

Diva Agarwal
Nov 8, 2021
Lab Section C

PUI Assignment 6B: Adding Functionality to a Website with Javascript

Client: Bun Bun Bake Shop

Link to Website: https://dagarwz.github.io/homework_6b/

Link to Source Code: https://github.com/dagarwz/dagarwz.github.io/tree/main/homework_6b

Reflection

I encountered several issues such as not really understanding how to implement local storage, some issues with html elements being in absolute position (so not being able to make them respond to additions or deletions in the cart), and creating an object and passing it as an array/object.

I was able to understand how to implement the above concepts with help from my *very helpful* TA - Jesse. I learned how to initialize a saved cart local storage by using `JSON.parse(localStorage.getItem("savedCart"))` and then ensuring that the cart array is initialized if the local storage is empty and if it's not empty - have it set to what's already in local storage. I also learned how to pass the cart array into local storage or set it to that by using `localStorage.setItem`

I also got some help with removing some elements from absolute position and putting them into tables so they are better formatted and more responsive to additions or deletions in the cart.

I created a product object in the javascript that uses `getElementById` to get data from the HTML for each product attribute such as product name and glaze. I then used `querySelector` to set the value of the attribute from the html (default value) to the updated value of whatever item was added from the product object.

Deleting objects was a struggle but I figured out how to do it by using `cart.splice` and `remove()` methods

`clone`

```
.querySelector(".itemDelete")
.addEventListner("click", function (event) {
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove()
    cart.splice(cart.indexOf(item))
    localStorage.setItem("savedCart", JSON.stringify(cart))
})
```

Programming Concepts

- Representing products as custom objects

```
const product = {
    // cartTotalItems: cartNumber,
```

```

    productName: "Original",
    productGlaze: document.getElementById("glaze").value,
    productQty: document.getElementById("quantity").value,
    productPrice: 1.75,
    productTotalPrice: 1.75 *
Number(document.getElementById("quantity").value),
  }

```

- Representing cart as an array of objects

```

let cart
if (savedCartLS == null) {
  cart = []
} else {
  cart = savedCartLS
}

```

- Using separate JS file for page-specific functionality
 - Used cart.js for cart.html in addition to main.js for overall site

```

<script defer src="cart.js"></script>

```

- Using the <template> tag for repeated HTML elements

```

<template id="line-item-box-template">
  <tr id="line-item-box">
    <!-- <div id="line-item-box"> -->
    <td>
      
      <h9 class="itemName">Original</h9>
    </td>
    <td id="price-r1" class="itemPrice">$1.75</td>
    <td class="itemQty" name="quantity"
id="qty-r1">1</td>

```

```

        <td class="itemGlaze" name="glaze"
id="glaze-r1">glaze</td>
        <td class="itemTotal" id="total-r1">$1.75</td>

        <td><h10 class="itemDelete">Delete</h10></td>
        <!-- </div> -->
    </tr>
</template>

```

- Adding event listeners in JS, instead of inline HTML

```

clone
    .querySelector(".itemDelete")
    .addEventListener("click", function (event) {
        var buttonClicked = event.target
        buttonClicked.parentElement.parentElement.remove()
        cart.splice(cart.indexOf(item))
        localStorage.setItem("savedCart", JSON.stringify(cart))
    })

```

Citations of External Resources Used

Images:

<https://www.svgrepo.com/svg/34854/cinnamon-roll>

<https://www.istockphoto.com/vector/cinnamon-roll-gm165058905-3408622>

<https://food52.com/recipes/85127-perfectly-pillowy-cinnamon-roll-recipe>

<https://www.foodnetwork.com/recipes/food-network-kitchen/almost-famous-cinnamon-buns-recipe-197300>