# TRUPTI PANDYA

AI Engineer | LLM & GenAI Applications | Agentic Workflows | Software Engineering
Leicester, UK | pandyatrupti531@gmail.com | truptipandya.dev | LinkedIn | GitHub

## Professional Profile

AI Engineer with 6 months of production experience at Zenithive building and deploying LLM-powered products for enterprise clients. MSc Cloud Computing graduate (Merit, University of Leicester). Designed and shipped four end-to-end GenAI applications across recruitment, healthcare, career coaching, and digital forensics — from prompt architecture and RAG pipeline design through to cloud deployment. Holds a Databricks Generative AI Fundamentals certification. Strong foundation in Python, TypeScript, and cloud infrastructure.

## Key Skills

**LLM & Prompt Engineering:** System prompt design, chain-of-thought / few-shot / zero-shot prompting, context window optimisation, structured output schemas, prompt evaluation (DeepEval)

**Agentic AI & Orchestration:** LangChain, LlamaIndex, Claude API (Anthropic), OpenAI GPT family, Gemini API, Ollama, HuggingFace, multi-step agentic pipelines, RAG architecture

**Dev & Infrastructure:** Python, TypeScript, Next.js, FastAPI, Streamlit, Docker, AWS (ECS, Lambda), Azure, Vercel, Railway, GitHub Actions CI/CD

**Vector Search:** Pinecone, Chroma, FAISS, Supabase, sentence-transformers, hybrid retrieval, structured metadata filtering

**AI Dev Tools:** Cursor, Claude Code (Anthropic), v0.dev, GitHub Copilot, Midjourney, DALL-E, Sora, Google Veo 3, MCP server integration

**Certification:** Databricks Academy — Generative AI Fundamentals

## Work Experience

**Junior AI Engineer (Remote)** | **Zenithive System Private Limited**
Oct 2025 – Apr 2026 | 6 months, full-time remote contract | Ahmedabad, India (Remote)

- Built and maintained production RAG pipelines (LangChain, Pinecone) for a multi-tenant document Q&A product — processing **1M+ tokens per ingestion run** across .pdf, .docx, .txt, and .json files with 512-token overlapping chunks; iterated system prompts for GPT-4o and Claude 3.5 Sonnet, improving answer accuracy by **18%** via DeepEval benchmarks across a labelled test set of 200+ queries.
- Engineered FastAPI LLM inference endpoints (auth, rate limiting, streaming) consumed by a Next.js frontend; reduced mean response latency by **~40%** through async processing and context window optimisation; deployed to AWS ECS and Lambda via Docker and GitHub Actions CI/CD.
- Integrated FAISS and Chroma vector search with metadata filtering for hybrid retrieval; tuned cosine similarity thresholds to cut low-confidence retrievals by **~30%**; achieved **74% pytest coverage** across AI service modules.
- Used Cursor and Claude Code for autonomous refactoring of legacy prompt chains; used v0.dev to prototype dashboard UI components, cutting build time by **~35%**; communicated all progress and decisions asynchronously via Slack, Notion, and GitHub PRs.

**UI/UX Intern** | **LinearLoop Private Limited**
Jun 2023 – Aug 2023 | 3 months, Internship | Ahmedabad, India (On-site)

- Designed client-facing UIs in Figma, Photoshop, and Illustrator, building product instincts that now inform how AI-generated interfaces are evaluated and refined into production-ready tools.
- Collaborated cross-functionally with engineers to translate requirements into working specifications — the same discipline applied when scoping and architecting AI prototype solutions.

## Projects

**AI Forensic Investigation Assistant — GenAI Evidence Analysis Platform** · Python, FastAPI, Streamlit, FAISS, sentence-transformers, OpenAI 2026

- RAG-powered platform for digital forensics investigators — ingests heterogeneous evidence (.log, .txt, .pdf, .json), reconstructs event timelines across multiple timestamp formats (ISO 8601, syslog, Unix epoch), detects anomalies, and generates structured investigation reports from natural-language queries.

- Built per-case isolated FAISS indexes (up to 50,000 vectors per case) with local sentence-transformer embeddings to eliminate cross-case data leakage; hybrid detection layer combines regex timestamp extraction, rule-based keyword scanning with explainable reasons, and statistical anomaly detection (event-rate spikes, hot IPs/users).
- System prompts constrain the LLM strictly to retrieved context with source citations; graceful extractive fallback when no API key is configured ensures the tool remains auditable in air-gapped environments; FastAPI backend (/upload, /chat, /timeline, /findings, /report) consumed by a Streamlit UI with per-case conversation memory.

**NurseChat — Medical Screening AI Chatbot** · Python, TypeScript, RAG, LLM Orchestration, Next.js, Supabase 2025

- End-to-end AI chatbot for nurse-led patient screening — structured question flows, ward/department recommendation engine, and anomaly detection to flag responses requiring clinical attention.
- Designed and implemented a full RAG pipeline (document ingestion, embedding, vector retrieval, context injection) enabling the chatbot to answer hospital-specific queries accurately across dynamic, multi-department knowledge bases.
- Built a nurse-facing oversight dashboard displaying chatbot decisions and patient details; human-in-the-loop architecture as a first-class design concern — anomaly flags route to senior staff review rather than autonomous escalation.

**Career Copilot — Multi-Turn Agentic AI Coaching Platform** · OpenAI, Gemini APIs, FastAPI, TypeScript, Next.js 2025

- Multi-turn agentic conversational system with persistent session context, turn-selection logic, and hierarchical context inclusion — ensuring coherent, on-topic agent responses across long interactions.
- System-level instructions constrain agent persona, response format, and topic scope; human-review loop flags low-confidence outputs before reaching the user; Ollama-served local models used for rapid, cost-free prompt iteration during prototyping before switching to production APIs.
- Feedback loop collects per-turn quality signals to continuously improve system prompts; designed with explicit separation between orchestration logic and LLM calls to keep the agentic pipeline testable and auditable.

**HireAI — Agentic LLM Recruitment Assistant (MSc Dissertation)** · Python, LangChain, Pinecone/FAISS, Supabase 2025

- Engineered a full agentic prompt architecture from scratch — system prompt design, context injection strategy, RAG pipeline (parsing → chunking → embedding → vector index), structured output schemas, and multi-stage prompt chains where each step (retrieval, ranking, synthesis, validation) has a defined role with explicit guardrails.
- Built a document ingestion pipeline with similarity thresholding to prevent low-confidence retrievals; implemented a systematic DeepEval-style evaluation loop benchmarking prompt variants against retrieval precision and answer quality metrics — iterating by data rather than intuition.
- Designed human oversight as a first-class concern: low-confidence agent outputs are flagged for review rather than surfaced directly; structured output schemas enforce consistent, auditable response formats across all agent steps.

## Education

**MSc Cloud Computing** | **University of Leicester, UK**
Sep 2024 – Jan 2026 | Merit | Dissertation: HireAI — Agentic LLM recruitment assistant with RAG and prompt engineering

**BEng Information Technology** | **Kadi Sarva Vishwavidyalaya (KSV), India**
Jun 2020 – May 2024

## Additional Information

- Broad hands-on AI tool stack spanning code generation, UI prototyping, and image/video generation (Midjourney, DALL-E, Sora, Google Veo 3)
- Builds AI agents end-to-end — from prompt design and RAG pipeline architecture through to cloud deployment and evaluation
- Shipped four production AI applications alongside 6 months of commercial AI engineering; fast, iterative experimenter
- Right to work in the UK (visa valid until February 2028) | References available upon request