# HTML:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>App Academy DOM API Tic-Tac-Toe Project</title>

  <link rel="stylesheet" href="./site.css">

</head>

<body>

  <div id="game">

    <h1 class="announcement" id="game-status"></h1>

    <div id="tic-tac-toe-board">

      <div id="square-0" class="square row-1 col-1"></div>

      <div id="square-1" class="square row-1 col-2"></div>

      <div id="square-2" class="square row-1 col-3"></div>

      <div id="square-3" class="square row-2 col-1"></div>

      <div id="square-4" class="square row-2 col-2"></div>

      <div id="square-5" class="square row-2 col-3"></div>

      <div id="square-6" class="square row-3 col-1"></div>

      <div id="square-7" class="square row-3 col-2"></div>

      <div id="square-8" class="square row-3 col-3"></div>

    </div>

    <div class="actions">

      <button>New Game</button>

      <div class="spacer"></div>

      <button>Give Up</button>

    </div>

  </div>

  <script src="./tic-tac-toe.js"></script>

</body>

</html>

# JS:

window.addEventListener("DOMContentLoaded", event => {

  //set title and local storage

  console.log(1);

  let title = document.getElementById("game-status");

  const ticTacToe = document.getElementById('tic-tac-toe-board');

  if(localStorage.getItem("title") === null){

    localStorage.setItem("title", "Tic-Tac-Toe!");

    localStorage.setItem('computerArray',JSON.stringify([0,1,2,3,4,5,6,7,8]));

    localStorage.setItem('computerTurn', JSON.stringify(Math.random() <= 0.5));

  }

  title.innerHTML = localStorage.getItem("title");

  let turn = document.createElement("div");

  turn.setAttribute("class", "announcement");

  turn.innerHTML = "";

  title.after(turn);

  //turn title function

  let changeTitle = function(bool\_val, array1) {

    console.log(8);

    if(array1.length < 9){

   let gameFinished = gameStatus();

    if (gameFinished){

      turn.innerHTML = "Game Complete!"

      return;

    }

  }

    if ((!computerTurn || computerArray.length === 0)){

      turn.innerHTML = "Player's Turn";

      return 1;

    }

    else{

      turn.innerHTML = "Computer's turn";

      return 0;

    }

  }

  //computer move

  let computerMove = function(){

    console.log(2);

    computerArray = JSON.parse(localStorage.getItem('computerArray'));

    computerTurn = JSON.parse(localStorage.getItem('computerTurn'));

    let bla = changeTitle(computerTurn, computerArray);

    if(bla){

      return;

    }

    let rand = Math.floor(Math.random() \* Math.floor(computerArray.length));

    let moveStr = "square-" + computerArray[rand];

    let computerImg = document.getElementById(moveStr);

    // addImage(computerImg.click());

    setTimeout(function(){

      computerImg.click(event => {

        addImage(event);

      });

    },500);

  }

  let addImage = function(event1) {

    console.log(3);

    let symbol = "/x.svg"

    let xArr = JSON.parse(localStorage.getItem("xArr"));

    let oArr = JSON.parse(localStorage.getItem('oArr'));

    let computerArray = JSON.parse(localStorage.getItem('computerArray'));

    let setArr = function () {

      if ((xArr.length + oArr.length) % 2 === 0) {

        symbol = "/x.svg"

        xArr.push(Number(event1.target.id[7]))

        computerArray.splice(computerArray.indexOf(Number(event1.target.id[7])), 1);

        let xStr = JSON.stringify(xArr);

        localStorage.setItem('xArr', xStr);

        localStorage.setItem('computerTurn', !JSON.parse(localStorage.getItem('computerTurn')));

      } else {

        symbol = "/o.svg"

        oArr.push(Number(event1.target.id[7]))

        computerArray.splice(computerArray.indexOf(Number(event1.target.id[7])), 1);

        let oStr = JSON.stringify(oArr)

        localStorage.setItem('oArr', oStr);

        localStorage.setItem('computerTurn', !JSON.parse(localStorage.getItem('computerTurn')));

      }

      let computerStr = JSON.stringify(computerArray);

      localStorage.setItem('computerArray',computerStr)

      let key = event1.target.id

      localStorage.setItem(key, symbol);

    }

    if (xArr === null || oArr === null) {

      initialStr = JSON.stringify([]);

      localStorage.setItem('xArr', initialStr);

      localStorage.setItem('oArr', initialStr);

      localStorage.setItem('computerArray',JSON.stringify([0,1,2,3,4,5,6,7,8]));

      addImage(event1);

    } else {

      setArr();

    }

  }

  //update storage.

