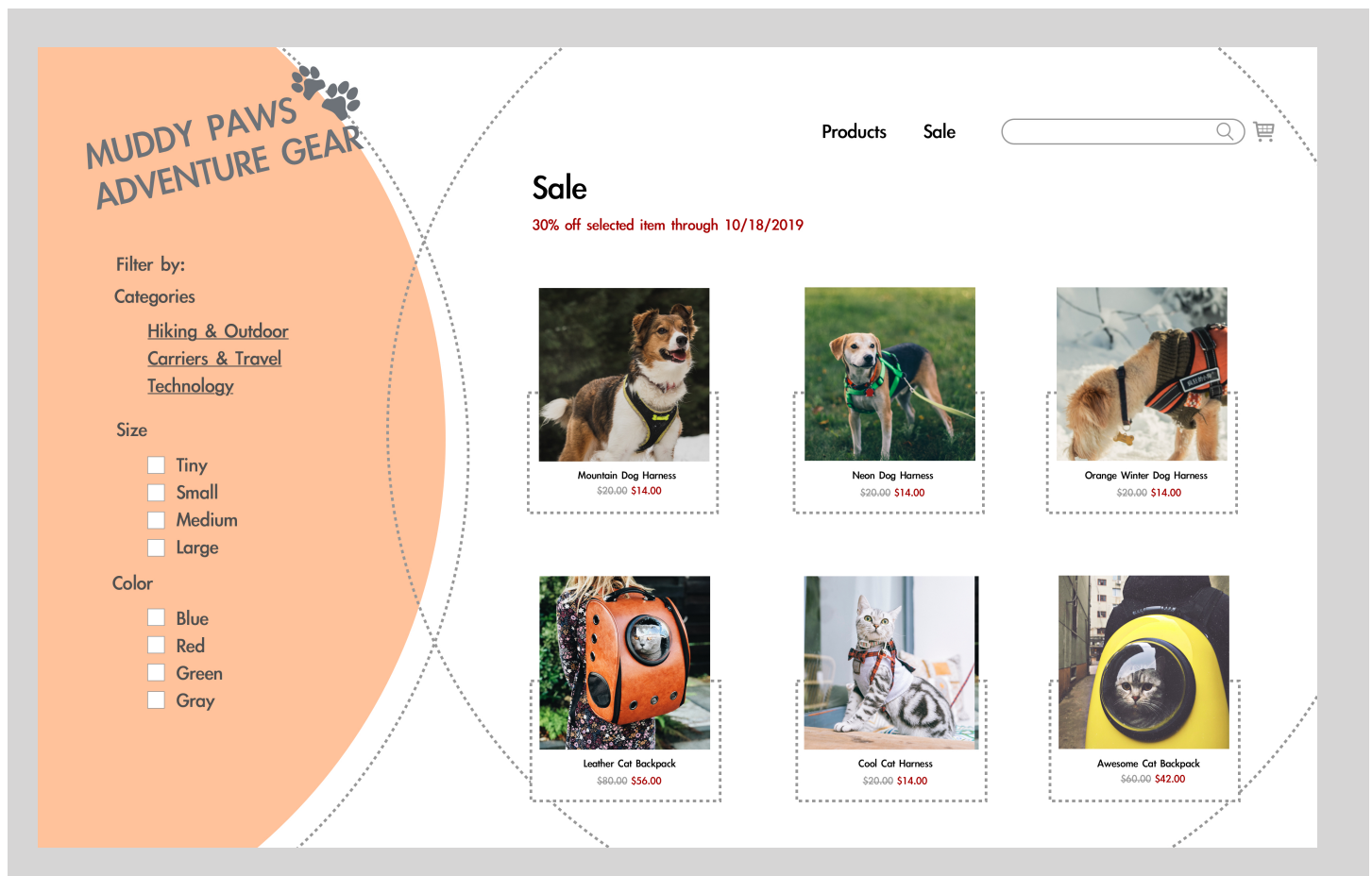
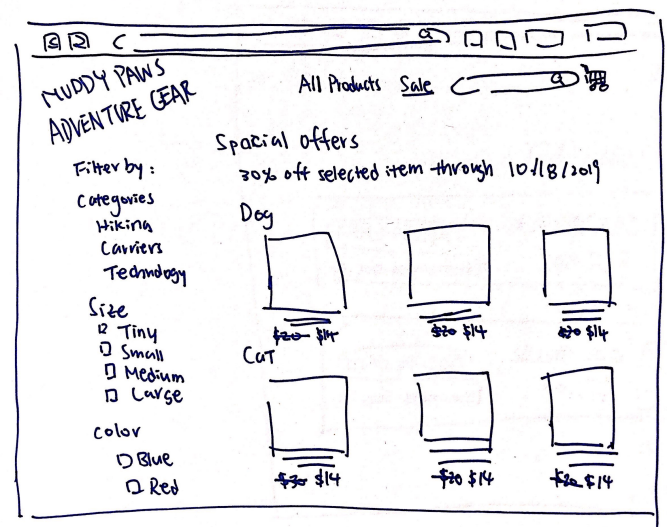


