## Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

**TIC TAC TOE GAME (TAILWIND)**

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Dr Raveesh Samkaria Samanyu Sharma

2210990771

G-12

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**INTRODUCTION-**

Tic Tac Toe is a classic paper-and-pencil game that has been enjoyed by people of all ages for generations. Here are some basic facts about Tic Tac Toe, including how it was made and its origins:

**Origins:** The origins of Tic Tac Toe can be traced back to ancient civilizations, with evidence suggesting that similar games were played in ancient Egypt around 1300 BCE. The game has also been found in ancient Roman and Greek cultures under various names.

**Names:** Tic Tac Toe is known by various names around the world. In the United Kingdom and some other English-speaking countries, it's often called "Noughts and Crosses." In other parts of the world, it's known as "Xs and Os" or simply "Three in a Row."

**Gameplay:** Tic Tac Toe is played on a grid usually consisting of a 3x3 square. Players take turns marking spaces in the grid with their symbol, typically an "X" or an "O," with the objective of getting three of their symbols in a row, either horizontally, vertically, or diagonally.

**Simplicity:** One of the reasons for Tic Tac Toe's enduring popularity is its simplicity. The rules are easy to understand, and the game can be played virtually anywhere with just a pen and paper.

**Strategy:** Despite its simplicity, Tic Tac Toe does involve some strategy, particularly in preventing your opponent from getting three in a row while trying to form your own winning line.

**Computational Complexity:** Despite its simple rules, Tic Tac Toe has been extensively studied in the field of mathematics and computer science. It's often used as a simple example to demonstrate concepts such as game theory, decision trees, and artificial intelligence algorithms.

**Digital Versions:** With the advent of computers and digital technology, Tic Tac Toe has been implemented in various software forms, including computer games, mobile apps, and websites. These digital versions often include features like single-player mode against a computer opponent or multiplayer mode against other players online.

**OBJECTIVE-**

1. Objective: The goal of Tic Tac Toe is to be the first player to form a line of three of your symbols (either X's or O's) horizontally, vertically, or diagonally on the grid.
2. Starting the Game: Players take turns placing their symbol on an empty square of the 3x3 grid.
3. Taking Turns: The game alternates between players, with one player placing X's and the other placing O's.
4. Winning the Game: If a player successfully aligns three of their symbols in a row, they win the game! If the entire grid is filled with symbols and no player has achieved a winning line, the game ends in a draw.

**NEED FOR THE GAME-**

1. **Entertainment:** Tic Tac Toe provides simple, yet engaging entertainment. It's a game that can be enjoyed by people of all ages, from children to adults, and it doesn't require extensive learning or setup. This makes it a popular choice for passing the time, whether it's during a break at work, a family gathering, or just for fun at home.
2. **Skill Development:** Despite its simplicity, Tic Tac Toe involves strategic thinking and planning. Players must anticipate their opponent's moves while planning their own to achieve victory. For children, it can be a valuable tool for developing critical thinking skills, spatial reasoning, and decision-making abilities.
3. **Accessibility:** Tic Tac Toe requires minimal resources to play. All that's needed is a pen and paper or a simple grid drawn on any surface. With the advent of digital versions, such as those made in JavaScript, Tic Tac Toe has become even more accessible, allowing people to play anytime, anywhere, using their computers, smartphones, or tablets.
4. **Educational Purposes:** Tic Tac Toe is often used as an introductory example in programming tutorials and courses. Building a Tic Tac Toe game helps beginners grasp fundamental programming concepts such as variables, loops, conditional statements, and event handling. It provides a practical and engaging way to apply theoretical knowledge to real-world projects.
5. **Community Building:** Tic Tac Toe is a social game that fosters interaction and camaraderie. Whether played in person or online, it encourages friendly competition and communication between players. Multiplayer versions of the game allow people to connect and compete with friends, family, or even strangers from around the world.
6. **Historical Significance:** Tic Tac Toe has a rich history and cultural significance. It's a game that has been played for centuries in various forms and under different names. Preserving and continuing to play games like Tic Tac Toe helps maintain a connection to our cultural heritage and traditions.

Overall, the need for a Tic Tac Toe game lies in its ability to provide entertainment, stimulate cognitive development, promote accessibility, serve educational purposes, facilitate social interaction, and preserve cultural heritage. Whether implemented in JavaScript or any other programming language, Tic Tac Toe remains a timeless classic that continues to captivate and engage people worldwide.

