PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Technology in Information Technology

By

AMIRNENI KRISHNA TEJA

(REG NO : 39120008)

DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI-600119, TAMIL NADU

April 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Amirneni Krishna teja(Reg. No: 39120008)** who carried out the project entitled "**Web Based Game With Two Players(XOX)**" under my supervision from February 2022 to April 2022.

Internal Guide

Dr.R.M.Gomathi ,M.Tech., Ph.D.

Head of the Department

Dr.R.SUBHASHINI, M.Tech.,Ph.D.

Submitted for Viva Voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I , **Amirneni Krishna teja** hereby declare that the project report entitled "**Web Based Game With Two Members"** done by me under the guidance of **Dr.R.M.Gomathi, Associate Professor IT Dept,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information technology.

DATE: Amirneni Krishna teja

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board **of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing**. Dr.Subhashini , M.E., Ph.D., Head of the Department** of **Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R.M.Gomathi, Associate Professor IT Dept.,** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

The game of Tic-tac-toe is one of the most commonly known games. This game does not allow one to win all the time and a significant proportion of games played results in a draw. Thus, the best a player can hope is to not lose the game. This study is aimed at evolving a number of no-loss strategies using genetic algorithms and comparing them with existing methodologies. To efficiently evolve no-loss strategies, we have developed innovative ways of representing and evaluating a solution, initializing the GA population, developing GA operators including an elite preserving scheme. interestingly, our GA implementation is able to find more than !" thousand no-loss strategies for playing the game. moreover, an analysis of these solutions has given us insights about how to play the game to not lose it. Based on this experience, we have developed specialized efficient strategies having a high win-to-draw ratio. The study and its results are interesting and can be encouraging for the techniques to be applied to other board games for finding efficient strategies.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE CONCEPT

In the fast growing field of software engineering and development and even more rapidly growing sector of game development the future is hard to predict. In general a software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a game project, the product is a game. But here comes the point: A game is much more than just its software. It has to provide content to become enjoyable. Just like a web server: without content the server is useless, and the quality cannot be measured. This has an important effect on the game project as a whole. The software part of the project is not the only one, and it must be considered in connection to all other parts: The environment of the game, the story, characters, game plays, the artwork, and so on.

Tic-tac-toe (American English), noughts and crosses (Commonwealth English), or Xs and Os (Irish English) is a paper-and-pencil game for two players who take turns marking the spaces in a three-by-three grid with *X* or *O*. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game, with a forced draw assuming best play from both players.

CHAPTER 2

AIM AND GAME RULES

2.1 AIM

Our objective is to evolve several Tic-tac-toe(XOX) strategies which will be played by two players. This makes the problem have a single objective of minimizing the number of losses. The evaluation of fitness of any strategy is done by first allowing it to play all possible games it could play, both as a first player and as a second player. For example, note from figure 1,2,3 that there are two possible ways a game can move for the first player and second player, depending on whether the opponent made the left or the right side move .The opponent also selects according to their choice. At the end displays the winner and gives a chance to reset.

2.2 GAME RULES

The game rules of XOX (tic tac toe) are that

(1)Each player has to select either x or o according to their choice. For example figure 1 the player has to select

(2) The grid contains of 3X3 matrix the player x and player o gets chance to select their element for examples Figure (2) and figure(3)

(3) Generally, for the sake of fair, each competition will run two times, thus each side will have chance been the first player or the second player, since being the first or the second will have the different advantages

