PROMINEO TECH

Front End Technologies Week 11 Coding Assignment

Points possible: 70

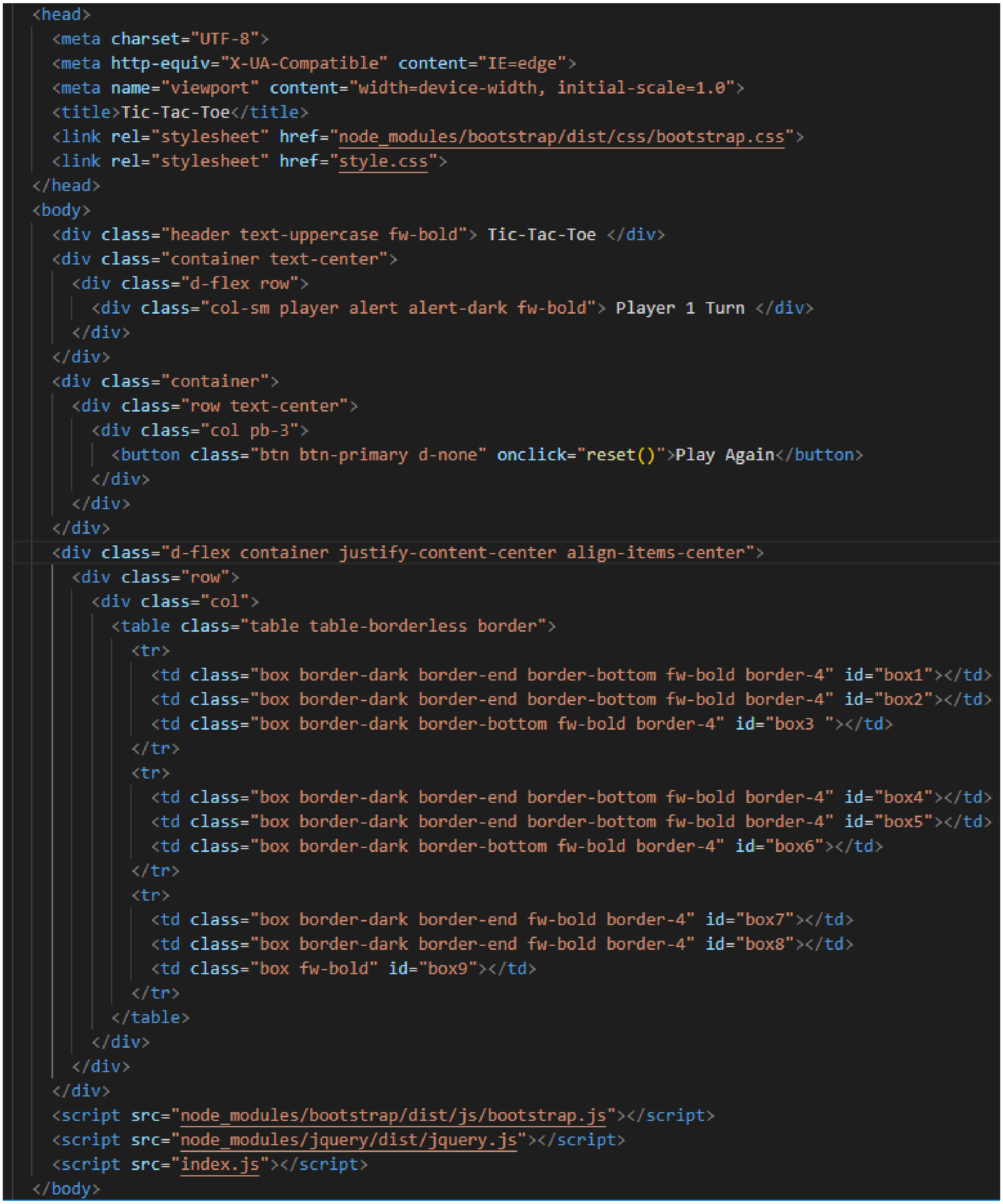
Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

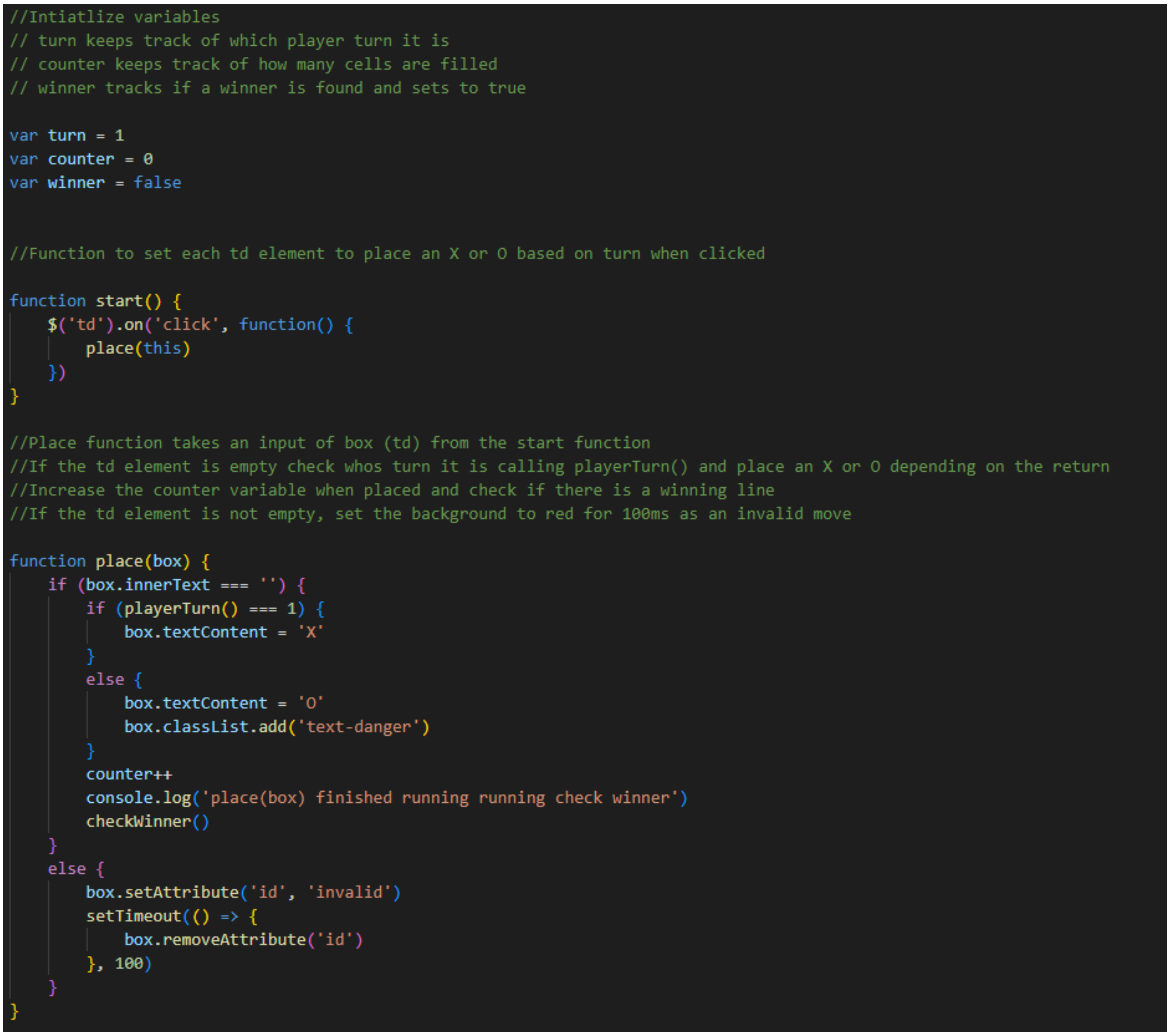
Coding Steps:

1. Using any of the tools you've worked with so far, create a game of tic-tac-toe.
   - **a.** A heading should say whether it is X's or O's turn and change with each move made.
   - **b.** Create a tic-tac-toe grid using your HTML element of choice. When a cell in the grid is clicked, an X or O should appear in that spot depending on whose turn it is.
   - **c.** A button should be available to clear the grid and restart the game.
   - **d.** When a player has won, or the board is full and the game results in a draw, a Bootstrap alert or similar Bootstrap component should appear across the screen announcing the winner.

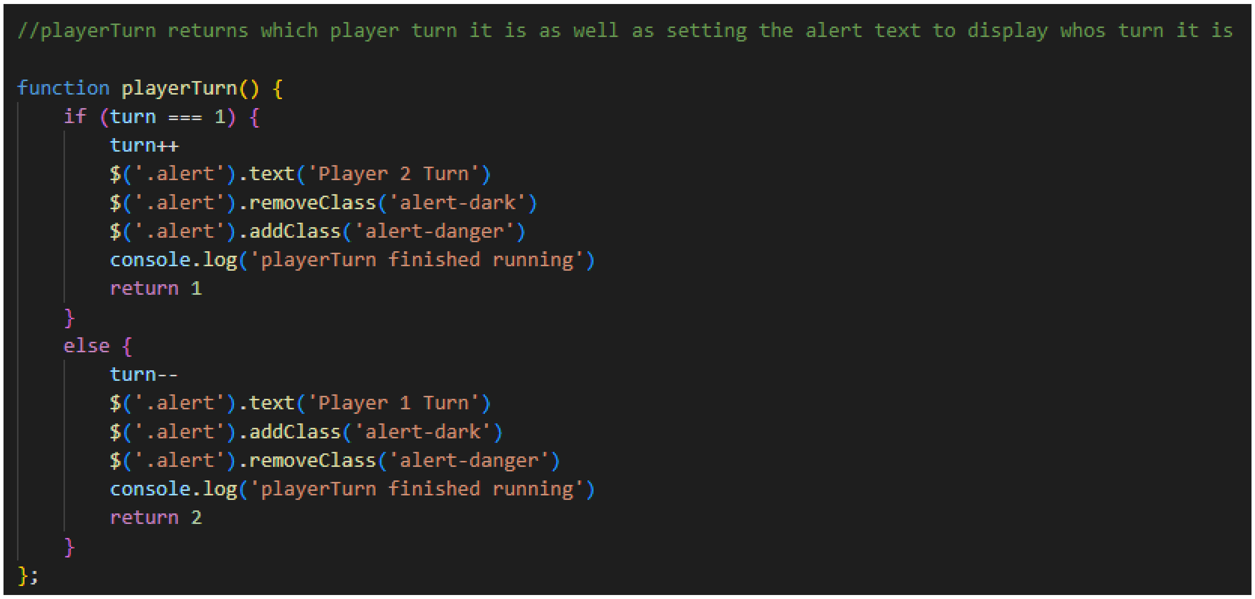
Screenshots of Code:



