Arpit Chauhan

Applied AI Engineer

✉ arpitchauhanofficial@gmail.com ☎ 9082784190 Mumbai, Maharashtra

portfolio.arpitdev.site linkedin.com/in/arpit-chauhan-3b0885250 github.com/botARPIT

Profile

Applied AI Engineer building LLM systems that handle failure gracefully. Focused on multi-agent workflows, RAG pipelines, and stateful agents under real-world constraints (token limits, latency, cost). Strong systems background across backend, edge, and distributed environments, enabling reliable AI integration beyond demos.

Technical Skills

Applied AI & LLM Systems

- LLM agents (LangGraph, Google ADK)
- Retrieval-Augmented Generation (RAG)
- Tool calling & agent orchestration
- Conversation memory & state handling
- LLM evaluation & hallucination analysis
- Vector databases (Chroma, FAISS)

Backend & Infrastructure

- Python, TypeScript, Node.js
- Cloudflare Workers, Express, Hono
- PostgreSQL, MongoDB, Redis
- Observability (Prometheus, logging)

Projects

Multi-Agent LLM System, *Tech Stack: Google ADK, Python, Gemini 2.5 Flash Lite*

- Added session persistence and **event compaction** to manage long-running conversations under token limits.
- Designed multi-agent workflows **(Sequential, Conditional)** for automated blog writing, fact-checking, and editorial review.
- Implemented tool-augmented agents using Google Search and custom functions with **retry logic** and failure handling.
- Optimized for **latency and cost**, selecting Flash Lite models and streaming outputs for multi-stage pipelines.

LLM Architecture & RAG Systems, *LangChain, FAISS, Chroma, Mistral, Qwen (Ollama)*

- Compared FAISS vs Chroma trade-offs (latency, persistence, scalability) and documented redesign paths (**hybrid retrieval, reranking**).
- Built **RAG** pipelines for YouTube transcript Q&A and document-based assistants using semantic retrieval.
- Implemented **parallel and conditional** chains to reduce latency and improve response relevance.

AgenticAI Workflows, *Teck Stack: LangGraph, LangGraph, Python, Ollama (Qwen 1.7B)*

- Built graph-based LLM workflows supporting **parallel** evaluation, **conditional** routing, and **iterative** refinement loops.
- Designed typed state schemas with checkpointing for fault tolerance and debuggability.
- Implemented iteration limits and structured outputs to control **non-deterministic** LLM behavior.

Blogify: Edge-Native Blogging Platform, *Tech Stack: Hono, Cloudflare Workers, PostgreSQL, Prisma Accelerate*

- Built a globally distributed backend on **Cloudflare Workers** under strict edge constraints.
- Integrated PostgreSQL via **Prisma Accelerate** for HTTP-based access at the edge.
- Reduced authentication CPU cost using native Web Crypto APIs (**~2-3x** faster sign-in).
- Enabled sub-minute global deployments via **CI/CD**.

E-Library REST API: Scalable Backend with Observability,

Tech Stack: Node.js, Express, MongoDB, Redis, Cloudinary, Prometheus, Docker, AWS EC2

- Added **Prometheus** metrics and structured logging for observability.
- Built an authenticated content platform with Redis **caching** and large file uploads (PDFs up to 10 MB).
- **Documented** scalability limits and remediation paths.

Education

B.E. Computer Engineering, Mumbai University Mumbai, India