Faye Fan
PUI HW6 Write-up
10/26/2018

1. **Low & High-Fidelity Prototypes**

In my low-fi prototype, made a shopping cart page that contains a cart section and an order summary section. The page follows the design patterns for the shopping cart of some popular e-commerce sites, that shopping carts have separate sections focusing on either items added to cart, or the price and order related information.

In my high-fidelity prototype, I changed my previous shopping cart mockup so that it meets the requirement of the assignment. Previously the shopping cart was designed to be a sidebar that will show up on the current page when user clicks on the shopping cart icon. However, in the requirement of this assignment, the shopping cart need to be a new page. Therefore, I redesigned the page following the grid and color theme of my other pages.

All the pictures are uploaded in github folder *hw6 -> prototype pictures* .

2. **Please see github for web prototype w/ js**

3. **Reflection**

One biggest challenge I encountered the organization of my local storage. Due to the nature that we are storing information as key-value pairs in the local storage, I first ran into a problem that my stored object was constantly replaced if I store a new object, because I didn't realize that I have been using the same key in the "setItem()" function. After I debugged the problem by adopting an index system to store my objects, I realized that the counter variable wasn't global in terms of being stored across multiple web pages. Therefore I had to store the counter into the local storage, so that it could be updated whenever I add/delete buns in these pages. However, the act of saving the counter variable in the storage made it more difficult to keep track of the total number of objects saved, and hence hard to decide what is the next available index to use as a key for save a new object. It was a messy journey of debugging and doing calculations of how to structure the indices, but I finally came up with an indexing system that works for my specific case. In the future, I think I would use a dictionary data structure to save and manage all of these individual objects, then save the one dictionary into my local storage.