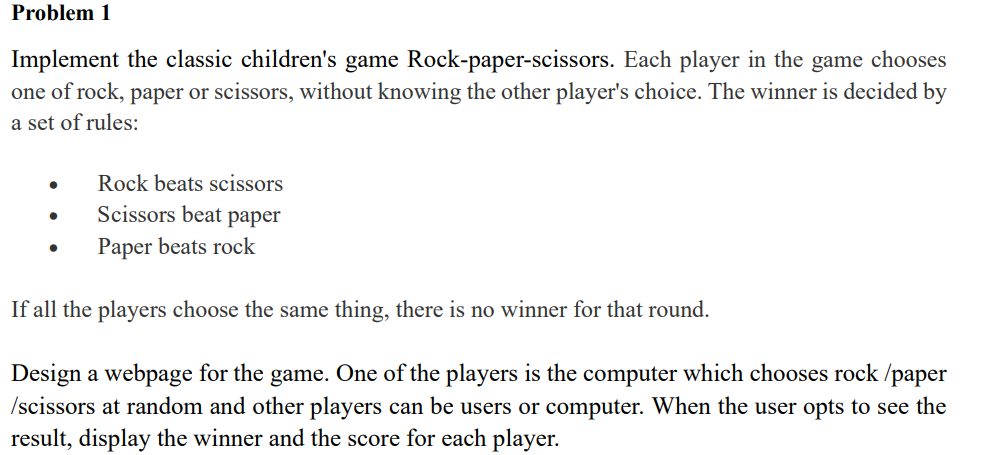
**ASSIGNMENT – 3C**



**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Rock Paper Scissors</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

  <style>

    \* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body {

    font-family: Arial, sans-serif;

    background-color: #f0f0f0;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    flex-direction: column;

}

.container {

    background-color: white;

    padding: 20px;

    border-radius: 10px;

    box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.1);

    text-align: center;

}

h1 {

    margin-bottom: 20px;

}

.score-board {

    display: flex;

    justify-content: space-around;

    margin-bottom: 20px;

}

.score-board div {

    font-size: 20px;

}

.choices {

    display: flex;

    justify-content: space-around;

    margin-bottom: 20px;

}

.choice-btn {

    padding: 10px 20px;

    font-size: 18px;

    margin: 10px;

    cursor: pointer;

    border: none;

    border-radius: 5px;

    background-color: #4CAF50;

    color: white;

    transition: background-color 0.3s;

}

.choice-btn:hover {

    background-color: #45a049;

}

.result {

    font-size: 24px;

    margin-bottom: 20px;

}

#play-again-btn {

    padding: 10px 20px;

    font-size: 18px;

    background-color: #f44336;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

}

#play-again-btn:hover {

    background-color: #e53935;

}

  </style>

<script>

  let playerScore = 0;

let computerScore = 0;

const choices = ['rock', 'paper', 'scissors'];

function userChoice(choice) {

    const computerChoice = getComputerChoice();

    const winner = determineWinner(choice, computerChoice);

    updateResult(winner, choice, computerChoice);

    updateScore(winner);

    document.getElementById('play-again-btn').style.display = 'inline-block';

}

function getComputerChoice() {

    return choices[Math.floor(Math.random() \* choices.length)];

}

function determineWinner(playerChoice, computerChoice) {

    if (playerChoice === computerChoice) {

        return 'draw';

    }

    if (

        (playerChoice === 'rock' && computerChoice === 'scissors') ||

        (playerChoice === 'scissors' && computerChoice === 'paper') ||

        (playerChoice === 'paper' && computerChoice === 'rock')

    ) {

        return 'player';

    } else {

        return 'computer';

    }

}

function updateResult(winner, playerChoice, computerChoice) {

    let resultText = '';

    if (winner === 'draw') {

        resultText = `It's a draw! You both chose ${playerChoice}.`;

    } else if (winner === 'player') {

        resultText = `You win! ${capitalize(playerChoice)} beats ${capitalize(computerChoice)}.`;

    } else {

        resultText = `Computer wins! ${capitalize(computerChoice)} beats ${capitalize(playerChoice)}.`;

    }

    document.getElementById('result').textContent = resultText;

