

Md. Al Mamun Mim

Senior Full-Stack Engineer

Dhaka, Bangladesh (UTC+6) · Remote · md.almamun.mim.dev@gmail.com · +880 1770540432
[almamun.codes](#) · github.com/MamunCrafts · linkedin.com/in/mamuncrafts

Summary

Senior full-stack engineer with 4 years at one company, promoted from intern to senior on a consumer social-commerce product with 600K+ registered users. I own the GraphQL API and data layer, and lead 5 engineers day to day on a team of 20.

My strongest work has been on the parts of a system that break quietly: query plans that scan whole collections, pipelines fed by third-party APIs that change without warning, and infrastructure whose cost scales with the wrong thing. I moved our chat system off Firestore onto in-house WebSockets when per-operation billing meant the product got more expensive as it got more popular.

Looking for a remote senior role with harder scale problems and a stronger engineering culture than I can find locally.

Experience

Talent Pro (Simura Group) — Dhaka, Bangladesh

Aug 2022 – present · promoted three times

- **Senior Software Developer** · Jan 2026 – present
- **Software Engineer** · Dec 2023 – Jan 2026
- **Junior Software Engineer** · Feb 2023 – Dec 2023
- **Software Engineer, Intern** · Aug 2022 – Feb 2023

Primary product: **Fanfare** — an Android social commerce app combining a social feed, video contests, a gamified points system, and in-app purchasing. Sells own-inventory products alongside brand advertising and collaborations. **1M+ downloads** · **600K+ registered accounts** · **~30K monthly active users** · **~1,000 video uploads per day**.

Leadership (senior, Jan 2026–present)

- Team of 20 engineers; **5 engineers report to me** for day-to-day work — task assignment, PR review, and technical direction
- Own architecture decisions on new services; mentor junior engineers on testing and API design

Architecture and cost

- **Rebuilt chat in-house on WebSockets, migrating off Firebase/Firestore.** Firestore bills per operation, and chat generates operations relentlessly — every message, presence update, and listener — so cost scaled with engagement rather than with revenue. Since a pricing model cannot be optimised away, I replaced it. Used a **Redis pub/sub adapter** so messages reach clients connected to any server instance, and brought message ordering and delivery under our control.
- Chose **MongoDB** for the social product (flexible document shapes for feed, posts, comments) and **PostgreSQL** for the data platform, where pipeline state and relational integrity matter more than schema flexibility.

Performance

- Diagnosed endpoints exceeding **5 seconds** using MongoDB **explain()**, finding full collection scans where documents examined far exceeded documents returned. Root cause was filtering and sorting on an unindexed field — acceptable at low volume, linear-cost as the collection grew. Added covering indexes (COLLSCAN → IXSCAN) and redesigned the schema to remove the expensive query shape. Production monitoring showed roughly **45% lower latency and 2× throughput** on the core API.
- Added **slow-query alerting** so degradation surfaces from monitoring rather than user reports.
- Introduced a **Redis caching layer** on read-heavy endpoints, and reduced API payload sizes to cut mobile data use for users on slow networks.

API and product

- Designed and shipped the **GraphQL schema** in NestJS serving feed, profiles, comments, and real-time notifications

- Built **RBAC** workflows and enabled e-commerce checkout
- Took critical service logic to **100% unit test coverage** with Jest
- Ran the **video transcoding pipeline** at ~1,000 uploads/day — handling queue backpressure, silent job failures, storage growth across renditions, and format-specific failures
- Built and maintained the internal **admin console** (Next.js, TypeScript, Redux, MUI) for the operations team; code-splitting and dependency pruning cut initial bundle size ~40% and load time ~25%

Data platform

- Built **ConversaAI**, an internal ETL / Reverse ETL platform integrating **Shopify, WooCommerce, Zoho, Brevo, Mailchimp, QuickBooks, and Wix Commerce**. Handled the failure modes third-party APIs guarantee: schema changes, rate limiting mid-sync, malformed input, and duplicate writes on retry — the last of which is an idempotency problem, not a retry problem.
- *Next.js · NestJS · PostgreSQL · Python · Airbyte · AWS EC2*

Engineering practice

- Implemented **CI/CD in GitHub Actions**; managed AWS infrastructure (EC2, S3, CloudWatch) with Docker and Nginx for containerised deployments
- Improved Lighthouse mobile usability from **70 to 92** on company web properties
- Contributed to sprint planning, architecture review, and code review on a cross-functional Agile team

Technical skills

Languages: TypeScript · JavaScript · Python (ETL) **Backend:** Node.js · NestJS · GraphQL · REST · Web-Sockets **Frontend:** React · Next.js · Redux · MUI · Tailwind CSS **Databases:** MongoDB (schema design, indexing, query plans) · PostgreSQL · Redis · Firebase **Infrastructure:** AWS (EC2, S3, CloudWatch) · Docker · Nginx · GitHub Actions **Testing:** Jest (unit and integration) **Tooling:** Git · Jira · PostHog · Google Analytics · Airbyte · Agile/Scrum

Certification: MongoDB Certified Associate Developer

Side project

RetryDock — self-hosted, multi-tenant webhook reliability platform, in active development.

Sits between a webhook producer (Stripe, GitHub, Twilio, any HTTP source) and its consumer. Persists every inbound event before processing, delivers to multiple destinations, retries on a backoff schedule with jitter, parks exhausted deliveries in an inspectable dead-letter queue, and supports replay without mutating history.

I wrote the full requirements specification before any code, because the retry semantics and delivery-state model needed to be settled first. It exists because I kept rebuilding the same reliability layer by hand while fighting flaky third-party APIs on ConversaAI.

Education

BSc in Information and Communication Engineering — University of Rajshahi · 2017 – 2022

Achievements

- **Tech Genius Award**, Talent Pro
- Competitive programming team contestant and leader, ICE Dept. — **600+ problems on Codeforces**, 100+ on LeetCode; IEEE Xtreme participant, 5th place domestic
- 3rd place, Rajshahi University National Science Fiesta 2021

Languages: English (professional working) · Bengali (native) · Hindi (professional working)

Available with 1 month notice. Working remotely from UTC+6 — full overlap with EU hours, and US East mornings.