WEB UI PRACTICAL

Experiment: Interactive SVG Drawing Tool with Mouse Event Handlers

AIM

Design and build a web-based drawing tool using SVG where users can draw by clicking and dragging the mouse.

COD

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SVG Drawing Tool</title>
  <style>
    body { font-family: Arial, sans-serif; padding: 20px; }
    .box { border: 2px solid #333; padding: 12px; }
    h1 { margin: 0 0 12px; }
    svg { background: #fafafa; border: 2px solid #555; display: block; margin: 12px; }
    .hint { color: #666; font-size: 14px; margin-left: 12px; }
  </style>
</head>
<body>
  <div class="box">
    <h1>SVG Drawing Tool</h1>
    <span class="hint">Drag inside the canvas to draw. Release to finish a stroke.</span>
    <svg id="canvas" width="800" height="450"></svg>
  </div>

  <script>
    const svgNS = "http://www.w3.org/2000/svg";
    const canvas = document.getElementById("canvas");
    let drawing = false;
    let pathEl = null;
    let d = "";

    function getPoint(evt) {
      const rect = canvas.getBoundingClientRect();
      const x = evt.clientX - rect.left;
      const y = evt.clientY - rect.top;
      return { x, y };
    }

    canvas.addEventListener("mousedown", (e) => {
      drawing = true;
      const { x, y } = getPoint(e);
      d = `M ${x} ${y}`;
      pathEl = document.createElementNS(svgNS, "path");
      pathEl.setAttribute("fill", "none");
      pathEl.setAttribute("stroke", "#1e88e5");
      pathEl.setAttribute("stroke-width", "3");
      pathEl.setAttribute("d", d);
      canvas.appendChild(pathEl);
    });

    canvas.addEventListener("mousemove", (e) => {
      if (!drawing) return;
      const { x, y } = getPoint(e);
      d += ` L ${x} ${y}`;
      pathEl.setAttribute("d", d);
    });

    window.addEventListener("mouseup", () => {
      drawing = false;
      d = "";
      pathEl = null;
    });
  </script>
</body>
</html>
```

OUTPUT