

Don (Nethum) Weerasinghe

(832) 276-8234 · nethumweerasinghe.nw@gmail.com · linkedin.com/in/nethum · github.com/nethum529 · nethum.dev

EDUCATION

Texas A&M University

College Station, TX

Bachelor of Science in Computer Science, GPA: 3.6/4.0

Aug 2024 – May 2028

- **Relevant Coursework:** Advanced Data Structures & Algorithms, Systems Programming, Machine Learning, Computer Architecture, Linear Algebra, Discrete Math, Functional & Object-Oriented Programming.

EXPERIENCE

Software Engineering Intern

May 2026 – Aug 2026

Global Shop Solutions

The Woodlands, TX

- Built a token-efficient **Model Context Protocol (MCP) server** for Traveler, enabling approximately **20 employees** to retrieve and update project tickets; over **90%** of ticket updates were completed through the MCP server.
- Co-built Traveler, a **Next.js/TypeScript** project-management platform designed to replace Monday.com; contributed to Scrum Master automations and the sprint feature, scaled to approximately **20 users**, and helped save **\$25K** annually.
- Refactored complex Data Access Managers in customer-facing production code, reducing code volume by over **60%** and cutting data-retrieval latency by **34%**.
- Replaced 5-minute polling with reliable webhook-triggered syncs between **GitHub** main and an internal **Docker/Kubernetes** hosting tool, cutting merge-to-deploy latency by **90%**.

Undergraduate Researcher

Jan 2026 – May 2026

Sketch Recognition Lab

College Station, TX

- Engineered an end-to-end **machine-learning data and evaluation pipeline** for medical-image segmentation, including preprocessing, embedding caching, training, **Dice/IoU/HD95** evaluation, and visualization.
- Ran training and evaluation on Texas A&M's Grace HPC cluster via **A100 GPU** jobs and **Slurm** scripts.

PROJECTS

Seer · Rust, QUIC, Tokio, PTY, Linux, macOS · github.com/nethum529/seer

Aug 2026 – Present

- Built **Seer**, a distributed multiplayer terminal multiplexer in **Rust** (29K lines, 5 crates) that shares live terminal sessions across **Linux** and **macOS**. Each user's shells run on their own machine and survive network loss, broker restarts, and crashes through atomic, **fsync-backed snapshots**.
- Designed a broker-routed peer architecture over end-to-end encrypted **QUIC** with NAT traversal and relay fallback. It sends many streams over one connection, hands off streams with one-time tokens, rate-limits connections per source, and allows only one publisher per user.
- Cut screen-sync bandwidth by **96–99%** by designing a scroll-aware frame-diff protocol, guided by the debug logs and performance probes built into the service.
- Shipped **190+ pull requests in 3 weeks** by running parallel AI coding agents (Claude Code, Codex) in isolated git worktrees. I validated each AI-generated change against strict gates before merge: Clippy with warnings denied, complexity limits, zero dead code, and no unwraps.

Zone · Swift, SwiftUI, Core NFC, Screen Time API, iOS · github.com/nethum529/zone

Oct 2026

- Built **Zone**, a native iOS app in **Swift 6** and **SwiftUI** with six targets that blocks chosen apps through Apple's Screen Time APIs until the user scans their own **NFC tag**. Shipped to public **TestFlight** 3 days after the first commit.

OPEN SOURCE CONTRIBUTIONS

NVIDIA/cudf · Open Source Contributor, 9 merged PRs

Jun – Oct 2026

- Fixed a **GCC 16 build break** in the C++ ORC and Parquet writers and kept decimal128 columns out of an overflow check that would have written wrong ORC statistics.
- Closed a **pandas API compatibility gap** by implementing DatetimeIndex.indexer_between_time for GPU DataFrames, covering time zones, midnight-spanning ranges, and endpoints with **22 tests** checked against pandas.

rapidsai/rmm · Open Source Contributor, 4 merged PRs

Jul – Oct 2026

- Eliminated false **out-of-memory failures** in the C++ GPU memory pool, letting size-capped pools reach **100% of their configured maximum** by returning free blocks upstream before growing, approved by 3 maintainers.
- Fixed **alignment forwarding** in the callback memory resource across the C++, Cython, and Python APIs (11 files), so user callbacks receive and validate the requested alignment.
- Made silent allocation-pointer leaks **compile-time errors** across **22 C++ memory-resource files** by adding `[[nodiscard]]` to `allocate` and `allocate_sync`.

TECHNICAL SKILLS

Languages: Python, C++, Rust, Java, JavaScript, TypeScript, HTML, CSS

Systems & Cloud: Linux, AWS, PostgreSQL, Docker, Kubernetes, Git

ML & Data: PyTorch, scikit-learn, Pandas, NumPy, cuDF

Frameworks & Libraries: React, Next.js, FastAPI, Spring Boot, SwiftUI