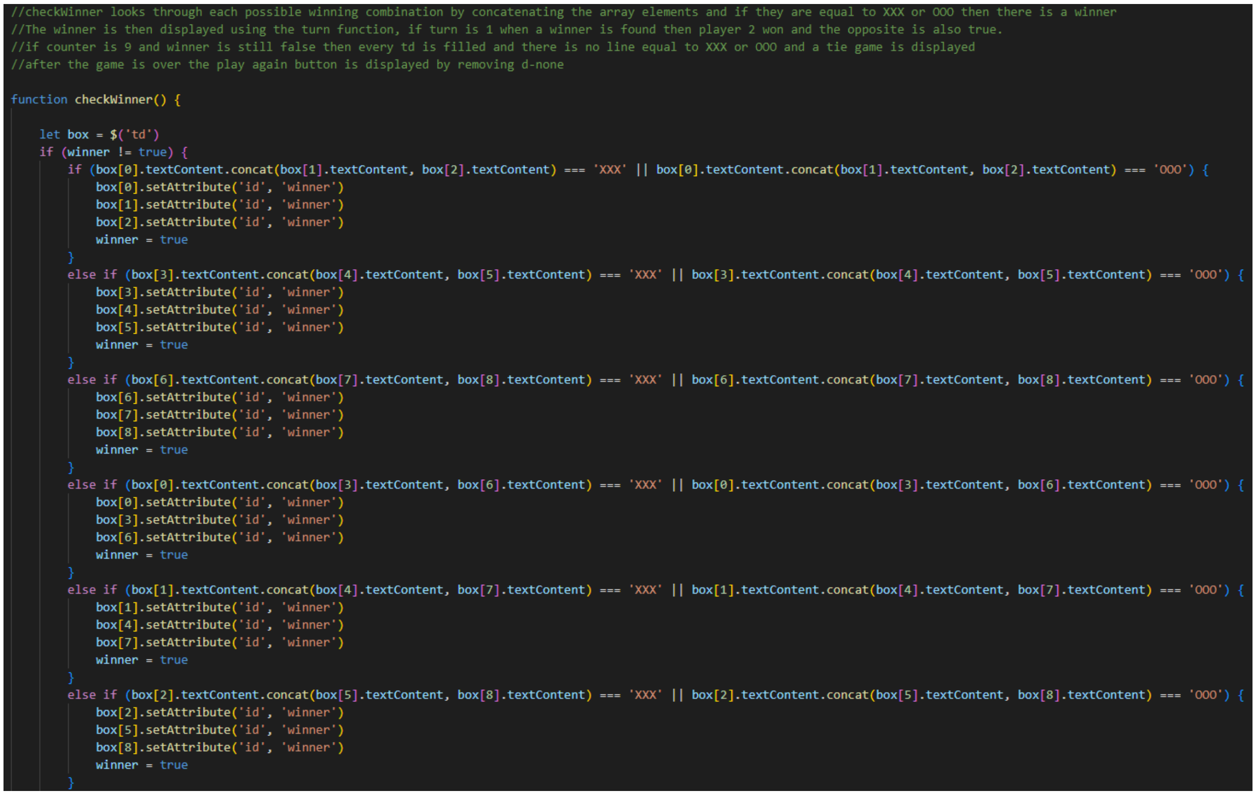
```html
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tic-Tac-Toe</title>
  <link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.css">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="header text-uppercase fw-bold"> Tic-Tac-Toe </div>
  <div class="container text-center">
    <div class="d-flex row">
      <div class="col-sm player alert alert-dark fw-bold"> Player 1 Turn </div>
    </div>
  </div>
  <div class="container">
    <div class="row text-center">
      <div class="col pb-3">
        <button class="btn btn-primary d-none" onclick="reset()">Play Again</button>
      </div>
    </div>
  </div>
  <div class="d-flex container justify-content-center align-items-center">
    <div class="row">
      <div class="col">
        <table class="table table-borderless border">
          <tr>
            <td class="box border-dark border-end border-bottom fw-bold border-4" id="box1"></td>
            <td class="box border-dark border-end border-bottom fw-bold border-4" id="box2"></td>
            <td class="box border-dark border-bottom fw-bold border-4" id="box3 "></td>
          </tr>
          <tr>
            <td class="box border-dark border-end border-bottom fw-bold border-4" id="box4"></td>
            <td class="box border-dark border-end border-bottom fw-bold border-4" id="box5"></td>
            <td class="box border-dark border-bottom fw-bold border-4" id="box6"></td>
          </tr>
          <tr>
            <td class="box border-dark border-end fw-bold border-4" id="box7"></td>
            <td class="box border-dark border-end fw-bold border-4" id="box8"></td>
            <td class="box fw-bold" id="box9"></td>
          </tr>
        </table>
      </div>
    </div>
  </div>
  <script src="node_modules/bootstrap/dist/js/bootstrap.js"></script>
  <script src="node_modules/jquery/dist/jquery.js"></script>
  <script src="index.js"></script>
</body>
```



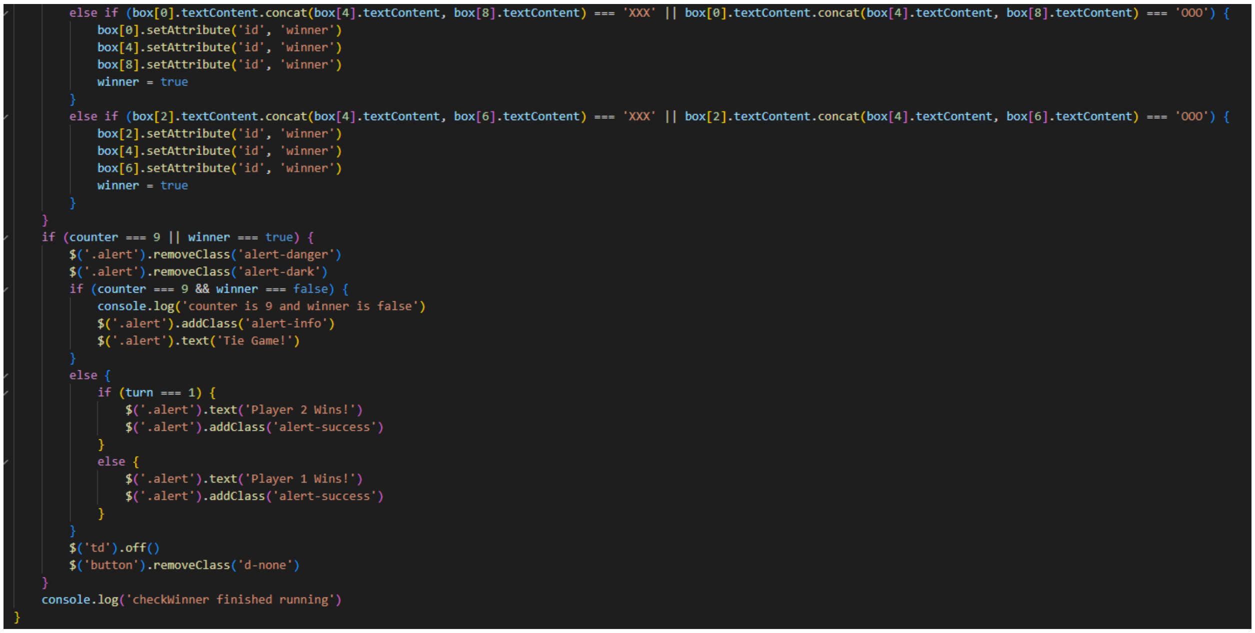
```
//Intiatlize variables
// turn keeps track of which player turn it is
// counter keeps track of how many cells are filled
// winner tracks if a winner is found and sets to true

var turn = 1
var counter = 0
var winner = false


//Function to set each td element to place an X or O based on turn when clicked

function start() {
    $('td').on('click', function() {
        place(this)
    })
}

//Place function takes an input of box (td) from the start function
//If the td element is empty check whos turn it is calling playerTurn() and place an X or O depending on the return
//Increase the counter variable when placed and check if there is a winning line
//If the td element is not empty, set the background to red for 100ms as an invalid move

function place(box) {
    if (box.innerText === '') {
        if (playerTurn() === 1) {
            box.textContent = 'X'
        }
        else {
            box.textContent = 'O'
            box.classList.add('text-danger')
        }
        counter++
        console.log('place(box) finished running running check winner')
        checkWinner()
    }
    else {
        box.setAttribute('id', 'invalid')
        setTimeout(() => {
            box.removeAttribute('id')
        }, 100)
    }
}
```



