Project report : Tic-Tac-Toe game

1. Introduction Tic Tac Toe is a classic two-player game where players take turns marking a 3×3 grid with their respective symbols (X or O). The objective is to form a horizontal, vertical, or diagonal line of three identical symbols. This project implements the game using HTML, CSS, and JavaScript.

2. Objectives

- To develop an interactive and user-friendly Tic Tac Toe game.
- To implement game logic using JavaScript.
- To provide a visually appealing design with CSS.
- To enhance user engagement with smooth transitions and animations.

3. Technologies Used

- HTML: Structure of the game interface.
- CSS: Styling and layout of the game.
- JavaScript: Game logic and interactivity.

4. Features

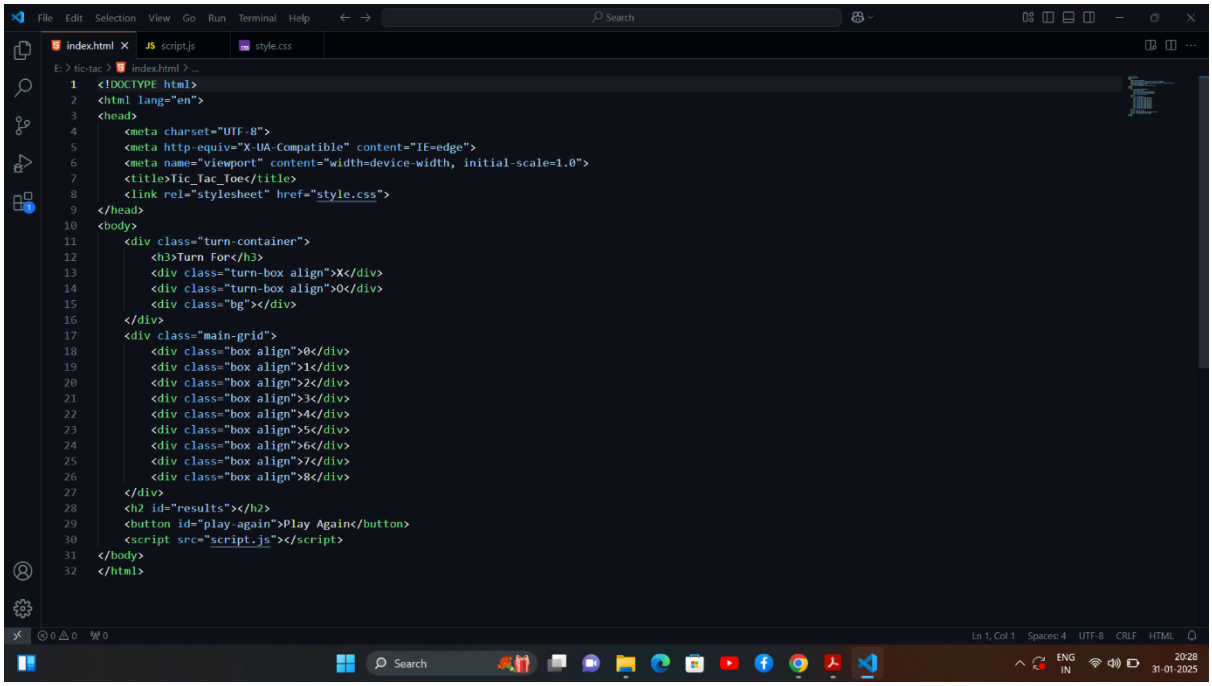
- Player turn indication.
- Winning condition detection.
- Draw detection.
- Play Again button to restart the game.
- Smooth animations for better user experience.

5. Implementation Details

- The HTML file (index.html) contains a structured layout with a 3×3 grid for the game board.
- The CSS file (style.css) ensures an attractive design, including grid styling and hover effects.
- The JavaScript file (script.js) manages game logic, including:
    - Handling player turns.
    - Checking for winning combinations.
    - Detecting a draw scenario.
    - Allowing users to restart the game.