GitHub Repo: <https://github.com/mixichen/mixichen.github.io/tree/master/assgn6>

To view: <https://mixichen.github.io/assgn6>

Low Fidelity and High Fidelity Prototypes of New Page

I created low fidelity sketch, designed high fidelity mockup and coded a new “Sale” page for my site to display all sale items. The general layout of the page is consistent with the product listing page so that users will have no trouble understanding the page. On the top of the page, there is information about the sale including discount and date. Then, on each individual items, I indicated the original price and the discounted price. Finally, I added two buttons on top near the search bar to link product listing page and listing page together.



Challenges and bugs

The assignment overall is very challenging. Initially, it was hard for me to think of how to build the whole functionality from a high level. However, by looking back to each PUI lab lecture, I slowly built out a constructor, then a local storage, and then functions to retrieve info. I found displaying product information on Shopping Cart page the most difficult, because I wasn't sure how I was supposed to write out the HTML code for each new container. To solve that challenge, I looked up tutorials on the Internet and went to TA office hours.

I also spent a lot of time debugging my code, by console logging almost every function to check what mistakes I've made in each step. Through this process, I learned to make sure to write "getElementById" instead of "getElementsById". Also, I forgot to call functions in HTML several times and was confused about why nothing happened, so I also learned to always make sure to add an event in HTML to call the Javascript function.

Programming Concepts I learned

1. Writing innerHTML with template literals

I learned to use back-ticks (`) and dollar sign and curly braces (`{expression}`) to write my HTML code for every new shopping cart container. It becomes very intuitive once I get familiar with it.

```
var container = `<div class="shopped-item" id="Item${i}">
  <div id="img">
    <a href="cool-harness.html">
    <h3>${cart[i].name}</h3>
    <h4>size: ${cart[i].size}</h4>
    <h4>color: ${cart[i].color}</h4>
    <h3>${cart[i].price}</h3>
  </div>
  <div class="quantity_remove">
    <br>Qty: ${cart[i].qty}<br>
    <h5 id="remove" onclick="removeItem('${i}')>
  </div>
</div>`
```

2.JSON.stringify and JSON.parse

I learned that it's important to convert a JavaScript object or value to a JSON string when storing information in local storage. This is content covered in PUI lab lecture but I only learned the importance of it when I'm coding this assignment.

3. `object.style.display = "block"` & `object.style.display = "none"`

I learned this method from the Internet to hide my “Your cart is empty” message and “Proceed to checkout” button depending on the condition.

4. Push()

I learned the push() method when trying to figure out how to add a new set of product information into shopping array. This method adds new items to the end of an array, and returns the new length.

5.textContent vs. value vs. src

I learned to pay attention to the type of information I need to get from a Node and use different properties accordingly. Initially I used .value for all elements and ended up with “undefined” for many of them, so I learned to use different properties based on the set up of an element and the type of information I need.

```
//Get information from product details page
var productName = document.getElementById("product-name").textContent;
var productPrice = document.getElementById("detailed-price").textContent;
var productQty = parseInt(document.getElementById("quantitySelection").value);
var productSize = document.getElementById("size_selected").textContent;
var productColor = document.getElementById("color_selected").textContent;
var productImage = document.getElementById("coolHarnessImg").src;
```