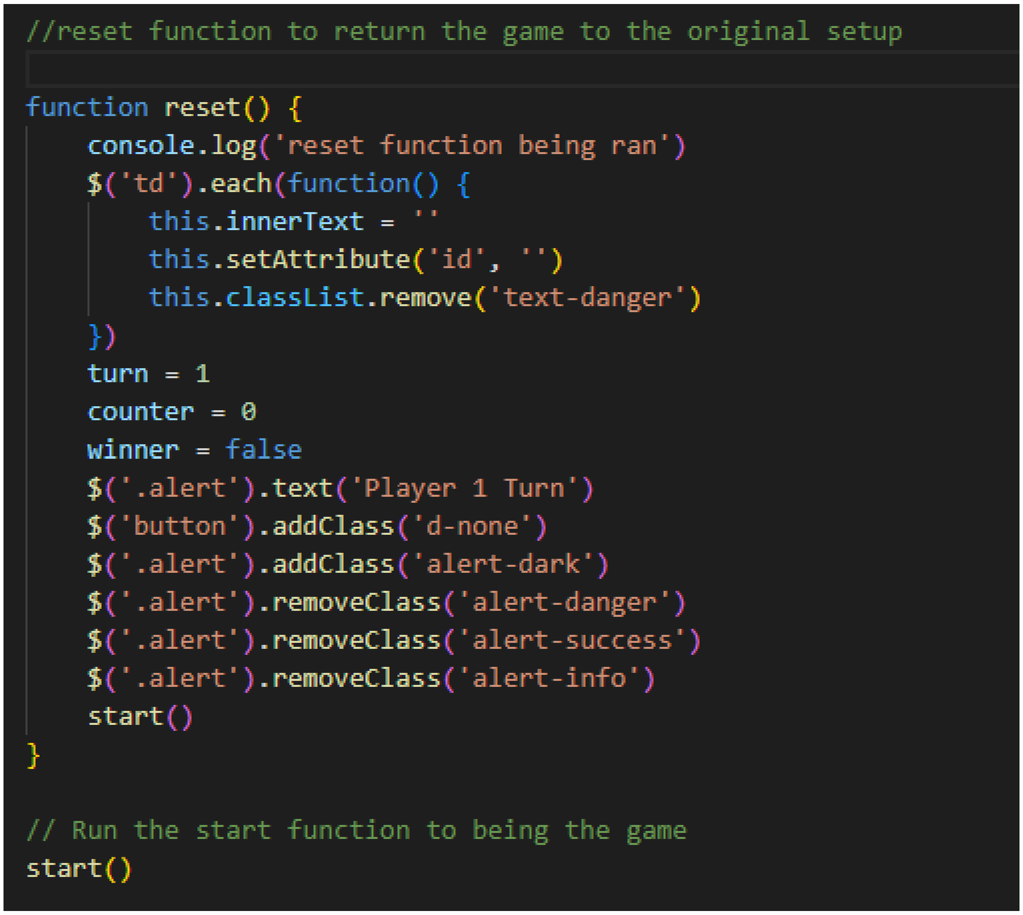
```
//playerTurn returns which player turn it is as well as setting the alert text to display whos turn it is

function playerTurn() {
    if (turn === 1) {
        turn++
        $('.alert').text('Player 2 Turn')
        $('.alert').removeClass('alert-dark')
        $('.alert').addClass('alert-danger')
        console.log('playerTurn finished running')
        return 1
    }
    else {
        turn--
        $('.alert').text('Player 1 Turn')
        $('.alert').addClass('alert-dark')
        $('.alert').removeClass('alert-danger')
        console.log('playerTurn finished running')
        return 2
    }
};
```



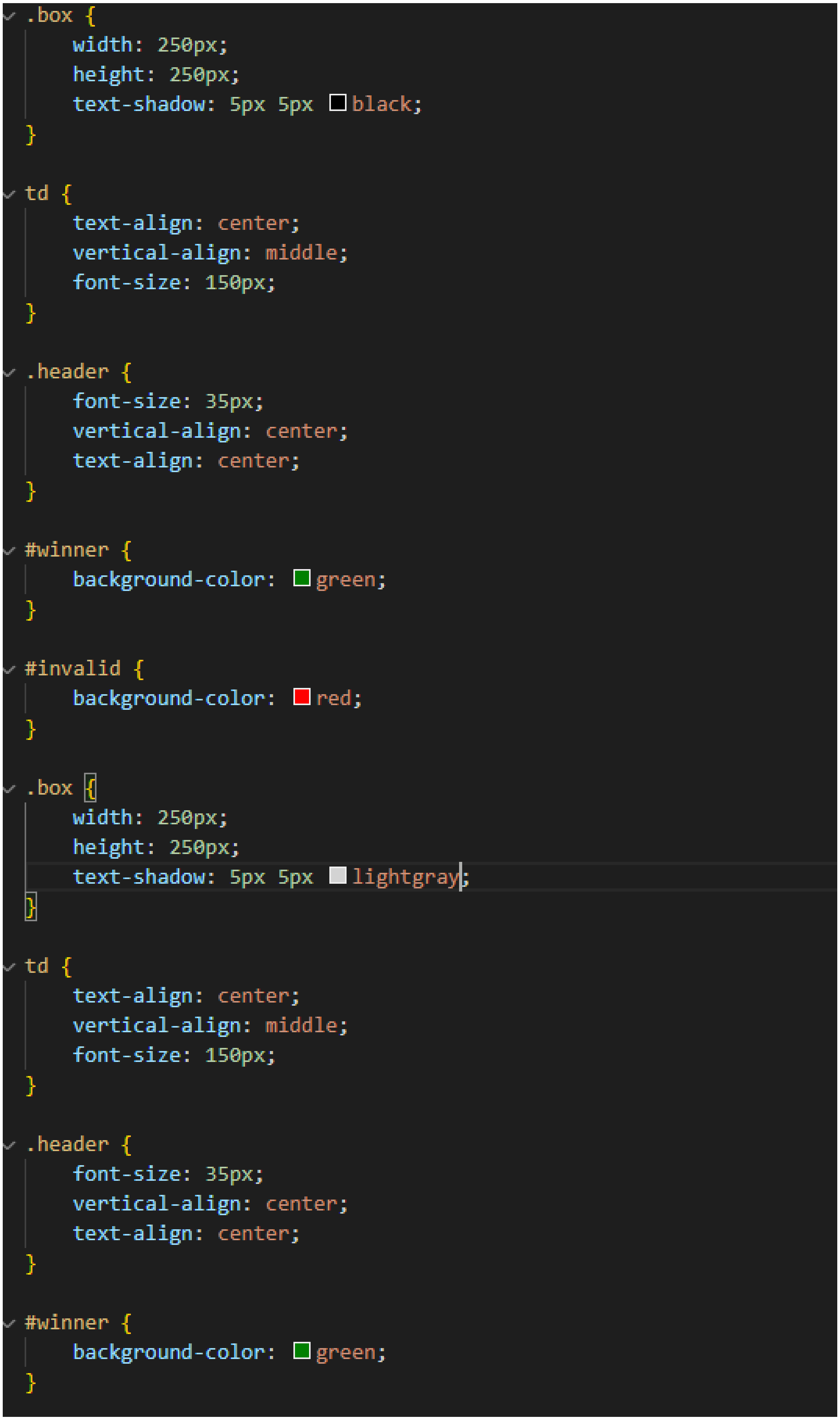
```
//checkWinner looks through each possible winning combination by concatenating the array elements and if they are equal to XXX or OOO then there is a winner
//The winner is then displayed using the turn function, if turn is 1 when a winner is found then player 2 won and the opposite is also true.
//if counter is 9 and winner is still false then every td is filled and there is no line equal to XXX or OOO and a tie game is displayed
//after the game is over the play again button is displayed by removing d-none

function checkWinner() {

    let box = $('td')
    if (winner != true) {
        if (box[0].textContent.concat(box[1].textContent, box[2].textContent) === 'XXX' || box[0].textContent.concat(box[1].textContent, box[2].textContent) === 'OOO') {
            box[0].setAttribute('id', 'winner')
            box[1].setAttribute('id', 'winner')
            box[2].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[3].textContent.concat(box[4].textContent, box[5].textContent) === 'XXX' || box[3].textContent.concat(box[4].textContent, box[5].textContent) === 'OOO') {
            box[3].setAttribute('id', 'winner')
            box[4].setAttribute('id', 'winner')
            box[5].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[6].textContent.concat(box[7].textContent, box[8].textContent) === 'XXX' || box[6].textContent.concat(box[7].textContent, box[8].textContent) === 'OOO') {
            box[6].setAttribute('id', 'winner')
            box[7].setAttribute('id', 'winner')
            box[8].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[0].textContent.concat(box[3].textContent, box[6].textContent) === 'XXX' || box[0].textContent.concat(box[3].textContent, box[6].textContent) === 'OOO') {
            box[0].setAttribute('id', 'winner')
            box[3].setAttribute('id', 'winner')
            box[6].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[1].textContent.concat(box[4].textContent, box[7].textContent) === 'XXX' || box[1].textContent.concat(box[4].textContent, box[7].textContent) === 'OOO') {
            box[1].setAttribute('id', 'winner')
            box[4].setAttribute('id', 'winner')
            box[7].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[2].textContent.concat(box[5].textContent, box[8].textContent) === 'XXX' || box[2].textContent.concat(box[5].textContent, box[8].textContent) === 'OOO') {
            box[2].setAttribute('id', 'winner')
            box[5].setAttribute('id', 'winner')
            box[8].setAttribute('id', 'winner')
            winner = true
        }
```



