Tom Wang

647-852-7705 | zxTomw.com | tom.wang3@uwaterloo.ca | linkedin.com/in/zxtomw | github.com/zxTomw

Education

University of Waterloo — Waterloo, ON

3rd Year Bachelor of Computer Science, Honours — *Sept. 2022 – May 2027 (Anticipated)*

- GPA: 3.72, Excellent Standing

Technical Skills

Languages: TypeScript/JavaScript, C/C++, Python, Java, SQL, Swift, HTML, CSS
Frameworks: React, Angular, NextJS, Remix Tailwind CSS, Remix, Qt, Node.js, Express, Flask, Django, SwiftUI
Developer Tools: Git, Docker, Vercel, Tailwind CSS, Make, Google Cloud, AWS, MS Azure, CI/CD Pipelines
Technologies: TanStack Query, Nginx, PostgreSQL, MongoDB, OpenAI SDK, WebSocket, Pandas, TensorFlow

Experience

Autonomous Robot Software Developer — Jan 2025 – Present

WATonomous — *Waterloo, ON*

- Implemented a robot pathfinding algorithm that accounts for detected obstacles using **ROS** and **C++**.
- Integrated a monocular depth estimation algorithm to the perception node of the autonomous vehicle.
- Researched machine learning algorithms for camera/sensor anomaly detection.

Remote Healthcare Software Engineer — Jan 2024 – Dec 2024

Mespere Lifesciences & Faculty of Science, University of Waterloo — *Waterloo, ON*

- Developed a full-stack web app for a remote healthcare system for patient data monitoring and remote diagnosing.
- Reduced streaming delay of real-time data by 15% by avoiding copying of dataset on the client side.
- Optimized rendering performance using **React 19** Compiler, resulting in significantly reduced rerendering counts.
- Utilized **NextJS 15** to optimize performance and implemented 20+ server components.
- Engineered a highly scalable **serverless** backend using **TypeScript**, **Azure Functions**, and **CosmosDB**.
- **Visualized** real-time sensor data into interactive graph components using **WebGL**, **React** and **Tailwind CSS**.
- Built CI/CD pipelines with **GitHub Actions** and **TurboRepo** for streamlined deployment and testing.

Projects

PersonalNotes | *TypeScript, NextJS, OpenAI, Flask, Google Cloud, PostgreSQL* — Hack the North 2024

- Developed an **AI agent** to aggregate and summarize documents and notes with **NextJS** and **Tailwind CSS**.
- Configured CI/CD pipelines for automated deployment to **Vercel** and **Google Cloud**.
- Leveraged OpenAI **vector embeddings** to construct knowledge bases from uploaded materials.
- Formulated effective end-to-end strategies surrounding storage organization with **Flask** and **PostgreSQL** to handle thousands of CRUD requests every month.

NaviGoose | *Python, OpenCV, YOLO, PyTorch, Django, Websocket, OpenAI* — GeeseHacks 2025

- Built an IoT AI hardware with **Raspberry Pi** and **Django** to help visual-impaired individuals avoid collisions.
- Trained an object detection network using **YOLO** models and **PyTorch** to identify 100+ types of hazards.
- Tailored personalized vocal warnings via **OpenAI** and **LangChain** SDK; achieved an average accuracy rate of 95% in identifying hazardous conditions while assisting over 50 test participants during field trials.

Lost and Found App | *JavaScript, ExpressJS, MongoDB, NodeJS, AWS*

- Developed a self-hosted application using **React** and **ExpressJS** to help people retrieve their lost items.
- Utilized **MongoDB** and **ExpressJS** to manage user and item data, and defined their shapes using schema.
- Designed referencing schema between the user and item clusters in the database to allow fast retrieval of data.

LiDAR Pathfinding Robot | *C++, CMake, Docker, ROS, Foxglove*

- Developed software for a differential wheeled robot that navigates independently on LiDAR data using **ROS**.
- Implemented the cost map module to analyze raw LiDAR inputs and mark-up dangerous areas on local maps.
- Utilized the **A*** algorithm to generate the shortest paths and avoid collisions.

Chess Game | *C++, GNU, CMake*

- Utilized the **RAII** paradigm, **OOP** principles (inheritances, polymorphism), and **MVC** design pattern.
- Innovated AI-player strategies employing alpha-beta pruning which resulted in 60% faster response times.
- **Optimized** the AI players and reduced processing time by 45% by refactoring the move ranking algorithms.