

Richard Bai

437-226-7128 | r25bai@uwaterloo.ca | [LinkedIn](#) | [Github](#) | [Personal Website](#)

EDUCATION

University of Waterloo
Bachelor of Computer Science

Waterloo, ON
Sep 2023 – Present

TECHNICAL SKILLS

Languages: JavaScript, TypeScript, HTML/CSS, Python, C++, C, C#, Java, Bash, Git
Developer Tools/Frameworks: NextJS, ReactJS, Node.js, ExpressJS, Tailwind, Postman, Docker
Libraries: Scikit-learn, Pandas, Matplotlib, PyGame, Selenium

EXPERIENCE

Software Development Intern

May 2022 – January 2023

Freedom Technologies

Beijing, China

- Developed a patented process using **neural networks** to identify and analyze different roof shapes and their respective dimensions and rebuild them in 3D space using **C++**, speeding up roof processing by **86%**
- Designed and implemented an algorithm using **k-means clustering** to get an image of a building's color palette in **Python** using **Scikit-learn** and **NumPy** providing a representative palette of 5 colors with over **90%** accuracy
- Leveraged this algorithm to condense satellite images of buildings to its most dominant colors, speeding up 3D modelling processing speeds by **20%** and reducing project processing loads by an average of over **60%**

Vice President of Technology

January 2021 – December 2023

Bright Youth Journey

Calgary, Alberta

- Organized and hosted over **50** remote weekly lessons to teach around **300 students** the fundamentals of **Java**, **Python** and **C++**, growing the non-profit's student base by over **2.2x** while recruiting **6** new tutors
- Coordinated summer camp initiatives for over **3 years**, leading math and programming summer camps, teaching **83 students** the fundamentals of Python and Java and raising over **\$7,100** for local charities and food banks
- Designed and maintained the organization's website using **HTML**, **CSS** and **JavaScript**, supporting traffic of over **550** users a month from **25** different countries, deploying the website on Netlify to **reduce latency by over 55%**

PROJECTS

Tennis AI | *PyTorch, OpenCV, Ultralytics, Python*

- Engineered a machine learning model using **Ultralytics YOLOv5l6u** to analyze tennis matches using object detection to track player and ball movements to compute and provide key match analytics
- Fine-tuned the model with a **convolutional neural network (CNN)** to extract and draw key-points using **PyTorch** and **OpenCV** to interpolate player and ball positions which are then converted to relative velocities in real-time

Beacons.us | *React Native, Express, MongoDB, Node.js, Git*

- Developed a full-stack social media mobile-application with **React Native** at **Deerhacks 2024**, allowing the user to plan events and implementing features like friend requests and live map modals to streamline UI/UX
- Engineered and documented a back-end API/database with **Express**, **MongoDB**, **Postman** and hosting it on **Render**, allowing the front-end to post, receive and store data efficiently and winning **Best Use of MongoDB**.

Code Wizard | *Vite, Express, MongoDB, Node.js*

- Built a full-stack conversational chat-bot with **React** and **Node.js** that utilizes **OpenAI's chat completions** to provide personalized feedback to coding inquiries, serving around **10** users and over **1,200 inquiries** monthly.
- Architected a cloud-based chat history feature which allows users to access, create and delete conversations using **MongoDB** for scalable data storage and integrating it with **AWS** ensuring **99.99%** up-time and global access.

Pulsar Chat-App | *React, Express, MongoDB, Node.js, Socket.io*

- Implemented a full-stack chat application with **React**, **Express** and **Node.js**, employing **B-crypt's** password hashing for secure user authentication along with providing users with a personalized experience.
- Utilized **Socket.io's** web-sockets to establish connections between users and host **real-time** conversations, integrating a cloud **MongoDB** server to remotely store, delete and access conversations amongst users.

Sorting Visualizer | *NextJS, Tailwind, Node.js, Vercel*

- Designed an interface with **NextJS** and **Tailwind** that allows a user to select different sorting algorithms, view relevant data like time complexity, adjust sorting speeds and play the animation of the selected algorithm.
- Leveraged **React Context API** to share properties across components, abstracting complex state management for global data and improving component re-usability, and deploying the application on **Vercel**.