once I changed the function and used .toFixed(2) to stop the float at the decimal place I wanted. This project was very challenging cognitively because I had to figure out how to store the information of the products, how to update these values according to user input, and how to display these values correctly in HTML. I thoroughly enjoyed this project as I know it made me a better programmer and problem solver.

Programming Concepts

1. localStorage and its corresponding functions

localStorage allows user input to be saved across HTML pages without needing to be reloaded every time a new page is opened. This is extremely helpful when trying to update the checkout cart according to user preferences. Before I would try to save the preferences on the individual product page, but this information would be erased as soon as I clicked out of the page.

```
function addProduct(productName, productObject) {
   localStorage.setItem(productName, JSON.stringify(productObject));

   let coffeeProfile = localStorage.getItem(productName);
   let savedCoffee = JSON.parse(coffeeProfile);
}
```

2. toString and parseInt() functions

toString converts an int to a String. This is helpful when trying to resave an int into an object that is held by localStorage because localStorage can only hold strings. parseInt() is the opposite function that converts a String to an int. This is helpful when trying to add up the subtotal of the individual product prices according to user preferences because addition of Strings concatenate the two Strings.

```
   var stringSubTotal = subtotal.toString();
```

```
       var stringTotal = coffee.price;
       var total = parseInt(stringTotal);
```

3. creating HTML tags and editing in Javascript

Creating HTML tags and being able to edit the innerHTML or text of those tags in Javascript is important for when you have information that you want to display but need to receive from the user before the page can be coded. I created multiple and <p> tags throughout my Javascript code so that I could display the user preferences accordingly.

```
var spanSubTotal = document.createElement('span');
   var stringSubTotal = subtotal.toString();
   spanSubTotal.style.fontWeight = "bold";
   spanSubTotal.appendChild(document.createTextNode(stringSubTotal));
```

4. creating objects in Javascript

While Javascript is not an OOP language entirely, objects are helpful when trying to save a large group of information related to one “object”. In my case, I wanted to store all the attributes for each individual product such as the quantity, price, and ground. I would then update these specific attributes according to user input and save to localStorage so that the system could remember and later display this updated information.

```
let chocolateLover = {
   coffee : 'Chocolate Lover',
   quantity : '1',
   price : '14.99',
```

```
   ground: 'Whole Bean',
   flavor : 'sweet / nippy / smoky',
   roast : 'Light',
   origin : 'South America',
   brewing : 'Espresso',
   description : 'Time for a taste. This is a kind of brew that's made for sharing and spilling some crazy stories. Full bodied, complex & smooth with notes of hazelnuts, smoky marshmallows, &  a sweet caramel drizzle. FOMO in a cup. Once you take a sip, you will go crazy and never go back'
};
```

5. changing CSS in Javascript

Changing CSS in Javascript is helpful when creating new tags in Javascript and you want to style them after they have been created. Styling is an important part of web development because it can help user readability with changes such as font weight.

```
   spanSubTotal.style.fontWeight = "bold";
```

HTML, CSS, and JS Files

See files in homework_6b folder.