# Web Programming Lab

**Ex-12**

**Name: Dhruv Vivek Gadre**

**Reg No: 23BCE5132**

**Q1)**

Code:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>23BCE5132</title>

  </head>

  <body>

    <canvas

      id="myCanvas"

      width="800"

      height="500"

      style="border: 2px solid black"

    ></canvas>

    <script>

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

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

      function drawSun() {

        ctx.beginPath();

        ctx.arc(700, 80, 40, 0, Math.PI \* 2);

        ctx.fillStyle = "yellow";

        ctx.fill();

        for (let i = 0; i < 12; i++) {

          let angle = (i \* Math.PI) / 6;

          let x1 = 700 + Math.cos(angle) \* 50;

          let y1 = 80 + Math.sin(angle) \* 50;

          let x2 = 700 + Math.cos(angle) \* 60;

          let y2 = 80 + Math.sin(angle) \* 60;

          ctx.beginPath();

          ctx.moveTo(x1, y1);

          ctx.lineTo(x2, y2);

          ctx.strokeStyle = "yellow";

          ctx.lineWidth = 2;

          ctx.stroke();

        }

      }

      function drawHouse() {

        ctx.fillStyle = "orange";

        ctx.fillRect(100, 250, 100, 100);

        ctx.strokeRect(100, 250, 100, 100);

        ctx.beginPath();

        ctx.moveTo(100, 250);

        ctx.lineTo(150, 200);

        ctx.lineTo(200, 250);

        ctx.closePath();

        ctx.fillStyle = "red";

        ctx.fill();

        ctx.stroke();

        ctx.fillStyle = "brown";

        ctx.fillRect(140, 300, 20, 50);

        ctx.fillStyle = "blue";

        ctx.fillRect(110, 270, 20, 20);

      }

      function drawPond() {

        ctx.beginPath();

        ctx.ellipse(400, 350, 150, 70, 0, 0, Math.PI \* 2);

        ctx.fillStyle = "lightblue";

        ctx.fill();

        ctx.strokeStyle = "blue";

        ctx.stroke();

      }

      function drawStones() {

        let stones = [

          { x: 20, y: 420, size: 15 },

          { x: 35, y: 430, size: 20 },

          { x: 50, y: 420, size: 25 },

          { x: 680, y: 460, size: 18 },

          { x: 700, y: 450, size: 22 },

        ];

        ctx.fillStyle = "gray";

        ctx.strokeStyle = "blue";

        stones.forEach((stone) => {

          ctx.beginPath();

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

          ctx.fill();

          ctx.stroke();

        });

      }

      function drawDuck() {

        ctx.beginPath();

        ctx.arc(600, 400, 25, 0, Math.PI \* 2);

        ctx.fillStyle = "yellow";

        ctx.fill();

        ctx.stroke();

        ctx.beginPath();

        ctx.arc(630, 390, 15, 0, Math.PI \* 2);

        ctx.fill();

        ctx.stroke();

*// Eye*

        ctx.beginPath();

        ctx.arc(635, 385, 3, 0, Math.PI \* 2);

        ctx.fillStyle = "black";

        ctx.fill();

*// Beak*

        ctx.beginPath();

        ctx.moveTo(645, 390);

        ctx.lineTo(660, 395);

        ctx.lineTo(645, 400);

        ctx.closePath();

        ctx.fillStyle = "orange";

        ctx.fill();

      }

      function drawFlower() {

        ctx.beginPath();

        ctx.moveTo(700, 450);

        ctx.lineTo(700, 400);

        ctx.lineWidth = 8;

        ctx.strokeStyle = "green";

        ctx.stroke();

        ctx.beginPath();

        ctx.ellipse(685, 420, 15, 10, Math.PI / 4, 0, Math.PI \* 2);

        ctx.fillStyle = "green";

        ctx.fill();

        ctx.beginPath();

        ctx.ellipse(715, 420, 15, 10, -Math.PI / 4, 0, Math.PI \* 2);

        ctx.fill();

        for (let i = 0; i < 5; i++) {

          let angle = (i \* Math.PI \* 2) / 5;

          let x = 700 + Math.cos(angle) \* 25;

          let y = 400 + Math.sin(angle) \* 25;

          ctx.beginPath();

          ctx.arc(x, y, 20, 0, Math.PI \* 2);

          ctx.fillStyle = "pink";

          ctx.fill();

        }

        ctx.beginPath();

        ctx.arc(700, 400, 10, 0, Math.PI \* 2);

        ctx.fillStyle = "yellow";

        ctx.fill();

