**TASK 3**

***HTML***

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Tic Tac Toe</title>

    <link rel="stylesheet" href="task3.css">

</head>

<body>

    <h1>Tic-Tac-Toe</h1>

    <div class="mode">

    <button onclick="setMode('pvp')">Player vs Player</button>

    <button onclick="setMode('cpu')">Player vs Computer</button>

    </div>

    <div class="game-board" id="board"></div>

    <h2 id="status"></h2>

    <button onclick="restartGame()">Restart</button>

    <script src="task3.js"></script>

</body>

</html>

***CSS***

body {

    font-family: Arial, sans-serif;

    text-align: center;

    background-color: #f4f4f4;

}

.game-board {

    display: grid;

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

    gap: 5px;

    justify-content: center;

    margin: 20px auto;

}

.cell {

    width: 100px;

    height: 100px;

    background-color: white;

    font-size: 2.5rem;

    display: flex;

    align-items: center;

    justify-content: center;

    cursor: pointer;

    border: 2px solid #333;

}

button {

    margin: 10px;

    padding: 10px 20px;

    font-size: 16px;

    cursor: pointer;

}

***JS***

let board = ['', '', '', '', '', '', '', '', ''];

let currentPlayer = 'X';

let gameActive = true;

let mode = 'pvp';

const winningCombos = [

  [0,1,2], [3,4,5], [6,7,8], // rows

  [0,3,6], [1,4,7], [2,5,8], // columns

  [0,4,8], [2,4,6]           // diagonals

];

const boardElement = document.getElementById('board');

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

function setMode(selectedMode) {

    mode = selectedMode;

    restartGame();

}

function createBoard() {

    boardElement.innerHTML = '';

    board.forEach((cell, index) => {

    const div = document.createElement('div');

    div.classList.add('cell');

    div.setAttribute('data-index', index);

    div.addEventListener('click', handleClick);

    div.textContent = cell;

    boardElement.appendChild(div);

});

}

function handleClick(e) {

    const index = e.target.getAttribute('data-index');

if (!gameActive || board[index] !== '') return;

    board[index] = currentPlayer;

    e.target.textContent = currentPlayer;

if (checkWin()) {

    statusText.textContent = `${currentPlayer} wins!`;

    gameActive = false;

    return;

}

if (board.every(cell => cell !== '')) {

    statusText.textContent = `It's a draw!`;

    gameActive = false;

    return;

}

    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';

    statusText.textContent = `${currentPlayer}'s turn`;

if (mode === 'cpu' && currentPlayer === 'O') {

    setTimeout(cpuMove, 500);

}

}

function checkWin() {

    return winningCombos.some(combo => {

    const [a, b, c] = combo;

    return board[a] && board[a] === board[b] && board[a] === board[c];

    });

}

function cpuMove() {

    let available = board.map((val, idx) => val === '' ? idx : null).filter(v => v !== null);

    let choice = available[Math.floor(Math.random() \* available.length)];

    board[choice] = 'O';

    document.querySelector(`.cell[data-index='${choice}']`).textContent = 'O';

if (checkWin()) {

    statusText.textContent = `O wins!`;

    gameActive = false;

    return;

}

if (board.every(cell => cell !== '')) {

    statusText.textContent = `It's a draw!`;

    gameActive = false;

    return;

}

    currentPlayer = 'X';

    statusText.textContent = `${currentPlayer}'s turn`;

}

function restartGame() {

    board = ['', '', '', '', '', '', '', '', ''];

    currentPlayer = 'X';

    gameActive = true;

    statusText.textContent = `${currentPlayer}'s turn`;

    createBoard();

}

    window.onload = () => {

    createBoard();

};