Jay Bhende

Applied AI Engineer | Full Stack

Mobile: +91-9325712924
Email: bhendejay5.bms@gmail.com

Github: JayBhende05
LinkedIn: Jay Bhende

Summary

Applied AI engineer building LLM systems with guardrails: a LangGraph support agent where deterministic code, not the model, decides what runs; a coding agent built from scratch over 4 LLM providers; and a RAG system with benchmarked hybrid retrieval and validated citations. Strong full-stack background in TypeScript, Python, FastAPI and PostgreSQL, with three client websites in production.

EDUCATION

- **Master of Computer Application (MCA)** *(Savitribai Phule Pune University)* — Maharashtra, India
  ZIBACAR College, Pune CGPA: 8.5 — *June 2024 - June 2026*

- **Bachelor of Computer Application (BCA)** *(Savitribai Phule Pune University)* — Maharashtra, India
  S.N. Arts, D.J. Malpani & B.N. Sarada College, Sangamner CGPA: 8.5 — *June 2020 - June 2024*

SKILLS SUMMARY

- **AI/LLM**: LangGraph, tool-calling agents, RAG, hybrid retrieval (vector + BM25), RRF, cross-encoder reranking, structured outputs (Pydantic), human-in-the-loop approval, Langfuse and OpenTelemetry tracing, Anthropic, Gemini and Groq APIs, ChromaDB
- **LLM Providers & Observability:** Anthropic, Gemini and Groq APIs, OpenAI SDK, Langfuse, OpenTelemetry
- **Languages**: Python, JavaScript, TypeScript, SQL
- **Frontend**: React, Next.js, Tailwind CSS, TanStack Query, React Flow, GSAP, Framer Motion
- **Backend**: Node.js, Express.js, FastAPI, REST APIs, WebSockets, Prisma, SQLAlchemy
- **Testing**: Jest, Vitest
- **Database**: MongoDB, PostgreSQL, MySQL, Redis
- **Deployment :** Vercel, EC2, Nginx, Docker Compose
- **DevOps/Tools**: Git, GitHub, Docker, AWS (Cloud Practitioner), Linux, VS Code

Certification

- **MERN Stack Application Development** – TechSaksham (Microsoft & SAP CSR Initiative, implemented by Edunet Foundation)
  Completed hands-on training in full-stack web development using MongoDB, Express.js, React.js, and Node.js.

- **AWS re/Start Cloud Practitioner Program** — Generation India
  Covered AWS Cloud Foundations, Linux, Networking, Security, Python basics, Databases, Web Servers, and Cloud Infrastructure concepts.

Experience

Freelance Full-Stack Developer
Self-Employed | Remote
2025 – Present

- Built and deployed 3 production websites with Next.js, React, TypeScript and Tailwind CSS:
  - Aquila Marketers (digital marketing agency)
  - Snappy Motions (motion design studio)
  - Kintsugi Village –((early-learning centre, including an admin access system).)
- Designed responsive and animation-rich user interfaces using GSAP and Framer Motion.
- Managed deployment workflows, performance optimization, CDN integration, and production maintenance.
- Collaborated directly with clients to understand requirements, implement features, and deliver scalable web solutions.

PROJECTS

- **Triage Agent - Human-in-the-Loop AI Support Agent** (Python, FastAPI, LangGraph, Pydantic, PostgreSQL, SQLAlchemy, React, TypeScript, Docker)
  Github :
  Built an **AI agent** that triages customer-support tickets, looks up customer and order data, and proposes one action (reply, refund, close as spam, or escalate). **Deterministic code controls execution**, and a human approves risky actions.
  - Designed a **LangGraph state machine** (triage → gather context → propose → risk gate → human approval → execute → finalize). The model only gets read-only tools; write tools (create_refund, draft_reply) run in graph nodes after the gate, providing **protection against prompt injection**.
  - Implemented a **deterministic risk policy with 6 rules**: refund approval, refund limits, sender/order-owner verification, a confidence threshold, and a loop guard on tool calls. Reviewer edits are validated by the same rules as model proposals.
  - Built **durable human-in-the-loop approval** using LangGraph interrupts checkpointed in PostgreSQL. Reviewers can approve, edit or reject from a React dashboard, and the agent **resumes after server restarts**.
  - Ensured **concurrency safety and idempotent refunds** (unique key + row lock), conditional UPDATE for ticket claiming, and SELECT … FOR UPDATE on decisions. Double approvals return **HTTP 409** instead of creating a second refund.
  - Added **Pydantic-validated structured outputs**, a tool runner with timeouts and retries with backoff and jitter, and a **complete audit trail** of every model call, tool call and decision with tokens, latency and cost (**~₹0.02 per ticket**). Tested with pytest against a real PostgreSQL database.

- **Coding Agent CLI - Terminal AI Coding Agent Built from Scratch** (TypeScript, Node.js, Anthropic SDK, OpenAI SDK, OpenTelemetry, Langfuse, Vitest)
  Github :
  Built a **terminal-based AI coding agent from scratch without any agent framework**. The model explores and edits a codebase through tools, streams its responses, and asks for confirmation before changing any file.
  - Engineered a **streaming tool-calling agent loop** that runs model turns, executes tool calls and feeds results back until the task completes (**20-turn cap**). Tool failures are returned to the model as errors so it can retry or change course.
  - Designed a **provider-agnostic LLM architecture supporting 4 backends** (native Anthropic, Groq, Gemini, and Anthropic's OpenAI-compatible endpoint). Each adapter converts a shared message format to its API's tool-calling schema, keeping the loop and tools provider-agnostic.
  - Implemented **7 developer tools** (read, ls, find, grep, write, edit, bash) with a **path guard preventing access outside the project root**, output truncation limits, and process-tree cleanup on command timeouts.
  - Added **human approval for file changes**: write and edit validate their input first, show a preview and wait for confirmation, and refuse to run in non-interactive terminals.
  - Integrated AGENTS.md project context into the system prompt and **OpenTelemetry tracing to Langfuse** for every model call.

- **ResearchPilot - Grounded RAG System for Research Papers** (Python, FastAPI, ChromaDB, BM25, BGE Reranker, Gemini API, Next.js, TypeScript)
  Github :
  Built a retrieval-augmented generation system that answers questions over technical research papers and returns the validated evidence behind each answer, developed across four benchmarked versions.
  - Implemented **hybrid vector + BM25 retrieval with RRF fusion** and **BGE cross-encoder reranking**, raising evidence **Top-3** from **70% to 90%** and **Top-5** from **80% to 92.5%** while holding **document retrieval at 100%.**
  - Built a **40-question benchmark across 4 papers** with document-level and page-level evidence labels, replacing subjective checks with Top-1/3/5 retrieval metrics.
  - Ran **8 controlled retrieval experiments** (chunking, query transformation, multi-query, hybrid search, reranking, pool size) and kept only the 2 that measurably improved results.
  - Added grounded generation with **Gemini structured output and citation validation**, so answers can only cite chunks the retriever supplied.
  - Served the pipeline through a **FastAPI** service with models loaded once at startup and **per-stage latency reporting**, and built a **Next.js** chat UI with a sources panel.

- **TriggerLab - Event-Driven Workflow Automation Platform** (TypeScript, Node.js, PostgreSQL, Prisma, Kafka, Webhooks, Solana)
  Github :
  Built an event-driven workflow automation platform where users can create webhook-triggered workflows and execute actions asynchronously through Kafka-based workers, including email and Solana transfers.
  - **Designed and built an event-driven workflow automation platform** with independent API, webhook, database-processing, and worker services.
  - Implemented an **Outbox-based asynchronous execution pipeline**: Webhook → Zap Run → Outbox → DB Processor → Kafka → Worker, decoupling event ingestion from action execution.
  - Built **Kafka workers with retry handling and offset management**, enabling failed actions to be reprocessed without losing workflow events.

∗ Implemented **ordered multi-action workflow execution**, where each compl eted action publishes the next action to Kafka for independent worker processing.
∗ Built extensible workflow execution supporting **dynamic inputs from webhook payloads**, with Email, Discord and Solana transfer actions and architecture designed for additional integrations.

ACHIEVEMENTS

- **Developed** a full-stack event management system for **TechMantra 2025**, handling **450+ registrations** and **₹20K+ transactions** with **real-time admin tracking** using **React, Node.js, and MongoDB**.
- Published a paper in **IJSREM Journal (Vol. 09)** on **"Impact of IPv6 Adoption on Internet Infrastructure"**
- **Ranked 1st** in Student Research Project presentation on **"Use of AI in Environmental Sustainability"**