

CARDOZO SHOFOLARIN CARRINSON

Junior Backend Engineer · Python (FastAPI) · AI / LLM Application Layer

Lagos, Nigeria (UTC+1) | +234 913 238 3099 | folacarrinson@gmail.com

[Linkedln](#) | [GitHub](#) | [Portfolio](#)

PROFILE

Backend developer focused on the application layer of LLM systems: retrieval pipelines, async task handling, and the API surfaces around them. Shipped a Retrieval-Augmented Generation support assistant on FastAPI, Celery, Redis, and Chroma DB, and a demo healthcare platform with JWT auth, Postgres migrations, ML prediction endpoints, and CI on a VPS I administer myself. Strong in Python, comfortable in React, and used to working from a Git repo, a Docker Compose file, and a README.

PROJECTS

Customer AI Support Assistant (B2B, Telegram) · FastAPI · Celery · Redis · ChromaDB · Groq · JavaFX · SQLite. [GitHub](#)

- Built a Telegram bot that answers customer questions from a 50-page PDF knowledge base, returning a grounded answer in under 3 seconds.
- Moved LLM calls off the request path with Celery workers over Redis, so the bot acknowledges each message in milliseconds and serves concurrent conversations without blocking.
- Implemented the RAG pipeline end-to-end: PDF chunking, embedding, Chroma DB vector retrieval, and prompt assembly for a Groq-hosted model.
- Built a JavaFX admin dashboard backed by SQLite for managing knowledge-base context and reviewing bot responses.

AI Healthcare Assistant & Disease Prediction Platform · FastAPI · PostgreSQL · Alembic · JWT · PyTorch · Docker · GitHub Actions · React Native (Expo) · React [Github](#)

- Designed and built the full backend: FastAPI service on PostgreSQL with Alembic migrations and JWT authentication (register / login / me) working end-to-end against a self-managed Ubuntu VPS.
- Shipped three ML-backed prediction endpoints (symptom triage, diabetes risk, heart risk) fronted by a red-flag safety layer that escalates emergency presentations before any model output is returned.
- Served self-hosted Hugging Face models on my own infrastructure rather than a hosted AI API, and provisioned the box myself — Docker, ufw, fail2ban, key-only SSH.
- Set up GitHub Actions CI to run the pytest suite on every push, covering the auth flow and the safety layer.
- Built the two clients against the same API: a React Native (Expo Router) patient app and a React + Vite clinician dashboard.

AI Request Validator API · FastAPI · Pydantic v2 · Layered service/router architecture. [GitHub](#)

- Schema-first validation service for LLM API requests: Pydantic v2 enums, field validators, and typed response models reject malformed payloads before they reach the model, preventing whole classes of downstream failure.
- Separated service and router layers behind typed contracts, so validation rules change without touching transport code.

TECHNICAL SKILLS

Backend: Python (FastAPI, Pydantic v2, SQLAlchemy, asyncio, Celery), REST API design, JWT authentication, Alembic migrations, WebSockets, schema-first API design, service/router layered architecture

AI / LLM: RAG pipelines, vector search (ChromaDB), embeddings and chunking strategies, LLM integration (Groq API), prompt engineering, PyTorch, self-hosting Hugging Face models, model deployment (Hugging Face Spaces)

Data & Infrastructure: PostgreSQL, MySQL, SQLite, Redis (caching, task queues), Docker & Docker Compose, GitHub Actions CI, pytest, self-managed Ubuntu VPS (ufw, fail2ban, key-only SSH), Railway, Linux

Front-End: React (Vite), React Native (Expo), JavaScript (ES6+), HTML5, CSS3, Bootstrap

Tools: Git & GitHub, uv, Postman, VS Code, Jupyter

Core: OOP, data structures & algorithms, relational schema design

EDUCATION

Advanced Diploma in Software Engineering, Aptech Computer Education, Lagos

Sept 2024 – Sept 2026