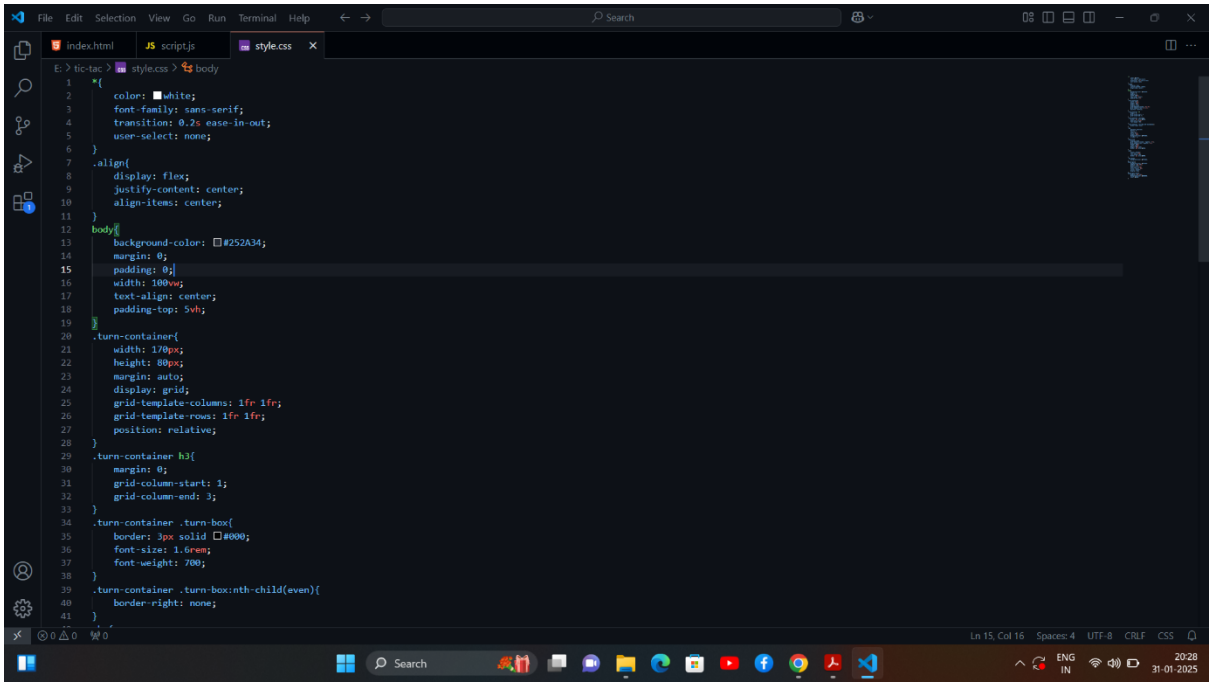
6. Code Breakdown

- **HTML:** Defines the grid structure and includes elements for displaying results.