}

function updateScore(winner) {

    if (winner === 'player') {

        playerScore++;

        document.getElementById('player-score').textContent = playerScore;

    } else if (winner === 'computer') {

        computerScore++;

        document.getElementById('computer-score').textContent = computerScore;

    }

}

function capitalize(str) {

    return str.charAt(0).toUpperCase() + str.slice(1);

}

function resetGame() {

    playerScore = 0;

    computerScore = 0;

    document.getElementById('player-score').textContent = playerScore;

    document.getElementById('computer-score').textContent = computerScore;

    document.getElementById('result').textContent = 'Choose your weapon!';

    document.getElementById('play-again-btn').style.display = 'none';

}

</script>

    <div class="container">

        <h1>Rock Paper Scissors</h1>

        <div class="score-board">

            <div class="player-score">

                <h2>Player</h2>

                <p id="player-score">0</p>

            </div>

            <div class="computer-score">

                <h2>Computer</h2>

                <p id="computer-score">0</p>

            </div>

        </div>

        <div class="choices">

            <button class="choice-btn" onclick="userChoice('rock')">Rock</button>

            <button class="choice-btn" onclick="userChoice('paper')">Paper</button>

            <button class="choice-btn" onclick="userChoice('scissors')">Scissors</button>

        </div>

        <div id="result" class="result">

            <p>Choose your weapon!</p>

        </div>

        <button id="play-again-btn" onclick="resetGame()" style="display:none;">Play Again</button>

