

Javohir (Ned)

Backend Developer & PostgreSQL DBA

+82 10-8864-3848 | nediro883@gmail.com | github.com/kicheol83 | KakaoTalk open.kakao.com/o/sa2bEaRi | javohir.dev | Seodaemun-gu, Seoul

Summary

I'm Javohir, a backend developer who builds reliable systems around PostgreSQL. I designed and deployed a double-entry ledger, a PostgreSQL monitoring platform, an IoT pipeline validated with real ESP32 sensors, and a live-commerce platform built on Saga and Outbox. My strengths are database design and performance tuning, concurrency control, and transactional integrity. Over four years I have worked with NestJS and Java /Kotlin/Spring Boot, including at Sigma Software and LeverX Group in Uzbekistan, and as a freelance backend and database developer for manufacturing, F&B and AI service companies in the UAE, Hong Kong and Bangladesh.

Skills

Languages	TypeScript, MicroPython, Java, Lua, Kotlin
Frameworks	NestJS, Prisma, zod, GraphQL, Spring Boot, Spring Data JPA, Bucket4j, Spring Cloud Gateway
Databases	PostgreSQL, TimescaleDB, MongoDB, Redis
Infrastructure	Docker, Caddy, node-pg-migrate, GitHub Actions, Flyway, Cloudflare, Toss Payments, OpenTelemetry, Jaeger
Messaging	Socket.IO, BullMQ, MQTT, Apache Kafka, WebSocket (STOMP)
Testing	Jest, Vitest, k6, JUnit 5, Testcontainers
Frontend	React, Vite, Tailwind CSS, Recharts, Leaflet, Next.js
Hardware	ESP32

Experience

Backend Developer (remote, Tashkent)

2024.09 - 2025.08

Sigma Software

- Order list API p95 1.4 s → 210 ms: added a composite index and merged N+1 queries into a single JOIN (Grafana p95 before/after)
- Redis cache cut repeated read queries to the primary database by about 45% (pg_stat_statements calls before/after)
- Took part in a Primary-Standby HA setup on streaming replication; failover to the standby completes in 5-10 s
- Designed order and invoice APIs for a European client in Java/Kotlin and Spring Boot, and introduced versioned Flyway migrations

Backend Developer & PostgreSQL DBA (remote, Tashkent)

2022.10 - 2024.05

LeverX Group

- Completed a PostgreSQL 13 → 15 major upgrade with logical replication and zero downtime
- Built an integration service for data exchange between internal systems, removing manual integration work
- Added pagination and batch export to large reports so they are processed in chunks without overloading the server
- Wrote unit and integration tests wired into CI, and optimized slow database queries

Backend & Database Developer (remote)

2021.11 - 2025.09

Freelance

- UAE, Manufacturing · 2021.11 - 2022.04 (6 months), Report load time 12 s → 0.8 s (~15x); overall database query load dropped
- Hong Kong, Restaurant / F&B · 2023.01 - 2023.04 (4 months), Unneeded superuser connections 7 → 0; restore from backup tested at under 15 minutes
- Bangladesh, AI services · WellDev Bangladesh Ltd. · 2025.03 - 2025.09 (6 months), INSERT p95 320 ms → 45 ms (~7x); active data 180 GB → 22 GB

Projects

PG Insight

2026.08 - 2026.10

An open multi-tenant monitoring service: anyone signs up, adds their own PostgreSQL and sees only the databases they added

NestJS, TypeScript, Prisma, PostgreSQL, TimescaleDB, Socket.IO, React, Vite, Tailwind CSS, Recharts, Jest, Vitest, Docker, Caddy

- Tenant isolation: a global guard checks the owner on every `:targetId` route and returns 404, not 403, for someone else's target; adding an unguarded mutating route fails a test
- SSRF protection: every IP in the DNS answer is checked and the connection is pinned to the verified IP (keeping TLS SNI); production rejected platform-db, 169.254.169.254 and 127.0.0.1

pginsight.javohir.dev | github.com/kicheol83/pg-insight

LedgerCore

2026.08 - 2026.10

A PostgreSQL double-entry ledger where money is never created or lost and no transfer is ever applied twice

NestJS, TypeScript, PostgreSQL, node-pg-migrate, zod, React, Vite, Vitest, k6, Docker, Caddy, GitHub Actions

- Correctness under concurrency: ordered row locks gave 0 deadlocks and 0 faults across 12/12 load runs, and a reversal race was found and fixed with a test
- Invariants in the database: debits = credits is checked by a trigger at COMMIT, with imbalance 0 both under load and in a database restored from backup

ledger.javohir.dev | github.com/kicheol83/ledger_core

Smart Farm

2026.01 - 2026.10

Real-time greenhouse IoT monitoring and automatic irrigation that closes the loop from ESP32 sensors through MQTT, a pipeline and rules to the pump and back to the device

NestJS, TypeScript, GraphQL, MongoDB, Redis, BullMQ, MQTT, Socket.IO, React, Vite, Leaflet, ESP32, MicroPython, Jest, Docker, Caddy

- Proving with measurements that IoT data is not lost: 24/24 through a 132 s MongoDB outage and 7/7 through a 20 s backend stop; the losses on the old code (1 of 12, 1 per deploy) were measured first
- A security audit and fixes: more than 12 issues, including ADMIN escalation at sign-up, arbitrary field writes, cross-member access, unlimited OTP guessing and a hard-coded JWT key, closed after first checking production for abuse

farm.javohir.dev | github.com/kicheol83/smart-farm

FlashDrop

2026.09 - 2026.10

First-come coupons with zero over-issue: four concurrency strategies implemented and measured with k6

Java, Spring Boot, Spring Data JPA, PostgreSQL, Redis, Lua, Flyway, Bucket4j, JUnit 5, Testcontainers, k6, Docker, Caddy, Cloudflare

- Proving the problem with numbers: the no-lock version issued 300 coupons instead of 50 and remaining_quantity stopped at 46, not 0
- Three correct solutions: the other strategies let exactly 50 of 300 requests through, with exactly 50 rows in the database

flashdrop.javohir.dev | github.com/kicheol83/flashdrop

LiveCommerce

2026.09 - 2026.10

A flash-sale platform for live broadcasts: 1,000 purchase attempts for 100 items, zero oversells

Kotlin, Spring Boot, Spring Cloud Gateway, Spring Data JPA, PostgreSQL, Apache Kafka, WebSocket (STOMP), Flyway, Toss Payments, OpenTelemetry, Jaeger, Next.js, TypeScript, Tailwind CSS, JUnit 5, k6, Docker, Caddy

- Correctness: 1,000 attempts for 100 items, repeated three times, with 0 oversells, 0 payment/sales mismatches and 0 HTTP errors; five invariants checked after every run
- Fault tolerance: stopping Kafka or the payment database for real lost 0 events and charged nobody twice (chaos tests)

live.javohir.dev | github.com/kicheol83/livecommerce-saga

Education

Seowon University

2022.08 - 2026.08

Bachelor of Business Administration, Global Business

MIT Group (DevEx Academy)

Backend development bootcamp

Languages

- Uzbek: Native
- Korean: TOPIK II Level 6, fluent at work
- English: Reads technical documents and follows discussions; basic speaking
- Russian: Understands; simple conversation