Build a complete, production-quality Vite project from scratch — a premium dark-editorial portfolio for a fictional AI creative director, centred on a full-screen WebGL fluid-trail reveal hero. Inspired by the codegrid LandoNorris reveal effect — but the design, copy, layout and visual direction must be ORIGINAL. Do not copy any existing site; use the reference only for the interaction (a mouse trail that masks a top portrait to reveal a bottom portrait beneath).
═══ TECH STACK (strict)
═══

* Vite 7 (vanilla JS template, NO React, NO Next, NO TypeScript)
* Three.js (latest stable, ESM)
* Custom GLSL shaders (no postprocessing libs)
* Plain CSS (no Tailwind, no preprocessors)
* Vanilla JavaScript modules
* One small inline <script> block in index.html for the menu + scroll reveal IntersectionObserver. Everything else lives in its own file.
═══ FILE STRUCTURE (exact, root-level — do NOT nest under src/)
═══ project-root/ ├── index.html ├── package.json ├── .gitignore (node_modules, dist, .DS_Store, .env) ├── script.js (entry: Three.js fluid trail + idle auto) ├── shaders.js (named exports: vertexShader, fluidFragmentShader, | displayFragmentShader) ├── styles.css └── public/ └── portrait_top.png, portrait_bottom.png (placeholders, 4:5 vertical, dark editorial portraits) package.json scripts: { "dev": "vite" }. Dependencies: three, vite. Do NOT put JavaScript inside HTML except the tiny menu/observer block. Do NOT inline styles. Keep each concern in its own file.
═══ HERO — FLUID TRAIL REVEAL (technical specification)
═══ The hero is a full-viewport canvas that displays portrait_top.png. The user's cursor (or an auto-driven synthetic cursor when idle) paints a soft trail into an off-screen mask; where the mask exceeds a threshold, portrait_bottom.png is revealed underneath. A subtle gray halo follows the leading edge of the trail so the motion is visible on top of the top image. Use this CONFIG verbatim at the top of script.js: const CONFIG = { // Simulation render-target size (square, ping-pong) simSize: 500, // Trail mask falloff decay: 0.97, lineWidth: 0.09, perFrameIntensity: 0.3, // Reveal threshold (display shader) revealThreshold: 0.02, edgeWidthBase: 0.004, // divided by uDpr in shader // Soft gray halo overlay (display shader) haloUpperMul: 2.0, // halo upper bound = revealThreshold * this haloMixStrength: 0.35, haloGray: [0.12, 0.12, 0.12], // Idle auto-trail idleThresholdMs: 2500, idleEaseInMs: 1500, autoLerp: 0.05, // Mouse stop detection stopAfterMs: 50, // Max texture size maxTextureSize: 4096, }; Implementation steps (in this order):

1. Select <canvas> inside .hero from the DOM.
2. Create a WebGLRenderer({ canvas, antialias: true, precision: 'highp' }). setSize(innerWidth, innerHeight). setPixelRatio(min(devicePixelRatio, 2)).
3. THREE.Scene, THREE.OrthographicCamera(-1, 1, 1, -1, 0, 1).
4. Two ping-pong WebGLRenderTargets at simSize x simSize: { minFilter: LinearFilter, magFilter: LinearFilter, format: RGBAFormat, type: FloatType } Clear both once at startup. Track currentTarget index (0/1).
5. Two THREE.Vector2 mouse states in normalized [0,1] canvas coords: `mouse` and `prevMouse` (both default 0.5, 0.5). isMoving boolean + lastMoveTime timestamp.
6. Two placeholder THREE.CanvasTextures (one filled #0000ff for top, one filled #ff0000 for bottom) so the spiral always has something to sample before real images load.
7. Load /portrait_top.png and /portrait_bottom.png via Image() with crossOrigin = 'Anonymous'. On load, write actual pixel dimensions into a Vector2 size uniform per texture. If width or height > maxTextureSize (4096), downscale via an offscreen 2D canvas before creating the final CanvasTexture. Replace the uniform texture value once ready. Log success and errors.
8. THREE.PlaneGeometry(2, 2) full-screen quad. Two ShaderMaterials: trailsMaterial (simulation) and displayMaterial (final composite). Add a displayMesh to the main scene; add a simMesh to a separate simScene that is only rendered into ping-pong targets. Render loop (every frame, requestAnimationFrame): const now = performance.now(); if (isMoving && now - lastMoveTime > CONFIG.stopAfterMs) isMoving = false; const idleTime = now - lastMoveTime; const autoActive = idleTime > CONFIG.idleThresholdMs; // Swap ping-pong const

prevTarget = pingPong[currentTarget]; currentTarget = (currentTarget + 1) % 2; const writeTarget = pingPong[currentTarget]; trailsMaterial.uniforms.uPrevTrails.value = prevTarget.texture; if (autoActive) { // see IDLE AUTO-TRAIL section } else { trailsMaterial.uniforms.uMouse.value.copy(mouse); trailsMaterial.uniforms.uPrevMouse.value.copy(prevMouse); trailsMaterial.uniforms.uIsMoving.value = isMoving; // mirror so the next idle cycle starts from user's last position autoMouse.copy(mouse); prevAutoMouse.copy(mouse); } renderer.setRenderTarget(writeTarget); renderer.render(simScene, camera); displayMaterial.uniforms.uFluid.value = writeTarget.texture; renderer.setRenderTarget(null); renderer.render(scene, camera); INPUT:

* mousemove: read getBoundingClientRect of the canvas. If the pointer is inside the rect, prevMouse.copy(mouse), update mouse.x / mouse.y to normalized canvas coords (mouse.y is flipped: 1 - ...), isMoving=true, lastMoveTime = now. Outside the rect, set isMoving=false. NEVER update lastMoveTime when outside — that's what lets the auto trail keep running while the user's cursor is elsewhere on the page.
* touchmove: identical logic using touches[0], with preventDefault(). RESIZE:
* renderer.setSize(innerWidth, innerHeight)
* displayMaterial.uniforms.uResolution updated
* displayMaterial.uniforms.uDpr updated

═══════════ SHADERS — shaders.js (named exports) ═══════════

vertexShader (shared by both passes): varying vec2 vUv; void main() { vUv = uv; gl_Position = projectionMatrix * modelViewMatrix * vec4(position, 1.0); } fluidFragmentShader (writes into ping-pong target each frame): uniform sampler2D uPrevTrails; uniform vec2 uMouse, uPrevMouse, uResolution; uniform float uDecay; uniform bool uIsMoving; varying vec2 vUv; void main() { vec4 prev = texture2D(uPrevTrails, vUv); float newValue = prev.r * uDecay; // 0.97 decay if (uIsMoving) { vec2 dir = uMouse - uPrevMouse; float len = length(dir); if (len > 0.001) { vec2 d = dir / len; vec2 toPx = vUv - uPrevMouse; float proj = clamp(dot(toPx, d), 0.0, len); vec2 closest = uPrevMouse + proj * d; float dist = length(vUv - closest); float lineWidth = 0.09; float intensity = smoothstep(lineWidth, 0.0, dist) * 0.3; newValue += intensity; } } gl_FragColor = vec4(newValue, 0.0, 0.0, 1.0); } The closest-point-on-segment math is critical: it draws a continuous line between prevMouse and mouse each frame instead of dots, so fast movement still produces a smooth trail. displayFragmentShader (composites the final pixel on screen): uniform sampler2D uFluid, uTopTexture, uBottomTexture; uniform vec2 uResolution, uTopTextureSize, uBottomTextureSize; uniform float uDpr; varying vec2 vUv; // Replicates CSS object-fit: cover in shader space vec2 getCoverUV(vec2 uv, vec2 ts) { if (ts.x < 1.0 || ts.y < 1.0) return uv; vec2 s = uResolution / ts; float scale = max(s.x, s.y); vec2 scaled = ts * scale; vec2 offset = (uResolution - scaled) * 0.5; return (uv * uResolution - offset) / scaled; } void main() { float fluid = texture2D(uFluid, vUv).r; vec2 topUV = getCoverUV(vUv, uTopTextureSize); vec2 bottomUV = getCoverUV(vUv, uBottomTextureSize); vec4 topColor = texture2D(uTopTexture, topUV); vec4 bottomColor = texture2D(uBottomTexture, bottomUV); float threshold = 0.02; float edgeWidth = 0.004 / uDpr; float t = smoothstep(threshold, threshold + edgeWidth, fluid); // Soft gray trail-visibility overlay (top image only). // Halo peaks just before the reveal threshold and fades out as // t -> 1 so the bottom image still appears clean inside the trail. float halo = smoothstep(0.0, threshold * 2.0, fluid) * (1.0 - t); vec3 trailGray = vec3(0.12); vec3 tintedTop = mix(topColor.rgb, trailGray, halo * 0.35); vec4 finalColor = mix(vec4(tintedTop, topColor.a), bottomColor, t); gl_FragColor = finalColor; }

═══════════ IDLE AUTO-TRAIL (synthetic cursor when the user is idle)

═══════════ State (above the animate loop): const autoMouse = new THREE.Vector2(0.5, 0.5); const prevAutoMouse = new THREE.Vector2(0.5, 0.5); When idleTime > idleThresholdMs (2500), the auto branch in the render loop runs: const easeIn = Math.min(1, (idleTime - idleThresholdMs) / idleEaseInMs); // Layered low-frequency sines on INCOMMENSURATE frequencies — the // resulting path is organic, never repeats, never spikes off-canvas. const t = now * 0.001; const targetX = 0.5
* 0.30 * Math.sin(t * 0.41)
* 0.12 * Math.sin(t * 0.93 + 1.3); const targetY = 0.5
* 0.28 * Math.cos(t * 0.37 + 0.5)
* 0.10 * Math.cos(t * 1.11 + 2.7); prevAutoMouse.copy(autoMouse); autoMouse.x += (targetX - autoMouse.x) * 0.05 * easeIn; autoMouse.y += (targetY - autoMouse.y) * 0.05 * easeIn; trailsMaterial.uniforms.uMouse.value.copy(autoMouse); trailsMaterial.uniforms.uPrevMouse.value.copy(prevAutoMouse);

trailsMaterial.uniforms.uIsMoving.value = true; // Mirror onto mouse/prevMouse so the next real move continues from // where auto left off — no stale long segment on the handoff frame. mouse.copy(autoMouse); prevMouse.copy(prevAutoMouse); RULES:
* Real mouse / touch always wins. Any real move updates lastMoveTime which drops idleTime back to ~0, branching to the user path next frame.
* Auto must NOT alter the shaders, ping-pong logic, or input handlers. It is purely additive: a synthetic cursor written into the same uniforms the real cursor uses.
* Page-load behaviour: lastMoveTime starts at 0, so idleTime ramps up naturally from page load — the trail wakes up smoothly on its own.
═══════════ BRAND + COPY (originality rules) ═══════════ The site is a portfolio for a FICTIONAL AI creative director — a synthetic mind that composes identity, motion and editorial systems. You must produce ORIGINAL:
* Studio / persona name (one word, display-typography-friendly, vaguely synthetic — e.g. ELYRA, NOEMA, AURIE, KORA, VESPER — invent your own; do not reuse these).
* Tagline like "Human form / Machine logic" — short, paired by a slash.
* One-line description: "synthetic creative director", "synthetic mind composing visual systems", etc.
* Three project names for Selected Works (e.g. Synthetic Identity, Motion Interface, Neural Campaign — invent your own equivalent).
* Section labels (use the set below OR invent equivalents): 01 About / Manifesto 02 Selected Works 03 Capabilities 04 Process 05 Experiments / Archive 06 Commission / Contact Tone: editorial-magazine, dry, confident, no marketing fluff, no emojis. Sentences feel printed, not typed. Italics are for emphasis in display serifs only (DM Mono italic counts). ═══════════ DESIGN INTENT — MOOD, MATERIALITY, ATMOSPHERE ═══════════ The site should feel like a printed art magazine that quietly happens to be online — not a tech product, not a portfolio template. Read every spec below as a tonal brief, not a checklist. The aesthetic target is EDITORIAL CALM WITH SYNTHETIC PRECISION: large quiet whitespace, hairline rules, deliberate typographic contrast, and one cool interaction the visitor discovers by moving. REFERENCE FAMILY (mood, not visual copy):
* High-end printed annual reports and studio monographs
* Independent art magazines (Apartamento, MacGuffin, 032c at rest)
* Pentagram-school identity decks — quiet authority, lots of margin
* Awwwards "editorial" or "minimal" winners — but slower, less busy
* Late-90s Swiss editorial systems re-rendered on a 2025 canvas VIBE DESCRIPTORS:
* Warm, paperlike, matte. Never cold corporate grey, never neon.
* Confident but not loud. Headlines are decisive, body is quiet.
* Dry, printed, considered — "this was set, not typed."
* One interaction (the hero reveal) is the only theatrical moment; everything else is still, restrained, magazine-like.
* The page should reward stillness, not aggressive scrolling.
═══════════ TYPOGRAPHY — CHARACTER & PAIRING (extended) ═══════════ The site runs on a deliberate two-typeface system that mirrors the brand premise (human form / machine logic): DISPLAY → a refined EDITORIAL SERIF with a beautiful italic. 'Righteous' is specified in the VISUAL SYSTEM block as the default, but the INTENT is "editorial serif with a strong italic" — 'Instrument Serif' is the production-ready alternative if Righteous reads too display-loud. Either font carries the same role: hero, section titles, work names, CTA, footer lockup. Letter-spacing slightly negative (~ -0.012em) so the display feels "set" rather than tracked. MONO → 'DM Mono' for body copy, all labels, meta, eyebrows, section folios, and italic accents. Use the italic 300 weight for "synthetic italic" lines that sit under display headlines and inside byline tags. Italic mono next to display serif is the single most important pairing on the site — it's the "machine logic" voice. ITALIC ACCENT RULE → whenever a display headline has a 2-line or 3-line stack, ONE line should be italic mono in muted tone, sized ~0.7em of the display line. This is the "synthetic whisper" that appears in About, CTA, footer. TYPE SCALE INTENT (encode as CSS variables): --t-display-xl hero / CTA / footer lockup — heroic, never less than ~2.8rem even on mobile --t-display-lg hero title — primary statement --t-display-md section titles — calm authority --t-display-sm work names, capability headings --t-display-xs process numbers (numerals are display-set) --t-body 1rem mono, line-height 1.65 — comfortable reading --t-meta 0.72rem mono uppercase, letter-spacing 0.22em — technical eyebrow voice ═══════════

EDITORIAL FLOURISHES — THE SMALL PRINT DETAILS
═══ These are the magazine-grade touches that separate this from a generic dark-portfolio template. Each should appear at least once:
* PAGE FOLIOS — small "p. 014"-style page numbers in mono, right- aligned in the section head. Sells the print-magazine illusion.
* DROP CAP — the About lede paragraph opens with a large display drop-cap (~3.6em, line-height 0.9, accent or ink color). One per page only; never repeat.
* RUNNING INDEX STRIP — a thin horizontal strip between the marquee and About reading "N — 01 · Index 2025 / 2026 · Edition 001 · In rotation", separated by short hairline rules. Like a magazine spine label.
* HAIRLINE RULES — every section head, every meta row, every card footer is separated by 1px var(--rule). The hairline IS the layout grammar.
* ACCENT COLOR — a single warm clay/terracotta tone (--accent: #c0392b family) used sparingly: the status-online dot, the drop-cap, the CTA hover state. Never two accents per viewport.
* SLASH TAGLINES — "Human form / Machine logic", "Trace the surface / Reveal underneath", "Online · Accepting commissions" — the slash and middle-dot are part of the brand language.
* NUMBER-FORWARD SECTIONS — section labels lead with 01/02/03, process steps lead with huge display numerals, archive items are "Study #001 … #006". Numbers do the orienting work that breadcrumbs would do in a normal site.
═══ COLOR & MATERIAL (extended palette intent)
═══
* BACKGROUND #ece9e1 — warm off-white, slightly creamy, PAPER. Not pure white, not grey. If you squint it reads like aged stock.
* INK #161513 — near-black with a brown undertone. Pure #000 is too harsh; the ink sits warm against the paper.
* MUTED #6a6760 / DIM #a8a59c — secondary and tertiary type, the voice of printed footnotes.
* ACCENT #c0392b — warm clay red. Used in flashes only. Status dot, drop-cap, occasional CTA accent.
* DARK SECTIONS #0a0a0a — only the CTA and footer. The transition from paper to night is part of the scrolling rhythm.
* HAIRLINES — rgba(10,10,10,0.14) on light, rgba(255,255,255,0.16) on dark. Always exactly 1px, never thicker. ═══ TEXTURE, DEPTH & GRAPHIC LANGUAGE
═══
* TOPOGRAPHIC CONTOUR SVG — fixed behind everything, ~15 horizontal sine-offset paths, opacity 0.07. Acts like printed paper grain or a faint contour-map watermark. Dark sections hide it naturally because their bg is opaque. It should never read as a "background pattern" — only as paper texture.
* WORK VISUAL CONTOURS — each Selected Work has an SVG overlay (concentric ellipses / horizontal flowing curves / grid + circles). These are the ONLY decorative graphics on the site; they substitute for the project hero images a normal portfolio would have. Stroke must be hairline white at low opacity.
* ARCHIVE STUDIES — six CSS-only pattern tiles (lines / rings / grid / waves / diag / dots). No images. The studies are "texture specimens" — the studio's process artifacts.
* NO PHOTOGRAPHY anywhere except the two hero portraits. No icon sets. No illustrations. The brand grammar is: serif + mono + hairline + occasional contour line. Restraint is the look. ═══ PACE & RHYTHM
═══
* The hero is the only moment of motion theatre. After it, the site slows down: long quiet sections, generous padding (5–9rem vertical), short paragraphs, lots of whitespace.
* Scroll-reveal animations are 1s, gentle cubic-bezier — never snappy, never bouncy. Things ARRIVE, they don't pop.
* Hover states are small and patient: a 0.4s expansion, a 4px nudge, a hairline that grows. No scale transforms, no rotations.
* The marquee scrolls at a slow 38s loop — readable, not a strobe.
* Scroll behaviour stays native; no momentum library. The page should feel like turning pages, not surfing. ═══ LAYOUT — sections below the hero
═══ Order, top to bottom, inside <main class="main">: [0] Marquee — single row of small uppercase monospace text

scrolling horizontally on infinite loop. Items separated by a small filled circle. Content like: [BRAND] • Synthetic Creative Direction • Portfolio 2025 / 2026 • Human Form — Machine Logic • Available For Commission • Duplicate the list inside the track so the loop is seamless. [1] About (01) — two-column grid (1.2fr / 1fr). Left: a 3-line display serif headline; the middle line is in italic DM Mono with muted color (synthetic italic accent). Right: 2 body paragraphs + a <dl> meta list with rows: Class / Origin / Focus / Status. [2] Selected Works (02) — section title + ordered list of 3 large work rows. Each row is a 2-col grid (1.1fr / 0.9fr) that ALTERNATES direction on :nth-child(even) using grid order. Left/right of each row: VISUAL: 4:3 dark gradient box (#0a0a0a → #1a1a1a), absolute SVG contour overlay (one of: concentric ellipses; horizontal flowing curves; vertical/horizontal grid + concentric circles), big display title centered on top, small mono SI/MI/NC index badge top-left. Light radial highlight ::after at bottom for cinema feel. INFO: meta row "(01 / 03) … 2025", display title, body paragraph (max-width ~38ch), pill tags (rounded-full hairline border), animated underline "View case →" CTA that expands left→right on hover. [3] Capabilities (03) — section title + 2×2 grid of 4 cards with hairline 1px gaps (use background: var(--rule); + 1px gap; individual cards have solid bg). Each card: small ↳ 01 index, display title, body, "tag list" footer separated by a top hairline. Card bg subtly lightens on hover. [4] Process (04) — section title + 4-column row of "movements": 01 Prompt 02 Concept 03 Refine 04 Experience Each column has a huge display number, a small uppercase title, and a body paragraph. Top/bottom borders are strong rules; each column has a right rule (last child no right). On hover, a 2px accent bar grows left→right across the top of the hovered column using a ::before transition. [5] Experiments / Archive (05) — section title + 3×2 grid of "studies". Each study is a figure: a 4:5 dark visual placeholder (solid #0a0a0a) decorated entirely with CSS backgrounds via data-pattern attribute — implement six variants: lines → repeating-linear-gradient(180deg, white@18% 0 1px, transparent 1px 14px) rings → stacked radial-gradients producing concentric outlines grid → crossed repeating-linear-gradients at 0deg and 90deg waves → radial top + bottom + faint repeating-radial concentric ring diag → repeating-linear-gradient(135deg, white@14% 0 1px, transparent 1px 16px) dots → radial-gradient dot at 18px tile size Below the visual, a figcaption row with "Study #00X" + tag (e.g. "Texture · Identity"). On hover, the figure lifts -4px and the pattern opacity intensifies. After the grid, a pill "Open full archive ↗" link. [6] CTA (06) — DARK section. Background switches to near-black (#0a0a0a), foreground to light cream. Big display heading, middle word in italic DM Mono in dim color. Sub-paragraph, then a pill button with a circular glyph inside that nudges right on hover. Below, a meta row: email + location, separated by hairlines in rule-light. [7] Footer — DARK. Top row: large display brand mark + italic-mono tagline. Middle row: 4 columns of links (Index / Channels / Contact / Index No.) with small uppercase mono labels and underlined hairline rules. Final row: enormous display brand lockup spanning the full viewport width (font-size: clamp(4rem, 22vw, 18rem)). ═══════════ NAV + SLIDE-DOWN MENU ═══════════
<nav> is position: fixed; full width; padded clamp; z-index 50; and uses mix-blend-mode: difference so it reads white on light AND on dark sections without recoloring on scroll. Left: display-serif brand mark. Right: a rounded-pill button labeled "Index" with a small 2-line glyph that morphs to an × when open. When clicked, a full-viewport dark panel (.site-menu) fades in, listing the 6 anchors as huge display-serif numbered rows: 01 About 02 Selected Works 03 Capabilities 04 Process 05 Archive 06 Commission Each row staggers in with a 0.05s delay step (transition on opacity + transform). Hover pads the anchor 1rem left and brightens it. Implement with a tiny inline <script> at the end of <body>: - aria-expanded toggling on the button - .is-open class on the panel - html.menu-open to lock body scroll - Esc key closes - Anchor click closes - 480ms removal delay so the fade-out plays before hidden returns
═══════════ SCROLL REVEAL ═══════════ Use IntersectionObserver (no GSAP, no Lenis — keep it lean): - Select [data-reveal] - Default styles: opacity:0; transform: translateY(28px); transition: opacity 1s, transform 1s cubic-bezier(.2,.8,.2,1). - Observer threshold 0.12, rootMargin "0px 0px -8% 0px" - Add .is-visible on intersect, then unobserve. - Respect prefers-reduced-motion: skip the observer entirely and mark all targets visible immediately. Add data-reveal to every section, every individual .work, the cta, the archive, etc. NEVER add it to the hero — hero is static, only its canvas animates.
═══════════ VISUAL SYSTEM (CSS variables + typography) ═══════════
PALETTE (derive your own values within these intents): --bg warm off-white in the #ece9e1

family --bg-soft slightly lighter card surface --bg-dark near-black #0a0a0a (dark sections only) --ink near-black for body text on light bg --ink-soft slightly softer body color --muted ~#6a6760 secondary --dim ~#a8a59c tertiary / on-dark muted --light ~#f4f1ea on-dark foreground --rule rgba(10,10,10,0.14) --rule-strong rgba(10,10,10,0.42) --rule-light rgba(255,255,255,0.16) TYPOGRAPHY (two Google Fonts only, plus mono italic for accents): --font-mono 'DM Mono' (with italic 300 + 400) --font-display 'Righteous' HERO + display titles use Righteous, sizes via clamp: hero-style display ~ clamp(2.4rem, 5.5vw, 5rem) section-title ~ clamp(2.25rem, 5.5vw, 4.5rem) cta-heading ~ clamp(2.8rem, 9vw, 8rem) Numbers/process numbers use Righteous too. Body uses DM Mono ~0.95rem, line-height 1.65, NOT uppercased. Section meta labels are DM Mono uppercase 0.72rem, letter-spacing 0.22em. Italic emphasis lines are DM Mono italic 300, muted color, ~0.7em of the parent display size — they sit BELOW the display line and feel hand-written next to the serif. FIXED TOPOGRAPHIC BACKGROUND: Inline <svg class="contour-bg"> as the FIRST child of <body>, with ~15 long horizontal stroke paths drawn at incommensurate sine offsets that span the full viewBox. Style: position: fixed; inset: 0; width: 100vw; height: 100vh; z-index: -1; pointer-events: none; opacity: 0.06; color: var(--ink); paths: fill: none; stroke: currentColor; stroke-width: 0.6; It sits behind the body content but above the html background, so sections with transparent bg show the contour through, and dark sections (cta + footer) hide it. ═══ RESPONSIVE ═══ Breakpoints: 960px, 560px. @media (max-width: 960px): - about-grid: 1 column - works: alternating layout collapses to single column, visuals always on top - capabilities-grid: 1 column - process-steps: 2 columns - archive-grid: 2 columns - footer-grid: 2 columns @media (max-width: 560px): - process-steps: 1 column - archive-grid: 1 column - footer-grid: 1 column - section padding tightens - hero/cta display sizes scale down to ~3.2rem max - footer big lockup font-size: 26vw prefers-reduced-motion: kill all transitions/animations, mark data-reveal visible immediately, set scroll-behavior: auto. ═══ NON-NEGOTIABLES ═══ - The hero shader architecture (vertex + fluidFragment + displayFragment), uniform names, ping-pong logic, decay value 0.97, line width 0.09, reveal threshold 0.02 — exactly as specified. Do not redesign. - Real mouse / touch interaction must not be removed or rewritten. The idle auto-trail is additive only. - Touch support intact (preventDefault inside the canvas rect only). - No frameworks. No CSS preprocessors. No Tailwind. No GSAP. No Lenis. - IntersectionObserver for reveals; that's all the animation library needed. - No JavaScript inside HTML except the menu / observer block at the end of <body>. - prefers-reduced-motion respected throughout. ═══ DELIVERABLES ═══ 1. Create every file in FILE STRUCTURE. 2. Run `npm install`. 3. Run `npm run dev` and verify both portrait images load (or the blue / red placeholders render if the user hasn't supplied images yet — the site must still animate). 4. At the end of your reply, print these six explanations: a. How the ping-pong simulation produces the trail mask. b. How the display shader composites top + bottom + halo. c. How the idle auto-trail wakes up and hands off to real input. d. How the topographic contour SVG is layered behind sections. e. How the mix-blend-mode: difference nav stays legible on both light and dark sections. f. Which CONFIG keys to tune for: trail thickness, reveal speed, halo strength, idle delay, and auto-cursor amplitude. ═══ LIVE PREVIEW (FINAL STEP — DO NOT SKIP) ═══ After every file is written and dependencies are installed, host the project so the user can click straight into it. 1. Start the Vite dev server as a BACKGROUND / LONG-RUNNING process: npm run dev Use background bash, detached process, or a preview MCP — never run it in the foreground and immediately exit. 2. Wait for Vite to print "ready in …ms" and the Local URL (http://localhost:5173/ or :5174). Verify with a quick fetch on `/`, `/script.js`, `/portrait_top.png` — all three should respond 200 (the image path may 404 if no images supplied; that's OK as long as `/` and `/script.js` return 200). 3. If your environment exposes a forwarded public URL, capture THAT URL. Otherwise use the plain localhost URL. 4. DO NOT stop, kill or restart the dev server. It must stay alive. 5. As the VERY LAST line of your reply — after the six explanations, after everything — print the live URL as a clickable Markdown link on its own line with a clear marker and nothing after it: ▶ Live preview: http://localhost:5173/ No closing summary, no "let me know if…", no farewell. The link is the final character of your reply. 6. If the dev server fails to start, diagnose and fix the root cause before ending. Never hand back a dead URL. Acceptable fallback for port collision only: change the Vite port and update the printed

URL to match. Go.