



UNIVERSITETI POLITEKNIK I TIRANES
FAKULTETI I TEKNOLOGJISE SE INFORMACIONIT
DEGA: INXHINIERI INFORMATIKE

Punë Laboratori Nr 3

Lënda: Programim Web

Tema: Krijimi i lojes Tic-tac-toe duke përdorur HTML,
CSS dhe JavaScript

Grupi: INF III-A

Punoi: Fabian Zerka

Pranoi: Msc Eduart Torba

Viti akademik 2024-2025

1. Pershkrimi i lojes: Të krijohet loja Tic-tac-toe duke përdorur HTML, CSS dhe JavaScript.

Loja mund të luhet me kompjuterin ose mes dy lojtarëve. Sapo të hapet faqja, përdoruesit do t'i kërkohet të zgjedhë nëse do të luajë me një lojtar tjetër apo do të luajë me kompjuterin. Nëse përdoruesi zgjedh të luajë me kompjuterin, atëherë lëvizja e kompjuterit do të jetë automatike dhe do të bëhet në mënyrë të rastësishme në një nga hapësirat e lira në lojë. Pas zgjedhjes së parë, do të shfaqet një prompt tjetër në ekran, ku do të vendosen emrat e lojtarëve dhe më pas do të nisët loja. Në një seksion të caktuar në faqe do të shfaqet statusi i lojës (p.sh., "Rradhën e lojtarit X", "Lojtari X fitoi", "Barazim"). Do të shfaqet një buton për të rinisur lojën, i cili bëhet i klikueshëm pasi loja ka përfunduar dhe rezultati është afishuar.

2. Llogjika e lojes te vendoset ne nje file me emrin “script.js”:

1. Zbatoni një sistem për të ndërruar mes dy lojtarëve (Lojtari 1, Lojtari 2 ose Kompjuteri). Lojtari 1 përdor "X" dhe Lojtari 2 përdor "O".
2. Kur një lojtar klikon në një qelizë të tabelës, duhet të shfaqet "X" ose "O" bazuar në atë që është radha e tij.
3. Zbuloni fituesin pasi një lojtar të mbushë një rresht, një kolonë ose një diagonale me simbolin e tij ("X" ose "O").
4. Nëse tabela është plot dhe nuk ka fitues, shfaqni një mesazh që tregon se ka ndodhur një barazim.
5. Kur një lojtar fiton ose ndodh një barazim, shfaqni një mesazh me emrin e fituesit dhe butoni luaj perseri behet I klikueshem. Kur klikohet butoni shfaqet nje pop up ne ekran me shkrimin: "Lojtari X fitoi! Dëshironi të luani përsëri?" dhe optionet Po, Jo.
6. Kur lojtari zgjidh "Po" pas përfundimit të lojës, beni reset tabelen e rezultateve dhe filloni një lojë të re.

3. Stilizimi: Krijoni një stylesheet të jashtëm për

1. Sigurohuni që tabela të duket e qartë, e thjeshtë për tu kuptuar që kemi të bëjmë me lojën tic tac toe dhe e pozicionuar në qendër të faqes.
2. Secila qelizë në grid duhet të jetë mjaft e madhe për të shfaqur një "X" ose "O" dhe të pozicionohet në mes të qelizës.
3. Përdorni ngjyra të ndryshme për lojtarin "X" dhe lojtarin "O".
4. Dorezimi (02/11/2024 – 13/11/2024): Te dorezohet kodi se bashku me relacionin mbi lojën, ku te sqarohet se si është implementuar loja. Te vendosen screenshots në relacionin tuaj.

File-I index.html permban:

1. Struktura bazë e dokumentit

- `<title>`: Vendosi titullin e faqes në shiritin e shfletuesit ("Tic-Tac-Toe").
- `<link rel="stylesheet" href="index.css" />`: Lidh dokumentin HTML me skedarin e stilimit index.css.

2. Pjesa kryesore e faqes (<body>)

1. Seksioni 1: Zgjedhja e mënyrës së lojës (#main)

- `<section id="main">`: Seksioni fillestar ku përdoruesi zgjedh mënyrën e lojës.
- `<h1>`: Mirëpret lojtarët me mesazhin "Welcome to Tic-Tac-Toe!".
- `<h2>`: Fton përdoruesit të zgjedhin modalitetin e lojës.
- `<button id="singleplayer">1 Player</button>`: Butoni për të zgjedhur modalitetin me një lojtar.
- `<button id="twooplayer">2 Players</button>`: Butoni për modalitetin me dy lojtarë.