FIG 2.1 player x turn

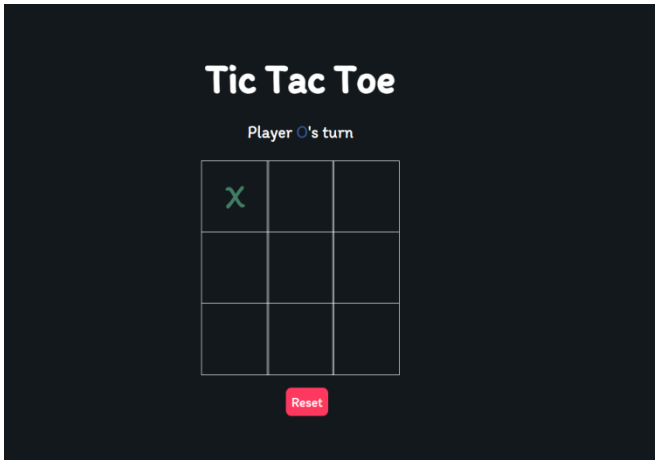

FIG 2.2 player o turn

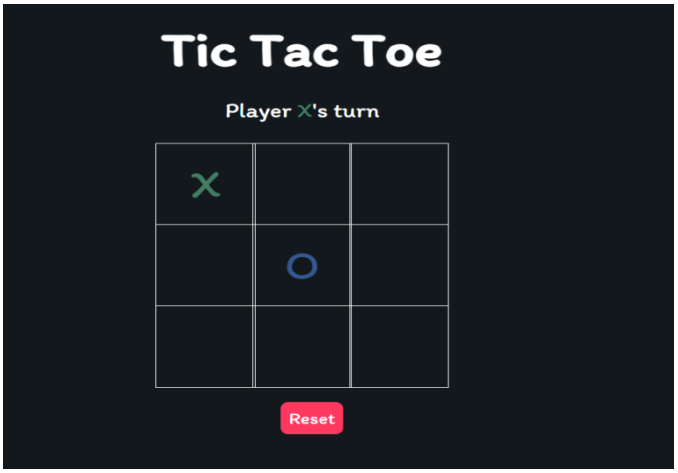

FIG2.3 player x turn

CHAPTER 3

3 Software Requirement Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the Game Development. It illustrate the purpose and complete description for the development of the system. It explains system constraints, interface. This document is primarily intended to be proposed for digital games and a reference for the upcoming students of the college.

3.2 Scope

Our project has made it easier for everyone to create their own digital games. Playing video game fulfills a purpose in their lives. This could include gaming for: relaxation, opportunities to exert control, enjoyment, creativity, socialization, prevent boredom, challenge, and achievement. It could also be used as a coping method or stress management.

3.3 Overview

The purpose of this document is to help the reader to visualize the solution to the project presented.

3.3.1 Product Perspective

Our project is mainly focused on entertainment purposes. This project can be implemented in any computer.

3.3.2 Product Functionality

In XOX two players can play at a time, where each player can handle a game at a time.

3.3.3 Users and Characteristics

In XOX there are two players each who can control their elements x and o to the opponent's side.

3.4 Specific Requirements

3.4.1 Hardware Requirements

- System: IBM-Compatible PC
- Processor: Intel® Pentium® IV 2.4 GHz or AMD 3500+
- Speed: Above 1GHz
- RAM capacity:2 GB
- Graphics: OpenGL 2.1 or OpenGL ES 2
- Keyboard: Standard
- Mouse: Standard

3.4.2 Software requirements

- 4 gb ram
- Operating system
- Windows 10
- Intel i3,10 generation

4 .TECHNOLOGIES

4.1.HTML: Hyper Text Markup Language

(Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content.

Other technologies besides HTML are generally used to describe a web page's

appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web.

By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , , <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video>, <ul>, <ol>, <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">".

The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents.

first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGML guid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

4.2.CSS: Cascading Style Sheets

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the and is standardized across Web browsers according to W3C specifications.

Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations.

You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

From CSS3, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much that it became more effective to develop and release recommendations separately per module.

Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

Inheritance in CSS is not the same as inheritance in class-based programming languages, where it is possible to define class B as "like class A, but with modifications". With CSS, it is possible to style an element with "class A, but with modifications". However, it is not possible to define a CSS class B like that, which could then be used to style multiple elements without having to repeat the modifications.

4.3.JavaScipt(JS)

JS is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions.

While it is most well-known as the scripting language for Web pages, many non browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat.

JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments.

For information about API specifics to Web pages, please see Web APIs and DOM.

The standards for JavaScript are the ECMAScript Language Specification (ECMA 262) and the ECMAScript Internationalization API specification (ECMA-402).

The JavaScript documentation throughout MDN is based on the latest draft versions of ECMA-262 and ECMA-402.

And in cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features.

Do not confuse JavaScript with the Java programming language.

Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries.

However, the two programming languages have very different syntax, semantics, and use.

