shengh@mit.edu
linkedin.com/in/shengh318/

Sheng Huang

(315) 704-9115

Profile

MIT 2025 B.S. and M.Eng. graduate of Computer Science and Engineering and US citizen with expertise in **full-stack development and distributed systems**, with a strong interest in **backend development**.

Education

Massachusetts Institute of Technology (MIT) — **Cambridge, MA**

M.Eng. in Computer Science — **GPA: 5.0 / 5.0** — **May 2025**

- Computer Networks; Systems Security; Distributed Systems; Robotics

B.S. in Computer Science — **GPA: 4.6 / 5.0** — **May 2024**

- Software Performance Engineering; Computational Architecture; Design of Algorithms; Systems Engineering; Network Security; Machine Learning

Experience

Back-end Software Engineer, Intern — **Capital One** — **Summer 2023 / 2024**

- Developed **Spring Boot API** to generate custom configs, resulting in dynamic web pages.
- Enhanced **back-end security** by implementing input validation layer.
- Implemented and deployed backend request logging, reducing resolution time by 30%.

AI Tutor Software Developer — **MIT Media Lab** — **Spring 2024**

- Boosted data storage efficiency by 30% using **SQL**.
- Designed API architecture, streamlining feature implementation through a clear roadmap.
- Automated context retrieval for the model by implementing a RAG parser.

Software Developer — **MIT CS Dept.** — **Summer 2022**

- Engineered filtering algorithms for identifying specific compounds, increasing performance by 25%.
- Implemented **caching**, boosting data retrieval speed by 50%.
- Optimized file storage by 75%, resulting in 20% faster loading.

Projects

Raft by Ongaro and Ousterhout — **Go / RPC / Parallelization**

- Implemented **fault tolerant Raft Distributed System**, using **RPC** and **Go**.
- Optimized worker and leader algorithms through parallelized RPC calls.
- Thoroughly tested the implementation using Go's robust testing framework.

Ray tracer Multi-Body Simulator — **C / C++ / AWS / Multi-Core**

- Utilized **OpenCilk** for **multi-core processing**.
- Optimized algorithms based on **Span** and **Work** of **paralleled code**.
- Utilized **AWS** for better performance testing.

StarBattle Video Game — **Typescript / Full-Stack**

- Implemented feature where the application handles **concurrent** inputs from users.
- Created **Python back end server** to retrieve and store games.

Tools

- **Languages**: Python (Strong); Java (Strong); Typescript; Go; C;
- **Libraries**: OpenCilk; Springboot; Fast API; Maven; Gradle
- **Stacks**: Git; S3; Lambda; PostgresQL; RDS