



REPORT ON
FRONT-END ENGINEERING PROJECT-
Tic-Tac-Toe Game using react js

Submitted By:

Akshat dhir

2110990125

Group- 19

Submitted To:

Mrs. pritpal kaur

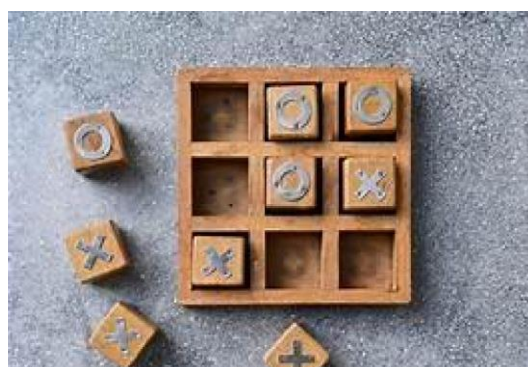
Department of Computer Science & Engineering
Chitkara University Institute of Engineering & Technology,
Rajpura, Punjab.

INDEX

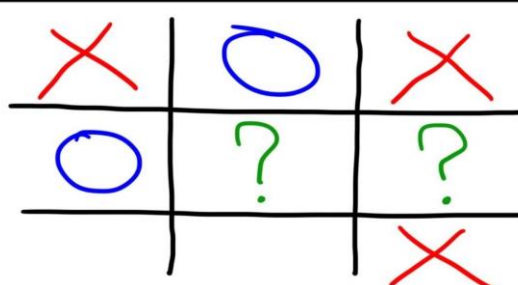
S. No.	Title	Page No.	Signature
1	Introduction		
2	Project Overview		
3	Project Components		
4	Counclusion		

Introduction

This is a project report on “ONLINE TIC TAC TOE GAME”



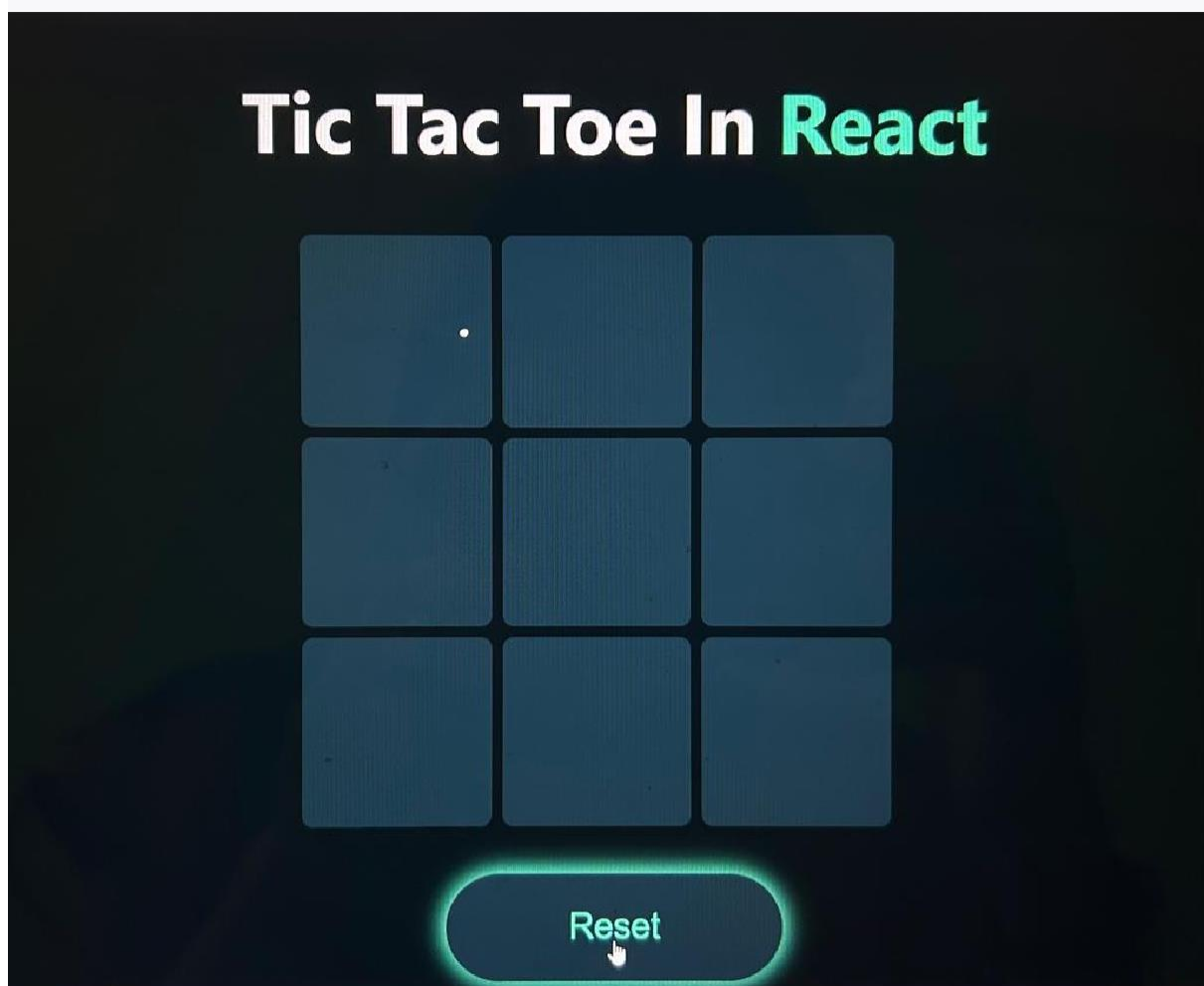
Where Should I Go?



