

SWARAJ CHATTARAJ

B.Tech in Artificial Intelligence (AGEMCO) — Jalpaiguri, West Bengal, India

+91-7501467804 | swarajchattaraj17402@gmail.com | github.com/SWARAJCHATTARAJ | linkedin.com/in/swaraj-chattaraj | swarajchattaraj.tech

SUMMARY

AI/ML developer specializing in agentic AI systems, Retrieval-Augmented Generation (RAG), and LLM application development. Built and shipped a 5-stage agentic RAG pipeline with automated evaluation guardrails (Ragas faithfulness, relevance, and context precision), cutting retrieval latency 63% and deploying production infrastructure across web, desktop, and Android surfaces on Azure.

EDUCATION

Alipurduar Government Engineering and Management College (AGEMCO) — affiliated with MAKAUT Expected 2027
B.Tech in Artificial Intelligence — 6th Semester (3rd Year)

PROJECTS

ARIA — Autonomous Research & Intelligence Assistant 2026 – Present

Python · LangGraph · LangChain · ChromaDB · FastAPI · React · Azure · Azure OpenAI · Supabase · Caddy · PyWebView · Gradle

Repository: github.com/SWARAJCHATTARAJ/ARIA | Live: aria.swarajchattaraj.tech

- Architected a 5-stage agentic RAG pipeline (Plan→Retrieve→Synthesize→Audit→Publish) on LangGraph, cutting retrieval latency 63% (11s→4s) via concurrent async retrieval across 4 sources (Wikipedia, OpenAlex, arXiv, DuckDuckGo).
- Built an automated Ragas-based evaluation suite measuring faithfulness, answer relevancy, and context precision; enforced retrieval-relevance guardrail tests requiring ≥50% recall on relevant documents and ≤50% leakage on keyword-collision traps (e.g., distinguishing “Python” the language from “Burmese Python”).
- Migrated backend infrastructure from Render to a dedicated Azure VM (Ubuntu, Caddy reverse proxy, systemd service) and provisioned Azure OpenAI (GPT-4o-mini) for planning, synthesis, and verification, removing free-tier rate-limit constraints.
- Migrated a Streamlit prototype to a production React/Tailwind SPA with JWT authentication and LangGraph-managed session persistence.
- Engineered async, parallelized I/O (asyncio, aiohttp) across all retrieval sources; packaged cross-platform as a PyWebView desktop build and a Gradle/TWA Android APK; PDF export (PyMuPDF/ReportLab) benchmarked at ~15.8ms for 8 sources with 11/11 unit tests passing.

RAGPRO — Modular Retrieval-Augmented Generation Pipeline 2026

Python · LangChain · FAISS · HuggingFace · FastAPI · OpenRouter

Repository: github.com/SWARAJCHATTARAJ/RAGPRO

- Engineered a modular, multi-format (PDF/DOCX/CSV/MD) ingestion pipeline with local FAISS persistence, achieving sub-100ms query response times.
- Served the pipeline via a FastAPI REST API using local HuggingFace all-MiniLM-L6-v2 embeddings for fully local, offline-capable semantic search execution.
- Decoupled preprocessing, vectorization, and retrieval into a modular MLOps architecture; integrated OpenRouter to support 100+ interchangeable LLMs (e.g., GPT-4o-mini, Claude 3 Haiku) via a single environment-variable swap, with zero code changes to the agent pipeline.

TECHNICAL SKILLS

Programming Languages: Python, C, C++, Java, SQL

Agentic AI & Orchestration: LangGraph, LangChain, OpenAI API, Azure OpenAI, OpenRouter, HuggingFace, Multi-Agent Pipelines (Plan/Retrieve/Synthesize/Audit/Publish)

ML / AI & NLP: TensorFlow, PyTorch, Scikit-learn, XGBoost, OpenCV, BERT, Sentence-Transformers, NLTK, spaCy, Generative AI / LLM Agents, Prompt Engineering, LoRA/QLoRA Fine-Tuning

Vector DBs & Retrieval: ChromaDB, FAISS, Vector Embeddings, Semantic Search, Hybrid Retrieval

Evaluation & Guardrails: Ragas (Faithfulness, Answer Relevancy, Context Precision), automated relevance-filtering test suites

Backend & APIs: FastAPI, Uvicorn, REST API Development, JWT Authentication, asyncio, aiohttp (async parallel I/O)

Frontend & App Wrappers: React, Vite, Tailwind CSS, Streamlit (rapid prototyping), PyWebView (Desktop), Gradle/TWA (Android packaging)

Databases & Storage: Supabase, Oracle, MySQL, MongoDB, ChromaDB, FAISS, Local JSON (session storage)

Cloud, MLOps & Deployment: Azure (VM provisioning, Azure OpenAI), AWS, GCP, Render, Vercel, Docker, CI/CD, Caddy, Postman, Linux

Data & Dev Tools: Pandas, NumPy, Matplotlib, Git, GitHub, PyMuPDF, ReportLab

Core CS: Data Structures & Algorithms, System Design Fundamentals, OOP, DBMS

ADDITIONAL

Languages: English (Professional), Bengali (Native), Hindi (Conversational)

Preparing for: GATE 2026 (Computer Science / Artificial Intelligence)