5 RESULT

5.1 welcome screen GUI

FIG 5.1 welcome screen

5.2 Game page at start

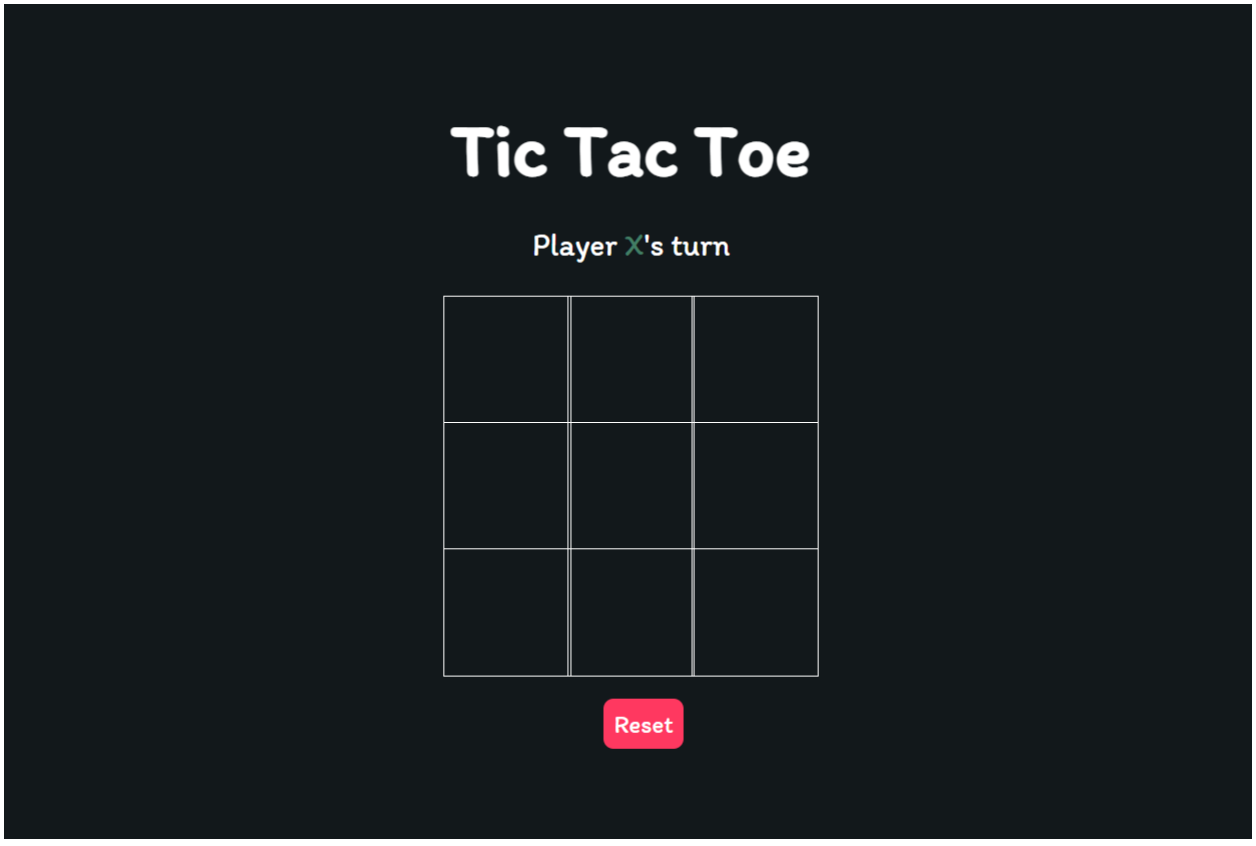

FIG 5.2 after gui screen

5.3 Game page at end

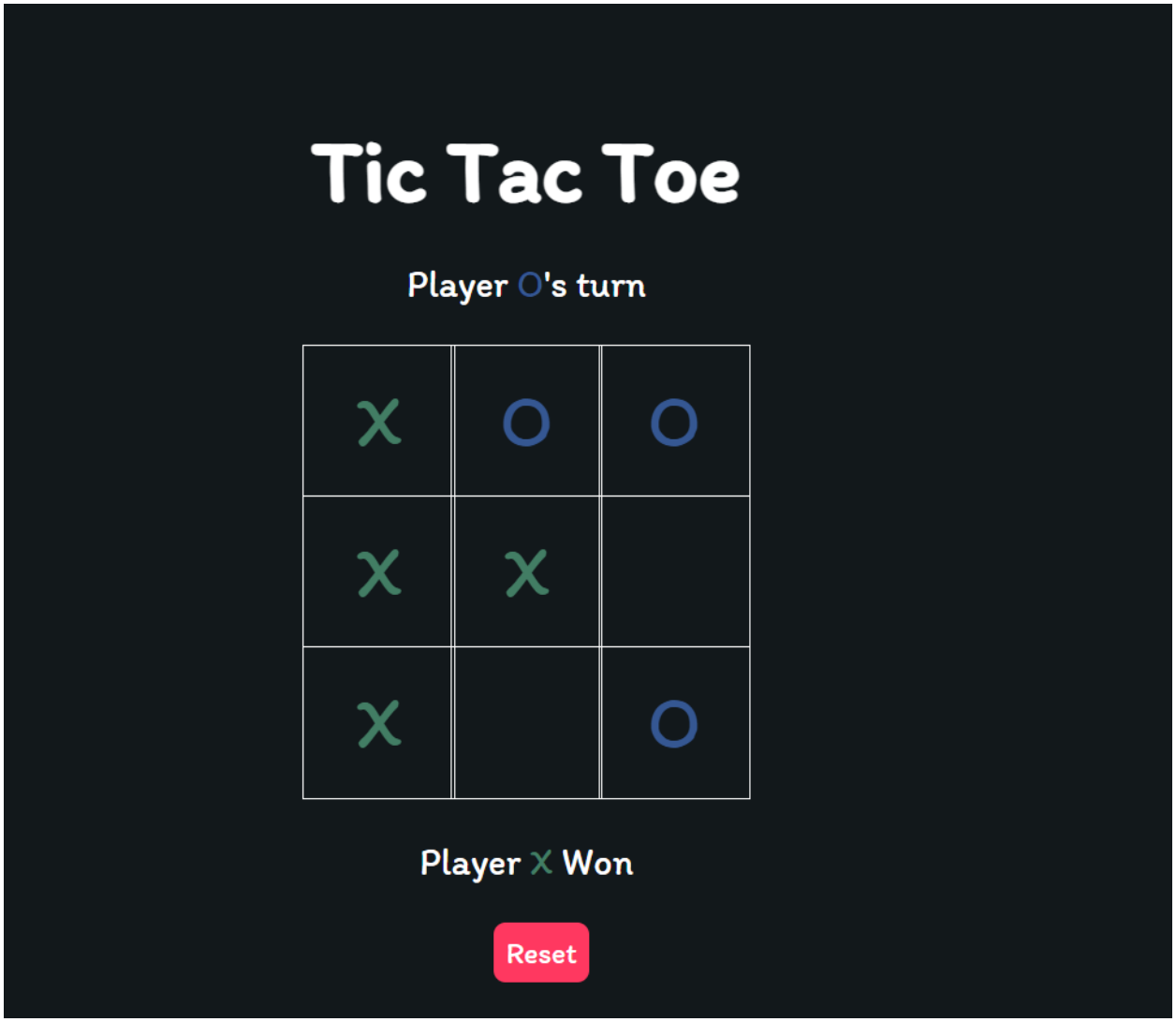

FIG 5.3 game at end page

6 Conclusion

The Tic Tac Toe game is most familiar among all the age groups. Intelligence can be a property of any purpose-driven decision maker. This basic idea has been suggested many times. An algorithm of playing Tic Tac Toe has been presented and tested that works in an efficient way. Overall the system works without any bugs.

7 SOURCE CODE

7.1 WELCOME SCREEN HTML CODE:

```
<!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">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lobster&display=swap"
rel="stylesheet">
   <link rel="stylesheet" href="style2.css">
   <div><span class="wlc" >WELCOME TO</span> <span class=``ping"
>XOX </span> </div>

<div class="credit">Created By <a
href=``https://www.linkedin.com/in/krishnateja-amirneni-6943a4219"
target=``_blank">KRISHNA TEJA</a></div>
   </div>
    <a href="./index.html"><button>Start the game</button>  </a>
</head>
<body>
```

```
</body>
</html>
```

7.2 WELCOME SCREEN CSS CODE:

```
   nt-family: 'Lobster', cursive;
   text-align: left;
   font-size: 80px;
   margin-top: 10%;
   border-color: black 15px;
}

.wlc{
   color: rgb(231, 31, 31);

}

.ping{

   color: antiquewhite;
}
.credit{
   font-size:x-large;
   margin-left: 25%;
   color: antiquewhite;
   font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif;

}
```

```
button{


    margin-left: 32%;

    margin-top: 120px;

    color: black;

    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;

    font-size: 20px;

    background-color: rgb(111, 150, 223);

}


body{

    background-color: #12181B;

    background-repeat: no-repeat;

    background-size: 1280px;

    background-position: center;
```

7.3 XOX GAME HTML CODE:

```
<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">
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Itim&display=swap" rel="stylesheet">
    <script src="./index.js"></script>
    <title>XOX</title>
</head>
<body>
    <main class="background">
        <section class="title">
            <h1>XOX</h1>
        </section>
        <section class="display">
            Player <span class="display-player playerX">X</span>'s turn
        </section>
        <section class="container">
            <div class="tile"></div>
            <div class="tile"></div>
            <div class="tile"></div>
```

```
            <div class="tile"></div>
            <div class="tile"></div>
            <div class="tile"></div>
            <div class="tile"></div>
            <div class="tile"></div>
            <div class="tile"></div>
        </section>
        <section class="display announcer hide"></section>
        <section class="controls">
            <button id="reset">Reset</button>
        </section>
    </main>
</body>
</html>
```