2. Seksioni 2: Futja e emrave të lojtarëve (#player-names)

- `<section id="player-names" style="display: none">`: Seksioni ku lojtarët futin emrat, që është fillimisht i fshehur.
- `<form id="player-form">`: Formulari për futjen e emrave të lojtarëve.
- `<div id="player-one-input">`: Përmban fushën e input-it për emrin e lojtarit të parë.
- `<div id="player-two-input">`: Përmban fushën e input-it për emrin e lojtarit të dytë. Nuk është "required", pasi përdoret vetëm në modalitetin me dy lojtarë.
- `<button type="submit">Start Game</button>`: Butoni për të filluar lojën.

3. Seksioni 3: Loja (#game)

- `<section id="game" style="display: none">`: Seksioni kryesor i lojës, fillimisht i fshehur.
- `<div id="status">`: Përdoret për të treguar statusin aktual të lojës, si "Player 1's turn" ose "Player 2 wins!".
- `<div id="board">`: Përmban tabelën 3x3 të lojës.
- `<table>`: Struktura e tabelës me tre rreshta dhe tre qeliza (`<tr>` dhe `<td>`). Secila qelizë përfaqëson një pozicion të lojës Tic-Tac-Toe.
- `<button id="reset" style="display: none">Play Again</button>`: Butoni për të rifilluar lojën. Fillimisht i fshehur, shfaqet kur loja përfundon.
- `<div id="result">`: Seksioni për të mbajtur rezultatin e lojës.
- `<table id="result-table">`: Një tabelë që përmban:
 - **Rreshti 1** (`<th>`): Kolona për emrin e lojtarit 1, barazimet, dhe lojtarin 2.
 - **Rreshti 2** (`<td>`): Vlerat fillestare për rezultatin e lojtarit 1, barazimet, dhe lojtarin 2 (të gjitha të vendosura në 0).

3. JavaScript

- `<script src="index.js"></script>`: Lidh skedarin e JavaScript index.js me dokumentin.

```
<!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="index.css" />
  </head>
  <body>
    <section id="main">
      <h1>Welcome to Tic-Tac-Toe!</h1>
      <h2>Select Play Mode</h2>
      <button id="singleplayer">1 Player</button>
      <button id="twoplayer">2 Players</button>
    </section>

    <section id="player-names" style="display: none">
      <h2>Enter Player Names</h2>
      <form id="player-form">
        <div id="player-one-input">
          <label for="player1">Player 1 Name:</label>
          <input type="text" id="player1" required />
        </div>
        <div id="player-two-input">
          <label for="player2">Player 2 Name:</label>
          <input type="text" id="player2" />
        </div>
        <button type="submit">Start Game</button>
      </form>
    </section>
  </body>
</html>
```

```
<section id="game" style="display: none">
  <div id="status"></div>
  <div id="board">
    <table>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
      <tr>
        <td></td>
        <td></td>
        <td></td>
      </tr>
    </table>
  </div>
  <button id="reset" style="display: none">Play Again</button>
  <div id="result">
    <table id="result-table">
      <tr>
        <th>Player 1</th>
        <th>Tie</th>
        <th>Player 2</th>
      </tr>
```

```

        <tr>
            <td id="player1-score">0</td>
            <td id="draws">0</td>
            <td id="player2-score">0</td>
        </tr>
    </table>
</div>
</section>

<script src="index.js"></script>
</body>
</html>

```

File-I index.css permban:

CSS përdor *id* (#idname), *klasa* (.classname) dhe selektorë të elementeve (element) për të kapur dhe stilizuar pjesë të ndryshme të faqes.

1. Përgjithësime të faqes:

- Stilimi i trupit të faqes (**body**) siguron font të thjeshtë, qendrim të përmbajtjes në mes dhe një sfond të lehtë për një pamje të pastër.

2. Seksionet kryesore:

- Çdo seksion kryesor si **main**, **player-names**, dhe **game** është i organizuar me hapësira për t'u shfaqur qartë dhe për të ndarë funksionet e ndryshme të lojës.

3. Tabela e lojës:

- Qelizat e tabelës janë të përcaktuara me madhësi të barabartë dhe kufij të dukshëm për të krijuar një pamje të strukturuar. Opsioni për të klikuar mbi qeliza sinjalizohet me një ndryshim të kursorit.

4. Butonat:

- Butonat kanë një dizajn të thjeshtë dhe intuitiv, me madhësi të përshtatshme dhe ngjyra që i bëjnë të dallohen lehtësisht.

