**KEY:**

\*: Indicates a design decision that will affect the structure of the code

-: Indicates a addition or deletion of different elements

**#:Indicates a start of a new section of code**

**$: Indicates a file/section of the project is completed**

&: Indicates a bug discovery and/or how it was handled

*Note: Start date was 3/17/2023. The days between then and now have been research on JavaScript, CSS, and HTML*

**Day 1 (3/21/2023):**

**Todays Goal: Get the Buttons functions implemented and working**

-------------------------------------------------------------------------------------------------------------------------------

-Set up development environment on Visual Studio Code and Cloned the Shine Solar Challenge from <https://github.com/ShineSolar/BYUI-technical-challenge>

to

<https://github.com/Programmer061703/ShineSolarChallange>

-------------------------------------------------------------------------------------------------------------------------------

- Added Website that can be run through the Visual Studio Extension “Live Server”

-------------------------------------------------------------------------------------------------------------------------------

- Implemented a basic button “without styling” with the ID button\_display, then tied it to the button1.JS file

-------------------------------------------------------------------------------------------------------------------------------

**#Began working on implementing the first button**

-------------------------------------------------------------------------------------------------------------------------------

\* button1.JS is going to be responsible for running through all the JSON information and displaying it to the screen.

-------------------------------------------------------------------------------------------------------------------------------

\* For the button1.JS file use 2 functions. The first function “clearTable” is responsible for clearing the table every time the button is pressed and the second function “displayAll” is responsible for appending all information from the JSON into the created table.

-------------------------------------------------------------------------------------------------------------------------------

\* The clearTable() function is designed to make sure no extra data is left over from the previous iteration of the button press in case new information is added to the JSON file. It probably will never be used in this environment, but it is added as a precaution.

-------------------------------------------------------------------------------------------------------------------------------

- clearTable() function successfully implemented

-------------------------------------------------------------------------------------------------------------------------------

\* For the displayAll() function, I chose to use a for loop to iterate through the JSON data. For every iteration of the For loop a new row variable and 4 box variables are created. Then each of the variables are assigned to a position i. Then they are appended to the table.

-------------------------------------------------------------------------------------------------------------------------------

- Implemented the displayAll() function

-------------------------------------------------------------------------------------------------------------------------------

-added a listener for when the button is pressed it runs the 2 functions clearTable() and displayAll()

-------------------------------------------------------------------------------------------------------------------------------

**$ button1.JS first iteration complete, some debugging still required but it displays the data correctly**

-------------------------------------------------------------------------------------------------------------------------------

& Fixed bug in displayAll() where the table would not display as the function was originally using the entire data from the JSON file and not a specific array.

Graphical user interface, text, application

Description automatically generated

Before fix, the for loop would not run as argument does not know the length of the entire JSON file.

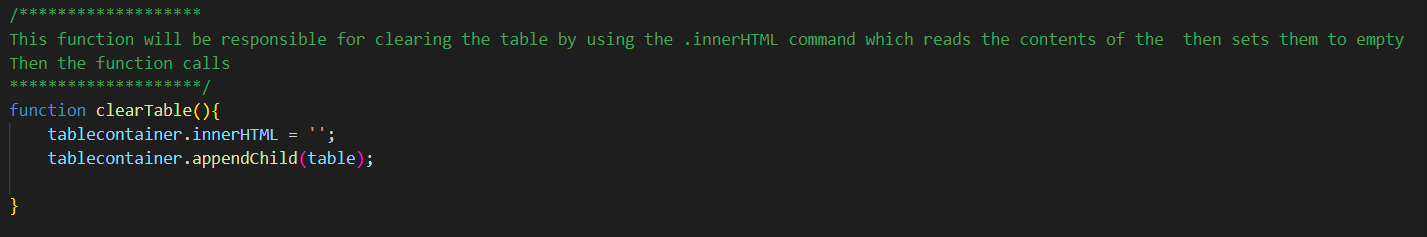
Text

Description automatically generated

Now the array “users” is stored in the const users, and the .length method is able to be used as an argument in the for loop

-------------------------------------------------------------------------------------------------------------------------------

& Found a bug where every time the button is pressed the table does not erase itself, but instead adds the same JSON data to the end of the table. It showed that the scrolling is working fine, but the clearTable() function is not.



I believe the error is that I am attempting to append table to tablecontainer, but table stores all the values of the JSON data after the displaylAl() function is called.

Option 1: remove every element of each row exept for the First One

Option 2: Create another table variable and leave it empty, then append the tablecontainer with the empty contents of the new table variable.

