

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Jeel Vaghasiya – Live Interactive Background</title>
  <style>
    /* Reset */
    *{box-sizing:border-box;margin:0;padding:0}
    html,body,#app{height:100%}
    body{
      font-family:Inter, system-ui, -apple-system, 'Segoe UI', Roboto,
'Helvetica Neue', Arial;
      overflow:hidden;
      background: linear-gradient(120deg,#081229 0%, #0b1e3a 50%, #08304f 100%);
      color: #fff;
    }

    /* Canvas covers full screen */
    canvas#bg-canvas{
      position:fixed;inset:0;width:100%;height:100%;display:block;z-index:0;
    }

    /* Main center container */
    .center {
      position:relative;z-index:2;height:100%;display:flex;align-
items:center;justify-content:center;flex-direction:column;padding:2rem;text-
align:center;
    }

    /* Name styling */
    .name {
      font-size: clamp(36px, 6vw, 96px);
      letter-spacing: 0.06em;
      font-weight:800;
      line-height:1;
      padding:0.25em 0.5em;
      backdrop-filter: blur(6px) saturate(1.1);
      -webkit-text-stroke: 1px rgba(255,255,255,0.06);
      text-shadow:
        0 6px 30px rgba(2,10,25,0.6),
        0 2px 8px rgba(0,150,255,0.06);
      display:inline-block;
      background: linear-gradient(90deg, rgba(255,255,255,0.95),
rgba(255,255,255,0.85));
      color: #022236;
      border-radius: 14px;
      transform: translateZ(0);
    }

    /* Subtitle */
    .sub {margin-top:0.65rem;font-size:clamp(12px,1.5vw,18px);opacity:0.9}

    /* Floating card with small controls */
    .controls{position:fixed;right:18px;top:18px;background:rgba(255,255,255,0.0
4);backdrop-filter:blur(6px);border-radius:10px;padding:10px 12px;font-
size:14px;z-index:3}
    .controls label{display:flex;align-items:center;gap:8px}
    .toggle{appearance:none;width:40px;height:22px;border-
radius:12px;background:linear-
gradient(90deg,#2b2b2b,#1b1b1b);position:relative;cursor:pointer}
    .toggle:after{content:"";position:absolute;top:2px;left:2px;width:18px;heigh
t:18px;border-radius:50%;background:#fff;transition:all .18s}
    .toggle[data-on="true"]{background:linear-gradient(90deg,#06f,#08f)}

```

```

.toggle[data-on="true"]:after{left:20px}

/* small instruction */
.hint{position:fixed;left:18px;bottom:18px;background:rgba(0,0,0,0.25);padding:8px 10px;border-radius:8px;font-size:13px;z-index:3}

/* butterflies (CSS-only decorative) */
.butterflies{position:fixed;inset:0;pointer-events:none;z-index:1}
.butterfly{
  position:absolute;width:36px;height:36px;filter:drop-shadow(0 6px 10px
  rgba(0,0,0,0.4));opacity:0.9;transform-origin:center;
  will-change:transform;mix-blend-mode:screen
}
.butterfly svg{width:100%;height:100%}

@media (max-width:600px){
  .controls{font-size:12px;padding:8px}
  .name{border-radius:10px}
}
</style>
</head>
<body>
  <canvas id="bg-canvas" aria-hidden="true"></canvas>

  <div id="app" class="center">
    <div class="name" id="mainName">JEEL VAGHASIYA</div>
  </div>

  <div class="controls" aria-hidden="true">
    <label title="Toggle particle motion"><span>Particles</span>
    <div id="particlesToggle" class="toggle" data-on="true"></div>
  </label>
</div>

  <div class="hint">Move your mouse / tap to interact • Click the name to
  pulse</div>

  <div class="butterflies" id="butterflies" aria-hidden="true"></div>

  <script>
    // ---- Canvas particle background with interactive attraction ----
    const canvas = document.getElementById('bg-canvas');
    const ctx = canvas.getContext('2d');
    let W = canvas.width = innerWidth;
    let H = canvas.height = innerHeight;

    // Settings
    const particleCount = Math.floor(Math.min(220, Math.max(80, (W*H)/60000)));
    const particles = [];
    const mouse = {x: W/2, y: H/2, down:false};
    let motionOn = true;

    function rand(min,max){return Math.random()*(max-min)+min}

    function Particle(){
      this.reset();
    }
    Particle.prototype.reset = function(){
      this.x = rand(0,W); this.y = rand(0,H);
      const s = rand(0.4,1.8);
      this.r = s*2.6; // radius
      this.vx = rand(-0.4,0.4);
      this.vy = rand(-0.4,0.4);
      this.life = rand(80,600);

```

```

    this.maxLife = this.life;
    this.hue = rand(190,230); // bluish
    this.alpha = rand(0.15,0.9);
}
Particle.prototype.update = function(){
    if(!motionOn){
        // slow drift
        this.x += this.vx*0.2; this.y += this.vy*0.2;
    } else {
        this.x += this.vx; this.y += this.vy;

        // attraction to mouse when mouse is moved or down
        const dx = mouse.x - this.x; const dy = mouse.y - this.y;
        const dist = Math.sqrt(dx*dx+dy*dy);
        if(dist < 180){
            const force = (180 - dist) / 180; // 0..1
            // if mouse down, attract stronger
            const k = mouse.down ? 0.6 : 0.25;
            this.vx += dx * 0.0006 * force * k;
            this.vy += dy * 0.0006 * force * k;
        }