**PROPOSED DESIGN-**

Designing a Tic Tac Toe game using Tailwind CSS and JavaScript involves structuring the HTML for the game board, styling it with Tailwind classes, and implementing the game logic using JavaScript. Here's a proposed design methodology:

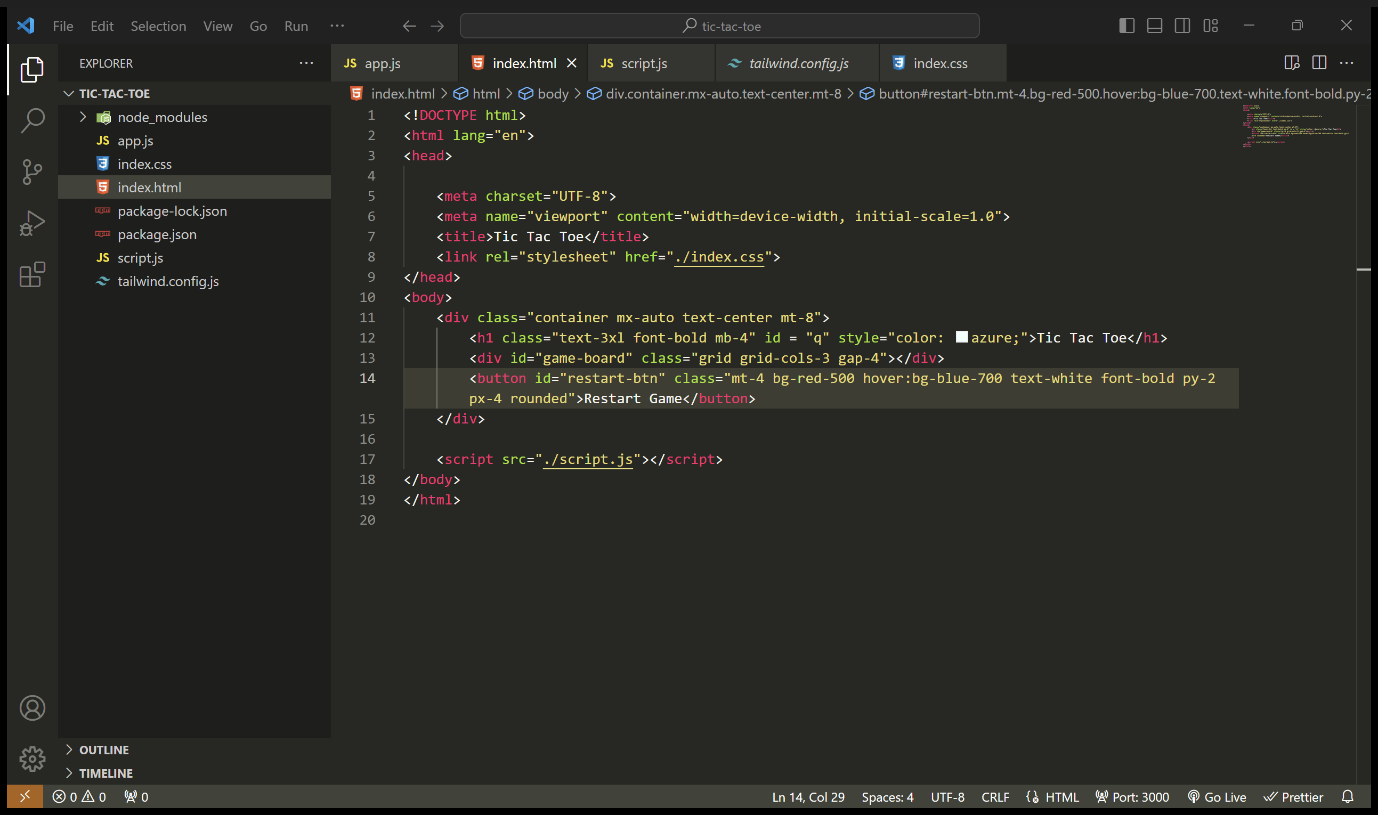
1. **HTML Structure:** Start by creating the HTML structure for the game board and interface elements. This includes:
   * A container for the game board.
   * A grid layout representing the 3x3 game board.
   * Buttons for starting a new game or resetting the current game.
   * Optionally, elements to display the current player's turn and game status (winner or draw).
2. **Tailwind CSS Styling:** Utilize Tailwind CSS utility classes to style the HTML elements. Customize the colors, fonts, and layout to create an attractive and responsive design. Consider using Tailwind's responsive design features to ensure the game looks good on various screen sizes.
3. **JavaScript Logic:** Implement the game logic using JavaScript. This includes:
   * Keeping track of the game state, such as the current player and the state of each cell on the game board.
   * Handling player moves when a cell is clicked.
   * Checking for win conditions after each move.
   * Handling end-game scenarios, such as declaring a winner or ending in a draw.
   * Implementing functionality for starting a new game or resetting the current game.
4. **Event Handling:** Use event listeners in JavaScript to detect when a cell on the game board is clicked or when the new game/reset buttons are clicked. When a cell is clicked, trigger the appropriate game logic to update the cell with the current player's symbol and check for win conditions.
5. **Testing and Refinement:** Test the Tic Tac Toe game extensively to ensure it functions correctly and is user-friendly. Test for different scenarios, including winning combinations, draws, and edge cases. Refine the design and functionality based on user feedback and testing results.
6. **Accessibility and Usability:** Ensure the game is accessible to all users, including those using assistive technologies. Use semantic HTML elements, provide appropriate alt text for images, and ensure the game is navigable using keyboard controls. Consider usability aspects such as providing visual feedback for player actions and clear instructions for playing the game.

