JavaScript Assignment - 01

Name – Soumitra Anil Kode
Roll no. – 42134 , BE- 06, Batch - P6

DOP :

Source Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8" />
 <title>Area of Shapes</title>
 <meta name="viewport" content="width=device-width, initial-scale=1" />
 <style>
  body {
   background: #f7f8fa;
   font-family: Arial, sans-serif;
   margin: 0;
   padding: 0;
  }
  .container {
   max-width: 440px;
   margin: 44px auto 20px auto;
   background: #fff;
   border-radius: 12px;
   box-shadow: 0 2px 16px #7f7f7f13;
   padding: 32px 30px 24px 30px;
   text-align: center;
  }
  h1 {
   margin-top: 18px;
   margin-bottom: 3px;
   font-size: 1.40em;
   font-weight: bold;
   color: #444e57;
   letter-spacing: 1px;
  }
  h3 {
   font-size: 1.09em;
   margin-bottom: 9px;
   color: #5c6b7a;
   letter-spacing: 1px;
  }
  .title {
   margin-bottom: 20px;
  }
```

```
.shape-section {
  margin-bottom: 27px;
  background: #f6f8fb;
  border-radius: 8px;
  padding: 14px 12px 18px 12px;
  box-shadow: 0 1.5px 8px #abbed333;
}
.shape-section label {
  font-weight: bold;
  display: block;
  margin: 10px 0 3px 0;
  text-align: left;
  font-size: 1em;
  color: #31415e;
}
.input-row {
  margin-bottom: 4px;
  margin-top: 2px;
}
.shape-section input[type="number"] {
  width: 100%;
  padding: 8px;
  border-radius: 6px;
  border: 1.6px solid #bdc5cc;
  font-size: 1em;
  background: #eef4fb;
  margin-bottom: 6px;
  transition: border-color 0.22s;
}
.shape-section input[type="number"]:focus {
  border-color: #5aa8fc;
  outline: none;
  background: #e3f1ff;
}
.area-output {
  margin: 6px 0 3px 0;
  font-size: 1em;
  color: #1e7a67;
  font-weight: 600;
  letter-spacing: 0.08em;
}
.error-msg {
  color: #d13a3a;
  font-weight: 600;
  margin-bottom: 4px;
  margin-top: 3px;
  font-size: 0.95em;
```

```
    }
    .btn-row {
      display: flex;
      justify-content: center;
      gap: 12px;
      margin-top: 8px;
    }
    button {
      padding: 6px 26px;
      background: #48546a;
      color: #fff;
      border: none;
      border-radius: 5px;
      font-weight: 600;
      font-size: 1em;
      cursor: pointer;
      transition: background 0.19s;
    }
    button:hover {
      background: #222c3b;
    }
    /* Responsive design for small screens */
    @media (max-width: 480px) {
      .container { max-width: 97vw; padding: 5vw 2vw; }
      button { padding: 6px 16px; }
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="title">
      <h1>Soumitra Kode</h1>
      <h3>42134 | P6 |  BE-06</h3>
      <h2 style="margin-bottom:12px; margin-top:8px;">Area of Shapes</h2>
    </div>
    <!-- Triangle -->
    <div class="shape-section" id="triangleSection">
      <div><b>Triangle</b></div>
      <div class="input-row">
        <label for="triangleA">Side A:</label>
        <input type="number" id="triangleA" min="1" step="any" />
      </div>
      <div class="input-row">
        <label for="triangleB">Side B:</label>
        <input type="number" id="triangleB" min="1" step="any" />
      </div>
      <div class="input-row">
```

