I. Bugs / Challenges Encountered

Summary Card
When designing the cart page, I had a vision for a card on the right that displays summary information about the order, including the total price, number of items in cart, etc. When implementing this design, however, I ran into difficulties because when I added more than three items to the cart, the checkout button went below the fold. To solve this issue, I I had to research how to implement scrolling the way I wanted, which involved testing many different types of scrolls. Once I learned how to implement this, I then had to design and size a div so that a certain section was scrollable, but the proceed to checkout button remained static, and always visible above the fold.

Remove Button
I ran into difficulties when trying to implement the Remove button on the Cart page. I originally tried to implement the function by setting it in the DOM with javascript code, calling it with parameters. The function was called every time the page was loaded, deleting all of the elements in the cart. To overcome this issue, I had to do research to learn that when setting the onclick function it should be called without parameters, and that when I write the function it takes in the parameter e. Then, I ran into the problem of not knowing how to delete a specific item from the cart. How would I know *which* item to remove? I had to do further research to realize that I can access the parent node from the "e" parameter that is the automatic parameter of the onclick function. Then, I created a new id onload for each of the elements in the cart, based on its index in the cart stored in memory. That made it straightforward to find which item in the cart I had to delete when the Remove button was clicked.

Reproducing Static HTML with Javascript Code
A final challenge i ran into was getting the items on the left hand side of the cart to show the way it did with static HTML. Overcoming this challenge involved a lot of translating between my static HTML and Javascript code to build the same structure in the DOM. I had to look up how to set elements a certain way (e.g. how to create a span element, how to create an onclick function, etc.). I also had to sketch out how my DOM looked in tree form, because I had so many divs and elements within elements in my static HTML, and translating these into lines of Javascript code was a very tedious task that involved making lots of mistakes

II. Programming Concepts Learned

1. How to save information in local storage

   Example: The cart, which is stored in local storage

2. How to implement scrolling in Javascript for a section of the card, while keeping the rest of the card intact

   Example: The summary card on cart page

3. How to implement an onclick function for an element that was added to the DOM, and thus set it within a Javascript function

   Example: the removeFromCart function, which is set as an onclick function within the updateItemsList function

4. How to build up a page originally written in static HTML using Javascript in the DOM

   Example: the left hand side of the cart, which is built up in the updateItemsList function)

5. How to utilize helper functions to keep good style

   a. *Example:* Because my static HTML for my cart page involved lots of divs within divs, my updateItemsList function, which built the same HTML page using Javascript, was very frustrating to look through. I learned the importance of breaking these longer functions into helper functions to be able to mentally manage all of the subpieces of very long functions