

Programme	: B.Tech.	Semester	: Winter24-25
Course	: BCSE203E: Web Programming Lab	Slot	: TE1/TE2
Faculty	: Dr. LM Jenila Livingston	Marks	: 10

Name: Sara Arora

Register no: 23BCE5036

Exercise 12

Question1:

Code:

- Html code:

```

EXPLORER  ...  index.html  # style.css  JS script.js
└─ 23BCE5036_EX11
  └─ index.html
     JS script.js
     # style.css

index.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>23BCE5036</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <canvas id="sceneCanvas" width="800" height="500"></canvas>
11     <script src="script.js"></script>
12 </body>
13 </html>
14

```

- Css file:

```

EXPLORER  ...  index.html  # style.css  JS script.js
└─ 23BCE5036_EX11
  └─ index.html
     JS script.js
     # style.css

# style.css > ...
1  body {
2      display: flex;
3      justify-content: center;
4      align-items: center;
5      height: 100vh;
6      background-color: white;
7  }
8
9  canvas {
10     border: 1px solid black;
11 }
12

```

- Javascript file:

```

EXPLORER  ...  index.html  # style.css  JS script.js  X
23BCE5036_EX11
  index.html
  JS script.js
  # style.css

JS script.js > ...
1  const canvas = document.getElementById("sceneCanvas");
2  const ctx = canvas.getContext("2d");
3
4  function drawPond() {
5      ctx.fillStyle = "#cce7ff";
6      ctx.strokeStyle = "blue";
7      ctx.beginPath();
8      ctx.ellipse(400, 300, 150, 70, 0, 0, Math.PI * 2);
9      ctx.fill();
10     ctx.stroke();
11 }
12
13 function drawBoat() {
14     ctx.fillStyle = "brown";
15     ctx.beginPath();
16     ctx.moveTo(370, 290);
17     ctx.lineTo(430, 290);
18     ctx.lineTo(420, 300);
19     ctx.lineTo(380, 300);
20     ctx.closePath();
21     ctx.fill();
22 }
23
24 function drawDuck() {
25     ctx.fillStyle = "yellow";
26     ctx.beginPath();
27     ctx.arc(600, 350, 20, 0, Math.PI * 2);
28     ctx.fill();
29
30     ctx.beginPath();
31     ctx.arc(630, 360, 10, 0, Math.PI * 2);
32     ctx.fill();
33
34     ctx.fillStyle = "black";
35     ctx.beginPath();
36     ctx.arc(635, 355, 2, 0, Math.PI * 2);
37     ctx.fill();
38 }
39
40 function drawSun() {
41     ctx.fillStyle = "yellow";
42     ctx.beginPath();
43     ctx.arc(700, 80, 40, 0, Math.PI * 2);
44     ctx.fill();
45     ctx.strokeStyle = "yellow";
46     for (let i = 0; i < 12; i++) {
47         let angle = (Math.PI / 6) * i;
48         let x1 = 700 + 45 * Math.cos(angle);
49         let y1 = 80 + 45 * Math.sin(angle);
50         let x2 = 700 + 60 * Math.cos(angle);
51         let y2 = 80 + 60 * Math.sin(angle);
52         ctx.beginPath();
53         ctx.moveTo(x1, y1);
54         ctx.lineTo(x2, y2);
55         ctx.stroke();

```

```

23BCE5036_EX11
  index.html
  JS script.js
  # style.css

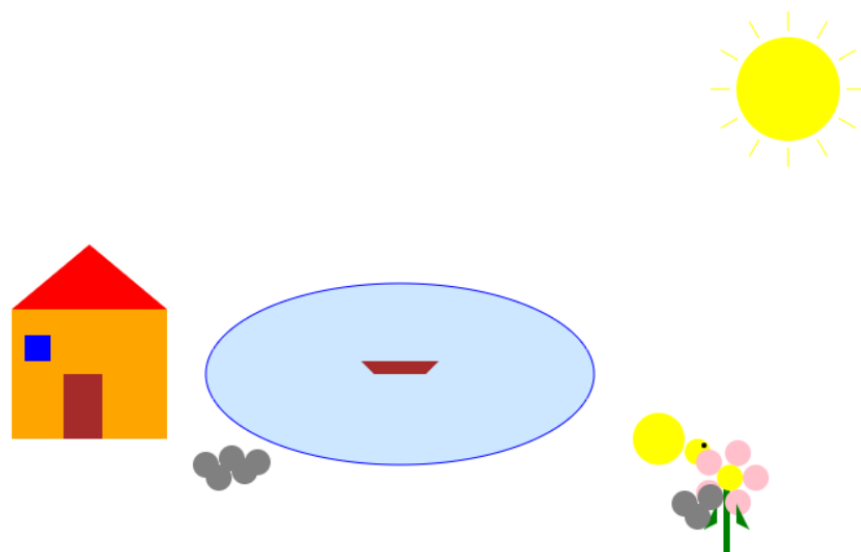
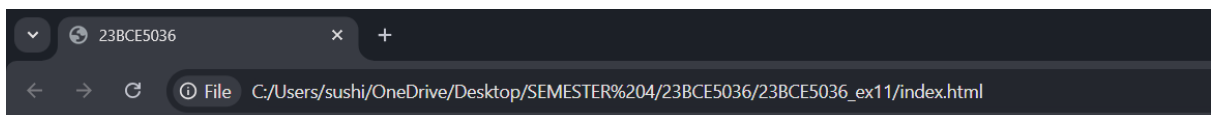
JS script.js > drawHouse
59  function drawHouse() {
60      ctx.fillStyle = "orange";
61      ctx.fillRect(100, 250, 120, 100);
62      ctx.fillStyle = "red";
63      ctx.beginPath();
64      ctx.moveTo(100, 250);
65      ctx.lineTo(160, 200);
66      ctx.lineTo(220, 250);
67      ctx.closePath();
68      ctx.fill();
69
70      ctx.fillStyle = "brown";
71      ctx.fillRect(140, 300, 30, 50);
72
73      ctx.fillStyle = "blue";
74      ctx.fillRect(110, 270, 20, 20);
75  }

