

Zain Asher

Software Engineer · Production AI Systems

+92 325 0288702 · zainasher42@gmail.com

linkedin.com/in/zain-asher-3960321b2 · github.com/zainasher42 · zainasher-dev.vercel.app

SUMMARY

Software engineer building production AI systems — multi-agent LLM platforms (LangGraph, ~40-agent orchestration), agentic RAG with hybrid retrieval, full-stack delivery on FastAPI and Next.js. Daily user of Cursor and Claude Code. Author of github.com/zainasher42/fastapi-docs-rag — a public production-pattern RAG demo demonstrating the patterns I use in client work.

EXPERIENCE

Ember AI — Software Engineer

Sept 2025 – Present

- Architected multi-agent litigation platform orchestrating ~40 specialized LLM agents through staged synthesis pipelines (LangGraph); foundation → discovery → motion practice → hearing prep → deposition → experts → final strategy. In production with paying users.
- Cost engineering (Anthropic prompt caching, dossier memoization, async-bounded retrieval) that cut token spend by roughly 60% on the multi-agent system.
- Built agentic RAG system for legal document analysis with hybrid retrieval (LLM-driven query generation plus curated seed queries), vision-based OCR via Gemini, ~20 single-responsibility services.
- Shipped multi-LLM media platform (artie.co) end-to-end — async FastAPI backend, Celery/Redis workers, React 18/Vite SPA, OpenAI/Gemini provider failover, deployed on AWS Amplify. Live with real users.
- Daily use of Cursor and Claude Code for AI-driven development — markdown specs as the unit of work, AI tools for implementation under engineer direction.

Devsinc — Associate Software Engineer

June 2025 – March 2026

- Built async web applications with FastAPI, MongoDB/PostgreSQL, and Redis-backed task queues; contributed to API design, Pydantic v2 schemas, and end-to-end feature delivery.

Conovo Technologies — Python Developer Intern

June 2024 – Sept 2024

- Developed web applications with Flask and Django; worked across REST APIs, ORM models, and templated UIs.

SELECTED PROJECTS

Multi-Agent Litigation Platform

Ember AI client · under NDA

FastAPI · LangGraph · Claude (prompt caching) · Gemini · OpenAI · PostgreSQL · SQLAlchemy 2.0 async · Pinecone · Celery/Redis · Next.js

- Ingests case documents and produces motions, discovery plans, deposition outlines, expert reports, and trial themes. ~40 specialized LLM agents composed into seven staged synthesis pipelines.
- Every handoff between stages is a Pydantic-validated schema with explicit status enums and mandatory evidence citations — structured contracts, not free-form text, are what keep ~40 agents consistent.
- Multi-LLM provider abstraction with cost optimizations: Anthropic prompt caching (1-hour TTL), dossier memoization, async Pinecone calls, cooperative cancellation tokens. ~60% token spend reduction versus naive first version.
- Pinecone RAG with llama-text-embed-v2 embeddings, semaphore-bounded concurrency, 40 KB metadata-cap handling; PostgreSQL + async SQLAlchemy 2.0 + Alembic migrations.

Legal Document Analysis Platform — Agentic RAG

Ember AI client · under NDA

FastAPI · Celery · Pinecone · Anthropic Claude · Google Gemini · MongoDB · GCP Storage · Next.js

- Automated extraction and analysis platform for venture-financing deal documents, with async FastAPI and Celery/Redis worker pipelines.
- Multi-LLM pipeline: Gemini for vision-based OCR and text extraction, Claude for legal reasoning and classification. Agentic RAG on Pinecone with hybrid retrieval (LLM-driven query generation plus curated seed queries) and per-question routing between full-text and RAG context.
- Service-oriented backend (~20 single-responsibility services) for PDF chunking, exhibit splitting, cross-document analysis, and a legal inference engine; export to Excel and PDF.

Ask FastAPI Docs — Production-pattern RAG Demo

github.com/zainasher42/fastapi-docs-rag

Python · FastAPI · async SQLAlchemy 2.0 · PostgreSQL + pgvector · Next.js · OpenAI

- Public RAG chatbot over the official FastAPI documentation. Demonstrates the patterns I use in production: hybrid retrieval (vector + keyword via Reciprocal Rank Fusion), structure-aware chunking with header-prefixed self-describing chunks, XML-structured prompts designed for caching.
- SSE streaming with structured event bus (status → citations → token × N → cost → done) for distinct retrieval and generation phases. Paired eval suite (18 hand-curated Q&A pairs, LLM-as-judge runner, 94% pass rate with honest failure reporting).

Artie — AI-Powered Brand & Campaign Platform

artie.co (live)

FastAPI · MongoDB (Motor) · Celery/Redis · OpenAI + Gemini · React 18 · TypeScript · Vite · AWS Amplify · S3

- Full-stack AI media-planning and brand-matching platform with async FastAPI backend, multi-LLM layer (OpenAI/Gemini with automatic fallback), SSE streaming for real-time output, and React 18 SPA deployed via AWS Amplify.
- Celery/Redis workers for brand-project matching, the proprietary Artie Score algorithm, campaign tactics engine, YouTube-data-driven reach calculation, and pitch-deck/script generation. JWT + Argon2 + Google OAuth auth.

SKILLS

- **Languages:** Python, TypeScript, JavaScript, SQL
- **Backend:** FastAPI (async), Flask, Django, Celery, Pydantic v2, SQLAlchemy 2.0 (async), Alembic, Uvicorn, asyncpg
- **Frontend:** React 18/19, Next.js, Vite, TypeScript, TailwindCSS, shadcn/ui, TanStack Query, Zustand
- **AI / LLM:** Anthropic Claude (prompt caching), OpenAI, Google Gemini, multi-LLM orchestration with fallback, LangGraph, Agentic RAG, Pinecone, pgvector, llama-text-embed-v2, prompt engineering with XML-structured prompts
- **AI-driven Development:** Cursor, Claude Code, markdown-spec-first workflows, eval-paired prompt iteration
- **Databases & Storage:** PostgreSQL, MongoDB (Motor), Redis, Pinecone; AWS S3, GCP Storage, MinIO
- **Infra & Tooling:** AWS Amplify, Docker, container deploys, SSE, JWT/Argon2/OAuth, Sentry, structlog, pytest + pytest-asyncio + httpx, document processing (pdfplumber, python-docx, python-pptx, openpyxl, reportlab)

EDUCATION

BS Computer Science — National University of Computer and Emerging Sciences, Lahore

2021 – 2025

- Relevant coursework: Data Structures, Algorithms, Database Systems, Artificial Intelligence, Data Science, Natural Language Processing, Bioinformatics.

LEADERSHIP

- Assistant Vice President, ACM NUCES Lahore Chapter (2023–2024)