Tom Wang

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

Education

University of Waterloo Waterloo, ON
3A Bachelor of Computer Science *Sept. 2022 – May 2027 (Anticipated)*

- Faculty average: 85%; Excellent Standing.

Experience

Attendee Sept. 2024
Hack the North 2024 *Waterloo, ON*

- Designed an AI learning system to aggregate and summarize documents and notes
- Implemented a full-stack web app with **NextJS** and **Flask** to handle file uploads and streaming **LLM** responses
- Employed **ShadCN** and NextJS app router to build and manage the login pages, file uploads, and dashboards
- Configured development environments and CI/CD tools for automated deployment of web app to **Vercel** and backend service to **Google Cloud**
- Leveraged OpenAI **vector embeddings** to construct knowledge bases from uploaded materials
- Built and managed a database using **PostgreSQL** and Flask for user authentication and file indexing

Software Engineering Research Assistant Jan 2024 – Dec 2024
Tang Nanotechnology Lab, Faculty of Science, University of Waterloo *Waterloo, ON*

- Developed a full-stack web app for a remote healthcare system for sensor data streaming and video conferencing
- Implemented real-time sensor data streaming to the web interface using **Websocket** and Azure IoT Hub
- Optimized the streaming of real-time data, reducing delay by 15% by avoiding copying of dataset on the client side
- Utilized **NextJS** and **TypeScript** to build a backend-for-frontend structure with server-side components to optimize performance and user experience
- Leveraged Azure Functions and CosmosDB to build and maintain a scalable **serverless** backend
- **Visualized** real-time sensor data into interactive graph components using **ChartJS** and **React**
- Rapidly iterated with **GitHub Actions** for streamlined deployment and testing

Projects

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

- streamlined message-based communications between node objects using the **ROS** Library and observer pattern
- Developed the cost map module to convert raw lidar inputs into coordinate-based values to be stored in local maps
- Leveraged **tf2** for coordinate transformations to align and aggregate local maps with the global map
- Utilized the **A*** algorithm to generate the shortest paths and avoid collisions
- Reduced the computation for updating the global map by 60% by optimizing the algorithm for aggregating maps

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

- Implemented a command line application following the **model-view-controller** pattern
- Constructed chess classes by adopting the **RAII** paradigm, **OOP** principles (inheritances, polymorphism), and various design patterns
- Designed the algorithms of the computer players of different levels for move prediction with **alpha-beta pruning**
- **Optimized** computer players (AI) and reduced their processing time by 45% by refactoring the prediction and ranking algorithms

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

- Developed **RESTful APIs** for the backend server using the Express framework, supporting database operations
- 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 to allow fast retrieval of independent data
- Leveraged **JWT** verification to implement user authentication. Implemented API-key-based authentication with **AWS** API Gateway

Technical Skills

Languages: C/C++, TypeScript/JavaScript, Java, SQL, Swift
Frameworks: NextJS, React, Node.js, Express, Flask, Django
Developer Tools: Git, Docker, CMake, Google Cloud Platform, AWS, MS Azure, Foxglove, Turborepo, GitHub Actions
Technologies: PostgreSQL, MongoDB, ROS, WebSocket, Supabase, Pandas, NumPy, TensorFlow