

Reflections

Link to the website: https://pvegired.github.io/homework_6b/

Link to the source code: https://github.com/pvegired/pvegired.github.io/tree/main/homework_6b

Remove Button:

I implemented the javascript functionality to remove a cart item. Overall I tried to target the parent row of the cart item using `getElementsByClassName` and subsequently reaching its parent element. Adding an event listener to the remove button I was able to write a function to remove the cart item row and also subsequently update the totals appropriately.

Bugs/Challenges faced:

1. Using Async:

One challenge that I faced is when I was trying to capture all the prices and inputs in an array, I `console.logged` and saw the divs that I'm targeting but `html collection elements` showed null. Digging into this further I found that as I'm keeping my javascript files in head and html elements in body which runs later, my javascript is running even when the html is not present. So I added "async" along with the `<script>` tag

```
<script src="cart.js" async></script>
</head>
```

2. "Event" is deprecated

```
var input = quantityInputs[1]
console.log(input)
input.addEventListener('click', function() {
  // ...
})

function quantity(event) {
  console.log(event)
  var input = event.target
  if (isNaN(input.value) || input.value <= 0) {
    input.value = 1
  }
  updateTotal()
}
```

'event' is deprecated. ts(6385)
lib.dom.d.ts(18286, 5): The declaration was marked as deprecated here.
var event: Event
@deprecated
No quick fixes available

After lot of time of debugging using `console.log` realized that this happening due to not writing event in the function like `function(event) {}` which is a silly mistake

3. Facing issues with having carts on a different page. I was able to do all the functionality if the cart was on the same page. Implemented with "product browsing" / "shop" page but facing a lot of issues with storing variables that I could use again in cart on a different page. May be that is easy to do with backend but it's being lot of hassle using only local storage variables to do so.

Programming Concepts:

```

console.log(price, quantity)
var priceNumber = parseFloat(price.innerText.replace('$', ''))
var quantityNumber = quantity.value
total = total + (priceNumber * quantityNumber)

```

1. Using 'innerText' to just extract the price
2. Using parseFloat to convert a string into a floating point number to count the total in a cart
3. Using event.target to write contextual functions when an event like changing the quantity or removing from the cart happens

```

function removeCartItem(event) {
    var buttonClicked = event.target
    buttonClicked.parentElement.parentElement.remove()
    updateTotal()
}

```

4. Using Math.round to round the total to two decimal places

```

document.getElementById('cart-subtotal').innerText = '$' + total
document.getElementById('cart-shipping').innerText = '$' + shippingRate
document.getElementById('cart-tax').innerText = '$' + Math.round((total * taxRate) * 100)/100
document.getElementById('cart-total').innerText = '$' + (total + shippingRate + (total * taxRate))

```

5. Using Arrays for implementing the carousel functionality in product detail page

```

var slideIndex = 1;
showDivs(slideIndex);

var slideIndex = 1;
showDivs(slideIndex);

function plusDivs(n) {
    showDivs(slideIndex += n);
}

function showDivs(n) {
    var i;
    var x = document.getElementsByClassName("broductn");
    if (n > x.length) { slideIndex = 1 }
    if (n < 1) { slideIndex = x.length }
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }
    x[slideIndex - 1].style.display = "flex";
}

```

Bonus Work:

1. Added javascript functionality to count the total after adding the tax and shipping



Longhui
Yarn Color: After School
Special; Foam: Memory
Foam

\$99

3

Delete

\$99



Limbhano
Yarn Color: Morning
Haze; Foam: Duckdown

\$99

1

Delete

\$99

Subtotal \$396
Tax (5%) \$19.8
Shipping \$15

Grand Total \$430.8

Continue to Checkout