By following this methodology, you can create a well-designed and functional Tic Tac Toe game using Tailwind CSS and JavaScript. Remember to iterate on the design and functionality based on user feedback to create the best possible gaming experience.

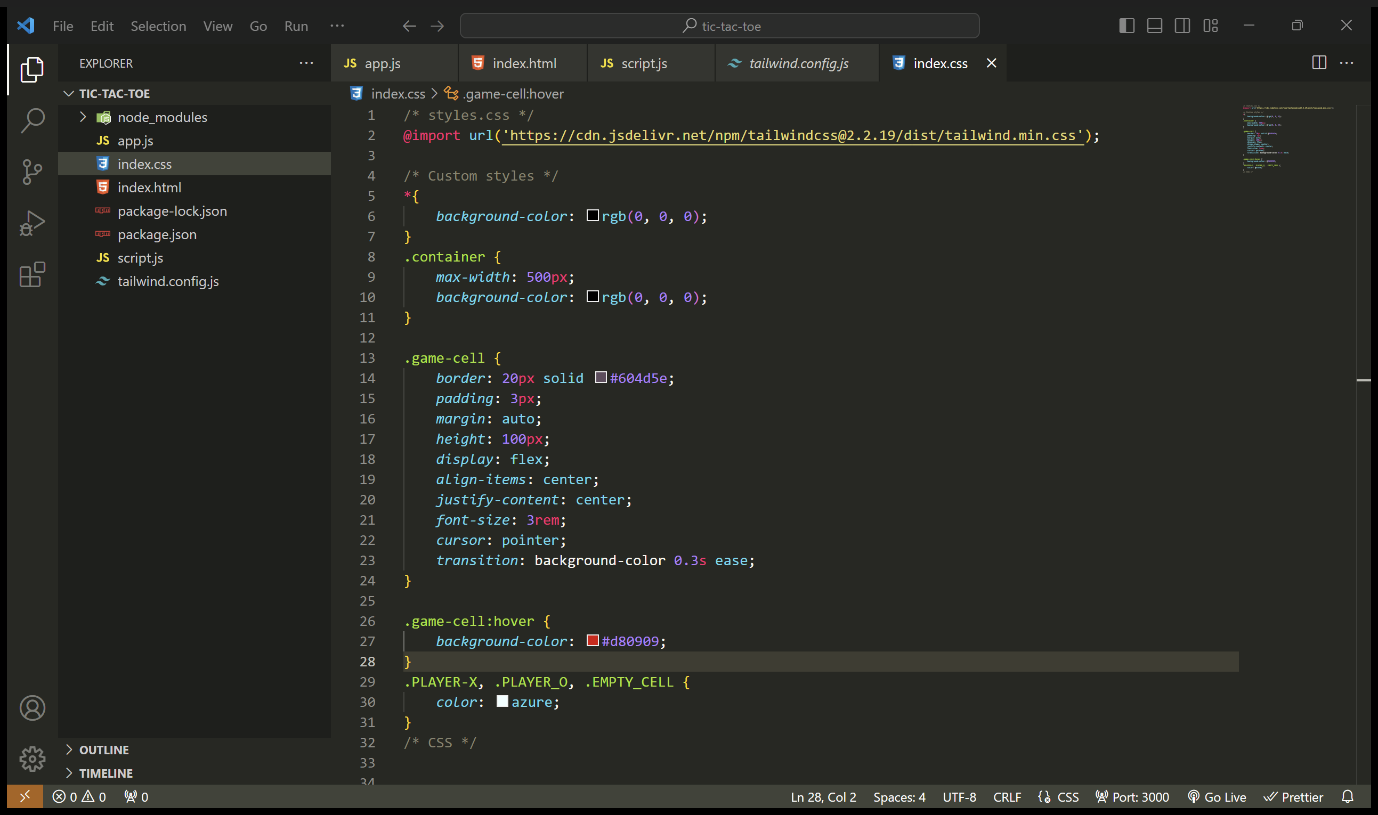
Top of Form

**CODE-**

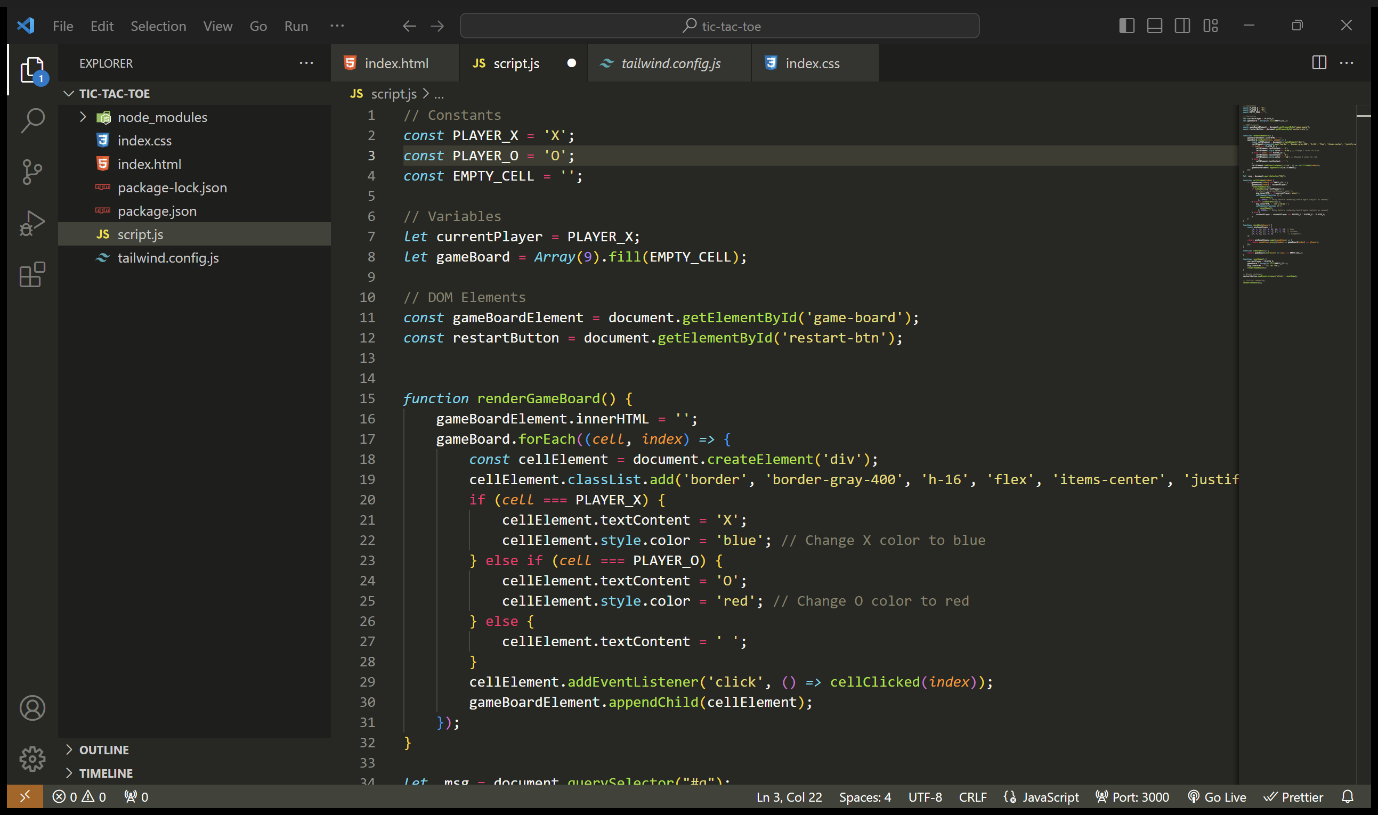
**HTML CODE-**

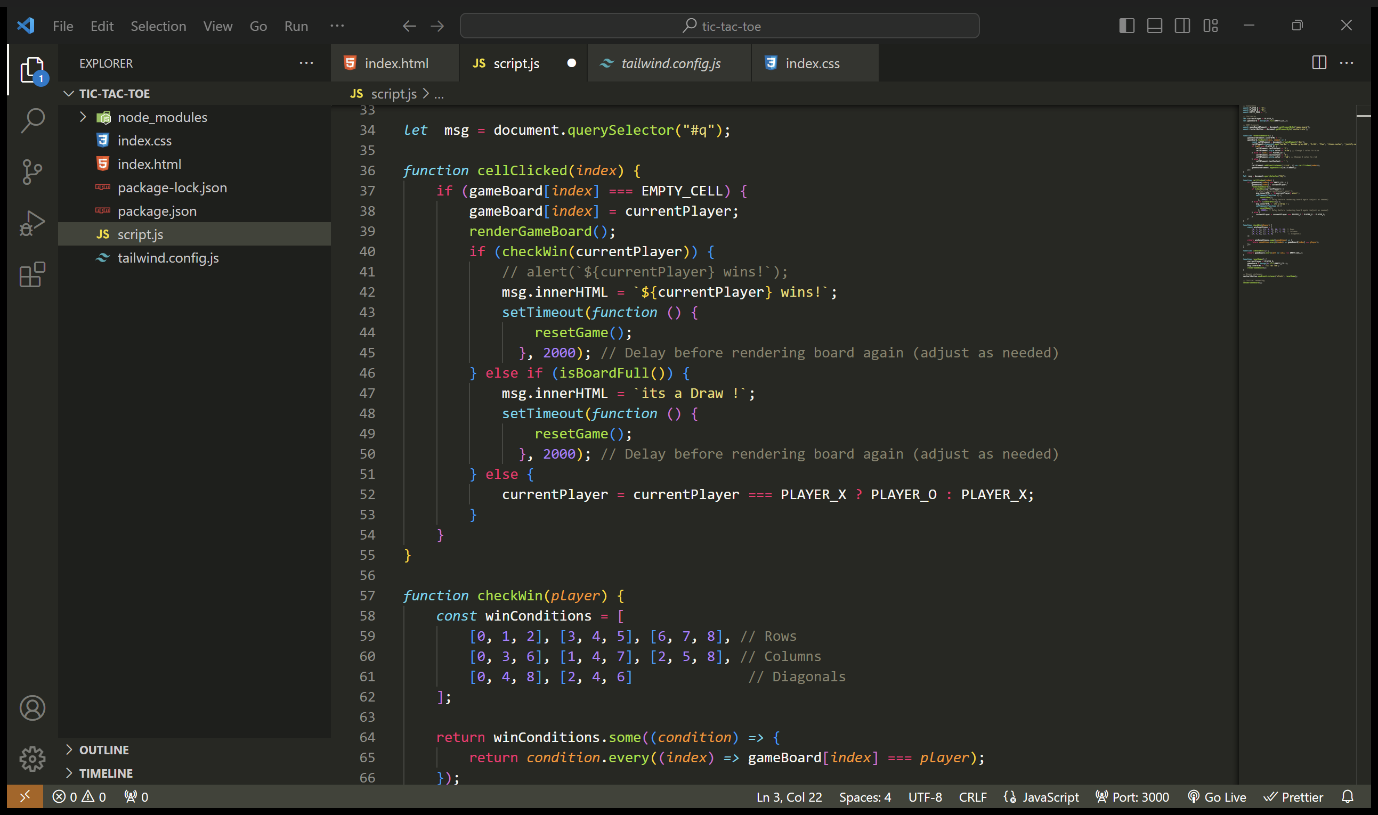
****

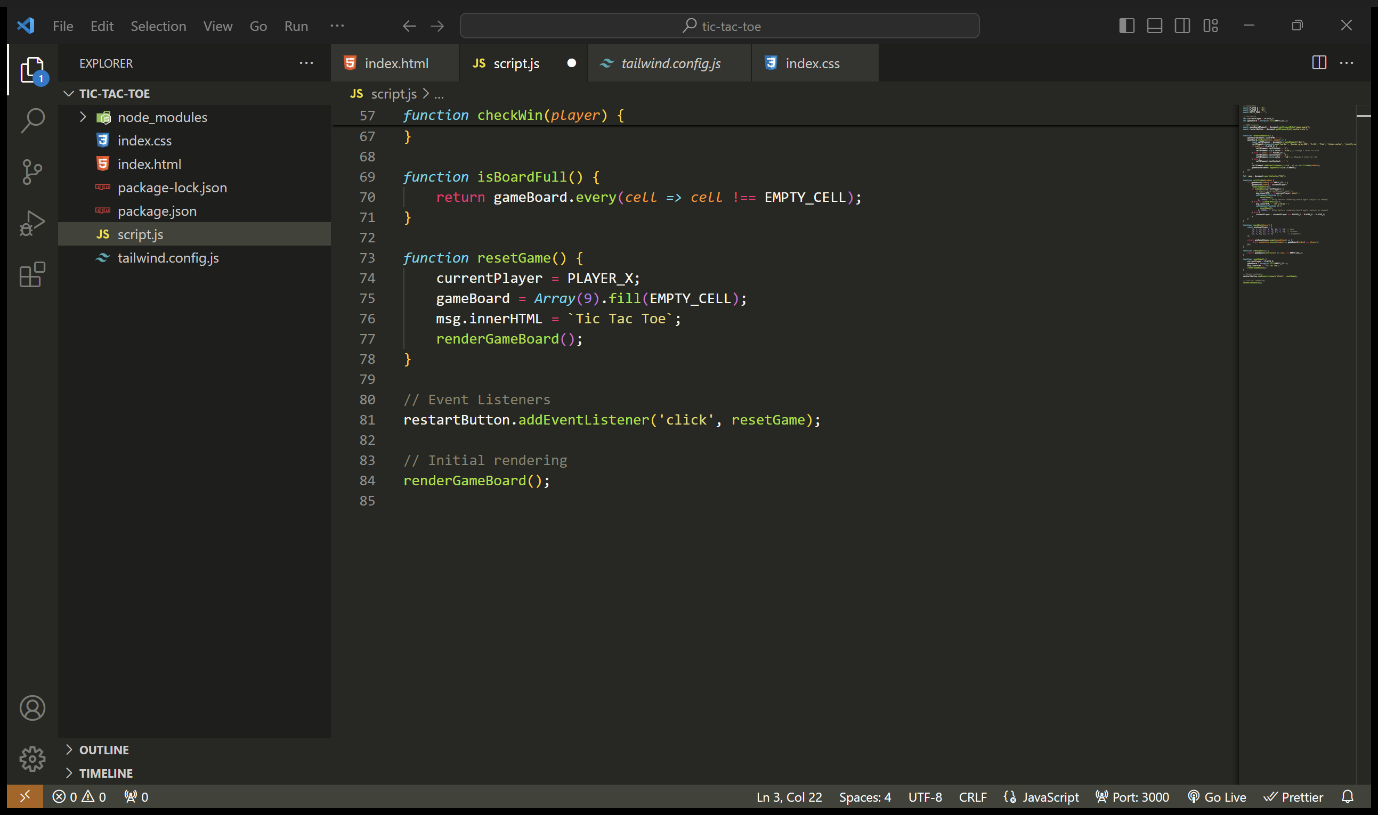
