

Eric Cheung

705-333-9268 | erriccheung@gmail.com | github.com/Erric-Cheung | linkedin.com/in/erriccheung

EDUCATION

Carleton University

Bachelor of Computer Science; High Distinction

Ottawa, ON

Sep. 2020 – May. 2025

TECHNICAL SKILLS

Languages: Python, Java, C, C++, JavaScript, TypeScript, SQL, HTML, CSS

Technologies: React, Node.js, Express.js, Next.js, MongoDB, Postman, WebRTC, AWS, Figma, Selenium

EXPERIENCE

Signority

Aug. 2022 – Dec. 2022

UI/UX Designer Intern

Ottawa, ON

- Prototyped user interfaces using Figma, adhering to modern design principles and patterns to revamp and enhance the UI/UX of the web application
- Communicated with developers in ensuring the technical feasibility of designs and for seamless and accurate implementation of UI components
- Worked closely with the product manager to translate requirements into high-fidelity prototypes that satisfied both user experience and product goals
- Conducted multiple walkthroughs to test and gain valuable stakeholder feedback of prototypes, continually refining and improving the design for usability and accessibility

PRO JECTS

VolleyTok | *AWS, Next.js, PostgreSQL*

- Developed a responsive web application designed for sharing volleyball content using **Next.js**, Tailwind, and a **PostgreSQL** database with 8+ entities
- Allowed users to register, log in, and create new video posts using **Auth0** for secure user management
- Implemented infinite scroll with **IntersectionObserver API** to achieve **lazy loading**, improving initial page load times and enhancing user experience
- Enabled secure video uploads directly to **AWS S3** for authenticated users using Presigned URLs, paired with CloudFront for content distribution
- Built a verification layer using **AWS Lambda** to verify files after upload, enhancing security and ensuring file authenticity

Discord Clone | *React, Node.js, Express.js, MongoDB*

- Built a full-stack web application that replicated the features of the Discord application with real-time communication
- Used **JSON Web Tokens** generated during the sign-in process to authenticate and authorize users
- Developed **REST API** with Node.js and Express.js which communicated with a MongoDB database to store user data and chat messages
- Created a simple messaging system using **WebSockets** to allow for real-time messaging and notifications between users
- Implemented **WebRTC** to establish peer to peer connections to allow exchanging of audio, allowing calls between users.

Sorting Visualizer | *React, Redux*

- Developed a front-end web application using React and **Redux** to visualize sorting algorithms by highlighting key parts of the array while being sorted
- Implemented merge sort, quick sort, bubble sort, and insertion sort by displaying each step of the sorting algorithm selected
- Styled each of the components in the web application using **CSS Modules** allowing all class names to be locally scoped
- Visualized each sorting algorithm by using the current state of the array being sorted and highlighting key steps of algorithms