CHAAS

05-430: PUI

Homework 8: Final Project Part 2

Part 1: Purpose

The purpose of the website is to visualize sorting algorithms. Additionally, it is intended to act as a portfolio project. The visualizer has an interface that allows the user to randomize the array of colors, select a sorting algorithm, and change the size of the array. After pressing the Sort button, the user can watch the sorting magic in real time. The target audience is potential employers who are checking out my portfolio website and want to view a project that showcases my experience.

Part 2: User Steps

- When loaded in, the user is presented with options on the left (or above if on mobile) and a random color array on the right.
- The user can press the New Array button to randomize the colors
- The user can select a sorting algorithm (only Bubble sort animation implemented thus far, the selection sort will only display the resulting sorted list).
- The user can use the slider to change the size of the array in real time.
- The user can press the Sort button to sort the list (Use Bubble Sort)
- The user can repeat these steps again.

Part 3: External Tools

- Jekyll
    - Used to create partials in a static web environment.
    - This plugin was used because my portfolio was previously built using this, it was only used for the HTML layout of the sorting.html page.
    - This technology allows for easy implementation of recurring elements by utilizing partials.
- jQuery (and a Swap plugin)
    - Used for the shorter JS functions, as well as a Swap function that animates two DOM elements switching.
    - I used it to manipulate document elements of the page, and also to display the swapping animation.
    - The additional plugin gave me a quick and easy way to swap the position of the element and animate the process with a given speed.

Part 4: Design Iteration

I did not iterate on the wireframes, but when implementing the website, I added a number to display the size of the array because it was easier for the user to understand the purpose of the slider. Additionally, I narrowed down the list of Sorting Algorithms because the animation process was more intensive than anticipated.

Part 5: Challenges

I encountered many challenges implementing this website because the animation process required multiple time sensitive functions. As such, I was dealing with async/await promises that did not line up with animations, and as a result, I had to limit the number of sorting algorithms. I was able to successfully animate the BubbleSort algorithm as it only iterates once at a time, but for other algorithms that required multiple iterations at once, the animations became increasingly difficult. I plan to refactor later and solve this problem at a later date.