

# STONE LIU

[617-792-6757](tel:617-792-6757)   [stoneliucs.github.io](https://stoneliucs.github.io)   [liu.sto@northeastern.edu](mailto:liu.sto@northeastern.edu)   [linkedin/stone-liu](https://linkedin.com/in/stone-liu)   [github.com/stoneliuCS](https://github.com/stoneliuCS)

## EDUCATION

### Northeastern University

Expected May 2026

*Candidate for Honors Bachelor of Science in Computer Science and Mathematics*

*Boston, Massachusetts*

- **Awards:** Northeastern Honors Program, Deans List
- **Relevant Courses:** Algorithms and Data Structures, Object Oriented Design, Software Development, Computer Systems

## TECHNICAL SKILLS

**Languages:** TypeScript/JavaScript, Python, Java, C, C++, SQL, Bash

**Technologies:** Nvim/Vim, Docker, Git, AWS S3 & Lambda, Nix

**Libraries:** Pyspark, Pandas, Polars, React.js, React-Native, Next.js, Vue.js, Nuxt.js

## EXPERIENCE

### Software Engineer Intern, Morse Corp, Cambridge MA

January 2025 - Present

- Built core infrastructure for data analysis pipelines aimed at algorithmic testing and evaluation of object detection models.
- Deployed containerized CV tools, supporting metrics through automating model inferencing and overlaying bounding boxes.
- Architected a custom test harness, utilizing data transformations to maintain invariants about model metrics across pipelines.

### Technical Lead, Generate Product Development Studio, Boston MA

September 2024 - April 2025

- Led 7 software engineers, engineering CI/CD pipelines to containerize deployments, automate builds and integration tests.
- Designed a distributed web server that supported multimedia compression/serving, scheduler services, and client-server auth.
- Continuously promoted learning through test driven development, pair-programming, workshops, and agile methodologies.

### Lead Lab TA for CS2500, Khoury College, Boston, MA

September 2024 - December 2024

- Helped 600+ students by teaching systematic program design with topics including structural recursion and accumulators.
- Held weekly lab sessions for 30+ students, reviewing course fundamentals, design concepts/paradigms, and exam reviews.
- Led engaging office hours, enabling students to proactively reason about self-referential data, and graph/tree algorithms.

### Software Development Intern, Spill Center, Hudson, MA

January 2024 – August 2024

- Tracked over 800,000 cargo tank facilities and tanks by architecting a centralized web application for cargo tank life events.
- Monitored 10,000+ incidents and alert groups by implementing a geospatial alert service through PostGIS spatial queries.
- Utilized database schemas which mapped regulatory reports to geographic areas, enabling for more detailed incident reports.

## PROJECTS

### Fluid-OAS | TypeScript

- *Fluid-OAS* is a functional domain specific language expressing type-safe HTTP APIs through the **OpenAPI** specification.
- Published a TypeScript **DSL** offering compile-time type-checking and modularization for writing **OpenAPI** specifications.
- Created macro-like functions using the TypeScript compiler to generate mixins for code completion and intellisense support.
- Architected a *Fluent API* for the **OpenAPI** specification by representing core JSON schemas as immutable builder objects.

### Dearly | TypeScript, React-Native, Docker, AWS S3/Lambda, Nix, Supabase, PostgreSQL

- Dearly is a private family-sharing app bridging generational gaps and makes staying connected easier and more meaningful.
- Created CI/CD pipelines that rigorously ran over 500 integration tests as well as containerized and deployed docker images.
- Delivered a type-safe **REST** backend using type generation and architected service abstractions over **AWS S3** and **Lambda**.
- Leveraged query caching and image/audio compression, leading to an 80% decrease in API calls and faster loading times.

### Bazaar | Java, Apache Maven, Bash, Google Gson

- Created a distributed trading card game with functional-style Java, where player mechanisms connect over **TCP/IP** protocols.
- Developed a tree-search algorithm that efficiently searched through millions of candidates using DFS and data accumulators.
- Created a robust server component which protected itself from DOS attacks and malformed JSON responses from clients.
- Built a robust testing harness, allowing serialization/deserialization of external JSON data representations for unit testing.