Assignment Reflections:

The main problem I encountered is storing the items data in localStorage. My first instinct was to store the items as an array of objects, but later I realized that I can only store strings in localStorage. After researching and reviewing course content, I decided to store items data as an object with index as the key, then use JSON.parse() and JSON.stringify() to manipulate the data.

It works pretty well at first, then after I'm implementing the remove() function for cart items, I started to find new problems - I cannot loop over all the items using index, because the indices performed as the object key and they are not updating after I remove any object elements. So I modified the renderCartItems function to loop over object keys to get information.

When reflecting, I don't think this solution is ideal since we're leaving "holes" between numbers after removing instead of updating the index. When the item number goes into a large amount, it might cause some overflow problems. But it works for this assignment so I will keep my implementation and keep researching about storing arrays in localStorage :D

New Programming Concepts:

1. localStorage manipulation:
   using localStorage.setItem to initialize and update the data field I need for the shopping cart

2. Handling edge cases:
   Initialize the data field when the user first comes to the page

```
Cart.js > addToCart
var itemCountDisplay = document.getElementById("item-count");
var items = [];

if (localStorage.itemCount === undefined || localStorage.items === undefined) {
    localStorage.setItem("itemCount", 0);
    localStorage.setItem("items", JSON.stringify(items));
    itemCountDisplay.innerHTML = 0;
} else {
    itemCountDisplay.innerHTML = localStorage.itemCount;
}
```

3. reload/redirect pages with JavaScript:

```
//remove the unwanted item
function remove(index) {
    let currentItems = JSON.parse(localStorage.items),
        currentCount= parseInt(localStorage.itemCount);
    delete currentItems[index];
    localStorage.setItem("items", JSON.stringify(currentItems));
    localStorage.setItem("itemCount", currentCount - 1);
    location.reload();
}
```

```
    let count = parseInt(itemCountDisplay.innerHTML);
    itemCountDisplay.innerHTML = count + 1;
    localStorage.setItem("itemCount", count + 1);

    window.location.href = "cart.html";

}
```

4. Enhanced for loops (for/of)

```
for (let content of contents) {
    let options = content.getElementsByClassName("pd-selection-option");
    for (let option of options) {
        option.addEventListener("click", function() {
            selectionToggle(this, options, content);
        });
```

5. Looping over an object by keys (for/in)

```
for (let key in currentItems) {
    let product = currentItems[key];
    subtotal += parseInt(product.price);
    document.getElementById("shopping-cart-items").innerHTML += `
    <div class="shopping-cart-item-detail">
    <div class="u-flex-row">
        <div class="item-detail-image">
            <img class="item-detail-image" src="
            `
```