

Halania Dixon

Full-Stack Developer · Full-Stack Engineer · AI Engineer

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

9 applications shipped to production · 84 database tables and 47 row-level security policies designed · 1,076 tests, including migrations verified against real Postgres · 15-agent event-driven AI system in production · 26 REST endpoints across 5 apps · 417 commits across 8 repositories

SUMMARY

Founder-operator turned engineer. 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 PostgreSQL. Since closing the salon I have shipped nine production applications solo — roughly 104,000 lines across 813 source files, 84 database tables, 47 row-level security policies, and 26 API endpoints. I own the whole stack on everything I build: schema design and migrations, auth, payments, background workflows, the API surface, accessible UI, CI, and deployment. I develop with Claude Code in an agent-assisted workflow — committed project instructions, feature branches, and pull-request review — which is how that volume shipped solo.

TECHNICAL SKILLS

Languages — TypeScript, JavaScript (ES6+), SQL, Python, HTML5, CSS3

Frameworks and runtimes — React 19, Next.js 16 (App Router, React Server Components, Server Actions, route handlers), Node.js, React Native, Expo, Vite, Deno

Databases and data modeling — PostgreSQL, Supabase, Neon, Drizzle ORM, schema design, database migrations, row-level security, indexing, partial unique indexes, CHECK constraints, stored procedures, triggers, full-text search (tsvector), pgvector

APIs and services — REST API design, serverless functions, webhooks with signature verification, rate limiting, cron and scheduled tasks, polling, pagination, Zod schema validation

Auth and payments — OAuth 2.0 (Google, GitHub, Twitter, Discord), JWT verification, session management, Clerk, Supabase Auth, Stripe Checkout, Stripe webhook reconciliation

AI engineering — Anthropic Claude API, OpenAI API, multi-agent orchestration, event-driven agents, human-in-the-loop approval design, function calling and tool use, structured outputs, versioned prompt engineering, embeddings (OpenAI and Voyage), semantic search, vector similarity, eval harness with golden sets, token-cost optimization

Background work — Inngest durable workflows, typed event bus, fan-out with per-source caps, scheduled agents

Testing and quality — Vitest, React Testing Library, unit and integration testing, migration testing against real Postgres via PGlite, ESLint, strict type checking, pull-request workflow, branch conventions, conventional commits

DevOps and AI-assisted development — Git, GitHub, GitHub Actions CI, Vercel, environment and secret management, Supabase Edge Functions (Deno), Claude Code (committed CLAUDE.md project instructions, agent-driven feature branches, pull-request review)

Front end — Tailwind CSS v4, design tokens and CSS custom properties, responsive layout (Flexbox, Grid), Zustand, React Context with useReducer, error boundaries, Vite, Three.js and React Three Fiber, Framer Motion, code splitting, lazy loading, memoization

Accessibility — WCAG 2.2 AA, ARIA, axe-core gating in CI, keyboard navigation, focus management, prefers-reduced-motion, color contrast, semantic HTML, progressive enhancement

EXPERIENCE

Halania R. Dixon — Full-Stack Developer · Founder & Operator · Atlanta, GA · 2016 – Present

Independent studio for full-stack product, AI systems, and accessible front-end. Engagements below listed most recent first.

STAXX 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 an event-driven system of 15 background agents on Inngest with a TypeScript-typed event map of 13 event types, so agents chain by emitted events rather than direct calls and any stage re-runs independently.
- Designed a 19-table Supabase PostgreSQL schema with 25 row-level security policies across 14 migrations, including storage-bucket policies for PDF uploads, with the service-role client isolated to server-only modules.
- Implemented semantic matching with pgvector, embedding documents at 1,536 dimensions behind an IVFFlat cosine index with a hand-written cosine scorer, and normalized 9 external sources into one table with a dedicated deduplication migration.
- Set WCAG 2.2 AA as a build gate by running axe-core against 5 routes in GitHub Actions, and built a credential-free demo mode so authenticated pages could be audited without secrets. Ships alongside a golden-set eval harness on a second workflow.

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.