5. Rezultatet dhe formulari:

- Tabela e rezultateve është vendosur në qendër.

```
body {  
    font-family: Arial, sans-serif;  
    text-align: center;  
    margin: 20px;  
    background-color: #f0f0f0;  
}
```

```
#main,  
#player-names,  
#game {  
    margin-top: 20px;  
}
```

```
#status {  
    margin-bottom: 20px;  
    font-size: 20px;  
    font-weight: bold;  
}
```

```
#board table {  
    border-collapse: collapse;  
    width: 400px;  
    height: 400px;  
    margin: 0 auto;  
}
```

```
#board td {  
    border: 1px solid black;  
    width: 33%;  
    height: 33%;
```

```
    text-align: center;
    vertical-align: middle;
    cursor: pointer;
}
```

```
#board td img {
    width: 80%;
    height: 80%;
}
```

```
button {
    margin: 10px;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
}
```

```
#reset {
    display: none;
    margin: 40px auto;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
}
```

```
#result {
    margin-top: 20px;
}
```

```
#result-table {
    margin: 0 auto;
```



```
border-collapse: collapse;
width: 20%;
height: 50%;
text-align: center;
}

#player-form {
display: flex;
flex-direction: column;
align-items: center;
margin-top: 20px;
}

#player-form input[type="text"] {
margin-bottom: 10px;
padding: 5px;
border: 1px solid #ccc;
border-radius: 5px;
font-size: 16px;
}

#player-form button[type="submit"] {
padding: 10px 20px;
background-color: #277a71;
color: white;
border: none;
border-radius: 5px;
font-size: 16px;
cursor: pointer;
}
```

File-I index.js permban:

- **Deklarimi i elementeve kryesore:**
 - **Elementët e DOM:** Këto elemente lidhen me pjesët kryesore të HTML për të manipuluar vizualisht seksionet dhe tabelën e lojës.
 - **Variabla të kontrollit si:**
 - **onePlayer, twoplayer:** Përcaktojnë mënyrën e lojës (një apo dy lojtarë).
 - **playerOne:** Tregon se cili lojtar është në radhë.
 - **player1Name, player2Name:** Ruajnë emrat e lojtarëve.
- **Events:**
 - **Klikimi mbi butonat e mënyrës së lojës:**
 - **1 Player:** Aktivizon mënyrën me një lojtar dhe fsheh input-in e lojtarit 2.
 - **2 Players:** Kërkon emrat e dy lojtarëve dhe i shfaq ata në seksionin e lojës.
 - **Formulari për emrat e lojtarëve:**
 - Mëson emrat e lojtarëve dhe kalon në seksionin e lojës.
 - **Klikimi mbi qelizat e tabelës:**
 - Në modalitetin me një lojtar, lojtari 1 (X) luan dhe më pas kompjuteri vendos një O në mënyrë të rastësishme.
 - Në modalitetin me dy lojtar, lojtarët ndërrohen duke vendosur X dhe O.
 - **Butoni për rifillimin e lojës:**
 - Pas përfundimit të lojës, tregon një popup (me confirm) që pyet nëse lojtarët duan të luajnë përsëri. Pas konfirmimit, rivendos tabelën dhe rifillon lojën.
- **Funksionet Kryesore:**
 - **insertIcon(cell, icon):** Vendos një ikonë (X ose O) në qelizën e zgjedhur dhe çaktivizon mundësinë për të klikuar sërish mbi të.
 - **autoInsertIcon():** Për mënyrën me një lojtar kompjuteri zgjedh një qelizë bosh në mënyrë të rastësishme dhe vendos O.
 - **checkWin(icon):** Kontrollon nëse lojtari me ikonën e dhënë (X ose O) ka fituar duke krahasuar modelet fituese të mundshme në tabelë.
 - **isDraw():** Kontrollon nëse të gjitha qelizat janë të mbushura pa një fitues (barazim).
 - **updateStatus(message):** Përditëson statusin e lojës me një mesazh për përdoruesit.
 - **resetBoard():** Pastron tabelën dhe rivendos statusin e lojës për një fillim të ri.
 - **updateTable():** Përditëson pikët në tabelën e rezultateve (fitore dhe barazime).

```

const mainSection = document.getElementById("main");
const playerNamesSection = document.getElementById("player-names");
const gameSection = document.getElementById("game");
const statusDiv = document.getElementById("status");
const tdElements = document.querySelectorAll("#board td");
const resetButton = document.getElementById("reset");
const playerForm = document.getElementById("player-form");
const player1Input = document.getElementById("player1");
const player2Input = document.getElementById("player2");
const timeout = 1000;

let onePlayer = false;
let twoplayer = false;
let playerOne = true;
let player1Name = "";
let player2Name = "Computer";

document.getElementById("singleplayer").addEventListener("click", ()
=> {
    mainSection.style.display = "none";
    playerNamesSection.style.display = "block";
    onePlayer = true;
    document.getElementById("player-two-input").style.display = "none";
});

document.getElementById("twoplayer").addEventListener("click", () => {
    document.getElementById("player2").required = true;
    mainSection.style.display = "none";
    playerNamesSection.style.display = "block";
    twoplayer = true;
});

```

```

playerForm.addEventListener("submit", (e) => {
    e.preventDefault();
    player1Name = player1Input.value;
    if (twoplayer) {
        player2Name = player2Input.value;
    }
    playerNamesSection.style.display = "none";
    gameSection.style.display = "block";
    updateStatus(`${player1Name}'s turn`);
});

