

# Kaichen Qu

• +1 (442) 423-8213 • Email: [kelsonqu@outlook.com](mailto:kelsonqu@outlook.com) • [LinkedIn](#)

## EDUCATION

**Northeastern University** | San Jose, CA  
Master of Computer Science | GPA: 4.0 / 4.0

Sep 2023 – May 2026

Coursework: Object-Oriented Design, Data Structures and Algorithms, Web Development, Database Management, Cloud Computing, Computer Networking

## WORK EXPERIENCE

**MIG Lab** | San Jose, CA

Feb 2025 – Present

Research Intern, Northeastern University | Python, Java, PyTorch, TensorFlow

- Built a multi-view UNet segmentation system using **TensorFlow/PyTorch** with edge-aware loss and Sobel filtering, improving MRI accuracy by 32%.
- Explored adaptive loss strategies by dynamically weighting Dice and Focal losses to enhance sensitivity to small and ambiguous structures.
- Built a Python-Java pipeline via **Py4J** for real-time image exchange, reducing model-GUI latency by 80% in MedVis-Suite integration.
- Optimized concurrent batch processing with Java's ExecutorService, added unit tests with **JUnit**, and maintained releases via **Git/GitHub**.

**BMW Group** | Shanghai, China

July 2022 – May 2023

Development Intern, R&D Department of New Technology | Python, MySQL, React

- Developed **Pytest** for automotive E/E chip validation, leveraged **PyQt** to build GUIs for real-time data visualization, generating more than 100 technical insights.
- Refactored complex **SQL** queries across internal startup systems (**MySQL** and **SAP HANA**), redesigned table joins and indexing strategies to reduce latency in cross-entity lookups for connectivity and security data and achieved a 30% reduction in query execution time during peak analysis hours.
- Optimized [BMW Startup website](#) by modularizing **React** components and decoupling UI logic, improving keyword-based search UX and cutting page latency by 30%, which led to a 20% increase in site traffic and 30% longer session durations.

**CDP Group** | Shanghai, China

July 2021 – Oct 2021

Development Intern | React, Node.js, Bootstrap, Taro, Jira, Jenkins

- Developed a CDP Youth platform by developing reusable **React** components and improving UX for modules like payroll and benefits, leading to **15% growth in daily active users** and increased client retention.
- Enhanced SaaS Platform backend services using **Node.js (Express)** and implemented asynchronous processing with **Kafka**, improving throughput and decoupling heavy-load payroll pipelines, reducing batch payroll load spikes by 25% and report generation time by 40% while handling tens of thousands of daily transactions.
- Enhanced **CI/CD** workflows with **Jenkins**, **Git**, and **Docker** to automate deployments and hotfix rollouts, boosting release velocity and stability; streamlined **Agile** processes in **Jira** to cut deployment times by 30% and post-release issues by 15%

## PROJECTS

**SaaS Short Link Platform** | Java, Spring Boot, Spring Cloud, Redis, MySQL, RocketMQ, Apache ShardingSphere, Sentinel

- Developed a high-performance short URL platform using **Spring**, **Redis**, and **MySQL**, enabling short link creation and rapid traffic redirection for millions of hits.
- Implemented asynchronous processing with **RocketMQ** and optimized caching, ensuring consistent low-latency performance under peak traffic loads.
- Integrated Apache **ShardingSphere** for horizontal sharding across multiple **MySQL** nodes, distributing read/write operations evenly and boosting query throughput by 3× during peak hours (eliminating single-node bottlenecks).
- Leveraged **Sentinel** for real-time traffic control and rate limiting, maintaining system stability during sudden traffic surges and achieved 10K+ link creations/sec and 50K+ redirects/sec at peak traffic, ensuring system reliability and SLA compliance.

**LiteKV – Lightweight Key-Value Database** | Go, Goroutines, Channels, TCP, Bitcask, Docker, Go Testing

- Designed and implemented the database in **Go** with storage engine, indexing, and network API components capable of handling 10M+ key-value pairs.
- Optimized disk I/O via **Bitcask-style log-structured design** and **memory-mapping files**, reaching 20K ops/sec write throughput and halving read latency.
- Implemented an in-memory **skip list** and **hash index** for fast lookups and range queries, delivering O(log n) query time on 5M keys with under 100MB memory use.
- Built a client-server architecture with **Goroutines**, handling 1K+ concurrent clients, ensuring thread-safe operations and sub-5ms 99th percentile query latency.

**Full-Stack Learning & Registration Platform** | Python, Django, Next.js, Node.js, MongoDB, React, Tailwind CSS, AWS (EC2, ELB, Auto Scaling, S3)

- Engineered a unified web platform for both course management and career fair registration using a **Node.js/Express** with **MongoDB** for flexible data storage.
- Built a responsive **React** frontend styled with **Tailwind CSS** for cross-device compatibility, building a highly modular UI component library (achieved 95% code reuse of components) to streamline development and maintenance.
- Secured all endpoints via **JWT** authentication, **OAuth2** logins, and fine-grained **RBAC** to protect 30+ distinct features, and managed global client state with **Redux**.
- Deployed on **AWS EC2** behind an **Elastic Load Balancer** and **Auto Scaling Groups**, offloaded static assets to **S3**, and optimized database queries to ensure zero-downtime scaling, high availability, and sub-second page loads under heavy traffic.

**Distributed RPC Framework** | Java, Netty, Zookeeper, Guava, FastJSON, Protobuf

- Developed a custom distributed RPC framework in **Java**, allowing microservices to register with a central registry and perform seamless remote procedure calls across nodes as if they were local (transparent service discovery and invocation).
- Designed a **proprietary communication protocol** to handle packet fragmentation and reassembly; built custom encoders/decoders supporting Java native **serialization**, **FastJSON**, and **Protobuf**, and leveraged **Netty** for non-blocking I/O to maximize throughput and minimize network latency.
- Integrated Apache **ZooKeeper** for dynamic service discovery and configuration management and implemented client-side caching with an LRU policy in **Redis** to speed up lookups; used consistent hashing to evenly distribute requests, eliminating single points of failure and improving overall scalability.
- Implemented robust fault-tolerance mechanisms: utilized **Guava Retry** for idempotent retry logic and added server-side rate limiting for graceful degradation and developed a client-side circuit breaker that isolates failing nodes and probes them for recovery, preventing cascade failures, ensuring high system availability.

## SKILLS

- **Languages:** Java, Python, Go, C/C++, SQL, JavaScript/TypeScript, HTML, CSS
- **Frameworks:** Spring Boot, Django, Node.js, Nest.js, Next.js, React, Gin, Bootstrap, Tailwind CSS, MyBatis, Hibernate, Protobuf, Redux
- **Tools:** Linux, Git, Docker, Kubernetes, Jenkins, Jira, MySQL, MongoDB, Redis, Zookeeper, Kafka, RocketMQ, gRPC, AWS, GraphQL, PyTorch, TensorFlow