Elliot Miller

734-223-7401 | Website | elliotgrantmiller@gmail.com | linkedin.com/in/elliotgmiller | github.com/elliotmiller18

Education

University of Michigan Ann Arbor, MI

Bachelor of Computer Science *Expected May 2027*

- Relevant Coursework: Formal Verification of Distributed Systems, Operating Systems, Web Systems, Computer Organization, Data Structures and Algorithms, Computer Theory, Compilers, Game Development, Programming Languages

Experience

Software Engineering Intern (Rust) Jan 2026 – May 2026

N1 → Founder's Fund *New York, NY*

- Incoming Software Engineering intern where my primary roles will be maintining a TypeScript targeting compiler and writing low-level networking code.

Backend Engineering Intern (Typescript) May 2025 – August 2025

Wise Pelican *Phoenix, AZ*

- Wrote a from-scratch fast JPEG metadata decoder, cutting off over 95% of runtime from the previous implementation by reducing data needed to process per JPEG from megabytes to a single kilobyte
- Decreased average runtime of PDF rendering engine from 10s to 2s by moving image generation to an asynchronous model, allowing smaller requests to be processed, generated, and responded to in the time that larger PDFs were rendering
- Migrated application from Azure to AWS ECS with a Codepipeline CI/CD setup, decreasing average runtime by removing AWS to Azure performance bottleneck in application

projects

GameBoy Emulator (C++) Github

- Simulated all GameBoy-specific hardware components/systems, including internal clock circuits, interrupts, and proprietary graphics
- Wrote a fully accurate Z80 Sharp CPU simulator to run ROMs including sub-instruction timing, hardware bugs, and a pipeline simulator
- Achieved consistent a consistent 300 frames per second using a custom SDL-integrating renderer on an M1 Macbook with integrated graphics

HTTP Server (Rust) Github

- Fully functional HTTP server built on top of the OS TCP layer
- Used the Rust core's type safe synchronization primitives to handle up to 20 concurrent connections

Network File System (C++)

- Created a simulated file system similar to that of the Windows operating system with full crash consistency and atomicity of all write operations, as well as support kernel swap operations
- Allowed for remote read/writes using OS socket/port syscalls

MapReduce Implementation (Python)

- Created an efficient recreation of the MapReduce algorithm using a distributed systems setup
- Wrote communication protocols between the Manager and Worker servers

Miscellaneous

- Was ranked #3 on the Stack Overflow global leaderboards (see my website for proof)
- Volunteer at underprivileged schools to teach coding classes to elementary-aged kids

Technical Skills

Languages: C++, C, Rust, x86_64, Dafny, Java, Python, SQL, OCaml, C#
Libraries/Frameworks: SDL.h, SQLite, PostgreSQL, AWS CDK/SDK, Unity
Developer Tools:Git, AWS, AWS Lambdas, Docker, MacOS, Linux, NVim