YASH RAJ PANDEY

yashpn62@gmail.com | linkedin.com/in/yashrajpandeyy | github.com/devYRPauli | yashrajpandey.com

Summary

AI engineer specializing in local-first LLM systems: agent runtimes, RAG and tool-calling platforms, and the production evaluation frameworks that keep them honest. I benchmark frontier and open-weight models, deploy self-hosted inference on-premise, and gate every release behind regression evals. Promoted from Software Engineer to AI Agents Architect in 13 months; an active open-source contributor on npm and GitHub.

Experience

University of Florida **Gainesville, FL**

AI Agents Architect *Apr 2026 – Present*

- Re-platforming inference onto institutional GPU infrastructure, self-hosting a ∼745B-parameter open-weight MoE (FP8, 1M-token context) to cut projected 5-year TCO ∼60% versus commercial APIs and per-query cost to ∼$0.01, fully on-premise.
- Architecting and deploying a production AI agent platform pairing RAG over a 10,000+ document technical corpus with an agentic tool-calling runtime that executes multi-step workflows autonomously, translating researchers' scientific workflows into natural-language access to domain data and analysis.
- Building a model-selection framework benchmarking 14 LLMs across 26 task-specific evaluations and 5 capability dimensions, identifying the top performer (88% multi-tool orchestration) and flagging hallucination failure modes before deployment.
- Building and maintaining a 24/24 regression eval suite that gates every model and pipeline change for zero-regression backend swaps, and mentoring engineers on the evaluation methodology behind it.
- Hardening the data foundation behind the platform: migrating a 50,000+ record corpus into a typed analytical store (DuckDB), auditing 1.4M+ cells with zero mismatches and surfacing 2,000+ hidden ID conflicts before lock-in.

University of Florida **Gainesville, FL**

Lead Software Engineer *Oct 2025 – Apr 2026*

- Led and scaled a genomics data platform (Django, React, PostgreSQL) to 5.38M+ records across 32 data models, 58 endpoints, and 35 indexes; it became the system of record for 30+ researchers across 5 labs (+40% daily active users).
- Tuned PostgreSQL indexing and caching to hold trait lookups at 23–29 ms and analytics aggregations at ∼222 ms over ∼10K records, with BLASTN genome searches at ∼602 ms on a 518 MB index.
- Shipped 7 ingestion and validation pipelines with automated R Markdown reporting (preprocessing from hours to ∼2 minutes); deployed on GCP (Kubernetes, Terraform, Docker) with JWT/RBAC, cut frontend load time from 8s to 3s, and grew the team from one to three.

Software Engineer *Mar 2025 – Oct 2025*

- Shaped the platform's original architecture: prototyped the first PostgreSQL schemas and Python ingestion, and wrote the validation checks and test fixtures the production system kept.

Open-Source Projects

Looma / *Local-first memory for coding agents* *github.com/devYRPauli/looma*

- Designed and built an MIT-licensed CLI that converts Claude Code, Codex, and Cursor transcripts into structured work items and resumable, token-budgeted context packs, on the Python standard library alone (SQLite + FTS5, zero runtime dependencies) with 134 passing tests.
- Designed and benchmarked the extraction pipeline to F1 0.86 (0.94 precision) with a dependency-free heuristic and an optional local-LLM extractor reaching 0.95; optimized the ingest daemon ∼27x and improved retrieval ranking (MRR +49%).

mddocs / *Git-native collaborative Markdown with an agent API* *github.com/devYRPauli/mddocs*

- Built and published (npm, MIT) a local-first, git-native collaborative Markdown editor combining a CLI, real-time multiplayer (Yjs/Hocuspocus CRDT), and a token-gated HTTP API that lets AI agents read, comment, suggest, and rewrite documents.
- Designed a token-authenticated agent API with an SSE event stream (backlog replay via Last-Event-ID, heartbeats) and standards-compliant rate-limit headers; shipped 8 semver releases behind a fresh-clone verification gate, and cut package size 87% (31.6 to 1.5 MB) with esbuild.

Technical Stack

AI & ML:	vLLM, Ollama, llama.cpp, MLX, RAG, Vector Search, Agent Runtimes, Tool-Calling, Model Benchmarking & Evals, Self-Hosted MoE Inference (FP8), Quantization, KV-Cache, Reranking, Embeddings, MCP, LangChain, LangGraph, LlamaIndex, OpenAI & Anthropic APIs
Languages:	Python, JavaScript, TypeScript, SQL, Bash, C/C++
Frameworks:	Django, FastAPI, React, Next.js, Node.js
Databases:	PostgreSQL, DuckDB, Qdrant, ChromaDB, Redis, SQLite
Infrastructure:	Docker, Kubernetes, Terraform, GCP, AWS, Linux, GPU / Self-Hosted Inference, Apple Silicon / MLX, Git, GitHub Actions, JWT / RBAC

Education

University of Florida **Aug 2023 – Dec 2024**

M.S. Computer & Information Science & Engineering *Gainesville, FL*

Jaypee University of Engineering and Technology **Jul 2019 – May 2023**

B.Tech. in Computer Science and Engineering *Guna, India*