      }

      let boatX = 370;

      let boatY = 340;

      let boatDir = 1;

      function drawBoat() {

        ctx.beginPath();

        ctx.moveTo(boatX, boatY);

        ctx.lineTo(boatX + 70, boatY);

        ctx.lineTo(boatX + 50, boatY + 20);

        ctx.lineTo(boatX + 20, boatY + 20);

        ctx.closePath();

        ctx.fillStyle = "brown";

        ctx.fill();

      }

      function animateBoat() {

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

        drawSun();

        drawHouse();

        drawPond();

        drawStones();

        drawDuck();

        drawFlower();

        boatX += boatDir;

        if (boatX > 430 || boatX < 370) {

          boatDir \*= -1;

        }

        drawBoat();

        requestAnimationFrame(animateBoat);

      }

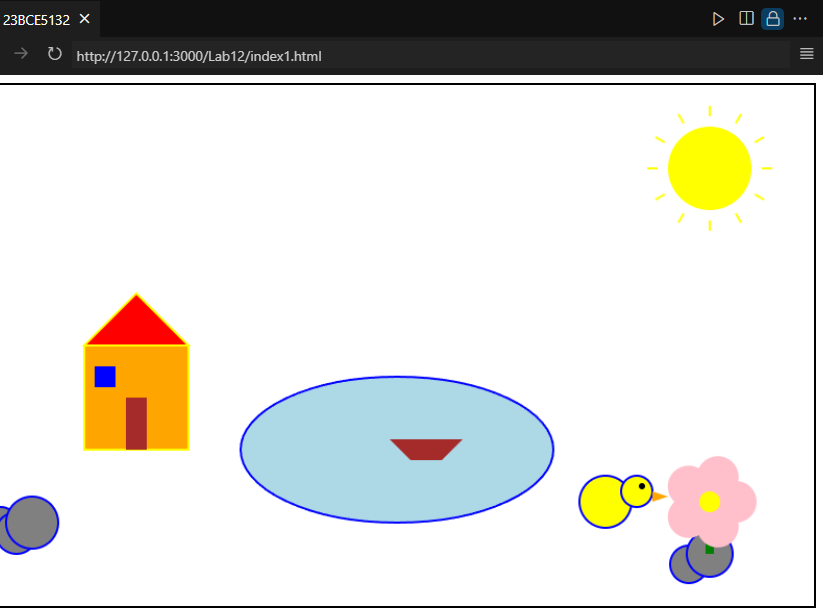
      animateBoat();

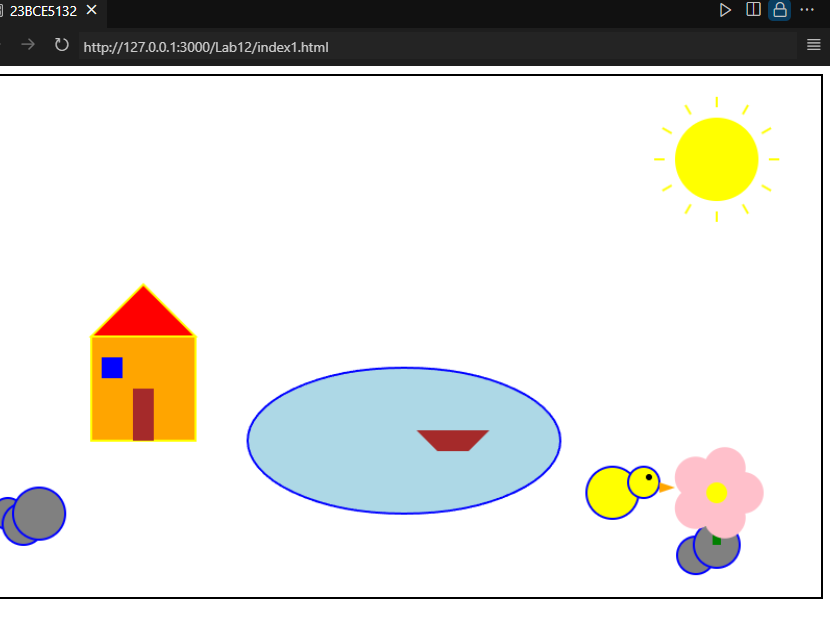
    </script>

  </body>

</html>

Output:

****

****

**Q2 and 3)**

Code:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>23BCE5132</title>

  </head>

  <body>

    <canvas id="clockCanvas" width="400" height="400"></canvas>

    <script>

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

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

      const centerX = canvas.width / 2;

      const centerY = canvas.height / 2;

      const radius = 150;

      function drawClockFace() {

          ctx.beginPath();

          ctx.arc(centerX, centerY, radius, 0, Math.PI \* 2);

          ctx.fillStyle = "beige";

          ctx.fill();

          ctx.lineWidth = 5;

          ctx.strokeStyle = "black";

          ctx.stroke();