```

```

tdElements.forEach((cell) => {
    cell.addEventListener("click", () => {
        if (onePlayer && !cell.innerHTML) {
            insertIcon(cell, "X");
            if (checkWin("X")) {
                updateStatus(`${player1Name} wins!`);
                resetButton.style.display = "block";
                disableCells();
            } else if (isDraw()) {
                updateStatus("It's a draw!");
                resetButton.style.display = "block";
                disableCells();
            } else {
                disableCells();
                updateStatus("Computer's turn...");
                autoInsertIcon();
            }
        } else if (twoplayer && !cell.innerHTML) {

```

```
if (playerOne) {
    insertIcon(cell, "X");
    if (checkWin("X")) {
        updateStatus(`${player1Name} wins!`);
        resetButton.style.display = "block";
        disableCells();
    } else if (isDraw()) {
        updateStatus("It's a draw!");
        resetButton.style.display = "block";
        disableCells();
    } else {
        playerOne = false;
        updateStatus(`${player2Name}'s turn`);
    }
} else {
    insertIcon(cell, "O");
    if (checkWin("O")) {
        updateStatus(`${player2Name} wins!`);
        resetButton.style.display = "block";
        disableCells();
    } else if (isDraw()) {
        updateStatus("It's a draw!");
        resetButton.style.display = "block";
        disableCells();
    } else {
        playerOne = true;
        updateStatus(`${player1Name}'s turn`);
    }
}
}
```

```

    });
});

resetButton.addEventListener("click", () => {
    let winnerMessage = "";
    if (checkWin("X")) {
        winnerMessage = `${player1Name} won! Do you want to play again?`;
    } else if (checkWin("O")) {
        winnerMessage = `${player2Name} won! Do you want to play again?`;
    } else {
        winnerMessage = "It's a draw! Do you want to play again?";
    }

    const popup = confirm(winnerMessage);

    if (popup) {
        updateTable();
        resetBoard();
    }
});

function updateTable() {
    const player1Score = document.getElementById("player1-score");
    const draws = document.getElementById("draws");
    const player2Score = document.getElementById("player2-score");

    if (checkWin("X")) {
        player1Score.textContent = parseInt(player1Score.textContent) + 1;
    } else if (checkWin("O")) {
        player2Score.textContent = parseInt(player2Score.textContent) + 1;
    }
}

```

```

    } else {
        draws.textContent = parseInt(draws.textContent) + 1;
    }
}

```

```

function insertIcon(cell, icon) {
    if (!cell.innerHTML) {
        const img = document.createElement("img");
        img.src = `images/${icon}.png`;
        img.alt = icon;
        cell.appendChild(img);
        cell.style.pointerEvents = "none";
    }
}

```

```

function autoInsertIcon() {
    setTimeout(() => {
        const emptyCells = Array.from(tdElements).filter((cell) =>
!cell.innerHTML);
        if (emptyCells.length > 0) {
            const randomIndex = Math.floor(Math.random() *
emptyCells.length);
            const randomCell = emptyCells[randomIndex];
            insertIcon(randomCell, "O");
        }
    });
}

```

```

enableCells();

```

```

if (checkWin("O")) {
    updateStatus(`${player2Name} wins!`);
    resetButton.style.display = "block";
}

```

```

        disableCells();
    } else if (isDraw()) {
        updateStatus("It's a draw!");
        resetButton.style.display = "block";
        disableCells();
    } else {
        updateStatus(`${player1Name}'s turn`);
    }
}, timeout);
}

function disableCells() {
    tdElements.forEach((cell) => {
        cell.style.pointerEvents = "none";
    });
}

function enableCells() {
    tdElements.forEach((cell) => {
        cell.style.pointerEvents = "auto";
    });
}

function checkWin(icon) {
    const board = Array.from(tdElements).map(
        (cell) => cell.querySelector("img")?.alt === icon
    );

    const winPatterns = [
        [0, 1, 2],

