



Front End Technologies Week 5 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:



PROMINEO TECH

```
<> index.html U X # styles.css U JS App.js U
<> index.html > html > body > main.background > section.container > div.box
1  <html lang="en">
2  <head>
3    <meta charset="UTF-8">
4    <meta http-equiv="X-UA-Compatible" content="IE=edge">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <link rel="stylesheet" href="styles.css">
7    <link rel="preconnect" href="https://fonts.gstatic.com">
8    <title>Tic-Tac-Toe</title>
9  </head>
10 <body>
11   <main class="background">
12     <section class="title">
13       <h1>Tic Tac Toe</h1>
14     </section>
15     <section class="display">
16       Player <span class="display-player playerX">X</span>'s turn
17     </section>
18     <section class="container">
19       <div class="box"></div>
20       <div class="box"></div>
21       <div class="box"></div>
22       <div class="box"></div>
23       <div class="box"></div>
24       <div class="box"></div>
25       <div class="box"></div>
26       <div class="box"></div>
27       <div class="box"></div>
28     </section>
29
30
31     <section class="display announcer hide"></section>
32     <section class="controls">
33       <button id="reset">Restart</button>
34     </section>
35   </main>
36
37   <script src="App.js"></script>
38 </body>
39 </html>
```



PROMINEO TECH

```
<> index.html U # styles.css U X JS App.js U
# styles.css > .container
1 * {
2   padding: 0;
3   margin: 0;
4 }
5
6 .background {
7   background-color: #af7ace;
8   height: 100vh;
9   padding-top: 1px;
10 }
11
12 .title {
13   color: #9b077be3;
14   text-align: center;
15   font-size: 75px;
16   /* margin-top: 5%; */
17   margin-bottom: 50px;
18 }
19
20
21 .display {
22   color: rgb(66, 5, 71);
23   font-size: 60px;
24   text-align: center;
25   margin-top: 1em;
26   margin-bottom: 1em;
27 }
28
29 .hide {
30   display: none;
31 }
32
33 .container {
34   margin: 0 auto;
35   display: grid;
36   grid-template-columns: auto auto auto;
37   grid-template-rows: auto auto auto;
38   max-width: 350px;
39 }
40
41
```



PROMINEO TECH

```
<> index.html U # styles.css U X JS App.js U
# styles.css > .container
42 .box {
43     border: 5px solid rgb(67, 4, 109);
44     min-width: 100px;
45     min-height: 100px;
46     display: flex;
47     justify-content: center;
48     align-items: center;
49     font-size: 80px;
50     cursor: pointer;
51 }
52
53 .playerX {
54     color: #822dbb;
55 }
56
57 .player0 {
58     color: #9b077be3;
59 }
60
61 .controls {
62     display: flex;
63     flex-direction: row;
64     justify-content: center;
65     align-items: center;
66     margin-top: 1em;
67 }
68
69 .controls button {
70     color: white;
71     font-weight: 700;
72     padding: 16px;
73     border-radius: 15px;
74     border: none;
75     font-size: 30px;
76     margin-left: 1em;
77     cursor: pointer;
78 }
79
80 .restart {
81     background-color: #498AFB;
82 }
```

```
83
84 #reset {
85     background-color: #b154d6;
86     border: #822dbb 3px solid;
87     width: 300px;
88     margin-top: 1%;
89 }
90
91
92
```



