Xinyu (Jack) Li

Lawrence, Kansas | Portfolio website | 818.964.7221| Xinyujackli@gmail.com

Education

Washington University in St. Louis, Missouri — **St. Louis, MO**
MS, BS Computer Science | GPA: 3.88 — *Aug 2021 - May 2024*
Related courses: Object-Oriented Software Development Laboratory, Multi-Paradigm Programming in C++, Operating Systems Organization, Systems Security, Machine Learning, Web Development

Sewanee: the University of the South — **Sewanee, TN**
BS Computer Science & Mathematics | GPA: 3.77 — *Aug 2018 - May 2024*

Professional Skills

Programming languages: C++, Python, C, Java, JavaScript, Golang, Swift, HTML/CSS
Software/frameworks: React, Xcode, Spring Boot/Cloud, AWS DynamoDB, Lambda, SNS, S3, CloudWatch, Redis, RocketMQ, ElasticSearch, Docker, Sanity, Git

Experience

Turing Medical, **Software Engineering Internship**, St. Louis, MO — *May 2023 – August 2023*

- Developed a customer portal for managing and distributing patient MRI images.
- Integrated **JavaScript**, **HTML**, **Tailwind CSS** with **React** to develop the frontend and designed user-friendly, reusable components for efficient development.
- Utilized **React Hooks**, including **useState** and **useEffect**, to manage component state and lifecycle in functional components, improving code readability and maintainability.
- Built a set of **RESTful API** for order models with **Golang**, serving order requests through **microservice architecture**. Integrated with filtering, sorting and pagination, and employed **MySQl** for persistent storage.
- Enhanced product search functionality with **ElasticSearch**, utilizing an **Inverted Index** Search feature that improved search speed by 50% compared to MySQL fuzz query.
- Implemented identity **authentication** and **authorization** functionality in the user management model using **JWT** to ensure secure login.
- Developed a **Python** batch job to integrate with the **CircleCI API**, automatically retrieving large sets of test case results from both the Engineering and R&D teams. Consolidated these results into a centralized, password-free persistent storage, thereby facilitating seamless access to artifacts between teams.
- Maintained 80% code coverage with unit tests using **Pytest** and leveraged **pytest-mock** for dependency mocking, ensuring the application remains secure and testable.

Sewanee: the University of the South, **Research Fellowship**, Sewanee,TN — *June 2022 – July 2022*

- Built and configured a self-organizing **Raspberry Pi** 3 cluster, testing **MPI**, **OpenMP**, and **mpi4py** programs on both the cluster and a Raspberry Pi 4.
- Assisted a faculty member in developing and testing **C++** simulations for drug testing and forest fires on the cluster to evaluate its performance.

Projects

Online Shopping System, Lawrence, KS — *July 2024 – August 2024*

- Developed a high-traffic online shopping system handling up to 10K QPS during Prime Day sales.
- Employed **Redis** and **Lua** scripts to establish a distributed locking for implementing the **caching inventory lock** and **Revert of Try-Confirm-Cancel pattern**, completely preventing overselling happens for sale events.
- Addressed ambiguous requirements and use cases by designing an **entity-relationship diagram** from scratch and converting it into an **online shopping schema**.
- Implemented an **event-driven architecture** leveraging **RocketMQ** to support decoupled and asynchronous order records on MySQL.
- Set up microservices with **Spring Cloud Gateway** and configured **Spring Cloud Consul**, creating Consul cluster nodes to support **service discovery** and **server registration** across different modules.
- Streamlined **Docker** image and container management with **Dockerfile** scripts, enhancing **CI/CD** pipelines for automating builds, testing, and deployments throughout various stages.

Image Processing with AWS, Lawrence, KS — *July 2024 - August 2024*

- Implemented functional **AWS Lambda** functions, leveraged the principle of least privilege via **IAM Role** with precise IAM policy setup to ensure optimal performance in serverless architectures.
- Automated the generation of anime-style images using AWS Lambda, triggered by **S3 object creation events**.
- Designed a **CloudWatch Metrics** dashboard, set up **CloudWatch Alarms** based on Lambda errors, enabling real-time detection of service issues and automated notifications to administrators.
- Set up AWS CloudWatch Event to trigger **CRON** jobs for sending emails and messages to customers via **AWS SNS**, providing daily summary reports.