```



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

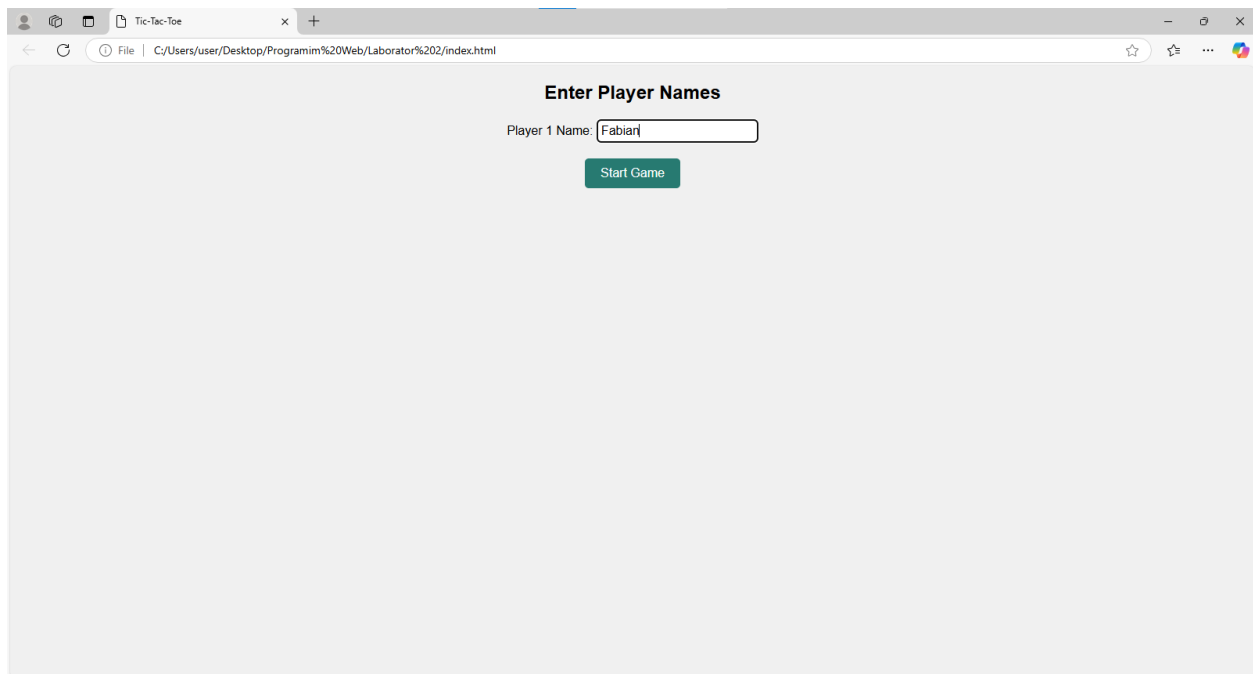
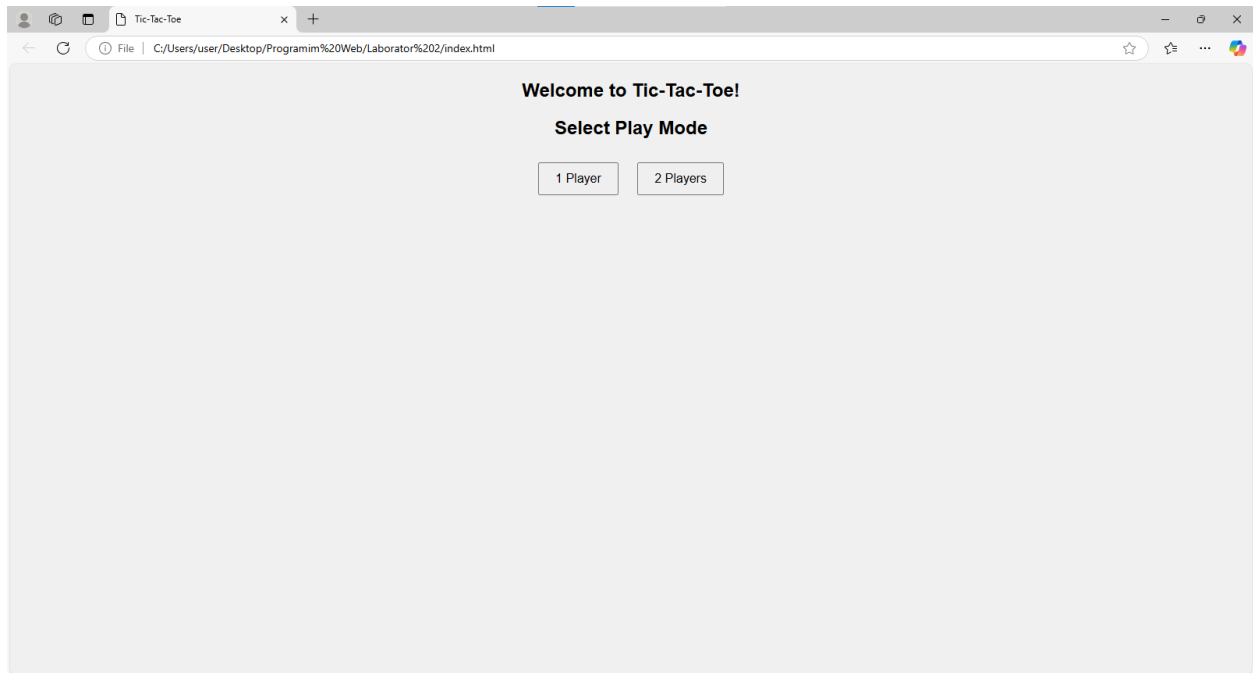
```
return winPatterns.some((pattern) => pattern.every((index) =>  
board[index]));  
}
```

