

Assignment – 1

Chess board:

```
<!DOCTYPE html>
<html>
  <head>
    <title>**Chess Board**</title>
    <meta charset="UTF-8">
    <style>
      * {
        margin: 0;
        padding: 0;
      }
      table {
        width: 500px;
        height: 500px;
        margin: auto;
      }
      table td {
        border: 1px solid;
      }
      .dark {
        background-color: black;
        width: 50px;
      }
      .white {
        background-color: white;
        width: 50px;
      }
    </style>
  </head>
  <body>
    <table>
      <tbody>
        <tr>
          <th></th>
          <th>a</th>
          <th>b</th>
          <th>c</th>
          <th>d</th>
          <th>e</th>
          <th>f</th>
          <th>g</th>
          <th>h</th>
        </tr>

        <tr>
          <th>1</th>
          <td class="light"></td>
          <td class="dark"></td>
          <td class="light"></td>
```

```

        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
    </tr>
    <tr>
        <th>2</th>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
    </tr>
    <tr>
        <th>3</th>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
    </tr>
    <tr>
        <th>4</th>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
    </tr>

```

```

    <tr>
        <th>5</th>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
        <td class="light"></td>
        <td class="dark"></td>
    </tr>

```

```

</tr>
<tr>
  <th>6</th>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
</tr>
<tr>
  <th>7</th>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
</tr>
<tr>
  <th>8</th>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
  <td class="dark"></td>
  <td class="light"></td>
</tr>
</tbody>
</table>
</body>
</html>

```

VIBGYOR :

```

<!DOCTYPE html>
<html>
  <head>
    <title>**VIBGYOR**</title>
    <meta charset="UTF-8">
    <style>
      * {
        margin: 0;
        padding: 0;
        padding-top: 20px;
      }
    </style>
  </head>
  <body>
    <table>

```

```

        width: 500px;
        height: 500px;
        margin: auto;
    }
    .Violet {
        background-color: #9400D3;
    }
    .Indigo {
        background-color: #4B0082;
    }
    .Blue {
        background-color: #0000FF;
    }
    .Green {
        background-color: #00FF00;
    }
    .Yellow {
        background-color: #FFFF00;
    }
    .Orange {
        background-color: #FF7F00;
    }
    .Red {
        background-color: #FF0000;
    }
</style>
</head>
<body>
<table>
<tbody>
<tr>
    <td class="Violet"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
</tr>
<tr>
    <td class="light"></td>
    <td class="Indigo"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
    <td class="light"></td>
</tr>
<tr>
    <td class="light"></td>
    <td class="light"></td>

```

```
<td class="Blue"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
</tr>
<tr>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="Green"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
</tr>
<tr>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="Yellow"></td>
<td class="light"></td>
<td class="light"></td>
</tr>
<tr>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="Orange"></td>
<td class="light"></td>
</tr>
<tr>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="light"></td>
<td class="Red"></td>
</tr>
</tbody>
</table>
</body>
</html>
```

TIC-TAC-TOE:

```
<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>Tic-Tac-Toe</title>
</style>
* {
  padding: 0;
  margin: 0;
  font-family: 'Itim';
}

.title {
  color: black;
  text-align: center;
  font-size: 40px;
  margin-top: 2%;
}

.display {
  color: black;
  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 black;
  min-width: 100px;
```

```

    min-height: 100px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 50px;
    cursor: pointer;
}

.playerX {
    color: #09C372;
}

.playerO {
    color: #498AFB;
}

.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;
}
</style>
</head>
<body>
    <main class="background">
        <section class="title">
            <h1>Tic Tac Toe</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>

```

```

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';
  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);
});

```

TIME TABLE:

```

<html>
<head>
    <title>time table</title>
</head>
<body bgcolor=#FFE5B4>
    <H1><FONT COLOR=#A020F0><CENTER>COLLEGE TIME TABLE</FONT></H1>
        <table border="5" cellspacing="20" align="center">
            <tr>
                <td align="center">
                    <td>09:00-09:50</td>
                    <td>09:50-10:40</td>
                    <td>10:40-11:00</td>
                    <td>11:00-11:50</td>
                    <td>11:50-12:40</td>
                    <td>12:40-01:20</td>
                    <td>01:20-02:10</td>
                    <td>02:10-03:00</td>
                    <td>03:00-03:50</td>
                </tr>

```



```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>REGISTER</title>

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous" />
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
integrity="sha384-fnmOCqbTIWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
    <script defer src="register.js"></script>
    <link rel="stylesheet" href="registration.css"/>
</head>
<body>
    <header>
        <div class="container-fluid p-0">
            <nav class="navbar navbar-expand-lg">
                <a class="navbar-brand" href="#">
                    <i class="fas fa-address-card"></i>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
REGISTRATION PAGE WITH HTML, CSS, AND JS</a>
                    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
                        <i class="fas fa-indent text-light"></i>
                    </button>
                </nav>
            </div>
            <div class="container text-center">
                <div class="row">
                    <div class="col-md-7 col-sm-12 text-white">
                        <form id="form" action="/">

                            <label>User Name</label>
                            <input type="text" name="username" id="username">
                            <div class="error"></div>
                            <label>Email</label>
                            <input type="text" name="email" id="email">
                            <div class="error"></div>
                            <label>Password</label>
                            <input type="text" name="password" id="password">
                            <div class="error"></div>
                            <button type="submit" onclick="alert('You have registered successfully')">register</button>

                        </form>
```

```

        </div>
        <div class="col-md-5 col-sm-12 h-25">
            
        </div>
    </div>
</div>
</header>
</body>
</html>