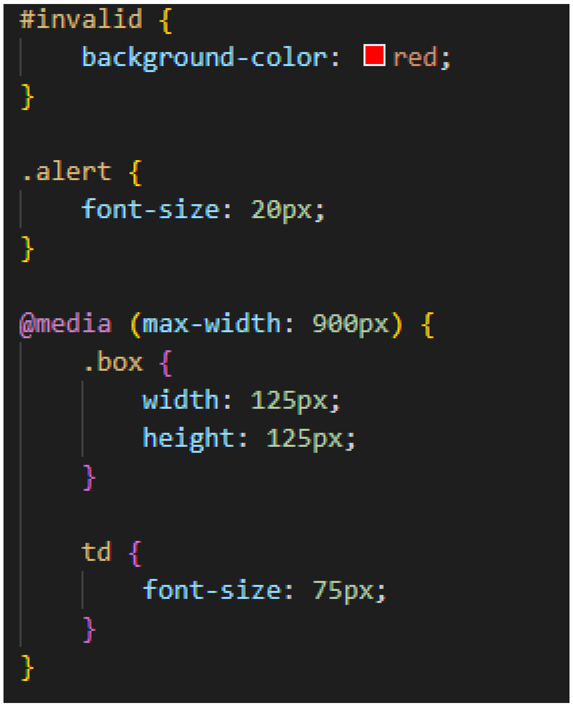
```
        else if (box[0].textContent.concat(box[4].textContent, box[8].textContent) === 'XXX' || box[0].textContent.concat(box[4].textContent, box[8].textContent) === 'OOO') {
            box[0].setAttribute('id', 'winner')
            box[4].setAttribute('id', 'winner')
            box[8].setAttribute('id', 'winner')
            winner = true
        }
        else if (box[2].textContent.concat(box[4].textContent, box[6].textContent) === 'XXX' || box[2].textContent.concat(box[4].textContent, box[6].textContent) === 'OOO') {
            box[2].setAttribute('id', 'winner')
            box[4].setAttribute('id', 'winner')
            box[6].setAttribute('id', 'winner')
            winner = true
        }
    }
    if (counter === 9 || winner === true) {
        $('.alert').removeClass('alert-danger')
        $('.alert').removeClass('alert-dark')
        if (counter === 9 && winner === false) {
            console.log('counter is 9 and winner is false')
            $('.alert').addClass('alert-info')
            $('.alert').text('Tie Game!')
        }
        else {
            if (turn === 1) {
                $('.alert').text('Player 2 Wins!')
                $('.alert').addClass('alert-success')
            }
            else {
                $('.alert').text('Player 1 Wins!')
                $('.alert').addClass('alert-success')
            }
        }
        $('td').off()
        $('button').removeClass('d-none')
    }
    console.log('checkWinner finished running')
}
```



```
//reset function to return the game to the original setup

function reset() {
    console.log('reset function being ran')
    $('td').each(function() {
        this.innerText = ''
        this.setAttribute('id', '')
        this.classList.remove('text-danger')
    })
    turn = 1
    counter = 0
    winner = false
    $('.alert').text('Player 1 Turn')
    $('button').addClass('d-none')
    $('.alert').addClass('alert-dark')
    $('.alert').removeClass('alert-danger')
    $('.alert').removeClass('alert-success')
    $('.alert').removeClass('alert-info')
    start()
}

// Run the start function to being the game
start()
```

