Yaseen Khatib

Senior MERN Stack Developer · AI Systems

I build MERN products end to end, then put the AI inside them.
Seven shipped solo, each one you can open.

yaseenkhatib04@gmail.com
+91 8208335028
linkedin.com/in/yaseen-yk
github.com/Yaseenyk
yaseenkhatib.streamerosai.com
Hyderabad, India · Remote / Hybrid / On-site

5+	7	94%	2x	34
YEARS FULL-STACK	PRODUCTS SHIPPED SOLO	SMALLER API PAYLOADS	FASTER DASHBOARDS	ARTICLES PUBLISHED
React · Node.js · MongoDB	in twelve months, each one open	React Flow + MongoDB serialization	MongoDB pipeline profiling	+35 syndicated on DEV.to

SUMMARY

I build full-stack products on **React, Next.js, TypeScript, Node.js, Express and MongoDB**, and for the last year I have been building the AI inside them. I started in 2021 on hospital information systems that clinical staff used every day. In 2023 I joined MSA Software as a full-stack developer, profiled slow MongoDB pipelines until dashboards ran **1.5–2x faster**, and moved the team to TypeScript and Next.js. Since 2025 I own the **LLM and agentic layer** of production MERN systems at Sparity: **RAG pipelines, function-calling agents, MCP servers**. On the side I have shipped **seven products solo**: an AI gym app with on-device form check, a React Flow workflow engine that cut payloads **94%**, a distilled AI decision model, and four more, each with a public link. I own a system from database schema to deployed UI, and I am open to senior remote, hybrid or on-site roles.

SELECTED WORK

full write-ups on pages 2–3

SpotAI — MOBILE + WEB · ON-DEVICE AI

AI gym buddy for Indian gyms. Squat form check runs on the phone, frames never uploaded; members and staff in one app, Indian diet plans. Row-level security per gym, 190 API tests.

Site · Android and iPhone apps ↗

FitRank — APPLIED ML · LIVE

23M MiniLM student distilled from a 9B teacher. Beats a rules-only baseline on hit@5 and pairwise ordering, served on CPU at $0 per decision.

fitrank.streamerosai.com ↗

streamerOS — DESKTOP · RUST

Rust + Tauri 2 desktop cockpit for streamers, written with Claude Code. 24k lines of Rust, 306 Rust tests, OBS automation, local LLM. Launching November 2026.

streamerosai.com ↗

EXPERIENCE

Senior Software Developer
06/2025 – Present

Sparity · Hyderabad · Hybrid

- Own the **LLM and agentic layer** of production MERN systems end to end: architecture, orchestration, deployment.
- Ship **Agentic RAG services** with schema-validated outputs, guardrails, and cost-aware routing between models.

Full Stack Developer
01/2023 – 05/2025

MSA Software · Remote

- Cut dashboard response times **1.5–2x** by profiling slow MongoDB aggregation pipelines, adding a caching layer, and lazy-loading heavy panels.
- Moved the team's legacy JavaScript to **TypeScript and Next.js**, removing a whole class of type-related production bugs.
- Migrated shared state from hand-rolled Redux to **Redux Toolkit slices**.

Software Developer (SDE 1)
09/2021 – 01/2023

Manorama Infosolutions · Healthcare software · Remote

- Built REST routing layers for hospital information systems on **ASP.NET + React**; **2x** payload throughput on the hot paths.
- Moved tangled request handling into MVC and hardened it for concurrent user sessions.

SKILLS

FRONTEND
React · Next.js · TypeScript · JavaScript (ES2023) · Redux Toolkit · Zustand · React Flow · React Native / Expo · Tailwind CSS · HTML / CSS

BACKEND
Node.js · Express · REST APIs · WebSockets · JWT auth · Python / FastAPI · Java / Spring Boot · ASP.NET

DATABASES
MongoDB · PostgreSQL · SQL Server · SQLite · Redis · Firestore · pgvector

AI ENGINEERING
LLM integration · RAG · Embeddings & vector search · Function calling · Agentic workflows · MCP · Prompt engineering · Fine-tuning / distillation · LLM evals · OpenAI · Gemini · Claude APIs · PyTorch · Ollama

