// DFI_BANK · hello@getsorce.xyz · Orlando, FL · Remote-Ready
github.com/lazereyeninjas-sys · jlpz.dev

J. Lopez

Full-Stack Developer · Product Engineer · Multi-Agent AI Systems

SUMMARY

Self-taught full-stack developer and solo founder architecting multi-agent AI development systems across production Web3 infrastructure. Designs, builds, and ships end-to-end — frontend, backend, database, infrastructure, and blockchain — from zero to paying users, no handoffs. Operates parallel Claude Code agent systems (Opus 5 orchestration, Sonnet 5 execution) across isolated git worktrees to ship faster than most teams. Deep experience with Next.js, Supabase, Railway, Vercel, Stripe, Python async services, Solana, and multi-agent AI pipelines. Thrives where ownership, shipping velocity, and systems thinking all matter.

EXPERIENCE

Founder & Full-Stack Developer — Sorce

2025 – Present · Orlando, FL (Remote) · app.getsorce.xyz

— Architected and operate a **6-agent Claude Code multi-agent development system** across parallel git worktrees — ORCHESTRATOR (Opus 5) coordinates and delegates via AGENTS_LOG.md; FE, BE, DB, DEVOPS, and QA agents (Sonnet 5) execute concurrently with injected CLAUDE.md context per agent. Ships features faster than a traditional team handoff model.

— Architected and shipped a multi-tier AI content SaaS from zero to paying users — full-stack ownership across Next.js 14 frontend, Supabase PostgreSQL backend, Stripe billing, and Railway background services.

— Engineered a complete AI content pipeline: Anthropic API for copy generation, Replicate/Flux for image generation, ElevenLabs for text-to-speech voiceover, with per-tier usage limits enforced at the database level.

— Built and deployed a social post scheduling system: Railway cron worker, inline scheduler UI, Telegram auto-posting bot with DM vs group context detection, and Schedule All bulk-publish feature.

— Implemented three authentication systems: Universal OTP (replacing magic links for Gmail compatibility), hCaptcha bot protection, and Phantom/Solflare wallet auth via Ed25519 signature verification.

— Designed and maintained full multi-service production infrastructure: Next.js App Router on Vercel, Supabase with RLS policies, Railway async workers, and end-to-end Stripe webhook pipeline.

— Architected the VP Marketing Hub: full-stack integration of Clay (prospect discovery), **OmniSocials** (multi-platform publishing, user-pay workspace model), Supermetrics (cross-platform analytics), and LunarCrush Social Radar (Web3 intelligence) — with a separate billing surface across 14 Stripe price IDs.

— Migrated social publishing infrastructure from Ayrshare to OmniSocials — per-user encrypted API key architecture, webhook handler, and publish route update. Ayrshare fully deprecated.

— Wired **Anthropic Batch API** for auto-run route with two-phase research + content architecture tracked in agent_batch_runs table. Cron hardened with auth guard and fail_stale cleanup.

— Built Agent OS — 23 modular system prompts with global skill inheritance architecture (base compliance + agent-build layers extending to Sorce-specific configs). Persona evolution system maps user tier to distinct PERSONA_TONE profiles, gating features and adjusting voice per persona.

— Deployed prompt caching with scale triggers and a /api/health endpoint monitoring 6 external services in real time — feeding a notify chain (Railway → SRCerer bot → Telegram DM) that pages before users notice.

— Configured GitHub Actions CI/CD pipeline — 3-job workflow (ESLint · TypeScript type-check · Next.js build) running on every push to main. Enforces type safety and lint compliance before production deployment.

Founder & Full-Stack Developer — LazerEye Temple

2025 – Present · Orlando, FL (Remote) · lazereyeninjas.xyz

— Built a browser-based 2D fighting game end-to-end in pure vanilla JS and Canvas API — no engine. 4-state CPU AI with per-character personality profiles, spritesheet animation system, parallax stage environments, and character select.

— Shipped Ninja Blaster — a mobile-first Telegram Mini App with real-time global leaderboards, weekly score resets, faction rankings, multi-character system, and full Telegram Bot API backend integration.

— Architected and maintained three concurrent Python async bot services on Railway: TempleBot, V5 Bot, and SRCerer — each with independent asyncio event loops, exponential backoff, and Supabase PostgreSQL integration.

