ISAAC ORESANYA

Applied AI Engineer · Python, LLM applications, retrieval and structured extraction

Nigeria (UTC+1) · Available now for remote full-time or contract work · isaacjoresanya@gmail.com
isaacjoresanya.com · github.com/TheDataIsaac · linkedin.com/in/isaac-oresanya

SUMMARY

Python developer who builds LLM applications that check their own output: retrieval answers whose quotes are verified against the source, and typed extraction that needs a supporting quote for each key field. Three public AI projects, one deployed. Paid reporting, scraping and data-cleaning work from 2024 to 2026.

AI PROJECTS

What Would Isaac Say? · grounded assistant for a newsletter **Live demo** **Code**
Python, LangGraph, FastAPI, Chroma, Docker

- Built and deployed an assistant that answers from 60 newsletter posts and reviews a reader's draft against their advice. It runs in Docker on a cloud server behind HTTPS.
- Plain Python checks that every quote it shows appears word for word in the source; if no quote survives, the assistant refuses instead of guessing.
- On an 88-question test set it answered 58 of the 60 questions the newsletter covers and refused all 12 unrelated ones. Adding a LangGraph relevance check raised refusals on 10 tricky near-topic questions from 5 to 9.

Job posting extraction and Q&A · structured extraction pipeline **Code**
Python, OpenAI API, Pydantic, asyncio, FastAPI

- Turned 8,595 freelance postings (deduplicated from 9,193 rows) into typed records with a strict Pydantic schema. The model returns null instead of guessing when a posting is silent.
- Required an exact supporting quote for budget, deadline, experience level and red flags, checked in code: 97.1% of records passed the automated quote and evidence checks.
- Runs 10 requests at a time, retries only errors that tend to clear (timeouts, rate limits) and saves each result as it arrives, so a stopped run resumes without paying for the same posting twice.
- Added a FastAPI service that extracts one posting per request, and a RAG command that answers questions over all postings with embedding search and cites the postings it used.

Hybrid search for freeCodeCamp lessons · retrieval and RAG **Code**
Python, NumPy, sentence-transformers

- Wrote BM25 and reciprocal rank fusion from scratch and combined them with local sentence embeddings over 144 lessons (698 chunks). RAG answers cite and link the lessons they came from.

EXPERIENCE

Reporting & Operations Analyst · Data Career Jumpstart (remote) Jan 2024 – Nov 2026

- Owned weekly reporting on 70+ business metrics. Cut preparation from about eight hours to five and raised on-time delivery from about 85% to 98% over 50+ cycles.
- Cut post-delivery corrections by 40% by validating source data before each report went out.
- Built Streamlit self-service views that cut answers to common performance questions from one working day to under 30 minutes.
- Product and QA lead on two internal web apps built with an AI coding agent: wrote the specs and prompts, tested in the live app and required proof from logs or the database before accepting each fix.

Data Analyst (Contract) · Novae, LLC Jun 2024 – Jan 2025

- Built Scrapy and Playwright collectors for three platforms and prepared 141,228 analysis-ready records with pandas.
- Found 60,784 duplicate rows (43.6%) in the first collection and added duplicate removal to each platform's processing step.

SKILLS

LLM applications: OpenAI API with structured outputs, RAG, LangGraph, prompt design, test sets, citation checks
Retrieval: BM25, sentence embeddings, reciprocal rank fusion, Chroma
Backend: Python, FastAPI, Pydantic, asyncio, SQLite, Docker, Git
Data: pandas, SQL, Scrapy, Playwright, Streamlit

EDUCATION

B.Sc. Physics · Federal University of Agriculture, Abeokuta · 2021 – 2024