const canvas = document.getElementById('imageCanvas');

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

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

const exportButton = document.getElementById('exportButton');

canvas.width = 800;

canvas.height = 600;

// Mock image (replace with a dynamic image loader)

const img = new Image();

img.src = 'https://via.placeholder.com/800x600.png'; // Placeholder image

img.onload = () => ctx.drawImage(img, 0, 0, canvas.width, canvas.height);

let annotations = [];

let isDrawing = false;

let startX, startY;

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

isDrawing = true;

startX = e.offsetX;

startY = e.offsetY;

});

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

if (isDrawing) {

ctx.drawImage(img, 0, 0, canvas.width, canvas.height); // Reset canvas

drawRectangle(startX, startY, e.offsetX - startX, e.offsetY - startY);

}

});

canvas.addEventListener('mouseup', (e) => {

if (isDrawing) {

isDrawing = false;

annotations.push({

x: startX,

y: startY,

width: e.offsetX - startX,

height: e.offsetY - startY,

color: colorPicker.value,

});

}

});

function drawRectangle(x, y, width, height) {

ctx.strokeStyle = colorPicker.value;

ctx.lineWidth = 2;

ctx.strokeRect(x, y, width, height);

}

exportButton.addEventListener('click', () => {

fetch('/saveAnnotations', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify(annotations),

})

.then(response => response.json())

.then(data => alert('Annotations saved successfully!'))

.catch(err => console.error(err));

});