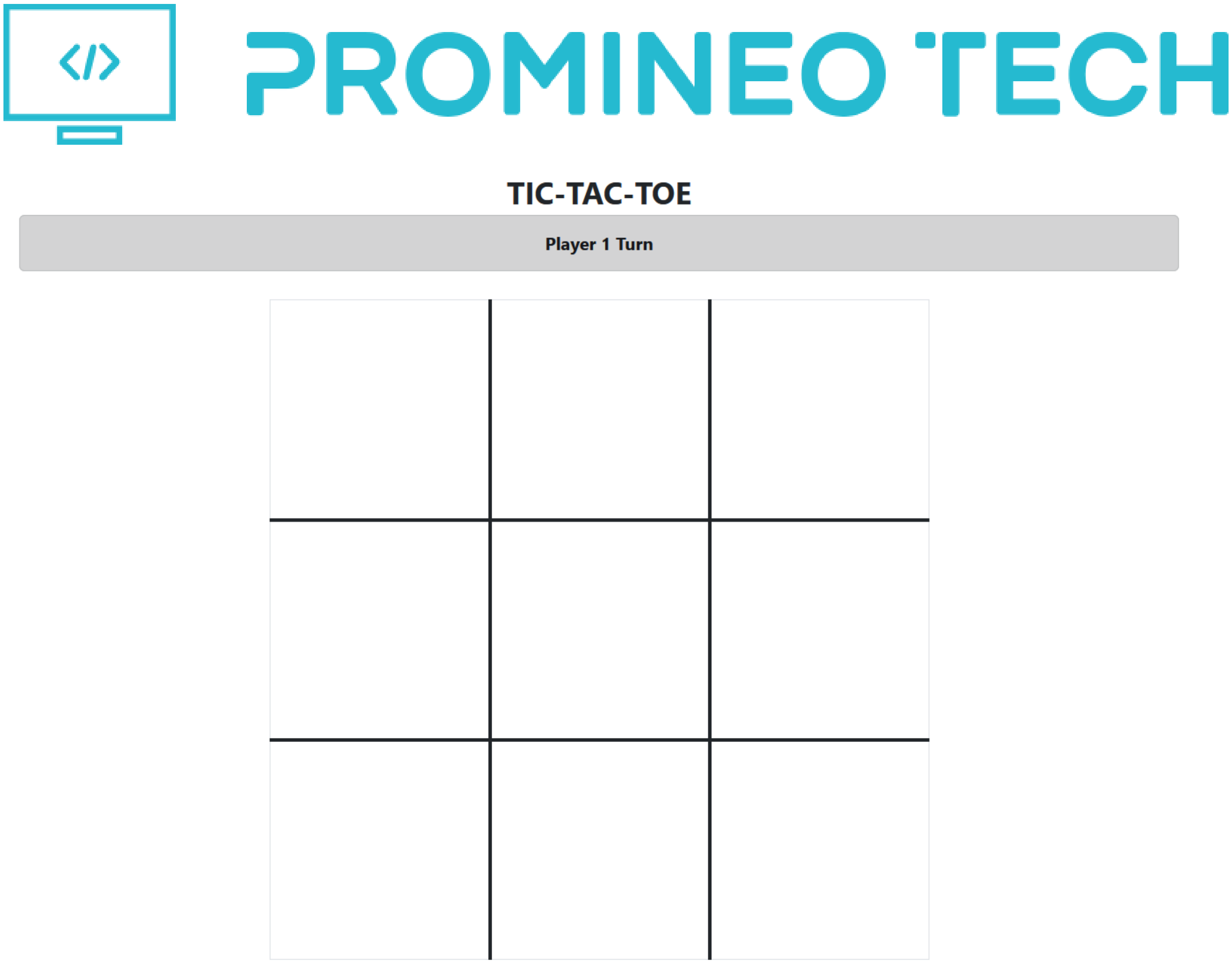


```css
.box {
    width: 250px;
    height: 250px;
    text-shadow: 5px 5px black;
}

td {
    text-align: center;
    vertical-align: middle;
    font-size: 150px;
}

.header {
    font-size: 35px;
    vertical-align: center;
    text-align: center;
}

#winner {
    background-color: green;
}

#invalid {
    background-color: red;
}

.box {
    width: 250px;
    height: 250px;
    text-shadow: 5px 5px lightgray;
}

td {
    text-align: center;
    vertical-align: middle;
    font-size: 150px;
}

.header {
    font-size: 35px;
    vertical-align: center;
    text-align: center;
}

#winner {
    background-color: green;
}
```



```css
#invalid {
    background-color: 🟥red;
}

.alert {
    font-size: 20px;
}

@media (max-width: 900px) {
    .box {
        width: 125px;
        height: 125px;
    }

    td {
        font-size: 75px;
    }
}
```

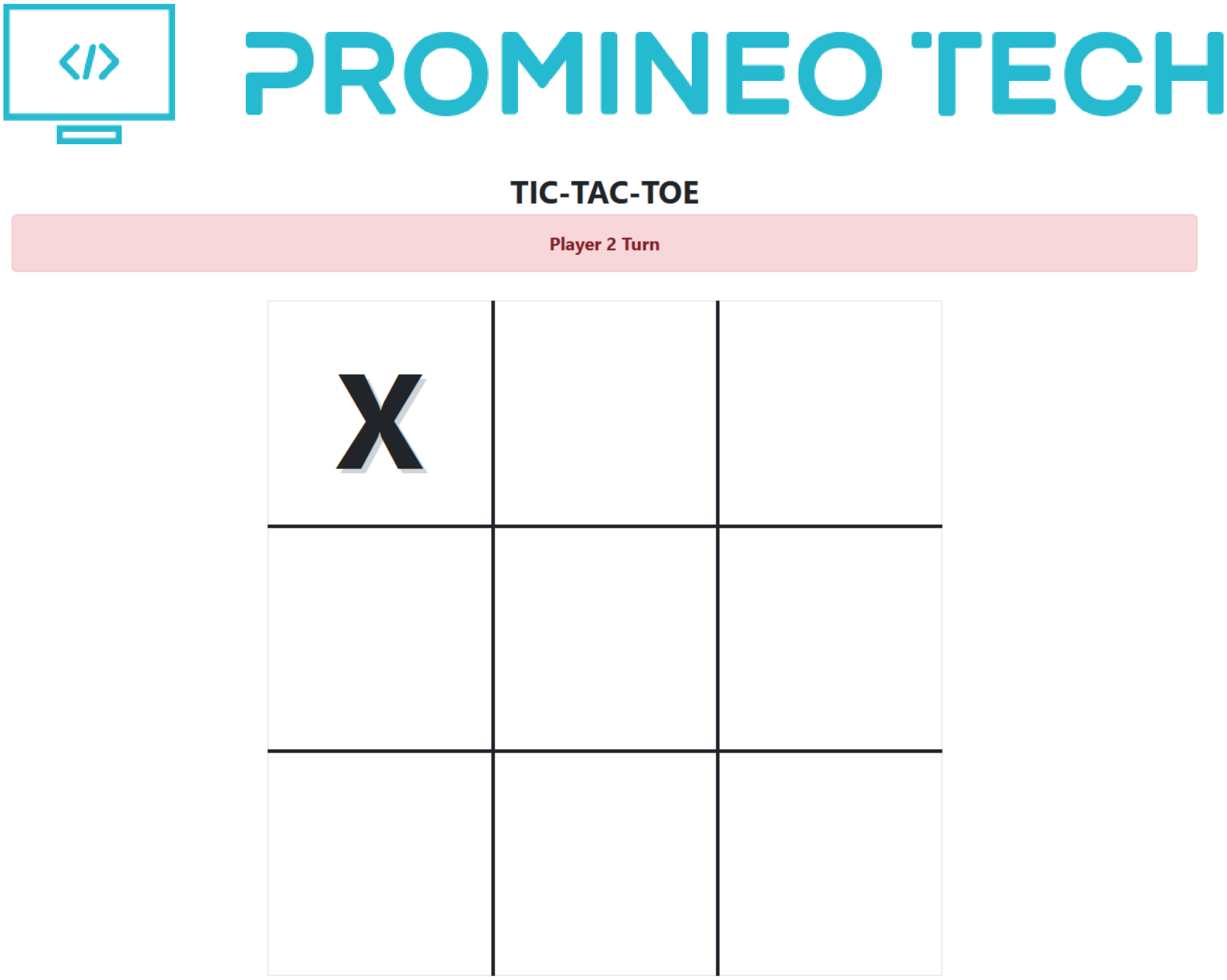
Screenshots of Running Application:



PROMINEO TECH

TIC-TAC-TOE

Player 1 Turn

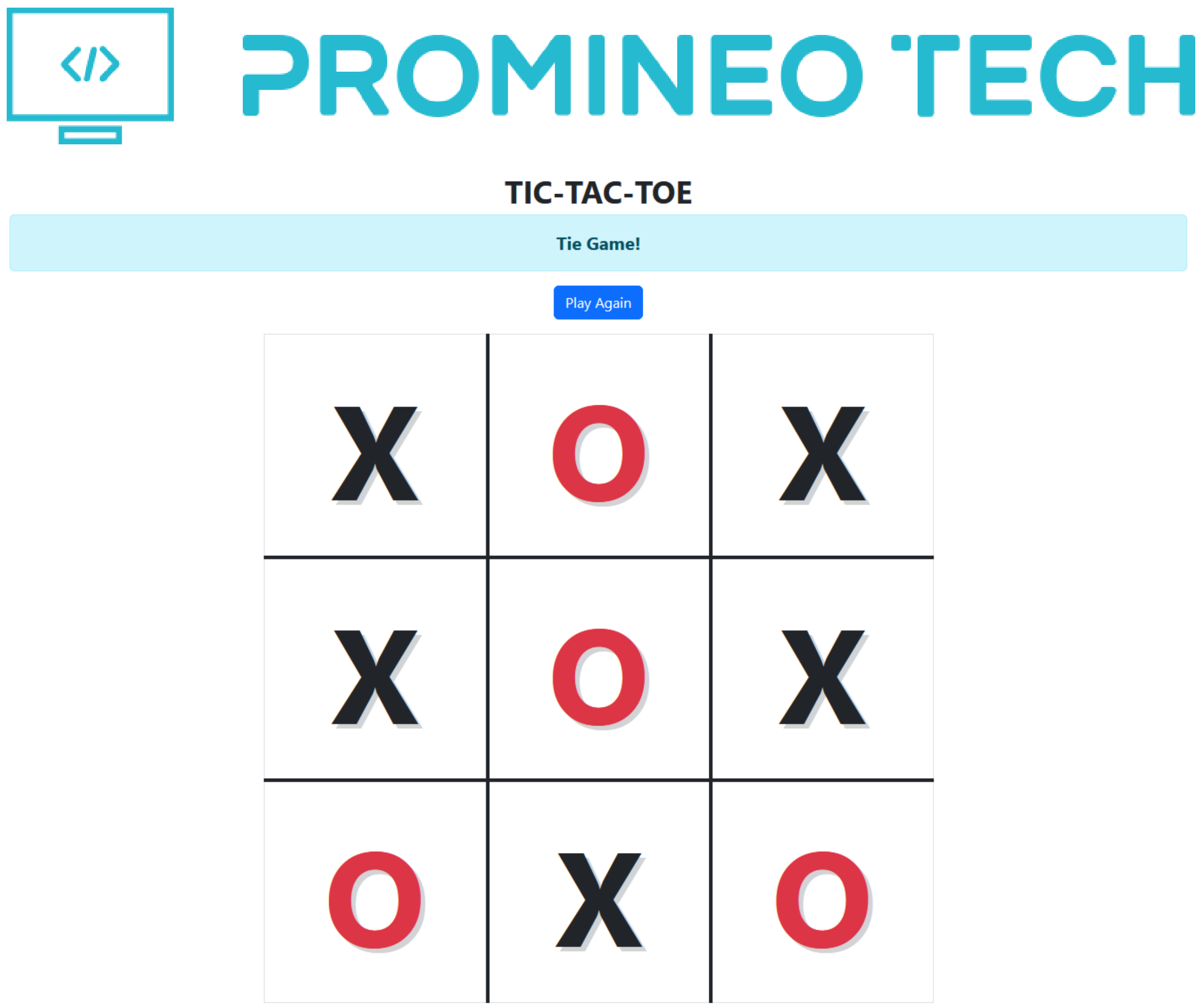


PROMINEO TECH

TIC-TAC-TOE

Player 2 Turn

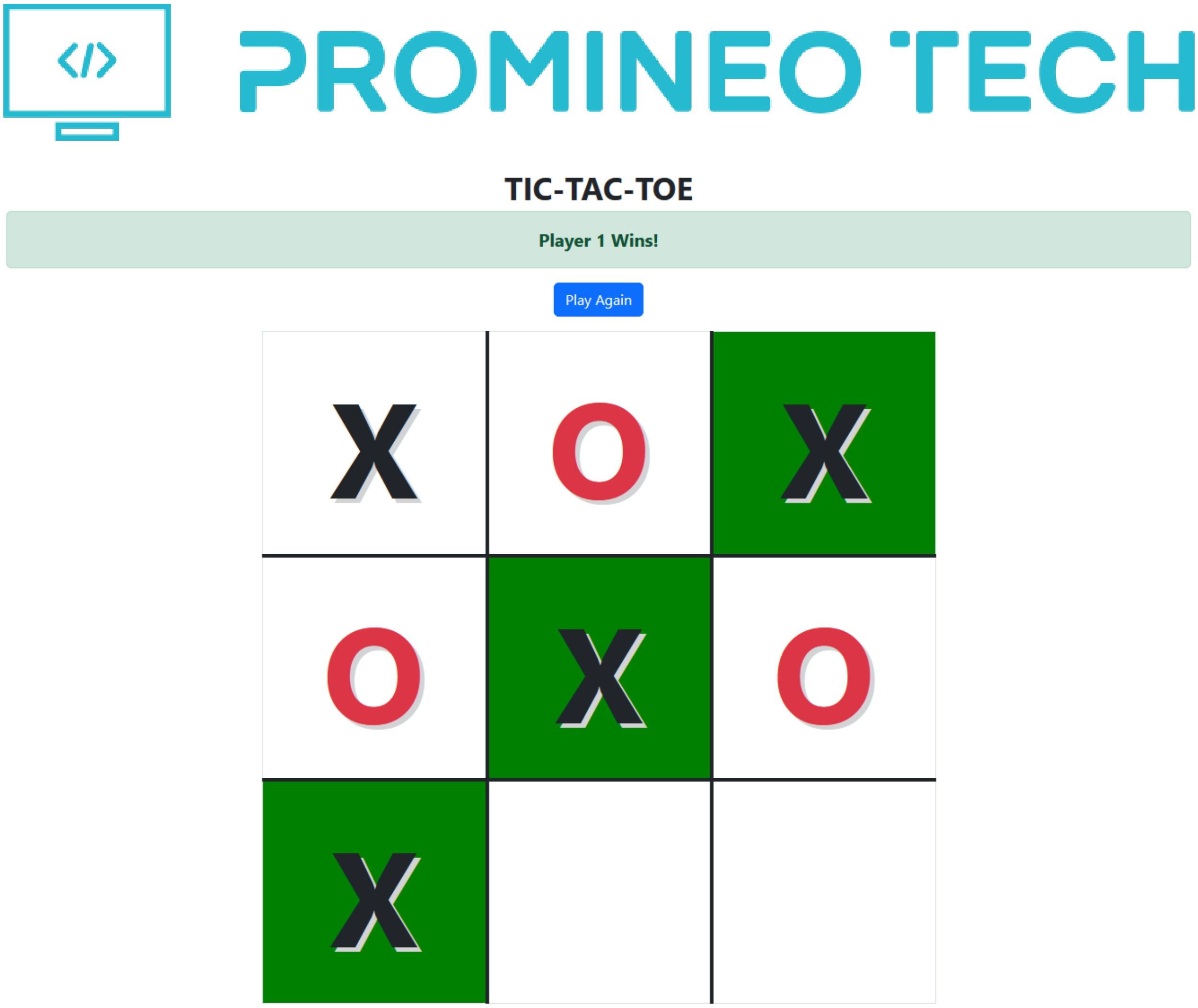
X		



TIC-TAC-TOE

Tie Game!

X	O	X
X	O	X
O	X	O