  let updateStorage = () => {

    console.log(4);

    ticTacToe.addEventListener("click",  event1=> {

      if ((event1.target.id !== 'image') && (title.innerHTML === "Tic-Tac-Toe!")) {

        addImage(event1);

        actions();

        computerMove();

        gameStatus();

        location.reload();

      }

    });

  }

  //show Storage.

  let showStorage = () => {

    console.log(5);

    let squares = document.querySelectorAll(".square");

    let ticTacArray = [];

    squares.forEach(square => {

      let value = localStorage.getItem(square.id);

      if (value !== null) {

        newImage = document.createElement('img');

        newImage.setAttribute('src', value)

        newImage.setAttribute('id', 'image')

        square.appendChild(newImage);

        ticTacArray.push()

      }

    });

    // document.getElementById(event1.target.id).appendChild(newImage);

  }

  //newGame

  let actions = () => {

    console.log(6);

    let button = document.querySelectorAll(".actions > button");

    if(title.innerHTML === "Tic-Tac-Toe!"){

      button[0].setAttribute("disabled", "true");

      button[1].removeAttribute("disabled");

    }

    else{

      button[0].removeAttribute("disabled");

      button[1].setAttribute("disabled", 'true');

    }

    button[0].addEventListener("click", eve => {

        localStorage.clear();

        location.reload();

    });

    button[1].addEventListener("click", eve => {

      let xArr = JSON.parse(localStorage.getItem("xArr"));

      let oArr = JSON.parse(localStorage.getItem('oArr'));

      if((xArr.length + oArr.length) % 2 === 0){

        localStorage.setItem("title", "O's Win!!");

      }

      else{

        localStorage.setItem("title", "X's Win!!");

      }

      location.reload();

    });

  }

  let gameStatus = () => {

    console.log(7);

    let xArr = JSON.parse(localStorage.getItem("xArr"));

    let oArr = JSON.parse(localStorage.getItem("oArr"));

    let connected = 0;

    let winConditions = [

      [0, 1, 2],

      [3, 4, 5],

      [6, 7, 8],

      [0, 3, 6],

      [1, 4, 7],

      [2, 5, 8],

      [0, 4, 8],

      [2, 4, 6]

    ]

    //Change Title

    for (let i = 0; i < winConditions.length; i++) {

      if (xArr.includes(winConditions[i][0]) && xArr.includes(winConditions[i][1]) && xArr.includes(winConditions[i][2])) {

        localStorage.setItem("title", "X's Win!!");

        return true;

      }

      if (oArr.includes(winConditions[i][0]) && oArr.includes(winConditions[i][1]) && oArr.includes(winConditions[i][2])) {

        localStorage.setItem("title", "O's Win!!");

        return true;

      }

    }

    if(xArr.length + oArr.length === 9){

      localStorage.setItem("title", "It is a tie!");

      return true;

    }

  }

  updateStorage();

  showStorage();

  computerMove();

  actions();

});

/\*

x

[ ['', ''] , ['', ''] ,[0, 2],

  [1, 0] , [1, 1] ,[1, 2],

  [2, 0] , [2, 1] ,[2, 2]

]

xarr

//

oarr

1 -- row 1 (0, 1, 2) array.every(func = () => {loop through 0,1,2}

2 -- row 2 (3,4,5) "345"

3 -- row 3 (6,7,8)

4 -- col 1 (0,3,6)

5 -- col 2 (1,4,7)

6 -- col 3 (2,5,8)

7 -- diag 1 (1,4,8)

8 -- diag 2 (2,4,6)

\*/

/\*

whenever we click a space, we push that onto an arry and remove it from 'potential move' array

generate random number from 0 to length of array

inserts into the position

let random = Math.floor(Math.rand() \* Math.floor(arr.length));

check game status after computer move

\*/

# CS:

body, button {

  font-family: Arial, Helvetica, sans-serif;

}

button {

  font-size: 1.25em;

}

#game {

  margin: auto;

  width: 402px;

}

#tic-tac-toe-board {

  display: grid;

  height: 400px;

  width: 400px;

  grid-template-columns: 1fr 1fr 1fr;

  grid-template-rows: 1fr 1fr 1fr;

  background-color: black;

  margin: 32px 0;

}

.actions {

  display: flex;

}

.announcement {

  font-size: 3em;

  text-align: center;

}

.col-1 {

  justify-self: start;

}

.col-2 {

  justify-self: center;

}

.col-3 {

  justify-self: end;

}

.row-1 {

  align-self: start;

}

.row-2 {

  align-self: center;

}

.row-3 {

  align-self: end;

}

.spacer {

  flex: 1 0 0px;

}

.square {

  background-color: white;

  height: 130px;

  width: 130px;

}