05-430 Programming User Interfaces

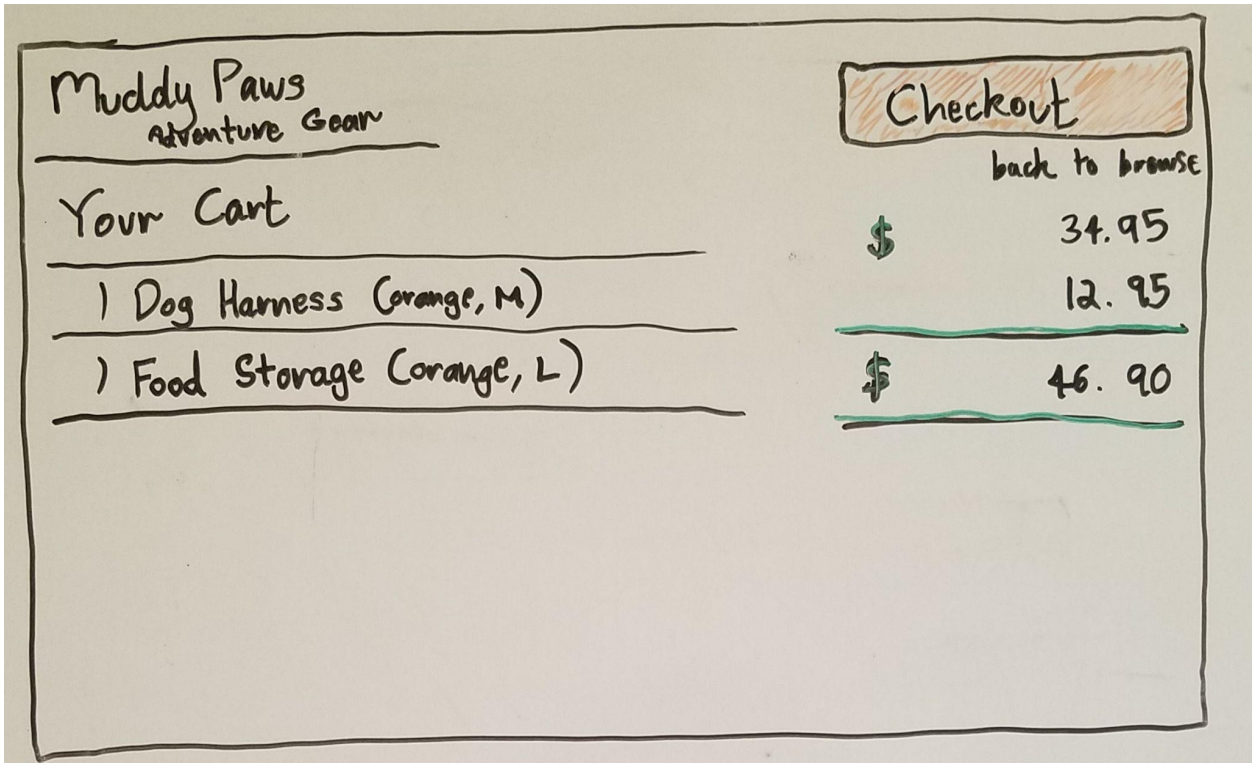
# Assignment 6B - Finishing the Shopping Cart

Rigved Deshpande (rdeshpan)

https://r1g.github.io/rdeshpan_pui_6b/

## Table of Contents

# Lo-Fi Prototype




Like the rest of the pages on the Muddy Paws site, I want to feature the logo at the top left of the window. There should also be a way to go back to the browsing page easily.

As for listing the products themselves, indicating the size and color after the product name is a quick way to read what the user has added.

# Hi-Fi Prototype


When going from lo-fi to hi-fi, here is what I had in mind:

1) Matching the bright, outlined, and cartoony visual style of the previous pages
2) Reducing some of the clutter in the lo-fi prototype

I realized that transparencies and drop shadows fit nicely into my visual style and decided to use them to further accentuate buttons and cart items.

# Reflection

## Issues/Bug Encountered

1. Removing a single item from the cart was removing every item from the cart

    By using console log statements starting from halfway through the codebase and narrowing down the source of error I realized that I was not clearing the cart correctly. In some cases I was treating the cart as a '|' delimited string and in other cases as an array. By treating the cart as an array consistently I was able to clear it correctly.

2. Clicking anywhere on the cart was causing items to be removed

    My approach to assigning events to several elements is to simply assign one event to a container of those events, and then looking at the event target for the individual event. This was easy on the previous pages, but the checkout page has a lot more whitespace between interactive elements and so clicking anywhere within the container was triggering events. Adding a simple if statement to check the event target's class name fixed this issue.

3. A blank entry was being drawn everytime the cart was redrawn

    Although the answer to this bug was trivial, it took me a surprising amount of time to figure it out. By using the split function to break the cart string into an array on the '|' element, I was creating an empty string element before the first '|' symbol. Starting my draw cart for-loop at 1 instead of 0 fixed this bug, although stripping the array of empty string elements would have sufficed as well.

5.Programming Concepts (5 points) a.Demonstrate 5 programming concepts that you learned in Javascript and used in this assignment with an example. b.Writing should use appropriate style and clearly to convey the writer's concepts (this includes grammar).

# Programming Concepts

1. ## Event Handling

   I did not know how event-based programming on the web is. I learnt how to assign functions that should execute on events such as ‘clicks’. Even CSS can be event-based, and I was able to make buttons look different when they were hovered over.

2. ## Parsing Between Data Types

   LocalStorage only gets and sets DOM strings as opposed to ints, arrays, etc. To make this work with my project I needed to parse strings into integers, split strings into arrays, join arrays into string, etc.

3. ## Order of Execution

   One of the tougher issues to debug was figuring out my script was running before my HTML page. In a previous Javascript course I would always load and execute my script by default on top of the page since I was not interacting with the HTML at all. Being mindful of whether the HTML elements I am interacting with in my script have loaded in advance was something new to me on this assignment.

4. ## Constant and Variable Data

   I did not really know the difference between the *const* and *var* until I got to this stage of the assignment. I was using a lot of constants before being told by the console that I was modifying data that I had marked as constant. Now I use a balance of *var*s and *const*s based on how I will be working with that data.

5. ## Trees with Variable Children

   The CS course I have taken so far are more theoretical than practical. The examples of trees that I worked with before were strictly binary trees, where there could only be up to 2 children for each node. In HTML documents however, there can be several children. Towards the end of of Assignment 6B I was inserting several child elements into containers via Javascript.