Zhenfeng Liao

zfliao.offers@gmail.com | 090-2212-7961 | linkedin.com/in/zhenfeng-liao | github.com/Zhenfeng7

Education

University of Toronto, BS in Computer Science — Sept 2020 – May 2025

- Recipient of **Dean's List** Academic Achievement Award

Waseda University, MS in Information and Data Engineering — Oct 2025 – June 2027

Skills

- **Language**: **Japanese** Intermediate Working Proficiency, **English** Full Professional Proficiency
- **Programming Languages**: Python, HTML, CSS, JavaScript, TypeScript, Java, C++, C, Haskell, Dart, SQL
- **Frameworks**: Django, React, Node.js, Bootstrap, Express, Flutter, Vite
- **Other Tools**: SQL, NoSQL, AWS: EC2, SES, Bedrock, Landing Zone, CloudTrail, RDS, S3, etc.

Experience

Cloud Engineer Intern, Softchoice – Toronto, ON, Canada — May 2023 – Aug 2023

- Cooperated with a group of 3 team to develop an AI-powered resume analysis application using **AWS Bedrock (Llama 2 Model), AWS Lambda,** and **AWS Textract** to efficiently extract and filter resumes based on key skills and experience.
- Optimized the application by restructuring **S3 storage** for better data organization, resolving content-mixing challenges, and adopting cost-effective solutions, saving recruiters over **70%** of their review time.
- Designed comprehensive unit tests to ensure the robustness and reliability of the **AWS Landing Zone** automation codebase, and successfully integrated unit tests with **AWS CloudTrail**, leveraging real-world scenarios to identify and resolve code bugs.
- Deployed and hosted websites on **AWS** infrastructure, leveraging services including **EC2, RDS, Route53, and S3**, ensuring **thousands** of daily website visits with high availability and reliability.

Quality Assurance Intern, Sichuan Digital Entertainment Company – China — May 2019 – Aug 2019

- Tested games on multiple OS platforms, including smart TV OS, Android, Windows, and IOS.
- Tracked the bugs on **Jira** to report to developers to prevent graphic defects before releasing.
- Shadowed the daily work of developers to understand **Agile methodology** of software development
- Contributed to black box testing of pre-released games to find bugs.

Projects

YouTube Looper *View in Chrome Store or Github*

- Built a Chrome extension that lets users set start/end times and loop any segment of a YouTube video inline in the player—no popups, no speed changes.
- Engineered ad-safe behavior (mutation observers suspend looping during ads and resume afterward) and SPA resilience (detects YouTube's single-page navigation, rebinds to the new video, resets loop state).
- Delivered a typed UI/loop stack (hh:mm:ss inputs, snap-to-start, deactivation on out-of-range seeks) with Vite multi-entry builds and Vitest coverage for loop logic and UI integration; packaged for the Chrome Web Store and published to GitHub.

Academix *Flutter, Dart, Firebase*

- Cooperated with a group of 5 and built a full-stack chatting, social and academic course planner **mobile** application for UofT students by **Flutter/Dart and Firebase.**
- Designed and refined a **Figma** app prototype by conducting a survey of **200** students to gather user needs and feedback, performing **usability testing** with **15 target users** to identify pain points. Ran **A/B tests** to optimize key features and layouts, resulting in a user-centric and data-driven app design.
- Designed and developed the UI of the chat page and implemented the chat feature for **real-time** notifications and **live chat** with texts, images, and files.

Currency Rate Alert App *Node.js, RESTful API*

- Built a product that lets users track currency pairs, set target exchange rates, and receive email alerts when their targets are hit.
- Designed the system around a scheduled rate fetcher and caching layer so users always see the latest available rates with clear timestamps instead of raw "real-time" noise.
- Implemented a web dashboard where users can create, edit, and pause alerts, and see which ones have recently triggered.
- Focused on reliability and cost control by limiting fetch windows and batching provider calls, keeping the app responsive while staying within free-tier limits.