```

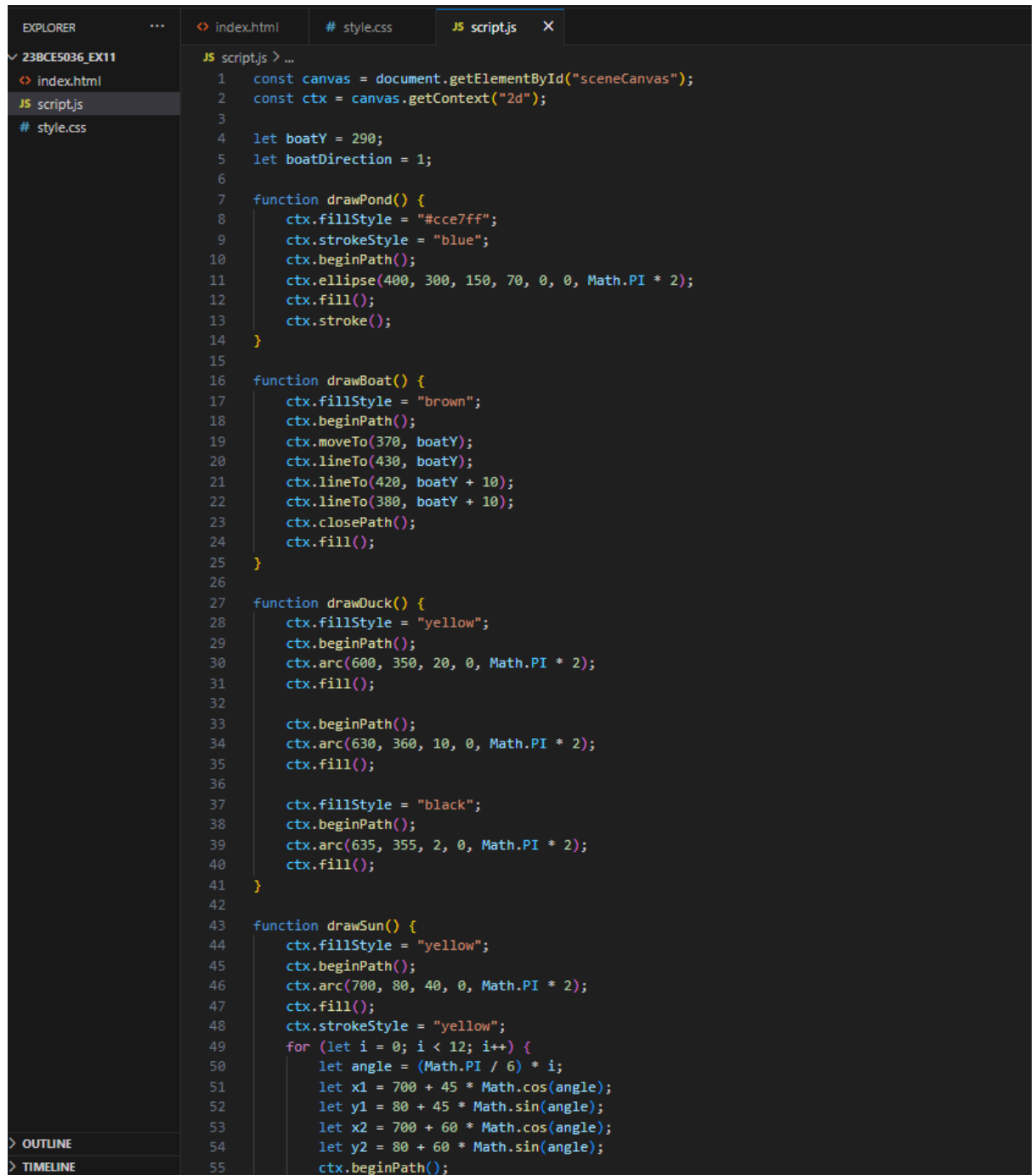
```
23BCE5036_EX11
index.html
script.js
style.css

JS script.js > ...
77 function drawFlower() {
78   ctx.fillStyle = "green";
79   ctx.fillRect(650, 380, 5, 60);
80
81   ctx.fillStyle = "pink";
82   for (let i = 0; i < 5; i++) {
83     let angle = (Math.PI / 2.5) * i;
84     let x = 655 + 20 * Math.cos(angle);
85     let y = 380 + 20 * Math.sin(angle);
86     ctx.beginPath();
87     ctx.arc(x, y, 10, 0, Math.PI * 2);
88     ctx.fill();
89   }
90
91   ctx.fillStyle = "yellow";
92   ctx.beginPath();
93   ctx.arc(655, 380, 10, 0, Math.PI * 2);
94   ctx.fill();
95
96   ctx.fillStyle = "green";
97   ctx.beginPath();
98   ctx.moveTo(645, 400);
99   ctx.lineTo(635, 420);
100  ctx.lineTo(645, 415);
101  ctx.fill();
102
103  ctx.beginPath();
104  ctx.moveTo(660, 400);
105  ctx.lineTo(670, 420);
106  ctx.lineTo(660, 415);
107  ctx.fill();
108 }
109 function drawStones() {
110   ctx.fillStyle = "gray";
111   let positions = [[250, 370], [260, 380], [270, 365], [280, 375], [290, 368]];
112   positions.forEach(pos => {
113     ctx.beginPath();
114     ctx.arc(pos[0], pos[1], 10, 0, Math.PI * 2);
115     ctx.fill();
116   });
117
118   let positions2 = [[620, 400], [630, 410], [640, 395]];
119   positions2.forEach(pos => {
120     ctx.beginPath();
121     ctx.arc(pos[0], pos[1], 10, 0, Math.PI * 2);
122     ctx.fill();
123   });
124 }
125 drawPond();
126 drawBoat();
127 drawDuck();
128 drawSun();
129 drawHouse();
130 drawFlower();
131 drawStones();
```

output:



Question 2:



```
EXPLORER    ...    index.html    # style.css    JS script.js    X
23BCE5036_EX11
  index.html
  JS script.js
  # style.css

JS script.js > ...
1  const canvas = document.getElementById("sceneCanvas");
2  const ctx = canvas.getContext("2d");
3
4  let boatY = 290;
5  let boatDirection = 1;
6
7  function drawPond() {
8    ctx.fillStyle = "#cce7ff";
9    ctx.strokeStyle = "blue";
10   ctx.beginPath();
11   ctx.ellipse(400, 300, 150, 70, 0, 0, Math.PI * 2);
12   ctx.fill();
13   ctx.stroke();
14 }
15
16 function drawBoat() {
17   ctx.fillStyle = "brown";
18   ctx.beginPath();
19   ctx.moveTo(370, boatY);
20   ctx.lineTo(430, boatY);
21   ctx.lineTo(420, boatY + 10);
22   ctx.lineTo(380, boatY + 10);
23   ctx.closePath();
24   ctx.fill();
25 }
26
27 function drawDuck() {
28   ctx.fillStyle = "yellow";
29   ctx.beginPath();
30   ctx.arc(600, 350, 20, 0, Math.PI * 2);
31   ctx.fill();
32
33   ctx.beginPath();
34   ctx.arc(630, 360, 10, 0, Math.PI * 2);
35   ctx.fill();
36
37   ctx.fillStyle = "black";
38   ctx.beginPath();
39   ctx.arc(635, 355, 2, 0, Math.PI * 2);
40   ctx.fill();
41 }
42
43 function drawSun() {
44   ctx.fillStyle = "yellow";
45   ctx.beginPath();
46   ctx.arc(700, 80, 40, 0, Math.PI * 2);
47   ctx.fill();
48   ctx.strokeStyle = "yellow";
49   for (let i = 0; i < 12; i++) {
50     let angle = (Math.PI / 6) * i;
51     let x1 = 700 + 45 * Math.cos(angle);
52     let y1 = 80 + 45 * Math.sin(angle);
53     let x2 = 700 + 60 * Math.cos(angle);
54     let y2 = 80 + 60 * Math.sin(angle);
55     ctx.beginPath();
```

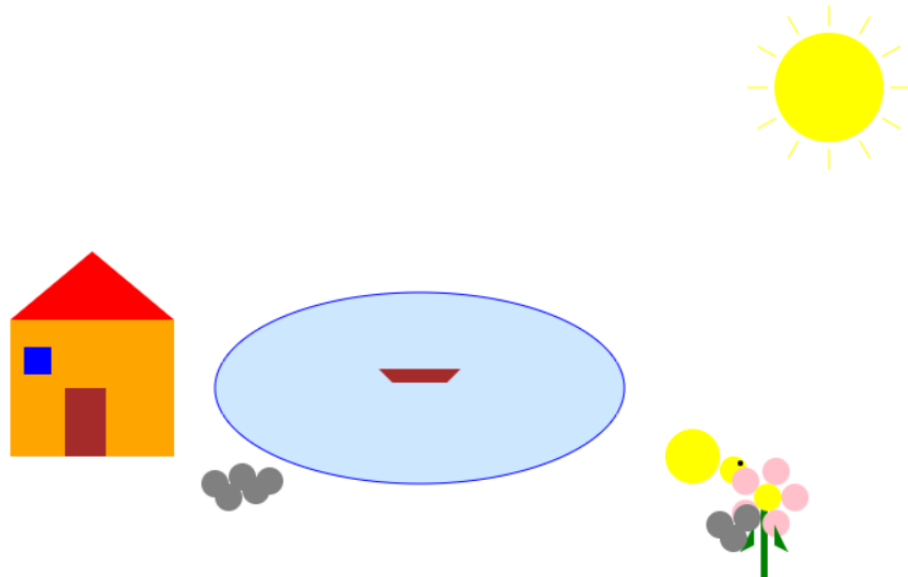
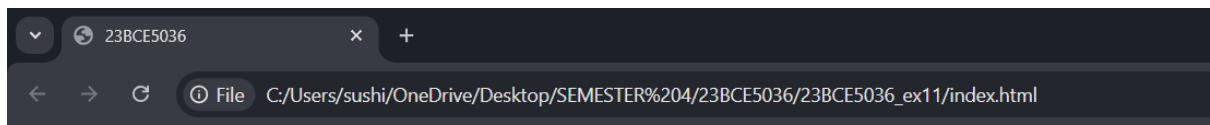
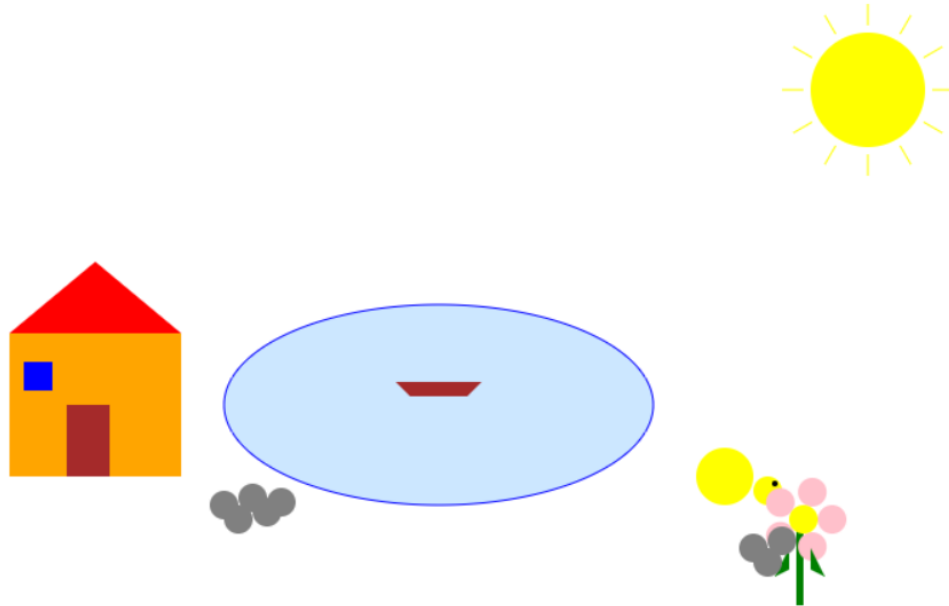
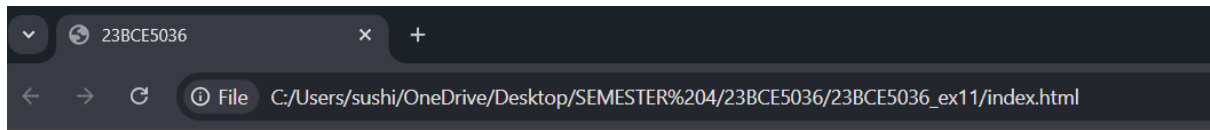
index.html
script.js
style.css

