Camille Chen

Seattle, WA • camilleching0317@gmail.com • 2062097666 • linkedin.com/in/qing-chen-camille/

Education

Northeastern University **Seattle, WA**

MS, Computer Science. GPA: 3.93/4.0 January 2023 - December 2025

Courses: Algorithm, Web Dev, Databases, Cloud Computing, Distributed System, Machine Learning, Computer System, Network

Skills: Java, JavaScript, TypeScript, Python, C/C++, React, Next, Node, SpringBoot, RESTful, Django, Redux, Valtio, Tailwind CSS

Tools: MySQL, MongoDB, Redis, Kafka, RabbitMQ, Firebase, AWS, Docker, Kubernetes, Nginx, Figma, Ant Design, Playwright

Experiences

Xiaomi Technology Wuhan, China

Frontend Engineer Intern May 2025 – August 2025

- Developed AI platform with **TypeScript**, **React**, Ant Design and **aHooks**, leveraging built-in hooks to accelerate development.
- Optimized **Webpack** configuration through advanced code splitting, dynamic imports, lazy loading, reducing initial load time**.**
- Architected scalable state management with **Valtio** for reactive UI updates and **useSWR** for intelligent data fetching with auto caching and revalidation, boosting API call performance.
- Reached **90%** end-to-end test coverage with **Playwright** by integrating parallel test execution, cross-browser testing and snapshot testing. Integrated into **CI/CD** using **Gitlab** Actions, reducing test cycles duration while maintaining UI stability.
- Improved release efficiency with **Docker** containerization for **Node.js** applications, **Kubernetes** orchestration with auto-scaling, rolling updates, health checks, and container registry management.

Peking University Institute for Computing and Digital Economy Changsha, China

Software Engineer Intern July 2024 – January 2025

- Developed AI-powered content generation platform with **React**, **TypeScript**, and **Next.js**, leveraging server-side rendering, static site generation, API routes, and middleware supporting **10+** daily active users for data labeling.
- Established reusable component library with **15+** encapsulated **React** components, reducing duplicate code by **40%**.
- Implemented streaming LLM responses via **WebSocket**, enabling real-time content preview as AI generates output.
- Improved **MySQL** efficiency by implementing composite indexing, query optimization, read replicas and database sharding, reducing query execution time by **20%**.
- Configured **Nginx** reverse proxy with caching layer and dual-environment routing for staging/production deployments.

A Round Entertainment Jersey City, NJ

Software Engineer Intern December 2023 – March 2024

- Implemented **Mapbox**-based map view allowing users to browse properties by location and apply real-time filters.
- Integrated **Whisper AI**-powered voice search functionality with advanced speech-to-text processing, **Natural Language Processing (NLP)** for query understanding, intent recognition algorithms, semantic search capabilities.
- Designed scalable **Google Firebase** backend with microservices architecture, handling real-time updates for 500+ concurrent users with **Firebase Authentication** and **Firebase Realtime Database**.

Projects

Full Stack Django E-commerce Website July 2024 – December 2024

- Deployed on **Heroku** utilizing **EC2 compute** with auto-scaling groups, **S3 object storage** for static assets and user-uploaded media files, **RDS** managed PostgreSQL database with automated backups and multi-AZ deployment for high availability.
- Implemented **JWT (JSON Web Token)** and **Role-Based Access Control** authentication with granular user roles.

Realtime Data Processing February 2023 – May 2023

- Designed and implemented scalable distributed system **using Java, Spring Boot, Apache Kafka,** and **MySQL** with event-driven architecture, enhanced system performance by 30% using **Redis** caching, **MySQL** query optimization.
- leveraging **Apache Kafka** architecture with **consumer groups** for parallel processing, and **dead letter queues** for error handling.
- Applied solid principles, design patterns (Factory, Singleton, Observer), dependency injection, comprehensive unit testing with **JUnit**, integration testing with **Mockito**, and containerized deployment with Docker.