Nathan Jen

November 8th, 2020

Assignment 6B

**Links:**

Repo: <https://github.com/NathanJen/assignment_6>

6b code is in the 6b branch, 6a code is in the master branch

**Live Site (Part B)**: https://admiring-poitras-cdc3b5.netlify.app/

Live Site (Part A): <https://nathanjen.github.io/assignment_6/>

**Challenges / Bugs:**

The way that I implemented the cart for this assignment is via React Context, which allowed me to pass data to any component in the component tree. While React Context was convenient in that I didn’t have to pass props manually down at every level, it presented the same challenges that global variables would present. In particular, I found that when I passed my cart context to all the necessary components, it became difficult to remember where it was being passed as well as how each component was using the data from the cart context. This inevitably led to a countless number of bugs, which was frustrating to debug as my intuition about where the bug was occurring was often wrong. This is usually a major challenge you have to deal with when you have data at a global level, and is often why global variables are dangerous to use. The way that I dealt with this challenge is through console.log. For instance, every time I call the cart data from the cart context in a new component, I would console.log the value to see whether the data that is passed to the component is the data that I am expecting. If not, I would try to follow the cart data as it gets passed throughout my application, subsequently adding a console.log line to see if the cart data is getting passed correctly. The moment the console.log differs from my expectations, I then investigate to see what’s causing the bug.

Another challenge that I faced during this assignment is the difficulty in deploying a React application. In particular, I found that when you want to deploy a React app, you have to follow a lot of steps correctly or else nothing will work. For instance, the first time that I deployed, I ran into an error revolving around setting the wrong homepage in my package.json file. Confusingly, you have to add your repo name at the end of the URL, while I intuitively thought that the homepage should just be the website name itself. Additionally, I kept forgetting to run my deploy command, and instead thought that pushing my code to my repo would just update my deployed site. Finally, I was having a problem with displaying my home component on the home page because my homepage had me repo name at the end of the URL, which did not match the home path that I gave to React Router. I eventually alleviated this issue by not giving my home route a specific path and putting my home route last, so React Router would try to render the other Routes before arriving to my home route. For many of these errors, I was able to resolve them by Googling my problem. Ultimately, I felt that deploying was so challenging for me because I felt that there isn’t a consistent behavior between my local site and my deployed site, as evidenced by my Router problem. Additionally, you have to perform extra steps to update your deployed site compared to your local site, which can be easy to forget. Personally, I found that I was rarely running the build and deploy command because I thought that pushing my code to my repo would update my deployed site automatically. I have yet to overcome this challenge and I hope my deployment challenges lessens or disappears as I gain more experience.

**Programming Concepts Learned:**

Because I created this assignment in React, the first programming concept that I learned is modularizing my code. Compared to the previous assignments, this assignment really helped me understand why modularizing your code is so powerful. For me, the best example is the header and footer components. In the previous assignment with static HTML and CSS, I had to copy and paste the same header and footer code into each HTML file. Not only did this make my code less readable as it appeared more cluttered, it also meant that I had to change the code in all my HTML files when I made even the smallest change. Additionally, I found that I would often times put my brain on autopilot and skip updating a HTML file or two, or even update my code incorrectly. With code modularization, I only had to update the header or footer once, and the change would be visible across all pages.

The next programming concept that I learned is working with JSON. One major decision that I had to make when starting this assignment was how to best store the data of my products, such as the assets, name, and price. I decided to use JSON since it’s incredibly popular today and easy to use. Since I have had experience using objects before, creating and getting data from my items JSON file came pretty naturally as I just had to remind myself that I was pretty much dealing with objects. For instance, when I made a item JSON object, I just had to add data through it via key value pairs. Additionally, getting data from my JSON file was extremely easy as I just had to export an array of my JSON objects, then use the item objects like I would use any other JS object.

Another programming concept that I learned from this assignment is handling multi-processing with React. More specifically, I found multi-processing to be incredibly simple in React since React gives the developer a lot of control over when re-renders occur. The best example that demonstrates this concept is changing the image on the product details page whenever a user selects a certain color. In vanilla JS, you would probably have to write a function that takes in a color and item, then returns the image of the item in the color you passed in. This becomes incredibly easy with react as you can just set a state component. Since you know that a re-render will occur whenever your component’s state changes, you can just make the product’s image depend on the state, and make the color buttons change the component’s state to match the color the user clicked on. In essence, this assignment really allowed me to learn how to handle multi-processing, which was made a lot easier by React since I had control over re-renders.

Additionally, one of the biggest programming concepts that I learned with this assignment is the power of using a popular library that has a lot of help and documentation online. For me, while I am not an expert at React, I was confident that I was going to be able to complete this assignment using React because of its good documentation (in my opinion), and the amount of help available online through websites such as stack overflow. In particular, this assignment was the first project that I used React Contexts. While I started this project with zero knowledge about Contexts, I was able to start using it to pass my cart data throughout my whole application pretty quickly. This was because I really like React’s documentation, and it helped me understand why I should use contexts as well as how to use it. Additionally, since I have never used contexts before, I ran into a couple bugs, mostly related to not passing my data properly. But, since React is so popular, I was able to Google my error and go on websites such as stack overflow to understand why the error was occurring as well as how to fix the error.

The last programming concept that I learned is using functional components. Since I did not use hooks in my assignment, the rule of thumb that I abided by was to use functional components for any component that did not need state. I really like the functional component programming concept because handling props feels more natural and clear compared to class components. For instance, in my CartItem functional component, I needed to pass data such as the item as well as callback function to update the cart item. For functional components, I was able to pass props as an argument of the function, which helped in reminding me the data I had to use in my component as well as what I had to pass in. Being able to pass props as an argument really resonated with me because it aligns with the standard way that you receive data when programming.