7.4 XOX GAME CSS CODE:

```
* {
    padding: 0;
    margin: 0;
    font-family: 'Itim', cursive;
}

.background {
    background-color: #12181B;
    height: 100vh;
    padding-top: 1px;
}
```

```
.title {
    color: white;
    text-align: center;
    font-size: 40px;
    margin-top: 10%;
}

.display {
    color: white;
    font-size: 25px;
    text-align: center;
    margin-top: 1em;
    margin-bottom: 1em;
}

.hide {
    display: none;
}

.container {
    margin: 0 auto;
    display: grid;
    grid-template-columns: 33% 33% 33%;
    grid-template-rows: 33% 33% 33%;
    max-width: 300px;

}
```

```
.tile {
    border: 1px solid white;
    min-width: 100px;
    min-height: 100px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 50px;
    cursor: pointer;
}

.playerX {
    color: #427d64;
}

.playerO {
    color: #355792;
}

.controls {
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
    margin-top: 1em;
}

.controls button {
```

```
    color: white;
    padding: 8px;
    border-radius: 8px;
    border: none;
    font-size: 20px;
    margin-left: 1em;
    cursor: pointer;
}

.restart {
    background-color: #498AFB;
}

#reset {
    background-color: #FF3860;
}
```

7.5 XOX GAME JAVA SCRIPT:

```
 window.addEventListener('DOMContentLoaded', () => {
const tiles = Array.from(document.querySelectorAll('.tile'));
const playerDisplay = document.querySelector('.display-player');
const resetButton = document.querySelector('#reset');
const announcer = document.querySelector('.announcer');

let board = ['', '', '', '', '', '', '', '', ''];
let currentPlayer = 'X';
let isGameActive = true;

const PLAYERX_WON = 'PLAYERX_WON';
const PLAYERO_WON = 'PLAYERO_WON';
const TIE = 'TIE';



/*
   Indexes within the board
   [0] [1] [2]
   [3] [4] [5]
   [6] [7] [8]
*/

const winningConditions = [
```

```
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6]
];

function handleResultValidation() {
    let roundWon = false;
    for (let i = 0; i <= 7; i++) {
        const winCondition = winningConditions[i];
        const a = board[winCondition[0]];
        const b = board[winCondition[1]];
        const c = board[winCondition[2]];
        if (a === '' || b === '' || c === '') {
            continue;
        }
        if (a === b && b === c) {
            roundWon = true;
            break;
        }
    }

if (roundWon) {
        announce(currentPlayer === 'X' ? PLAYERX_WON : PLAYERO_WON);
```

```
        isGameActive = false;
        return;
    }

if (!board.includes(''))
    announce(TIE);
}

const announce = (type) => {
    switch(type){
        case PLAYERO_WON:
            announcer.innerHTML = 'Player <span class="playerO">O</span> Won';
            break;
        case PLAYERX_WON:
            announcer.innerHTML = 'Player <span class="playerX">X</span> Won';
            break;
        case TIE:
            announcer.innerText = 'Tie';
    }
    announcer.classList.remove('hide');
};

const isValidAction = (tile) => {
     if (tile.innerText === 'X' || tile.innerText === 'O'){
       return false;
    }

   return true;
```

```
};

 const updateBoard =  (index) => {
     board[index] = currentPlayer;
}

const changePlayer = () => {
    playerDisplay.classList.remove(`player${currentPlayer}`);
    currentPlayer = currentPlayer === 'X' ? 'O' : 'X';
    playerDisplay.innerText = currentPlayer;
    playerDisplay.classList.add(`player${currentPlayer}`);
}

const userAction = (tile, index) => {
    if(isValidAction(tile) && isGameActive) {
        tile.innerText = currentPlayer;
        tile.classList.add(`player${currentPlayer}`);
        updateBoard(index);
        handleResultValidation();
        changePlayer();
    }
}

const resetBoard = () => {
    board = ['', '', '', '', '', '', '', '', ''];
     isGameActive = true;
    announcer.classList.add('hide');
```

```
        if (currentPlayer === 'O') {
            changePlayer();
        }

        tiles.forEach(tile => {
            tile.innerText = '';
            tile.classList.remove('playerX');
            tile.classList.remove('playerO');
        });
    }

    tiles.forEach( (tile, index) => {
        tile.addEventListener('click', () => userAction(tile, index));
    });

    resetButton.addEventListener('click', resetBoard);
});
```