60 }
61
62 function drawHouse() {
63 ctx.fillStyle = "orange";
64 ctx.fillRect(100, 250, 120, 100);
65 ctx.fillStyle = "red";
66 ctx.beginPath();
67 ctx.moveTo(100, 250);
68 ctx.lineTo(160, 200);
69 ctx.lineTo(220, 250);
70 ctx.closePath();
71 ctx.fill();
72
73 ctx.fillStyle = "brown";
74 ctx.fillRect(140, 300, 30, 50);
75
76 ctx.fillStyle = "blue";
77 ctx.fillRect(110, 270, 20, 20);
78 }
79
80 function drawFlower() {
81 ctx.fillStyle = "green";
82 ctx.fillRect(650, 380, 5, 60);
83
84 ctx.fillStyle = "pink";
85 for (let i = 0; i < 5; i++) {
86 let angle = (Math.PI / 2.5) * i;
87 let x = 655 + 20 * Math.cos(angle);
88 let y = 380 + 20 * Math.sin(angle);
89 ctx.beginPath();
90 ctx.arc(x, y, 10, 0, Math.PI * 2);
91 ctx.fill();
92 }
93
94 ctx.fillStyle = "yellow";
95 ctx.beginPath();
96 ctx.arc(655, 380, 10, 0, Math.PI * 2);
97 ctx.fill();
98
99 ctx.fillStyle = "green";
100 ctx.beginPath();
101 ctx.moveTo(645, 400);
102 ctx.lineTo(635, 420);
103 ctx.lineTo(645, 415);
104 ctx.fill();
105
106 ctx.beginPath();
107 ctx.moveTo(660, 400);
108 ctx.lineTo(670, 420);
109 ctx.lineTo(660, 415);
110 ctx.fill();
111 }

23865086 EXH
index.html
script.js
style.css

112 JS scripts / animate
113 function drawStones() {
114 ctx.fillStyle = "gray";
115 let positions = [[250, 370], [260, 380], [270, 365], [280, 375], [290, 368]];
116 positions.forEach(pos => {
117 ctx.beginPath();
118 ctx.arc(pos[0], pos[1], 10, 0, Math.PI * 2);
119 ctx.fill();
120 });
121
122 let positions2 = [[620, 400], [630, 410], [640, 395]];
123 positions2.forEach(pos => {
124 ctx.beginPath();
125 ctx.arc(pos[0], pos[1], 10, 0, Math.PI * 2);
126 ctx.fill();
127 });
128 }
129
130 function animate() {
131 ctx.clearRect(0, 0, canvas.width, canvas.height);
132
133 drawPond();
134 drawBoat();
135 drawDuck();
136 drawSun();
137 drawHouse();
138 drawFlower();
139 drawStones();
140
141 boatY += boatDirection;
142 if (boatY > 295 || boatY < 285) {
143 boatDirection *= -1;
144 }
145
146 requestAnimationFrame(animate);
147 }
148
149 animate();
150

