DHANESH VASHISTH

Agentic AI Engineer | LangGraph · Multi-Agent Systems · RAG · MCP · FastAPI | Python

Hisar, Haryana, India | contact@dhaneshvashisth.com | +91 9813558866 | dhaneshvashisth.com

https://github.com/dhaneshvashisth | https://linkedin.com/in/dhaneshvashisth

PROFESSIONAL SUMMARY

Agentic AI Engineer with 5+ years of experience designing and deploying production-grade autonomous AI systems using Python. Expert in LangGraph multi-agent orchestration, RAG architectures, agent reasoning and reflection patterns, and MCP-aligned tool interoperability. Delivered agentic AI solutions for a US based client including self-correcting NL-to-SQL pipelines (planner → executor → validator pattern) that reduced manual analytics workload by 70%, and Redis semantic cache systems with role-scoped guardrails that cut repeated LLM cost by 40%. Experienced in A2A agent coordination, structured Pydantic output validation, short-term and long-term agent memory, RBAC-enforced tool access, and containerized multi-agent deployments. Focused on building reliable, observable, and guardrailed agentic systems at enterprise scale.

TECHNICAL SKILLS

AI Engineering : LangGraph, LangChain, RAG Pipelines, Multi-Agent Systems (A2A), Agent Orchestration, MCP (Model Context Protocol), Structured Output Validation, Agent Memory (short-term/long-term/episodic), Guardrails, OpenAI APIs, Prompt Engineering, RAGAS

Backend: FastAPI, Django, REST APIs, AsyncIO, asyncpg, Pydantic

Databases : PostgreSQL, Redis (semantic cache), Qdrant

Distributed Systems : Kafka, Zookeeper

DevOps / Infra: Docker, Docker Compose, GIT, GitHub, AWS EC2 CI/CD

Security: OAuth2, JWT, RBAC, API Key Auth, multi-tenant isolation patterns

PROFESSIONAL EXPERIENCE

AI Engineer (Direct Hire) | Excel Texas Wireless, TX, USA (Remote) *Nov 2025 – Present*

- Directly hired after 3.5 years of successful delivery as a consultant on their project — converted from consulting engagement to full-time AI engineering role
- Built a self-correcting agentic NL→SQL system using LangGraph (planner → executor → validator) with agent reflection via validation-retry loop, reducing manual analytics query effort by 70%
- Implemented structured tool access (MCP-aligned) via Qdrant embeddings for schema-aware retrieval, ensuring only relevant tables were passed to the LLM, improving accuracy and reducing token cost.
- Role-based access control enforced at SQL level — WHERE clauses injected programmatically from JWT role before execution, bypass-proof at API level
- Designed Redis semantic caching with role-scoped guardrail keys, reducing repeated LLM calls by ~40% while enforcing cross-user data isolation as a hard agent safety constraint.

Software Consultant | Rahul Tech Services (Technology Consultancy) *May 2021 – Oct 2025*

Client: Excel Texas Wireless, TX, USA

- Delivered 5 production systems across telecom operations, dealer platforms, automation pipelines, and admin portals for a US client over 3.5 years.
- Designed high-performance PostgreSQL schemas for concurrent dealer operations with sub-100ms query response times.
- Automated telecom refill workflows end-to-end, reducing manual processing time from 3–5 minutes per transaction to near-zero.
- Implemented OAuth2 + JWT with multi-level RBAC (Admin, Supervisor, Agent) for gaming platforms and partner APIs, enabling secure access, scoped dashboards, and audit trails.

PROJECTS

NL-to-SQL RAG Pipeline - github.com/dhaneshvashisth/NL-SQL-RAG-pipeline

Stack: LangGraph · FastAPI · Qdrant · Redis · PostgreSQL · GPT-4o-mini · Streamlit · Docker

- Designed 5-node agentic LangGraph workflow (planner → schema retriever → SQL generator → validator → executor) implementing agent reflection via self-correcting retry loop; reduced manual query requests by 70%.
- Implemented Qdrant schema retrieval as structured tool access (MCP-aligned pattern) so only semantically relevant tables were passed to the LLM, improving accuracy and reducing token cost.
- Enforced RBAC at SQL level using JWT role-aware query filtering across admin, supervisor, and agent roles.
- Designed Redis semantic cache with role-scoped guardrails, reducing repeated LLM calls by ~40% while enforcing cross-user data isolation as a safety constraint.

Multi-Tenant E-Commerce RAG System - github.com/dhaneshvashisth/multi-tenant-ecommerce-rag

Stack: LangGraph FastAPI Qdrant Kafka Redis PostgreSQL FlashRank RAGAS Streamlit Docker

- Built a 9-container production agentic RAG platform with event-driven agent coordination via Kafka, short-term conversation memory via Redis, and RAGAS-based agent evaluation for output quality monitoring.
- Designed tenant-isolated retrieval architecture for Amazon, Flipkart, and Myntra with secure API-key access.
- Implemented dual-layer agent memory: Redis semantic cache (short-term, cosine similarity) + conversation history store (episodic), reducing repeated query latency by 70%.
- Built feedback-driven prompt optimization loop: Kafka consumers ingest user ratings, evaluator agent scores responses, optimizer agent rewrites prompts and version-switches automatically — a full agentic self-correction cycle.

Multi-Agent Code Review System - github.com/dhaneshvashisth/code-review-agent

Stack: LangGraph FastAPI PostgreSQL asyncpg GPT-4o-mini Docker SHA256 Cache

- Designed A2A multi-agent orchestration: planner/orchestrator delegates tasks in parallel to 3 specialized agents (Security Checker, Bug Detector, Quality Checker), each returning Pydantic-validated structured outputs — reducing turnaround time by 66%.
- Enforced agent guardrails via temperature=0 determinism, Annotated reducers for state validation, and structured exception handling to prevent partial or malformed agent outputs reaching the API.
- Implemented SHA256-based agent memory in PostgreSQL to deduplicate identical submissions, eliminating redundant LLM calls and enforcing consistent outputs for repeated inputs.
- Added agent observability features: request tracing middleware, structured logging, paginated audit history, and retry policies for reliable production agent execution.

EDUCATION

Master of Computer Applications (MCA)
Guru Jambheshwar University of Science and Technology | 2014 – 2016

Bachelor of Computer Applications (BCA)
Kurukshetra University | 2011 – 2014