John Nnamchi

+1 (415)-823-4294 · john.nnamchi@simvo.io
30 Otis St., San Francisco, CA 94103

BSc. Major in Biochemistry from **McGill University**
Currently employed at Bloomberg LP under **TN US VISA** status

PROFESSIONAL SUMMARY

Experienced engineer specializing in **Python backend infrastructure** and **high-frequency-low-latency systems** for **quantitative workflows**. After optimizing Index Exchange's real-time ad exchange platform to handle **350 billion requests per day**, I joined Bloomberg to help build **BQuant**, Bloomberg's data science and quantitative investment platform. By completing various **Python, Database, Authentication and Infrastructure** projects, I've helped our department achieve **BQuant's browser migration** to a cloud provisioned web app available to all Bloomberg users.

SKILLS

Python Backend Development.	(Flask, Tornado & multi-thread workflows, DB interactions, JupyterLab)
Software Infrastructure, Cloud Provisioning	(GCP & AWS) & **Authentication** Workflows
Instrumentation and Telemetry Infrastructure.	(Kafka/Influx/Grafana, LogRocket
Statistical Modeling and **Data Visualization**	(Numpy, BQuantViz)
Golang Backend Development	(go concurrency, gRPC microservices)
Front-end Development	(React, VueJs, Typescript)

PROFESSIONAL EXPERIENCE

BLOOMBERG LP - Software Engineer

Oct 2021 - Present

BQuant Browser App Migration - 6 month long department-wide effort to deprecate BQuant Desktop App
Task For users to migrate to the BQuant browser app, they needed to be able to fully manage their project workflows which include creating & publishing projects, changing ownership, reverting to commits, and customizing environments
Implemented the backend infrastructure for these workflows: provisioning Python API services, exposing them to the internet, handling authentication, and programming the workflow specific logic (Database pulls/inserts for project snapshots and commit history, microservice calls for secondary workflows such as transferring projects, etc.)
Impact The BQuant Desktop app was deprecated, all existing users migrated to the browser, and BQuant is now made accessible to all Bloomberg Terminal users. It is now the base version when a user runs BQNT in the terminal.

BQuant Instrumentation Pipeline
Task Product / leadership had difficulty understanding user behavior and habits when working on a project, as well as tracing a user's actions that led to reported bugs
Implemented a message dispatch mechanism by hooking onto executed JupyterLab commands (cell run, notebook save, etc), collaborated with product teams to produce a standardized message format, and developed a common session ID to track the user across the various BQuant platforms (main projects view, JupyterLab & help/docs page)
Impact Telemetry provided a direct access to a user's trace actions leading to reduced user issue investigation time. The telemetry pipeline also initiated the conception of various BQuant features, such as git integration.

INDEX EXCHANGE - Software Engineer

June 2018 - June 2021

Built and deployed high-frequency-low-latency solutions that power real-time advertising auctions, to distributed systems on 6000 LINUX servers in 10 DCs worldwide, processing 350+ billion requests a day

Project Boomerang - optimizing auction latency to maximize advertising spend
• Implemented **code profiling tools** to identify various auction setup workflows as optimization targets
• Designed an **experimental framework** to quickly deploy experimental releases, paired with a **real-time data pipeline (Kafka / InfluxDB / Grafana)** to analyze optimization outcomes (advertiser bid response rates / spend)
• Identified key optimizations as (1) disabling non-voluntary context switches and (2) replacing a compression algorithm
Impact after MVP3 - auction latency **dropped by 40ms (25%), translating to $110K / day ad spend increase, and PROMOTED TO SOFTWARE ENGINEER II after project completion**

Bid Distribution - analyzing auction dynamics & bidding behaviour to reduce
• Wrote a **Numpy statistical algorithm** that produced a **Competitive Index** based on winning vs losing bids distribution
• This reduced the min price per ad leading to 10% savings for advertisers (10% more auctions won with same budget).