```
    <label for="triangleC">Side C:</label>
    <input type="number" id="triangleC" min="1" step="any" />
   </div>
   <div class="area-output" id="triangleOutput">Area of Triangle: </div>
   <div class="error-msg" id="triangleError"></div>
   <div class="btn-row">
    <button onclick="calcTriangleArea()">Submit</button>
    <button onclick="resetTriangle()">Reset</button>
   </div>
  </div>
  <!-- Circle -->
  <div class="shape-section" id="circleSection">
   <div><b>Circle</b></div>
   <div class="input-row">
    <label for="circleRadius">Radius of Circle:</label>
    <input type="number" id="circleRadius" min="0" step="any" />
   </div>
   <div class="area-output" id="circleOutput">Area of Circle: </div>
   <div class="error-msg" id="circleError"></div>
   <div class="btn-row">
    <button onclick="calcCircleArea()">Submit</button>
    <button onclick="resetCircle()">Reset</button>
   </div>
  </div>
  <!-- Square -->
  <div class="shape-section" id="squareSection">
   <div><b>Square</b></div>
   <div class="input-row">
    <label for="squareLength">Length of Square:</label>
    <input type="number" id="squareLength" step="any" />
   </div>
   <div class="area-output" id="squareOutput">Area of Square: </div>
   <div class="error-msg" id="squareError"></div>
   <div class="btn-row">
    <button onclick="calcSquareArea()">Submit</button>
    <button onclick="resetSquare()">Reset</button>
   </div>
  </div>
  <!-- Rectangle -->
  <div class="shape-section" id="rectangleSection">
   <div><b>Rectangle</b></div>
   <div class="input-row">
    <label for="rectangleLength">Length of Rectangle:</label>
    <input type="number" id="rectangleLength" step="any" />
   </div>
   <div class="input-row">
    <label for="rectangleBreadth">Breadth of Rectangle:</label>
```

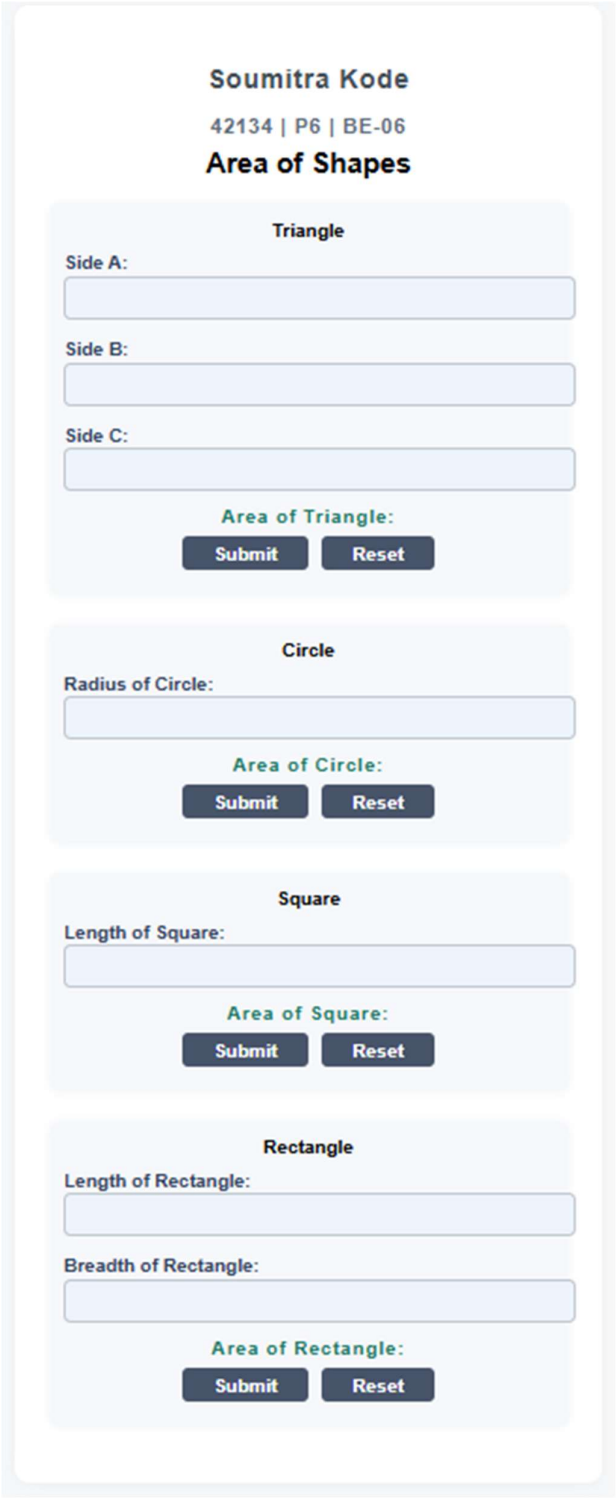
```
      <input type="number" id="rectangleBreadth" step="any" />
     </div>
     <div class="area-output" id="rectangleOutput">Area of Rectangle: </div>
     <div class="error-msg" id="rectangleError"></div>
     <div class="btn-row">
      <button onclick="calcRectangleArea()">Submit</button>
      <button onclick="resetRectangle()">Reset</button>
     </div>
    </div>
   </div>
<script>
  function calcTriangleArea() {
    const a = parseFloat(document.getElementById("triangleA").value);
    const b = parseFloat(document.getElementById("triangleB").value);
    const c = parseFloat(document.getElementById("triangleC").value);
    const output = document.getElementById("triangleOutput");
    const error = document.getElementById("triangleError");
    output.textContent = "Area of Triangle: ";
    error.textContent = "";

