

TAWHID AHMED

Full-Stack Software Engineer | Java · Spring Boot · Vue 3 · AI / LLM Engineering

tawhidahmed98@gmail.com | +880 1407 835375 | Dhaka, Bangladesh | [LinkedIn](#) | [GitHub](#)

PROFILE SUMMARY

Full-stack software engineer with over two years of professional experience delivering enterprise fintech platforms across **capital markets, cross-border payments**, and multi-tenant SaaS, and author of a **peer-reviewed journal article** and a **2025 arXiv preprint**. Works end to end in **Java 21 / Spring Boot** microservices and **Vue 3 / Quasar** frontends, and owns production **LLM features** — **tool-calling agents, RAG**, and streaming assistants — embedded directly into business applications, from database schema through API design to the user interface.

TECHNICAL SKILLS

Languages: Java 21, Python, TypeScript, JavaScript, SQL

Backend: Spring Boot 3, Spring Cloud, Spring Modulith, Spring Security / OAuth2 (JWT), Spring Data JPA, FastAPI, REST APIs, Microservices

Frontend: Vue 3 (Composition API), Quasar, Pinia

Data & Messaging: PostgreSQL, Redis, RabbitMQ, WebSocket / STOMP

AI & ML: LangChain, LangGraph, RAG, FAISS, Chroma, Sentence-Transformers, OpenAI / Azure OpenAI, Groq, Ollama, MCP, tool-calling agents, prompt engineering, TensorFlow, PyTorch, Scikit-learn, OpenCV, Transformers (BERT, CNN, BiLSTM)

Cloud & DevOps: AWS, Docker, Docker Compose, Git, CI/CD, OpenAPI / SpringDoc

AI-Assisted Development: Cursor and Claude for pair-programming, refactoring, and test scaffolding, with human review before merge

PROFESSIONAL EXPERIENCE

Junior Software Engineer

Sep 2024 – Present

Proficient Information Systems | Dhaka, Bangladesh

- Deliver full-stack features across three ongoing enterprise platforms — a capital markets OMS, a cross-border remittance wallet, and an internal task-management SaaS — working from database schema through Spring Boot services to the Vue 3 interface, and **own the embedded AI / LLM features end to end**, with tests, API documentation, and code review throughout.

Capital Markets Order Management System (OMS) & Trading Platform

Spring Boot microservices platform for Dhaka Stock Exchange brokerage operations, covering the full trading lifecycle: order entry, pre-trade risk controls, **FIX** exchange execution, and real-time market data.

- Developed order-lifecycle, client-management, and pre-trade restriction endpoints across the OMS and CAT services within a **nine-service microservices** architecture, and added message consumers handling execution reports and order-status events.
- Built the trading terminal front end — order entry, blotter, watchlist, and portfolio — streaming live order, execution, and market-data updates into the UI over **WebSockets**.
- Owned the platform's AI features: a **conversational trading assistant** with LLM tool-use over **MCP** and streamed responses, and per-symbol technical analysis computing 30-day RSI, SMA, and VWAP indicators.

Stack: Java 21, Spring Boot 3, Spring Cloud, RabbitMQ, PostgreSQL, Redis, QuickFIX/J, STOMP WebSockets, OAuth2 / JWT, Azure OpenAI, MCP, Vue 3, Quasar, Pinia, Docker

Cross-Border Remittance & Digital Wallet Platform

Fiat ↔ USDC remittance platform integrating the Stellar blockchain and the MoneyGram anchor, backed by a **double-entry ledger** and a KYC compliance workflow.

- Implemented REST endpoints and service-layer logic across the user, transaction, KYC, and ledger modules, contributing to a double-entry ledger with idempotent postings and a transaction state machine spanning 20+ deposit, withdrawal, transfer, hold, and reversal states.
- Built the MoneyGram deposit and withdrawal integration over **Stellar SEP-10 / SEP-24**, with scheduled jobs polling provider status and reconciling inbound USDC payments streamed from Horizon.
- Developed the wallet front end — authentication, dashboard, transactions, and KYC — including secure identity-document upload through backend presigned URLs.

