LIMING(Felix) ZHENG

www.linkedin.com/in/felix-cs | https://zlm23github.github.io/portfolio/ | 531-739-9370 | felixneucs95@gmail.com

EDUCATION

Northeastern University | MS in Computer Sciences (GPA: 3.82) **09/2022-12/2024**

- **Coursework:** Object-Oriented Design, Algorithm, Web/Mobile Development, Database Management, Game AI

University of Rochester | MS in Finance **06/2021-05/2022**

- **Coursework:** Foundations of Python, Financial Market & Investments, Financial Spreadsheet Modeling

University of Nebraska, Lincoln / BS in Finance **09/2016-12/2020**

SKILLS, ACTIVITIES & INTERESTS

Programming Language: Java, Python, JavaScript, TypeScript, HTML/CSS
Frameworks: React, React Native, Spring, Node.js, .Net Core
Tools: Elasticsearch, PostgreSQL, MySql, MongoDB, Redis, GCP, AWS, Azure, Kafka, Docker, Git, RPC, Linux

PROFESSIONAL EXPERIENCE

WorldEngine AI | *Software Engineer* **United States |** 01/2025-Present

- Developed internal Operation Management platform to assist Operator Managers in tracking task distribution and monitoring teleoperation data collection. ·
- Architected and implemented 15+ RESTful APIs using **Python** and FastAPI, ensuring high performance and security with **JWT** authentication and role-based access control (**RBAC**).
- Designed and optimized **PostgreSQL** database schemas, implementing indexing strategies, query optimization, and connection pooling with a singleton pattern, improving database query efficiency by 45%.
- Integrated **Elasticsearch** to enable full-text search and real-time analytics for task tracking and teleoperation data.
- Built frontend UI using **React** and **Ant Design**, developing 20+ components and implementing **Redux** for state management.
- Deployed microservices using **Docker** on Google Kubernetes Service (**GKE**), leveraging **BigQuery** for large-scale data analysis.

Verze LLC | *Software Engineer, Intern* **United States |** 06/2024 – 08/2024

- Developed microservices-based e-commerce platform (**C# + TypeScript**) deployed on **AKS** for real-time inventory management.
- Developed 10+ RESTful APIs (**.NET Core + Azure CosmosDB**) for product uploads, searches, and purchases.
- Optimized communication by combining **gRPC** and **RESTful APIs**, reducing request latency by 31%.
- Implemented **Redis** caching, reducing CosmosDB queries by 82%, improving data retrieval speed and reducing database load.
- Performed unit, integration, and load testing to ensure system reliability under high concurrency

Joblogic - X | *Software Engineer, Intern* **United States |** 06/2023 – 08/2023

- Built a video-sharing platform (**Node.js + TypeScript**) with user authentication and real-time data fetching, deployed on **AWS**.
- Developed 15+ RESTful APIs, optimizing video uploads, retrieval, and user interactions, reducing API response time by 35%.
- Leveraged AWS MediaConvert for video transcoding, reducing video processing time.

PROJECT EXPERIENCE

Distributed RPC Framework Implementation 12/2024-01/2025

- Implemented RPC framework in Java, focusing on service registration, publishing, and remote invocation in a distributed system.
- Developed a custom communication protocol to solve packet attachment issues and created encoders/decoders and serializers for Java native, **FastJSON**, and **Protobuf**.
- Leveraged **Netty** for high-performance NIO-based communication, optimizing network interaction and improving system efficiency.
- Integrated **Zookeeper** as a distributed registry and configuration center, reducing the load by leveraging Watchers for real-time updates to client-side service address cache, minimizing frequent direct queries to the registry.
- Implemented multiple **load-balancing** strategies, including service polling, random access, and consistent hashing to optimize service distribution.

Distributed Employee Management System 09/2024-11/2024

- Architected a distributed Employee Management System using microservices architecture to ensure scalability.
- Developed employee management services using **Java**, **Spring Boot**, and **MySQL**, implementing CRUD operations by RESTful APIs.
- Utilized **Apache Kafka** for asynchronous communication between services, ensuring efficient data processing and decoupling service dependencies.
- Configured **HAProxy** for load balancing, distributing incoming requests across multiple service instances to enhance system performance.
- Employed **Docker** for containerization, facilitating seamless deployment and enabling automatic scaling of microservices on AWS EC2 instances using Auto Scaling Groups.