PROMINEO TECH

```
< index.html U # styles.css U JS App.js U X
JS App.js > window.addEventListener('DOMContentLoaded') callback
1 window.addEventListener('DOMContentLoaded', () => {
2   const boxes = Array.from(document.querySelectorAll('.box'));
3   const playerDisplay = document.querySelector('.display-player');
4   const resetButton = document.querySelector('#reset');
5   const announcer = document.querySelector('.announcer');
6
7   let board = ['', '', '', '', '', '', '', '', ''];
8   let currentPlayer = 'X';
9   let isGameActive = true;
10
11   const PLAYERX_WON = 'PLAYERX_WON';
12   const PLAYERO_WON = 'PLAYERO_WON';
13   const TIE = 'TIE';
14   /*
15    * Indexes within the board
16    * [0] [1] [2]
17    * [3] [4] [5]
18    * [6] [7] [8]
19    */
20
21   const winningConditions = [
22     [0, 1, 2],
23     [3, 4, 5],
24     [6, 7, 8],
25     [0, 3, 6],
26     [1, 4, 7],
27     [2, 5, 8],
28     [0, 4, 8],
29     [2, 4, 6]
30   ];
31
32   function handleResultValidation() {
33     let roundWon = false;
34     for (let i = 0; i <= 7; i++) {
35       const winCondition = winningConditions[i];
36       const a = board[winCondition[0]];
37       const b = board[winCondition[1]];
38       const c = board[winCondition[2]];
39       if (a === '' || b === '' || c === '') {
40         continue;
41       }
42       if (a === b && b === c) {
43         roundWon = true;
44         break;
45       }
46     }
47   }
48 }
```



PROMINEO TECH

```
index.html U # styles.css U JS App.js U X
JS App.js > window.addEventListener('DOMContentLoaded') callback
47
48   if (roundWon) {
49       announce(currentPlayer === 'X' ? PLAYERX_WON : PLAYERO_WON);
50       isGameActive = false;
51       return;
52   }
53
54   if (!board.includes(''))
55       announce(TIE);
56   }
57
58   const announce = (type) => {
59       switch(type){
60           case PLAYERO_WON:
61               announcer.innerHTML = 'Player <span class="player0">1</span> Won !!!';
62               break;
63           case PLAYERX_WON:
64               announcer.innerHTML = 'Player <span class="playerX">2</span> Won !!!';
65               break;
66           case TIE:
67               announcer.innerText = 'It\'s a Tie!!';
68       }
69       announcer.classList.remove('hide');
70   };
71
72   const isValidAction = (box) => {
73       if (box.innerText === 'X' || box.innerText === 'O'){
74           return false;
75       }
76
77       return true;
78   };
79
80   const updateBoard = (index) => {
81       board[index] = currentPlayer;
82   }
83
84   const changePlayer = () => {
85       playerDisplay.classList.remove(`player${currentPlayer}`);
86       currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
87       playerDisplay.innerText = currentPlayer === 'X' ? '2' : '1';
88       playerDisplay.classList.add(`player${currentPlayer}`);
89   }
90
```



PROMINEO TECH

```
91     const userAction = (box, index) => {
92       if(isValidAction(box) && isGameActive) {
93         box.innerText = currentPlayer;
94         box.classList.add(`player${currentPlayer}`);
95         updateBoard(index);
96         handleResultValidation();
97         changePlayer();
98       }
99     }
100
101     const resetBoard = () => {
102       board = ['', '', '', '', '', '', '', '', ''];
103       isGameActive = true;
104       announcer.classList.add('hide');
105
106       if (currentPlayer === '0') {
107         changePlayer();
108       }
109
110       boxes.forEach(box => {
111         box.innerText = '';
112         box.classList.remove('playerX');
113         box.classList.remove('player0');
114       });
115     }
116
117     boxes.forEach( (box, index) => {
118       box.addEventListener('click', () => userAction(box, index));
119     });
120
121     resetButton.addEventListener('click', resetBoard);
122   });
```

Screenshots of Running Application:

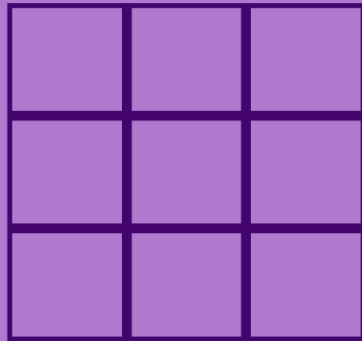
https://github.com/JoanaBarao7/FET_Week5



PROMINEO TECH

Tic Tac Toe

Player X's turn



Restart



PROMINEO TECH

Tic Tac Toe

Player 2's turn

X	O	X
O	O	X
X	O	

Player 1 Won !!!

Restart

URL to GitHub Repository: