<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Gesture Heart Effect with Love Letter and Poem</title>

<style>

html, body {

height: 100%;

margin: 0;

background: #000;

overflow: hidden;

font-family: 'Segoe UI', sans-serif;

color: #fff;

}

/\* LOGIN SCREEN \*/

#loginScreen {

position: absolute;

width: 100%;

height: 100%;

background: radial-gradient(circle, #ffccda, #ffb6c1);

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

z-index: 10;

}

#loginScreen h1 {

font-size: 2em;

margin-bottom: 20px;

color: #b0004d;

}

#yesBtn {

padding: 10px 20px;

font-size: 1em;

border: none;

background-color: #ff4d88;

color: white;

border-radius: 10px;

cursor: pointer;

transition: background 0.3s ease;

}

#yesBtn:hover {

background-color: #e6005c;

}

/\* LOVE LETTER PAGE \*/

#letterPage {

position: absolute;

width: 100%;

height: 100%;

background: #ffc0cb; /\* pink \*/

display: none;

flex-direction: column;

align-items: center;

justify-content: center;

padding: 40px 20px;

box-sizing: border-box;

overflow-y: auto;

z-index: 9;

}

/\* Small heading "Love Letter" \*/

#letterPage h2 {

margin: 0 0 20px 0;

font-size: 1.2em;

color: #b22222;

font-weight: bold;

}

/\* Letter box \*/

#letterBox {

background: #b22222; /\* red \*/

padding: 20px;

border: 4px solid transparent; /\* outer border space \*/

box-sizing: border-box;

max-width: 600px;

border-image-source: linear-gradient(to bottom right, #ff4d88, #800000);

border-image-slice: 1;

border-style: solid;

border-width: 4px;

color: #fff;

font-size: 1.3em;

line-height: 1.4em;

position: relative;

z-index: 10;

}

/\* Inner border aligned inside the box \*/

#letterBox::before {

content: "";

position: absolute;

top: 8px;

left: 8px;

right: 8px;

bottom: 8px;

border: 2px solid #ffb6c1;

pointer-events: none;

}

/\* Hanging hearts behind letter page \*/

#heartsContainer {

position: absolute;

width: 100%;

height: 100%;

pointer-events: none;

overflow: visible;

top: 0;

left: 0;

z-index: 1;

}

.heart {

position: absolute;

width: 40px;

height: 40px;

background: #ff4d88;

transform: rotate(-45deg);

animation: floatUp 6s linear infinite;

}

.heart::before,

.heart::after {

content: "";

position: absolute;

width: 40px;

height: 40px;

background: #ff4d88;

border-radius: 50%;

}

.heart::before {

top: -20px;

left: 0;

}

.heart::after {

left: 20px;

top: 0;

}

@keyframes floatUp {

0% {

opacity: 1;

transform: translateY(0) rotate(-45deg);

}

100% {

opacity: 0;

transform: translateY(-300px) rotate(-45deg);

}

}

/\* Smaller >>> button centered bottom \*/

#nextLetterBtn {

font-size: 1.2em;

color: #b22222;

background: #ffd6dd;

border-radius: 8px;

padding: 6px 20px;

border: 2px solid #800000;

position: absolute;

bottom: 20px;

left: 50%;

transform: translateX(-50%);

cursor: pointer;

user-select: none;

transition: background 0.3s ease, color 0.3s ease;

text-align: center;

box-shadow: 0 0 6px #b22222;

z-index: 10;

}

#nextLetterBtn:hover {

background: #b22222;

color: #fff;

box-shadow: 0 0 12px #ff4d88;

}

/\* POEM PAGE \*/

#poem {

position: absolute;

top: 10%;

width: 100%;

text-align: center;

font-size: 1.5em;

color: #ffe6f0;

line-height: 1.5em;

white-space: pre-line;

z-index: 9;

opacity: 0;

transition: opacity 1s ease-in-out, transform 2s ease;

pointer-events: none;

transform: translateY(50px);

}

#poem.shimmering {

animation: shimmer 2s infinite;

pointer-events: auto;

opacity: 1 !important;

transform: translateY(0) !important;

}

/\* Shimmer effect \*/

@keyframes shimmer {

0% {

text-shadow: 0 0 5px #ffccda, 0 0 10px #ffccda, 0 0 15px #ff4d88;

}

50% {

text-shadow: 0 0 20px #ffccda, 0 0 30px #ff4d88, 0 0 40px #ffb6c1;

}

100% {

text-shadow: 0 0 5px #ffccda, 0 0 10px #ff4d88, 0 0 15px #ffb6c1;

}

}

/\* Name at bottom \*/

#name {

position: absolute;

bottom: 20px;

width: 100%;

text-align: center;

font-size: 2em;

color: #ffe6f0;

font-family: 'Brush Script MT', cursive;

letter-spacing: 2px;

z-index: 10;

opacity: 0;

transform: translateY(50px);

transition: opacity 1s ease-in-out, transform 2s ease;

}

#name.shimmering {

animation: shimmer 2s infinite;

opacity: 1 !important;

transform: translateY(0) !important;

}

/\* Canvas for hearts \*/

canvas {

position: absolute;

width: 100%;

height: 100%;

display: block;

z-index: 0;

}

</style>

</head>

<body>

<!-- Login Screen -->

<div id="loginScreen">

<h1>Do you love me?</h1>

<button id="yesBtn">Yes ❤️</button>

</div>

<!-- Letter Page -->

<div id="letterPage">

<h2>Love Letter</h2>

<div id="letterBox">

To You!<br><br>

You never know my feelings for you. I like you, heaven knows, earth knows, everyone knows except you. Have you ever thought you are very silly? When you are happy, you talk happily with me, when you are sad, I don't dare to say a word. Being friends with you is the thing I regret the most in my life. When will I have enough courage to break this shell called friendship and express my feelings to you?

</div>

<div id="nextLetterBtn">&gt;&gt;&gt;</div>

<div id="heartsContainer"></div>

</div>

<!-- Poem -->

<div id="poem">

My love for you runs dark and deep,<br>

It wakes me still, it sings in sleep.<br>

No time, no fate could pull apart,<br>

The thread you’ve woven through my heart.

</div>

<!-- Heart Animation Canvas -->

<canvas id="pinkboard"></canvas>

<!-- Name -->

<div id="name">Here's a small peom for u💞</div>

<script>

// Show letter page on Yes click

document.getElementById('yesBtn').addEventListener('click', () => {

document.getElementById('loginScreen').style.display = 'none';

document.getElementById('letterPage').style.display = 'flex';

});

// Show poem page on >>> click

document.getElementById('nextLetterBtn').addEventListener('click', () => {

document.getElementById('letterPage').style.display = 'none';

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

poem.style.opacity = 1;

poem.style.transform = 'translateY(0)';

poem.classList.add('shimmering');

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

name.style.opacity = 1;

name.style.transform = 'translateY(0)';

name.classList.add('shimmering');

poem.style.pointerEvents = 'auto';

});

// Create floating hearts on letter page behind letter box

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

const heartCount = 12;

function randomRange(min, max) {

return Math.random() \* (max - min) + min;

}

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

const heart = document.createElement('div');

heart.classList.add('heart');

heart.style.left = `${randomRange(5, 95)}%`;

heart.style.top = `${randomRange(80, 100)}%`;

heart.style.animationDuration = `${randomRange(5, 10)}s`;

heart.style.animationDelay = `${randomRange(0, 5)}s`;

heartsContainer.appendChild(heart);

}

// --- Heart animation code from original (fixed) ---

var settings = {

particles: {

length: 500,

duration: 2,

velocity: 100,

effect: -0.75,

size: 30

}

};

function Point(x, y) {

this.x = x || 0;

this.y = y || 0;

}

var Particle = function() {

this.position = new Point();

this.velocity = new Point();

this.acceleration = new Point();

this.age = 0;

};

Particle.prototype.initialize = function(x, y, dx, dy) {

this.position.x = x;

this.position.y = y;

this.velocity.x = dx;

this.velocity.y = dy;

this.acceleration.x = dx \* settings.particles.effect;

this.acceleration.y = dy \* settings.particles.effect;

this.age = 0;

};

Particle.prototype.update = function(deltaTime) {

this.velocity.x += this.acceleration.x \* deltaTime;

this.velocity.y += this.acceleration.y \* deltaTime;

this.position.x += this.velocity.x \* deltaTime;

this.position.y += this.velocity.y \* deltaTime;

this.age += deltaTime;

};

Particle.prototype.draw = function(context, image) {

function ease(t) {

return (--t) \* t \* t + 1;

}

var size = image.width \* ease(this.age / settings.particles.duration);

context.globalAlpha = 1 - this.age / settings.particles.duration;

context.drawImage(image, this.position.x - size / 2, this.position.y - size / 2, size, size);

context.globalAlpha = 1;

};

var ParticlePool = function(length) {

this.length = length;

this.particles = [];

this.firstActive = 0;

this.firstFree = 0;

for (var i = 0; i < length; i++)

this.particles.push(new Particle());

};

ParticlePool.prototype.add = function(x, y, dx, dy) {

var particle = this.particles[this.firstFree];

particle.initialize(x, y, dx, dy);

this.firstFree++;

if (this.firstFree === this.length)

this.firstFree = 0;

if (this.firstActive === this.firstFree)

this.firstActive++;

if (this.firstActive === this.length)

this.firstActive = 0;

};

ParticlePool.prototype.update = function(deltaTime) {

var i = this.firstActive;

while (i !== this.firstFree) {

var particle = this.particles[i];

if (particle.age >= settings.particles.duration) {

this.firstActive++;

if (this.firstActive === this.length)

this.firstActive = 0;

} else

particle.update(deltaTime);

i++;

if (i === this.length)

i = 0;

}

};

ParticlePool.prototype.draw = function(context, image) {

var i = this.firstActive;

while (i !== this.firstFree) {

this.particles[i].draw(context, image);

i++;

if (i === this.length)

i = 0;

}

};

var canvas = document.getElementById('pinkboard');

var context = canvas.getContext('2d');

var particles = new ParticlePool(settings.particles.length);

var time;

var position = new Point();

const heartImg = new Image();

// Base64 small pink heart icon

heartImg.src = 'data:image/svg+xml;base64,';

function resize() {

canvas.width = window.innerWidth;

canvas.height = window.innerHeight;

}

window.addEventListener('resize', resize);

resize();

function addParticle() {

// Emit hearts near bottom center on letter or poem page

var direction = new Point(

(Math.random() - 0.5) \* settings.particles.velocity,

(Math.random() - 0.5) \* settings.particles.velocity

);

particles.add(position.x, position.y, direction.x, direction.y);

}

function loop(timestamp) {

if (!time) time = timestamp;

var deltaTime = (timestamp - time) / 1000;

time = timestamp;

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

// Only emit if letter or poem page is visible

const letterVisible = document.getElementById('letterPage').style.display === 'flex';

const poemVisible = document.getElementById('poem').style.opacity === '1';

if (letterVisible || poemVisible) {

position.x = window.innerWidth / 2;

position.y = window.innerHeight \* 0.75;

addParticle();

}

particles.update(deltaTime);

particles.draw(context, heartImg);

requestAnimationFrame(loop);

}

// Start animation loop only after image loaded

heartImg.onload = () => {

requestAnimationFrame(loop);

};

</script>

</body>

</html>