

PROFESSIONAL SUMMARY

Full-Stack & AI Engineer specialized in modern web applications and AI agent systems using TypeScript, Next.js, React, Node.js, and PostgreSQL. Experienced in end-to-end product development with a strong focus on clean code, security, and reliable performance.

TECHNICAL SKILLS

- Programming Languages:** TypeScript, JavaScript, Python, SQL
- Frontend:** React, Next.js, TanStack Query, Zustand, Tailwind CSS, Framer Motion
- Backend & DB:** Node.js, PostgreSQL, Better Auth, Supabase, Convex, Zod
- AI & Agents:** Vercel AI SDK, Agentic Tool-Calling, Multi-Provider Pipelines, Prompt Engineering, Structured Outputs
- Tools & Testing:** Vitest, Docker, Git, GitHub Actions, Vercel

WORK EXPERIENCE

Full-Stack & AI Engineer (Independent / Freelance) Jun 2025 — Present

Designed, built, and shipped multiple production web applications end-to-end, with a focus on AI agent systems and full-stack development.

Strata AI (Agentic Workspace Studio) Demo Code

NEXT.JS · VERCEL AI SDK 7 · GOOGLE GEMINI & FIREWORKS (DEEPSEEK) · DEXIE.JS (INDEXEDDB) · ZOD

- Agentic Tool-Calling Engine:** Built an 8-tool agentic workspace (6 file ops + Tavily web search & page extraction) using closure-captured tool contexts for stateless multi-step mutations, with multi-pass auto-continuation extending single requests to ~75 effective steps.
- 3-Tier Surgical Edit Engine:** A pure-TS cascading string matcher (exact → whitespace-normalized → 2-point anchor bounded) with ambiguity guards, reliably applying LLM code/document edits without file corruption.
- Token & Context Engineering:** Slashed per-step context with metadata-only file listings and an on-demand readFile protocol, added an active 128k-token context guardrail with a live usage meter, per-model USD cost tracking from provider-reported usage, and a /compact command streaming structured summaries with server-side history pruning, keeping long agent sessions bounded, cost-visible, and stable.

Aurora (Luxury E-Store & Admin Panel) Demo Code

NEXT.JS · REACT · TAILWIND CSS · POSTGRESQL · BETTER AUTH · LEMON SQUEEZY

- Payment Orchestration:** Structurally eliminated oversells and double charges: stock is reserved under SELECT ... FOR UPDATE row locks with 35-minute soft holds and debited only on verified webhook delivery, while a UNIQUE-constrained event ledger and timing-safe HMAC verification prevent double-processing. Prices are re-fetched from the DB server-side, making client tampering impossible.
- N+1 Query Elimination:** Eliminated N+1 catalog reads by compiling nested product data (images, sizes, details) into a single json_agg SQL roundtrip, cutting detail-page latency by 70–95% with zero ORM overhead, paired with use cache directives (60s–60min TTLs), and achieving a 99/95/100/100 Lighthouse performance profile.
- Security & Testing:** Production hardening across the stack: 2-tier RBAC (user=0/admin=10) enforced at the edge proxy, server layout, and every admin endpoint (401/403), DB-backed sliding-window rate limiting on auth, newsletter, and checkout, and an immutable audit_logs trail. Backed by 230+ automated tests across 24 test suites.

Break It Down (AI Task Architect) Demo Code

NEXT.JS · SUPABASE · GROQ & GEMINI · MOTION · ZUSTAND

- Asymmetric Dual-LLM Pipeline:** Built a resilient task generation system using two distinct SDK paths, Groq via generateText with automated fallback to Gemini via generateObject with native schema enforcement, maintaining identical output contracts across both providers. Integrated an adaptive prompt engine that dynamically adjusts step granularity and tone based on user energy levels.
- Optimistic Cache Mutations:** Implemented TanStack Query optimistic updates with context-captured rollback, eliminating perceived network latency across all task and step mutations.
- Production AI Guardrails:** Constructed a multi-layer output pipeline: custom JSON sanitizer for malformed LLM responses, Zod runtime schema enforcement for structural validity, and typed error classification (SyntaxError → 502, ZodError → 502, generic → 500), preventing hallucinated or broken AI output from reaching users.

CERTIFICATIONS & LANGUAGES

- Next.js & React Architecture:** Scrimba Coursework
- Spoken Languages:** English (Professional), Urdu (Native)