

Ruiquan (Richard) Su

506-886-8834 | r38su@uwaterloo.ca | [LinkedIn](#) | [Website & Portfolio](#)

EDUCATION

University of Waterloo

Computer Engineering, Hon. Coop, President's Scholarship with Distinction, GPA: 4.0/4.0

Waterloo, ON

Sept. 2024 - Present

EXPERIENCE

Software Developer

September 2025 - December 2025

Lynkr Inc.

Toronto, ON

- Developed, along with **only one other developer**, the **core company product**, Lynkr Workbench, from **scratch** to **beta**, fetching over **100,000 CAD** in revenue and over **2800 clients** for the startup.
- Designed the **architecture** for Workbench, then used **FastAPI** and **Next.js** to implement the app, with secure routing and **two AI agent streams** that handle the creation & execution of **AI agents** that calls & executes services autonomously, storing user data with **PostgreSQL** and manages secrets using **Google Cloud**.
- Implemented and seeded **vector embedding** for the available service endpoints, and a **cosine similarity search** along with **LLM routing** to quickly fetch services & endpoints based on a natural language prompt.
- Used **MCP servers** to integrate over **100** services and over **3000** endpoints, reflecting a **1600%** increase.

Control Software Developer

January 2025 - April 2025

SIOUX Technologies (Asia), Jiangsu Industrial Technology Research Institute (JITRI)

Suzhou, China

- Cut on-site deployment time **from 98 hours to 20 hours** by designing and developing concurrent **streamlined installation and upgrade processes**, reducing manual overhead via GitLab **CICD pipelines** and **scripting**.
- Refactored existing services into **organized packages** and developed **Blazor-based API controllers** to automate installations, enabling **direct customer access**.
- Organized and led **company-wide Git training sessions** to enhance version control proficiency across teams.

PROJECT SELECTION

SnapLearn | *Flask, PyMuPDF, SciKit-learn, Next.js, Auth0, spaCy*

January 2025 - May 2025

- Built an app with **Next.js** and **Flask** to take in a textbook uploaded by the user and the time they plan to study every day, and regenerate the topics the textbook covered into a curriculum that fits the user's schedule.
- Designed a Python **algorithm** that takes in any variety of topics, and uses **NLP techniques**, as well as **cosine similarity search**, to determine their dependency relationship in learning, and creates a directed acyclic graph.
- Trained a **small-sized language model** to estimate the time needed for an average person to study a certain topic with all pre-requisite knowledge acquired.
- Developed a **greedy algorithm** to fit the course content into the user's schedule, while also balancing it with practice sessions. Then used an **LLM** to generate the course content with the specified length.

QuickSilver | *Python, Socket, IPAddress, re, pybluez, Tkinter, PyInstaller*

April 2025 - May 2025

- Built a **broadcaster-receiver pair** that broadcasts information to **every receiving machine in the LAN**
- Developed a file **sending and receiving system** via a **TCP or RFCOMM connection** that can validate and send files to **any computer** on the LAN or with Bluetooth, and a validation mechanism when receiving files.
- Maintained a **CLI implementation** of the application, while also designing a **user-friendly UI** using **Tkinter**, integrating **all CLI functions** into it, and packaged the program into a **standalone EXE**.
- **Automated** the build process on release using **CICD pipelines** via **GitHub Actions**.

WatIsLife | *Python, JavaScript, React, Flask, GSM*

December 2024 - January 2025

- Developed a **web API** that **records** the user's speech and uses **Google Speech Module** to count how many times a certain phrase was said and **stores** the count in a **database**.
- Fine-tuned a **Llama LLM** model to translate the slang in the user's speech into common English. Data sourced from scraping several teenager subreddits.
- Built a simple **React front-end** that displays a **leaderboard** for all users.

TECHNICAL SKILLS

Languages: Python, C/C++, C#, SQL (Postgres, MySQL), JavaScript/Typescript (React, Next), Powershell, YAML

Developer Tools: Git, Node, Yarn, Docker, VS Code, Jupyter, Anaconda, NSIS, NuGet, Vim, Jira, IntelliJ

Libraries: pandas, NumPy, Matplotlib, Tkinter, PyKrig, Flask, FastAPI, MCP, Pydantic, Tensorflow, sklearn, spaCy