

Ahmed Lotfy Abousalem

Obour City, Sixth Settlement, Egypt

+201003962652 | ahmedlotfy122@gmail.com | github.com/ahmed-lotfy-abousalem

ahmed-lotfy-abousalem.github.io/portfolio | linkedin.com/in/ahmed-lotfy-abosalem

OBJECTIVE

Software Engineering graduate (Ain Shams University, June 2026) building applied AI systems — retrieval-augmented generation pipelines, LLM-based agents, and multi-agent orchestration using MCP and LangGraph. Combines hands-on experience across the RAG/agent stack (LangChain, LangSmith, vector databases, embedding models) with a security engineering background, bringing a security-conscious approach to designing and deploying AI systems. Seeking a junior Applied AI Engineer role to build and ship production-grade LLM-powered products.

SKILLS

LLMs & Generative AI: Anthropic Claude API, OpenAI API, Ollama (local LLM serving), Claude Skills (SKILL.md authoring), prompt engineering, LLM orchestration.

Agentic Systems: MCP (Model Context Protocol) — server & tool design, LangGraph (agent/graph orchestration, branching & parallel tool routing), LangChain (prompt templates, LCEL, retriever chains), LangSmith (tracing & observability).

RAG & Retrieval: Retrieval-Augmented Generation pipeline design, embedding models (sentence-transformers), ChromaDB, PostgreSQL + pgvector, semantic search, document chunking & indexing.

ML/Distributed Learning: Python, Flower (Federated Learning), on-device/edge model training, privacy-preserving ML.

Backend & Infra: FastAPI, Flask, REST API design, Celery (async task queues), Docker/Docker Compose, PostgreSQL, MongoDB, Prometheus & Grafana.

Security: OWASP Top 10, secure API/authN design, network security fundamentals (Cisco CyberOps, Huawei HCIA-Security) — applied to hardening AI system attack surfaces (data exfiltration, prompt injection awareness, tool-call authorization).

Languages/Tools: Python, C/C++, JavaScript, Bash, Git, Linux, Postman.

AI PROJECTS

Holdco OS — MCP-Based Multi-Agent Orchestration System

- Designed an agentic system that manages multiple sub-companies through tool calls, exposing each sub-company's operations as MCP (Model Context Protocol) tool servers built on Flask.
- Orchestrated agent control flow with **LangGraph** for native branching, routing, and parallel tool execution, with **LangChain** handling prompt templating and retriever wiring.
- Built a local retrieval layer using **sentence-transformers** (all-MiniLM-L6-v2, 384-dim) embeddings persisted in **ChromaDB**, avoiding external embedding API dependency and cost.
- Authored agent capabilities as portable **Claude Skills** (SKILL.md convention) for modular, human-readable tool documentation instead of bespoke tooling.
- Integrated **Anthropic Claude** (Haiku for fast/cheap iteration, Sonnet for the production demo) with an **Ollama** (Llama 3.2) local fallback to keep development cost near zero.
- Instrumented the full agent pipeline with **LangSmith** for run tracing and debugging of multi-step tool-calling decisions.
- Technologies: MCP, LangGraph, LangChain, LangSmith, ChromaDB, sentence-transformers, Anthropic Claude API, Ollama, Flask, Python.*

Mini-RAG Assistant — Retrieval-Augmented Q&A System

- Built a production-style RAG question-answering service: a **FastAPI** layer handling document upload, chunking, embedding, and indexing, with vector storage in **PostgreSQL + pgvector**.
- Implemented the augmented-answer generation flow, combining retrieved context with LLM responses via the **OpenAI API**, with **Ollama** as a local model option for offline/low-cost inference.
- Offloaded document ingestion and indexing to **Celery** workers with a Beat scheduler, keeping the API responsive under load.
- Containerized the full stack with **Docker Compose** and added **Prometheus/Grafana** monitoring for task-queue health and service metrics.
- *Technologies: FastAPI, PostgreSQL, pgvector, Celery, Docker, OpenAI API, Ollama, Prometheus, Grafana.*

Federated Learning-based IDS for Autonomous Vehicles (Graduation Project)

- Building a privacy-preserving Intrusion Detection System that detects anomalies in automotive CAN bus traffic using **Federated Learning** — no raw vehicle sensor data ever leaves a node.
- Implementing local model training on **Raspberry Pi** clients to simulate resource-constrained vehicle ECUs, validating detection performance under edge compute constraints.
- Built the coordination layer as a **Flask REST API** that aggregates model updates from distributed edge clients using the **Flower** federated learning framework, eliminating a centralized data attack surface.
- Producing requirement-traceable documentation covering system design, component interfaces, and verification results.
- *Technologies: Python, Flower (Federated Learning), Flask, Raspberry Pi, CAN bus.*

EDUCATION

B.S., Software Engineering, Ain Shams University

08/2021 – Jun 2026

- Relevant Coursework: Machine Learning, Algorithms, Software Engineering, Operating Systems, Network Security, Cryptography.

B.Sc. Software Engineering, University of East London (Dual Degree)

2021 – Jun 2026

- Dual degree awarded alongside the Ain Shams University programme.

Digital Egypt Pioneers Initiative (DEPI)

2024 – 2025

- Track: Penetration Testing — security foundations now applied to hardening AI system and agent tool-call surfaces.

WORK EXPERIENCE

Technical Instructor, STS

06/2023 – Present

- Taught software fundamentals, secure coding, and system design to 30+ students, translating complex technical concepts into clear, structured training material.
- Mentored ICoolRoboCup teams through full project development cycles — from requirements through integration and testing.

Embedded Systems Developer, Amit

07/2021 – 02/2024

- Developed and maintained firmware for 20+ embedded systems in C, applying structured requirement analysis, testing, and defensive programming — engineering discipline now carried into designing reliable, verifiable AI pipelines.

CERTIFICATIONS

- **Cisco CyberOps Associate**, NTI (01/2024 – 06/2024) — network security monitoring and threat analysis.
- **Huawei HCIA-Security**, NTI (01/2025 – 06/2025) — enterprise network security and threat mitigation.
- **Backend Developer Course**, Sprints (01/2023 – 06/2023) — REST API design, authentication, and data protection, underpinning the FastAPI/Flask services behind both AI projects above.
- **HCDDA-Tech Essentials**, Huawei Cloud (valid until 12/2028) — cloud infrastructure and deployment fundamentals.

LANGUAGES

Arabic: Native | English: Fluent