BRODY KRETZ

AI Engineer | Agents, retrieval, ML | U.S. citizen
Open to remote or relocation · 561-818-8486 · brodyeckretz@gmail.com · brodykretz.com · linkedin.com/in/brody-kretz · github.com/BrodyKretz

EDUCATION

University of Colorado Boulder | B.S. Information Science, Business minor Expected May 2027

Coursework: Applied Machine Learning, Web Data Science, Statistics for Information Science, Programming for Information Science I and II. Google Data Analytics Professional Certificate (2026).

SUMMARY

Software and AI engineer who ships to production: a client CRM that runs a school's sponsorship program, the backend of a newsletter with 900+ subscribers, and ChessReflex, a live chess trainer with more than 1,800 tests in CI. I build LLM features with the Claude API (RAG, tool calling, structured output, multi agent workflows, eval sets) on TypeScript and Python stacks (Next.js, FastAPI, Node, Postgres, Supabase), and bring a BI and analytics background to data work: SQL, pandas, data imports, migrations and backups on live systems.

EXPERIENCE

BlueSteel AI Solutions | Developer (Contract) Remote · Apr 2026 – Present

- Took over a multi-tenant sponsorship CRM for a private high school (Express, SQLite, React, Tailwind) and have been its only engineer since July 2026. It has run the school's sponsorship program on real data since August, and in October I shipped renewal protection (staged reminders, a Renewal pipeline stage, and a revenue forecast that follows each outcome).
- Backend developer on The Morning Dump, a weekday newsletter with 900+ subscribers (Next.js, MongoDB, Vercel, beehiiv). Own email delivery: the beehiiv integration and its signed webhook, send-safety checks that hold empty or unapproved issues, and getting each issue out on schedule.

PROJECTS

ChessReflex | chess opening trainer, solo build | chessreflex.com Jun 2026 – present

- Built and run an opening trainer that drills only the positions where the opponent has a choice: about 1,350 decisions to learn instead of 33,700 moves to replay across the library (Next.js, TypeScript, Supabase, Stockfish WASM, Vercel).
- Wrote the content pipeline. Scripts branch over Stockfish's top replies to generate 61 courses (about 3,700 lines), and a depth 18 pass rejects any line where our move loses more than 150 centipawns. Puzzles come from the user's own chess.com and Lichess games, with a 34,009 puzzle fallback.

Armoire | AI stylist for one real closet, solo build Next.js, Supabase, Claude · Sep–Oct 2026

- Phone web app that builds each morning's outfit from what someone actually owns, using only clean pieces and dressing for the weather. Claude Haiku tags each photographed piece; Claude Sonnet picks the look with the closet as a cached prompt prefix, and code checks every look (real, clean, complete, weather-sane) and retries with the problems spelled out.
- Background removal with U²-Net on ONNX Runtime, picked from a bake off of seven models to fit a 2 GB serverless function (about 0.5 s per photo). Laundry, wears and a daily AI allowance are Postgres functions under row level security, tested against the real migrations in PGlite. A look costs about a cent.

linesight | a new desktop for macOS Swift, AppKit, WebKit, JavaScript · Oct 2026

- macOS app that turns the desktop into a working layer drawn on the wallpaper: a hover-to-open folder tree with file previews, docked browsers, a terminal strip, video wallpaper, and live Claude Code sessions whose prompts are approved in place.
- Designed and directed a multi-agent Claude Code workflow (builder, parallel reviewer and tester, fixer, adversarial review) over 11 feature batches in 5 days; 735 XCTests, a 246-check UI suite in Chromium and WebKit, zero-warning builds, about 1% idle CPU.

Evoke | agentic RAG research assistant Next.js, Claude API · May 2026

- Retrieval augmented generation over academic literature, modeled on Elicit: expands a question into queries, retrieves papers from Semantic Scholar and CORE, screens each abstract with an LLM against an editable inclusion rule, extracts findings into a table, and writes a report grounded in and citing the retrieved papers. An agent mode runs its own search loop, following references until it has enough.
- Model routing by stage: Haiku screens every paper, Sonnet extracts, Opus writes the report, at about 15 to 25 cents a run. Structured output through forced tool calls so every response parses, with progress streamed live.

SKILLS

AI systems: Claude API (tool use, streaming), LangGraph, LangChain, MCP servers, RAG with ChromaDB and pgvector, local embeddings, eval sets and threshold tuning
ML and data: pandas, NumPy, SciPy, scikit-learn, XGBoost, LightGBM, BigQuery, SQL, Tableau
Engineering: Python, TypeScript, FastAPI, Next.js, Postgres, Supabase, SQLite, pytest, Vitest, GitHub Actions, Docker, Vercel, Fly.io