Assignment 6B

Submission Links

Github: https://github.com/jamiejliu/homework_6b
Firebase: https://homework-6b-dece8.web.app

Reflection

When working on this assignment, I faced a couple of challenges, especially when implementing the cancel functionality. The largest challenge that I faced occurred when my cancel function worked on one appointment, but not two appointments in a row. In other words, if users had two appointments, they could only cancel one. They needed to refresh the page to be able to cancel the remaining appointment. While debugging, I realized this was happening because removing an appointment from the array changed the array indexes. Because of this, the index of the appointment from the initial load no longer matched the modified array index. I discovered that I could solve this problem by adding a line of code in the function that refreshed the page. This ensured that the indexes lined up correctly after each appointment was cancelled.

I also faced challenges building onto my existing code as I had to modify parts of it to fit this new functionality. For example, I initially implemented a for-loop to iterate through my appointments array and populate my html page. To cancel an appointment, I would need to determine the index of the appointment that would be cancelled. However, I faced a challenge because variables are limited to the functions. To overcome this challenge, I first moved my appointments array out of a function, so that I could easily call it. I also passed the index of the for-loop into the appointmentModal function and the cancelAppointment function. Finally, I had to modify those functions to take the index as a variable.

The final large challenge that I faced was that the text illustrating the number of upcoming appointments did not appear after the appointments were cancelled, when the total appointments became zero. I realized that this was happening because my original if-else statement that populated the text did not consider the condition when the array was 0. The if statement only addressed when the appointments array was null. Once I figured out that this was the problem, it was simple to find a solution because I just had to add an or-condition to account for when the array has a length of zero.

Now that I have encountered these challenges, I will be able to spend less time debugging these mistakes in the future. I will remember that I should refresh a page after slicing an array, and I will be more aware when writing if-else statements to ensure that I address all of the possible conditions.

Programming Concepts

When working on homework 6, I was able to experiment with and implement many new Javascript concepts, including the following five:

1. Store Values into Local Storage:
    a. I used this programming concept to store the information regarding the appointment type, date, and time based on the user selection. Users are able to schedule new appointments, and their selected information is saved into local storage whenever they click the “Confirm Appointment” button.
2. Conditional Statements:
    a. I used an if-else conditional statement for my Appointments page. I wanted two different types of text to appear, depending on how many appointments were scheduled. If the user had no appointments scheduled, then I wanted the text on the page to read: “You have no upcoming appointments scheduled.” Otherwise, I wanted the text to read “You have x upcoming appointments,” with x being the number of appointments. I achieved this by learning how to implement a conditional statement in Javascript.
3. For Loops:
    a. Within the “else” of my conditional statement, I had a for loop to populate the page. Since the number of appointments was variable, I used a for loop to execute the same block of code a certain number of times, depending on how many appointments were scheduled. This for loop cycled through the indexes of the appointments in the appointments array, and then added each one and its details onto the page.
4. Set Attributes:
    a. I was able to set new values to attributes using the programming concept setAttribute. This was immensely useful because it allowed me to pass the type, date, time, and index of specific appointments from one function to another. I used this to populate the modals and add the cancel functionality.
5. Splice Arrays:
    a. In addition to passing the index from one function to another, I also used the splice function when building my cancel function. With my prior programming knowledge in Python, I knew that typical programming languages would have the ability to edit a list or array. In Javascript, I learned

that this was done using splice. By passing in the index of the selected appointment, I was able to remove it from the array.

Peer Feedback:

After receiving my peer feedback, I made a few changes to my code. My peers found that some of the information in the tables on the Home page was confusing, especially regarding the test status. Initially, I displayed that a lab result was available with a status of "Complete | View Test Results." I received feedback that the vertical line suggested that this should be two links, not one. As such, I simplified the status to just "Complete" to avoid confusion. I also received a suggestion that it would be nice to be able to click on the headers to go to each page. I really liked this idea, and I implemented it.

Additionally, I did find that I received some contradicting feedback for some pretty major parts of the page. For example, I received compliments on the overall website color scheme as well as feedback to replace the tan with a different color. I also received feedback that the home page and appointment were consistent with each other as well as feedback stating they were too similar in appearance. This was a little confusing, but I attributed it to personal preference. Since these were pretty major changes, and I am pretty happy with how my website looks, I decided not to make these visual changes for now. Further research could be conducted to see if these changes would impact overall website usability.

References:

https://www.w3schools.com/jsref/jsref_splice.asp
https://www.w3schools.com/jsref/met_loc_reload.asp
https://www.w3schools.com/js/js_if_else.asp
https://www.w3schools.com/js/js_loop_for.asp
https://www.w3schools.com/jsref/met_element_setattribute.asp