Tic-Tac-Toe Game

Name: K.R.DeviSree

Internship Project

Abstract

The Tic-Tac-Toe Game is a timeless, easy-to-learn puzzle game where two players alternately mark the spaces in a 3×3 grid with X and O. The goal is to place three respective marks in a horizontal, vertical, or diagonal row. This project brings the game to the digital realm using fundamental web technologies like HTML, CSS, and JavaScript. The implementation allows two users to interact in real time on the same device, with visual feedback on each move and immediate status updates. The core logic includes detecting winning combinations, preventing overwriting of already-marked cells, and checking for draws. This simple yet strategic game helps developers understand event-driven programming, DOM manipulation, and user interface responsiveness. The project also serves as an excellent entry point for learning front-end development and lays the groundwork for more advanced game development in the future.

Introduction

Tic-Tac-Toe, also known as Noughts and Crosses, has long been a staple of pen-and-paper games and logical thinking. Its rules are simple enough for young children, yet it provides enough strategic depth to remain engaging for players of all ages. The objective of this project is to create a web-based version of Tic-Tac-Toe, which can be played by two players on a browser without the need for installation or back-end support. By translating this game into a browser-compatible application, we not only bring nostalgic gameplay to the digital world but also demonstrate how core programming principles apply to user interaction. Through this project, I have deepened my understanding of how to manipulate HTML elements using JavaScript, apply dynamic styling through CSS, and implement real-time interaction logic. This project aims to balance functionality with user experience and design responsiveness.

Objectives

The main objective of the project is to design and develop a fully interactive, user-friendly Tic-Tac-Toe game using only client-side technologies.

The additional goals include:

Implementing alternating turn-taking for two players with visual feedback.

Designing a responsive 3x3 grid that clearly shows player moves.

Validating user actions to avoid overwriting existing marks.

Detecting win conditions efficiently using JavaScript logic.

Displaying relevant game messages like win, draw, or player turns.

Providing a restart button that resets the game board and state for new rounds.

**Code Implementation**

**HTML:** Show your game board structure

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Tic Tac Toe Game</title>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<h1>Tic Tac Toe Game</h1>

<div class="game-board">

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

<div class="cell" data-cell></div>

</div>

<div class="game-info">

<p class="status"></p>

<button id="restartButton">Restart</button>

</div>

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

</body>

</html>

CSS: Styling the Board and Feedback

body {

font-family: Arial, sans-serif;

text-align: center;

margin-top: 50px;

}

.game-board {

display: grid;

grid-template-columns: repeat(3, 100px);

grid-gap: 10px;

justify-content: center;

}

.cell {

width: 100px;

height: 100px;

background-color: #f0f0f0;

font-size: 3em;

display: flex;

align-items: center;

justify-content: center;

cursor: pointer;

user-select: none;

}

.cell.x {

color: #e74c3c;

}

.cell.o {

color: #3498db;

}

.status {

font-size: 1.5em;

margin: 20px;

}

#restartButton {

padding: 10px 20px;

font-size: 1em;

cursor: pointer;

}

JavaScript: Game Logic (Turns, Win Check, Restart)

const cells = document.querySelectorAll('[data-cell]');

const statusText = document.querySelector('.status');

const restartButton = document.getElementById('restartButton');

let isXTurn = true;

let board = ["", "", "", "", "", "", "", "", ""];

const winningCombos = [

[0, 1, 2],

[3, 4, 5],

[6, 7, 8],

[0, 3, 6],

[1, 4, 7],

[2, 5, 8],

[0, 4, 8],

[2, 4, 6]

];

function startGame() {

cells.forEach(cell => {

cell.textContent = '';

cell.classList.remove('x', 'o');

cell.addEventListener('click', handleClick, { once: true });

});

isXTurn = true;

statusText.textContent = "X's turn";

board.fill("");

}

function handleClick(e) {

const cell = e.target;

const currentPlayer = isXTurn ? 'x' : 'o';

const cellIndex = Array.from(cells).indexOf(cell);

if (board[cellIndex] !== "") return;

board[cellIndex] = currentPlayer;

cell.textContent = currentPlayer.toUpperCase();

cell.classList.add(currentPlayer);

if (checkWin(currentPlayer)) {

statusText.textContent = `${currentPlayer.toUpperCase()} Wins!`;

endGame();

} else if (board.every(cell => cell !== "")) {

statusText.textContent = "Draw!";

endGame();

} else {

isXTurn = !isXTurn;

statusText.textContent = `${isXTurn ? 'X' : 'O'}'s turn`;

}

}

function checkWin(player) {

return winningCombos.some(combo => {

return combo.every(index => board[index] === player);

});