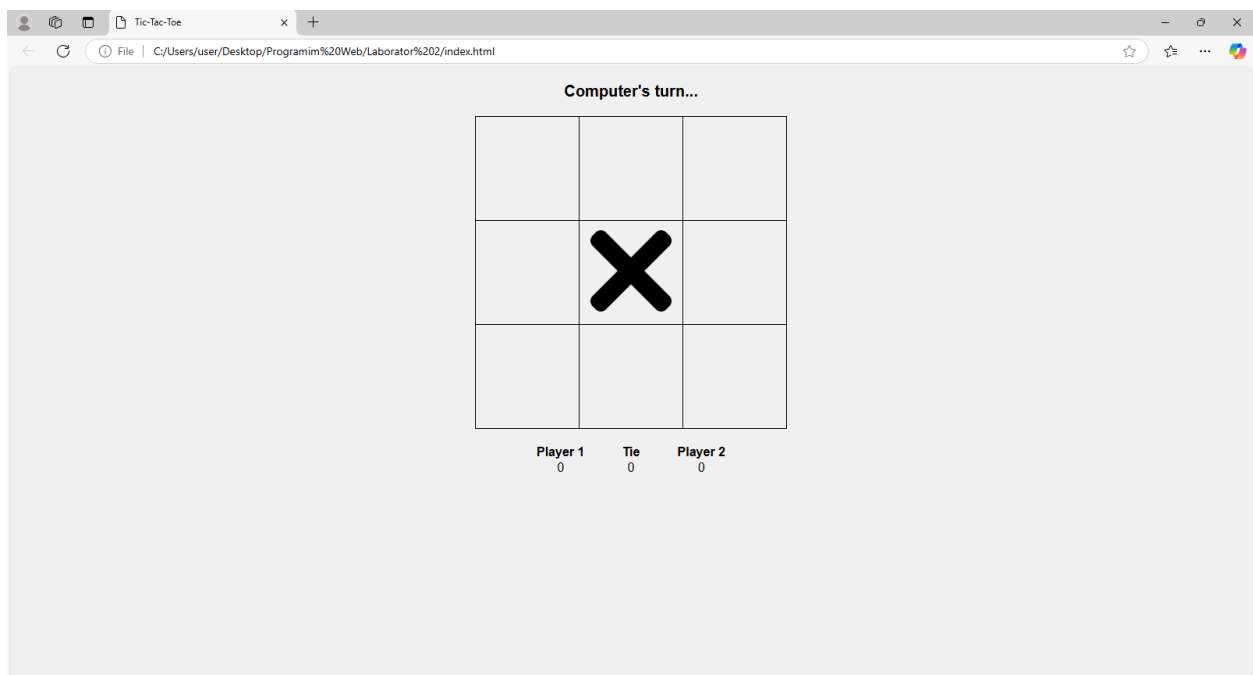
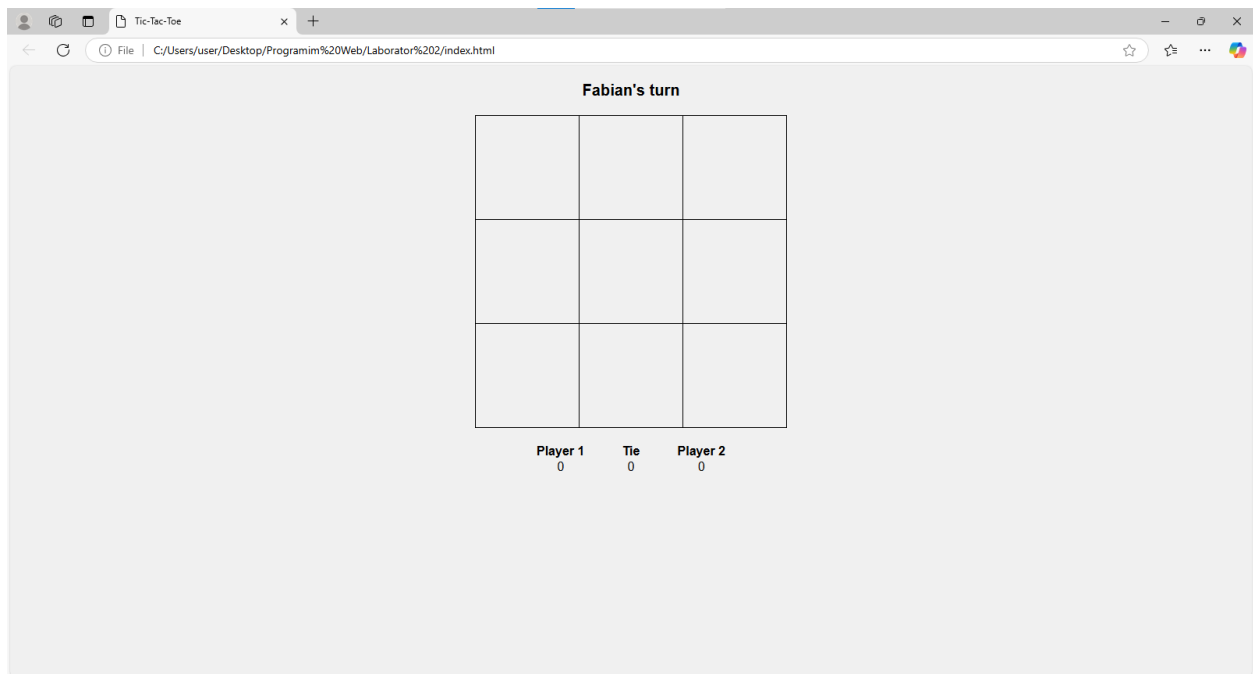
```
function isDraw() {  
    return Array.from(tdElements).every((cell) => cell.innerHTML);  
}
```