Option 3: Use a for loop to iterate through the table and clear the data before new data populates it

-------------------------------------------------------------------------------------------------------------------------------

\* Option 3 was chosen as other methods of displaying the table would create many errors in the system with my current ability of coding

-------------------------------------------------------------------------------------------------------------------------------

**Ending Notes:**

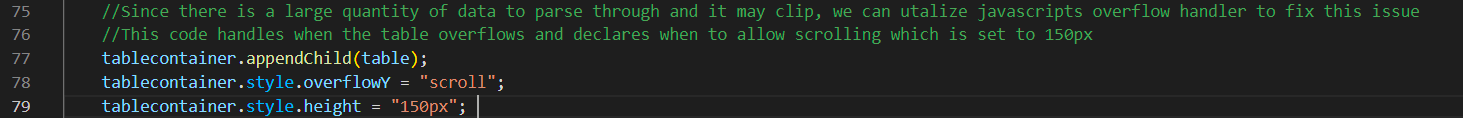
Implemented the first button functionality and it successfully displays the data of the JSON file. Still need to fix the bug of removing the data only. So if the JSON file changes then the table clears the data and populates it with new data instead of continuously populating it.

**Day 2:**

**Today’s Goal: Finish fixing the bugs on the button1.JS, then get the alignment of the first table completed**

-------------------------------------------------------------------------------------------------------------------------------

& Bug found when the table scrolls the header goes with it, the fix for this was to move the following implementations of the style of the table from the js file to the css file.



Old implementation attempted to handle the styling of the table inside the button1.JS file.

Text

Description automatically generated

New implementation utilizes the style.CSS file to handle pinning the first row to the top by declaring that the ‘th’(header cells) are to remain sticky at position 0. Then #table targets the HTML element with id table (table container) to declare when overflow is detected, and what to do when overflow is detected on the y axis.

*Note: The colors chosen were to make the table pop and will be changed in the final product*

-------------------------------------------------------------------------------------------------------------------------------

& To fix the bug of when the button is pressed multiple times, it appends new content to the end of the table, a new variable will be created that is tied to every ‘tr’ value. This will allow me to create a for loop to iterate through the table and clear all the contents without touching the header as it is a ‘th’ value.

-------------------------------------------------------------------------------------------------------------------------------

& Instead of using a for loop for the remove table elements a while loop will be used

-------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------

& Fixed the Bug that deletes the contents of the table except the header by using a while loop

Text

Description automatically generated with medium confidence

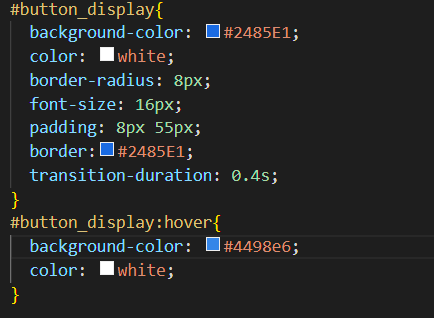
-------------------------------------------------------------------------------------------------------------------------------

**#Began work on styling the website and getting alignment correct**

-------------------------------------------------------------------------------------------------------------------------------

- Implemented the design of the button and made it so if you hover over it changes color to a lighter blue and changed the name of the button from press to click.

Reference: <https://www.w3schools.com/CSSref/sel_hover.php>





-------------------------------------------------------------------------------------------------------------------------------

\* Beginning work on the styling of the website. Going to use a <div> to encapsulate the button and table to encapsulate these components. Then using css make the <div> pop out to the user.

-------------------------------------------------------------------------------------------------------------------------------

- Used the box-shadow property of css to add a 3-D box around the content ( button and table)

Text

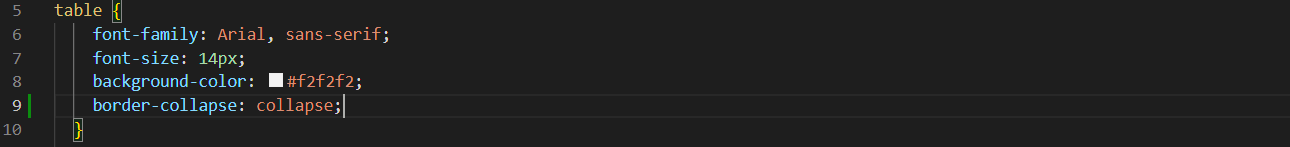
Description automatically generated

Background pattern

Description automatically generated

-------------------------------------------------------------------------------------------------------------------------------

& Fixed a bug when the table is displayed the first row “Wiggles” a little bit instead of remaining fixed in place. After some research I discovered that the first row is default set to boarder-collapse: separate and that there was some wiggle room in the first row that made it seem to “Wiggle” when the user scrolls. To fix this a declaration in the section of the css file table{} the “boarder-collapse: collapse” was set. This fixed the issue



