Name: Likhith B A
Reg. No: 23BCE1098

# Experiment 11

# JavaScript DOM

## Question 01 and 02

### Code:

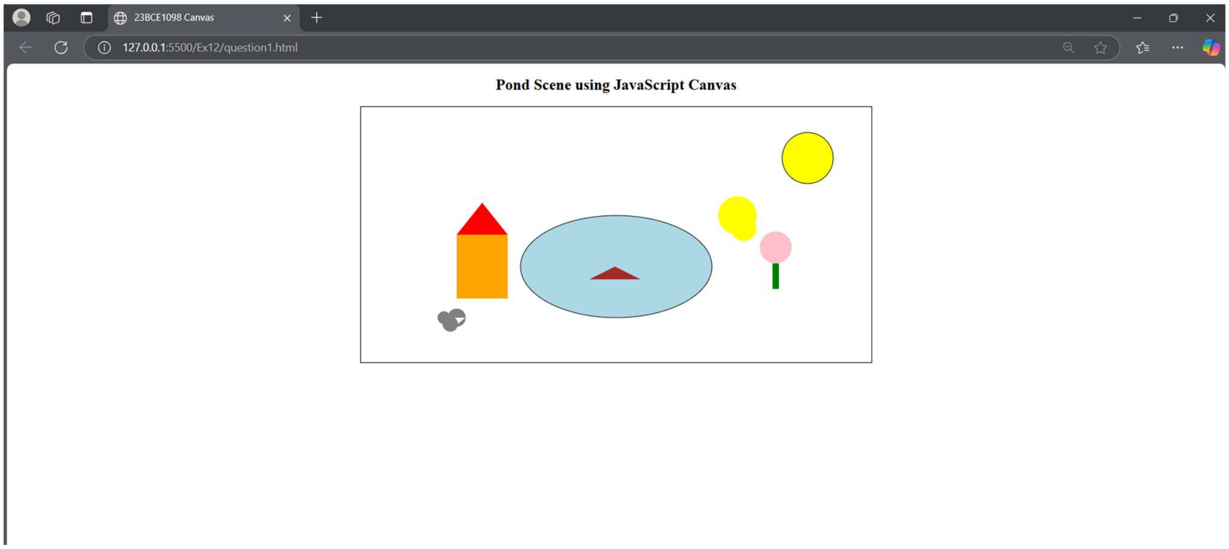
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>23BCE1098 Canvas</title>
    <style>
      body {
        text-align: center;
      }
      canvas {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h2>Pond Scene using JavaScript Canvas</h2>
    <canvas id="pondCanvas" width="800" height="400"></canvas>
    <script src="script.js"></script>
  </body>
</html>
```

### JAVASCRIPT:

```
const canvas = document.getElementById("pondCanvas");
      const ctx = canvas.getContext("2d");
      let boatX = 350,
        direction = 1;
      function drawScene() {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        ctx.fillStyle = "lightblue";
        ctx.beginPath();
        ctx.ellipse(400, 250, 150, 80, 0, 0, Math.PI * 2);
        ctx.fill();
        ctx.stroke();
        ctx.fillStyle = "brown";
ctx.beginPath();
ctx.moveTo(boatX, 250);
ctx.lineTo(boatX + 40, 270);
ctx.lineTo(boatX - 40, 270);
ctx.closePath();
ctx.fill();
ctx.fillStyle = "yellow";
ctx.beginPath();
ctx.arc(700, 80, 40, 0, Math.PI * 2);
ctx.fill();
ctx.stroke();
ctx.fillStyle = "orange";
ctx.fillRect(150, 200, 80, 100);
ctx.fillStyle = "red";
ctx.beginPath();
ctx.moveTo(150, 200);
ctx.lineTo(230, 200);
ctx.lineTo(190, 150);
ctx.closePath();
ctx.fill();
ctx.fillStyle = "yellow";
ctx.beginPath();
ctx.arc(590, 170, 30, 0, Math.PI * 2);
ctx.fill();
ctx.arc(600, 190, 20, 0, Math.PI * 2);
ctx.fill();
ctx.fillStyle = "red";
ctx.beginPath();
ctx.moveTo(610, 190);
ctx.lineTo(620, 200);
ctx.lineTo(625, 205);
ctx.closePath();
ctx.fill();
ctx.fillStyle = "pink";
ctx.beginPath();
ctx.arc(650, 220, 25, 0, Math.PI * 2);
ctx.fill();
ctx.fillStyle = "green";
ctx.fillRect(645, 245, 10, 40);
ctx.fillStyle = "gray";
ctx.beginPath();
ctx.arc(130, 330, 10, 0, Math.PI * 2);
ctx.arc(140, 340, 12, 0, Math.PI * 2);
ctx.arc(150, 330, 14, 0, Math.PI * 2);
ctx.fill();
ctx.fill();
```