This is a web based application which helps people to play “TIC-TAC-TOE” online. The game is very popular and is fairly simple by itself. In this game, there is a board with $n \times n$ squares. In our game ,it is 3×3 squares. The goal of Tic-Tac-Toe is to be one of the players to get three same symbols in a row-horizontally, vertically or diagonally- on a 3×3 grid.

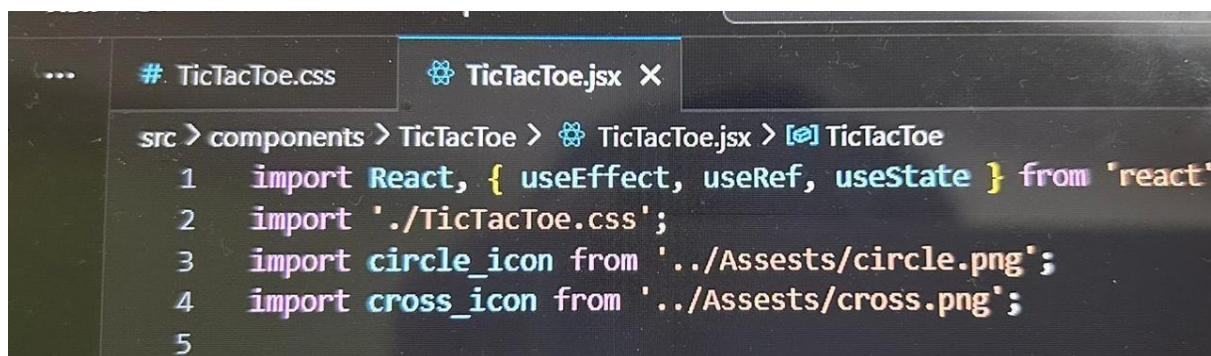
OVERVIEW

A player can choose between two symbols with his opponent, usual games use “X” and “O”. If first player choose “X” then the second player have to play with “O” and vice versa .A player marks any of the 3x3 squares with his symbol (may be “X” or “O”) and his aim is to create a straight line horizontally or vertically or diagonally with two intentions :a) Create a straight line before his opponent to win the game. b) Restrict his opponent from creating a straight line first. In case logically no one can create a straight line with his own symbol, the game results a tie. Hence there are only three possible results – a player wins, his opponent wins or it’s a tie.



Project Components

- **React Components:** The project consists of a single React component named `TicTacToe`. This component handles the game logic, rendering the game board, and managing game state.
- The Three hooks used are `useEffect`, `useRef` and `useState`.
- **CSS Styling:** The project uses CSS for styling. The `TicTacToe.css` file is used to define the visual layout and appearance of the game.
- **Assets:** The project includes image assets for the X and O icons.



```
src > components > TicTacToe > TicTacToe.jsx > TicTacToe
1  import React, { useEffect, useRef, useState } from 'react'
2  import './TicTacToe.css';
3  import circle_icon from '../Assets/circle.png';
4  import cross_icon from '../Assets/cross.png';
5
```

2. Game Logic The game logic is implemented within the `TicTacToe` component. Key components of the game logic include:

- Maintaining the game board state using the `data` state variable.
- Tracking the number of moves using the `count` state variable.
- Checking for a win or a draw using the `checkWin` function.
- Handling the end of the game and displaying the result.