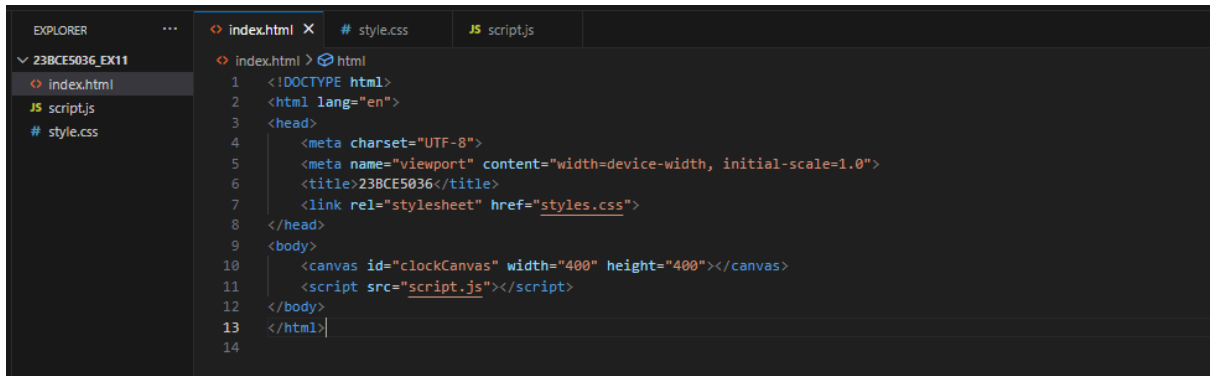
Output:



Question3

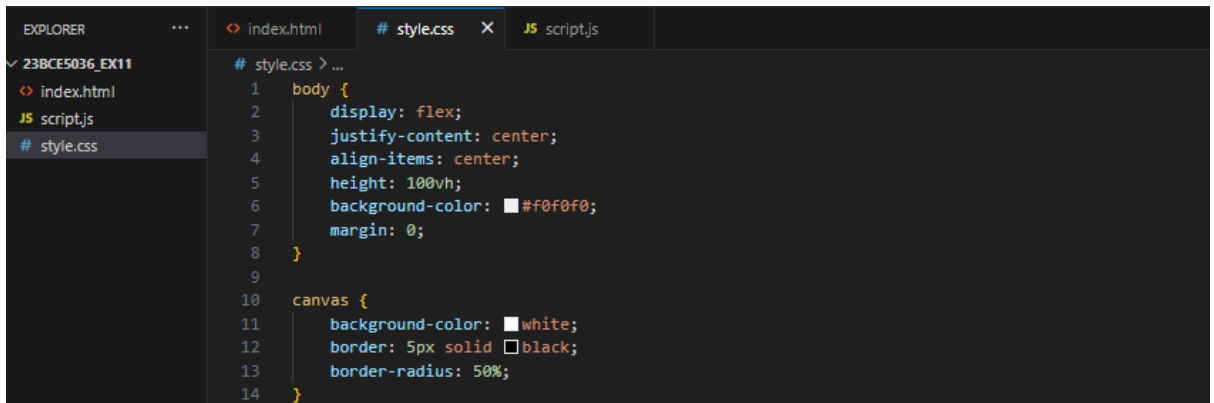
Code:

- Html file:



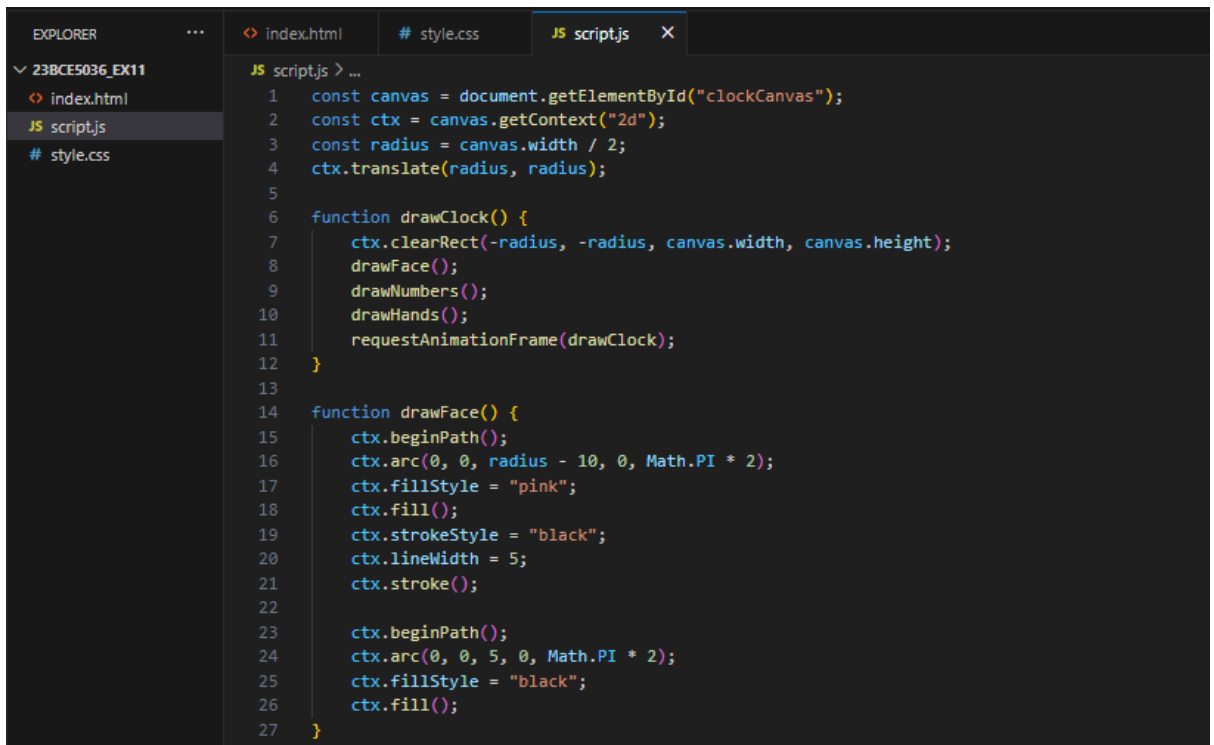
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>23BCE5036</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <canvas id="clockCanvas" width="400" height="400"></canvas>
11  <script src="script.js"></script>
12 </body>
13 </html>
```

- Css file



