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.
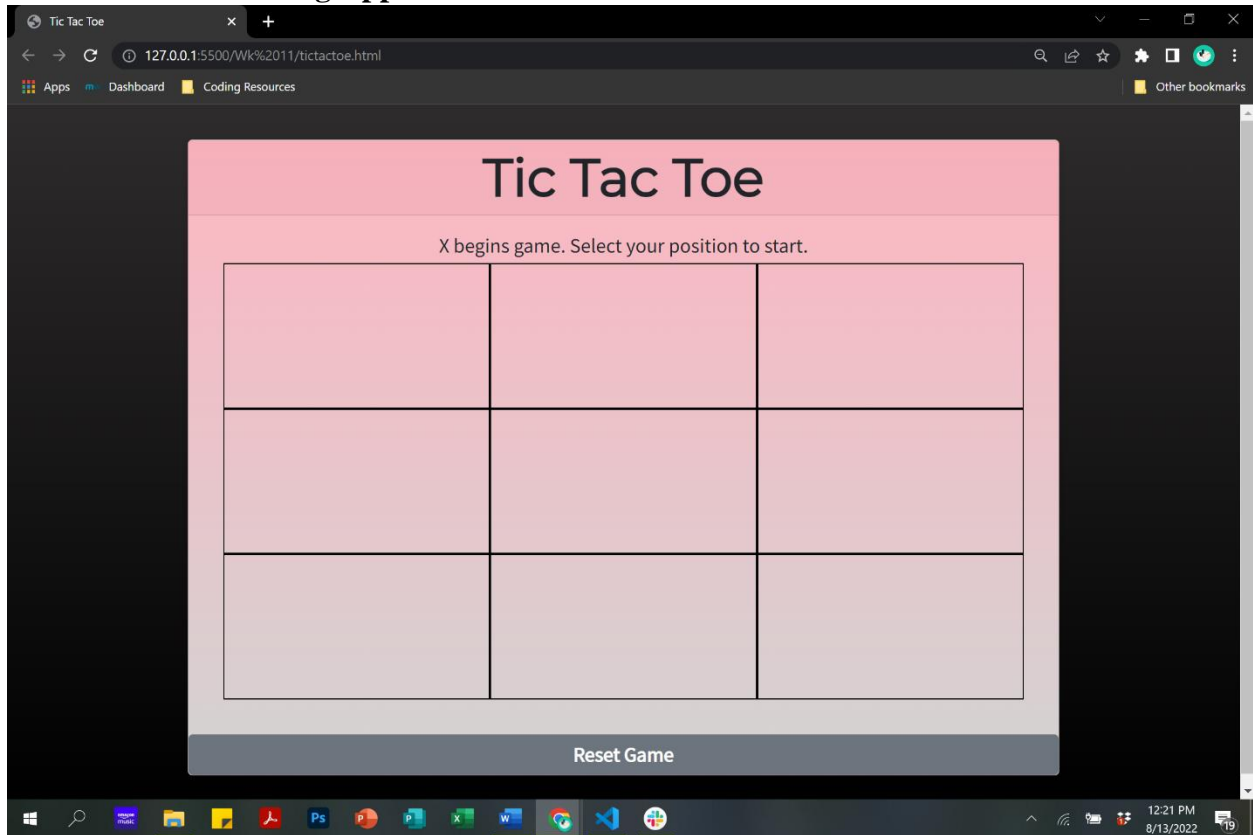
Screenshots of Code:

```javascript
// Top-left editor (lines 70–107)
    currentPlayer = player1; //starts game as Player X
    return currentPlayer;
}

// Game Operator
function clickSelector(clicked_Id) {
    let boxId = clicked_Id;
    console.log(boxId);
    document.getElementById(boxId).innerHTML = currentPlayer;
    document.getElementById(boxId).removeAttribute("onClick"); //selected

    for (let i = 0; i < openBoxes.length; i++) {
        //iterates through available move options and removes whatever was p
        if (openBoxes[i] == boxId) {
            openBoxes.splice(i, 1);
        }
    }

    if (currentPlayer == player1) {
        heading.text(`It is O turn.`); //switches heading
        pX.push(boxId); // adds player selection to array

        for (let i = 0; i < winCombos.length; i++) {
            if (
                pX.includes(winCombos[i][0]) &&
                pX.includes(winCombos[i][1]) &&
                pX.includes(winCombos[i][2])
            ) {
                gridDivs.removeAttr("onClick"); //makes all squares unclickable
                heading.text("Game Over"); //changes heading text
                winAlert.show(); //shows bs alert
                winAlert.text("X Wins!"); //text on alert
            } else if (openBoxes.length == 0) {
                //detects if a game has tied out
                heading.text("Game Over");
                winAlert.show();
                winAlert.text("Tie game!");
            }
```

```javascript
// Top-right editor (lines 107–137)
        }
    }
    //switches player turn
    currentPlayer = player2;
    return currentPlayer;
}

if (currentPlayer == player2) {
    heading.text("It is X turn.");
    pO.push(boxId);
    for (let ii = 0; ii < winCombos.length; ii++) {
        if (
            pO.includes(winCombos[ii][0]) &&
            pO.includes(winCombos[ii][1]) &&
            pO.includes(winCombos[ii][2])
        ) {
            gridDivs.removeAttr("onClick");
            heading.text("Game Over");
            winAlert.show();
            winAlert.text("O Wins!");
        } else if (openBoxes.length == 0) {
            heading.text("Game Over");
            winAlert.show();
            winAlert.text("Tie game!");
        }
    }
    currentPlayer = player1;
    return currentPlayer;
}
}
```

```javascript
// Bottom-left editor (lines 1–37)
const player1 = "X";
const player2 = "O";

let heading = $("#heading"); //jquery to identify heading
let currentPlayer = player1;

let box1 = $("#box1")[0].id; //jquery to identify boxes
let box2 = $("#box2")[0].id;
let box3 = $("#box3")[0].id;
let box4 = $("#box4")[0].id;
let box5 = $("#box5")[0].id;
let box6 = $("#box6")[0].id;
let box7 = $("#box7")[0].id;
let box8 = $("#box8")[0].id;
let box9 = $("#box9")[0].id;

// positions open to select
let openBoxes = [
    "box1",
    "box2",
    "box3",
    "box4",
    "box5",
    "box6",
    "box7",
    "box8",
    "box9",
];

// arrays for player selections
const pX = [];
const pO = [];

// Winning Combos
const winCombos = [
    [box1, box2, box3],
    [box4, box5, box6],
```

```javascript
// Bottom-right editor (lines 34–70)
// Winning Combos
const winCombos = [
    [box1, box2, box3],
    [box4, box5, box6],
    [box7, box8, box9],
    [box1, box5, box9],
    [box3, box5, box7],
    [box1, box4, box7],
    [box2, box5, box8],
    [box3, box6, box9],
];

let gridDivs = $(".col"); //jquery to target all squares.