}

function endGame() {

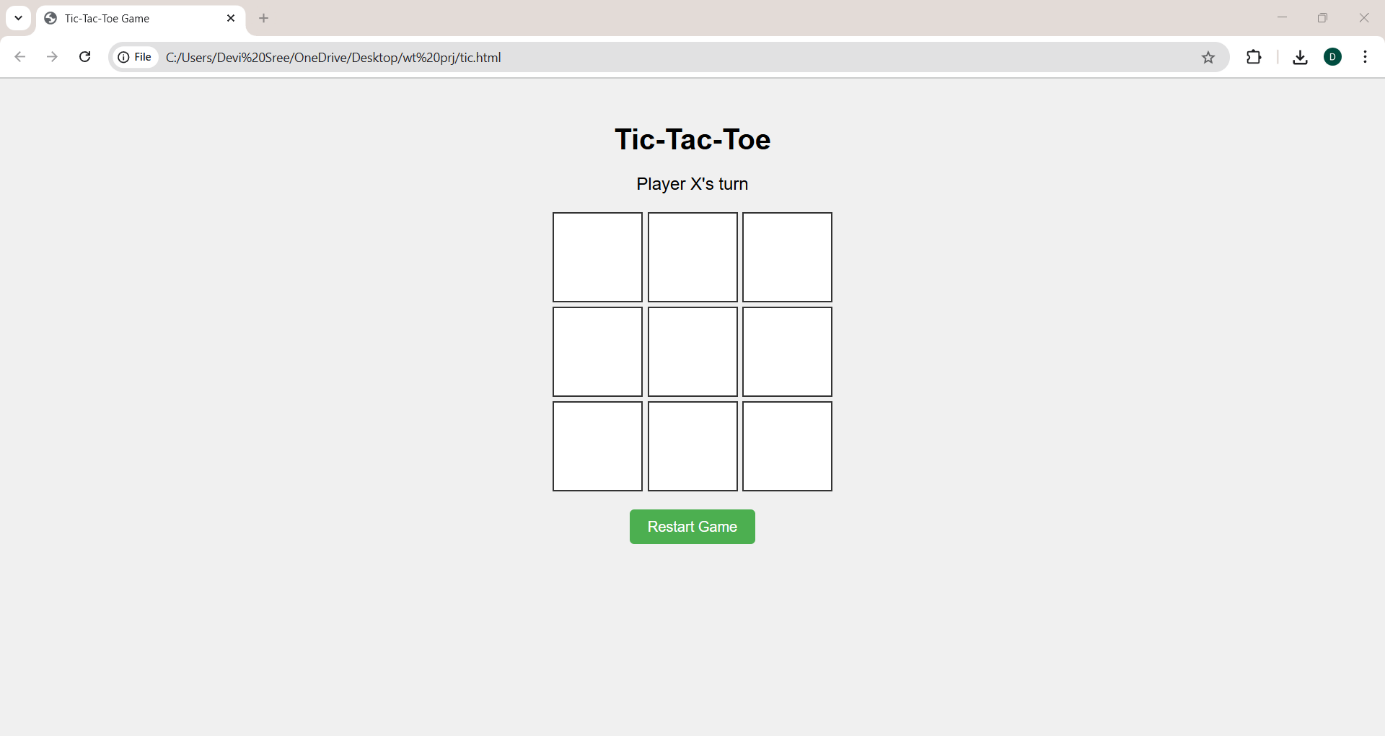
cells.forEach(cell => cell.removeEventListener('click', handleClick));