```
    boatX += direction * 2;
    if (boatX > 450 || boatX < 350) direction *= -1;
    requestAnimationFrame(drawScene);
  }
  drawScene();
```

## Output:



Pond Scene using JavaScript Canvas

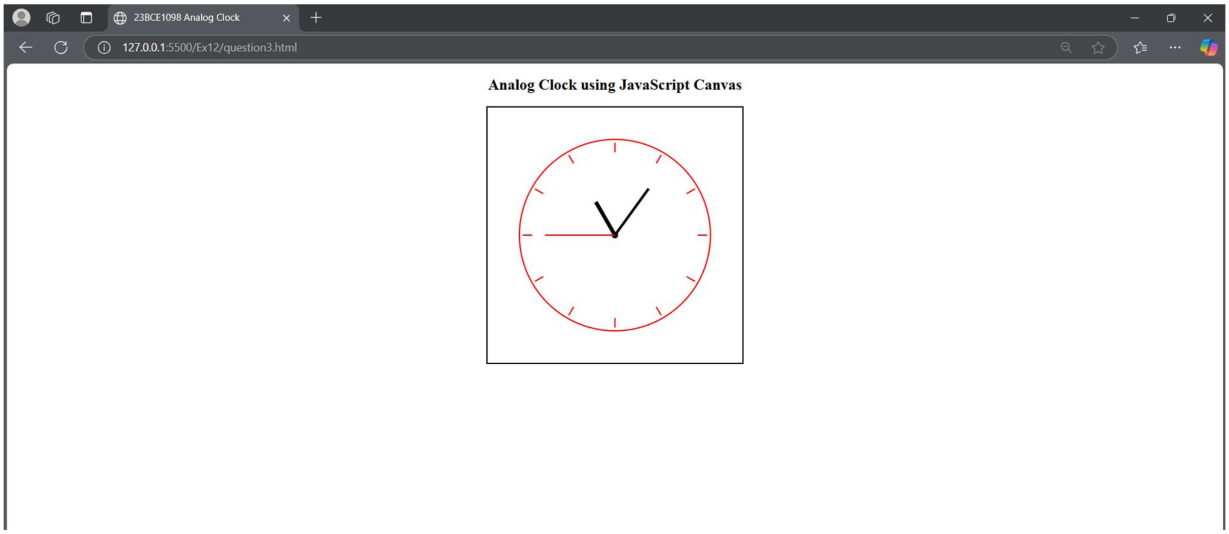
Name: Likhith B A
Reg. No: 23BCE1098

# Question 03

**Code:**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>23BCE1098 Analog Clock</title>
    <style>
      body {
        text-align: center;
      }
      canvas {
        border: 2px solid black;
        background: white;
      }
    </style>
  </head>
  <body>
    <h2>Analog Clock using JavaScript Canvas</h2>
    <canvas id="clockCanvas" width="400" height="400"></canvas>
    <script>
      const canvas = document.getElementById("clockCanvas");
      const ctx = canvas.getContext("2d");
      function drawClock() {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        ctx.fillStyle = "white";
        ctx.beginPath();
        ctx.arc(200, 200, 150, 0, Math.PI * 2);
        ctx.fill();
        ctx.stroke();
        ctx.fillStyle = "black";
        ctx.beginPath();
        ctx.arc(200, 200, 5, 0, Math.PI * 2);
        ctx.fill();
        for (let i = 0; i < 12; i++) {
          let angle = (i * Math.PI) / 6;
          let x1 = 200 + Math.cos(angle) * 130;
          let y1 = 200 + Math.sin(angle) * 130;
          let x2 = 200 + Math.cos(angle) * 145;
          let y2 = 200 + Math.sin(angle) * 145;
          ctx.beginPath();
          ctx.moveTo(x1, y1);
          ctx.lineTo(x2, y2);
          ctx.stroke();
        }
        let now = new Date();
        let secAngle = (now.getSeconds() / 60) * 2 * Math.PI - Math.PI / 2;
        let minAngle = (now.getMinutes() / 60) * 2 * Math.PI - Math.PI / 2;
        let hourAngle =
          ((now.getHours() % 12) / 12) * 2 * Math.PI - Math.PI / 2;
        drawHand(hourAngle, 60, 6);
        drawHand(minAngle, 90, 4);
        drawHand(secAngle, 110, 2, "red");
        requestAnimationFrame(drawClock);
      }
      function drawHand(angle, length, width, color = "black") {
        ctx.strokeStyle = color;
        ctx.lineWidth = width;
        ctx.beginPath();
        ctx.moveTo(200, 200);
        ctx.lineTo(
          200 + Math.cos(angle) * length,
          200 + Math.sin(angle) * length
        );
        ctx.stroke();
      }
      drawClock();
    </script>
  </body>
</html>
```