Stack: Java 21, Spring Boot 3.5, Spring Modulith, PostgreSQL, Stellar SDK / Horizon, OAuth2 / JWT, Vue 3, TypeScript, Quasar, Pinia, Docker

SuperCards — “Task AI Coach” In-App AI Assistant (designed and built end to end)

Conversational assistant for a multi-tenant task-management SaaS that grounds an LLM in live backend data — giving management insight into task states and workload distribution, and helping individual users track and improve their own performance.

- Designed and built the assistant as a FastAPI + LangChain / **LangGraph** microservice that grounds LLM answers in live REST-API data through a **bounded tool-calling loop**, removing an entire class of hallucinated figures.

- Implemented **nine domain tools** behind a view-scoped registry so each page exposes only the relevant tools, with deterministic pre-LLM logic and scope labelling to keep reported totals accurate.
- Built **tenant-scoped RAG** for tag resolution using per-tenant vector indexes with hybrid literal and semantic retrieval, and shipped the chat widget with persisted per-view threads and server-driven charts.
Stack: Python, FastAPI, LangChain, LangGraph, FAISS, Sentence-Transformers, Groq / Ollama, Laravel REST API, Vue 3, Quasar, Pinia, ApexCharts

Programmer Intern

May 2024 – Aug 2024

Proficient Information Systems | Dhaka, Bangladesh

- Built responsive web applications with Vue.js, Quasar, and JavaScript, improving UI performance and usability.
- Implemented features under senior review, debugged cross-device frontend issues, and took part in code reviews and team design discussions.

Industrial Trainee

Nov 2020

BRACNet Limited | Dhaka, Bangladesh

- Completed a 15-day industrial training covering data-centre services, IoT, cybersecurity, ISP technology, and cloud applications.

EDUCATION

M.Sc. in Computer Science & Engineering

2023 – Present | CGPA 3.58

Bangladesh University of Engineering and Technology (BUET) | Dhaka — coursework completed 2024, thesis in progress

B.Sc. (Eng.) in Information & Communication Engineering

2017 – 2022 | CGPA 3.66

Noakhali Science and Technology University (NSTU) | Noakhali

PUBLICATIONS

- [1] A Unified BERT-CNN-BiLSTM Framework for Simultaneous Headline Classification and Sentiment Analysis of Bangla News, 2025. DOI: [10.48550/arXiv.2511.18618](https://doi.org/10.48550/arXiv.2511.18618)
- [2] Ahmed, T., Uddin, M., Khan, M.A.R. and Hasan, A.R.M. Offline Handwritten Character Recognition Including Compound Character from Scanned Document. Asian Journal of Research in Computer Science, 14(4), 119–129, 2022. DOI: [10.9734/ajrcos/2022/v14i4297](https://doi.org/10.9734/ajrcos/2022/v14i4297)

PERSONAL PROJECTS

Bangla OCR — Offline Handwritten Character Recognition [GitHub](#)

- Built an end-to-end deep-learning system for offline recognition of handwritten Bangla characters, including compound characters, covering image preprocessing, text segmentation, and **CNN-based classification**.
- Reached **~96% validation accuracy**, evaluated with precision, recall, F1-score, and confusion matrices.

Stack: Python, TensorFlow, Keras, OpenCV, NumPy, Pandas, Matplotlib, Scikit-learn

arXiv RAG Assistant — Research Paper Q&A with Inline Citations [GitHub](#)

- Built a RAG service that ingests arXiv papers and answers questions with inline source citations, spanning a FastAPI backend, an ingestion pipeline (PDF parsing, token-aware chunking, embedding, vector indexing), and a Streamlit chat interface.
- Implemented **hybrid retrieval** combining BM25 sparse search with dense embeddings and a cross-encoder reranker, plus a guardrail that declines to answer when retrieval confidence is low.
- Designed a **pluggable LLM** provider abstraction covering OpenAI, Anthropic, Hugging Face Inference, and local Ollama, with SSE token streaming, per-request token and cost accounting, and optional tracing.
- Built an **evaluation harness** reporting hit-rate, recall@k, and MRR across dense, hybrid, and reranked ablations; the codebase carries 18 pytest modules, ruff and mypy gates, and a Dockerised GitHub Actions CI pipeline.

Stack: Python, FastAPI, Streamlit, Sentence-Transformers, Chroma, BM25, Langfuse, Docker Compose, GitHub Actions, pytest