**CSS CODE-**

****

**JAVSCRIPT CODE-**

****

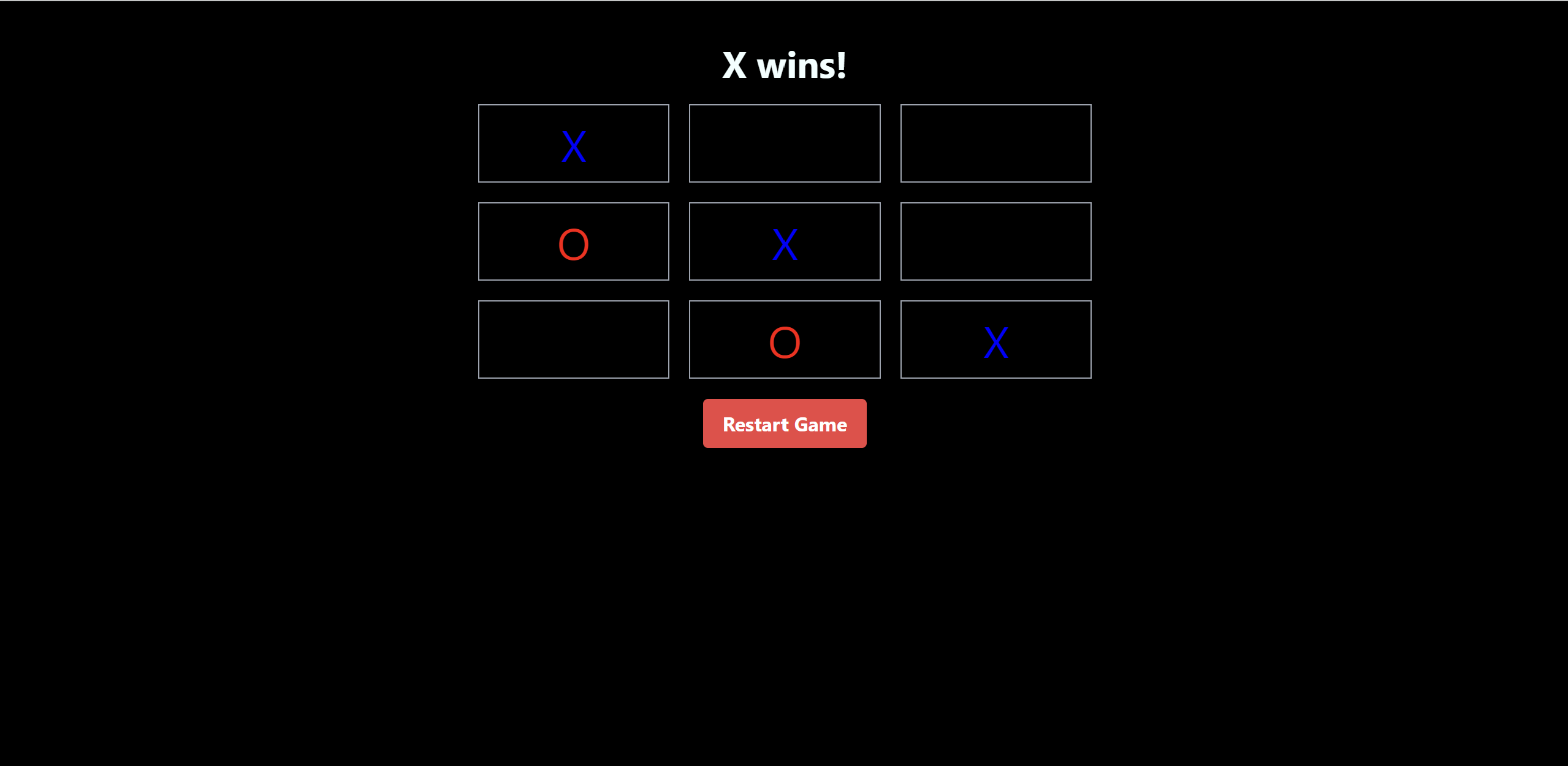
****

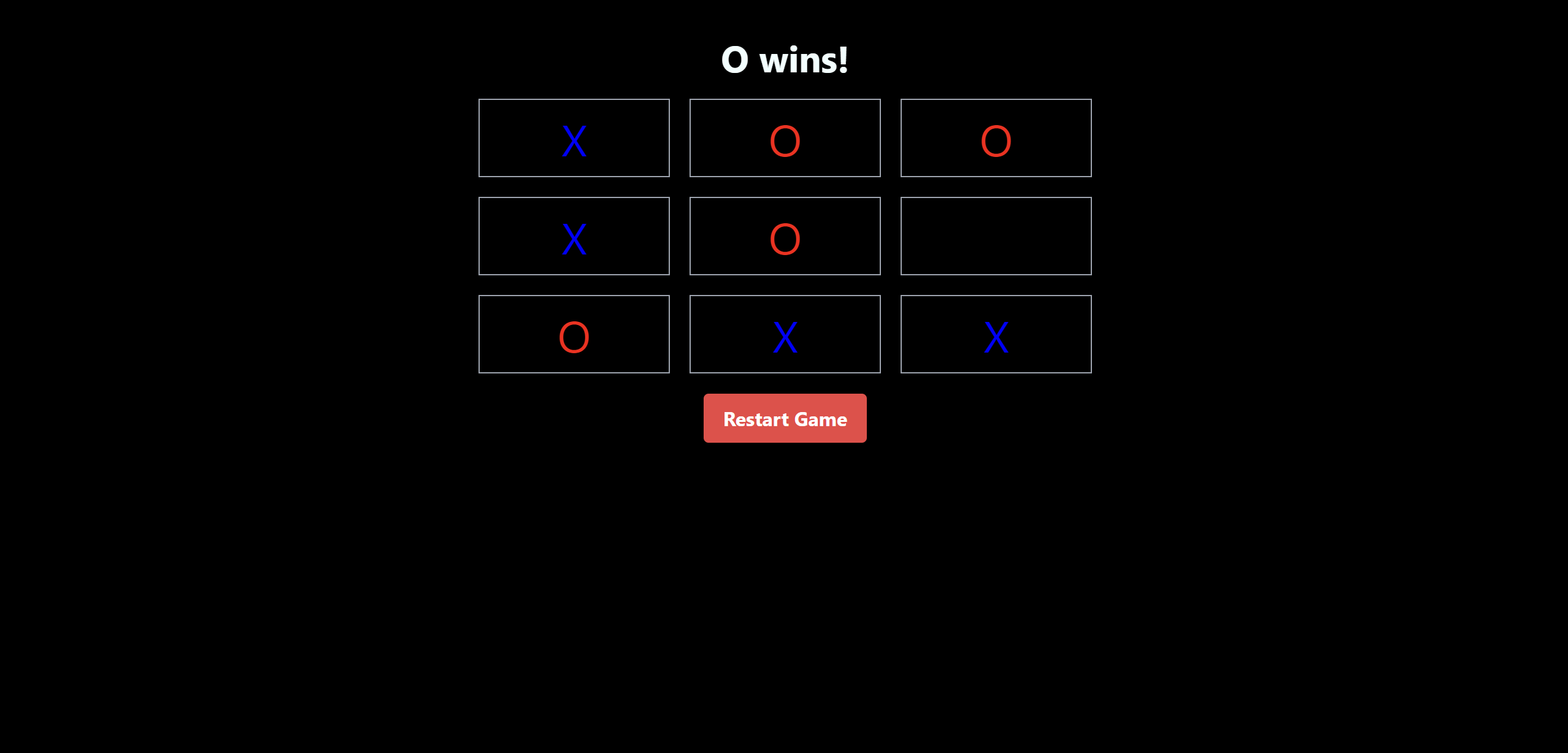
****

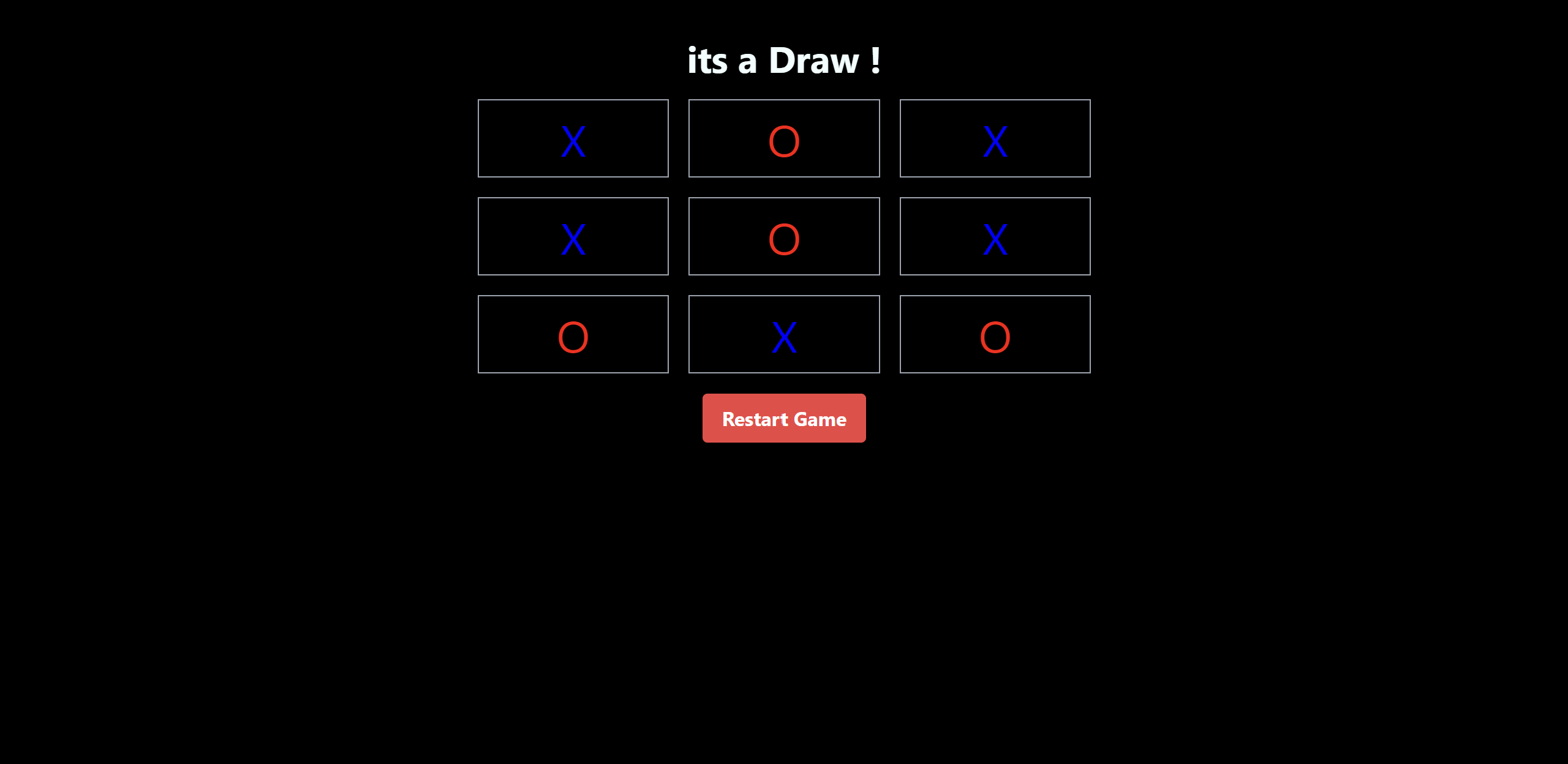
**BREAKDOWN OF JS CODE-**

1. **Constants:**
   * **PLAYER\_X**, **PLAYER\_O**: Constants representing the symbols for player X and player O.
   * **EMPTY\_CELL**: Constant representing an empty cell on the game board.
2. **Variables:**
   * **currentPlayer**: Variable to keep track of the current player (initially set to player X).
