Swetha Kannan
Homework 6B
Programming usable interfaces

Github repository: https://github.com/swethakan/pui-fall-2021/tree/main/homework_6b
Live website: http://thisisswetha.com/pui-fall-2021/homework_6b/

External Images used:
Images are open sourced from FontAwesome.com, Flickr.com, and from Unsplash.com(creators: Timothy Meinberg, Erik Mclean, and Pi Supply)

Reflection

Homeworks 6A and 6B were harder than others because they primarily required the use of javascript. I'm familiar with python and other languages so programming does not intimidate me. However, this was the first time I was working with storage across multiple pages.

The biggest thing that I struggled with in this assignment was allowing the user to customize items that were already in their cart. This required that I know where each cart item was in the json array and that I can map changed values to their place in the array. This gave me some headaches and I tried multiple solutions to make this possible. In the end, I don't think my chosen solution was very good only because it required some hardcoding. The way I found which element was being updated was by using the function '`Array.prototype.indexOf.call('parent','child);`' which I found in a stack overflow forum. It's not the best because it required me to know exactly what the parent div was and what the child was. I previously tried using the find() function we learned in the lab but I could not make it work for my purposes here. I'll explain more about how I used this new function below when I detail the five programming concepts I learned with this assignment.

Another challenge was how disorganized my code quickly became - it's hard for me to work with large code files because I hate having to comb through everything just to find a function. If I forget the name of my function, for example, I wouldn't even be able to use the 'find' command to easily jump to it. This was solved, somewhat by the use of multiple javascript files that separated all my functions into manageable chunks. In the future, however, I think I will need to start my coding with an organizational structure already in mind. This will prevent unneeded restructuring down the line.

Programming concepts I learned as a part of the assignment:

1) **Get a child element's index** - As mentioned before, I was able to use the '`Array.prototype.indexOf.call('parent', 'child);`' command in order to find the index of a child element as compared to the parent element. Using this information, I was able to map the elements of my cart to their respective location in the Json array of the cart.

2) **LocalStorage** - I also learned how to save items using localstorage. This was especially important for my design because I had two pages which displayed the content of my cart - my product and checkout pages. Because of this, I had to make sure that both pages were 1) checking if there was content in the cart when the page loaded and 2) displaying the content. I found that the hardest part was updating the contents of the cart everytime the user edited it while it was in the cart. Rather than updating what was in local storage, the best system was to move the localStorage values into a variable on the page, only fiddle and change this variable, and then trigger a function every time the user left the page to replace the localStorage values with this new variable.

3) **Template elements** - I've worked with javascript a few times before and the way I primarily made templates was to adjust the "innerHTML" attribute which forced me to write a lot of unnecessary HTML in my javascript page. By comparison, creating a template element and cloning it using javascript is a *much* easier way to do things. In this case, I used template elements in 3 instances: to make all my products, to populate the shopping cart in my product page, and to populate the cart at checkout.

4) **Using variables across multiple javascript files** - The more functionality I added to my project the more my javascript file became overcrowded and hard to read. This was when I began searching for how other developers organized their code and found that multiple scripts can be added to an HTML file. Not only that, but each script knows what the one before it has done. So, if I defined a global variable in one .js file, every other file I open afterwards will be aware of this variable and can use or modify it. This was incredibly helpful for organizing my code. Not only that, but it allowed me to separate out my code so that pages of my site that did not need all the functionality that my main page had, did not need to call a script file that had all those extras.

5) **Functions can serve multiple purposes** - This is not something I learned straight away in my deliverable and therefore isn't properly demonstrated in my code. My website works but the javascript side of it is messier than I initially thought it would be. Through trying to rework it to be more organized, I realized that it would have been helpful for me to make a list of all the functions each page would need (not just my products page) and figure out what the overlaps were. This framing helped me organize my *dropdown.js* file and because of this, I feel that the file has much more utility because it is made to work on both my product page and in the checkout page. In other words, It saved me from having to make dropdown functions specifically for one page because I identified from

the beginning that both pages will need to use these functions and therefore crafted them to be used that way.

Bonus/Extra credit:

For a summary of extra elements on this page, please read the README.md file in my repository. Thank you!