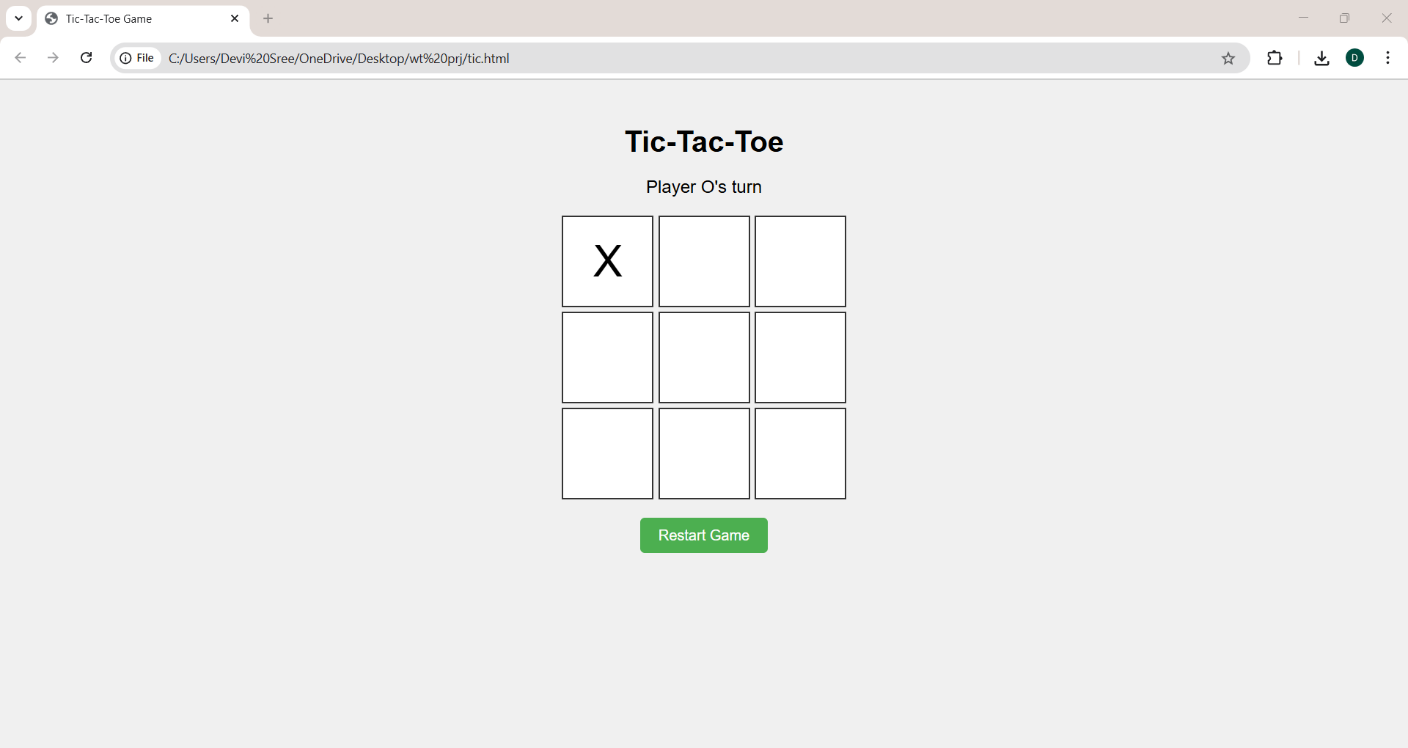
}

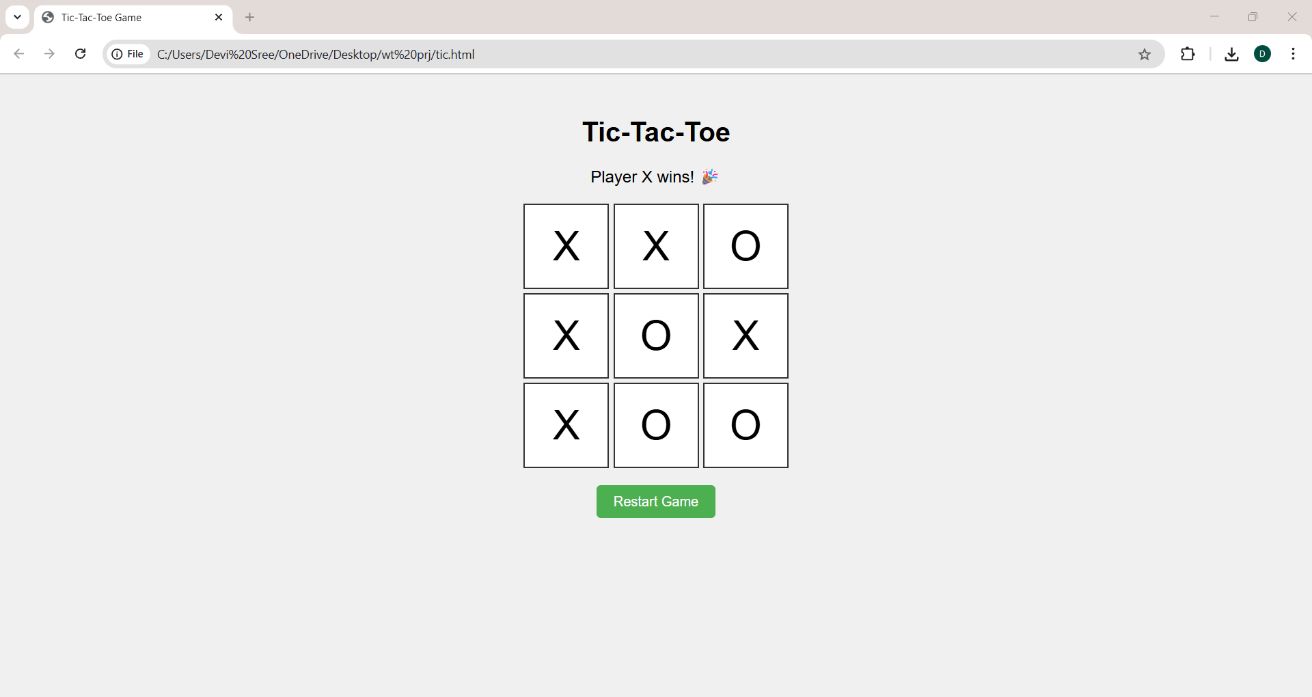
restartButton.addEventListener('click', startGame);