-------------------------------------------------------------------------------------------------------------------------------

\* Changing the table and button to not use fixed pixel values, instead they will use percentages so any monitor can properly display the correct alignment

-------------------------------------------------------------------------------------------------------------------------------

\*The previous method does not work so will be scrapped, instead the box{} in the .CSS file will be changed to achieve the

-------------------------------------------------------------------------------------------------------------------------------

- After testing various methods the best way to correctly align the button and table was to set the position of the box to sticky. This causes the button and table to correctly align to their position based off the size of the box

Graphical user interface, text, application

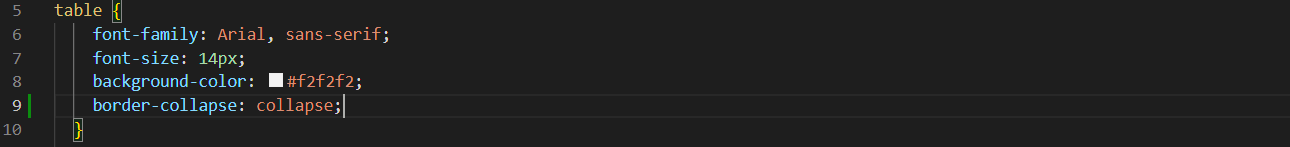
Description automatically generated

-------------------------------------------------------------------------------------------------------------------------------

-Did Some changes to the way the data is displayed to the screen along with some alignment of the box and tables to look more like the examples given

-------------------------------------------------------------------------------------------------------------------------------

-Changed the way the table is displayed, now the table does not stay at the top of the screen. Instead, it scrolls with the rest of the table. Also discovered how to add borders to the elements.



This is the Old table container that used border-collapse:collapse

Text

Description automatically generated with medium confidence

Now it uses border-collapse: separate to allow there to be border spacing which allows for spaces between the rows.

-------------------------------------------------------------------------------------------------------------------------------

-Added methods to add spacing between the rows, and the color changes when you hover over a row.

Graphical user interface

Description automatically generated

-------------------------------------------------------------------------------------------------------------------------------

**Ending Notes:**

Today the first button was successfully implemented, and its logic is working correctly. Then further development on the formatting of the HTML webpage was done. 3-D look was added to the container box that holds the Table and Button. The Table was worked on to add scrolling, highlighting of rows when the user hovers, and adjusted the code so the top header moves when the table is scrolled. Also adjusted the location of the button. Now when the website is opened on different monitors it should adjust the location of the button and table according to the size of the box. Tomorrow will be when the first section of the three sections gets completed. The plan is to format the size of the box along with the size and look of the table. Then begin working on the implementation of the other two buttons which should follow the same logic as the first button.

-------------------------------------------------------------------------------------------------------------------------------

**Day 3:**

**Todays Goal: Add curved edges on the ends of the rows, and add spacing between each row to make the table stick out. Design the logic for the next two buttons and implement designs into .CSS and html files. Finish all alignment and turn project in.**

-------------------------------------------------------------------------------------------------------------------------------

-Discovered a way to use first-child and last-child to apply a one-sided curve to the left of the first cell and right of the last cell. Making the row an oval shape. Also changed the color of the table to a light grey.

Reference: <https://dev.to/temmietope/rounded-edges-on-table-rows-1d0n>

Text

Description automatically generated

Text

Description automatically generated

-------------------------------------------------------------------------------------------------------------------------------

&Bug found where each row would not be highlighted, to fix this I changed the tr:hover{} to tr:hover td so it targets each cell in the row and highlights them when the row is hovered over. The color is temporary. The same property was also applied to the th row.

Graphical user interface

Description automatically generated

-------------------------------------------------------------------------------------------------------------------------------

**# Began Work on Button 2**

-------------------------------------------------------------------------------------------------------------------------------

**\***Button 2 will utilize the same stile as Button 1 but in a separate file and will access a different array of elements.

-------------------------------------------------------------------------------------------------------------------------------

**$ Implemented the button2.JS, may go through and rename some of the variables for clarity, but button2.JS is essentially button1.JS with different variable/function names and is tied to the orders array instead of the users array.**

-------------------------------------------------------------------------------------------------------------------------------

# **Beginning work on Button 3**

-------------------------------------------------------------------------------------------------------------------------------

\* Going to use the button2.JS file to implement another display function that will be tied to button 3 and uses a if statement to search for the users Jane Doe so there isn’t a ton of .JS files all over the place. After the buttons are implemented going to finish the ..CSS and looks of the page. Luckily the table variables are shared across tables so if I change 1 table it changes them all.

