YIFEI SHANG

919-559-8735 | Availability: '25 Summer Internship / '25 Full-time
fayfeinekoshang@gmail.com | linkedin.com/in/Yifei | github.com/Yifei

EDUCATION

Northeastern University | **Seattle, WA**   Expected Dec. 2025
Master of Computer Science   GPA: 4.0/4.0

Dalian University of Technology | **Dalian, China**   Jun. 2023
Bachelor of Engineering   GPA: 3.7/4.0

Arizona State University | **Tempe, AZ**   May 2022
Exchange Student in Software Engineering   GPA: 4.0/4.0

TECHNICAL SKILLS

Languages: Java, Python, C/C++, SQL, JavaScript, HTML/CSS
Frameworks: Streamlit, SpringBoot, Flask, Django, React.js, Node.js, Express.js
Developer Tools: Heroku, Kubernete, Docker, AWS

INTERNSHIP EXPERIENCE

Koch Industries (INVISTA)   Jul. 2022 – Feb. 2023
Software Development Engineer Intern   Shanghai

- **Developed an access control function** using Java Spring Boot, enabling communication with over 500 physical devices via **Netty for TCP/IP socket communication**. Implemented a bi-directional protocol for sending user credentials and retrieving device statuses.
- **Implemented an authorization workflow** using Spring Workflow and Java, supporting multi-stage approval processes for 100+ users. Integrated **JavaMail** for sending notifications on status updates, with automated reminders for pending approvals. Developed state transition logic to handle 10+ authorization stages.
- **Built a React-based management dashboard** that supported **real-time monitoring** of devices for over 50 administrators. Utilized **WebSockets** to push updates within ms delay, and **React Hooks** for state management. Implemented responsive design with **Material-UI**, allowing seamless permission management and monitoring across desktops and mobile devices.
- **Developed a logging and audit module** using Java with Spring Data JPA and PostgreSQL, securely storing over 500,000 log entries per month. Implemented **encryption** for sensitive data and role-based access control to ensure that only authorized personnel could view or retrieve logs.

PROJECTS AND RESEARCH

LLM generated Lyrics   Summer 2024
Course Project   Seattle

- Developed a **RESTful API using Java SpringBoot**, enabling seamless integration with a custom LLM to generate sheet music based on user input. Designed the API endpoints to handle requests for different music styles and formats.
- Integrated LLM model hosted on **AWS SageMaker**, utilizing pre-trained models for musical composition. Configured **Lambda functions** to process API requests and dynamically generate musical notations, reducing response times for generating sheet music by optimizing memory.
- Built a **responsive frontend using React** with a focus on user interaction for real-time music generation. Implemented drag-and-drop features for modifying musical elements and a live music preview using **Web Audio API**, allowing users to interact with the generated compositions.
- Deployed the application on AWS using **Elastic Beanstalk** for backend scalability and **S3** for storing generated music files. Leveraged CloudFront for low-latency content delivery, ensuring smooth performance across different regions.
- Designed an **intuitive UI with React**, incorporating state management with **Redux** to efficiently handle user sessions and music composition data.

Food Ordering System   Spring 2024
Course Project   Seattle

- Developed a **Java SpringBoot** backend to manage restaurant order processing, utilizing **Spring Data JPA** for efficient database interaction. Implemented **asynchronous processing** to handle high volumes of concurrent orders.
- Built a React frontend with real-time order tracking, utilizing **WebSocket** to update order status dynamically. Integrated **Redux for state management**, ensuring efficient synchronization between the frontend and backend for order and payment updates.
- Integrated **Stripe API** for payment processing, creating **custom middleware in SpringBoot** to securely handle transaction requests. Configured **Auth0 for authentication**, ensuring secure user access to payment and order data.
- Deployed the system with **Docker containers**, ensuring consistent application environments. Configured **Kubernetes** to automatically scale backend services during high traffic periods, maintaining system reliability and uptime.