YASH VIJAY

Backend & AI Engineer | Full-Stack Capable

Jaipur, Rajasthan, India | +91 7976882006 | yashvijay049@gmail.com

Portfolio | LinkedIn | GitHub

PROFESSIONAL SUMMARY

Backend & AI Engineer specializing in distributed systems, fault-tolerant infrastructure, and production-grade agentic AI / Hybrid RAG platforms. Track record architecting distributed workflow orchestration engines, resilient queueing systems, and sub-millisecond API infrastructure using Node.js, Fastify, Python, and Redis. Fluent across the full agentic AI stack — LLM orchestration, RAG pipeline design, and multi-agent workflows (n8n, Langflow, MCP) — from architecture through production deployment. Full-stack range across Angular, React, Next.js, and .NET, currently applied to InsightFirst, an enterprise analytics and AI-driven workflow product, at Evalueserve.

TECHNICAL SKILLS

- **Backend & Agentic AI:** Node.js, Fastify, Express.js, .NET Web API, FastAPI (Python), Prisma ORM, Agentic Workflows, RAG (Retrieval-Augmented Generation), n8n, Langflow, MCP (Model Context Protocol)
- **Frontend:** React, Next.js, Angular 19, Tailwind CSS, TypeScript, HTML/CSS
- **Databases & Systems:** PostgreSQL, MongoDB, SQL, Redis, Qdrant (Vector DB), System Design
- **DevOps & Tools:** Docker, Kubernetes, CI/CD, Git, GitHub, Postman , Aws (EC2 , S3)
- **Languages:** Python, Java, JavaScript

PROFESSIONAL EXPERIENCE

Evalueserve India — Software Engineer (Full-time)

Aug 2025 – Present

- Own end-to-end feature delivery for InsightFirst, an enterprise analytics and AI-driven workflow product — designing React-based dashboards and Python/.NET Web API services for production use.
- Lead design and implementation of agentic workflows in Langflow, automating multi-step AI-driven processes and reducing manual process overhead within the product.
- Architected resilient REST/WebSocket interfaces across Python and .NET services, cutting end-to-end API response latency by 35% and supporting multi-step agent execution.
- Drive feature delivery across Angular 19, .NET Web API, SQL, and the MERN stack within a cross-functional agile team, taking features from design through production release.

Evalueserve India — Software Engineering Intern

Feb 2025 – Aug 2025

- Built a secure, company-wide Learning Management System (LMS) using Angular 19, Bootstrap, .NET Web API, and SQL, with JWT authentication, Google reCAPTCHA v3, and Microsoft 2FA.
- Implemented role-based access control enabling seamless redirection for Users, Admins, SMEs, and TAs on a secure internal company server.
- Developed dual dashboards — User (course enrollment, quizzes, assignments, certificates) and Admin (department-based course management, SME tools, reporting with charts).

FEATURED PROJECTS

AetherFlow — Distributed Agentic Workflow Engine & Enterprise Hybrid RAG Live Demo

TypeScript, Node.js (Fastify), Next.js 16, React Flow, PostgreSQL, Prisma, Redis, BullMQ, Google Gemini

- Decoupled long-running LLM and document-ingestion pipelines into asynchronous BullMQ worker pools across Redis, reducing initial API acknowledgment latency from ~8s to <150ms while streaming live node execution state to the client over WebSockets.
- Designed an interactive drag-and-drop visual workflow canvas (React Flow) supporting 7+ extensible node primitives — autonomous AI agents, sandboxed Python/JS execution, foreach iterators, guardrails, webhooks — cutting manual workflow setup time from hours to under 10 minutes.
- Engineered a multi-stage Hybrid RAG pipeline combining hierarchical parent-child chunking (3,000-char parent context / 600-char child vectors), dense vector retrieval on PostgreSQL, and cross-encoder neural reranking, achieving 92%+ retrieval accuracy on complex documents.
- Developed a resilient Dead Letter Queue (DLQ) with automated backoff and state-preserving replay, preventing mid-workflow data loss during third-party LLM rate limits (HTTP 429/500) and enabling 100% recovery of failed steps.

