

YASH KUMAR

AI Application & Full-Stack Engineer · Chatbots, LLM Integration, Web Apps · Founder, VixitAI Labs

Ranchi, Jharkhand, India · vixitaiash@gmail.com · +91 70042 24293 · LinkedIn · Portfolio: studios.vixitai.com

SUMMARY

AI application and full-stack engineer, four years shipping and then operating what I build. I run a live LLM chatbot in production — a multilingual clinical assistant on a client's website, answering patients in Hindi, English or Hinglish and steering them into the booking flow — and a retrieval-augmented knowledge-base service with a document ingestion pipeline, pgvector search, cross-encoder reranking and an evaluation harness that gates CI. I integrate LLM APIs across Anthropic, OpenAI, Google and DeepSeek, and build the web applications and REST APIs around them: 39 services and 17 production databases on a single VPS, all mine end to end.

Measured on the live estate, 13 September 2026: **39 services in production, 17 PostgreSQL databases, 1,572 API routes and 527 tables across six products, 142 SQL migrations, 23 live hostnames on one VPS.** Counted by a script kept with this résumé's source, third-party code excluded. Every system named below is running now and can be demonstrated end to end.

AI & CHATBOT ENGINEERING

“Dr Anshuman” clinical assistant · production LLM chatbot on a client website geetasriretina.org

- Shipped a multilingual patient-facing chatbot that is live on a real clinic's website, answering in Hindi, English or Hinglish — it mirrors whatever language the patient types in, and three UI pills can force a language, implemented as server-side prompt overrides appended after the base rule so the override wins even when the patient writes in another language.
- Built the whole path myself: a Node/Express proxy so the API key never reaches the browser, a whitelisted-role message validator that caps history at the last 10 turns and 2,000 characters per turn, a per-IP limit read from the trusted client IP rather than a spoofable header, a 20-second abort, and a fallback on every failure branch that still answers the patient and points them at booking instead of surfacing an error.
- Made the model drive the interface rather than just talk: the system prompt emits a structured [[LEARN: <key>]] tag from a fixed eight-key whitelist, which the client parses to open the matching patient-education panel — so an answer about screen fatigue also opens the eye-exercise content. Clinical safety is in the prompt contract too: general guidance only, a translated disclaimer on every medical answer, and an explicit escalation to an in-person visit on red-flag symptoms.
- Closed the loop from assistant to revenue: the bot routes patients into a booking flow that issues a queue token, fires a same-day alert to the clinic's inbox and sends the patient a confirmation with arrival instructions — replacing three notification paths that looked like they worked and reached nobody (browser push with zero subscriptions, a socket event needing the queue screen open, and a WhatsApp share link with no recipient). Reception calls back from the alert; the clinic reports roughly ₹1 lakh in additional revenue in the first 19 days after launch.

VIXIT Cortex · retrieval-augmented knowledge base over your own documents cortex.vixitai.com

- Built the full RAG pipeline end to end — PDF, DOCX and CSV parsing with OCR fallback, normalisation, multi-level chunking, batched embedding and pgvector storage — then two-stage retrieval: vector cosine search narrowed by a Voyage rerank-2.5 cross-encoder. Embeddings run on Google text-embedding-005 or Voyage, behind one provider interface so a model swap is a config change.
- Designed the seven-layer query router so only two of seven layers ever call an LLM — schema, saved-query, cache and statistical lookups answer first — which keeps the common question cheap and fast instead of billing a model for it.
- Made answer quality a build gate rather than a vibe: an evaluation harness with retrieval metrics, an LLM-as-judge suite and baseline comparison, wired to a CI command that fails the build on regression, with 20 dated runs kept on disk as evidence.

AI text-analysis services · Python, FastAPI, model serving

- Built Python and FastAPI services serving transformers, torch and sentence-transformers models behind HTTP APIs across 24 endpoints, ensembling a fine-tuned RoBERTa classifier with a from-scratch GPT-2 perplexity and burstiness signal so a single model's bias cannot decide the answer alone, alongside plagiarism, grammar and style analysis.

EXPERIENCE

Founder & Full-Stack Engineer · VixitAI Labs, Ranchi

2022 to Present

Independent software work since 2022, formally registered as VixitAI Labs in 2026 (Udyam + GST). Architecture, engineering, DevOps, security and product for a multi-product estate, single-handed.

VIXIT Clinic ERP · multi-tenant clinic and hospital platform clinic.vixitai.com

- Shipped 60 clinical modules across 813 API routes and a 207-table PostgreSQL schema — EMR, pharmacy POS with GST returns and a narcotics register, lab, radiology with DICOM, IPD, OT, blood bank, TPA, HR and a 56-endpoint patient portal — as the only engineer, on a two-instance PM2 cluster behind Nginx with per-tenant isolation enforced across 52 files.
- Took the platform from records to revenue by building Razorpay subscription billing with live recurring mandates, GST-inclusive invoicing and module-level licensing, so a plan maps to a feature set without a code change.
- Turned India's health-data standard from a blocker into a shipped surface: 28 ABDM/ABHA endpoints covering consent, HIP callbacks and FHIR bundles, with AES-256-GCM encryption at rest for imaging.
- Hardened the stack myself — least-privilege PostgreSQL roles, audit logging, JWT with token-version revocation that ends every session on password reset, a client-secret header enforced across 192+ calls, fail2ban and Nginx WAF rules at the edge.

Upper Bazaar · livelihood and local-commerce marketplace upperbazaar.com

- Built a bilingual (English and Hindi) marketplace — 288 API routes, 104 tables, 68 forward-only migrations — covering worker profiles, shops, orders, delivery dispatch, scheduled slots, wage panels and government-scheme guidance, with a Next.js 16 / React 19 storefront and four single-file operator consoles non-technical staff run without me.
- Replaced a test suite that passed silently — the repository's configured test command matched no files at all — with 100 end-to-end scripts against an isolated stack owning its own database, Redis and ports, now permanently running rather than a folder of files.
- Cut mobile Lighthouse performance from 47 to the high 90s and largest-contentful-paint from 7.4s to 2.2s through responsive image renditions and font-payload work.

Jantar Mantar · short-video platform with a native Android client

- Built a short-video platform — 116 routes, 43 tables, 38 migrations — with HLS transcoding, a ranking worker, creator upload, a moderation console and Cloudflare R2 storage, verified by 11 end-to-end suites carrying 416 assertions.

Rentlelo · peer-to-peer rental marketplace

Freelance Web Developer · Independent

2021 to Present

- Built and operate the patient portal for Geeta Sri Retina Institute: OTP sign-in, self-serve registration, booking, reschedule and cancel, a live queue, per-visit medical-record PDFs, automated patient and clinic email, a Hindi interface, and a multilingual assistant proxied server-side so no API key reaches the browser.
- Delivered a local-SEO overhaul for the same clinic (11 articles, an indexable Hindi section, structured data), and built a client-managed photo slider on Cloudflare R2 so Ekdant Dental updates its own gallery without me.

SELECTED ENGINEERING

SPARSE · CPU-only detection of resident space objects in neuromorphic event-camera streams

- Held per-window detection to 1.01ms mean and 1.46ms p99 against a 40ms real-time budget on two CPU cores with no GPU, by staying in the event domain end to end — noise filtering, $O(N)$ grid clustering with sub-pixel centroids, then a constant-velocity Kalman filter with probabilistic data association — instead of rasterising events into frames for a CNN.
- Made it reproducible by a stranger: 2,114 lines of NumPy and SciPy with 60 tests, packaged as a non-root Docker image on a read-only registry host, so `docker pull vixitai.com/sparse:1.0` reproduces the reported numbers exactly.

INFRASTRUCTURE & OPERATIONS

- Sole administrator of one Ubuntu VPS carrying 39 PM2 services, 17 PostgreSQL databases, 23 Nginx vhosts and 22 auto-renewed TLS certificates — every product and client site on this page.
- Cut the public attack surface to ports 22, 80 and 443 in a hardening pass, moved secrets out of web-reachable paths, and pinned JWT algorithms after an authentication audit found an unpinned alg and an OTP bypass.
- Found a backup that had been silently failing for five days while reporting success — offsite uploads were dead and the job still returned COMPLETE — and added the verification step that would have caught it on day one.

COMPETITIONS & RECOGNITION

- **TII OrbitSight Challenge** (US\$50,000): submitted SPARSE as a reproducible container, with its results and its limits both stated in the artefact.

TECHNICAL SKILLS

AI & LLM: Chatbots and assistants on live sites; LLM APIs (Anthropic, OpenAI, Google, DeepSeek); prompt engineering (server-side system prompts, language overrides, structured output contracts); RAG and knowledge bases (pgvector, Voyage and Google embeddings, cross-encoder reranking, document ingestion); LLM-as-judge evaluation with a CI gate; transformers, torch and sentence-transformers behind FastAPI; AI-assisted development (Claude Code, daily)

Languages: TypeScript, JavaScript (Node.js and browser), Python, SQL, Solidity, HTML/CSS, Bash

Backend & APIs: Node.js, Express, Fastify, FastAPI, REST API design, BullMQ and Redis job pipelines, Socket.IO, webhooks, payment integration

Data: PostgreSQL schema design, forward-only migrations, query and index tuning, PgBouncer, PostGIS, pgvector, pg_trgm, Redis, multi-tenant isolation

Frontend & Mobile: React 19, Next.js 16, Tailwind CSS, Three.js and React Three Fiber, GSAP, React Native and Expo with over-the-air updates, Android APK and AAB builds, progressive web apps with service-worker strategies and VAPID push

DevOps & Infra: Ubuntu, Nginx, PM2 clustering, Docker and self-hosted registries, systemd, Cloudflare and R2, Certbot and TLS, deploy scripting, backups, monitoring and alerting

Security: Least-privilege database roles, JWT with token revocation, OTP and session auth, AES-256-GCM encryption at rest, audit logging, consent capture, PII masking, fail2ban, Nginx WAF rules, rate limiting

Testing & Practice: Vitest, Jest, e2e harnesses against isolated stacks, scripted release verification, PRDs and design documents written before code, Git, Figma, AI-assisted development (Claude Code, daily)

EDUCATION

- Self-taught: four years of full-stack development, system design, databases, DevOps and application security — measured by what is running, not by coursework.
- Higher Secondary Certificate (Class 12), CBSE, 2021.