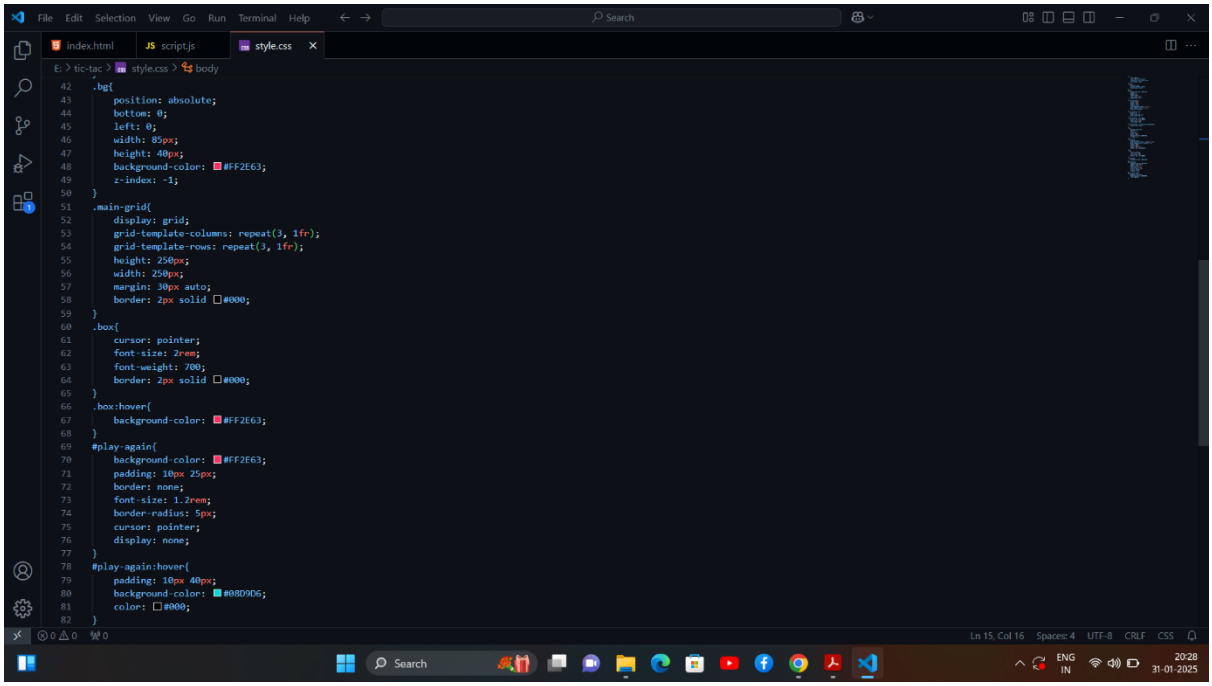


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tic_Tac_Toe</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="turn-container">
        <h3>Turn For</h3>
        <div class="turn-box align">X</div>
        <div class="turn-box align">O</div>
        <div class="bg"></div>
    </div>
    <div class="main-grid">
        <div class="box align">0</div>
        <div class="box align">1</div>
        <div class="box align">2</div>
        <div class="box align">3</div>
        <div class="box align">4</div>
        <div class="box align">5</div>
        <div class="box align">6</div>
        <div class="box align">7</div>
        <div class="box align">8</div>
    </div>
    <h2 id="results"></h2>
    <button id="play-again">Play Again</button>
    <script src="script.js"></script>
</body>
</html>
```

- **CSS:** Provides a modern and responsive look with colors, grid positioning, and animations.

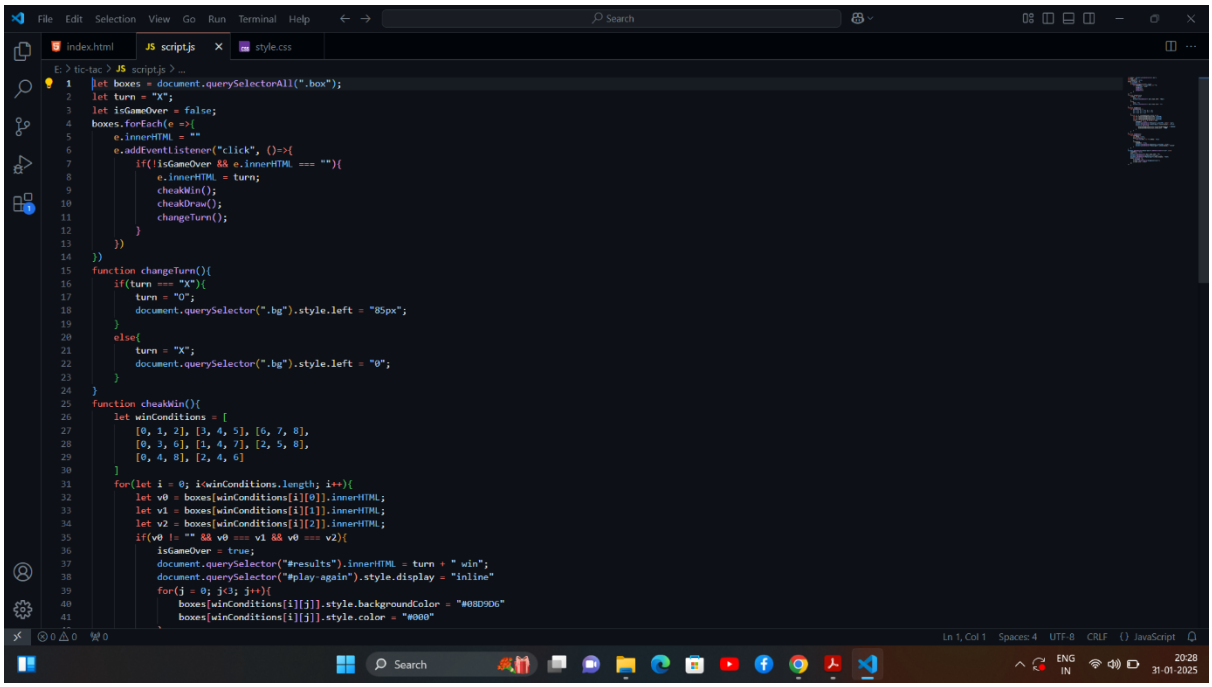


