// Global background movement effect

document.body.addEventListener('mousemove', (e) => {

const x = (e.clientX / window.innerWidth) \* 100;

const y = (e.clientY / window.innerHeight) \* 100;

document.body.style.backgroundPosition = `${x}% ${y}%`;

});

// Section fade-in on scroll

document.querySelectorAll('section').forEach(sec => {

sec.style.opacity = 0;

sec.style.transform = 'translateY(40px) rotateX(10deg)';

window.addEventListener('scroll', () => {

const rect = sec.getBoundingClientRect();

if (rect.top < window.innerHeight - 100) {

sec.style.transition = 'all 0.8s ease-out';

sec.style.opacity = 1;

sec.style.transform = 'translateY(0) rotateX(0deg)';

}

});

});

window.dispatchEvent(new Event('scroll'));

// Particle effect for moving dots

const canvas = document.getElementById('particle-canvas');

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

// Set canvas size to match window

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

// Resize canvas on window resize

window.addEventListener('resize', () => {

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

});

// Particle class to manage each dot

class Particle {

constructor() {

this.x = Math.random() \* canvas.width;

this.y = Math.random() \* canvas.height;

this.size = 1; // Fixed size for smaller, uniform dots

this.speedX = Math.random() \* 2 - 1; // Random speed between -1 and 1

this.speedY = Math.random() \* 2 - 1;

}

update() {

this.x += this.speedX;

this.y += this.speedY;

// Bounce off edges

if (this.x < 0 || this.x > canvas.width) this.speedX \*= -1;

if (this.y < 0 || this.y > canvas.height) this.speedY \*= -1;

}

draw() {

ctx.fillStyle = 'rgba(255, 255, 255, 0.8)'; // White dots with slight transparency

ctx.beginPath();

ctx.arc(this.x, this.y, this.size, 0, Math.PI \* 2);

ctx.fill();

}

}

// Create an array of particles

const particles = [];

const particleCount = 100; // Adjust for more or fewer dots

for (let i = 0; i < particleCount; i++) {

particles.push(new Particle());

}

// Animation loop

function animate() {

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

particles.forEach(particle => {

particle.update();

particle.draw();

});

requestAnimationFrame(animate);

}

animate();