TOOLS & DELIVERY
Git · GitHub Actions · CI/CD · Docker · Cloudflare Workers · Unit & integration testing · System design · Agile delivery · Rust / Tauri 2 (using Claude Code)

EDUCATION

B.E. Computer Science · 2021
D. Y. Patil College of Engineering

Diploma, Computer Science · 2018
D. Y. Patil Polytechnic

Certifications (Coding Ninjas): Front End Architecture · React Production Development · Advanced Backend Systems

SpotAI · AI gym buddy for Indian gyms

Mobile + Web · On-device AI

One product, two sides. Members get a squat coach that runs MediaPipe pose on the phone and sends only rep counts and per-rep flags to the API, never a frame, plus a workout log, an opt-in gym leaderboard and buddy matching. Owners get live arrivals with dues flagged, UPI payments, busy hours and a trainer console that reviews every AI tip. Each purpose is a separate consent with its notice version stored.

PROOF One Vue 3 + Capacitor app for members and staff, NestJS on PostgreSQL with **row-level security per gym** (two gyms asserted side by side in tests), NATS outbox, model gateway with identity redaction. **190 API tests**, browser tests with WCAG AA checks on every screen.

Vue 3 · Capacitor | NestJS | PostgreSQL RLS | MediaPipe | NATS | Astro + GSAP

Android app ↗ iPhone app ↗ Case study ↗

FitRank · Typed AI decisions for staffing

Applied ML · Live

Ranks people for open tasks with typed decisions instead of a chatbot's opinion. Code applies the hard rules; a 23M-parameter MiniLM-L6 student with five linear heads, distilled from Qwen3.5-9B with a KL loss against the teacher's full distributions, returns a probability for every option; thresholds band each person shortlist / review / hidden, and a manager makes every final call. Managers' accepts and rejects become the next model's training data.

PROOF Evaluated against a code-only baseline on every release (35 held-out tasks, synthetic data): **hit@5 94% vs 91%**, **pairwise 0.88 vs 0.86**, NDCG@10 0.89 vs 0.89, and **hit@1 54% vs 74%**, which is why it shortlists rather than assigns. FastAPI router → service → repository, Postgres + pgvector, Alembic, JWT + refresh-cookie auth with roles, rate limits, a budget guard, Docker, GitHub Actions CI, **93.5% backend test coverage**.

Knowledge Distillation | MiniLM / PyTorch | FastAPI | Postgres + pgvector | React + TypeScript | Docker · CI

Live ↗ Case study ↗

streamerOS · Local-first streaming cockpit

Desktop · Rust

A Rust + Tauri 2 desktop app for Twitch and YouTube streamers. Chat is read locally, velocity and sentiment are scored in Rust off the UI thread against the channel's own baseline, and scene automation drives OBS over WebSocket v5. No account, no cloud; pull the network cable and it keeps working. Rust-core-to-webview IPC is batched MessagePack rather than per-event JSON; the assistant runs on Ollama at 127.0.0.1.

PROOF Rust written with Claude Code, reviewed and tested by me. **187 commits** since May 2026, 70 Rust files / **24,292 lines of Rust**, 42,733 lines of TypeScript, **306 Rust tests**. The app repository is private; the marketing site and its Tier-1 support agent (a Hono router on Cloudflare Workers that embeds the product docs into Upstash Vector and streams a gemini-flash answer constrained to that context) are public.

Rust | Tauri 2 | OBS WebSocket v5 | MessagePack IPC | Ollama | Next.js

Website ↗ Case study ↗ Site + agent source ↗

Portfolio Concierge · Grounded RAG API + MCP server

AI Infrastructure · Live

One zero-dependency Cloudflare Worker, two interfaces. A grounded RAG API answers questions about my work from a Vectorize index of the site's 80-document corpus (34 articles, 46 project pages) and refuses below a similarity threshold rather than guess. A public MCP server exposes search_portfolio, get_products and get_availability, so any recruiter can add the URL to Claude and interview this résumé from inside their own AI. The corpus re-embeds automatically after every deploy.

