[Code:](https://soko7awen.github.io/n220LabS23/Labs/tictactoe/index.html)

https://soko7awen.github.io/n220LabS23/Labs/tictactoe/index.html

Repository:

<https://github.com/soko7awen/n220LabS23/tree/main/Labs/tictactoe>

Resources:  
https://stackoverflow.com/questions/4649699/is-it-possible-to-get-elements-numerical-index-in-its-parent-node-without-loopi

- I used this stack overflow question to find out how to get the index of an element within the array of its parent’s children.

https://www.w3schools.com/howto/howto\_js\_rangeslider.asp

- I used this to remind myself how the html slider works.

Design:

This program is intended to be a recreation of the classic game tictactoe. It should alternate between the first players placement of their corresponding letter and the second players placement of their corresponding letter. When three of the same letter are in a row, the game will stop and that player will be declared the winner. If all of the spaces are filled and there is no winner, then the game will be declared a draw. I have also chosen to add in customization in the form of letter coloring.

Design Notes:

I worked on this project for two very long sessions and a final session for polish. The first session I spent doing all of the background logic, which I could then feed it into the HTML. One problem I had for this session was that I wanted to integrate the diagonal checks into the checks of the rows and columns. I was able to solve this by checking one square of the diagonal for every row/column.

The second session I created the html. I had to integrate my background logic into the front-end. Doing this involved first creating the requisite elements and matching them to their respective places on the matrix. I also had to add the text for who the current player is, as well the text for when a win/draw is reached.

The last session I added some polish, this was mostly just prettying up the css and adding a color feature.