Yashwanth Reddy Katipally

San Jose, CA | 669-340-9742 | yashwanthreddykatipally@gmail.com
linkedin.com/in/yashwanth-katipally | github.com/katipally | yashwanth.net

Summary

AI Engineer dedicated to shipping robust, end-to-end multi-agent systems. Tech-agnostic and highly adaptable, with a strong focus on core problem-solving over framework loyalty. Deep technical fluency in building custom agent loops, structured tool calling, MCP server and client integrations, and context compaction. Quick to adopt new technologies, deploying AI products across any surface, including the terminal, cloud, macOS, iOS, and Electron desktop.

Education

San Jose State University — San Jose, CA
M.S. Applied Data Intelligence — *Aug 2024 – May 2026*

- Coursework: Generative AI, Machine Learning, Deep Learning, Distributed Systems

Presidency University — Bangalore, India
B.Tech. Computer Engineering, AI/ML Specialization — *Aug 2020 – May 2024*

Experience

Datasaur — San Francisco, CA
Software Engineering Intern — *Jun 2025 – Dec 2025*

- **Workforce AI Agent:** Built and shipped, alongside a small team, a conversational workspace agent (LangGraph, FastAPI, React 18) that unified Slack, Gmail, Notion, and Calendar into one interface used daily by three internal teams. Owned the tool registry and OAuth 2.0 scope design. Median task time dropped from **8 minutes to under 1** across a **4-month production pilot**.
- Built a hybrid retrieval pipeline pairing **pgvector** dense embeddings with **BM25** sparse scoring over a **5,000-document** internal knowledge base. Search that finds what you mean, not just what you said. Recall@5 moved from **0.72 baseline to 0.89** on a **200-query** offline evaluation set.
- Deployed to AWS on EC2 and RDS with Docker and CI/CD. Hard timeouts, fallback LLM routing, and structured observability hold p95 inference under **1.2s** and the daily token bill under **five dollars** at production load.
- **Contact Consolidation Tool:** Co-designed a 6-agent LangChain and Ollama pipeline for column mapping, deduplication, and email enrichment, orchestrated through a fault-tolerant DAG with async workers and exponential retries. Four hours of manual CRM prep became **45 minutes**.
- An LLM-as-a-judge agent evaluation harness, built with the team, caught extraction hallucinations before they shipped and held **98 percent accuracy** across **50+ production batch runs**.

Projects

WOS (WorkOS) | Local-First AI Agent Desktop — *2026*

- *Electron, React 19, TypeScript, SQLite, MCP, Anthropic / OpenAI SDKs*
- Built a custom agent harness, the same while-loop-based ReAct pattern that powers Claude Code. Picked a custom loop over LangGraph because the framework was too limited and restrictive, and not flexible enough for the looping, branching agent behavior the product needed.
- Integrated Slack, GitHub, Jira, and Google Workspace through a typed tool surface with **AES-256** credential storage, **OAuth 2.0** flows, and per-app snapshot caching. The agent pulls the right context into the right conversation.
- Shipped an LLM-agnostic provider layer with per-conversation model selection, streaming token delivery, a real-time intent classifier that filters the tool surface before each turn, lightweight agent evaluations per turn, a **five-stage context-compaction pipeline**, and a persistent SQLite-backed memory service that recalls facts across sessions. The whole stack runs locally, with no cloud dependency, no telemetry, and no accounts.

JARVIS (J.A.R.V.I.S) | macOS Notch AI Assistant — *2026*

- *Swift 6.2, Python 3.12, FastAPI, LangGraph, DeepAgents, MCP, Ollama*
- Designed and built a privacy-first AI assistant that lives inside the Mac notch, with a four-state SwiftUI overlay (closed, open, expanded, working) and a hybrid Swift 6.2 plus Python 3.12 FastAPI backend communicating over HTTP and SSE. Real-time thinking tokens, tool calls, and agent progress stream straight into the native UI.
- Designed the agent architecture around **LangGraph** and **DeepAgents** with an async SQLite checkpointer for per-conversation isolation and fault-tolerant mid-run resumption. **75+ tools** spanning file ops, shell, git, macOS automation, system app control, and OCR. A LangGraph interrupt gate pauses the agent before anything destructive, so the user approves every risky move.
- Local inference by default through **Ollama**, with OpenAI, Anthropic, and Google Gemini as opt-in fallbacks. Supports thinking models (Qwen3, DeepSeek-R1) with configurable reasoning budgets. The MCP layer ships **three built-in servers** and converts any MCP JSON schema into a typed LangChain StructuredTool at runtime.