Cloudflare Workers | Vectorize | Workers AI | MCP | Grounded RAG | TypeScript

Try it ↗ MCP ↗ Code ↗

IntegrateX · Workflow automation engine

React Flow · Live Sandbox

A node-based workflow canvas (triggers, processing layers, directional edge bindings) backed by a Zustand store. The naive design stored the React Flow render model; a custom Serialization Adapter separates the view from the transport record and cut stored payloads by **94%**, round-trip lossless. The architecture runs live in the browser: drag nodes, rewire the graph, watch the serialization bench.

React Flow | Zustand | TypeScript | Serialization Adapter | MongoDB

Live sandbox ↗ Write-up ↗

Sable · Local-first AI financial agent

Mobile · AI Agent

A React Native / Expo personal-finance agent where the model is structurally incapable of touching the money. 100% of financial data stays on-device in SQLite with no cloud backend; an OpenAI function-calling agent tracks peer-to-peer debts and pushes proactive briefings, and every database mutation renders as a Review & Confirm card: it proposes, only a human commits. Bank-SMS ingestion uses the Adapter pattern with a p-queue serialising SQLite writes.

React Native / Expo | SQLite | OpenAI Function Calling | Local RAG | TypeScript

Case study ↗

APPLIED & CLIENT WORK

ShelfSight AI · Retail vision for a kirana shop

Computer Vision · Open Source

Four retail vision problems in one on-premise system: phantom stock, planogram compliance, produce freshness and expiry-date OCR. A manager lays out a rack once; after that a photo counts every facing and staff are told which shelf to walk to. Its most useful property is what it refuses to claim: the detector localises products without naming them, so the planogram endpoint returns 409 rather than a score it cannot support. **650 tests**.

FastAPI · YOLOv8 · MobileNetV2 · Next.js 14 · Docker · Ollama

Backend ↗ Dashboard ↗

AISA · Student association platform

Full-Stack · Live

Arrived as one 4,019-line HTML file with a live Firebase key in the source and every permission decided in the browser. Rebuilt as a Next.js static export on GitHub Pages and a Spring Boot 3 / Java 25 API on Render, both on Firestore. Firebase Auth answers who you are; the server decides what you may do, the half the original had backwards. Five auth states, **29 tests**.

Next.js · Java 25 · Spring Boot 3 · Firestore · Firebase Auth

Live ↗ Frontend ↗ Backend ↗

Path Saathi LMS · Vocational-skilling platform

Client · 1-Day Delivery

A free vocational-skilling LMS for a client sitting on a library of course videos and PDFs. Brief and MVP sketch arrived on a Monday; a working platform with auth, course delivery and a self-paced flow was live on the dev environment the next day.

Next.js · React · Tailwind · TypeScript

Live (dev) ↗

Autonomous Content Machine · Three pipelines, zero servers

Pipelines · $0 Infra

Three cooperating pipelines on GitHub Actions, running unattended at no cost: an AI writer that drafts a technical article with a brand-locked cover on Mon/Wed/Fri, commits it and deploys the static site; a LinkedIn agent that turns each post into scheduled, UTM-tracked content Tue–Thu with its state ledger committed to Git; and a nightly DEV.to syndicator that cross-posts with canonical URLs pointing home.

GitHub Actions · Python · Next.js · DEV.to API · Zero-Cost Infra

Case study ↗ LinkedIn agent ↗

WRITING & TEACHING

Engineering blog

34 articles

First-person write-ups of the systems above: serialization adapters, grounded RAG, agent tracing with OpenTelemetry, prompt caching, WebSocket telemetry at scale.

Read the blog ↗

DEV.to

35 articles

Syndicated automatically every night with canonical URLs pointing home.

dev.to/yaseenyk04 ↗

AI Engineering Journey

16 days · 64 hours

A curriculum I authored, from embeddings and vector search through RAG, agents, evaluation and fine-tuning to production, as 16 interactive chapters with diagrams and a built-in presentation mode.

Open the curriculum ↗

Open source

54 public repos

Since 2022, including the full source of this portfolio, its RAG/MCP worker, the content pipelines, ShelfSight and AISA.

github.com/Yaseenyk ↗

Every link in this document was checked on the date of generation. Repositories not linked are private. Metrics are the latest published evaluation, including where the model loses.