Emile Li

Toronto, ON • 647-335-8899 • emile.litim@mail.utoronto.ca • github.com/emile29 • linkedin.com/in/emile-li

CAREER SUMMARY

With a passion for programming, adept problem-solving skills, and the desire to learn and improve, I always strive for excellence in every new project or challenge that I take on. As a software engineer, I also highly value teamwork as being a good player is what brings the next big idea from concept to realization.

EDUCATION

B.S. in Computer Science

University of Toronto

Sep 2018 – Dec 2022 (exp.)

SKILLS

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

Frameworks:
React, Angular, NodeJS, SpringBoot, JAXB, Docker, GraphQL, MongoDB, MySQL, MEAN Stack

Concepts:
SDLC, Scrum, Agile Methodology, Object-oriented programming, Design Patterns, REST API, Unit Testing

RELEVANT COURSES

- Data Structures
- Software Design
- Web Development
- Introduction to Databases
- Design and Analysis of Algorithms
- Engineering Large Software Systems

WORK EXPERIENCE

CGI | Software Developer (Co-op)

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.
- Minimized rollbacks by incorporating a Java script, into their continuous delivery pipeline, that merges multiple JSON collections into one in order to speed up the integration testing and produces a performance report at the end using Postman's Newman API.
- Ensured at least 90% test coverage of the codebase using JUnit and Mockito, and conducted E2E testing of microservices in Postman.

Leonardo Inc. | Full Stack Developer (Co-op)

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 AWS API Gateway using NodeJS, GraphQL, gRPC and Protocol Buffers, that redirects to the right microservice.
- 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, 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 chat feature.
- Use of three-tiered docker architecture to facilitate fast development and improve scalability and reliability.
- Ensured secure connection between server and client using good security practices such as sanitizing user input, using the right cookie flags and using signed and verified certificates.

Fanlinc **(w/ Angular, MEAN Stack)** github | demo

- Built a platform that works in similar fashion to Reddit with its most basic features such as post, comment, and upvote/downvote.
- Created a consistent UI theme from scratch while following industry standards for proper CSS management and hierarchy.
- Carefully designed database schemas and their fields to account for all relations between several schemas while keeping scalability in mind.