```
1 body {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   height: 100vh;
6   background-color: #f0f0f0;
7   margin: 0;
8 }
9
10 canvas {
11   background-color: white;
12   border: 5px solid black;
13   border-radius: 50%;
14 }
```

- Javascript file

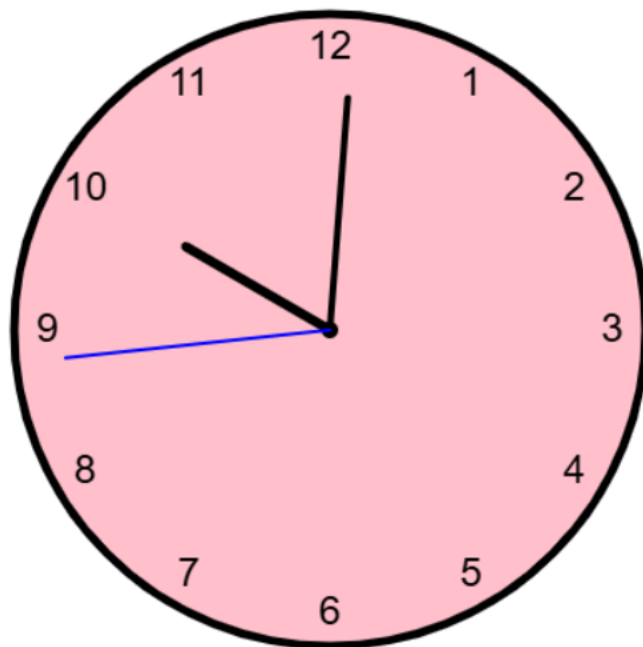
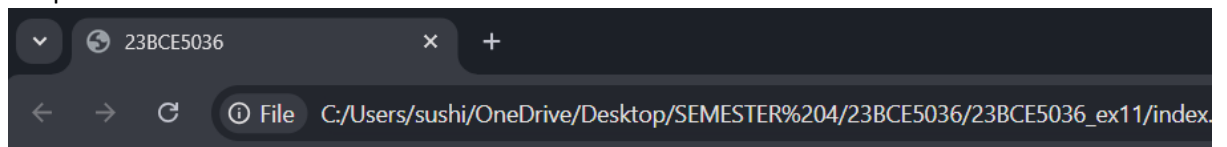


```
1 const canvas = document.getElementById("clockCanvas");
2 const ctx = canvas.getContext("2d");
3 const radius = canvas.width / 2;
4 ctx.translate(radius, radius);
5
6 function drawClock() {
7   ctx.clearRect(-radius, -radius, canvas.width, canvas.height);
8   drawFace();
9   drawNumbers();
10  drawHands();
11  requestAnimationFrame(drawClock);
12 }
13
14 function drawFace() {
15   ctx.beginPath();
16   ctx.arc(0, 0, radius - 10, 0, Math.PI * 2);
17   ctx.fillStyle = "pink";
18   ctx.fill();
19   ctx.strokeStyle = "black";
20   ctx.lineWidth = 5;
21   ctx.stroke();
22
23   ctx.beginPath();
24   ctx.arc(0, 0, 5, 0, Math.PI * 2);
25   ctx.fillStyle = "black";
26   ctx.fill();
27 }
```

```
23BCE5036_EX11
  index.html
  script.js
  style.css

JS script.js > ...
14 function drawFace() {
15     // ...
27 }
28
29 function drawNumbers() {
30     ctx.font = "24px Arial";
31     ctx.fillStyle = "black";
32     ctx.textAlign = "center";
33     ctx.textBaseline = "middle";
34     for (let num = 1; num <= 12; num++) {
35         let angle = ((num - 3) * Math.PI) / 6;
36         let x = Math.cos(angle) * (radius - 30);
37         let y = Math.sin(angle) * (radius - 30);
38         ctx.fillText(num, x, y);
39     }
40 }
41
42 function drawHands() {
43     const now = new Date();
44     const hour = now.getHours() % 12;
45     const minute = now.getMinutes();
46     const second = now.getSeconds();
47
48     drawHand((hour + minute / 60) * 30, radius * 0.5, 6, "black");
49     drawHand((minute + second / 60) * 6, radius * 0.7, 4, "black");
50     drawHand(second * 6, radius * 0.8, 2, "blue");
51 }
52
53 function drawHand(angle, length, width, color) {
54     ctx.beginPath();
55     ctx.lineWidth = width;
56     ctx.lineCap = "round";
57     ctx.strokeStyle = color;
58     ctx.moveTo(0, 0);
59     ctx.lineTo(length * Math.cos((angle - 90) * (Math.PI / 180)), length * Math.sin((angle - 90) * (Math.PI / 180)));
60     ctx.stroke();
61 }
62
63 drawClock();
64
```

Output:



Question 4:

Code:

- Html file:

```
EXPLORER  ...  index.html X  # style.css  JS script.js

23BCE5036_EX11
  index.html
  script.js
  style.css

index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>23BCE5036</title>
7      <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
8      <link rel="stylesheet" href="style.css">
9  </head>
10 <body>
11     <h2>23BCE5036</h2>
12     <div class="chart-container">
13         <div id="barChart"></div>
14         <div id="lineChart"></div>
15         <div id="pieChart"></div>
16         <div id="donutChart"></div>
17     </div>
18     <script src="script.js"></script>
19 </body>
20 </html>
```

- Css File

