# Vivek Kumar

Noida, India | +91-7088980706 | vivekumar2003bsr@gmail.com
**vivekumar08.com** | github.com/vivekumar08 | linkedin.com/in/vivekumar08 | medium.com/@vivekumar08

**Best Launchpad Award 2024–25** (Bytive) | **Led the team that delivered the Kisna Diamonds platform** to production

## SUMMARY

Backend Engineer with **3 years** building **event-driven microservices** and multi-tenant commerce platforms in **Node.js, TypeScript and Go** on **AWS**, across **MongoDB, PostgreSQL, MSSQL and Redis**. Own problems end to end — design spec, benchmark, tests, production — and ship the **React/Next.js** surfaces on top.

## EXPERIENCE

**Software Engineer** Jul 2023 – Present
*Bytive Technologies* *Noida, India*

**Jewellery E-Commerce Platforms** — Kisna Diamonds, Iraah, PC Jewellers | Node.js, Go, MongoDB, Redis, AWS

- Cut nightly repricing of a **600K-variant** catalogue from **2.6 hours to 33 minutes** (**179x** fewer queries, **99.5%** hit rate) and a second tenant's **300K** run to **15–35 min**, by censusing queries before coding and adding a two-tier **Redis** master-data cache with batched writes.
- Cut API latency **40%** and eliminated dropped **ERP** ↔ **e-commerce** orders across **10K+ SKUs** by moving synchronous flows onto **event-driven microservices** (**AWS SQS**, **Azure Service Bus**) with idempotent consumers, DLQs and replay.
- Took a **WhatsApp commerce assistant** live to customers with a **Salesforce CRM** layer — cached OAuth2 token, audit log per call, every outbound channel behind an environment flag for staged rollout.
- Improved catalogue relevance and exposed zero-result searches by building **AI-powered hybrid search in Go** over **OpenSearch** — **agentic LLM query understanding** (intent, filters, multilingual terms, regex fallback), text + image embeddings behind a swappable **OpenAI/Gemini** provider, KNN vector + full-text ranking, and a query-analytics console.
- Closed the storefront-to-CRM loop with a fire-and-forget **event beacon** on CMS pages (per-block impressions and clicks, stable session id) and **SQS-driven syncing** of customers, products and gold rates from **MongoDB into MSSQL** for **Salesforce** ingestion.
- Delivered features **end to end** on Iraah: order-approval workflow with role attribution, family products, and a batched **PDF export** pipeline.
- Diagnosed production data races with **zero downtime** via a **Winston** → **Loki** → **Grafana** pipeline, and cut deploy time **40%** automating **CloudFormation** and **CI/CD**.

**Sadashree — Multi-Vendor Marketplace** — vendor listings, commission and extra-charge management | React, Node.js, Cloudflare R2

- Protected vendor-negotiated pricing with super-admin-gated **vendor price locking** in Price Master, and corrected extra-charge handling so vendor order totals settled correctly.
- Removed the application server from the upload path via **direct browser-to-R2 uploads** (presigned, bounded concurrency, byte-level progress) behind one env-driven storage abstraction over **DigitalOcean Spaces, S3 and R2**.

## PROJECTS

**VTempo — Event-Driven Accountability SaaS** *TypeScript, Express, PostgreSQL, Prisma, Docker*

- Built the escalation engine: rule predicates over bounded windows and miss thresholds, a delayed queue and worker, and consumers on `occurrence.missed`/`.skipped` that **claim an event before sending** so a retry cannot double-notify.
- Modelled groups, members, invites and plan tiers on **PostgreSQL/Prisma** — a downgrade **archives** excess groups rather than deleting them, and a four-level **visibility engine** clamps shared payloads to the strictest floor, previewable live before saving. Design spec and TDD plan written per phase; NotFound-before-Forbidden so the API never leaks existence.

**qr-infra — Product Identity Resolution API** *TypeScript, PostgreSQL, Drizzle, Redis, Next.js*

- **GS1 Digital Link** / Digital Product Passport resolver in a ports-and-adapters monorepo — domain invariants enforced under a row lock, so "a revoked code cannot be repointed" holds for every caller — with auth, rate-limit and **idempotency** middleware and three entrypoints (API, scan resolver, worker) from one image.
- Kept the Redis read model correct with a **transactional outbox**, not a dual write — each mutation writes Postgres and enqueues in one transaction, swept by **BullMQ** — and enforced inward dependencies with a **CI architecture checker** barring the resolver from the database, which let the hot path move off Cloudflare Workers unchanged.

## TECHNICAL SKILLS

- **Languages & Backend:** TypeScript, JavaScript, Go, Python, SQL | Node.js, Express, gRPC, REST, AWS Lambda
- **Distributed Systems:** Event-driven architecture, AWS SQS/SNS, Azure Service Bus, Redis queues, idempotency, DLQs & replay, fault tolerance
- **Data:** MongoDB, PostgreSQL, MSSQL, MySQL, Redis, OpenSearch, S3, Cloudflare R2
- **AI & Search:** Agentic LLM query understanding, OpenAI & Gemini, text + image embeddings, vector/KNN search, RAG-style retrieval
- **Frontend:** React, Next.js, Redux Toolkit & RTK Query, Tailwind CSS, MUI, shadcn/ui
- **Cloud & Practice:** AWS, Docker, CI/CD, CloudFormation, Grafana & Loki, profiling & benchmarking, RBAC, design specs before code, feature-flagged rollouts

## EDUCATION

- **B.Tech, IT & Mathematical Innovations** — Cluster Innovation Centre, University of Delhi | **CGPA 8.981/10** | 2020–2024