Friday (friday-code) | Terminal AI Coding Agent *2026*

- *TypeScript, Bun, Ink/React, MCP, 7-provider LLM SDKs*
- Built a hand-rolled agent harness from scratch, structured as a while-loop ReAct pattern. The loop calls the LLM, parses the tool calls, executes them, feeds results back, and repeats until the model emits a stop signal. LangGraph was too heavy for a terminal-first workflow, so the harness was a custom build from day one.
- **47 typed tools**, exposed through a single Zod-validated registry, give the agent full read, write, and execute reach over the developer's environment. The ReAct loop logs every step (thought, action, observation) and gates any state-mutating call behind an explicit confirmation, so the agent cannot ship code without the developer nodding.
- Auto-compact watches token consumption against each model's context window, triggers an LLM-based summarization pass when approaching the limit, and splices the summary back into the conversation. Long sessions stay coherent. Conversations persist, resume across terminal restarts, and pipe into CI for non-interactive runs. MCP support out of the box, so any MCP-compliant server drops in and the agent learns new tools at runtime.

Doom-Coder | AI Agent Lifecycle Tracker, macOS + iOS *2026*

- *Swift 6, SwiftUI, CloudKit (iCloud), IOKit, Carbon Events, Sparkle*
- Tracks **6+ AI coding agents** in real time, including Claude Code, Cursor, VS Code Copilot, Windsurf, and Codex CLI, and pushes a native macOS notification the moment any of them finish, fail, or need your attention. Built a hook socket listener, an event normalizer, and a per-agent process monitor with PID liveness checks, all running on a Swift 6 + SwiftUI menu-bar overlay.
- Shipped a free **iOS companion app** that mirrors agent state to your iPhone over a private **iCloud (CloudKit)** container. No accounts, no servers, no telemetry. The iOS app is a full standalone product with its own prompt library, on-device AI rewriting, notes, and a live Mac presence heartbeat.
- Engineered a smart "Auto" keep-awake mode that holds a single kernel-level IOPMAssertion only while a tracked agent is actively working, then releases after a **10-minute grace**. Carbon Events global hotkey, Sparkle auto-updates, signed and notarized DMG, hardened runtime. Distributed on GitHub Releases.

Technical Skills

Languages: Python, TypeScript, Swift, JavaScript, C/C++, SQL
Agent Systems: Agent Architecture and Design, Multi-Agent Systems, MCP, Tool-Use Orchestration, Human-in-the-loop (HITL), Prompt Engineering, LLM-as-a-Judge, Guardrails AI, custom agent loops, context compaction, agent evaluations, system design; LangGraph, LangChain, DeepAgents, Ollama, DSPy, ReAct; OpenAI / Anthropic / Gemini / Bedrock / Vertex / Azure SDKs
RAG and Retrieval: pgvector, FAISS, GraphRAG, BM25, hybrid retrieval, embeddings, sqlite-vec, semantic search
ML and Fine-Tuning: PyTorch, Hugging Face, Unsloth, QLoRA, PEFT, DPO, KTO, AWQ, vLLM, TensorRT-LLM, Weights and Biases
Agent Infrastructure: FastAPI, SSE streaming, WebSockets, Pydantic, asyncio, SQLite (WAL), Redis, Celery
App and Product: Electron 41, React 19, Next.js, SwiftUI, AppKit, iOS (UIKit / SwiftUI), CloudKit, Tailwind, Zustand, Vite, Xcode, XcodeGen
Data and Cloud: PostgreSQL, MongoDB, AWS (EC2, RDS, Bedrock, Lambda, S3), GCP, Docker, Kubernetes, CI/CD, Airflow, dbt, Terraform
Testing and Quality: Vitest, Playwright, pytest, Biome