Andy Wang

1930 Mount Vernon Court, Mountain View, CA 94040
510-449-8925 andygqwang@gmail.com linkedin.com/in/guanqiao-wang-9955b8270 github.com/andygqw

Education

University of California, Davis **Dec 2023**
Bachelor of Science in Computer Science *Davis, California*

Relevant Coursework

- Algorithm Design
- Data Structures
- Operating Systems
- Computer Networks
- Parallel Architecture
- Programming Languages
- Computer Architecture
- Machine Learning

Technical Skills

Languages: Java, C#, C/C++, Python, GoLang, SQL, Typescript, JavaScript, HTML/CSS, Bash
Frameworks: Spring Boot, .NET, React, Node.JS, Flask
Technologies: Agile, Amazon Web Services, CloudFlare, Linux, Git, Github Actions, MySQL, MS SQL Server, Redis, RabbitMQ, Docker, CI/CD, Postman, Wireshark, CUDA, Jupyter

Certificate

2024 | aws AWS Associate Solution Architect(SSA-C03)

Experience

Technology Services & Solutions, Santa Clara County **July 2023 – July 2024**
Software Developer Intern III | Fullstack: .NET, MS SQL Server, Github Actions *San Jose, CA*

- Spearheaded the development of employee management application using the .NET framework, MS SQL Server with Dapper, and optimized SQL queries to enable efficient data searches and seamless questionnaire for over 14,000 clients.
- Architected and implemented a user access management for the county services, and developed REST APIs to enhance security by accommodating varying access requests from different groups and roles.
- Optimized the deployment of the user access management backend using GitHub Actions and integrated Postman CLI tests to ensure reliability, resulting in a 70% improvement in deployment.
- Created a NuGet .NET package to streamline API response processing, significantly reducing redundant coding efforts for the entire team, now actively utilized in over 25 instances across the codebase.

Blueboard **Jul 2022 – Sep 2022**
Software Engineer Intern | Backend: Spring Boot, MySQL, Redis, RabbitMQ, Linux *Remote, US*

- Developed backend REST APIs to process order information on Linux (Ubuntu) using Redis and MySQL, optimizing concurrent request handling with multi-threading Java, reducing API response time from 3 seconds to 0.6 seconds.
- Implemented Redis caching and a delayed double-deletion strategy to optimize queries on product inventories, reducing MySQL load and ensuring data consistency with distributed locks in a multi-server environment.
- Integrated backend APIs with RabbitMQ to establish a microservices architecture, effectively decoupling services for improved scalability and maintainability.

Projects

Efficiency Boost Tool | *Javascript, React, Node.JS, CloudFlare* **June 2024**

- Designed, developed, and deployed a comprehensive web solution that enables users to manipulate text and words, access cloud file storage, utilize a Facebook type posting/replying playground, and access online resources, all in a fully self-built environment.
- Integrated a Single Sign-On (SSO) service for multiple services, and implemented JWT for user authentication and management through Cookies, significantly improving login/logout processes and request handling efficiency.
- Deployed Cloudflare Workers as a serverless backend to a registered domain, utilizing Cloudflare D1 and R2 for relational database and durable object storage. Configured R2 object lifecycle policies to maintain optimal file storage health.
- Utilized React in CloudFlare Pages with Material UI to implement a modernized UI/UX design, significantly enhancing the user interaction experience.

Overflowed Subtitle Generator | *Java, OpenAI, Concurrent, IntelliJ Idea* **May 2024**

- Architected AI subtitle generator specifically designed for multiple long videos content, providing an efficient and accurate solution for AI generating subtitles.
- Leveraged the OpenAI Whisper Speech-to-Text AI model to interpret speech information with generative accuracy. Enhanced accuracy significantly by fine-tuning with roles and effectively prompting the model through API integration.
- Configured Java Thread Pool with Semaphore to handle parsing, request making, and combining audio with generated texts, ensuring thread safety and achieving a minimum 25x speedup in processing time.