

# Emile Li

---

☎ 647-335-8899 | ✉ [emile.litim@mailutoronto.ca](mailto:emile.litim@mailutoronto.ca) | [github.com/emile29](https://github.com/emile29) | [linkedin.com/in/emile-li](https://linkedin.com/in/emile-li)

## Education

**University of Toronto** | Sep 2018 – Dec 2022

B.Sc. in Computer Science (Software Engineering Stream Co-op)

## Skills

**Languages:** JavaScript, TypeScript, CSS/SCSS, HTML5, Java, C, Python

**Frameworks:** React, Angular, NodeJS, Express, SpringBoot, Docker, GraphQL, MongoDB, MySQL

**Concepts:** SDLC, Scrum, Agile Methodology, OOP, REST API, AJAX, UI/UX Design

## Experience

### Software Developer

**CGI** | Toronto, ON | May 2021 – Dec 2021

- In charge of a project that helped speed up their manual E2E testing process by building a program, using Postman's Newman API, that automates the testing of several microservices all at once and producing a test report at the end.
- Worked with multilayer API components such as gateway, controller and repository.
- Use of JUnit for microservices test coverage and postman for E2E testing.

### Full Stack Developer

**Leonardo Worldwide Corporation** | Toronto, ON | Jan 2020 – June 2020

- Involved in developing a highly modular and configurable component-based UI framework using ReactJS, SCSS, Material UI, and Vue Patterns.
- Involved in developing an API Gateway using NodeJS, GraphQL, gRPC and Protocol Buffers.
- Maintained an efficient and fast delivery of large data between backend microservices to clients which consisted of online media galleries for travel agencies.

## Projects

**YouTube Watch Party (w/ React, MongoDB, GraphQL)** | [github](#)

- Use of web sockets to create virtual rooms and synchronize the currently playing video for everyone in the party with minimum delay.
- Use of PeerJS to create WebRTC connections for voice chats.
- Dockerized frontend and backend containers to be deployed on Digital Ocean.
- Implemented a secure connection between server-client using good security practices such as sanitizing user input, using the right cookie flags and using signed https certificates.

**Fanlinc (w/ Angular, MongoDB)** | [github](#) | [demo](#)

- Built a platform that works in similar fashion to Reddit with its most basic features such as posting, commenting, and upvoting or downvoting a post.
- Carefully designed database schemas and their fields to account for all relations between several schemas such as fandoms, users, posts and comments.

**YouTube Playlist Player (w/ Angular)** | [github](#) | [demo](#)

- Built an online audio player that retrieves playlists from either a youtube channel or your personal account and let you play your playlist of choice.
- Honed SCSS skills by carefully designing the layout to be responsive on different sized screens while providing the best possible user experience.