# Nishant

## Backend Engineer

TypeScript · Node.js · Python · PostgreSQL

nishanttgithub@gmail.com devnishantt.github.io github.com/devnishantt

TRY TO BREAK MY BACKENDS

### SUMMARY

Backend engineer focused on the parts of a system that are expensive to get wrong: **concurrent money movement, idempotent payment APIs, exact money arithmetic, tamper-evident audit logs and approval workflows**. Failure cases are covered by tests, every project below runs CI on each push, and my portfolio lets you try to break these guarantees in five live labs.

### SELECTED PROJECTS

code links point to the exact lines

#### Bank Management System

TypeScript · Express 5 · PostgreSQL · Prisma · Docker

*Retail banking REST API: accounts, deposits, withdrawals and transfers.*

- Locked every account a request touches with **SELECT ... FOR UPDATE in id order**, so racing transfers queue instead of losing updates or deadlocking; a test fires **10 parallel withdrawals** of 30.00 at 100.00 and asserts exactly 3 succeed. ↗ CODE ↗ TEST
- Built **idempotent payments**: an `Idempotency-Key` replays the original result on retry, a reused key with a new body returns 422, and a unique index settles racing duplicates. ↗ CODE
- Money as **BigInt minor units** in a double-entry ledger whose credits minus debits must equal total balances, with `CHECK` constraints as a last line of defence. ↗ CODE ↗ CODE
- Refresh tokens stored only as SHA-256 hashes; replaying a rotated token **revokes the whole token family**. ↗ CODE

#### Ledgerline

React · TypeScript · Hono · Cloudflare D1 · Zod

*Reconciles bank statements against ledger records without silently signing off on differences.*

- Matching that **refuses to guess**: tied or conflicting candidates stay unresolved for a human; money is exact integer minor units per currency. ↗ CODE
- **Independent review** (a run's creator cannot approve it) and a **hash-linked audit log** with chain verification, missing-tail detection and SQL immutability triggers; 150+ tests in CI. ↗ CODE ↗ CODE

#### Aftermark

TypeScript · Fastify · Bun · Cloudflare D1 · OpenAPI

*Evidence-driven incident response API.*

- Closing an incident needs **two non-creator approvals of the current SHA-256 evidence manifest**; new evidence makes earlier approvals stale. ↗ CODE ↗ CODE
- Optimistic concurrency: each write claims the next revision and a conflicting writer's whole batch rolls back; SQL triggers keep audit events append-only. ↗ CODE ↗ CODE

#### Tracebench

Python · DuckDB · Pydantic · Typer · pytest

*Local-first dataset contracts and drift reports for CSV, JSONL and Parquet.*

- Drift checks report **"incomparable"** instead of a false "stable" when the data cannot support a verdict; bounded ingestion, fingerprinted reports, 100+ tests. ↗ CODE

#### Hospital Management API

TypeScript · Express · Prisma · PostgreSQL · Redis

*Hospital operations backend with payments and role-based access.*

- Razorpay payments verified by **HMAC-SHA256 signature**; TOTP two-factor auth, a Redis token blacklist and role-based access. ↗ CODE

### OPEN SOURCE

merged upstream

- **foss42/apidash** #1828 / Kept disabled environment variables from being lost after pre-request scripts ran.
- **CrossGL/crosstl** #3612 / Aligned HIP `__constant__` binding numbering with the CUDA importer.

### SKILLS

| | | | |
|---|---|---|---|
| LANGUAGES | TypeScript, JavaScript, Python, SQL | BACKEND | Node.js, Express 5, Fastify, Hono, Bun, REST, OpenAPI, Zod, Pydantic |
| DATA | PostgreSQL, Prisma, SQLite / Cloudflare D1, Redis, DuckDB, MySQL, MongoDB | RELIABILITY | Row locking, optimistic concurrency, idempotency keys, double-entry ledgers, hash-chained audit logs |
| SECURITY | JWT with refresh rotation, RBAC, HMAC signature checks, TOTP 2FA, rate limiting, input validation | DELIVERY | Vitest, pytest, Playwright, GitHub Actions, Docker, Cloudflare |

Portfolio: 5 live labs, unit and browser tests run on every deploy. devnishantt.github.io