

# Ken Jiang

+1 647 936 9283 | [ken.jiang@uwaterloo.ca](mailto:ken.jiang@uwaterloo.ca) | [linkedin.com/in/ken-jiang](https://www.linkedin.com/in/ken-jiang) | [github.com/kenzoengineer](https://github.com/kenzoengineer) | [kenjiang.ca](https://kenjiang.ca)

## EDUCATION

### University of Waterloo

Honours Computer Engineering - B.A.Sc Candidate

Academic Exchange: Hong Kong Polytechnic University - Electronic and Information Engineering

Waterloo, Ontario

Expected April 2025

## SKILLS

**Languages:** TypeScript, JavaScript, Python, C#, C, C++, Java, SQL, HTML, CSS

**Frameworks/Libraries:** React, Vue, Next, Jest, Node, Tailwind, TypeORM, .NET

**Tools:** Git, SQL Server, PostgreSQL, MongoDB, Docker, Enzyme, AWS (S3, EC2), Unity

## EXPERIENCE

### Software Engineer

Vontive

January 2024 – April 2024

San Francisco, California

- Introduced aggregated real estate and market pre-qualification data via a **Vue** and **TypeScript** dashboard preventing resource wastage on unserviceable loans
- Owned full design of a third-party electronic document signing integration, lowering time to dataset completion for **6000+** annual documents by **40** times
- Implemented a **webhook** handler which automatically extracted data from signed documents, and uploaded them to **AWS S3** and **PostgreSQL**, eliminating manual user intervention
- Proxied sensitive data through **Very Good Security**, offloading encryption of Social Security Numbers, Account Numbers and other PII

### Software Developer

Senstar

January 2023 – April 2023

Waterloo, Ontario

- Owned design and development of all hardware device integrations into Senstar's client allowing for seamless addition of new perimeter security hardware through a **.NET C#** and **C++** back-end
- Built a reusable upgrade pipeline automatically converting legacy hardware implementations to newer integrations stored in **PostgreSQL** and **SQL Server**, increasing client security, feature adoption
- Enhanced a **TCP-based** access control security system event handler to support 100+ concurrent connections using **multi threaded sockets**, allowing many clients to send security commands simultaneously
- Revamped a **TypeScript** canvas control to allow for precise pixel-level masking of video feeds, allowing a user to restrict the processing region of face recognition and license plate detection increasing speed and accuracy

### Software Developer

Shoplogix

May 2022 – August 2022

Oakville, Ontario

- Developed an extensible settings management system in **C#** and **TypeScript**, allowing plant, area, and machine level configuration with support for overriding and inheritance massively simplifying company-wide customization
- Built 5+ API **endpoints** in **C#** with **Nancy** which fetch data from **PostgreSQL**, exposing per user configuration options and simplifying machine **OPC** data collection
- Synthesized a **React** component to handle 100+ routing combinations simplifying future development
- Created and validated unit tests in **Jest** and **Enzyme**, achieving 96% code coverage over 5+ features

### Application Developer

QBuild

September 2021 – December 2021

Markham, Ontario

- Improved an Engineering Change Notice management web app to simplify design iteration and logistics
- Parallelized initial data fetching with task-based logic to speed up latency by up to 15 times
- Created a responsive image upload feature which processed files into **SQL Server** in under 100ms for a seamless user experience
- Built an extensible comments section from scratch using **Telerik UI**, complete with email notifications and employee tagging

## OTHER EXPERIENCES

### WATonomous - Lead Full Stack Developer | Next.js, Tailwind, Vercel

January 2023 - Present

- Founding** lead developer of WATonomous' new website initiative, creating an appealing brand and source of information for prospective members and sponsors
- Managed a team of 5 developers, maintaining a high standard of code quality and prioritizing effective feedback
- Built 30+ **Tailwind** styled components from scratch in **Next.js** and managed deployment with **Vercel**