+++++++++++++++++++HTML CODE+++++++++++++++++++++++++++++++++=

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="tic.css">

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Tic-Tac-Toe</title>

</head>

<body>

<div class="main">

<button class="btn" id="btn1"></button>

<button class="btn" id="btn2"></button>

<button class="btn" id="btn3"></button>

<button class="btn" id="btn4"></button>

<button class="btn" id="btn5"></button>

<button class="btn" id="btn6"></button>

<button class="btn" id="btn7"></button>

<button class="btn" id="btn8"></button>

<button class="btn" id="btn9"></button>

</div>

<script src="tic.js"></script>

</body>

</html>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JAVASCRIPT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

let chance = "X"; // "X" always starts first

// Function to change the turn between players

function change\_chance() {

chance = (chance === "X") ? "O" : "X"; // Toggle between "X" and "O"

}

// Function to handle click events on buttons

function handleClick(button) {

if (button.innerText === "") { // Check if the button is empty

button.innerText = chance; // Set the inner text of the button to the current player's chance

if (checkWin()) {

alert(chance + " wins!"); // Notify if the player wins

resetGame(); // Reset the game after a win

} else if (checkDraw()) {

alert("It's a draw!"); // Notify if it's a draw

resetGame(); // Reset the game after a draw

} else {

change\_chance(); // Change turn if no winner or draw

}

}

}

// Attach the click event to each button

document.getElementById("btn1").onclick = function() { handleClick(document.getElementById("btn1")); };

document.getElementById("btn2").onclick = function() { handleClick(document.getElementById("btn2")); };

document.getElementById("btn3").onclick = function() { handleClick(document.getElementById("btn3")); };

document.getElementById("btn4").onclick = function() { handleClick(document.getElementById("btn4")); };

document.getElementById("btn5").onclick = function() { handleClick(document.getElementById("btn5")); };

document.getElementById("btn6").onclick = function() { handleClick(document.getElementById("btn6")); };

document.getElementById("btn7").onclick = function() { handleClick(document.getElementById("btn7")); };

document.getElementById("btn8").onclick = function() { handleClick(document.getElementById("btn8")); };

document.getElementById("btn9").onclick = function() { handleClick(document.getElementById("btn9")); };

// Function to check for a win

function checkWin() {

const winConditions = [

[document.getElementById('btn1'), document.getElementById('btn2'), document.getElementById('btn3')], // Row 1

[document.getElementById('btn4'), document.getElementById('btn5'), document.getElementById('btn6')], // Row 2

[document.getElementById('btn7'), document.getElementById('btn8'), document.getElementById('btn9')], // Row 3

[document.getElementById('btn1'), document.getElementById('btn4'), document.getElementById('btn7')], // Column 1

[document.getElementById('btn2'), document.getElementById('btn5'), document.getElementById('btn8')], // Column 2

[document.getElementById('btn3'), document.getElementById('btn6'), document.getElementById('btn9')], // Column 3

[document.getElementById('btn1'), document.getElementById('btn5'), document.getElementById('btn9')], // Diagonal 1

[document.getElementById('btn3'), document.getElementById('btn5'), document.getElementById('btn7')] // Diagonal 2

];

for (let condition of winConditions) {

if (condition[0].innerText === condition[1].innerText &&

condition[1].innerText === condition[2].innerText &&

condition[0].innerText !== "") {

return true; // A win condition is met

}

}

return false; // No win condition met

}

// Function to check if the game is a draw

function checkDraw() {

for (let i = 1; i <= 9; i++) {

if (document.getElementById('btn' + i).innerText === "") {

return false; // There is still an empty button

}

}

return true; // All buttons are filled

}

// Function to reset the game

function resetGame() {

for (let i = 1; i <= 9; i++) {

document.getElementById('btn' + i).innerText = ""; // Clear all buttons

}

chance = "X"; // Reset the game to player X starting

}

**Explanation & Documentation**

**1. Variables and Game State**

* chance: A string variable that tracks the current player's turn. It is initially set to "X".

**2. Functions**

* **change\_chance()**:
  + This function toggles the turn between player "X" and "O". After each valid move, it switches the current player.
* **handleClick(button)**:
  + This function is called when a player clicks a button on the board.
  + If the button is empty, it places the current player's symbol ("X" or "O") on the button and checks for a win or draw:
    - **checkWin()**: Checks if there is a winning combination on the board.
    - **checkDraw()**: Checks if all buttons are filled, which would result in a draw.
    - If no one has won or drawn, the turn is passed to the next player using change\_chance().
* **checkWin()**:
  + This function checks the eight possible winning combinations (three rows, three columns, and two diagonals) to see if the current player has won. It returns true if a player has won, otherwise false.
* **checkDraw()**:
  + This function checks if all buttons are filled. If the game board is full and no player has won, it returns true (indicating a draw), otherwise it returns false.
* **resetGame()**:
  + This function resets the game by clearing all button texts and setting the chance variable back to "X".

**3. HTML Button Elements**

* The btn1, btn2, ..., btn9 are <button> elements that represent the game board. Each button has an onclick event that calls handleClick(), which manages the game logic.

**4. Game Flow**

* The game starts with player "X" and alternates turns between players "X" and "O".
* When a player clicks a button, the corresponding function checks for a win or a draw.
* After a win or a draw, an alert is shown, and the game is reset.

**Conclusion:**

This code implements a fully functional Tic-Tac-Toe game with alternating players and checks for win and draw conditions. The game can be played by two players, with automatic reset after each game round.

s