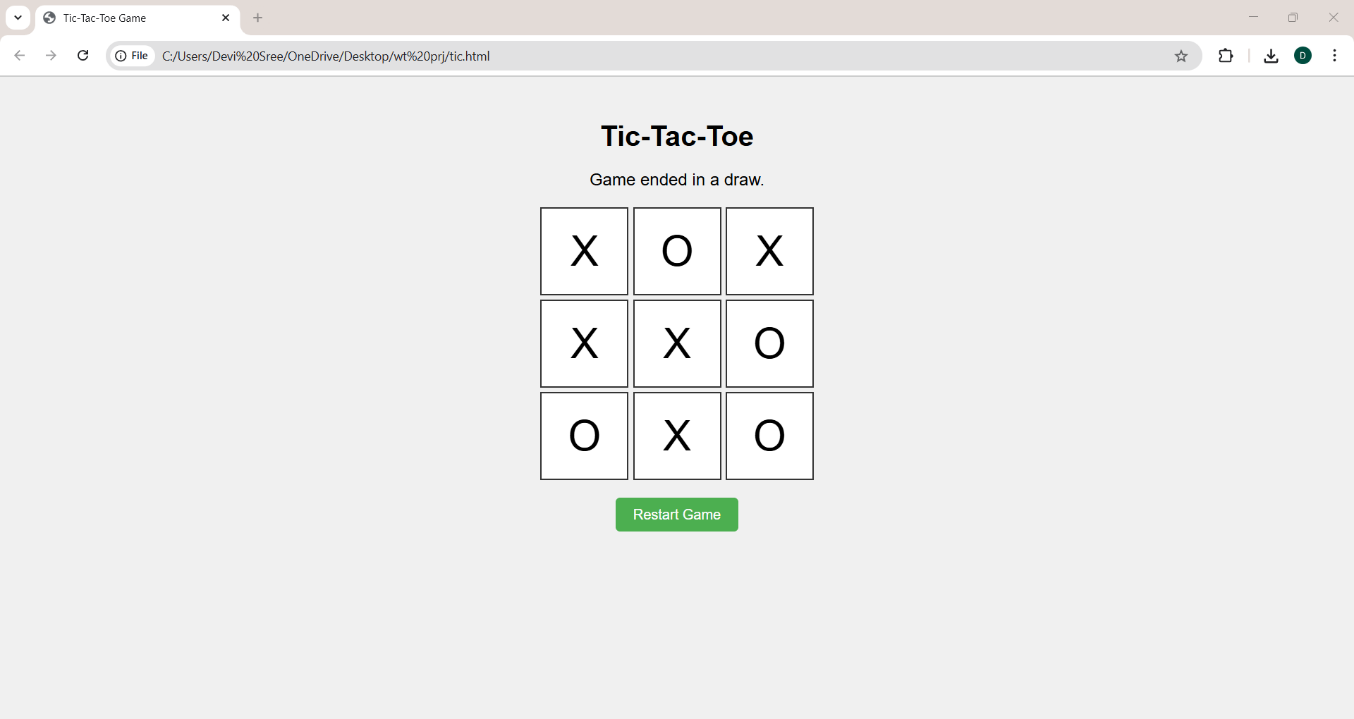
startGame();

OUTPUT









CONCLUSION

The Tic-Tac-Toe web application successfully demonstrates the fundamental concepts of front-end web development using HTML, CSS, and JavaScript. Through this project, I gained hands-on experience in building an interactive user interface, managing game state logic, and implementing event-driven programming. The game effectively handles user inputs, enforces rules like preventing overwriting of cells, detects winning and draw conditions, and provides immediate feedback to the players.

This project reinforced my understanding of DOM manipulation and dynamic styling, as well as how to create a responsive and user-friendly interface. Moreover, it highlighted the importance of structuring code logically to maintain readability and ease of debugging.

Overall, the Tic-Tac-Toe game serves as an excellent foundation for exploring more complex game development and client-side programming challenges. Future enhancements could include adding AI opponents, score tracking, or online multiplayer capabilities. This internship project has been a valuable step in my journey toward becoming a proficient front-end developer.