SOLUTION – CODING ASSIGNMENT 6.1

App.js

import "./tictac.css";

import { useState } from "react";

export default *function* App3() {

*const* [squares, setSquares] = useState(*Array*(9));

*const* nextValue = Nextvalue(squares);

*const* winner = Possibility(squares);

*const* status = Status(winner, squares, nextValue);

*const* finalWinner=final(winner, squares, nextValue)

*function* defaultSquare(*square*) {

    if (winner || squares[*square*]) {

      return;

    }

*const* squaresCopy = [...squares];

    squaresCopy[*square*] = nextValue;

    setSquares(squaresCopy);

  }

*function* renderSquare(*i*) {

    return (

      <button className="square" action={squares[*i*] ? squares[*i*]:"+"} onClick={() *=>* defaultSquare(*i*)}>

        <h1 className="txt" >{squares[*i*] ? squares[*i*]:"+"}</h1>

      </button>

    );

  }

*function* Status(*winner*, *squares*, *nextValue*) {

    return *winner*

      ? `Next player: Player ${*nextValue*}`

      : *squares*.every(*Boolean*)

      ? `Next Player:Player X`

      : `Next player: Player ${*nextValue*}`;

  }

*function* final(*winner*, *squares*, *nextValue*){

    return *winner*

    ?`Winner is Player ${*winner*}`

    :`Winner is No Winner Yet`

  }

*function* Nextvalue(*squares*) {

    return *squares*.filter(*Boolean*).length % 2 === 0 ? "X" : "O";

  }

*function* Possibility(*squares*) {

*const* lines = [

      [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 < lines.length; i++) {

*const* [tic, tac, toe] = lines[i];

      if (*squares*[tic] && *squares*[tic] === *squares*[tac] && *squares*[tic] === *squares*[toe]) {

        return *squares*[tic];

      }

    }

  }

  return (

    <>

    <div

      className="game-board">

<div className="grid-row" >

        { renderSquare(0)}

        {  renderSquare(1)}

        { renderSquare(2)}

      </div>

      <div className="grid-row">

        { renderSquare(3)}

        {  renderSquare(4)}

        { renderSquare(5)}

      </div>

      <div className="grid-row">

        { renderSquare(6)}

        {  renderSquare(7)}

        { renderSquare(8)}

      </div>

      <div>

      <h1 className="status"> {status}</h1>

      <h1  className="status">{finalWinner}

      </h1>

      </div>

    </div>

    </>

  );

}

Tictac.css

body{

*margin-top*: 1%;

}

.App {

*font-family*: sans-serif;

*text-align*: center;

}

button {

*background-color*: #0c80f3;

*border*: 6px solid #0ff166;

*border-radius*: 10px;

}

.game-board {

*width*: 600px;

*height*: 600px;

*margin*: 0 auto;

*background-color*:gray ;

*color*: rgb(204, 28, 28);

*border*: 6px solid #0b0d0f;

*border-radius*: 10px;

*display*: grid;

*grid-template-rows*: 1fr 1fr 1fr;

*grid-template-columns*: 1fr;

}

.grid-row {

*border*: 6px solid #739dc7;

*border-radius*: 6px;

*font-family*: Helvetica;

*font-weight*: bold;

*font-size*: 4em;

*display*: grid;

*grid-template-rows*: 1fr;

*grid-template-columns*: 1fr 1fr 1fr;

*position*: sticky;

*background-color*: rgb(175, 70, 70);

}

.block {

*background-color*: rgb(219, 28, 28);

*border*: 6px solid green;

*border-radius*: 6px;

*font-family*: Helvetica;

*font-weight*: bold;

*font-size*: 20em;

}

Sample Screenshot





