Marvin Li

li.marvin.nn@gmail.com | +1 437-259-6652 | Ottawa, Canada
github | linkedin | portfolio

Education

BASc Software Engineering - Year 3 **Currently Enrolled**

University of Ottawa *CGPA - 8.3*

Skills

Programming Languages Python | Java | C++ | C | C# | JavaScript | Go

Technologies FastAPI | PostgreSQL | Docker | Kubernetes | REST APIs | Git | React.js

Professional Skills Agile/Scrum | API Design | System Debugging | Technical Documentation

Work

Fisheries and Oceans Canada (DFO) **August 2025 - December 2025**

Software Engineering Student *Ottawa, ON*

- Engineered a dual-interface Grants and Contributions Management System with C# ASP.NET MVC, replacing manual, paper-based workflows with a secure web platform used by internal administrators and external applicants.
- Collaborated within an Agile Scrum team to modernize deployment pipelines; utilized Azure DevOps for CI/CD, managed source code with Git, and supported containerized application deployment via Kubernetes.
- Leveraged Entity Framework to design and implement a structured data architecture, facilitating the migration of manual grant tracking into a centralized, relational database system.

Fisheries and Oceans Canada (DFO) **January 2025 - May 2025**

Data Analyst Student *Ottawa, ON*

- Migrated large quantities of data across systems, ensuring data integrity, consistency, and compliance with standards.
- Collaborated with cross-departmental teams to address technical challenges in data management and system integration.
- Utilized Azure DevOps for Agile project management, version control, and efficient team coordination, ensuring timely completion of data migration tasks.

Notable Projects

TutorMonkey - Personal Project **December 2025 - January 2026**

Retrieval-Augmented Generation (RAG) | *Prompt Engineering* | *Vector Databases* | *Full-Stack* *https://github.com/iilou/topaz*

- Implemented a scalable Retrieval-Augmented Generation (RAG) pipeline for biology textbooks that embeds and retrieves semantically relevant academic content from a vector database to ground LLM responses, using ChromaDB with Gemini and OpenAI models.
- Engineered a secure, production-style full-stack web application enabling students to interact with an AI biology tutor in real time, featuring JWT-based authentication, FastAPI microservices, and a PostgreSQL-backed persistence layer.
- Developed an automated data ingestion and preprocessing pipeline for large biology PDF corpora using LangChain and Docling, including parsing, normalization, chunking, and metadata indexing to support efficient semantic search.

star.stylla.moe - Personal Project **April 2025 - May 2025**

Typescript | *Next js* | *Python* | *PostgreSQL* | *Docker* *https://github.com/iilou/starrailproject_fe*

- Built a full-stack web app with an interactive UI that ingests and aggregates player profiles, character builds, and item data from third-party APIs.
- Implemented a character leaderboard system that ranks builds across over 100,000 data points using stat-weighted formulas, with efficient server-side aggregation and sorting logic.
- Developed a FastAPI backend to manage API calls, perform real-time score calculations, and handle data persistence with PostgreSQL, efficiently adding and removing game data to ensure quick access and minimal latency.

Tetris AI - Personal Project **August 1, 2024 - August 31, 2024**

C++ | *SDL2* | *Machine Learning* *https://github.com/iilou/tetr.ai*

- Developed an advanced Tetris AI capable of being locally and autonomously trained achieving performance comparable to the top 10% of players of active players when set at a speed of 4 pieces/second after 300 generations of training.
- Engineered the AI model in vanilla C++ utilizing a heuristic-based algorithm to calculate the optimal placement for each Tetris piece, dynamically adapting to future piece sequences and real-time board states.
- Integrated SDL2 to build a comprehensive UI, allowing users to train AI locally with 3 distinct reinforcement strategies, compete against AI opponents at 5 different speeds, and adjust key parameters such as processing depth for up to 4 simultaneous pieces.