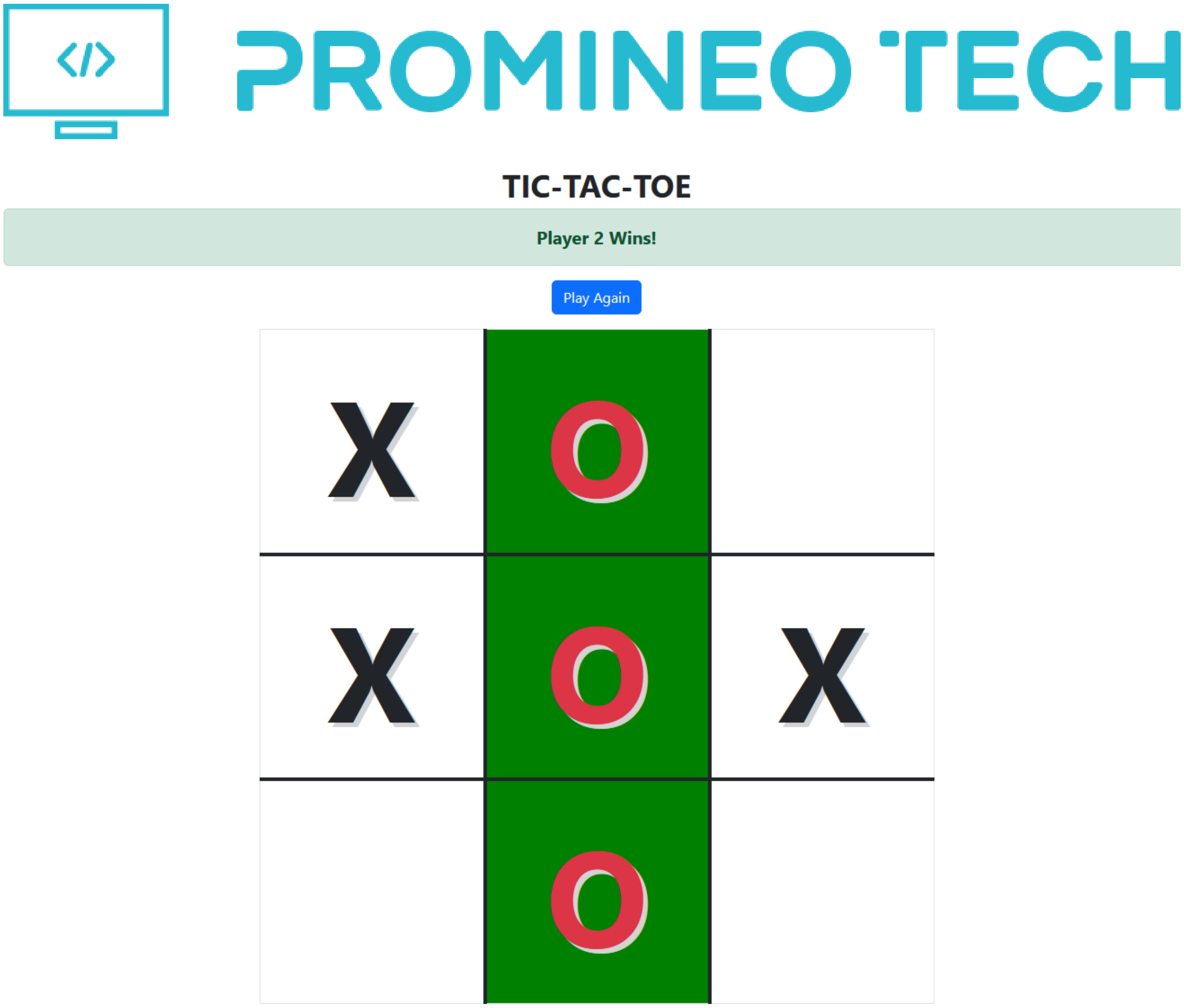


PROMINEO TECH

TIC-TAC-TOE

Player 1 Wins!

X	O	X
O	X	O
X		



TIC-TAC-TOE

Player 2 Wins!

X	O	
X	O	X
	O	

URL to GitHub Repository: https://github.com/n-harvey/Week11CodingAssignment