




# Brian Chang

 [github.com/bchang99](https://github.com/bchang99)  [linkedin.com/in/bchang99](https://linkedin.com/in/bchang99)  [bchang99@gmail.com](mailto:bchang99@gmail.com)  [bchang99.github.io](https://bchang99.github.io)

## Skills

### Frontend:

React   React-Router   React-Redux   Next.js  
TypeScript   MaterialUI   Ant Design   Tailwind

### Languages:

JavaScript   Python  
HTML/CSS   C++

### Backend:

Node.js   Express.js  
MongoDB

### Dev Tools:

Git   Puppeteer   Jest   Postman   NVDA/JAWS

## Education

### University of Waterloo | Mechatronics Honours | Bachelors of Engineering

Cumulative GPA: 85

#### Relevant Courses:

SYDE 522: Foundations of Artificial Intelligence	GENE 121: Digital Computation
MTE 262: Microprocessors and Digital Logic	MTE 140: Data Structures and Algorithms
SYDE 548: User Centered Design Methods	SYDE 543: Cognitive Ergonomics

### Meta Marketing Analytics Professional Certificate Course

#### Relevant Courses:

Data Analytics Methods for Marketing	Statistics for Marketing
Python Data Analytics	Data Analysis with Spreadsheets and SQL

## Experience

### Freelance Developer [ December 2023 - Present | Remote ]

Node.js

Puppeteer.js

Git

#### Real Estate Report Extraction

- Developed a Node.js solution to automate data extraction from real estate reports using pixel positioning, eliminating on average 4 hours of manual time, then standardizing the data into a user-friendly CSV format for visualization.
- Collaborated closely with project stakeholders to create scalable and versatile solutions for handling missing and outdated data, enabling future data aggregation across various real estate markets.

#### Jobs Data Web Scraper

- Led a web automation project for a multi-million-dollar real estate firm to gather job postings across Greater Vancouver, developing custom scraping scripts with Puppeteer to gather postings by industry, salary, location, and remote work availability.
- Optimized scraping algorithms to reduce runtime by 50% from 1.5 hours to 45 minutes and implemented a modular code structure for scalability, while coordinating with stakeholders to continuously refine project requirements.

### Dozr [ Developer | May - September 2021 | Kitchener, Canada ]

Node.js

EJS

MongoDB

React

React-Redux

TypeScript

Postman

Git

- Solely accomplished overhaul of the Node.js backend and EJS email infrastructure, resulting in consolidation of follow-up emails for purchases across all Front-end platforms, accelerated completion to under 1 month for shareholder presentation.
- Re-implemented user purchase tracking, creating new order summary pages using React and Node.js, delivering a production-ready front end within 2 weeks.
- Fully implemented user analytics through tools such as Google Analytics and Bing Ad Conversion, connecting all user interactions within Dozr's primary ecommerce platform to the company's CRM for future marketing campaigns.

### BenQ [ Developer | May - August 2019 & January - April 2020 | Taipei, Taiwan ]

Node.js

React

Socket.io

WebRTC

Git

- Developed an extensive chrome extension for Mantis Bug tracker using React and JS, accelerating QA team velocity and reducing daily time spent reporting bugs by 10% (48 minutes).
- Prototyped stream broadcasting services using WebRTC and Socket.io, completing versatile proof of concepts within 1 month for presentation to multiple product leads.

### Newtopia [ Developer | September - December 2018 | Toronto, Canada ]

React

React Native

React Redux

Jest

Git

- Achieved improved error handling and user engagement by designing, creating, and implementing a Redux-based log for developer observation.
- Increased efficiency and compatibility of the mobile app's endpoint by refactoring and adding JSDoc documentation to all endpoints
- Enhanced code coverage from 80% to 90% of the entire code base by writing nearly 100 unit tests using Jest.
- Ensured WCAG compliance with accessibility enhancements across 20 web pages, as well conducted meticulous NVDA and JAWS screen reader testing, resolving all issues within 3 weeks.
- Routinely fixed critical bugs, including forking 3rd-party open source dependencies and editing/contributing to their code.