    if (isNaN(a) || isNaN(b) || isNaN(c) || a <= 0 || b <= 0 || c <= 0) {
      error.textContent = "Error! Please check your values!";
      return;
    }
    if ((a + b <= c) || (a + c <= b) || (b + c <= a)) {
      error.textContent = "Error! Sides do not form a triangle.";
      return;
    }
    const s = (a + b + c) / 2;
    const area = Math.sqrt(s * (s - a) * (s - b) * (s - c));
    output.textContent = "Area of Triangle: " + area.toFixed(2) + " sq units";
  }

  function resetTriangle() {
    document.getElementById("triangleA").value = "";
    document.getElementById("triangleB").value = "";
    document.getElementById("triangleC").value = "";
    document.getElementById("triangleOutput").textContent = "Area of
Triangle: ";
    document.getElementById("triangleError").textContent = "";
  }

  function calcCircleArea() {
    const radius = parseFloat(document.getElementById("circleRadius").value);
    const output = document.getElementById("circleOutput");
    const error = document.getElementById("circleError");
    output.textContent = "Area of Circle: ";
```

```
  error.textContent = "";
  if (isNaN(radius) || radius < 0) {
   error.textContent = "Error! Please check your values!";
   return;
  }
  const area = Math.PI * radius * radius;
  output.textContent = "Area of Circle: " + area.toFixed(2) + " sq units";
 }

 function resetCircle() {
  document.getElementById("circleRadius").value = "";
  document.getElementById("circleOutput").textContent = "Area of Circle: ";
  document.getElementById("circleError").textContent = "";
 }

 function calcSquareArea() {
  const len = parseFloat(document.getElementById("squareLength").value);
  const output = document.getElementById("squareOutput");
  const error = document.getElementById("squareError");
  output.textContent = "Area of Square: ";
  error.textContent = "";
  if (isNaN(len) || len <= 0) {
   error.textContent = "Error! Please check your values!";
   return;
  }
  const area = len * len;
  output.textContent = "Area of Square: " + area.toFixed(2) + " sq units";
 }

 function resetSquare() {
  document.getElementById("squareLength").value = "";
  document.getElementById("squareOutput").textContent = "Area of Square:
";
  document.getElementById("squareError").textContent = "";
 }

 function calcRectangleArea() {
  const l = parseFloat(document.getElementById("rectangleLength").value);
  const b = parseFloat(document.getElementById("rectangleBreadth").value);
  const output = document.getElementById("rectangleOutput");
  const error = document.getElementById("rectangleError");
  output.textContent = "Area of Rectangle: ";
  error.textContent = "";
  if (isNaN(l) || isNaN(b) || l <= 0 || b <= 0) {
   error.textContent = "Error! Please check your values!";
   return;
  }
```

```
  const area = l * b;
  output.textContent = "Area of Rectangle: " + area.toFixed(2) + " sq units";
 }

 function resetRectangle() {
  document.getElementById("rectangleLength").value = "";
  document.getElementById("rectangleBreadth").value = "";
  document.getElementById("rectangleOutput").textContent = "Area of
Rectangle: ";
  document.getElementById("rectangleError").textContent = "";
 }
</script>
</body>
</html>
```

Output –

A. Initial Landing Page –



B. Desired Output –