      }

      function drawNumbers() {

          ctx.font = "20px Arial";

          ctx.fillStyle = "black";

          ctx.textAlign = "center";

          ctx.textBaseline = "middle";

          for (let i = 1; i <= 12; i++) {

              let angle = (i \* Math.PI) / 6;

              let x = centerX + Math.cos(angle) \* (radius - 35);

              let y = centerY + Math.sin(angle) \* (radius - 35);

              ctx.fillText(i, x, y);

          }

      }

      function drawTicks() {

          for (let i = 0; i < 60; i++) {

              let angle = (i \* Math.PI) / 30;

              let x1 = centerX + Math.cos(angle) \* (radius - 10);

              let y1 = centerY + Math.sin(angle) \* (radius - 10);

              let x2 = centerX + Math.cos(angle) \* (radius - (i % 5 === 0 ? 20 : 10));

              let y2 = centerY + Math.sin(angle) \* (radius - (i % 5 === 0 ? 20 : 10));

              ctx.beginPath();

              ctx.moveTo(x1, y1);

              ctx.lineTo(x2, y2);

              ctx.lineWidth = i % 5 === 0 ? 3 : 1;

              ctx.strokeStyle = "black";

              ctx.stroke();

          }

      }

      function drawHand(angle, length, width, color) {

          ctx.beginPath();

          ctx.moveTo(centerX, centerY);

          let x = centerX + Math.cos(angle) \* length;

          let y = centerY + Math.sin(angle) \* length;

          ctx.lineTo(x, y);

          ctx.lineWidth = width;

          ctx.strokeStyle = color;

          ctx.lineCap = "round";

          ctx.stroke();

      }

      function drawClock() {

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

          drawClockFace();

          drawNumbers();

          drawTicks();

          let now = new Date();

          let hours = now.getHours() % 12;

          let minutes = now.getMinutes();

          let seconds = now.getSeconds();

          let hourAngle = ((hours + minutes / 60) \* Math.PI) / 6 - Math.PI / 2;

          let minuteAngle = ((minutes + seconds / 60) \* Math.PI) / 30 - Math.PI / 2;

          let secondAngle = (seconds \* Math.PI) / 30 - Math.PI / 2;

          drawHand(hourAngle, radius \* 0.5, 6, "black");

          drawHand(minuteAngle, radius \* 0.7, 4, "black");

          drawHand(secondAngle, radius \* 0.9, 2, "red");

          ctx.beginPath();

          ctx.arc(centerX, centerY, 5, 0, Math.PI \* 2);

          ctx.fillStyle = "black";

          ctx.fill();

      }

      setInterval(drawClock, 1000);

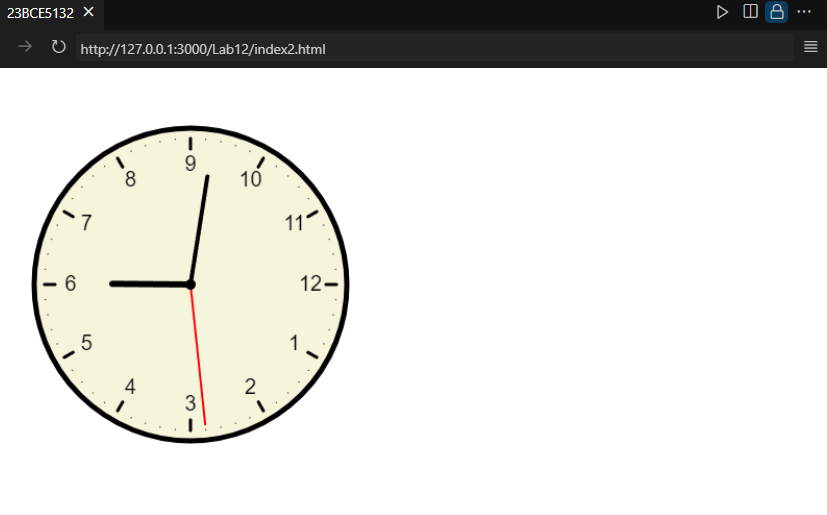
      drawClock();