3. Gameplay

- Players take turns to click on empty cells on the game board to place their 'X' or 'O'.
- The game keeps track of the number of moves and checks for a win or a draw after each move.

- When a player wins, the game displays a congratulatory message and locks further moves.
- If the game ends in a draw, a message is displayed, and further moves are locked.

4. User Interface

- The game interface is user-friendly and responsive.
- The title of the game is dynamically updated to reflect the game status, such as who won or if it's a draw.
- X and O icons are displayed in the cells to represent the moves of the players.

5. Reset Functionality

- The 'Reset' button allows users to restart the game. When pressed, it resets the game state and clears the innerHTML of the cells to start a new game.

```
const reset = () => {
  setLock(false);
  setWinner(null);
  setData(Array(9).fill(''));
  setTitle('Tic Tac Toe In <span>React</span>');
  // Clear innerHTML after setting the state
  setTimeout(() => {
    boxArr.forEach((ref) => {
      if (ref.current) {
        ref.current.innerHTML = '';
      }
    });
  }, 0);
};
```

6. Refactoring and Best Practices

- The code structure is well-organized, making use of React state management and useEffect for checking the game state.
- Variables and functions are named descriptively, enhancing code readability.
- The use of ref callbacks to access DOM elements for updating innerHTML is in line with React best practices.

```
new Go Run Terminal Help
# TicTacToe.css TicTacToe.jsx
src > components > TicTacToe > TicTacToe.jsx > toggle
15
16 useEffect(() => {
17   checkWin();
18 }, [count]);
19
20 const toggle = (num) => {
21   if (lock || data[num] !== '') {
22     return;
23   }
24   const newData = [...data];
25   newData[num] = count % 2 === 0 ? 'x' : 'o';
26   setCount((prevCount) => {
27     setData(newData);
28     return prevCount + 1;
29   });
30 };
31
32 const checkWin = () => {
33   const winPatterns = [
34     [0, 1, 2],
35     [3, 4, 5],
36     [6, 7, 8], // rows
37     [0, 3, 6],
38     [1, 4, 7], // columns
39     [2, 5, 8],
40     [0, 4, 8],
41     [2, 4, 6], // diagonals
42   ];
43 }
```

7. Future Improvements

- Implementing a scoring system to keep track of wins for each player.
- Enhancing the user interface with more interactive features.
- Implementing an option to play against an AI opponent.

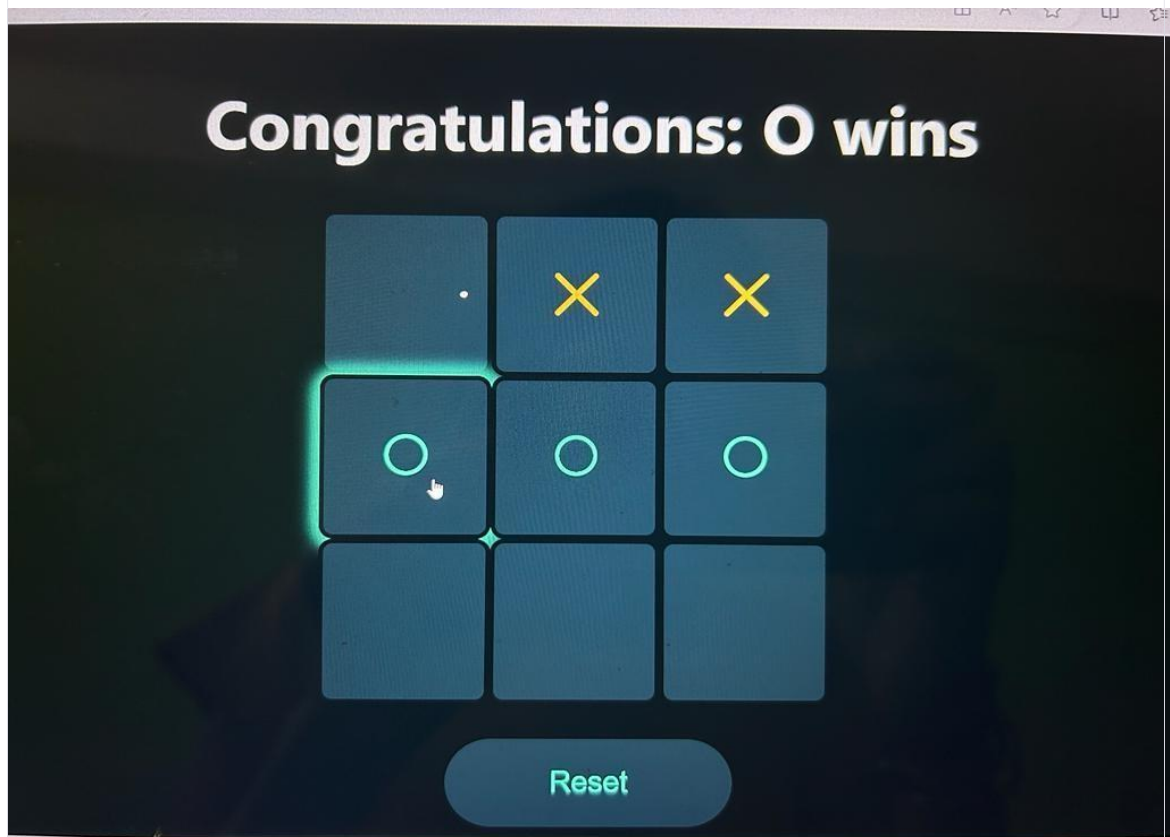
8. **Conclusion** The Tic Tac Toe game in React is a well-implemented project that provides an enjoyable gaming experience. It demonstrates the use of React for building interactive web applications and serves as a foundation for potential enhancements and features.