## Output:



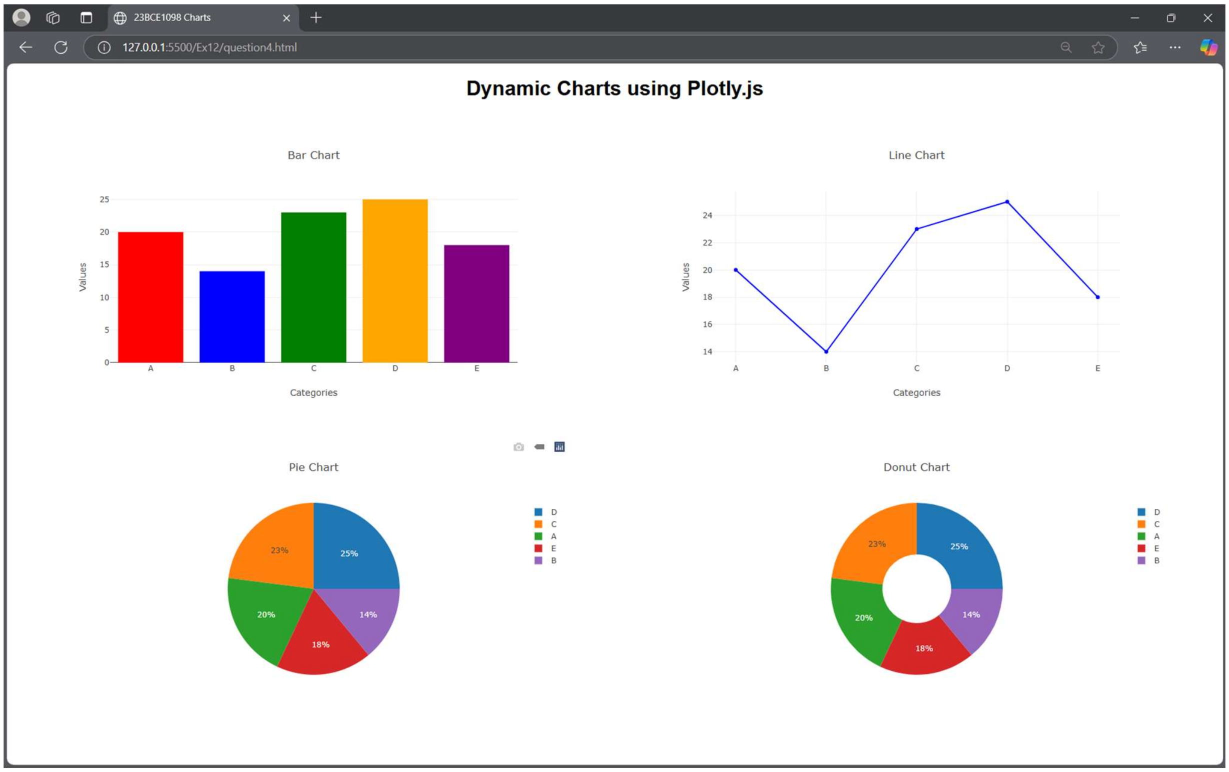
Analog Clock using JavaScript Canvas

# Question 04

**Code:**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>23BCE1098 Charts</title>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
    <style>
      body {
        text-align: center;
        font-family: Arial, sans-serif;
      }
      .chart-container {
        width: 800px;
        margin: 20px auto;
      }
      .box{
        display: flex;
        flex-wrap: wrap;
      }
    </style>
  </head>
  <body>
    <h1>Dynamic Charts using Plotly.js</h1>
    <div class="box">
      <div id="barChart" class="chart-container"></div>
      <div id="lineChart" class="chart-container"></div>
      <div id="pieChart" class="chart-container"></div>
      <div id="donutChart" class="chart-container"></div>
    </div>
    <script>
      function createCharts() {
        let xData = ["A", "B", "C", "D", "E"];
        let yData = [20, 14, 23, 25, 18];
        Plotly.newPlot(
          "barChart",
          [
            {
              x: xData,
              y: yData,
              type: "bar",
      marker: { color: ["red", "blue", "green", "orange", "purple"] },
    },
  ],
  {
    title: "Bar Chart",
    xaxis: { title: "Categories" },
    yaxis: { title: "Values" },
  }
);
Plotly.newPlot(
  "lineChart",
  [
    {
      x: xData,
      y: yData,
      type: "scatter",
      mode: "lines+markers",
      marker: { color: "blue" },
    },
  ],
  {
    title: "Line Chart",
    xaxis: { title: "Categories" },
    yaxis: { title: "Values" },
  }
);
Plotly.newPlot(
  "pieChart",
  [
    {
      labels: xData,
      values: yData,
      type: "pie",
    },
  ],
  { title: "Pie Chart" }
);
Plotly.newPlot(
  "donutChart",
  [
    {
      labels: xData,
      values: yData,
      type: "pie",
      hole: 0.4,
    },
          ],
          { title: "Donut Chart" }
        );
      }
      createCharts();
    </script>
  </body>
</html>
```

# Output:



Dynamic Charts using Plotly.js

