Jan Garong
Email: jan.garong@mail.utoronto.ca
Portfolio: https://jangarong.github.io/
GitHub: https://github.com/jangarong

Skills

- **Concepts:** Data Structures, Algorithms, Software Design, Design Patterns, SOLID, Agile, Cloud Development, Unix/Shell.
- **Languages:** JavaScript, TypeScript, C#, Java, Python, Solidity, HTML, CSS, SQL.
- **Technologies:** Amazon Web Services (AWS), Lambda Functions, DynamoDB, Google Cloud Platform (GCP), Firebase, Node.js, React.js, Angular.js, Nest.js, Next.js, Unity, Jest, JUnit, SQLite, Docker, git, Flask, GitHub CI/CD.

Education

• University of Toronto — Toronto, ON

Honours Bachelor of Science — *September 2019 – November 2023*

- Majored in the Computer Science Specialist - Software Engineering stream.
- Received a $2000 entrance scholarship, was part of the Dean's List for all 4 years, and graduated with Distinction.
- Learned algorithms and data structures, as well as basic software engineering and security.

Work Experience

• Trend Micro — Ottawa, ON

Software Developer — *January 2024 – Current*

- Working on Trend Vision One, an XDR solution that protects against cyber attacks. Currently assigned on improving the Email Security feature.
- Currently learning C++ and email protocols such as SMTP and IMAP, as well as common security issues such as phishing.
- Planning to use AWS services such as SQS, DynamoDB, and EC2 to help contribute to the product.
- Currently learning new DevOps tools such as Docker and Kubernetes.

• Blackberry — Mississauga, ON

Software Developer Student — *September 2023 – December 2023*

- Worked on CylanceGUARD, which is an AI-driven Managed Detection and Response (MDR) used to detect exploits on client devices.
- Implemented new integrations in the data pipeline with AWS Lambda functions and the AWS Simple Queue Service.
- Created features for multi-tenancy and alerts for the Java Spring Boot microservices, which adheres to MVC architecture.
- Stored critical alerts with PostgreSQL and DynamoDB and wrote automated tests with JUnit and jest.

• CertiK — New York, NY

Software Engineering Intern — *May 2022 – August 2022*

- Worked as a full-stack developer on Skynet, a platform that generates statistics related to the trustworthiness of blockchain projects, and Bug Bounty, a platform that rewards white-hat hackers for finding bugs in smart contracts.
- Wrote React.js code using the Next.js framework for developing server side rendered web UIs, and Vercel for CI/CD.
- Aided with improving user experience via Google Lighthouse to measure UI metrics such as performance and accessibility.
- Built microservices with JavaScript, AWS Serverless and Nomad for ease of scalability.

Security Engineering Intern (Part Time) — *October 2021 – December 2021*

- Conducted audits of EVM smart contracts, which including ERC20s, ERC721s, decentralized swaps and staking platforms, to identify vulnerabilities and centralization issues.
- Published many critical and major issues related to security and centralization, which prevented loss of funds.

• dApp Technology Inc. — Toronto, ON

Co-op Blockchain Full Stack Developer — *May 2021 – December 2021*

- Wrote smart contracts to implement common EIPs and helper functions, and used ethers.js to query blockchain nodes.
- Created full stack applications using Nest.js and React.js, and deployed them on the Google Cloud Platform (GCP).

Research Experience

• University of Toronto — Toronto, ON

Research Opportunity Program (Part Time) — *May 2022 – August 2022*

- Worked on a web application that uses artificial intelligence to group students in collaborative courses.
- Helped in creating REST API endpoints and UI components for a web application using Flask, Firebase, and React.js.
- Integrated an artificial intelligence model written in Python for forming groups based on personal student data.
- Used Docker for easy deployment, independent of the server environment.

Projects

• ZodiacTail — *July 2023 – January 2024*

- Designed and developed the core gameplay segments, as well as helped with fixing bugs in the UI and game itself.
- Used design patterns and object oriented principles in C# such as inheritance, observer and polymorphism, SOLID to write clean and reusable code.
- Used the Agile process and organized scrum meetings to keep track of progress.