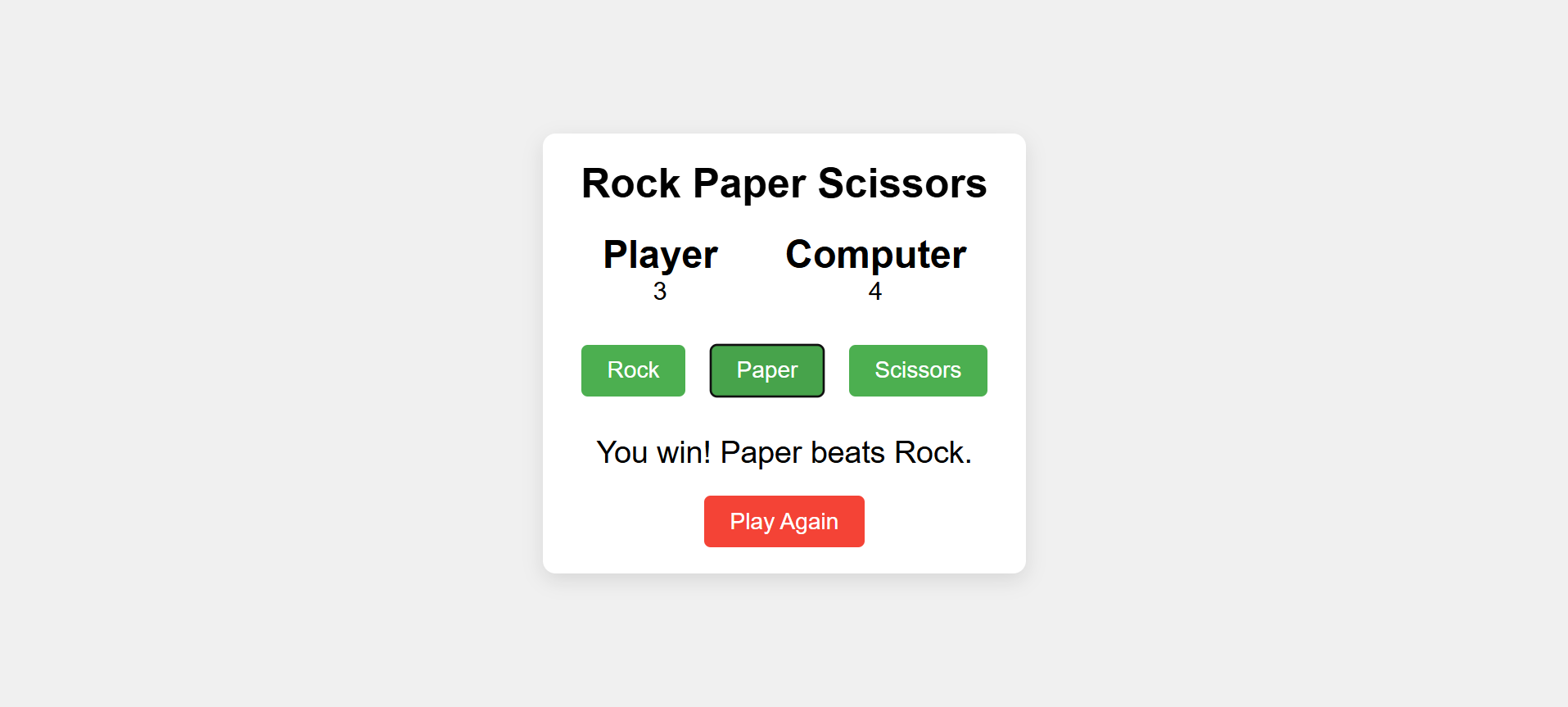
    </div>

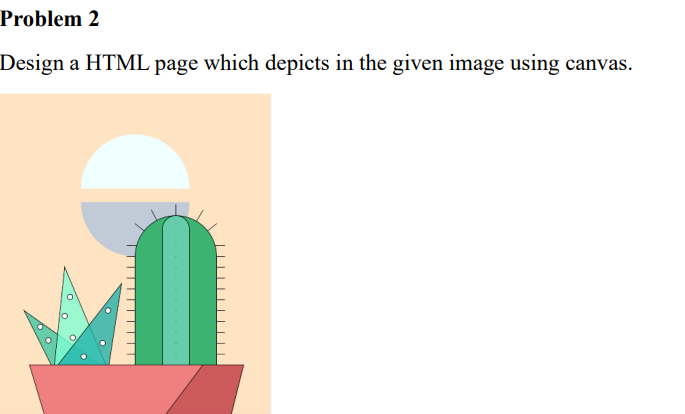
    <script src="script.js"></script>

</body>

</html>

**OUTPUT:**





**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Cactus using Canvas</title>

</head>

<body>

    <h3>Cactus using Canvas</h3>

    <canvas id="myCanvas" width="300" height="400"></canvas>

    <script>

        var canvas = document.getElementById("myCanvas");

        var ctx = canvas.getContext("2d");

        ctx.fillStyle = "#F7E7C3";

        ctx.fillRect(0, 0, canvas.width, canvas.height);

        ctx.beginPath();

        ctx.arc(150, 100, 60, 0, Math.PI, true);

        ctx.fillStyle = "white";

        ctx.fill();

        ctx.beginPath();

        ctx.arc(150, 110, 60, Math.PI, 2 \* Math.PI, true);

        ctx.fillStyle = "lightgrey";

        ctx.fill();

        ctx.beginPath();

        ctx.moveTo(20, 320);

        ctx.lineTo(280, 320);

        ctx.lineTo(240, 400);

        ctx.lineTo(60, 400);

        ctx.closePath();

        ctx.fillStyle="#C19A6B ";

        ctx.fill();

        ctx.beginPath();

        ctx.moveTo(220, 320);

        ctx.lineTo(280, 320);

        ctx.lineTo(240, 400);

        ctx.lineTo(160, 400);

        ctx.closePath();

        ctx.fillStyle = "#A0522D";

        ctx.fill();

        ctx.fillStyle = "#5DAE7F";

        ctx.beginPath();

        ctx.moveTo(60, 320);

        ctx.lineTo(45, 200);

        ctx.lineTo(80, 320);

        ctx.closePath();

        ctx.fill();

        ctx.fillStyle = "#A0C55F";

        ctx.beginPath();

        ctx.moveTo(60, 320);

        ctx.lineTo(70, 180);

        ctx.lineTo(80, 320);

        ctx.closePath();

        ctx.fill();

        ctx.fillStyle = "#3F95C0";

        ctx.beginPath();

        ctx.moveTo(60, 320);

        ctx.lineTo(100, 200);

        ctx.lineTo(80, 320);

        ctx.closePath();

        ctx.fill();

        ctx.fillStyle = "white";

        for (let i = 0; i < 5; i++) {

            let x = 60 + Math.random() \* 20;

            let y = 220 + Math.random() \* 80;

            ctx.beginPath();

            ctx.arc(x, y, 3, 0, 2 \* Math.PI);

            ctx.fill();

