# Yifan Jiang

yifanjiang.me | yifan.jiang@duke.edu | LinkedIn | GitHub | (984) 327-9560 | GHC 23

## Education

**Duke University** Aug 2022 – May 2024
*M.Eng. in Computer Engineering | Concentration: Software Development* *Durham, NC*

**University of Liverpool** Sep 2018 – Jul 2022
*B.S. in Information and Computing Science | First Class Honour | GPA: 3.81/4.00* *Liverpool, UK*

## Technical Skills

**Languages**: Java, JavaScript, C, C++, Python, HTML, CSS, TypeScript, SQL, LaTeX, Visual Basic, Verilog
**Frameworks**: Vue.js, AngularJS, React.js, Node.js, Spring Boot, MongoDB, Django, OpenGL, Unity, PyTorch
**Developer Tools**: Docker, CI/CD, Mockito, AWS, Microsoft Azure, Postman, MySQL, MyBatis, Redis, Linux, GDB, Valgrind

## Experience

**Software Engineer Intern** May 2023 – Aug 2023
*Advanced Institute of Information Technology, Peking University* *Hangzhou, China*

- Developed a front-end UI for a power supply e-commerce website, utilizing **Vue.js** and **TypeScript**. Enhanced website performance using lazy and asynchronous loading, reducing page loading time by **25%**.
- Built various features including search, filter, and smart recommendations by employing **Elasticsearch** queries and indexing strategies, leading to a **83%** improvement in search efficiency and a **46%** increase in user engagement.
- Optimized back-end operations in **Spring Boot** by implementing **MySQL** and **MyBatis** for efficient retrieval and storage of electricity pricing data and **Redis** for caching, reducing response time for **RESTful** API endpoints by **52%**.
- Resulted in a launch with a user base exceeding **12,000** in a few months and an **80%** increase in web traffic after deploying the website onto Alibaba Cloud.

**Software Engineer Intern** Jun 2021 – Sep 2021
*Hikvision Digital Technology Co., Ltd.* *Hangzhou, China*

- Enhanced monitoring system for Starbucks with customer-driven features using **AngularJS** and **Vue.js**, including a critical mask detection feature to raise safety awareness during the pandemic, boosting in-store mask compliance by **50%**.
- Implemented the project with **Webpack** for efficient bundling and optimization of assets, **ES6** for modern **JavaScript** capabilities, and the integration of **Sass/Less** for maintainable and scalable **CSS**.
- Deployed the system across **500 stores** successfully, serving approximately **10,000** daily visitors, and achieved an impressive adoption rate of **99%** for the mask detection functionality.

## Projects

**RISC Game | Java, JavaFX, TCP Socket, CI/CD, Mockito, Docker** Feb 2023 – Apr 2023

- Organized a team of 3 to develop a multi-player game which enabled users to attack territories and obtain resources, move and upgrade soldiers. Developed backend server with Java and frontend UI with JavaFX and **MVC**.
- Improved server performance through synchronization and thread management, TCP sockets, and Docker containerization for accelerated deployment, contributed to maintaining **99%** server uptime.
- Attained a flawless **100%** unit test coverage with Mockito, reducing deployment time by **40%** through a CI/CD pipeline.
- Applied **OOP** and software engineering principles to design and draw UML diagrams and prototypes for effective system visualization, resulting in a **25%** boost in user satisfaction.

**Mini Amazon | Python, Django, PostgreSQL, Google Protocol Buffers, Bootstrap** Apr 2023 – May 2023

- Utilized Django and PostgreSQL to build a full-stack web app simulating Amazon, incorporating sequential consumption and transactional messaging in **RocketMQ** to ensure **100%** orders were processed, packaged, and delivered.
- Elevated development efficiency by **82%** via the application of Google Protocol Buffers to establish efficient and reliable communications across different warehouse simulators and delivery systems.
- Employed **TCP sockets** with the **ACK** mechanism to maintain app functionalities and performance at simulation speeds up to **100** times faster than the standard pace, tolerating at most **99%** flakiness of the world simulator.

**HTTP Caching Proxy | C++, Emacs, Multi-Threading, GDB, Valgrind, Docker, Git** Feb 2023 – Mar 2023

- Implemented an HTTP caching proxy server in C++ using Emacs, supporting GET, POST, and CONNECT **HTTP methods**.
- Designed a sophisticated **LRU**-based caching mechanism, compliant with **RFC7234** standards, achieving an impressive server cache hit rate of over **90%** for GET responses.
- Boosted server responsiveness by introducing **Mutex** locks and multi-threading, achieving a remarkable **400%** decrease in request processing time, capable of handling up to **1000** concurrent requests per second.
- Established robust resource management using **RAII**, leading to a **24%** reduction in memory leaks. Rigorously debugged the system using GDB and Valgrind’s Memcheck, enhancing overall system stability and reliability.