

Krishna Adhikari

SENIOR FULL STACK ENGINEER

thekadhikari@gmail.com · +977 9866544323 · krishna-adhikari.com.np · github/linkedin: @thekadhikari

PROFILE

Backend-focused full-stack engineer, currently **Engineering Lead at SoftUp** and still in the codebase every day. NestJS, Node.js, TypeScript, PostgreSQL and Redis, with most of my attention on the parts that are expensive to get wrong — data models, API contracts, service boundaries, and the failure paths nobody writes tickets for.

I've led a four-surface product platform from architecture through to on-call, shipped a full e-commerce system solo, and owned the data model behind a large enterprise ERP. Recently that has extended into production AI: retrieval over pgvector, MCP tool servers, and agent loops allowed to write only under explicit scope, confirmation and budget limits.

I'm drawn to problems that are ambiguous before they're difficult — an unclear domain, a schema that has to survive years of edits, a system that fails quietly rather than loudly. I lead through design and review rather than process, and I'd rather ship something the next engineer can safely change than something only I understand.

EXPERIENCE

SoftUp — Dallas, Texas, US · Remote

May 2023 — Present

Engineering Lead May 2025 — Present · *Sr. Full Stack Engineer* May 2023 — May 2025

- Lead the engineering team — architecture standards, system-design reviews, code review and mentoring — while staying hands-on on the backend daily.
- **Led a four-surface product platform** — public web, iOS and Android, a professional portal and an admin console — owning the back-end, the domain engine, the infrastructure and the release process across four codebases, with roughly 450 documented endpoints over ~50 modules so client teams built against a contract rather than guessing at shapes.
- Kept the correctness-critical domain logic in **framework-free packages** with no database, ORM or environment coupling, and enforced that boundary in CI rather than in review, so the most important code in the product has the fastest and most trustworthy tests in it.
- Backend architecture on **NestJS** — REST and GraphQL surfaces, BullMQ workers for anything that must survive a restart, Redis for cache and pub/sub, WebSockets for realtime — across PostgreSQL, MySQL and MongoDB.
- **Zero-downtime releases by design**: replicas roll in sequence behind health gates, migrations and builds run before anything is swapped, and a failed gate rolls back automatically. Staging and production run the identical pipeline.
- **Own the AI platform** across product backends: MCP tool servers, a provider-portable plan→act→observe agent loop that behaves identically across Gemini, OpenAI, Anthropic and DeepSeek, and retrieval over **pgvector** that degrades to lexical search when embeddings are unavailable.
- Designed the governance layer for agents that can write — scope gating, plan-and-confirm, hard stops on destructive operations, transcripts and budgets.

Websoft Technology Nepal — with sister company RS Tech · Pokhara, Nepal

Aug 2022 — May 2023

Full Stack Developer & Project Manager

- **Led the architecture and owned the complete data model** for a large internal ERP used by a construction and property development group in South Asia — delivery orders, purchase orders, receipts, bank account management, loans and employee management. Made the normalisation and performance calls, and guided the web team through implementing against that design.
- Built that platform's notification service in Node.js and Express — templating, delivery retries, and the deduplication rules that stop a retry from becoming a second message to a real person.
- Built backend services for e-commerce and social products in Node.js, Spring Boot and Spring Framework with Spring Security, shipped cross-platform mobile in Flutter, and owned MySQL and Redis performance: query plans, index design, and the cache layer in front of the hot paths.
- Provisioned and operated AWS for an overseas client — EC2, VPC, RDS, S3 and SES — and built the GitHub Actions pipelines that deployed to it.
- Held delivery ownership across both companies — scoping, estimation, sequencing and client communication — alongside the engineering work, and mentored junior developers on backend patterns, REST design and database structure.

Informatics College Pokhara — Pokhara, Nepal · Part-time

Oct 2023 — Jun 2024

Final Year Project Supervisor

- Supervised final-year capstone projects end to end — API design, backend architecture, data modelling and deployment — for students building their first production-shaped system.
- Pushed the work from "it runs on my laptop" to "it holds up": code review, error handling, environment configuration and real deployment, which is where most academic projects quietly fall over.
- Taught diagnosis rather than answers — how to read a stack trace, isolate the failing layer, and form a hypothesis before changing code. Most students arrived able to write code and unable to debug it.

