**Interactive Painting or drawing application**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Interactive Drawing App</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="toolbar">

<input type="color" id="colorPicker" value="#000000">

<button id="pencilButton">Pencil</button>

<button id="eraserButton">Eraser</button>

<button id="clearButton">Clear</button>

<label for="brushSize">Brush Size:</label>

<input type="range" id="brushSize" min="1" max="20" value="5">

</div>

<canvas id="drawingCanvas" width="800" height="600"></canvas>

<script src="app.js"></script>

</body>

</html>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

.toolbar {

padding: 10px;

background-color: #f5f5f5;

text-align: center;

}

canvas {

border: 1px solid #ccc;

display: block;

margin: 20px auto;

background-color: white;

}

const canvas = document.getElementById("drawingCanvas");

const ctx = canvas.getContext("2d");

const colorPicker = document.getElementById("colorPicker");

const pencilButton = document.getElementById("pencilButton");

const eraserButton = document.getElementById("eraserButton");

const clearButton = document.getElementById("clearButton");

const brushSize = document.getElementById("brushSize");

let isDrawing = false;

let tool = "pencil"; // default tool

let currentColor = "#000000";

let currentSize = 5;

// Handle pencil and eraser buttons

pencilButton.addEventListener("click", () => {

tool = "pencil";

});

eraserButton.addEventListener("click", () => {

tool = "eraser";

});

// Handle color picker

colorPicker.addEventListener("change", (e) => {

currentColor = e.target.value;

});

// Handle brush size

brushSize.addEventListener("change", (e) => {

currentSize = e.target.value;

});

// Clear canvas

clearButton.addEventListener("click", () => {

ctx.clearRect(0, 0, canvas.width, canvas.height);

});

// Start drawing when the mouse is pressed

canvas.addEventListener("mousedown", (e) => {

isDrawing = true;

ctx.beginPath();

ctx.moveTo(e.offsetX, e.offsetY);

});

// Stop drawing when the mouse is released

canvas.addEventListener("mouseup", () => {

isDrawing = false;

ctx.closePath();

});

// Draw on the canvas when the mouse moves

canvas.addEventListener("mousemove", (e) => {

if (isDrawing) {

ctx.lineCap = "round";

ctx.lineWidth = currentSize;

if (tool === "pencil") {

ctx.strokeStyle = currentColor;

} else if (tool === "eraser") {

ctx.strokeStyle = "#ffffff"; // erasing with white

}

ctx.lineTo(e.offsetX, e.offsetY);

ctx.stroke();

}

});