9. **Project Repository** The project's source code is hosted on a code repository. Developers and contributors can find the project's code and resources there for further study and collaboration.

GAME OUTCOMES

1) If one of the two players wins:

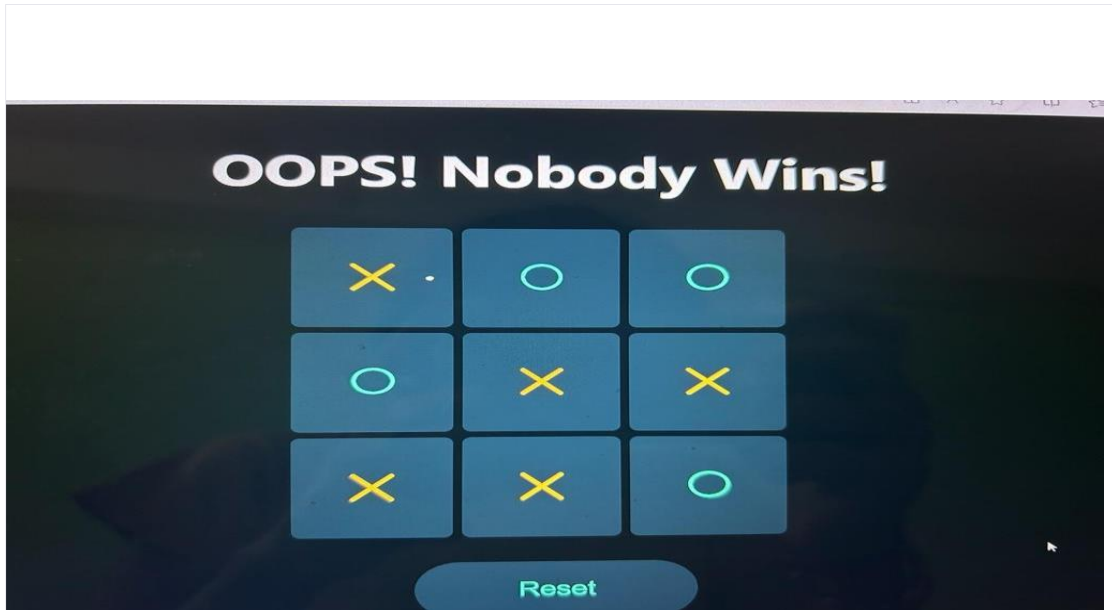
```
const handleWin = (winner) => {
  setLock(true);
  const message = winner === 'x' ? 'Congratulations: X wins' : 'Congratulations: O wins';
  setTitle(message);
  setWinner(winner);
};
```



2) If it's a tie:

```
if (data.every((cell) => cell !== '') && !winner) {
  setLock(true);
  setTitle('OOPS! Nobody Wins!');
}
;
```

I



Please note that this project report provides an overview of the Tic Tac Toe game in React, highlighting its key components, features, and potential for future improvements.