   * **gameBoard**: Array representing the state of the game board, initialized with empty cells.
3. **DOM Elements:**
   * **gameBoardElement**: Reference to the HTML element representing the game board.
   * **restartButton**: Reference to the HTML button for restarting the game.
4. **Rendering Function (renderGameBoard):**
   * Clears the existing game board HTML.
   * Iterates over the game board array and creates HTML elements for each cell.
   * Assigns appropriate styling and text content based on the cell's value (X, O, or empty).
   * Adds event listeners to each cell to handle click events.
5. **Click Handler Function (cellClicked):**
   * Handles click events on game board cells.
   * Checks if the clicked cell is empty.
   * Updates the game board state with the current player's symbol.
   * Renders the updated game board.
   * Checks for win conditions or a draw.
   * Switches the current player if the game is still ongoing.
6. **Win Condition Check Function (checkWin):**
   * Defines an array of win conditions (combinations of cell indices for winning lines).
   * Checks if any of the win conditions are met for the current player.
7. **Board Full Check Function (isBoardFull):**
   * Checks if all cells on the game board are filled with symbols (no empty cells remaining).
8. **Reset Game Function (resetGame):**
   * Resets the game state:
     + Sets the current player back to player X.
     + Resets the game board to all empty cells.
     + Clears any messages displayed.
     + Renders the updated game board.
9. **Event Listeners:**
   * Adds an event listener to the restart button to reset the game when clicked.
10. **Initial Rendering:**
    * Calls the **renderGameBoard** function to initialize the game board on page load.

**PREVIEW OF THE CODE-**

****

****

****

**REFERENCES-**

**TAILWIND-** <https://tailwindcss.com/docs/object-fit>

JAVA SCRIPT- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>