-------------------------------------------------------------------------------------------------------------------------------\* Instead of making a function I am just going to make another .JS file that makes the table then use a filter to filter out all other orders than Jane Does

-------------------------------------------------------------------------------------------------------------------------------

**-** Added button3.JS which takes the same code from button2.JS and modifies the way the array is fed into the const orders. It uses Java Scripts built in .filter application.

This is what the original method of storing the array of orders into the const orders.



This is the new method of storing the array of orders into the const orders. It looks at the orders array and checks the userId by using the .filter() method. Then stores only values with a userId of 4 into the orders const.



-------------------------------------------------------------------------------------------------------------------------------

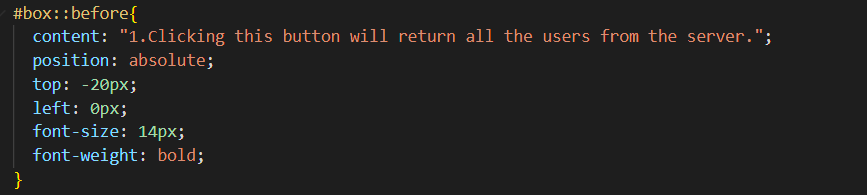
**$ Finished button3.JS**

-------------------------------------------------------------------------------------------------------------------------------

**# Finalizing the styling of the Webpage**

-------------------------------------------------------------------------------------------------------------------------------

-Discovered that I can use .CSS psudoelements to insert text inside the box margin using the code #box::before{}. Position: absolute; is necessary so the top -20px; and left: 0px; are applied.



Reference: <https://www.w3.org/TR/CSS2/generate.html#before-after-content>

*Note: This is not the final font or size, will be changed in the final draft*

-------------------------------------------------------------------------------------------------------------------------------

-Added text to all the tops of each box and changed the color to match the reference provided by Matthew Picket

-------------------------------------------------------------------------------------------------------------------------------

-Changed the colors and the sizes of the displayed text, and table.

-------------------------------------------------------------------------------------------------------------------------------

-Finished Final Touches and styling

-------------------------------------------------------------------------------------------------------------------------------

-Added curved edges on the boxes

-------------------------------------------------------------------------------------------------------------------------------

-Moved the buttons to better match the provided examples

-------------------------------------------------------------------------------------------------------------------------------

-Changed the color when you hover over the row of a table to be lighter than the standard color

-------------------------------------------------------------------------------------------------------------------------------

**$Finished Project**

-------------------------------------------------------------------------------------------------------------------------------

**Final Notes:**

This project was successful in its attempt to query data and display it into a html table onto the screen. Then it was successful in displaying a design similar to the one provided by Matthew Picket and contains the same functionality as the provided design. This project took 3days to complete with 3 days of researching JavaScript, CSS, and HTML. The most difficult part of this project was getting the formatting correct using CSS.

-------------------------------------------------------------------------------------------------------------------------------

**References:**

**Helped Overall:**

<https://stackoverflow.com/> **-** Life saver

<https://www.w3schools.com/> - Main resource used

**JavaScript:**

<https://www.w3schools.com/js/js_htmldom_document.asp> **-** Used for creating the table and assigning the tags to variables

<https://www.w3schools.com/js/js_functions.asp> - Helped to understand the syntax of the functions in JavaScript

<https://www.w3schools.com/jsref/jsref_filter.asp> - Helped with understanding filtering so the table 3 could be implemented

<https://www.w3schools.com/jsref/met_node_appendchild.asp> - Helped with the append child and being able to append data to the table.

**HTML:**

<https://www.w3schools.com/TAGs/> **-**Specifically the ‘th’,’td’,and ‘tr’ tags

<https://www.w3schools.com/htmL/html_id.asp> - Needed for creating id’s for the javascript and css to manipulate

**CSS:**

<https://www.w3schools.com/css/css_selectors.asp> **-** Backbone of the styling of this project

<https://www.w3schools.com/css/css_boxmodel.asp> - More insight on boxes

<https://www.w3.org/TR/CSS2/generate.html#before-after-content> – Used for learning how to insert text in the margin of a box

<https://dev.to/temmietope/rounded-edges-on-table-rows-1d0n> - Helped to understand how to make the rows rounded

<https://www.w3schools.com/cssref/pr_class_position.php> - Helped with understanding how positioning works

<https://www.w3schools.com/cssref/pr_border-collapse.php> - Helped to understand how Border-Collapse works and can be implemented in the table

<https://www.w3schools.com/CSSref/sel_hover.php> - Helped to understand how Hover works