- Built full-stack applications on Node.js and Express backed by Firebase and Cloud Firestore, alongside backend services in Java and cross-platform mobile in Flutter.
- **Packaged recurring functionality into reusable internal modules** — authentication flows, file handling, shared API clients and common UI — so the team stopped rewriting the same three things at the start of every project.
- Took ambiguous requests and broke them into work a small team could actually ship, which usually meant asking what a feature was for before asking how to build it.

SELECTED PROJECTS

Joishi joishi.com · iOS, Android, Web · project lead
Nepal-first calendar and Vedic astrology platform — public web app, iOS and Android apps, an astrologer portal and an admin console, over a computation engine written from scratch. I led the project and owned the backend, the engine, the infrastructure and every deployment: four codebases, **1.2M+ lines**, shipping since 2023. The engine is deliberately framework-free and tests without a database or a web server, with CI failing the build on any import that would break that boundary. Four locales designed in from the start — English, Nepali, Hindi and Sanskrit — plus durable queues, WebSocket realtime, live market rates and payments. Releases roll two replicas in sequence behind health gates and roll back on their own if a gate fails. The AI assistant sits on top, with Gemini context caching keyed by content hash.

OpenBoxZone openboxzone.com · solo build
Live e-commerce platform, built and shipped solo — catalogue and product models, inventory and stock alerts, pricing and promotions, cart and checkout, orders, payments, shipping, returns, warranty, trade-in, B2B accounts, sourcing and marketplace channels, on NestJS with Prisma, Redis, BullMQ and CASL permissions. On top sits a product composer: the model researches an item and returns a schema-shaped draft, then a **deterministic materialiser with no LLM in it** creates the product, its variants and its seeded inventory through the same catalogue services a human goes through — so a composed listing is indistinguishable from a hand-made one. Anything missing an image or stock is held as a draft with reasons attached rather than auto-published. The model proposes; the platform still decides.

Bishwopatra bishwopatra.com · solo build
Nepali news and publishing platform across three codebases — a NestJS API, a Next.js newsroom admin and the public portal, all three mine. The newsroom is the real product: nine editorial roles, a full story lifecycle with an optional copydesk, and CASL authorisation enforced in the API rather than trusted from the client, across **283 documented operations**. It sells white-label, so the platform is config-over-code — a layered theme registry with draft-mode preview, a clone kit that stands up a client’s own deployment, social syndication, and passkey sign-in. The AI operator layer exposes those same 283 operations as MCP tools called *as the signed-in user*, so permissions stay correct by construction — reads automatic, writes confirmed, destructive operations stopped.

@teispacenext-themes npm · ~11K weekly downloads · MIT
Drop-in replacement for the original **next-themes**, which stopped shipping releases in early 2025 while tens of thousands of apps still depended on it. The public API matches, so migrating is a one-line dependency change. Internally it is rebuilt: the no-flash head script is a typed function serialised at render time rather than a build-time literal, and hybrid storage mirrors the theme into a cookie as well as local-Storage so a server component can resolve it in the root layout — moving the flash window from “until hydration” to never. Plus native view transitions and fully typed themes.

@teispacenext-maker npm · 73 releases · MIT
Scaffolding CLI backed by a manifest of every file it writes. That manifest makes two hard things possible: **doctor** reports how far an existing install has drifted from the current template, and **remove <feature>** cleanly reverses a feature it previously added — the part most starters can't do after week one.

EDUCATION

London Metropolitan University — studied at Informatics College Pokhara

BSc (Hons) Computing

Aug 2019 — Dec 2022

- Coursework included software engineering, data structures and algorithms, web development, cloud computing, mobile app development and artificial intelligence.

TECHNICAL SKILLS

LANGUAGES	TypeScript · JavaScript · Java · Dart · SQL
BACKEND	NestJS · Node.js · Express · Spring Boot · REST · GraphQL · WebSockets · microservices · event-driven architecture
DATA	PostgreSQL · pgvector · MySQL · MongoDB · Redis · Prisma · Firestore · schema design & normalisation
AI / LLM	RAG · embeddings · vector search · MCP · agent orchestration · prompt engineering · context caching · Gemini / OpenAI / Anthropic / DeepSeek APIs
FRONTEND	React · Next.js · Tailwind CSS · Redux Toolkit · Flutter
INFRA	AWS (EC2, VPC, RDS, S3, SES) · Docker · nginx · GitHub Actions · BullMQ · DigitalOcean · Linux
PRACTICE	system design · data modelling · code review · technical mentoring · CI/CD · observability