let winAlert = $("#winAlert"); //jquery for win alert at end of game.

// Reset Button Function
function resetButton() {
    winAlert.hide(); //clears alert
    heading.text(`X begins game. Select your position to start.`); //reset
    gridDivs.text(" "); //clears X and O
    gridDivs.attr("onClick", "clickSelector(this.id)");
    pX.splice(0, pX.length); //clears array
    pO.splice(0, pO.length); //clears array
    openBoxes.splice(0, openBoxes.length); //clears array
    openBoxes.push(
        "box1",
        "box2",
        "box3",
        "box4",
        "box5",
        "box6",
        "box7",
        "box8",
        "box9"
    );
    console.log(openBoxes);
    currentPlayer = player1; //starts game as Player X
```

Screenshots of Running Application:
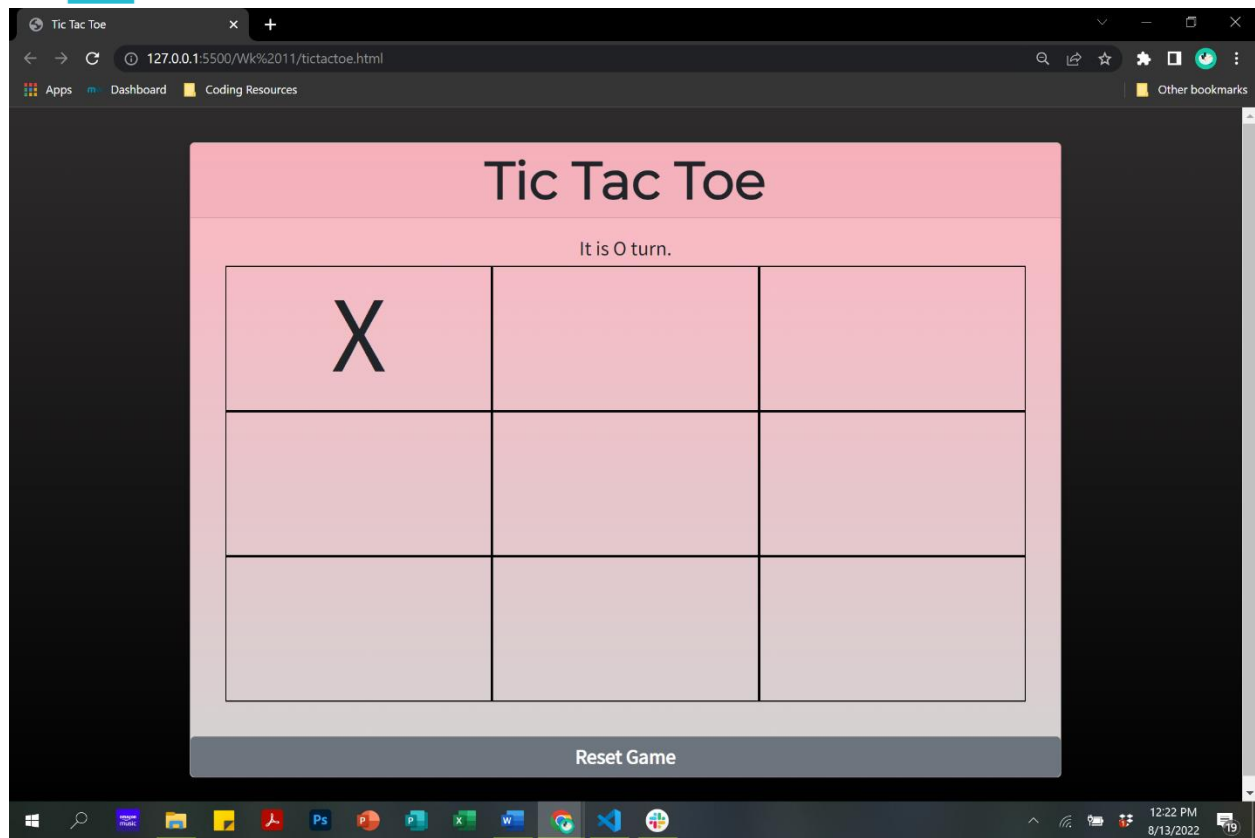
URL to GitHub Repository: https://github.com/jenahern/wk-11-Tic-Tac-Toe/tree/main