Jay Fan

(416) 858-1488 | jingyangfan16@gmail.com | LinkedIn

Work Experience

Software Engineer 2022 – Present

Laserfiche

- Migrated our **AWS EC2** containers to a new architecture utilizing **Kubernetes** services, **AWS SQS** queues, and **AWS SNS** topics to improve the user experience and scalability of our application. Users were able to get instant feedback upon making requests, and the new service has not dropped a single request since the migration.
- Enhanced the performance of two **C#** microservices that made requests to our **PostgreSQL** database by implementing batch and asynchronous **RESTful APIs**. Our biggest clients with more than 20,000 users observed a **runtime reduction of 93%** when performing large-scale operations.
- Created **RESTful APIs** using **Typescript** and **Node** to serve client requests on our account control system. Daily unit, functionality, and webapi test results are monitored via **Grafana** dashboards, and they currently have a success rate of **98% over the last 90 days**.
- Implemented a user feedback system in **Python** and deployed via **AWS Lambda** functions. The system stored messages in **AWS Redshift** and **S3**, and allowed users to attach image and GIF files. After testing, the system was deployed to all user-facing products and is the primary way that we gather feedback from clients.
- Created an internal web application using **React** and **Flask**, enabling non-technical users to track their events in Redshift via drop-down menus. Managed the deployment with **Docker** containers, using **Nginx** as the reverse proxy for efficient application delivery.
- Executed a project to overhaul all user dialogues in our client-facing application using **React** and **Typescript**. Created reusable and extensible components that improved the developer experience, which led to the creation of user stories to refactor other front-end components in the same manner.

Software Developer 2021 – 2022

City of Toronto

- Led a project to create a web application using **Javascript** and **React** to visualize park utilization across the city. The application was used internally by city planners and policy makers to drive decisions on park maintenance and amenity development.
- Wrote and deployed **Python** scripts to automate the quarterly extraction of recreation program performance metrics, transitioning away from manual SQL queries. This automation reduced human error and decreased analysis times by **90%**.
- Designed and created data visualization dashboards using **Plotly** and deployed internally with **Docker**. Optimized data accessibility and reduced internal ad-hoc data requests by **over 80%**.
- Created and documented an internal system to automate our data pipelines using a **Flask** server run on a remote host. Allowed our dashboards to be updated on a fixed schedule without manual intervention.

Software Developer 2020

Doxim Solutions

- Developed pattern recognition scripts in **C#**, streamlining the conversion process of legacy **PowerShell scripts** into concise and reusable **C# components**. This significantly reduced manual effort, enhanced code maintainability, and promoted codebase modernization.
- Extracted, transformed, and analyzed credit union user data from internal databases using **SQL** queries. The insights derived from this analysis contributed to strategic decision-making processes.
- Spearheaded the design and implementation of a dynamic **C# data pipeline**, facilitating seamless integration with a third-party database. This initiative improved the usability, reliability, and performance of our internal data flow.

Education

Master of Data Analytics

Western University · Ontario, Canada · 2019 – 2020

Bachelor of Medical Science, Biochemistry of Infection and Immunity

Western University · Ontario, Canada · 2015 – 2019

Technical Skills

Programming Languages: Python, Typescript, Javascript, C#, SQL, Bash

Frameworks: React, Angular, Entity Framework, Vue, RxJS, Express, Flask, Django, Plotly

Tools & Technologies: Kubernetes, PostgreSQL, Node, Docker, Cloud Formation, Redshift, S3, DynamoDB, Nginx, Kibana, Grafana, Git