```
*{
    color: white;
    font-family: sans-serif;
    transition: 0.2s ease-in-out;
    user-select: none;
}
.align{
    display: flex;
    justify-content: center;
    align-items: center;
}
body{
    background-color: #252A34;
    margin: 0;
    padding: 0;
    width: 100vw;
    text-align: center;
    padding-top: 5vh;
}
.turn-container{
    width: 170px;
    height: 80px;
    margin: auto;
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-template-rows: 1fr 1fr;
    position: relative;
}
.turn-container h3{
    margin: 0;
    grid-column-start: 1;
    grid-column-end: 3;
}
.turn-container .turn-box{
    border: 3px solid #000;
    font-size: 1.6rem;
    font-weight: 700;
}
.turn-container .turn-box:nth-child(even){
    border-right: none;
}
```

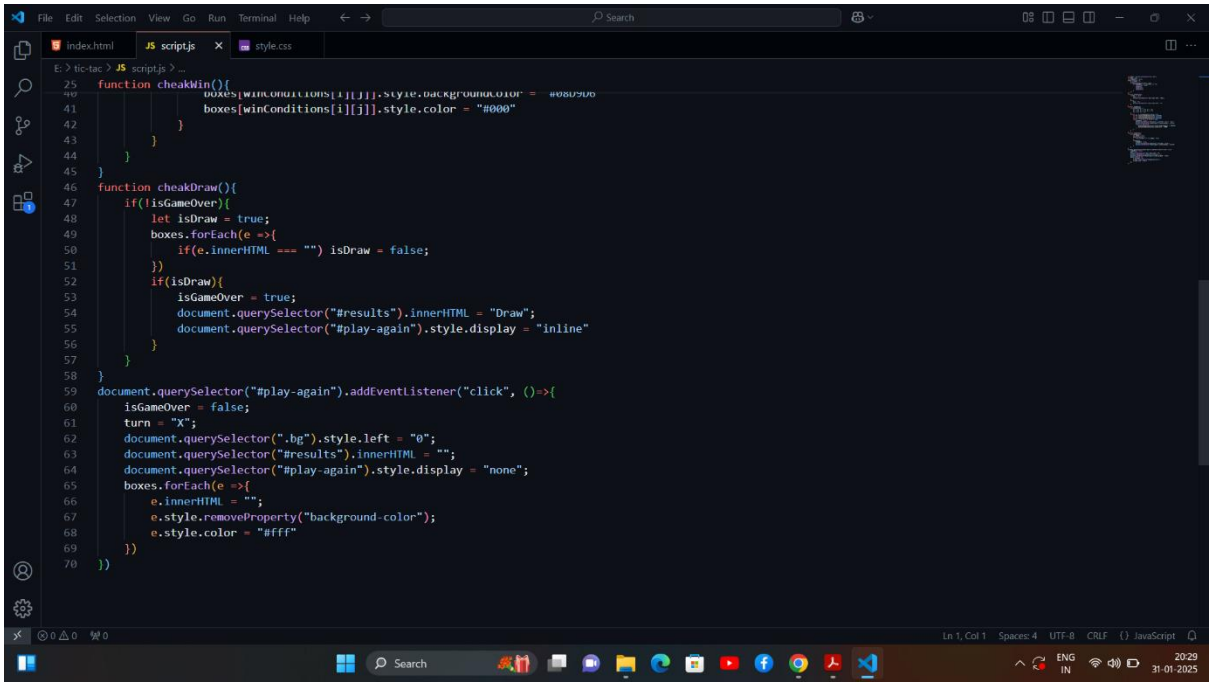