```
EXPLORER  ...  index.html  # style.css X  JS script.js

23BCE5036_EX11
  index.html
  script.js
  style.css

# style.css > ...
1  body {
2      font-family: Arial, sans-serif;
3      text-align: center;
4      background-color: #f8f9fa;
5  }
6
7  h2 {
8      margin-top: 20px;
9  }
10
11 .chart-container {
12     display: grid;
13     grid-template-columns: repeat(2, 1fr);
14     gap: 20px;
15     padding: 20px;
16 }
17
18 #barChart, #lineChart, #pieChart, #donutChart {
19     width: 100%;
20     height: 400px;
21 }
```

- Javascript file:

```
EXPLORER  ...  index.html  # style.css  JS script.js X

23BCE5036_EX11
  index.html
  script.js
  style.css

JS script.js > drawBarChart > trace > marker > color
1  document.addEventListener("DOMContentLoaded", function() {
2      drawBarChart();
3      drawLineChart();
4      drawPieChart();
5      drawDonutChart();
6  });
7
8  function drawBarChart() {
9      let trace = {
10         x: ["Category A", "Category B", "Category C", "Category D"],
11         y: [10, 15, 7, 12],
12         type: "bar",
13         marker: { color: ["pink", "red", "teal", "orange"] }
14     };
15 }
```

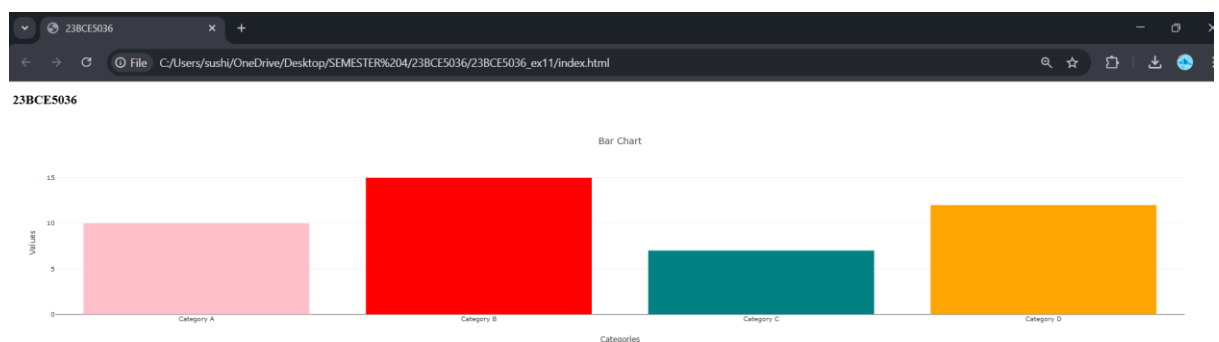
```
23BCE5036_EX11
index.html
JS script.js
# style.css

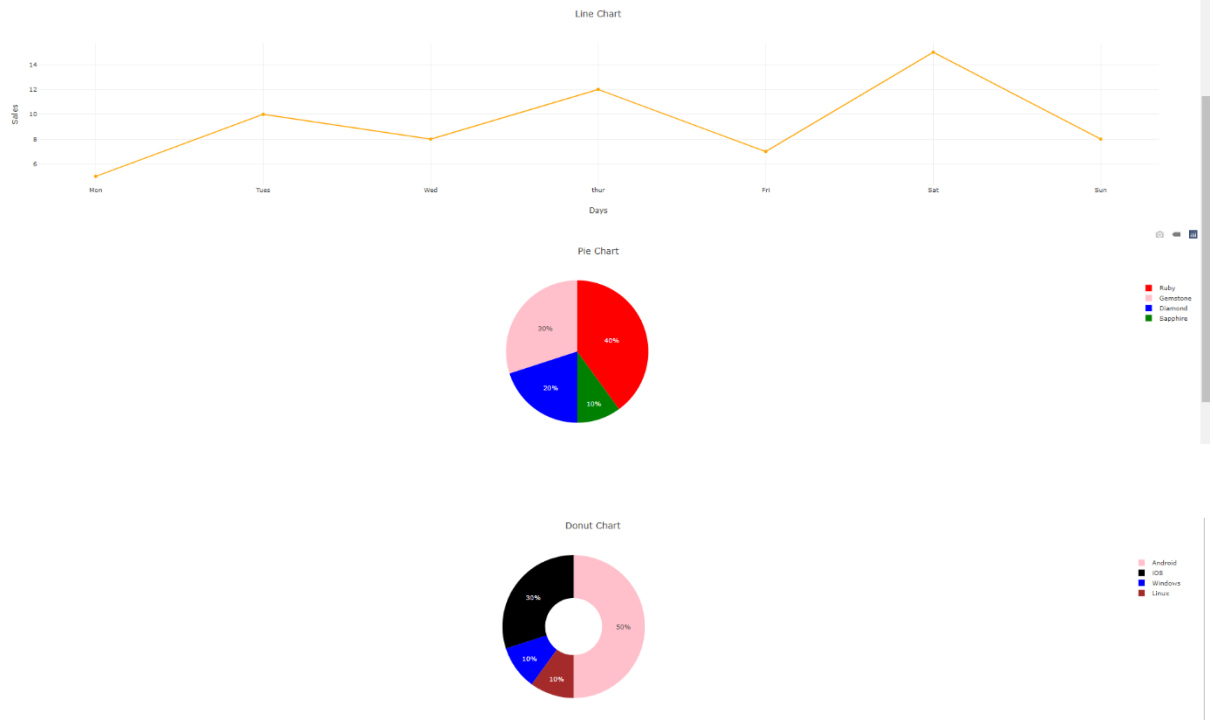
JS script.js
8 function drawBarChart() {
15
16     let layout = {
17         title: "Bar Chart",
18         xaxis: { title: "Categories" },
19         yaxis: { title: "Values" }
20     };
21
22     Plotly.newPlot("barChart", [trace], layout);
23 }
24
25 function drawLineChart() {
26     let trace = {
27         x: ["Mon", "Tues", "Wed", "thur", "Fri", "Sat", "Sun"],
28         y: [5, 10, 8, 12, 7, 15, 8],
29         type: "scatter",
30         mode: "lines+markers",
31         marker: { color: "Orange" }
32     };
33
34     let layout = {
35         title: "Line Chart",
36         xaxis: { title: "Days" },
37         yaxis: { title: "Sales" }
38     };
39
40     Plotly.newPlot("lineChart", [trace], layout);
41 }
42
43 function drawPieChart() {
44     let trace = {
45         labels: ["Gemstone", "Diamond", "Ruby", "Sapphire"],
46         values: [30, 20, 40, 10],
47         type: "pie",
48         marker: { colors: ["pink", "Blue", "Red", "Green"] }
49     };
50
51     let layout = {
52         title: "Pie Chart",
53         showlegend: true
54     };
55 }
```