— Built and maintain **LazerSniper** — a production Solana sniper bot with 12-layer token filter stack, Jito MEV bundle execution, 4-wallet rotation, per-position TP/trailing stop/SL override system with price alerts stored in Supabase, and on-chain copy trading via Helius wallet tracking with dynamic win-rate scoring. Expanded into prediction markets via **Kalshi** (RSA-PSS auth, live order placement) and **Polymarket** market feed — both surfaced in a real-time Cloudflare Pages terminal with Phantom wallet auth. BOT SETTINGS panel exposes Railway env vars editable directly from the UI. Security hardened: secrets scrubbed from git history via git-filter-repo, all keys rotated. Repo privatized.

— Operates a **5-agent Claude Code multi-agent system** for LazerSniper development — ARCHITECT, FRONTEND, BACKEND, DATABASE, and TESTER agents running in parallel git worktrees with shared AGENTS_LOG.md and injected per-agent CLAUDE.md. CI/CD pipeline enforces Black, Flake8, and mypy on every push.

— Architecting **TOKENEYEZ** — a 7-layer multi-chain livestream tokenization protocol: Livepeer capture, IPFS storage, LangGraph AI agents, Wormhole bridge, and dedicated smart contract agents per chain — Solana, EVM, TON, Stacks, and XRP (XRPL native).

KEY PROJECTS

Sorce

app.getsorce.xyz

Next.js 14, Supabase, Stripe, ElevenLabs, Replicate, Anthropic API, Railway, Vercel, Clay, OmniSocials, LunarCrush, Supermetrics, GitHub Actions, Batch API, Multi-Agent Claude Code, 6-Agent Worktree System

LazerEye Fighter

lazereyeninjas.xyz

Vanilla JS, Canvas API, Sprite Animation, CPU AI State Machine, Cloudflare Pages

Ninja Blaster

Telegram Mini App

Telegram Bot API, Python, Supabase PostgreSQL, Canvas API, Railway, Cloudflare Pages

LazerSniper

lazersniper.pages.dev

Python, Solana, Jito Bundles, FastAPI, Jupiter API, Kalshi, Polymarket, Railway, Cloudflare Pages, Multi-Agent Claude Code

LazerOps

AI Agent Command Center

Next.js, Supabase, Railway Worker, Telegram Bot API, Python, 25-Agent Orchestration

TOKENEYEZ

In Development

Solana, EVM, TON, Stacks, XRP, Livepeer, IPFS, LangGraph, Wormhole, Dynamic.xyz, Multi-Chain Smart Contract Agents

School of Fresh

schooloffresh.com · Live

Next.js, Supabase, Supabase Realtime, Three.js / React Three Fiber, GSAP ScrollTrigger

1000.1

Local App · In Development

Tauri, Rust, ONNX Wake Word, Silero VAD, whisper.cpp, Piper TTS, Claude API, React

SCOUT-WORKER

Internal Tooling

Python, Supabase, Anthropic Sonnet 5, GitHub API, Railway Cron

Sorce Composer

Chrome Extension

Manifest V3, Service Worker, Chrome Side Panel API, Content Scripts, Smart Inject, Sorce API

TECHNICAL SKILLS

Frontend Next.js 14, React, Vanilla JS / ES6+, Canvas API, CSS, Tailwind, HTML5

Backend Python, Node.js, REST API Design, Webhook Architecture, Async/Await

Database Supabase, PostgreSQL, Row-Level Security, Schema Design, SQL Migrations

Infrastructure Vercel, Railway, Cloudflare Pages, CI/CD, Environment Management

Auth & Payments OTP Auth, Wallet Auth (Phantom/Solflare), Stripe Billing, Stripe Webhooks

AI & APIs Anthropic API (multi-agent, Batch API), ElevenLabs, Replicate/Flux, Telegram Bot API

Blockchain Solana RPC, Jito MEV Bundles, Jupiter API v6, Pump.fun, Token-2022, WebSocket pool detection, simulateTransaction, Kalshi RSA-PSS, Polymarket, XRPL

Tools Git, GitHub, Supabase Studio, Railway CLI, Vercel CLI

AI Dev Systems Multi-Agent Claude Code, Git Worktree Architecture, Opus 5 Orchestration, Sonnet 5 Execution, AGENTS_LOG workflow