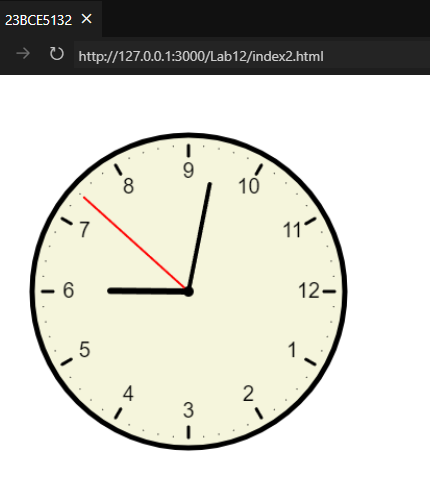
    </script>

  </body>

</html>

Output:

****

****

**Q4)**

Code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>23BCE5132</title>

    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

</head>

<body>

    <h2>Charts</h2>

    <div id="barChart" style="width: 600px; height: 400px;"></div>

    <div id="lineChart" style="width: 600px; height: 400px;"></div>

    <div id="pieChart" style="width: 600px; height: 400px;"></div>

    <div id="donutChart" style="width: 600px; height: 400px;"></div>

    <script>

        const categories = ["A", "B", "C", "D", "E"];

        const values = [10, 20, 30, 40, 50];

        const colors = ['red', 'blue', 'green', 'purple', 'orange'];

        Plotly.newPlot("barChart", [{

            x: categories,

            y: values,

            type: "bar",

            marker: { color: colors }

        }], { title: "Bar Chart", xaxis: { title: "Categories" }, yaxis: { title: "Values" } });

        Plotly.newPlot("lineChart", [{

            x: categories,

            y: values,

            type: "scatter",

            mode: "lines+markers",

            marker: { color: "blue" }

        }], { title: "Line Chart", xaxis: { title: "Categories" }, yaxis: { title: "Values" } });

        Plotly.newPlot("pieChart", [{

            labels: categories,

            values: values,

            type: "pie",

            marker: { colors: colors }

        }], { title: "Pie Chart" });

        Plotly.newPlot("donutChart", [{

            labels: categories,

            values: values,

            type: "pie",

            hole: 0.4,

            marker: { colors: colors }

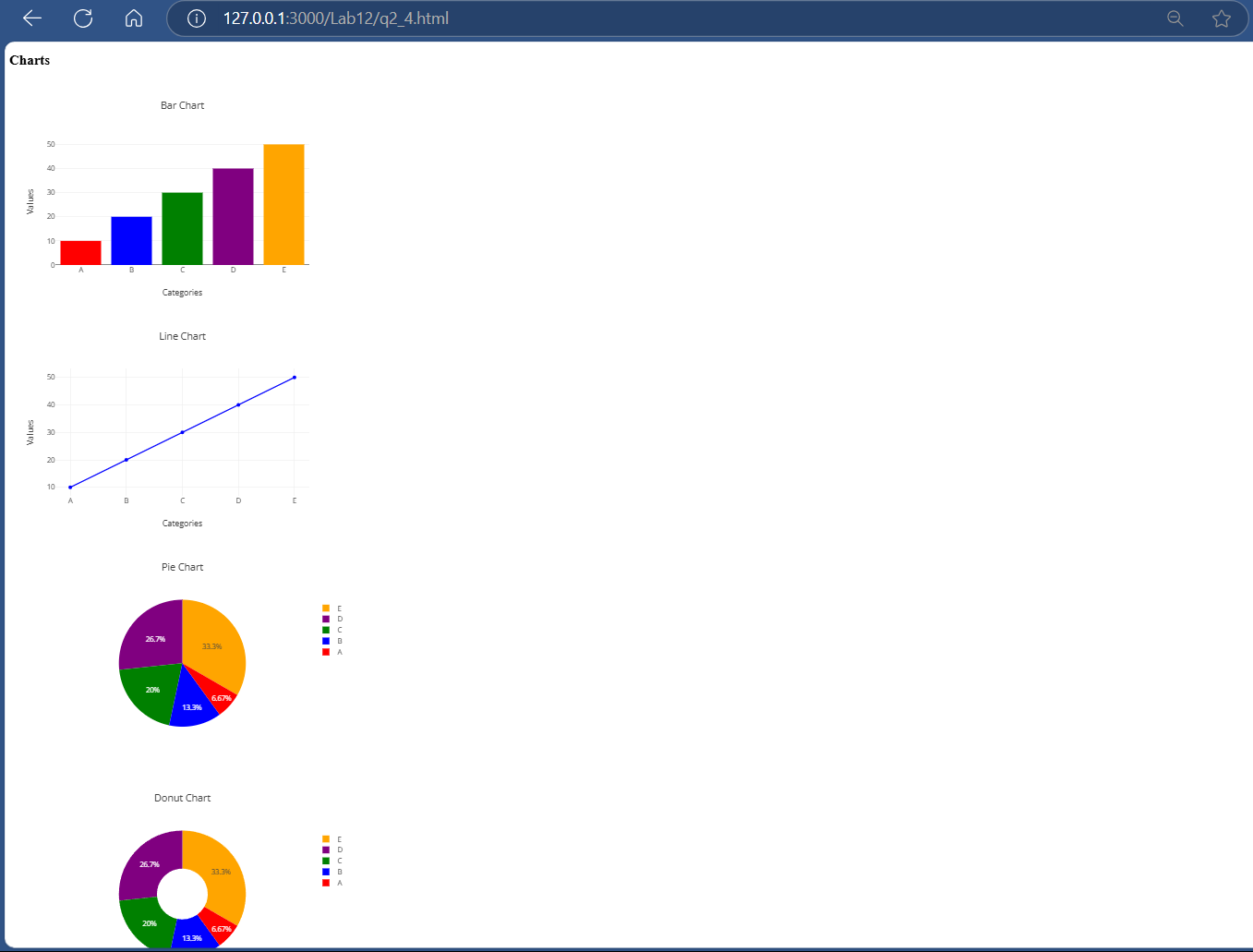
        }], { title: "Donut Chart" });