- Designed a 29-table PostgreSQL schema in Drizzle ORM across 20 migrations, including CHECK constraints, two partial unique indexes, and a generated tsvector column for full-text search.
- Wrote 1,076 tests in Vitest, including an integration test that applies production migrations to Postgres compiled to WebAssembly through PGlite, verifying every table, constraint, and index before any DDL reaches the hosted database.
- Fixed a silent production failure where Vercel Cron sent bearer tokens that Clerk middleware answered with a 307 redirect, which Vercel reports as success; exempted the three cron routes and gated them on a required shared secret that refuses to run when unset.
- Built vision-model chart reading with Zod-enforced structured output, then a guardrail chain that sanitizes, checks banned phrases, validates level ranges, and declines low-confidence output rather than displaying it.

AI Makers Generation Full-Stack 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 Stripe Checkout payment flow with signature-verified webhook handling, writing a pending application row at session creation and reconciling it to paid when the verified webhook event arrives.
- Found and fixed an authorization vulnerability where an AI chat endpoint read caller identity from the request body and executed database-write tools with the service-role key, bypassing row-level security; moved identity to server-verified Supabase JWTs, added per-tool authorization checks, scoped edits to author ownership, and removed a wildcard CORS header.
- Designed a 7-table Supabase schema with 15 row-level security policies, implemented OAuth across 4 providers, and wrote 2 Deno edge functions for SMTP mail and content updates so those credentials stay off both the client and the serverless layer.

Selah Sole Developer

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

- Designed a 29-table Supabase PostgreSQL schema with row-level security, database functions, and triggers, and shipped 72 screens to iOS, Android, and a progressive web app from one React Native and Expo codebase.
- Architected the app local-first, defining every backend service as an interface with a mock implementation, so the product runs fully without a backend and Supabase is additive rather than required.
- Kept a third-party API key out of the client bundle by deferring the call to a Supabase Edge Function behind the same interface, with the public anon key justified by row-level security on every personal table.

Attune Sole Developer

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

- Built 11 REST endpoints in Next.js route handlers with a single shared rate limiter applied uniformly across every AI route rather than per-route checks.
- Designed a Claude tool-use contract returning each conversational turn as structured data — 8 state values, the reason each changed, behavior flags, and goal progress — removing free-text parsing from the response path.
- Implemented graceful degradation for text-to-speech, returning a 503 with a machine-readable fallback flag when the provider key is absent so the client drops to browser speech synthesis at zero API cost.

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.

- Produced two build shapes from one codebase by branching next.config on an environment flag, using pageExtensions to compile server route handlers out of the static export that Capacitor bundles — no dead-code guards and no second codebase.
- Shipped a progressive web app with a programmatic manifest, service worker, offline support, and a JSON export and import path, plus a Capacitor 8 iOS shell with native background notifications.

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 application as an installable progressive web app with offline capability, tracking habits and revenue against targets. Personal production build, used daily.

AURA Sole Developer aura-apparel-shop.vercel.app

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

- Built the checkout route to recompute every subtotal from the trusted server-side catalog and reject unknown product identifiers, so a client-sent price is never honored. Stripe-ready: the server-computed total is what a PaymentIntent would charge.
- Put virtual try-on behind a provider-adaptor pattern, so the interface calls one internal route and runs in demo mode until a provider environment variable is set, then goes live with no interface changes.
- Wrote a 163-line Python asset pipeline (rembg background removal, bounding-box trim, WebP conversion, JSON manifest) feeding roughly 42 products across 33 components, 6 pages, and 3 API routes.

Destination College Park Commissioned destinationcollegepark.com

A 501(c)(6) destination-marketing nonprofit for College Park, Georgia.

- Automated a municipal RFP proposal deliverable with 15 composable Python scripts using python-pptx, each owning one transformation of the deck so content changes were re-runnable and the final document reproducible; dropped to raw OOXML manipulation where the library did not expose the needed control.

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

A booking and storefront platform for independent beauty professionals.

- Booking-and-payments platform for beauty professionals, built on Next.js, React, TypeScript, and Supabase. Designed its template system so storefronts are generated from a small set of layout components and themes rather than built page by page.

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. The domain knowledge OYRB is built on.

EDUCATION

Savannah State University — Business

Virginia College — Cosmetology