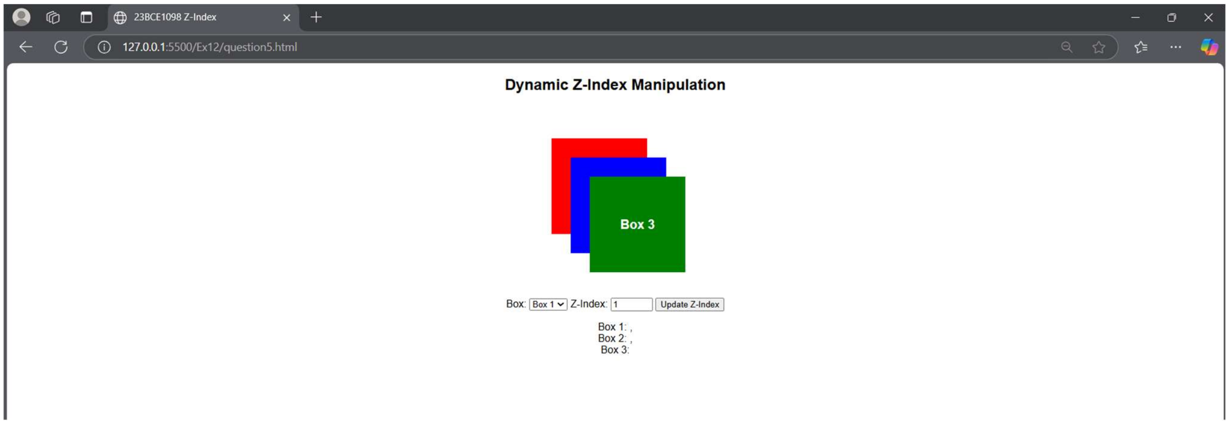
Name: Likhith B A
Reg. No: 23BCE1098

# Question 05

**Code:**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>23BCE1098 Z-Index</title>
    <style>
      body {
        text-align: center;
        font-family: Arial, sans-serif;
      }
      .container {
        position: relative;
        width: 300px;
        height: 300px;
        margin: auto;
      }
      .box {
        width: 150px;
        height: 150px;
        position: absolute;
        text-align: center;
        line-height: 150px;
        font-size: 20px;
        color: white;
        font-weight: bold;
      }
      #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>Dynamic Z-Index Manipulation</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
        >Box:
        <select id="boxSelect">
          <option value="box1">Box 1</option>
          <option value="box2">Box 2</option>
          <option value="box3">Box 3</option>
        </select>
      </label>
      <label
        >Z-Index:
        <input type="number" id="zIndexInput" value="1" min="1" max="10"
      /></label>
      <button onclick="changeZIndex()">Update Z-Index</button>
    </div>
    <p id="zIndexDisplay"></p>
    <script>
      function updateDisplay() {
        document.getElementById("zIndexDisplay").innerText = `Box 1: ${
          document.getElementById("box1").style.zIndex
        },
            Box 2: ${document.getElementById("box2").style.zIndex},
            Box 3: ${document.getElementById("box3").style.zIndex}`;
      }
      function changeZIndex() {
        let selectedBox = document.getElementById("boxSelect").value;
        let newZIndex = document.getElementById("zIndexInput").value;
        document.getElementById(selectedBox).style.zIndex = newZIndex;
        updateDisplay();
      }
      updateDisplay();
    </script>
  </body>
</html>
```

# Output:



Dynamic Z-Index Manipulation

Box: Box 1 Z-Index: 1 Update Z-Index

Box 1: ,
Box 2: ,
Box 3: