Lahore, Pakistan
GitHub ↗
LinkedIn ↗
Portfolio ↗
Email ↗
Phone no : 03114086626

Muhammad Adil Usmani

SOFTWARE ENGINEER

SUMMARY

AI/ML Engineer with hands-on experience building RAG pipelines, self-correcting retrieval systems, and FastAPI-based backends. Has shipped evaluation benchmarks, threshold-based routing logic, and end-to-end LLM applications across vector and graph datastores.

SKILLS

Languages
Python · SQL

Backend & APIs
FastAPI · LangChain · LangGraph · LangSmith

AI / ML
RAG Pipelines · LLM Applications · Embeddings · Retrieval Systems · Prompt Engineering · Deep Learning

Databases
SQLite · ChromaDB · Neo4j

Tools & Infra
Github · Docker · GitHub Actions · Render · Vercel · Azure · GCP · Github copilot · Gemini CLI · Deepwiki

Libraries
Pandas · NumPy · Matplotlib

EDUCATION

BS Computer Science
University of Central Punjab
Lahore, Pakistan

Intermediate (2022)
Unique Group of Institutions

Matriculation
Lahore Grammar School

Relevant Coursework
Deep Learning · Big Data Analytics · Database Systems · DSA · Cryptography

CERTIFICATIONS

- Supervised Machine Learning: Regression & Classification — ✓ Done
- Introduction to Generative AI — ✓ Done
- ML Specialization — Stanford University-Coursera — In Progress
- LLMOps Specialization — Duke University-Coursera — In Progress

PROJECTS

AeroSphere — Air Quality Forecasting System

aerosphere.earth ↗

NASA TEMPO data · LSTM forecasting · Azure deployment

- Built LSTM model forecasting 72-hour PM2.5 across 45 cities; achieved 85%+ accuracy, 30% above baseline.
- Designed Airflow pipeline processing 1.2M+ records; added GPT-based NL summaries for public air quality reports.
- Contributed to model training, batch scheduling, and end-to-end Azure cloud deployment.

Hybrid Graph RAG vs Vector RAG — SEC 10-K (FinTech)

GitHub ↗

ChromaDB · Neo4j · Azure OpenAI · LLM-as-a-judge

- Independently designed and implemented a HybridRAGService over Apple and Tesla SEC 10-K filings using parallel retrieval from ChromaDB and Neo4j.
- Applied late fusion to merge vector and graph contexts before LLM synthesis, improving structured financial reasoning.
- Built a 53-question evaluation benchmark; graded answers across accuracy, comprehensiveness, diversity, empowerment, and directness.
- Showed hybrid retrieval matches or outperforms single-modality RAG via statistical analysis of benchmark results.

Corrective RAG — Adaptive Self-Correcting Pipeline (CRAG)

GitHub ↗

LangGraph · FAISS · Tavily web search · GPT-4o-mini

- Developed threshold-based routing: CORRECT (≥0.7) answers directly; INCORRECT (<0.3) triggers Tavily web search; AMBIGUOUS runs both in parallel.
- Incorporated content from three distinct books to enhance diversity in the RAG dataset.
- Implemented LangGraph stateful workflow with embedding caching and sentence-level document refinement.
- Achieved 3–6s latency for direct retrievals and 5–8s for web-augmented answers over geopolitical PDF documents.

News Article Summarization System

- Built a FastAPI service wrapping an T5 summarization pipeline; handled long-form input with structured JSON output for downstream consumption.

BACKEND & DEPLOYMENT

- Exposed machine learning models and RAG pipelines to frontend applications by creating RESTful API endpoints using FastAPI.
- Integrated third-party APIs (OpenAI, Tavily, Azure) and managed lightweight SQLite databases for application state.
- Containerized AI applications using basic Dockerfiles and utilized GitHub Actions to automate deployments to hosting platforms like Render and Vercel.