        }

        ctx.fillStyle = "#2E8B57";

        ctx.fillRect(150, 200, 100, 120);

        ctx.beginPath();

        ctx.arc(200, 200, 50, Math.PI, 0, false);

        ctx.fill();

        ctx.fillStyle = "#76C76C";

        ctx.fillRect(190, 180, 20, 140);

        ctx.beginPath();

        ctx.arc(200, 180, 10, Math.PI, 0, false);

        ctx.fill();

        ctx.strokeStyle = "#333";

        ctx.lineWidth = 2;

        let spikeLength = 8;

        for (let y = 210; y <= 310; y += 15) {

            ctx.beginPath();

            ctx.moveTo(145, y);

            ctx.lineTo(145 - spikeLength, y);

            ctx.stroke();

            ctx.beginPath();

            ctx.moveTo(255, y);

            ctx.lineTo(255 + spikeLength, y);

            ctx.stroke();

        }

        for (let angle = Math.PI; angle <= 0; angle += Math.PI / 10){

            let x = 200 + 50 \* Math.cos(angle);

            let y = 200 + 50 \* Math.sin(angle);

            let dx = Math.cos(angle - Math.PI / 2) \* spikeLength;

            let dy = Math.sin(angle - Math.PI / 2) \* spikeLength;

            ctx.beginPath();

            ctx.moveTo(x, y);

            ctx.lineTo(x + dx, y + dy);

            ctx.stroke();

