# Yang Zhao

Email: youngerstar213@gmail.com | Phone: 510-282-0057 | [linkedin.com/in/yang-zhao2](linkedin.com/in/yang-zhao2) | [github.com/IonZhao](github.com/IonZhao)

## EDUCATION

**Northeastern University** <span style="float:right">Oakland,CA</span>
*Master of Computer Science |GPA:4/4* <span style="float:right">Sep. 2023 – Now</span>
**Courses:** Algorithm, Database Management, Programming Paradigm, Web Development

**Zhengzhou University** <span style="float:right">Zhengzhou, China</span>
*B.Eng in Computer Science and Technology* <span style="float:right">Sep. 2018 – Jul. 2022</span>
**Courses:** Data Structures, Computer Network, Object-Oriented and Language

## TECHNICAL SKILLS

**Languages**: Java, JavaScript, Python, C, SQL, HTML/CSS, R
**Frameworks**: Spring, React, Node.js, Flask, PyTorch, JUnit, Bootstrap, jQuery
**Techniques**: AWS, Git, MySQL, Redis, Elasticsearch, Prisma, MyBatis, Nginx, Linux, REST API, Docker

## PROJECTS

**BlogMe - A Dynamic Blogging Platform** | *React, Node, Express, MySql, JWT* <span style="float:right">Jan.2024 –- Mar.2024</span>
- Developed a comprehensive blogging platform with modern frontend and backend technologies, enabling users to create, share, and interact with content seamlessly.
- Enhanced application security by implementing user authentication using JSON Web Tokens (JWT), resulting in a **50% reduction in unauthorized access attempts**.
- Built the frontend interface using React, ensuring a **responsive and dynamic** user experience.
- Engineered the backend logic using Express.js, streamlining HTTP request handling and reducing response times.
- Implemented RESTful APIs ensuring seamless communication between the frontend and backend systems.
- Optimized data storage and retrieval processes by leveraging MySQL, **resulting in a 40% decrease in database query times** and improving overall system performance.

**ForumHub - A Forum System** | *Kafka, Redis, Spring, Security, Mybatis, Elasticsearch* <span style="float:right">Aug.2023 – Nov. 2023</span>
- Implemented user authentication and authorization using Spring Security, with state checks performed through interceptors to protect resources that require login.
- Leveraged Redis as a local cache for frequently accessed data, resulting in a **70% improvement in system performance and faster response times** for user requests.
- Redis's set data type is utilized to address functionalities such as liking, disliking, and mutual following. Transaction management is used to ensure data correctness and data consistency with the cache and database.
- Utilized Kafka as a message queue to enqueue user actions like liking, commenting, and following, and delivered them asynchronously as system notifications, **decoupling the system and handling peak loads**.
- Stored post content information in Elasticsearch, enabling **fast searching of posts** and improving user experience by providing quick access to relevant content.
- Implemented a two-level cache using Caffeine + Redis and optimized the access to popular posts, achieving **a single-server QPS of up to 8000**.

**Luxury Shopping Platform** | *Nginx, MySql, SpringBoot, Redis, AWS, Master–slave* <span style="float:right">Feb.2023 — Jun.2023</span>
- Implemented user account creation and login using Alibaba Cloud SMS service. Implemented filters to allow access to login required resources and check login status.
- Integrated Redis caching with Spring-Cache to optimize server **response times by 40%** for frequently accessed product information, improving user experience and system performance.
- Adopted a frontend-backend separation approach and asynchronous data transmission, resulting in a **30% reduction in API response times** and improved scalability for handling concurrent user requests.
- Implemented Sharding-JDBC for **read-write separation**, enhancing database performance by 35% and ensuring **data consistency and availability**.
- Deployed the project on two AWS Cloud ECS servers, leveraging Nginx reverse proxy for frontend deployment and Spring Boot's Tomcat server for backend deployment, achieving **99.9% uptime** and improved system reliability.