

JIAN XU

736 Serra St, Stanford, CA 94305

☎ 650-519-7039

✉ xj1998@stanford.edu

📄 [jian-xu-4b9710253](https://jian-xu-4b9710253.github.io)

🔄 [JianXimple](#)

🏠 [Jian's Home](#)

Education

Stanford University

Sep 2022 – Jun 2024

M.S. Environmental Data, Statistics and Modeling GPA:3.8/4.0

Stanford, CA

Courses: Web Applications, Machine Learning, NLP, Deep Generative Models

Nanjing University

Sep 2016 – Jun 2022

B.S. Computer Science and Technology, B.S. Environmental Engineering GPA:4.4/5.0

Nanjing, China

Courses: Algorithm Design, Operating Systems, Networks, Advanced Programming, Database, Compiler

Experience

Renesas Electronics

May 2023 – Aug 2023

Intern: Software Development Engineer

San Jose, CA

- Designed AI-based solutions to identify fraudulent activities in sample order requests on Renesas' E-commerce platform.
- Implemented a data collection pipeline on **AWS**, using **Lambda Functions**, triggered by data uploads to an **S3** bucket. This process efficiently dispatched API fetch requests to **SQS**, with the resultant data stored in **DocumentDB**.
- Conducted data preprocessing, data augmentation, feature engineering, and trained **weighted-ensemble ML models** for fraud detection, achieving a **95%** accuracy post-production.
- Engineered a serverless cloud workflow on AWS, integrating **API Gateway** and Lambda functions to create a **RESTful** backend API, effectively managing order request processing.
- Deployed **SageMaker** endpoints to detect fraudulence for **real-time fraud detection** in order requests, which prevented **60%** financial loss, reduced **50%** labor cost, and decreased the overall order request processing time by **50%**.

NHB: Intelligent Medical Systems

Jul 2021 – Mar 2022

Intern: Software Development Engineer

Nanjing, China

- Fine-tuned an **OCR** model specifically for invoice image using PaddlePaddle, with an accuracy increase of **15%**.
- Implemented interactive bounding boxes using **React** on invoice images, enabling dragging and resizing adjustments around target area. This process segments the image into smaller sections, based on the bounding boxes' coordinates.
- Developed **RESTful APIs** using **Django** for image processing, formatting OCR model outputs, and saving to **MySQL** databases. Successfully onboarded 4 hospitals to the semi-automatic invoice recognition tool, achieving a **75%** reduction in manual invoice input time.

Projects

Concert Sharing and Ticket Selling Website | *React, Express, MongoDB, Redis, Docker, Kubernetes, NATS* Sep 2023

- Developed a single-page application for concert sharing and ticket selling, featuring sorting, filtering, and dark mode
- Utilized **Tailwind CSS** for styling, **Redux** and **React Query** for state management, and **memoization** techniques for performance optimization.
- Designed an event-driven microservices backend using **Express**, **MongoDB** and **Redis**, integrated **NATS** for event bus communication, and incorporated **Jest** for unit testing and **JWT** for secure authentication.
- Achieved efficient and scalable deployment on multiple clusters using **Docker** and **Kubernetes**.

Photo Sharing Website | *React, Spring Cloud, Kafka, Docker*

Jan 2023

- Engineered a responsive photo-sharing web application utilizing **React** and **Material UI**, incorporating features such as customizable photo uploads, commenting capabilities, and friend tagging
- Implemented microservices with **Spring Cloud**, managed configurations with the **Config Server**, and set up load balancing through **Eureka**.
- Utilized **Zipkin** to monitor distributed services and enhance the system availability.
- Developed message queue services and built a high-throughput log collection platform with **Kafka**.

I386 simulator | *C, Linux*

Dec 2020

- Developed a software simulation of an i386 computer system in C and successfully ran an RPG game on the simulator.
- Implemented the x86 instruction set, **cache scheduling** algorithms (**LRU** and **FIFO**), integrated memory management techniques (**segmentation** and **paging**), **multi-threading** and I/O device management.
- Emulated i386 components, including ALU, memory, serial ports, I/O, **interrupts**, exceptions, and **system calls**.

Technical Skills

Languages: C, C++, Java, Python, Swift(iOS), SQL, JavaScript, TypeScript, HTML/CSS, R, Matlab

Technologies/Frameworks: Pytorch, Sklearn, Tensorflow, CoreML, Qt(Cpp/Python), MongoDB, Express, React, Node.js, Selenium, Django, AWS, Azure, GCP, Spring Cloud, RabbitMQ, Kafka, Docker, Kubernetes