

HW 6B Reflection

Assignment Challenges Reflection

The first issue I ran into was regarding implementing localStorage. I had initially thought there was a bug in my localStorage implementation because the cart on a separate page would not update when a new Appointment object should have been added to the array of upcoming appointments. I then used console.log() and viewed my localStorage using Developer Tools and saw that my localStorage was actually getting saved correctly. It was just that for some reason when I opened my Cart page (aka the Upcoming Appointments page), the localStorage would get reset. Upon reviewing my code, I realized that I was resetting my array in localStorage to an empty array every time. I then had to figure out how to initialize my array in localStorage to an empty array only at the beginning so that it didn't reset every time, and I did this by simply using a conditional any time I tried accessing the array to see if localStorage.getItem() would return null.

An issue with my interface that I would like to fix is on my appointment time slot listings page. In order for users to see which date each time slot corresponded to without also including the date in the button text, I organized the time slots in columns in a table (each column corresponded to a day of the week). However, in order to add a <td> element in column 3 for example, I would have to make sure there is a column 0, 1, and 2 already. I wasn't sure how to implement this and so simply created a new row for each new appointment. It allows the website to function, but because of the unnecessary space, it's a bit unwieldy and confusing.

Programming Concepts

1. I learned how to insert rows into a table using JavaScript without manually adding <tr> and <td> elements as text via innerHTML. I was able to learn about the insertRow() and insertCell() feature by looking online if this functionality was possible. This proved extremely helpful since I have several tables throughout my website, since tables make grid layouts much simpler.
2. I learned how to add and access entire JavaScript objects as arguments in a function by using array indexing. Previously, I had simply been passing selected attributes of an object as function arguments (i.e. just the time attribute of an Appointment object) because I wasn't sure how to include the entire object as an argument. After talking to the TA, I changed my code to pass in the array index of the object as the function argument. This way, I wouldn't have to add the entire object and could instead just access the object by indexing into the upcoming appointments array in a separate function. It made my code a lot cleaner and would still allow me to access all the attributes of an object.
3. An important concept I learned in regards to programming style was how to split up a single line of code into several lines. This is helpful for breaking up a very long line of code so that it's easier to read without scrolling. I now include the "/" to indicate that a

string is continuing onto the next line and is part of the same command. This way, my code is generally under 80 characters long and has better readability.

4. To improve site efficiency, I changed my function to only reload part of a page rather than the entire page. Previously my `cancelAppointment()` function would pop the selected appointment from the upcoming appointments array and reload the entire page so that the number of appointments displayed in the cart could get recalculated. I disliked this because it was unnecessary and took a bit longer, so I changed the function to simply call `loadCart()` again. Because this would just add additional HTML to the existing cart, I also added a line to reset the `innerHTML` of the table every time the function is called. This almost completely got rid of the lag after canceling an appointment that would have been introduced if the entire page had to be reloaded.
5. Lastly, I was able to improve my code by keeping track of fewer variables in `localStorage`. I originally had kept an upcoming appointments array as well as an integer variable that would hold the number of upcoming appointments to display on the cart page. I decided to get rid of this second variable because I could instead use the `length` attribute of the array to get the same value. This could be slightly less efficient, but also a lot more sustainable for future additions to the code since there would be less (or rather, no) room for discrepancies in the data.