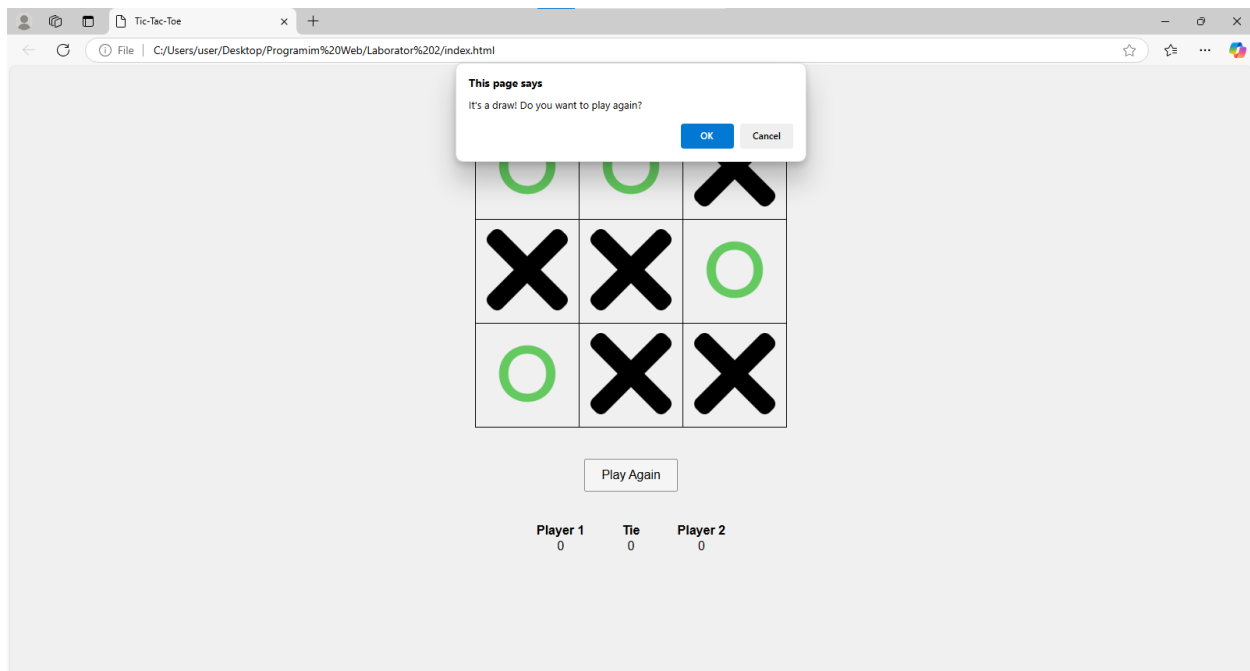
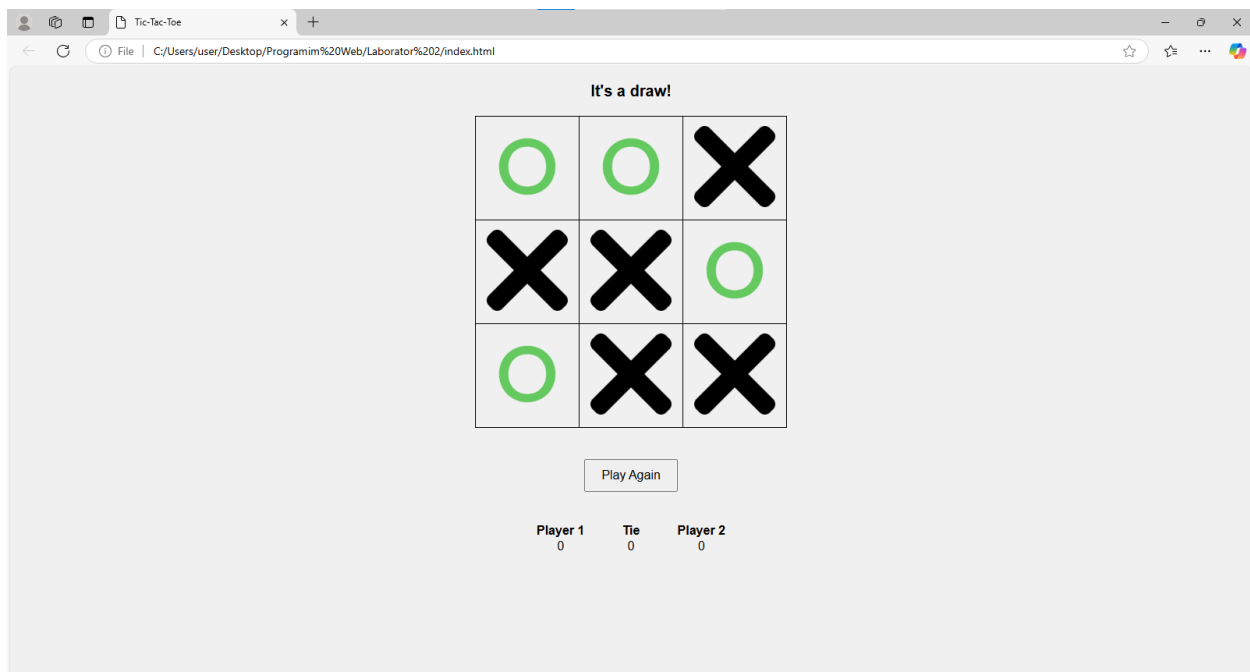
```
function updateStatus(message) {  
    statusDiv.textContent = message;  
}
```