        }

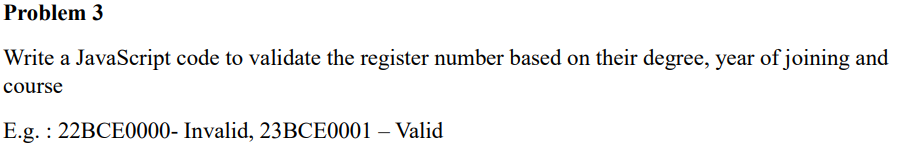
    </script>

</body>

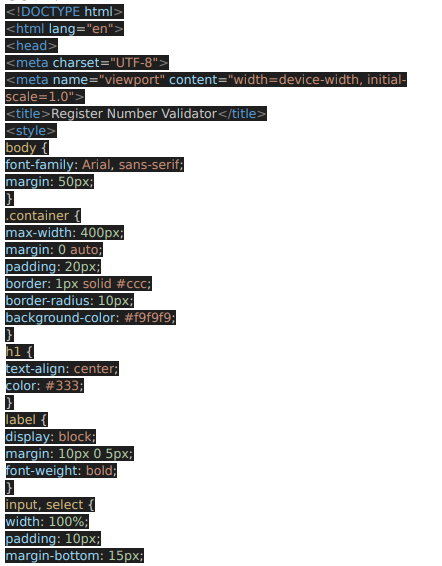
</html>

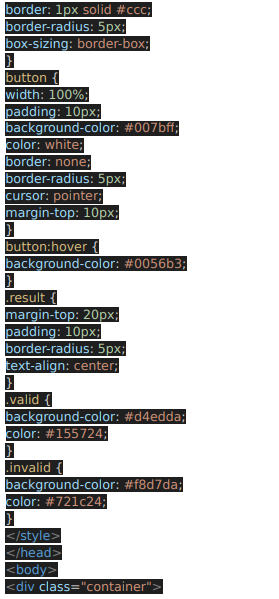
**OUTPUT:**

****

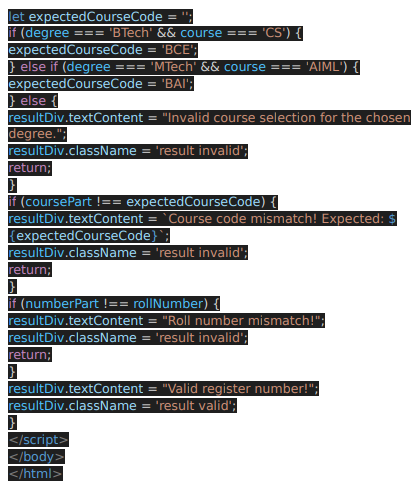
****

**CODE:**

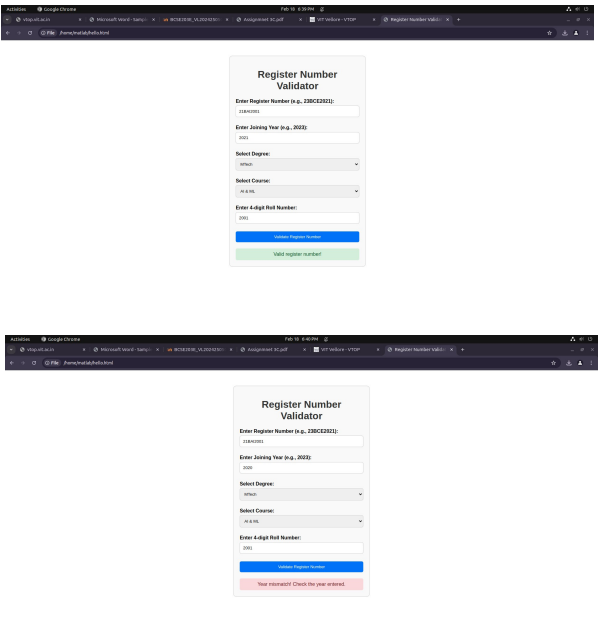
****

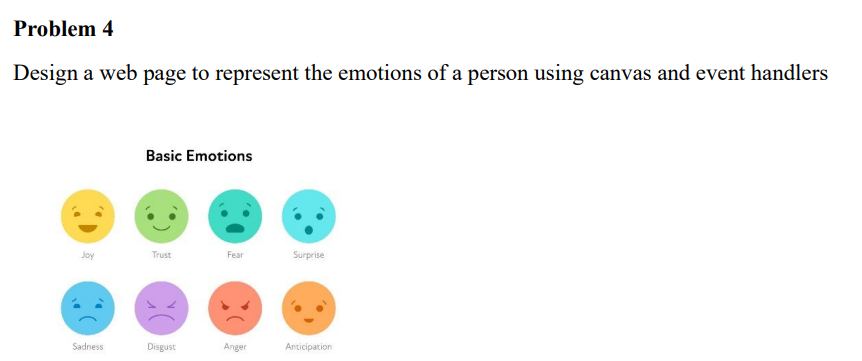
****

****

****

**OUTPUT:**

****

****

**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Emotion Representation</title>

    <style>

        body {

            text-align: center;

            font-family: Arial, sans-serif;

        }

    </style>

</head>

<body>

    <h2>Basic Emotions</h2>

    <canvas id="emotionCanvas" width="500" height="300"></canvas>

    <p id="emotionText"></p>

    <script>

        const canvas = document.getElementById("emotionCanvas");

        const ctx = canvas.getContext("2d");

        const emotions = [

            { name: "Joy", color: "#FFD700", x: 80, y: 80 },

            { name: "Trust", color: "#90EE90", x: 200, y: 80 },

            { name: "Fear", color: "#5F9EA0", x: 320, y: 80 },

            { name: "Surprise", color: "#87CEEB", x: 440, y: 80 },

            { name: "Sadness", color: "#4682B4", x: 80, y: 200 },

            { name: "Disgust", color: "#9370DB", x: 200, y: 200 },

            { name: "Anger", color: "#FF6347", x: 320, y: 200 },

            { name: "Anticipation", color: "#FFA500", x: 440, y: 200 }

        ];

        function drawEmotions() {

            ctx.clearRect(0, 0, canvas.width, canvas.height);

            emotions.forEach(emotion => {

                ctx.beginPath();

                ctx.arc(emotion.x, emotion.y, 40, 0, Math.PI \* 2);

                ctx.fillStyle = emotion.color;

                ctx.fill();

                ctx.closePath();

            });

        }

        canvas.addEventListener("click", function(event) {

            const rect = canvas.getBoundingClientRect();

            const clickX = event.clientX - rect.left;

            const clickY = event.clientY - rect.top;

            emotions.forEach(emotion => {

                const distance = Math.sqrt((clickX - emotion.x) \*\* 2 + (clickY - emotion.y) \*\* 2);

                if (distance < 40) {

                    document.getElementById("emotionText").innerText = "Selected Emotion: " + emotion.name;