```

```

const form = document.getElementById('form');
const username = document.getElementById('username');
const email = document.getElementById('email');
const password = document.getElementById('password');

```

```

form.addEventListener('submit', e => {
    e.preventDefault();

```

```

    validateInputs();
});

```

```

const setError = (element, message) => {
    const inputControl = element.parentElement;
    const errorDisplay = inputControl.querySelector('.error');

    errorDisplay.innerText = message;
    inputControl.classList.add('error');
    inputControl.classList.remove('success')
}

```

```

const setSuccess = element => {
    const inputControl = element.parentElement;
    const errorDisplay = inputControl.querySelector('.error');

    errorDisplay.innerText = '';
    inputControl.classList.add('success');
    inputControl.classList.remove('error');
};

```

```

const isValidEmail = email => {
    const re = /^(^<>()[]\.\,\:\s@""]+(\.^[^<>()[]\.\,\:\s@""]+)*)(\."")@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|([a-zA-Z-0-9]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(String(email).toLowerCase());
}

```

```

const validateInputs = () => {
    const usernameValue = username.value.trim();
    const emailValue = email.value.trim();
    const passwordValue = password.value.trim();

```

```

if(usernameValue === "") {
    setError(username, 'Username is required');
} else {
    setSuccess(username);
}

if(emailValue === "") {
    setError(email, 'Email is required');
} else if (!isValidEmail(emailValue)) {
    setError(email, 'Provide a valid email address');
} else {
    setSuccess(email);
}

if(passwordValue === "") {
    setError(password, 'Password is required');
} else if (passwordValue.length < 6 ) {
    setError(password, 'Password must be at least 6 character.')
} else {
    setSuccess(password);
}

};

@import url("https://fonts.googleapis.com/css?family=Acme|Lobster|Patua+One|Rubik|Sniglet");
@import url('https://fonts.googleapis.com/css2?family=Tiro+Bangla:ital@1&display=swap');
* {
    box-sizing: border-box;
}

header {
    overflow-x: hidden;
}

:root {
    --Sniglet-font: "Sniglet", cursive;
    --Rubik: "Rubik", cursive;
    --bggradient: linear-gradient(to bottom, #FF6DA4, #1DC28B);
}

header a {
    font-family: var(--Sniglet-font);
    font-size: 6rem;
    color: whitesmoke;
}

header .navbar-brand {
    padding-left: 8rem;
}

```

```
header .navbar-brand:hover{
  color: black;
}
```

```
header .row .col-md-7 {
  padding: 25vmin 1vmin;
  padding-bottom: 128px;
}
```

```
header .row .col-md-5 {
  padding: 4.2vmin 1vmin;
}
```

```
header .row .col-md-5 img {
  width: 90%;
}
```

```
header {
  background: #C21D54;
  /* fallback for old browsers */
  background: var(--bggradient);
}
```

```
header .container .col-md-7 {
  font-family: "Rubik", sans-serif;
}
```

```
form{
  width: 30rem;
  height: 18rem;
  display: flex;
  flex-direction: column;
  background: rgba(255,255,255,0.06);
  box-shadow: 0 8px 32px 0 rgba(31,38,135,0.37);
  border-radius: 30px;
  border-left: 1px solid rgba(255,255,255,.3);
  border-top: 1px solid rgba(255,255,255,.3);
}
```

```
form:before{
  content: "";
  position: absolute;
  top: 2%;
  right: 50%;
  width: 50px;
  height: 50px;
  background: #728FCE;
  border-radius: 20%;
  z-index: -1;
  opacity: .8;
```

```
}
```

```
label{  
  font-size: 20px;  
  color: black;  
  text-shadow: 2px 2px 4px rgba(0,0,0,.2);  
}
```

```
input{  
  width: 90%;  
  margin: 2% auto;  
  margin-bottom: 4%;  
  border: none;  
  outline: none;  
  background: transparent;  
  color: #000000;  
  border-bottom: 1px solid rgb(0,0,0);  
}
```

```
button{  
  width: 50%;  
  margin: 3% auto;  
  color: black;  
  font-size: 15px;  
  opacity: .9;  
  background: rgba(255,255,255,0.06);  
  padding: 10px 30px;  
  outline: none;  
  border-radius: 20px;  
  text-shadow: 2px 2px 4px rgb(0,0,0);  
  box-shadow: rgba(31,38,135,.37);  
}
```