

Haozhe Li

📞 +1 (437) 661-7680 | 📩 lihaozhe013@gmail.com | 🗂 Portfolio | 💬 lihaozhe013 | 🌐 lihaozhe013

EDUCATION

University of Toronto

B.A.Sc in Computer Engineering + PEY Co-op

Toronto, ON, Canada

Sep. 2023 – Apr. 2028 (Expected)

Relevant Coursework: Algorithms & Data Structures, Computer Hardware, Operating Systems, Introduction to Databases, Computer Organization, Software Design & Communication, Communication Systems, Digital Systems

TECHNICAL SKILLS

Programming Languages: C, C++, TypeScript, JavaScript, Java, Go, Python, RISC-V Assembly, MATLAB

Developer Tools: Git, Bash, VS Code, Cursor, Vim, Chrome DevTools, Makefile

Frameworks: Node.js, React.js, Spring Boot, Vite, Tailwind CSS, GTK, Electron Framework

DevOps & Cloud: Nginx, GitHub Actions, Docker, AWS

Other Tools: Figma, Microsoft Office, L^AT_EX, Google Workspace

Hardware / Digital Design: Simulink, FPGA board, Verilog, LTSpice, Quartus Prime, ModelSim, DESim

EXPERIENCE

Full Stack Engineer (Contract)

Sep. 2025 – Present

Hangzhou MYF Technology Co., Ltd.

Remote

- Developed and maintained proprietary enterprise modules (private repositories) extending the open-source core, while implementing server-side logging and alerting mechanisms to monitor system health and minimize downtime.
- Orchestrated hybrid cloud infrastructure across AWS (Pre-prod) and Alibaba Cloud ECS (Production), utilizing Docker and PM2 to ensure high availability and resource efficiency for the company's core trading platform.
- Designed and implemented a CI/CD pipeline via GitHub Actions, automating testing, build processes, and Docker image publishing, which reduced deployment turnaround time and streamlined the release cycle.
- Hardened system security by configuring Nginx as a reverse proxy with automated SSL renewal (Certbot), ensuring compliant HTTPS data transmission for enterprise users.

Frontend Developer (Intern)

May. 2025 – Aug. 2025

Hangzhou EagleCloud Security Technology Inc.

Hangzhou, Zhejiang, China

- Developed new front-end features for an Electron-based enterprise cybersecurity desktop application and implemented UI functions for an admin web console. Utilized TypeScript, React.js, Ant Design, within a 7-person front-end team.
- Utilized Cursor (AI IDE), optimized coding workflows and enhanced software quality by applying advanced prompt engineering methodologies.
- Leveraged advanced Git workflows and GitHub PRs to manage code integration into the CI pipeline, successfully resolved merge conflicts in both QA and pre-production environments.
- Utilized DevOps and CI/CD pipelines for testing, self-tested code in the pre-production environment to ensure the feasibility and robustness, successfully contributed features to a SaaS release.

AI Lab Research Assistant (Intern)

Jun. 2024 – Aug. 2024

Shenzhen Research Institute of Big Data

Shenzhen, Guangdong, China

- Automated research environment by scripting the one-time cleanup and reinstallation of Conda environments and key packages (PyTorch/TensorFlow), reducing setup time and enabling teams to immediately run new models on idle computing capacity.
- Reinstalled Ubuntu and Debian systems on lab computers to fix compromised software environments, and configured a seamless model deployment workflow by integrating SSH with the research team's web console, repairing numerous computers that the research team couldn't use for experiments.
- Developed a comprehensive guide and configured runtime environments for the research team to run open-source models for reference, reducing the time research teams spend on configuration.

PROJECTS

TradeFlow System (Open Source) <i>TypeScript, Node.js, SQLite, React.js</i>	Jul. 2025 – Present
<ul style="list-style-type: none">Architected a full-stack trade management solution with a modular Monorepo structure, featuring JWT-based RBAC authentication and i18n support to facilitate order and financial management.Engineered a high-performance backend using Node.js and Express, utilizing Decimal.js for precise financial calculations and ESBuild for optimized build artifacts.Developed a responsive frontend with React and Ant Design, adopting a component-driven architecture to accelerate feature delivery and maintain code reusability.	
GIS Route Optimization Application – Course Project <i>C++, GTK, Git, A*, Dijkstra</i>	Jan. 2025 – Apr. 2025
<ul style="list-style-type: none">Developed a Geographic Information System (GIS) desktop application in C++ with GTK on Mate Desktop as a course project in a 3-person team, implemented map rendering, geographical name search, shortest path and multi-stop path finding features.Leveraged multithreading to preprocess 4GB of raw coordinate data, converting it into structured point, line, and polygon formats for canvas rendering in under 50 seconds on a 16-core test machine.Implemented pathfinding algorithms including A* for shortest path computation, and Dijkstra, multi-start greedy, and simulated annealing for multi-stop route optimization, achieving 90% of the technical evaluation score.Integrated real-time traffic visualization by consuming TomTom API data and incorporating the libcurl HTTP module, enabling dynamic display of traffic congestion during navigation for enhanced route planning.Maintained a clean Git workflow with feature branches and used Makefile-based build scripts to streamline collaboration and ensure efficient development across the team.	
StreamFile Server <i>Go, Gin, Node.js, TypeScript, Express, Multer, Video.js</i>	Jan. 2025 – Present
<ul style="list-style-type: none">Developed a Go-based, database free server for static resource hosting, providing private link generation, file upload, HTTP Range support, and file search features.Implemented frontend features such as Markdown rendering, video/audio playback, static webpage hosting.Optimized most frontend components using only Tailwind CSS and native HTML DOM, using ESBuild for JavaScript minification, creating an ultra-lightweight frontend, significantly reducing page load time on low specification devices.	
Runner Game (FPGA Board Game) – Course Project <i>C, RISC-V Assembly, FPGA Board, CPUlator</i>	Mar. 2025
<ul style="list-style-type: none">Developed a 2D runner game in C on a DE1-SoC FPGA board, implementing core game logic, PS/2 keyboard, audio components, and video components, and delivered a playable game in 3 weeks.Integrated a PS/2 keyboard for real-time user control, utilizing CPU interrupts to achieve low-latency input handling.Built a lightweight, OS-independent 2D graphics engine capable of rendering color images, animations, and text at up to 60 FPS.Developed a basic square wave synthesizer to enable dynamic sound effects during gameplay events such as scoring and game completion.	
Greedy Mouse Game – Course Project <i>Verilog, FPGA Board, ModelSim, Quartus Prime</i>	Nov. 2024
<ul style="list-style-type: none">Designed and implemented a 2D Greedy Mouse game in Verilog for the DE1-SoC FPGA board, featuring interactive keyboard input and VGA graphics output.Developed a PS/2 keyboard controller for real-time user input, and integrated it with a finite state machine (FSM) and on-chip memory modules to drive VGA output, enabling smooth display of bitmap animations and enhancing gameplay visuals.Built a lightweight, OS-independent 2D graphics engine capable of rendering color images, animations, and text at up to 60 FPS.Used ModelSim and DESim for early-stage simulation and debugging; finalized the design using Quartus Prime for synthesis and deployment to the FPGA board, followed by iterative optimization.	