                }

            });

        });

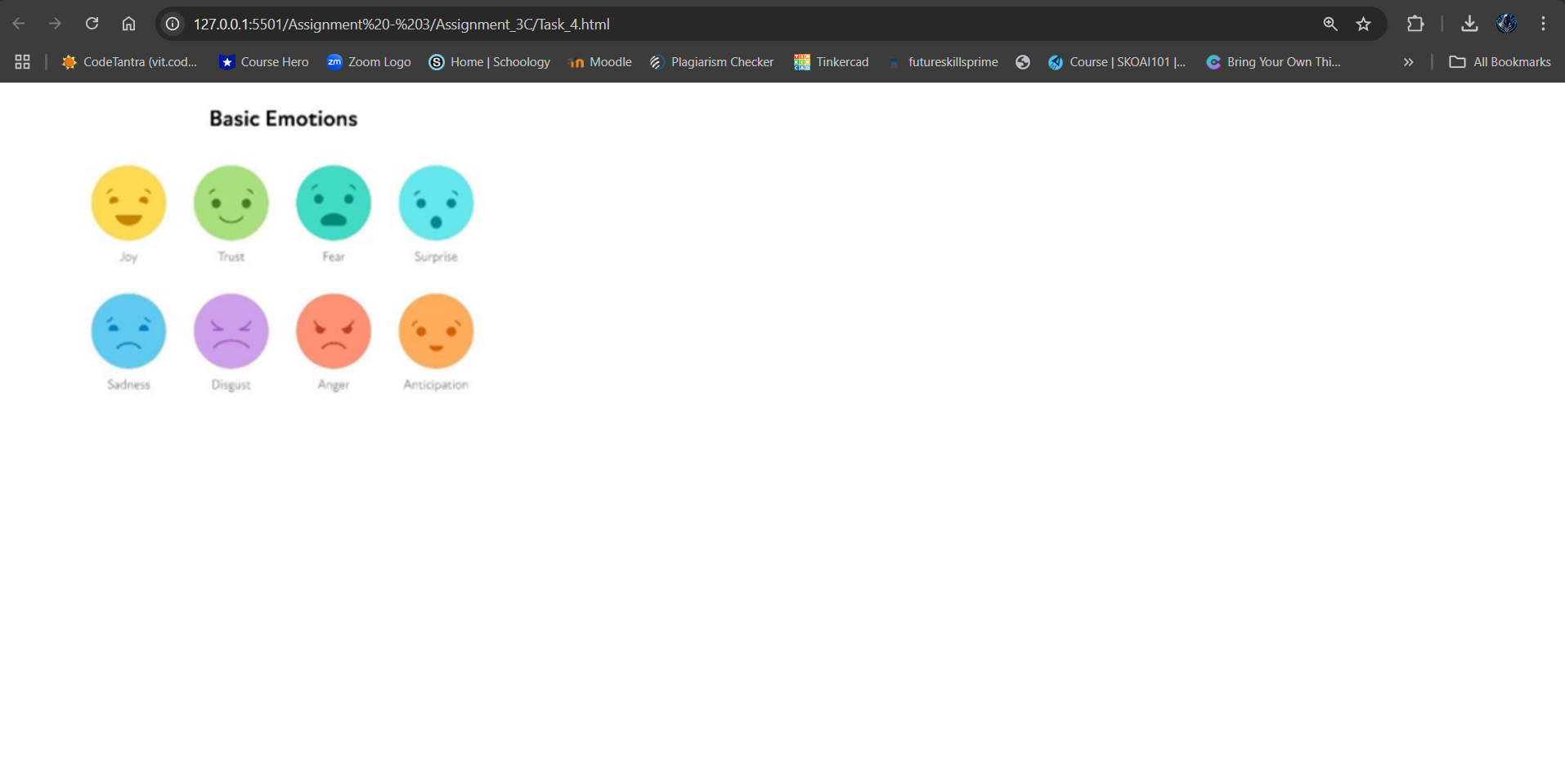
        drawEmotions();

    </script>

</body>

</html>

**OUTPUT:**

****