

SHIYAO WANG

swangwang2000@gmail.com | (+64)0212305686 | [GitHub](#) | [LinkedIn](#)

TECHNICAL

Software	GIT, Docker, MongoDB, MS Office Suite
Software Languages	Javascript/Typescript, Python, Golang, SQL, NoSQL
Spoken Languages	English, Mandarin

Experienced in using Windows and Linux operating systems

PERSONAL STATEMENT

Full-Stack Engineer experienced in developing SaaS products following CI/CD practices. Have worked extensively with Dev-Ops tools such as GitHub, Jira, Figma, and Confluence when collaborating with the engineering and design teams.

WORK EXPERIENCE

AsBuilt <i>Junior Software Engineer</i>	<i>November 2020 - January 2025</i>
---	-------------------------------------

- Create/fix/maintain features and packages across frontend(React + js/ts) and backend (echo + Golang)
- Taken the lead on some larger features and run stand-ups/retros
- Work closely with the design team to resolve technical issues and come up with alternative solutions
- Experience in using MS Azure services (blob storage, dockerized image management, PIM)

AsBuilt <i>Software Engineer Intern</i>	<i>December 2019 - February 2020</i>
---	--------------------------------------

Research and development of some POC applications based around Autodesk's Forge-Viewer API

- Use the Forge-Viewer API to manipulate and interact with 3D elements on a web application
- API data ingestion of IOT devices and display them as meaningful graphs
- Meta data extraction of uploaded JPEG images and plot them geographically on a web application

EDUCATION

University of Auckland, Auckland Bachelor of Engineering (Hons) in Computer Systems Engineering	<i>March 2018 - November 2021</i>
---	-----------------------------------

PROJECTS

Blog Site: ribbit.moe	<i>2024</i>
--	-------------

Host a blog site where users can make an account and share posts publicly or privately

- React framework for front-end written in TypeScript
- Echo framework for back-end written in Golang
- NoSQL Database hosted with MongoDB
- Webapp hosted on Google Cloud using Cloud-run running dockerized images

(References available on request)