Devraj Gadhvi

Applied AI & AI Automation Engineer

Gujarat, India (UTC+5:30) · Open to remote worldwide · Immediate start
hello@dev4aibots.com · linkedin.com/in/devraj0770 · github.com/dev4aibots · dev4aibots.com
Independent AI engineer, Dev4AIBots (June 2026 – present) — Udyam-registered studio (UDYAM-GJ-29-0019103)

Summary

I build AI systems you can verify — not just demo. Permission-aware RAG pipelines, n8n + FastAPI business automations, LiteLLM model routing, and tool-using AI agents. Every claim backed by runnable code, reproducible benchmarks, and open-source repos with CI. Every number in my READMEs links to a command that produces it, or is explicitly labelled as unverified.

Flagship Systems

Production RAG System — Applied AI

Python · FastAPI · Hybrid retrieval · Eval regression gate · github.com/dev4aibots/production-rag-system

Production-grade RAG with verified citations, hybrid retrieval (dense + BM25 + RRF), cross-encoder reranking, and a McNemar eval regression gate that fails CI on real drops. 13 tests, all passing. Every answer chained to verbatim source quotes; unsupported claims return insufficient_evidence.

AI Agent System — Agents & Automation

Python · FastAPI · Tool use · Human-in-the-loop · github.com/dev4aibots/ai-agent-system

Tool-using agent with allow-listed tools, exponential backoff retries, idempotency keys (duplicate requests never double-execute), human-in-the-loop approval on write actions, max-step loop guard, and hash-chained audit ledger. 14 tests, all passing.

Multi-Tenant AI Employee Platform — AI Automation

n8n + FastAPI + LiteLLM + Postgres RLS · Solo · github.com/dev4aibots/ai-automation-platform

Six n8n workflows with a FastAPI microservice for structured extraction, pgvector RAG, and OWASP prompt-injection screening. Row-level security at the database layer, idempotent webhooks, human-in-the-loop gates, synthetic replay harness.

PRAMAAN — Zero-Trust RAG

Next.js 14 + PGlite/pgvector + Transformers.js · Team Madmax hackathon + solo v2 rebuild · github.com/dev4aibots/Pramaan

5-layer security pipeline: ingestion scan, retrieval-time authorization inside the vector query, evidence-grounded synthesis, egress PII sanitizer, tamper-evident audit ledger. Live demo at pramaan-dev4aibots.vercel.app.

Semantic Code Search — Applied AI

AST chunking + RRF hybrid fusion · github.com/dev4aibots/semantic-code-search

Repository-scale search mapping natural language to AST nodes. Live API at semantic-code-search-demo.vercel.app.

LLM Gateway Benchmarks — LLM Eval & Routing

191 models, 5 providers · Python · github.com/dev4aibots/llm-benchmarks

Measured latency and success for 191 models across OpenRouter, Groq, Cerebras, Gemini, and NVIDIA NIM. Used to build a LiteLLM routing table with provider failover.

What I Work With

Retrieval: Hybrid dense + lexical search, RRF fusion, AST-aware chunking, pgvector with filtered HNSW, per-row authorization inside the query

Agents & Automation: n8n multi-tenant workflows, LiteLLM routing with failover, MCP tool servers, human-in-the-loop approval gates, idempotency keys, dead-letter paths

Security: Row-level security, envelope encryption, HMAC verification with replay windows, prompt-injection screening, PII redaction, hash-chained audit logs

Evaluation: Rule graders with explicit expectations, cost and latency per run, reproducible benchmarks, McNemar regression gates

Delivery: TypeScript/Next.js, Python/FastAPI, PostgreSQL, Docker, GitHub Actions CI, Vercel

Credentials

18 vendor-verified credentials across Microsoft, Google, IBM, DeepLearning.AI, Vanderbilt, UiPath, Edureka, and JetBrains. No prior paid employment claimed — engineering work is independent work at Dev4AIBots since June 2026. All claims verifiable at dev4aibots.com.

Honesty contract: deterministic interface prototypes are labelled as simulations. Team work (PRAMAAN / Team Madmax) is labelled as team work. Every README number links to a runnable command or explicit provenance label.