DAO TRONG NGUYEN

BACKEND & SYSTEMS ENGINEER

CONTACT

+84 879 914 364

nguyen10t2lhp@gmail.com

github.com/nguyen10t2

EDUCATION

Hanoi University of Science and Technology
Computer Engineering
CPA: **3.56**
GPA: **3.89**
2023 - Present

Le Hong Phong High School for the Gifted
Mathematics Major
2020 - 2023

EXPERIENCE

VNet Joint Stock Company
Backend Developer Intern
May 2024 – Oct 2024 (5 mos)

LANGUAGES

English
TOEIC 600 (R-L)

CORE SKILLS

Languages:
Rust, C/C++, Python (Intermediate)
TypeScript, Java (Basic)

Systems & Networking:
Linux, Network Routing (Netplan, iptables), RTOS (ESP32), QUIC (Quinn)

Backend & Databases:
Tokio, Axum, Bun, Node, ElysiaJs, PostgreSQL, Redis, MQTT

Systems Thinking:
Concurrency, Low-level Memory, Zero-Copy I/O, Cache Optimization, DSP

OBJECTIVE

Aspiring Backend & Systems Engineer with a deep focus on distributed systems, high concurrency, and low-latency network protocols. Passionate about leveraging systems programming (Rust, C/C++) to architect high-throughput data pipelines, network infrastructure, and resilient production backend services.

PROJECTS & TECHNICAL RESEARCH

QUIC-Chat (vchat) - High-Performance E2EE Messaging *VNet JSC*
Tech: Rust, QUIC (Quinn), Tokio, Rustls, SQLite, X25519/ChaCha20

- Architected E2EE messaging over single QUIC/UDP connections, multiplexing control, chat, presence datagrams, and file transfer with stream-priority QoS, achieving >**150 MB/s** with zero head-of-line blocking.
- Engineered actor-based Tokio lifecycles with zero-copy `Bytes` slicing, backpressure, and Double-Ratchet/X25519 encryption.

bamboo-core - Vietnamese IME Engine Core *Open Source / Crates.io*
Tech: Rust, C-FFI, WebAssembly, DFA, Zero-Allocation

- Authored and published **bamboo-core** on crates.io, an ultra-fast Vietnamese Input Method Engine (IME) core in Rust supporting Telex, VNI, and VIQR.
- Designed a hybrid Deterministic Finite Automaton (DFA) state machine and rule-based parser featuring a **zero-allocation hot path** for $O(1)$ syllable-level transformation.
- Engineered cross-platform C-FFI and WebAssembly (WASM) interfaces for frictionless embedding into native desktop, mobile, and browser environments.

Dynomite STM32 - Embedded Bubble-Shooter Game *Academic / Team*
Tech: C/C++, STM32F429 (Cortex-M4), TouchGFX, DMA2D, SDRAM

- Designed the **TouchGFX MVP** architecture and screen-flow state machine, separating UI rendering, game logic, and hardware abstraction.
- Engineered core game physics: **hex-grid** snapping with $O(1)$ LUT neighbor lookup, **BFS Match-3** chain detection, orphan-drop, and sqrt-free squared-distance collision with **EMA** aim smoothing at 60 FPS.
- Optimized CPU-bound hot paths: **sub-stepping** continuous collision to prevent tunneling, **dirty-flag** partial rendering, $O(1)$ cached egg count, LUT-based color mapping, and local grid buffer copies to minimize cache misses.

SplitDebt - Group Expense Splitting Backend *Personal*
Tech: TypeScript, Bun, ElysiaJS, PostgreSQL, Drizzle ORM, Redis

- Built a REST API backend solving the **debt simplification problem**: greedy creditor-debtor matching algorithm minimizes the number of settlement transactions across group members.
- Structured codebase into feature-based modules (users, groups, expenses, settlements) with Strategy pattern for split logic (Equal/Exact/Percentage), JWT + OAuth authentication, Redis-backed rate limiting, and Redis cache invalidation.

AI INTEGRATION IN DEVELOPMENT WORKFLOW

- **Reverse Engineering & R&D:** Leveraged LLMs for parsing complex network protocol specifications (RFCs), hardware datasheets, and low-level code inspection.
- **Automated QA & Testing:** Synthesized automated test suites, concurrency regression tests, and edge-case property-based validation.