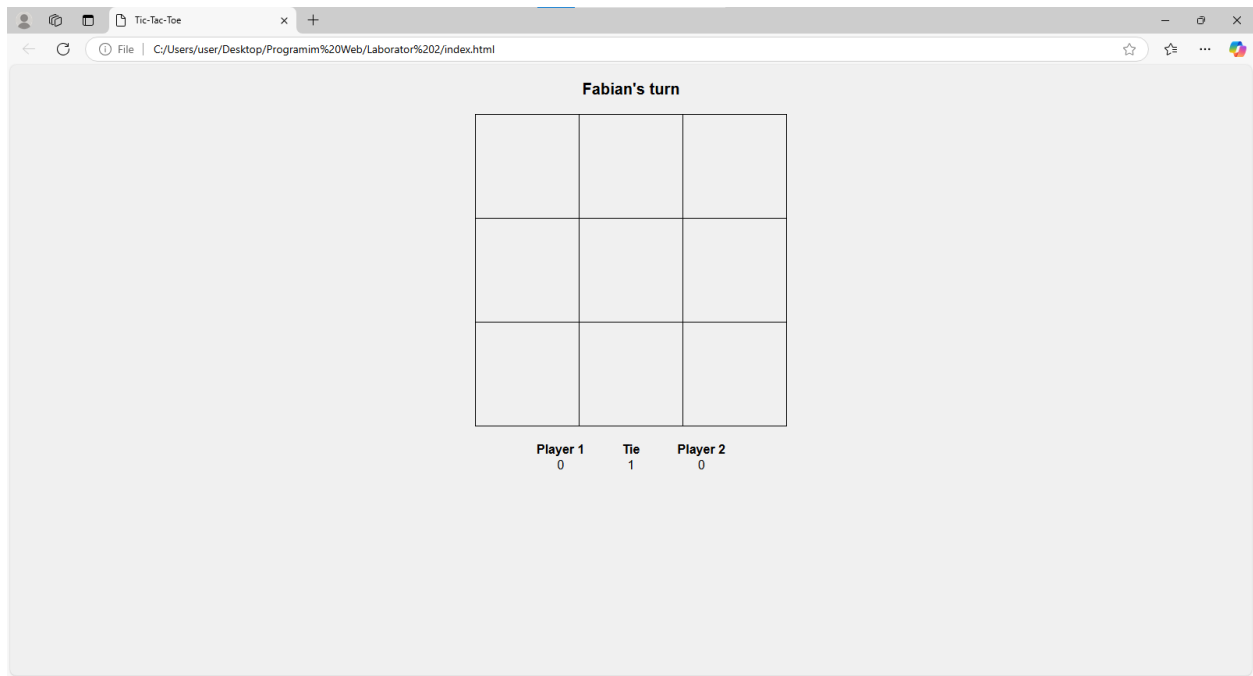
```
function resetBoard() {  
    tdElements.forEach((cell) => {  
        cell.innerHTML = "";  
        cell.style.pointerEvents = "auto";  
    });  
    resetButton.style.display = "none";  
  
    playerOne = true;  
    updateStatus(`${player1Name}'s turn`);  
}
```

Rezultati per 1 player mode:









Rezultati per 2 player mode:

