

Halania Dixon

Front-End UI Developer • Accessibility-First Design Engineer

Atlanta, GA (Remote-ready) • halania.r.dixon@gmail.com • (470) 727-4888
www.HalaniaDixon.com • linkedin.com/in/halaniadixon

6 production applications shipped solo • 124 hand-built components across 130 screens with no component library • WCAG 2.2 AA gated by axe-core in CI • ARIA markup across 116 files • reduced-motion paths in 6 projects

01 SUMMARY

Front-end UI developer who builds interfaces that measure. I built my first website in 2021 for the salon I owned, ran and maintained it for four years, then moved from a site builder into React, TypeScript, and design systems. Since then I have shipped six production applications solo, hand-building 124 components across 130 screens with no component library. Accessibility is a build requirement rather than a retrofit: WCAG 2.2 AA gated by axe-core in continuous integration, ARIA markup across 116 files, keyboard equivalents for every gesture, and reduced-motion paths in six projects.

02 TECHNICAL SKILLS

Core — TypeScript, JavaScript (ES6+), semantic HTML5, CSS3, GLSL, React 19, Next.js 16 (App Router, React Server Components, Server Actions), Vite, React Router

Styling and systems — Tailwind CSS v4, PostCSS, CSS custom properties and design tokens, multi-palette theming with light, dark, and reduced-stimulation modes, responsive layout (Flexbox, CSS Grid), 124 hand-built components; dependency-light implementation — no component library, custom GLSL written from scratch rather than library presets, hand-rolled SVG gauge and Kanban interactions

Accessibility — WCAG 2.2 AA, ARIA labeling, axe-core gating in CI, keyboard navigation including button equivalents for every swipe gesture, focus management, prefers-reduced-motion, color-contrast remediation, semantic landmarks, progressive enhancement, graceful degradation

State and interaction — Zustand, React Context with useReducer, useOptimistic and optimistic updates, error boundaries, Framer Motion, GSAP, Lenis, touch gestures, Web Speech API

3D, shaders, and creative — Three.js, React Three Fiber, drei, postprocessing, custom GLSL fragment shaders (simplex noise written from scratch, uniform-driven), shaderMaterial, vite-plugin-gls, scroll-scrubbed video, 360-degree panoramas, WebGL capability gating with non-WebGL fallback

Mobile and PWA — React Native, Expo Router, progressive web apps, service workers, Web Push, offline-first, app manifests, responsive breakpoints

Performance — code splitting, dynamic imports with SSR disabled, lazy loading, memoization, cache-header strategy, device-capability gating, real-device performance debugging

Behind the interface — PostgreSQL and Supabase, row-level security, REST API design, Zod validation, OAuth 2.0, Stripe Checkout, Anthropic Claude API, rate limiting

Tooling and AI-assisted development — Git, GitHub, GitHub Actions CI, Vercel, ESLint, strict type checking, Vitest, React Testing Library, pull-request workflow, conventional commits, Claude Code (committed CLAUDE.md project instructions, agent-driven feature branches)

03 EXPERIENCE

Halania R. Dixon — Front-End UI Developer • Founder & Operator • Atlanta, GA • 2016 – Present

Independent studio for accessible front-end, design engineering, and immersive web. Engagements below listed most recent first.

Selah **Sole Developer**

A cross-platform wellness app covering exercise programming, readiness scoring, nutrition tracking, and journaling.

- Built 72 screens in React Native and Expo Router with TypeScript, shipping to iOS, Android, and a progressive web app from one codebase with a custom post-export step that injects the PWA manifest and service worker Expo does not generate.
- Wrote 196 accessibility props across 21 components and 72 screens, covering labels, roles, hints, and state, and resolved contrast failures found in a dedicated legibility QA pass.
- Implemented a design token system with 3 color palettes and 3 theme modes including a reduced-stimulation mode, backed by a dedicated accessibility store and a user-facing settings screen.

STAXK **Creator & Sole Developer**

A job-search platform that sources postings, matches them semantically against a resume, drafts outreach, and holds every message for human approval before it sends.

- Built 29 React components in TypeScript and Tailwind CSS v4 on a custom design token system, including a hand-built dual-ring SVG gauge and a Kanban pipeline with full keyboard navigation and a button equivalent for every swipe gesture.
- Set WCAG 2.2 AA as a build gate by running axe-core against 5 routes in GitHub Actions, made possible by building a credential-free demo mode so authenticated pages could be audited without secrets. The only measured accessibility pipeline in my portfolio.
- Added optimistic updates with React useOptimistic across the approval queue and card views so actions render instantly while the server action settles.

The Meridian **Creative Developer** realtor-scroll.vercel.app

A luxury home-builder marketing site built as a scroll-driven cinematic experience.

- Wrote a custom GLSL fragment shader implementing Ashima simplex noise from scratch, 86 lines, driven by time, scroll, and reveal uniforms through a react-three-fiber shaderMaterial rather than a library preset.
- Built "Step Inside": a GSAP crash-zoom through a home's cover image that cross-fades into a draggable 360-degree interior panorama mapped onto the inside of a Three.js sphere.
- Built on Vite 8, React 19, TypeScript 6, and React Router across 4 routes, with Three.js, drei, postprocessing, GSAP, Lenis, and vite-plugin-gls.

AURA **Front-End Developer** aura-apparel-shop.vercel.app

A streetwear storefront with a full browse-to-checkout flow and a 3D product gallery.

- Built 33 components across 6 pages in Next.js 16, React 19, and Tailwind v4, with an Object Mode 3D product gallery in Three.js and react-three-fiber over a catalog of roughly 42 products.
- Managed cart and saved-item state with React Context and useReducer persisted to local storage, shipped an error boundary, and closed the build with an axe accessibility pass.
- Wrote a 163-line Python asset pipeline that removes product backgrounds with rembg, trims to bounding box, converts to WebP, and emits the JSON manifest the storefront reads.

HalaniaDixon.com — Inside Her Head **Creative Developer** halaniadixon.com

My personal site and portfolio, built as a scroll-driven WebGL experience.

- Built a scroll-driven WebGL portfolio with Three.js and React Three Fiber, scrubbing an intro video by scroll position and cross-fading on its own final black frame into a live 3D scene so the transition reads as one continuous shot.
- Held the 3D gallery's drag and wheel controls dormant until the scroll-scrubbed intro completed, resolving an input conflict where gallery controls would otherwise consume the scroll driving the video.
- Choreographed roughly 15 viewport-heights of scroll across 7 named timing constants, and dynamically imported the WebGL canvas with server rendering disabled behind a color-matched placeholder to avoid hydration errors and load flash. Debugged in-app browser lag, iOS video pop-out, and low-end device performance with a runtime capability gate.

TradeSprout **Sole Developer** 115 commits

A trading journal and market-education app that reads chart screenshots with a vision model and explains them in plain language.

- Built a 17-page Next.js 16 application in React 19 and TypeScript, keeping 53 percent of components as React Server Components so data flows from database queries into server components with no client-side fetching layer.
- Wrote 44 components with accessibility markup in 44 files, including 13 prefers-reduced-motion handlers, 6 aria-live regions, and a hit-area test that enforces WCAG target sizing on every glossary term in the app copy.
- Built a two-layer glossary and read-aloud system that makes any term in the product's own copy tappable, covered by React Testing Library tests inside a 1,076-case Vitest suite.

Attune **Sole Developer**

A real-time voice application for conversation practice that tracks emotional state and goal progress turn by turn.

- Built a real-time voice interface with the Web Speech API, wrapping speech recognition with pause-detection auto-send and pausing the microphone during synthesized playback so the app does not transcribe its own voice.
- Fixed silent audio playback on mobile Safari by unlocking the audio context inside a user gesture and routing synthesized speech through the Web Audio API.
- Built 10 pages and 9 components in React 19 and Tailwind CSS v4 with dual input modality, so the product stays fully usable by typing when a browser does not support speech recognition.

Breadcrumb **Sole Developer**

An offline-first personal task app with native background notifications, shipped as both a web app and an iOS build from one codebase.

- Built a progressive web app in Next.js 16 and React 19 with a programmatic manifest, service worker, offline support, and a JSON export and import path so a local-storage-only app is not a data-loss risk.
- Set no-store cache headers on the service worker so a stale worker can never pin users to an old build, and applied nosniff, frame-deny, and referrer-policy headers on the server build only, since a static export has no proxy to set them.

6 Figure Playbook **Creator & Sole Developer** ask for demo link

A daily accountability app for tracking habits and revenue against targets.

- Built a daily accountability interface as an installable progressive web app with offline capability, tracking habits and revenue against targets, designed for repeat daily use. Personal production build.

AI Makers Generation **Front-End Developer** 111 commits · collaborative repository · aimakersgeneration.com

The membership, payments, and course platform for an Atlanta AI education community running weekly meetups, workshops, and cohorts.

- Built a React 19 and Vite single-page member application with React Router, and deliberately kept the two public marketing surfaces as hand-written HTML because client-rendered shells are not reliably indexed by non-Google crawlers.
- Implemented Supabase OAuth sign-in across 4 providers with explicit redirect handling back into the member app, and routed 3 hostnames from one deployment through platform configuration alone.

OYRB (Own Your Brand) **Founder & Sole Developer** oyrb.space

A booking and storefront platform for independent beauty professionals.

- Booking platform for beauty professionals. Designed its template system so storefronts are generated from a small set of layout components and themes rather than built page by page, with plain-language documentation for non-technical users.

Glambox Room — Website **Designer & Developer**

The website for the salon I owned and operated.

- Designed, built, and maintained the salon's website on Wix, writing custom HTML and CSS for layout and styling beyond the platform defaults. Ran and updated it for four years. First site built end to end, and the start of my development practice.

Glambox Room Salon **Founder & Owner** 2016 – 2025

An independent Atlanta salon I founded and ran for nearly a decade.

- Independent Atlanta salon for nearly a decade: operations, pricing, brand, and a celebrity clientele grown by word of mouth.

04 EDUCATION

Savannah State University — Business

Virginia College — Cosmetology