Haozhe Li

+1 (437) 661-7680 | lihaozhe013@gmail.com | Portfolio | lihaozhe013 | lihaozhe013

Education

University of Toronto Toronto, ON, Canada
B.A.Sc in Computer Engineering + PEY Co-op *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, Spring Boot, Vite, Tailwind CSS, GTK, Electron Framework
DevOps & Cloud: Nginx, GitHub Actions, Docker, AWS
Other Tools: Figma, Microsoft Office, LaTeX, 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*

- Led requirements discovery and authored a comprehensive PRD; designed **Figma** wireframes and delivered a Node.js + JavaScript MVP upon client sign-off.
- Built and maintained full-stack tradeflow enterprise modules on top of a **Node.js** open-source core using **TypeScript**; implemented server-side observability (structured logging, metrics, alerting) to monitor health and reduce downtime.
- Orchestrated cloud infrastructure across **AWS** (Pre-prod) and **Alibaba Cloud ECS** (Production), utilizing **Docker** to ensure high availability and resource efficiency for the company's core trading platform.
- Implemented a **CI/CD** pipeline with **GitHub Actions** to automate testing, builds, and Docker image publishing, accelerating deployments and streamlining release cadence.
- Strengthened security by fronting services with **Nginx** reverse proxy and automated SSL renewal (Certbot), enforcing compliant HTTPS 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**, **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.
- Worked within standardized development workflows, including **Gitflow branching** (resolving merges and conflicts across QA and pre-production), **GitHub Pull Requests** with assigned reviewers and feedback resolution, and **CI/CD integration** testing on Alibaba Cloud (Yunxiao) for build consistency.

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) | *TypeScript, Node.js, SQLite, React* Jul. 2025 – Present

- 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 | *C++, GTK, Git, A*, Dijkstra* Jan. 2025 – Apr. 2025

- 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 | *Go, Gin, TypeScript, Express, Multer, Video.js* Jan. 2025 – Present

- 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 | *C, RISC-V Assembly, FPGA Board, CPUlator* Mar. 2025

- 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 | *Verilog, FPGA Board, ModelSim, Quartus Prime* Nov. 2024

- 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.