- Implemented a self-healing distributed Redis cache with mutex locks to eliminate thundering-herd API spikes, cutting redundant LLM embedding calls and token costs by ~45% while serving cached queries in <10ms.
- Enforced enterprise multi-tenancy with strict workspace data isolation, granular RBAC (Admin/Member/Single), AES-256-GCM encrypted credential vaults, and RFC 6238 TOTP two-factor authentication (2FA).

AgentGuard — Policy Governance & Execution Control for AI Coding Agents Live Demo

Python, FastAPI, PostgreSQL, Docker, AWS EC2, Open Policy Agent (OPA), Rego, LangGraph, E2B Sandboxing, React, TypeScript

- Engineered an enterprise-grade execution control and policy governance platform for autonomous coding agents, intercepting 100% of LLM tool invocations into structured Action Intents to prevent unauthorized repository mutations and runaway cloud costs.
- Built a sub-15ms policy evaluation engine using declarative OPA Rego policies, budget ceilings ($0.50/run), and a deterministic risk-scoring model to govern every agent action in real time.
- Designed a dual-custody quarantine workflow that automatically pauses agent execution on critical-path modifications, routing changes to an interactive diff dashboard for human sign-off (HITL).
- Guaranteed 100% webhook idempotency and race-condition immunity through PostgreSQL row-level locking and per-PR serialization on a FastAPI backend deployed on AWS EC2 with Neon Postgres and Docker.
- Isolated test reproduction in ephemeral E2B micro-sandboxes and maintained tamper-proof, cryptographic audit ledgers across every autonomous reasoning and execution phase.

Enterprise Knowledge Intelligence Platform GitHub

FastAPI, Next.js, Qdrant, PostgreSQL, Redis, Python, TypeScript

- Engineered an enterprise Hybrid RAG retrieval pipeline combining dense vector embeddings (Qdrant) and sparse lexical search (BM25) with Reciprocal Rank Fusion (RRF, k=60), lifting retrieval Recall@5 by ~30% over standalone semantic search.
- Integrated a cross-encoder neural reranking stage (BAAI/bge-reranker-base) and sliding-window recursive chunking (512-token windows, 64-token overlap), reducing irrelevant context injection and hallucination rates by ~40%.
- Built a real-time Server-Sent Events (SSE) streaming chat service in FastAPI, reducing perceived response latency by ~75% (<350ms Time-to-First-Token) with dynamic citation mapping to exact source pages and documents.
- Architected three-tier multi-tenant isolation across SQL, Vector (Qdrant payload filters), and Lexical layers, with in-memory singleton caching achieving sub-150ms hybrid retrieval latency across concurrent user sessions.

Distributed, Low-Latency Rate Limiter Service GitHub

Fastify, TypeScript, Redis, Lua Scripting, Distributed Systems, Circuit Breaker

- Engineered a standalone, stateless rate-limiting microservice using Fastify and Redis, achieving an ultra-low latency overhead of <2ms for centralized API traffic governance.
- Designed an atomic execution engine using native Redis Lua scripts, guaranteeing 100% data integrity and eliminating double-spend race conditions under high-concurrency loads.
- Optimized resource consumption by replacing active background timers with an O(1) lazy-evaluation math algorithm, using aggressive Time-To-Live (TTL) cache purging to minimize memory footprint.
- Architected a Fail-Open circuit breaker with a strict 200ms network timeout threshold, ensuring core business traffic remains entirely uninterrupted during security-layer infrastructure degradation.

Resilient Redis-Queue (ZestQueue) GitHub

Node.js, TypeScript, Redis, ioredis, Fastify, Atomic Operations, Transaction Pipelines

- Engineered a highly available background processing engine capable of scaling to 10,000+ concurrent tasks per minute using non-blocking event loops and in-process Redis connection pooling.
- Guaranteed absolute data consistency for distributed ledgers during simulated catastrophic server failures by engineering O(1) idempotency locks via Redis SETNX, achieving a 0% duplicate transaction rate.
- Automated system recovery and observability, reducing orphaned-task resolution time to under 5 minutes using a custom visibility-timeout Sweeper, monitored via a sub-millisecond Fastify API.

EDUCATION

Chandigarh University — *Mohali, Punjab, India*
Bachelor of Engineering in CSE, CGPA: 8.03 | 2021 – 2025

CERTIFICATIONS

- Programming in Python — Coursera, 2023
- Probability and Statistics — Elite NPTEL Certificate, 2023