

# William Qin

Software Engineering - University of Waterloo

2019 – 2024 | 96.6% GPA, 1<sup>st</sup> in class Fall 2020

 [hello@williamqin.com](mailto:hello@williamqin.com)

 [williamqin.com](http://williamqin.com)

 [github.com/WilliamLQin](https://github.com/WilliamLQin)

## Skills

---

**Languages** TypeScript, JavaScript, Python, Java, SQL, C/C++, C#, Go, Rust, Scala, HTML, CSS/SCSS

**Web** Node, React/React Native, Vue, AWS, PostgreSQL, MongoDB, Kubernetes, Terraform, Firebase

**Tools** Snowflake, Jest, Mocha, Cypress, Docker, Unity, Git, Bash, Unix, Jenkins

**Tech** Arduino, Raspberry Pi, OpenCV, VR, ARCore/ARKit, Mixed Reality Toolkit, Blockchain

## Experience

---

**Full Stack Developer** | [Aven Financial](#) | Burlingame, CA Jan 2023 – Present

- Launched cardholder promotions using **TypeScript Node** backend, yielding a **2.5x ROI** and engaging **53% redeemers**
- Created initiatives in **TypeScript Vue** web app to move **20.5% of users** to enable Auto Pay, reducing delinquency rates
- Integrated **MethodFi** debt API to add new debt sources to balance transfer flow and pull data into **PostgreSQL** DB
- Validated new features with **Jest** unit tests, **Cypress** frontend tests, and endpoint integration and data quality tests

**Blockchain Integration Developer** | [BitGo](#) | Palo Alto, CA (Remote) May 2022 – Aug 2022

- Integrated a **top 25** cryptocurrency to the core platform with **Node** and **Typescript** to boost revenue by over **\$300k**
- Packaged a blockchain full node using **Docker** to integrate custom bug fixes and generate test cases
- Wrote extensive tests in Typescript **Mocha** to ensure the safety of the code refactors needed for the new cryptocurrency

**Chassis Controls Firmware Developer** | [Tesla](#) | Palo Alto, CA (Remote) Sep 2021 – Dec 2021

- Built a **Python** test framework to run custom checks on **7500+** vehicle simulations, improving validation confidence
- Integrated test framework with **Jenkins** and **SCons** to auto-run tests and ensure their continued use and maintenance
- Enhanced simulation with a modified interpolation search in **Rust** to implement new open loop playback functions

**Full Stack Developer** | [Spatial](#) | San Francisco, CA (Remote) Jan 2021 – Apr 2021

- Developed backend endpoints in **Go** with **MongoDB** to support new pro features, driving growth of paying users by **10x**
- Added no auth flow to **TypeScript React** webapp for joining rooms by link, **converting 27%** of users to create an account
- Migrated web services to **Kubernetes** using **Terraform** and **Docker** for a declarative and more reliable infrastructure

**Mobile Augmented Reality Developer** | [Spatial](#) | San Francisco, CA (Remote) Apr 2020 – Aug 2020

- Ported the entire Spatial holographic meetings app to a **mobile iOS** and **Android AR app** to significantly expand meeting accessibility to users without AR/VR headsets, now making up **20% of all active users**
- Built a **TypeScript React Native** project on top of a **Redux**-style state system in **Unity C#** for a modular and native UI layer

**Software Lead** | [FIRST Robotics Team 4308](#) | Mississauga, ON Sep 2017 – Jun 2019

- Developed an **autonomous** robot driving mode with **Java** that led the team to become **world semi-finalists** in 2018
- Created a **Firebase web app** to collect, display, and analyze team stats, informing match strategy and alliance selection

## Projects

---

**Vision Motion** | Android Application Oct. 2017 – Jan. 2020

- Used real-time **OpenCV** computer vision to launch motion graphing app with **2270+** installs & **400+** peak active users
- Integrated **Firebase** database and authentication to allow users to save and upload their data for use across devices

**Project BRETT** | Drone Application | First Year Design Project Nov. 2019

- Built an **NXP microcontroller** drone with an infrared sensor and camera to autonomously map heat sources in an area
- Developed a drone program in **C/C++** to collect temperature data in real-time and create a file for offline processing