

# Kayan Shih

Providence, RI | [kayan\\_shih@brown.edu](mailto:kayan_shih@brown.edu) | [linkedin.com/in/kayans](https://www.linkedin.com/in/kayans) | [github.com/kayans](https://github.com/kayans) | +1-401-248-8545

## EDUCATION

**Brown University** September 2023 – May 2025  
*Master of Science in Computer Science* GPA: 4.0/4.0 Providence, RI

- **Relevant Coursework:** Computer Networks, Computational Linguistics, Fundamentals of Computer Systems

**New York University** September 2019 – May 2023  
*Bachelor of Arts in Computer Science* GPA: 3.7/4.0 New York, NY

- **Relevant Coursework:** Operating System, Database, Computer System Organization, Machine Learning, Deep Learning, Responsible Data Science, Algorithm, Data Structure
- **Honor:** Dean's list 2019 & 2021

## PROFESSIONAL EXPERIENCE

**SlowMist** June – July 2023  
*Smart Contract Engineer Intern* Xiamen, China

- Improved security of DeFi products and NFT applications for 10+ projects by auditing and maintaining Ethereum-based infrastructure and smart contracts using **Solidity** and token issuance protocols ERC20 and ERC1155.
- Employed **Agile** methodologies for iterative development and cross-functional collaboration with 3 teams.

## PROJECT EXPERIENCE

**Distributed Key-Value Storage System** May – August 2024

- Designed and developed a distributed Key-Value storage system with 3 Nines of availability and support 1M QPS utilizing **Multi-Raft** architecture with linearizable K-V read/write operations using **Golang**.
- Leveraged **consistent hashing** to ensure even data distribution across shards and implemented robust data migration strategies for data transfer between different **Raft Groups** to enable dynamic load balancing.
- Implemented **async Apply** to process log entries without blocking, utilizing **ReadIndex** to perform read operations without leader election and integrated **FollowerRead** to allow followers to serve read requests, reducing the latency of read by **42%**.
- Integrated multiple storage engines, including **RocksDB**, B+ trees, and hash tables for different storage needs and performance optimizations; using **MVCC** to provide concurrency control.

**Event Ticketing Platform** May – August 2024

- Designed and developed a scalable ticket booking web application to handle ticket browsing, ordering, reselling, and payment expiration using **Node** and **Express**; containerized the microservices with **Docker** in **Kubernetes**.
- Developed and implemented the **NATS Streaming** Server as an event bus, promoting reliable event transfer between microservices and handling concurrent transactions with **optimistic locking**.
- Developed payment and subscription functionalities based on Stripe API, webhook and concurrent control.
- Created the dynamic fronted pages with **React**, **Next JS**, and **TypeScript**, using **JQuery** to respond to users' events and **MongoDB** for order management.
- Built a CI/CD pipeline using GitHub Actions to automate the testing process. Created unit tests with **Jest** and **Super test**. Performed end-to-end usability tests on concurrency control mechanisms using **JMeter**.

**AI-Powered Recipe Generation App** May – August 2024

- Built a serverless web application using **AWS Amplify** and **Amazon Bedrock**, integrating the Claude 3 Sonnet foundation model for AI-powered recipe generation; hosting frontend on CDN.
- Implemented **GraphQL API** with custom types to handle dynamic input such as arrays of ingredients, optimizing queries with batching and efficient resolvers to process user request.
- Created an efficient data model in **DynamoDB** for user and recipe data with sparse indexing, enabling Auto Scaling for the database to scale.
- Implemented backend using **AWS Lambda**, handling requests for user-submitted ingredients and invoking functions after user sign-up confirmation.
- Enhanced the web application's 2Auth by implementing an Email one-time-password authentication with **JWT** using **Amazon Cognito**.

## SOFTWARE AND LANGUAGE SKILLS

*Programming Language:* Python, Java, C/C++, Golang, JavaScript, TypeScript, Solidity

*Frameworks & Libraries:* React, Next.js, JQuery, Node.js, Express.js, Jest, Supertest, JMeter

*Tools & Platforms:* AWS, Docker, Kubernetes, GitHub

*Databases:* MySQL, PostgreSQL, RocksDB, MongoDB, DynamoDB, Redis