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, C#, Go, Python, RISC-V Assembly
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, LaTeX, Google Workspace
Hardware / Digital Design: FPGA board, Verilog, LTSpice, Quartus Prime, ModelSim, DESim

Experience

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 from GitHub, reducing the time research teams spend on configuration.

Projects

TradeFlow System | *TypeScript, Node.js, SQLite, React.js, Vite* Jul. 2025 – Present

- Developed a full-stack trade management system using Node.js for an integrated circuit sales company, featuring JWT-based authentication, RBAC, internationalization (i18n), and Excel export capabilities.
- Built the backend with TypeScript, Express.js, SQLite, Decimal.js, and JWT libraries, establishing a modular project architecture, clean Git commit history, and automated build and minification build scripts powered by ESBuild.
- Engineered the frontend with TypeScript, React, Vite, React Router, and Ant Design, adopting a component-driven "Vibe Coding" approach to accelerate MVP delivery in early product stages.

TradeFlow System Infrastructure | *CI/CD, GitHub Actions, AWS, Nginx, Docker* Aug. 2025 – Present

- Managed and deployed cloud infrastructure across AWS EC2 (t2.micro) for pre-production and Alibaba Cloud ECS for production, both containerized with Docker and managed via PM2 for process reliability and concurrency control.
- Configured Nginx reverse proxy, automated environment setup scripts, and managed SSL certificates.
- Implemented CI/CD pipeline with GitHub Actions, automating build and test processes and streamlining semi-automated Docker-based deployments to cloud environments.
- Developed lightweight log management and alerting mechanisms within the backend to monitor system stability.

StreamFile Server | *Go, Gin, Node.js, 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.

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.
- Utilized A* algorithm for shortest path finding, Dijkstra, multi-start greedy method and simulated annealing for multi-stop path finding, ultimately achieved 90% of the technical grade.
- Maintained a clear Git branching strategy utilizing feature branches, which resulted in timely mergers and effective conflict resolution, leading to each milestone being completed ahead of schedule.

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 as part of a 2-person team, implementing core game logic, VGA display, and audio components, and delivered a fully functional game as the course project.
- Utilized Git for version control and CPUlator for simulation and debugging, optimized online collaboration, and reduced integration issues, which enabled the team to complete all development work in just 2 weeks.
- Compiled, deployed, and optimized the game on the FPGA board, ensuring stable performance during a 5-hour continuous run, demonstrating system reliability and robustness.

Greedy Mouse Game (FPGA Board Game) – Course Project | *Verilog, FPGA Board, ModelSim* Nov. 2024

- Developed a 2D Greedy Mouse game in Verilog 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.
- Proactively managed partner's extended absence and non-cooperation by escalating the situation to the TA and swiftly adjusting the project plan. Accelerated development to independently complete the project within the deadline, achieving a grade of 80% and demonstrating strong adaptability and problem-solving skills.
- Utilized ModelSim and DESim for game development and testing, compiled and deployed the game on the board using Quartus Prime, and presented a playable demo in the final presentation.