

## EXP-6

### Interactive SVG Drawing Tool with Mouse Event Handlers:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Interactive SVG Drawing Tool</title>

  <!-- Tailwind CSS for styling -->

  <script src="https://cdn.tailwindcss.com"></script>

  <!-- Google Fonts for a clean, modern look -->

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600&display=swap"
rel="stylesheet">

  <style>

    /* Use the Inter font family */

    body {

      font-family: 'Inter', sans-serif;

    }

    /* Style for the drawing canvas */

    #drawing-canvas {

      border: 2px solid #4A5568; /* gray-700 */

      cursor: crosshair;

      touch-action: none; /* Prevents scrolling on touch devices */

    }

  </style>

</head>

<body class="bg-gray-900 text-gray-200 flex items-center justify-center min-h-screen p-4">

  <!-- Main container card -->
```

```

<div class="w-full max-w-3xl p-8 bg-gray-800 rounded-2xl shadow-lg">

  <!-- Header Section -->

  <div class="mb-6 text-center">

    <h1 class="text-3xl font-bold text-gray-100">SVG Drawing Tool</h1>

    <p class="text-gray-400 mt-2">Click and drag on the canvas below to draw.</p>

  </div>


  <!-- SVG Drawing Area -->

  <svg id="drawing-canvas" class="w-full h-96 bg-gray-100 rounded-lg"></svg>


</div>


<script>

  // --- DOM ELEMENTS ---

  const svgCanvas = document.getElementById('drawing-canvas');


  // --- STATE VARIABLES ---

  let isDrawing = false;

  let currentPath = null;

  let pathData = '';


  // --- FUNCTIONS ---


  /**
   * Gets the mouse/touch coordinates relative to the SVG canvas.
   * @param {MouseEvent|TouchEvent} event - The mouse or touch event.
   * @returns {Object} An object with x and y coordinates.
   */

  function getCoordinates(event) {

    const rect = svgCanvas.getBoundingClientRect();

```

```

// Handle both mouse and touch events

const clientX = event.clientX || event.touches[0].clientX;
const clientY = event.clientY || event.touches[0].clientY;

return {
  x: clientX - rect.left,
  y: clientY - rect.top
};
}

/**
 * Handles the start of a drawing action (mousedown or touchstart).
 * @param {MouseEvent|TouchEvent} event - The event object.
 */
function startDrawing(event) {
  event.preventDefault();

  isDrawing = true;

  const { x, y } = getCoordinates(event);

  // Start a new path
  pathData = `M ${x} ${y}`;

  currentPath = document.createElementNS('http://www.w3.org/2000/svg', 'path');
  currentPath.setAttribute('stroke', '#3B82F6'); // A nice blue color
  currentPath.setAttribute('stroke-width', '3');
  currentPath.setAttribute('fill', 'none');
  currentPath.setAttribute('stroke-linejoin', 'round');
  currentPath.setAttribute('stroke-linecap', 'round');
  currentPath.setAttribute('d', pathData);

  svgCanvas.appendChild(currentPath);
}

```

```

/**
 * Handles the drawing action as the mouse or finger moves.
 * @param {MouseEvent|TouchEvent} event - The event object.
 */
function draw(event) {
    if (!isDrawing) return;
    event.preventDefault();
    const { x, y } = getCoordinates(event);

    // Append a line segment to the current path
    pathData += ` L ${x} ${y}`;

    if (currentPath) {
        currentPath.setAttribute('d', pathData);
    }
}

/**
 * Handles the end of a drawing action (mouseup, mouseleave, touchend).
 */
function stopDrawing() {
    isDrawing = false;
    currentPath = null;
    pathData = "";
}

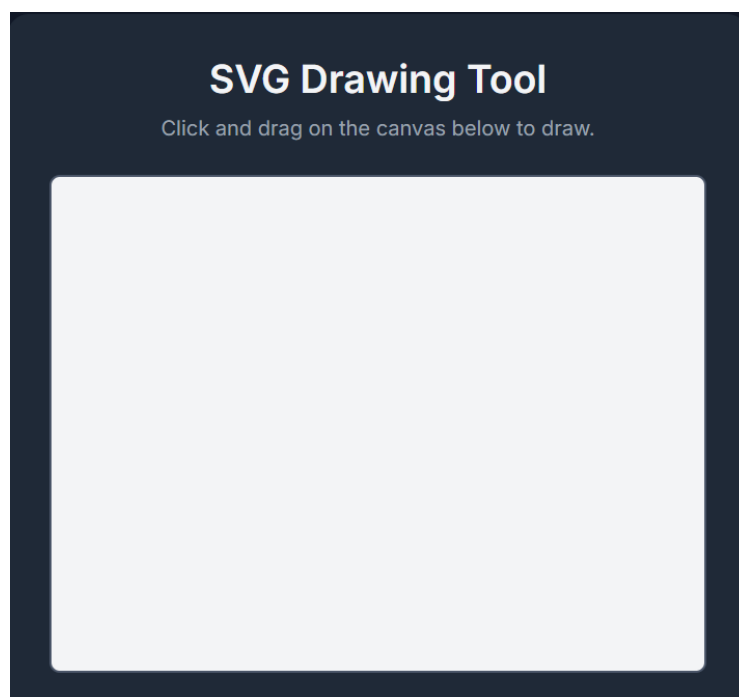
// --- EVENT LISTENERS ---

// Mouse events
svgCanvas.addEventListener('mousedown', startDrawing);

```

```
svgCanvas.addEventListener('mousemove', draw);  
svgCanvas.addEventListener('mouseup', stopDrawing);  
svgCanvas.addEventListener('mouseleave', stopDrawing); // Stop if mouse leaves canvas  
  
// Touch events for mobile support  
svgCanvas.addEventListener('touchstart', startDrawing);  
svgCanvas.addEventListener('touchmove', draw);  
svgCanvas.addEventListener('touchend', stopDrawing);  
svgCanvas.addEventListener('touchcancel', stopDrawing);  
  
</script>  
</body>  
</html>
```

## OUTPUTS:



*Figure 1 Empty Canvas*

## SVG Drawing Tool

Click and drag on the canvas below to draw.



*Figure 2 Drawing on Canvas*