

SHARIF ABUSAD

AI / ML Engineer | Generative AI & LLM Applications

Delhi, India | abusharif866@gmail.com | [+91-7408289840](tel:+91-7408289840)

linkedin.com/in/sharif-abusad | github.com/Sharif-Abusad | sharif-portfolio-flax.vercel.app

PROFESSIONAL SUMMARY

AI/ML fresher with hands-on project experience building generative AI and machine learning systems, including RAG pipelines, multi-agent LLM workflows, and deep learning models for forecasting and NLP. Proficient in Python, LangChain/LangGraph, FastAPI, and vector search (FAISS, ChromaDB), with working knowledge of MLOps practices (MLflow, DVC, CI/CD) for reproducible model workflows. Seeking entry-level AI/ML Engineer or Generative AI Engineer roles.

TECHNICAL SKILLS

Programming: Python, SQL

Machine Learning: Regression, Classification, Clustering, Feature Engineering, Hyperparameter Tuning, Ensemble Learning, XGBoost, Scikit-learn

Deep Learning: TensorFlow, Keras, PyTorch, ANN, CNN, LSTM (and variants), Transformers

NLP / Generative AI: RAG, AI Agents & Agentic Workflows, Prompt Engineering, Sentence Transformers, Semantic Search, Hugging Face, Whisper

Frameworks & Libraries: LangChain, LangGraph, FastAPI, Pandas, NumPy, Matplotlib, Seaborn

Databases & Vector Stores: PostgreSQL, SQLite, FAISS, ChromaDB

Tools / MLOps: Git, GitHub, Docker, CI/CD, MLflow, DVC, AWS, Jupyter Notebook, VS Code

PROJECTS

DocMind — Intelligent RAG Chatbot | *Python, FastAPI, LangChain, LangGraph, Hugging Face, FAISS, SQLite* [GitHub](#) | [Live-Demo](#)

- Developed an enterprise-style RAG chatbot enabling natural-language querying over large document sets with grounded, context-aware answers.
- Implemented semantic retrieval using FAISS vector search and Hugging Face embeddings to improve passage relevance over keyword search.
- Architected multi-turn conversational memory with LangGraph, maintaining context across extended interactions.
- Built a modular FastAPI backend (ingestion, retrieval, chat endpoints) with a Streamlit front end supporting streaming responses and tool-calling.

Meeting Intelligence — AI Video Assistant | *Python, Whisper, ChromaDB, LangChain, FastAPI* [GitHub](#)

- Built an AI assistant that transcribes YouTube videos and meeting recordings into structured, searchable knowledge using OpenAI Whisper.
- Designed LangChain pipelines to generate meeting summaries and extract action items, decisions, and open questions from transcripts.
- Implemented RAG-based conversational Q&A over transcripts using ChromaDB for semantic search.
- Developed modular FastAPI services separating ingestion, transcription, and retrieval workflows.

AI Blog Writing Agent | *LangGraph, LangChain, FastAPI, LLMs* [GitHub](#)

- Designed a multi-agent content pipeline automating research, drafting, SEO optimization, and editorial review.
- Implemented graph-based agent workflows in LangGraph with dedicated research, writing, and review agents.
- Automated keyword research and on-page SEO recommendations to improve content discoverability.
- Configured the pipeline to output publication-ready Markdown articles, reducing manual editing effort.

AtmoSense — AI Weather Forecasting System | *TensorFlow, XGBoost, FastAPI, Open-Meteo API* [GitHub](#) | [Live-Demo](#)

- Built a hybrid forecasting platform combining an XGBoost classifier (rainfall likelihood) and a Bidirectional LSTM (temperature trends).
- Integrated live data from the Open-Meteo API with automatic location detection for localized forecasts.
- Delivered hourly and weekly forecast views through a FastAPI-backed dashboard.

CineMatch — Movie Recommendation System | *Python, Streamlit, FastAPI, Scikit-learn, TMDB API* [GitHub](#) | [Live-Demo](#)

- Built a content-based recommendation engine using TF-IDF vectorization and cosine similarity over plot, genre, and metadata.
- Integrated the TMDB API to enrich recommendations with posters, ratings, and metadata.
- Developed a Streamlit interface backed by a FastAPI service for real-time recommendation retrieval.

Quora Duplicate Question Pair Detection | *Python, NLP, Scikit-learn* [GitHub](#) | [Live-Demo](#)

- Built an NLP classification model to detect semantically duplicate question pairs.
- Engineered text-similarity features (TF-IDF, token overlap, fuzzy-matching statistics) to strengthen classifier signal.
- Compared multiple algorithms and refined preprocessing to improve duplicate-detection performance.

EDUCATION

Master of Computer Applications (Specialization in AI/ML) — Pursuing

Lovely Professional University

Bachelor of Computer Applications — CGPA 8.8

Shibli National College | 2023 – 2026