    </script>

</body>

</html>

Output:

**  
Q5)**

Code:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>23BCE5132</title>

    <style>

        .container {

            position: relative;

            width: 400px;

            height: 300px;

            border: 2px solid black;

        }

        .box {

            width: 100px;

            height: 100px;

            position: absolute;

            text-align: center;

            line-height: 100px;

            font-weight: bold;

            color: white;

        }

        #box1 { background: red; top: 50px; left: 50px; z-index: 1; }

        #box2 { background: blue; top: 80px; left: 80px; z-index: 2; }

        #box3 { background: green; top: 110px; left: 110px; z-index: 3; }

    </style>

</head>

<body>

    <h2>Control the Z-index</h2>

    <div class="container">

        <div id="box1" class="box">Box 1</div>

        <div id="box2" class="box">Box 2</div>

        <div id="box3" class="box">Box 3</div>

    </div>

    <div>

        <label for="zBox">Select Box:</label>

        <select id="zBox">

            <option value="box1">Box 1</option>

            <option value="box2">Box 2</option>

            <option value="box3">Box 3</option>

        </select>

        <label for="zValue">Set z-index:</label>

        <input type="number" id="zValue" value="1">

        <button onclick="changeZIndex()">Update</button>

    </div>

    <h3>Current z-index values:</h3>

    <p id="zValues"></p>

    <script>

        function changeZIndex() {

            let boxId = document.getElementById("zBox").value;

            let zIndex = document.getElementById("zValue").value;

            document.getElementById(boxId).style.zIndex = zIndex;

            updateZValues();

        }

        function updateZValues() {

            let zInfo = `

                Box 1: ${document.getElementById("box1").style.zIndex}<br>

                Box 2: ${document.getElementById("box2").style.zIndex}<br>

                Box 3: ${document.getElementById("box3").style.zIndex}

            `;

            document.getElementById("zValues").innerHTML = zInfo;

        }

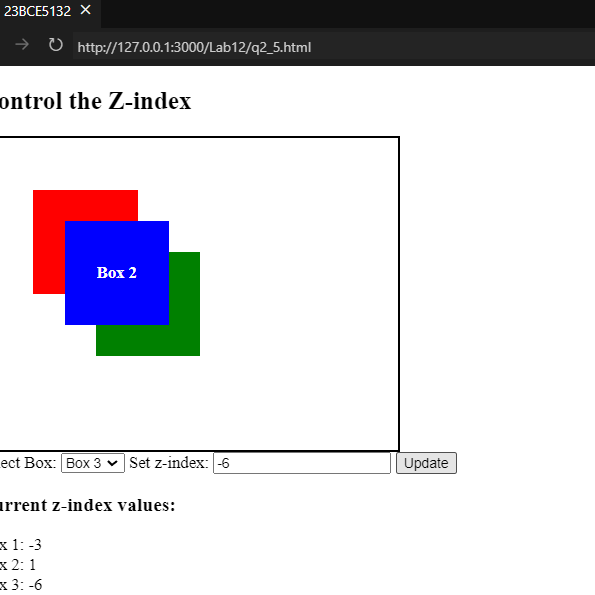
        updateZValues();

    </script>

</body>

</html>

Output:

****