

Emile Li

☎ 647-335-8899 | ✉ emile.litim@mail.utoronto.ca | github.com/emile29 | 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, SpringBoot, JAXB, Docker, GraphQL, MongoDB, MySQL

Concepts: SDLC, Scrum, Agile Methodology, OOP, Design Patterns, REST API, Unit Testing

Experience

Software Developer

CGI | Toronto, ON | May 2021 – Dec 2021

- Worked with multilayer REST API components in Java such as gateway, controller and repository, while using corresponding Spring beans and JAXB configurations.
- 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 produces a test report of all collections at the end.
- Ensured at least 90% test coverage using JUnit and Mockito, and conducted E2E testing of microservices in Postman.

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 such as the State Management Pattern.
- Involved in developing an API Gateway using NodeJS, GraphQL, gRPC and Protocol Buffers.
- Maintenance of high-availability backend microservices in Java (SpringBoot, gRPC) that sort the data that is to be displayed on the client's website based on user preferences.

Projects

Watch Party App (w/ React, Node.js, GraphQL, MongoDB) | [github](#)

- Use of web sockets to create virtual rooms and synchronize the currently playing video for everyone in the party with minimum delay and use of 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, Node.js, 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.

Playlist Player App (w/ Angular, Node.js) | [github](#) | [demo](#)

- Built an online audio player using YouTube's Data API, that retrieves playlists from a youtube channel and lets you play your playlist of choice with interesting features like shuffle and loop.
- Honed CSS skills by making the layout responsive on different sized screens while following industry standards for managing a good and organized hierarchy of CSS classes and files.