TASK 03 TIC-TAC-TOE WEB APPLICATION

index.html

```
<!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</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div class="container">
    <h1>Tic Tac Toe</h1>
    <div id="board" class="board">
      <div class="cell" data-index="0"></div>
      <div class="cell" data-index="1"></div>
      <div class="cell" data-index="2"></div>
      <div class="cell" data-index="3"></div>
      <div class="cell" data-index="4"></div>
      <div class="cell" data-index="5"></div>
      <div class="cell" data-index="6"></div>
```

```
 <div class="cell" data-index="7"></div>
 <div class="cell" data-index="8"></div>
 </div>
 <p id="status">Player X's turn</p>
 <button id="reset">Restart
Game</button>
 </div>
 <script src="script.js"></script>
</body>
</html>
```

style.css

```
body {
  margin: 0;
  padding: 0;
  font-family: 'Segoe UI', sans-serif;
  background-color: #1e1e2f;
  color: #fff;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  text-align: center;
}

h1 {
  margin-bottom: 20px;
}

.board {
 display: grid;
 grid-template-columns: repeat(3, 100px);
```

```css
 grid-template-rows: repeat(3, 100px);
 gap: 10px;
 justify-content: center;
 margin-bottom: 20px;
}

.cell {
 background-color: #2c2c44;
 border-radius: 10px;
 display: flex;
 align-items: center;
 justify-content: center;
 font-size: 2.5rem;
 cursor: pointer;
 transition: background 0.3s;
}
.cell:hover {
 background-color: #3a3a5c;
}
#status {
 font-size: 1.2rem;
 margin-bottom: 10px;
}
```

```
button {
 padding: 10px 20px;
 background-color: #4caf50;
 border: none;
 border-radius: 8px;
 color: white;
 font-size: 1rem;
 cursor: pointer;
 transition: background 0.3s;
}
button:hover {
 background-color: #45a049;
}
```

script.js

```
const cells = document.querySelectorAll(".cell");
const statusText = document.getElementById("status");
const resetBtn = document.getElementById("reset");

let board = ["", "", "", "", "", "", "", "", ""];
let currentPlayer = "X";
let gameActive = true;

const winConditions = [
  [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
];
function handleCellClick(e) {
 const index = e.target.getAttribute("data-index");
 if (board[index] !== "" || !gameActive) return;
 board[index] = currentPlayer;
 e.target.textContent = currentPlayer;
 if (checkWinner()) {
 statusText.textContent = `Player ${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 = `Player ${currentPlayer}'s
turn`;
}
function checkWinner() {
 return winConditions.some(condition => {
 const [a, b, c] = condition;
 return board[a] && board[a] === board[b] &&
board[b] === board[c];
 });
}
function resetGame() {
 board = ["", "", "", "", "", "", "", "", ""];
 currentPlayer = "X";
 gameActive = true;
 statusText.textContent = `Player ${currentPlayer}'s
turn`;
 cells.forEach(cell => (cell.textContent = ""));
}
cells.forEach(cell => cell.addEventListener("click",
handleCellClick));
resetBtn.addEventListener("click", resetGame);
```