```css
.bg{
    position: absolute;
    bottom: 0;
    left: 0;
    width: 85px;
    height: 40px;
    background-color: #FF2E63;
    z-index: -1;
}
.main-grid{
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 1fr);
    height: 250px;
    width: 250px;
    margin: 30px auto;
    border: 2px solid #000;
}
.box{
    cursor: pointer;
    font-size: 2rem;
    font-weight: 700;
    border: 2px solid #000;
}
.box:hover{
    background-color: #FF2E63;
}
#play-again{
    background-color: #FF2E63;
    padding: 10px 25px;
    border: none;
    font-size: 1.2rem;
    border-radius: 5px;
    cursor: pointer;
    display: none;
}
#play-again:hover{
    padding: 10px 40px;
    background-color: #08D9D6;
    color: #000;
}
```

- **JavaScript:** Implements event listeners, turn-based mechanics, win condition checks, and reset functionality.



```js
let boxes = document.querySelectorAll(".box");
let turn = "X";
let isGameOver = false;
boxes.forEach(e =>{
    e.innerHTML = ""
    e.addEventListener("click", ()=>{
        if(!isGameOver && e.innerHTML === ""){
            e.innerHTML = turn;
            cheakWin();
            cheakDraw();
            changeTurn();
        }
    })
})
function changeTurn(){
    if(turn === "X"){
        turn = "O";
        document.querySelector(".bg").style.left = "85px";
    }
    else{
        turn = "X";
        document.querySelector(".bg").style.left = "0";
    }
}
function cheakWin(){
    let winConditions = [
        [0, 1, 2], [3, 4, 5], [6, 7, 8],
        [0, 3, 6], [1, 4, 7], [2, 5, 8],
        [0, 4, 8], [2, 4, 6]
    ]
    for(let i = 0; i<winConditions.length; i++){
        let v0 = boxes[winConditions[i][0]].innerHTML;
        let v1 = boxes[winConditions[i][1]].innerHTML;
        let v2 = boxes[winConditions[i][2]].innerHTML;
        if(v0 != "" && v0 === v1 && v0 === v2){
            isGameOver = true;
            document.querySelector("#results").innerHTML = turn + " win";
            document.querySelector("#play-again").style.display = "inline"
            for(j = 0; j<3; j++){
                boxes[winConditions[i][j]].style.backgroundColor = "#08D9D6"
                boxes[winConditions[i][j]].style.color = "#000"
```



7. Testing and Debugging

- The game has been tested for responsiveness and functionality.
- Bugs related to incorrect win conditions and event handling were fixed.
- The UI has been refined for a better user experience.

8. Future Enhancements

- Adding an AI opponent for single-player mode.
- Implementing a scoreboard to track wins.
- Enhancing UI with sound effects and animations.
- Supporting multiplayer gameplay over a network.

9. Conclusion This project successfully demonstrates the implementation of a simple Tic Tac Toe game using front-end technologies. The game is fully functional, visually appealing, and offers a smooth user experience. Future updates can further improve gameplay and interactivity.