+880 1321073452
adilshamim696@gmail.com

ADIL SHAMIM

AI Engineer

Portfolio: adilshamim.me
LinkedIn: adilshamim8
GitHub: AdilShamim8

AI Engineer with 2+ years building and shipping production LLM systems, RAG pipelines, and agentic workflows. Published 1st-author conference paper achieving 56% inference speedup (BUET CSE Fest 2026). Kaggle Master, Top 1% globally out of 4,082 competitors. Founded Toolly — a live AI tools platform with 500+ tools, built and maintained solo. Comfortable across the full AI stack: multi-provider LLM orchestration, vector databases, agent architecture, Dockerized deployment, and production monitoring.

SKILLS

Agentic AI Engineering: LangChain · LangGraph · Tool-calling agents · MCP (Model Context Protocol) · Memory & state management · Failure-mode-aware agent design · Qdrant vector store

LLM & Generative AI: Multi-provider orchestration · RAG pipeline design · Retrieval quality evaluation · Context window management · Prompt engineering at production scale

ML/DL & Speech AI: PyTorch · TensorFlow · HuggingFace Transformers · Fine-tuning & domain adaptation · Whisper · wav2vec2 · pyannote.audio · WeSpeaker · XGBoost · LightGBM · scikit-learn

MLOps & Deployment: Docker · MLflow · ZenML · GitHub Actions · REST APIs · CI/CD · A/B testing pipelines · Systems Design

Programming & Frameworks: Python · JavaScript (functional) · SQL · FastAPI · Flask · Streamlit

PUBLICATION

Bangla Diarizz: Domain-Adapted Speaker Diarization for Bengali Long-Form Audio | 1st Author

BUET CSE Fest 2026 — DL Sprint 4.0 Bengali Speaker Diarization Competition · PyAnnote · WeSpeaker · wav2vec2 · HuggingFace Transformers

- Fine-tuned segmentation model on competition dataset; replaced speaker embeddings with WeSpeaker ResNet34-LM for targeted domain adaptation over language-agnostic baselines.
- Achieved 56% wall-clock speed improvement—reduced inference time from 1h 22m to ~36m across 14 test audio files. Competitive DER on Bengali-Loop benchmark.

EXPERIENCE

Founder & AI Engineer — **Toolly** · toolly.tech — Jun 2025 — Present

- Designed and shipped toolly.tech — a live AI tools directory with 500+ tools, 15 categories, a community submission pipeline, and an integrated Learn AI hub; all built and maintained solo.
- Engineered the full production stack: frontend, tool submission and moderation system, search and filter logic, and usage analytics from scratch.
- Built and deployed Toolly Studio — a Streamlit + Bria AI image generation app with batch export, Docker packaging, and a one-command demo flow for non-technical users.

PROJECTS

Production-Grade RAG Pipeline | GitHub | Production GenAI · LangChain · Qdrant · FastAPI · Inngest · OpenAI · Multi-provider LLM + embeddings · Durable workflows · Local-first resilience

- PDF upload → recursive chunking → multi-provider embeddings (OpenAI, Gemini, Ollama, local) → Qdrant vector store with deterministic IDs for idempotent re-ingestion → top-K retrieval → source-aware LLM generation with grounded, auditable answers. Architected a full production RAG pipeline:
- Production-ready RAG API with FastAPI, LangChain, OpenAI embeddings, and Qdrant vector store — covering chunking strategy, retrieval pipeline, and structured API response contracts.
- Implements query routing, context window management, and source attribution; designed for deployment behind a real inference endpoint with documented latency characteristics.

QuantScope — Global Quantitative Stock Analysis Platform | GitHub | Python · FastAPI · LangChain · Docker · pytest

35+ exchanges · 6 LLM providers · 33 tests ·Full monitoring stack

- Enforced strict architectural separation: core/ (indicators, risk, data models) has zero imports from llm/ or api/ — business logic is fully testable in CI without a running server, live API key, or LLM provider, eliminating test environment fragility.
- Engineered a 6-provider LLM fallback chain (OpenAI → Anthropic → Google → Ollama → Mistral → Cohere) with automatic provider switching on failure and template-based static fallback—the system degrades gracefully through all six providers before ever returning an error to the user.

Training Data Bot — Automated LLM Fine-Tuning Dataset Pipeline | GitHub | LLM Engineering · Fine-Tuning Pipeline · PDF / URL Ingestion · Quality Scoring

- Designed to solve the data curation bottleneck in LLM fine-tuning: ingests raw documents (PDF, plain text, URLs) → applies multi-signal quality scoring (length, deduplication, coherence) → outputs structured datasets formatted for direct use in fine-tuning runs — no manual curation step required.
- *Demonstrates full* ML engineering lifecycle thinking from raw data ingestion through training-ready structured output—the pipeline a production ML team would build before starting a fine-tuning project, not after.

EDUCATION

Computer Science & Engineering — BNIST — Feb 2023 — Present

- **Relevant coursework:** Linear Algebra · Calculus · Probability & Statistics · Data Structures & Algorithms · Operating Systems · Database Systems · System Design Projects · Artificial Intelligence · Machine Learning · Data Science

Languages: English (Fluent) · Bengali (Native) · Hindi (Conversational)