```
23BCE5036_EX11
index.html
JS script.js
# style.css

JS script.js
43 function drawPieChart() {
55
56     Plotly.newPlot("pieChart", [trace], layout);
57 }
58
59 function drawDonutChart() {
60     let trace = {
61         labels: ["Android", "iOS", "Windows", "Linux"],
62         values: [50, 30, 10, 10],
63         type: "pie",
64         hole: 0.4,
65         marker: { colors: ["pink", "black", "blue", "brown"] }
66     };
67
68     let layout = {
69         title: "Donut Chart",
70         showlegend: true
71     };
72
73     Plotly.newPlot("donutChart", [trace], layout);
74 }
75 }
```

output:





Question 5:

Code:

- Html file

```

EXPLORER
23BCE5036_EX11
  index.html
  script.js
  style.css

index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>23BCE5036</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10   <div class="container">
11     <div id="box1" class="box red">Box 1</div>
12     <div id="box2" class="box blue">Box 2</div>
13     <div id="box3" class="box green">Box 3</div>
14   </div>
15
16   <div class="controls">
17     <label for="elementSelect">Select Element:</label>
18     <select id="elementSelect">
19       <option value="box1">Box 1</option>
20       <option value="box2">Box 2</option>
21       <option value="box3">Box 3</option>
22     </select>
23
24     <label for="zIndexValue">Set Z-Index:</label>
25     <input type="number" id="zIndexValue" min="0" max="10" value="1">
26     <button onclick="changeZIndex()">Apply</button>
27   </div>
28
29   <div id="status"></div>
30
31   <script src="script.js"></script>
32 </body>
33 </html>

```

- Css file

```

# style.css > ...
1  body {
2      font-family: Arial, sans-serif;
3      text-align: center;
4  }
5
6  .container {
7      position: relative;
8      width: 300px;
9      height: 300px;
10     margin: 50px auto;
11 }
12
13 .box {
14     width: 150px;
15     height: 150px;
16     position: absolute;
17     text-align: center;
18     line-height: 150px;
19     font-size: 20px;
20     font-weight: bold;
21     color: white;
22     border: 2px solid black;
23 }
24
25 .red { background: red; top: 40px; left: 40px; z-index: 1; }
26 .blue { background: blue; top: 80px; left: 80px; z-index: 2; }
27 .green { background: green; top: 120px; left: 120px; z-index: 3; }
28
29 .controls {
30     margin-top: 20px;
31 }
32
33 button {
34     padding: 8px 15px;
35     margin-left: 10px;
36     cursor: pointer;
37 }
38
39 #status {
40     margin-top: 20px;
41     font-weight: bold;
42 }

```

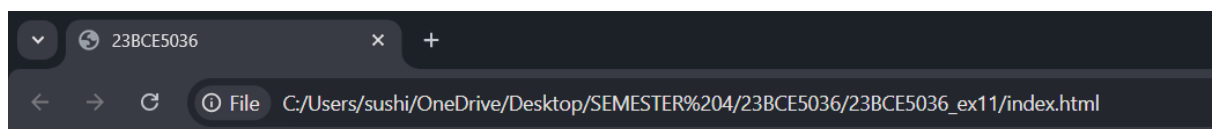
- javascript file

```

# script.js
1  updateStatus
2  function changeZIndex() {
3      let selectedElement = document.getElementById(document.getElementById("elementSelect").value);
4      let newZIndex = document.getElementById("zIndexValue").value;
5
6      selectedElement.style.zIndex = newZIndex;
7      updateStatus();
8  }
9
10 function updateStatus() {
11     let box1Z = document.getElementById("box1").style.zIndex;
12     let box2Z = document.getElementById("box2").style.zIndex;
13     let box3Z = document.getElementById("box3").style.zIndex;
14
15     document.getElementById("status").innerHTML = `Box 1 Z-Index: ${box1Z} | Box 2 Z-Index: ${box2Z} | Box 3 Z-Index: ${box3Z}`;
16 }
17
18 window.onload = updateStatus;

```

Output:



Box 1

Box 2

Box 3

Select Element: Box 3 ▼ Set Z-Index: Apply

Box 1 Z-Index: 10 | Box 2 Z-Index: 8 | Box 3 Z-Index: 1

