# Nguyen Thang Minh Dat

Fullstack Developer (Java & React/Angular)

0867684603 nguyenthangdat84@gmail.com github.com/nguyenthangminhdat84-glitch

Thu Duc City, Ho Chi Minh City

## SUMMARY

**Fullstack developer** focused on **Java Spring Boot microservices**. Designed, built and deployed a **12-service sports platform** that handles real bookings and payments, using **Kafka**, **Camunda 8**, **Keycloak** and Redis, with **CI/CD**, **monitoring** and nightly backups. Comfortable owning a feature from database schema to **Angular** or **React UI**, and keeping it reliable once it is live.

## EXPERIENCE

**FPT IS (Ho Chi Minh)** May 2026 - Aug 2026

**Fullstack Developer Intern**

Contributed to the development of enterprise **Banking System (**FIS Bank). Participated in implementing business **workflows, authentication, reporting, and microservices** while collaborating with cross-functional teams in an **Agile** environment.

- Developed and maintained backend services using **Java Spring Boot** following **Microservices Architecture**.
- Built and consumed **RESTful APIs** with **OpenFeign** for inter-service communication.
- Integrated **Keycloak** for authentication, authorization, and role-based access control **(RBAC)**.
- Developed business workflows using **Camunda BPM** by implementing BPMN processes and service tasks.
- Implemented reporting features using **JasperReports** to generate PDF/XLSX reports for banking operations.
- Participated in developing frontend modules using **Angular with Micro Frontend Architecture**.
- Containerized applications with **Docker** to ensure consistent development and deployment environments.

## EDUCATION

**Industrial University of Ho Chi Minh City** Sep 2022 - Mar 2027

Software Engineering; Major GPA: 3.32/4.0

- Relevant coursework: Data Structures and Algorithms, Database Design, Web Development

## CERTIFICATIONS

**Language:** English - **TOEIC 765/990** Feb 2026

## SKILLS / TOOLS

| | |
|---|---|
| **Backend:** | Java 21, Spring Boot 4, Spring Security (OAuth2/JWT), Spring Data JPA, Spring Cloud (Gateway, Eureka, OpenFeign), WebSocket/STOMP. |
| **Messaging & Workflow:** | Apache Kafka (outbox, retry/DLT), Camunda 8 (Zeebe), Keycloak (custom SPI). |
| **Frontend:** | TypeScript, Angular 21, React 19 / Next.js, Tailwind CSS, shadcn/ui. |
| **Database & Caching:** | PostgreSQL, Redis, MongoDB, Liquibase. |
| **Devops:** | Docker / Compose, GitLab CI, GitHub Actions, Jenkins, Jib, AWS (S3, Elastic Beanstalk), Cloudflare (R2, Tunnel), Prometheus / Grafana / Loki, k6. |
| **Concepts:** | OOP, RESTful APIs, Basic Microservices, Event-Driven Architecture, Clean Architecture. |

## PROJECTS

**GOAT SPORT - SPORT PLATFORM** Aug 2026 - Nov 2026

**Role: Fullstack Developer | Team of 2 - Java 21, Spring Boot, Spring Cloud (Gateway, Eureka), Keycloak (Custom SPI), Kafka, Camunda 8, PostgreSQL, Redis, Liquibase, Angular 21, Docker, GitLab CI, Jenkins, Cloudflare R2.**

- **Architected** 12 microservices behind Spring Cloud Gateway and Eureka. The gateway validates JWTs from HttpOnly cookies, enforces path-based RBAC, revokes tokens via a Redis blacklist and forwards trusted user-identity headers downstream.
- **Prevented** double bookings with pessimistic row locks and 15-minute slot holds. **Integrated** payOS payments with checksum-verified, idempotent webhooks and a transactional outbox that publishes payment events to Kafka.
- **Modeled** the booking lifecycle and cancellation/refund flow as Camunda 8 BPMN processes with Zeebe job workers.
- **Built** event-driven flows on Kafka with retry topics and DLT. A custom Keycloak Event Listener SPI syncs users to the backend, and storage consumers running on Java 21 virtual threads move and clean up files in Cloudflare R2 via presigned-URL uploads.
- **Delivered** with GitLab CI → Jib → Docker Hub → Jenkins onto a Docker Compose VPS behind Cloudflare Tunnel. **Monitored** with Prometheus, Grafana and Loki, plus nightly backups to R2.
- **Engineered** 3 Angular apps (player, admin, auth) using Clean Architecture/DDD, with real-time chat and notifications over STOMP WebSocket.

**Github:** https://github.com/Goat-Sports-Group
**Demo:** https://youtu.be/NBRLJoYHHsE?si=_cWC_RXiiGc824vN

**GOAT SOCIAL - SOCIAL NETWORKING PLATFORM** Nov 2025 - Mar 2026

**Role: Fullstack Developer | Team of 5 - Java 21, Spring Boot, Next.js 16 (React 19), PostgreSQL, AWS (DynamoDB, S3, Elastic Beanstalk), Redis, Liquibase, Agora RTC, WebSocket (STOMP/SockJS), Docker, CI/CD.**

- **Architected** a hybrid database architecture combining PostgreSQL (via Spring Data JPA) for structured relational business data and AWS DynamoDB to store and query high-volume, real-time messaging data.
- **Implemented** centralized Identity & Access Management (IAM) using Spring Security OAuth2 Resource Server to validate JWTs from client cookies via a custom Bearer Token Resolver, establishing strict Endpoint-level authorization and integrating a custom Redis-based Token Blacklist filter for secure session revocation/logout.
- **Designed** a distributed real-time presence & notification engine leveraging Redis Keyspace Notifications (listening to key-expiration events), dynamically updating user online/offline status, executing delayed/scheduled notification workflows, and broadcasting events via WebSockets (STOMP & SockJS).
- **Integrated** Google Gemini AI (Google GenAI SDK) to build intelligent features: automated resume/CV evaluation (parsing files to return scores, strengths, weaknesses, and key skill gaps), custom role-based smart chat assistants, automated blog tag generation, and inline message translation.
- **Developed** a high-performance Next.js 16 client (React 19) styled with TailwindCSS v4, shadcn/ui, and Framer Motion micro-animations,managing global state using Redux Toolkit/Redux Persist, and integrating Agora RTC Web SDK for real-time video/audio group calling directly within chat rooms.
- **Established** an automated CI/CD pipeline via GitHub Actions to build, compile, and containerize the Java 21 Spring Boot application using Docker, publishing images to Docker Hub, and executing blue-green style deployments to AWS Elastic Beanstalk.

**Github:** https://github.com/trgcghai/Goat-JobHunter-Fe | https://github.com/trgcghai/Goat-JobHunter-Be