

# Austin Chanhsavang

auschanh@gmail.com | [Website](#) | [Github](#) | [Linkedin](#) | Milton, ON (Canadian Citizen, can relocate)

## Skills

---

**Languages:** Python, Java, JavaScript, Node, HTML, CSS, SQL, Unix, Linux

**Frameworks:** React, Apache Kafka, Flask, Tailwind, SharePoint Framework, Socket.io, Express

**Developer Tools:** Git, PostgreSQL, Docker, SharePoint, Power Automate, Confluent

## Certifications

---

**Confluent Certified Administrator for Apache Kafka**

December 2024

## Education

---

**Bachelor of Science, Computer Science**

June 2023

**Toronto Metropolitan University**

*Toronto, ON*

(formerly known as Ryerson University)

GPA: 3.48

## Work Experience

---

**Software Developer**

May 2021 – August 2022

**Ontario Ministry of Children, Community and Social Services**

*Toronto, ON*

- Developed customized **SharePoint** site with **SharePoint Framework (SPFx)**, and **React** ensuring compliance with Web Content Accessibility Guideline (WCAG), enhancing site usability
- Modernized team tracking ticket system using **Power Automate**, **MS Forms** and **SharePoint**, substantially reducing task assignment and data input time by **over 100%**
- Selected as a speaker to **300+** coworkers, delivering insights on daily client engagement and website creation, highlighting effective communication and project development abilities

## Projects

---

**Call Signs Game**

March 2024 – Present

- Developed **API** with **Express**, to generate a random word, and apply natural language processing techniques with regex to process user input
- Designed frontend Game page layout using **Shadcn**, **Tailwind** and **React** - created a vertical carousel for each Game phase, with a slider indicating current phase
- Implemented scoring system, processing data in backend and render to frontend with sockets

**React, JavaScript, Tailwind, Express, Socket.io, HTML, CSS**

**Book Review Website**

December 2019 – January 2020

[Website](#)

- Handled login, logout and session interactions with a relational (**SQL**) database using **Python (Flask)** for backend logic
- Utilized Openlibrary **API** using **REST** to obtain data for users to search, comment and rate
- Stored all comments and ratings by users in **database** - displays all user comments and ratings on account page using **SQL** statements, with join and like

**Python, Flask, HTML, CSS, PostgreSQL**