YASH GUPTA

+91 7024025543 | ygupta9132@gmail.com | Raipur, Chhattisgarh
linkedin.com/in/ygupta9132 | github.com/ygupta9132

SUMMARY

Software developer shipping features end to end: a multi-tenant NestJS and **Next.js** WhatsApp platform serving **170+ connected business accounts**, a Django ERP, and a CRM. Built **User Journeys** from schema to UI, took campaign throughput **14x**, made a core API **17x** faster, and root-caused **30+ production incidents**.

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, Python, SQL
Backend: NestJS, Node.js, Express, Django, Celery, BullMQ, REST APIs, WebSockets, Webhooks, OAuth 2.0
Frontend: React 19, Next.js 15 (App Router), TypeScript, Zustand, Tailwind CSS, Radix UI, Socket.IO
Data and Infra: PostgreSQL, Redis, Drizzle ORM, Docker, AWS S3, Linux
Concepts: Transactional Outbox, Idempotency, HMAC Signing, Multi-Tenancy, RBAC, Queues
AI: Claude Code (headless runs, tool allowlists), MCP tool servers, LLM tool calling, RAG, embeddings, vector search (pgvector, ChromaDB), LangChain, Ollama

EXPERIENCE

Software Development Engineer, GimBooks May 2026 - Present
ChatSocial (multi-tenant WhatsApp platform), GimBooks ERP, Jewellery CRM

- Owned **User Journeys v1** end to end and alone: Postgres schema, event-ingest API, dispatch scheduler and conversion windows, plus **the entire Next.js UI** (journey builder, audience selection, analytics). **Live in production** behind per-tenant feature flags.
- Designed a **cross-service event pipeline** from the Django ERP to the NestJS platform: a **transactional outbox** with a Celery sweeper, **HMAC-SHA256 signing** and idempotent at-least-once delivery, so payment events enrol merchants into journeys without dual-write loss.
- Re-architected campaign delivery onto a **Redis-backed job queue** (Bull), replacing a serial loop that lost work on every deploy: claims and state transitions are atomic **Lua** scripts, a reconciler re-adds jobs the queue drops, and pause and resume are idempotent. Measured in production, send throughput went from **~56 to ~790 messages per minute (14x)** - **18,615 contacts in 23.6 minutes** - with **zero duplicates and zero double charges** across 20,637 sends.
- Reduced a core contacts-list API from **515ms to 30ms (17x)** on a **528k-row** production dataset with covering indexes on two unindexed join tables and an index sort-order fix (DESC vs DESC NULLS LAST).
- Found and fixed two security flaws in a multi-tenant CRM: a **cross-tenant data leak** where JavaScript `&&` in query filters dropped the tenant condition and let contact sync read **every tenant's customer PII** (now covered by tests on the generated SQL), and a production **Docker image that shipped the `.env`**.
- Closed a set of **multi-tenant authorization holes** in production: an endpoint that let an enterprise client mint credentials, clients acting on records that were not theirs, and a guard that validated only the **first** company id in a request instead of every one.
- Root-caused **30+ production incidents**: an **8-day silent outage** from a webhook returning HTTP 200 on failure, a **duplicate delivery webhook charging customers twice**, and every scheduled job firing **11 times a tick** from a scheduler registered once per module.
- Own the **payments** paths: **iOS in-app purchases** verified against **Apple's App Store Server API**, with a **partial unique index** making a replay a no-op, and **metered billing** that charges each reply once from the delivery webhook and gates sends at 402 when credits run out.

PROJECTS

Clause - Contract Review Assistant with Verified Citations
Next.js 15, TypeScript, Postgres + pgvector, BullMQ, Docker | github.com/ygupta9132/Clause-Contract-Review

- Built a **multi-tenant contract reviewer**: a contract is checked against an 8-point checklist and **every model quote is re-verified against the source text before display**, so a paraphrase is never shown as a quotation. Hybrid **pgvector** and lexical retrieval, queued ingestion, per-user scoping on every table. **16/16** on a scoring harness, **52 tests**.

PerformNicely MCP - Claude Integration for Internal Task Management
TypeScript, Node.js, Model Context Protocol SDK, Zod, Claude Code

- Built an **MCP server** giving Claude **14 schema-validated tools** over the company's internal task manager, plus a **scheduled headless Claude Code run** that logs each day's work from git across 5 repositories under an explicit tool allowlist and a dry-run mode. Running unattended on **64 days since September**, every run logged.

EDUCATION AND ACHIEVEMENTS

B.Tech, Computer Science Engineering | O.P. Jindal University, Raigarh | 2022-2026 | **CGPA 9.0/10**
NPTEL Reinforcement Learning, Elite + **Top 2%** | NPTEL Cryptography and Network Security, Elite | **Notice period:** 30 days