        // gentle friction
        this.vx *= 0.995; this.vy *= 0.995;
    }

    // wrap around
    if(this.x < -20) this.x = W + 20;
    if(this.x > W + 20) this.x = -20;
    if(this.y < -20) this.y = H + 20;
    if(this.y > H + 20) this.y = -20;

    this.life -= 1;
    if(this.life <= 0) this.reset();
}
Particle.prototype.draw = function(ctx){
    ctx.beginPath();
    ctx.fillStyle = `hsla(${this.hue},85%,60%,${this.alpha})`;
    ctx.arc(this.x,this.y,this.r,0,Math.PI*2);
    ctx.fill();
}

function initParticles(){
    particles.length = 0;
    for(let i=0;i<particleCount;i++) particles.push(new Particle());
}

function resize(){
    W = canvas.width = innerWidth; H = canvas.height = innerHeight;
    // do not change particleCount on resize to avoid chaos; keep existing
}

window.addEventListener('resize', resize);

// Mouse
window.addEventListener('mousemove',
e=>{mouse.x=e.clientX;mouse.y=e.clientY});
window.addEventListener('mousedown',
e=>{mouse.down=true;mouse.x=e.clientX;mouse.y=e.clientY});
window.addEventListener('mouseup', e=>{mouse.down=false});
window.addEventListener('touchstart', e=>{mouse.down=true;const
t=e.touches[0];mouse.x=t.clientX;mouse.y=t.clientY});
window.addEventListener('touchend', e=>{mouse.down=false});
window.addEventListener('touchmove', e=>{const

```

```

t=e.touches[0];mouse.x=t.clientX;mouse.y=t.clientY});

// Toggle control
const toggle = document.getElementById('particlesToggle');
toggle.addEventListener('click',()=>{motionOn = !motionOn;
toggle.setAttribute('data-on', String(motionOn))});

// Name interaction - pulse on click
const nameEl = document.getElementById('mainName');
nameEl.addEventListener('click', ()=>{
  // create a burst of temporary particles
  for(let i=0;i<28;i++){
    const p = new Particle();
    p.x = nameEl.getBoundingClientRect().left + nameEl.offsetWidth/2 +
rand(-30,30);
    p.y = nameEl.getBoundingClientRect().top + nameEl.offsetHeight/2 +
rand(-20,20);
    p.vx = rand(-3,3); p.vy = rand(-3,3); p.life = rand(40,120); p.alpha =
1; particles.push(p);
  }
});

// Main loop
function step(){
  ctx.clearRect(0,0,W,H);
  // subtle background gradient overlay
  const grad = ctx.createLinearGradient(0,0,W,H);
  grad.addColorStop(0, 'rgba(2,6,23,0.35)');
  grad.addColorStop(1, 'rgba(2,14,28,0.35)');
  ctx.fillStyle = grad; ctx.fillRect(0,0,W,H);

  // draw connecting lines for nearby particles
  for(let i=0;i<particles.length;i++){
    const a = particles[i];
    a.update();
    a.draw(ctx);
    for(let j=i+1;j<particles.length;j++){
      const b = particles[j];
      const dx = a.x-b.x; const dy = a.y-b.y; const d =
Math.sqrt(dx*dx+dy*dy);
      if(d < 120){
        ctx.beginPath();
        ctx.moveTo(a.x,a.y); ctx.lineTo(b.x,b.y);
        ctx.strokeStyle = `rgba(100,160,255,${(120-d)/300})`;
        ctx.lineWidth = 0.6; ctx.stroke();
      }
    }
  }

  requestAnimationFrame(step);
}

// Create some decorative butterflies (SVG) that float and slightly rotate
function makeButterfly(x,y,scale=1,delay=0){
  const div = document.createElement('div'); div.className='butterfly';
  div.style.left = x+'px'; div.style.top = y+'px'; div.style.transform =
`scale(${scale})`;
  div.innerHTML = `
    <svg viewBox="0 0 64 64" xmlns="http://www.w3.org/2000/svg" aria-
hidden="true">
      <path d="M32 32c6-10 18-14 22-10s2 14-8 16-16 2-14-6z"
fill="rgba(255,200,100,0.95)"/>
      <path d="M32 32c-6-10-18-14-22-10s2 14 8 16 16 2 14-6z"
fill="rgba(255,120,180,0.95)"/>
  `

```

```

        <circle cx="32" cy="26" r="3" fill="rgba(30,30,40,0.95)" />
    </svg>`;
    document.getElementById('butterflies').appendChild(div);

    // animate using CSS transforms
    const dur = rand(6000,14000);
    div.animate([
        { transform: `translateY(0px) scale(${scale}) rotate(0deg)` },
        { transform: `translateY(${rand(-40,60)}px) scale(${scale}) rotate($
{rand(-30,30)}deg)` },
        { transform: `translateY(0px) scale(${scale}) rotate(${rand(-10,10)}
deg)` }
    ], {duration: dur, iterations: Infinity, easing: 'ease-in-out', delay: delay});
}

// init
initParticles();
// place a few butterflies at random positions
for(let i=0; i<4; i++){
    makeButterfly(rand(30,W-60), rand(30,H-60), rand(0.6,1.1), i*400);
}

requestAnimationFrame(step);

// small accessibility: allow keyboard focus on name
nameEl.setAttribute('tabindex','0');
nameEl.addEventListener('keydown', e=>{ if(e.key=== 'Enter' || e.key=== ' ')
nameEl.click(); });

// friendly hint: center the name better when window loads
window.addEventListener('load', ()=>{
    // smooth intro
    document.getElementById('app').animate([{opacity:0,
transform:'translateY(8px)'},{opacity:1, transform:'translateY(0)'}],
{duration:600,easing:'ease-out'});
});
</script>
</body>
</html>

```