YILONG(NICK) WU

587-372-1183 | yilongwu17@gmail.com
Linkedin Profile | Porfolio

EDUCATION

University of Alberta, Edmonton, Canada *09/2020 - 06/2022*
Master of Engineering (MEng) *Software Engineering*

Northern Arizona University, Flagstaff, USA *06/2018 - 12/2018*
Exchange student

Chongqing University of Post and Telecommunication, China *09/2016 - 06/2020*
Bachelor of Engineering (BEng) *Electrical and Computer Engineering*

WORK EXPERIENCE

Associate Developer, Intern, Northwestern Polytechnical University, Xi'an *09/2019 - 11/2019*
- Assist lab researchers in the development of a project on scale invariant based digital image feature recognition, matching and fusion.
- Participated in the development of image preprocessing functions, including histogram equalization processing, image rotation, image scaling and image radial distortion correction.

SKILLS AND HIGHLIGHTS

- Strong knowledge of Java, data structure, Spring framework and algorithms.
- 1 year back-end web development experience.
- Experience with SpringBoot, JavaScript, Python, Rust, MySql, Hibernate, Docker, Mybatis, Elastic Search, Maven, Git, RabbitMQ, Bootstrap, Node, SpringCloud, Microservice, Mongoose, Redis, MongoDB, Express, Thymeleaf, Vue

PROJECTS

Happymall: A distributed microservice e-commerce website
- *Project Demo* *Github Repo*
- Happymall is a distributed architecture e-commerce project based on microservices, this project implements SpringCloud architecture and uses SpringBoot to build. Using modular development, with different functions divided into separate microservice modules, like order service, cart service, user service, etc. Use Nacos as service discovery and configuration.
- Use Spring OpenFeign to handle remote procedure calls between services and services, SpringCloud Gateway acts as a routing gateway.
- The project includes a front-dest shopping mall system and a backstage management system. The functions of the mall system include user registration/login, product search, cart, place order, and other functions. The Backstage management system can manage products, orders, and users.
- Introduce CompletableFuture to implement asynchronous computing to speed up the rendering of product detail pages.
- Combining RabbitMQ message TTL and Dead-Letter-Exchange to realize RabbitMQ delay queue (scheduled task), realize the automatic closing of expired orders and ensuring the eventual consistency of the distributed system.
- Use elastic search as the full-text search engine, and optimize the ES storage structure to improve the ability of high concurrency.
- Utilize Redis as a cache to solve the scenario of reading more and writing less, and use MyBatis-Plus as a persistence framework.
- Use Lua scripts and a unique token to prevent users from submitting the same order repeatedly, and ensure the idempotency of the interface.
- Integrate GitHub OAuth2, use SpringSession and Redis to solve session cross-domain problems in the distributed system.

Jupiter: Personalization-based recommendation engine for events
- *Project Link (Deploy on AWS EC2)* *Github Repo*
- Developed a personalization-based recommendation engine for events, provided an interactive web page for users to search events, update preference and view recommended events.
- Develop the backend using java, which includes three servlets, and utilized MySQL to store user preference and event information.
- Developed a web service using (Java Servlet, REST API) to fetch event data from TicketMaster API.
- Designed a content-based recommendation algorithm to recommend events based on the categories of user's favorite events.
- Improved precision of recommendation by ordering events based on distance and matched categories.

GoCamp: A campground sharing website based on Node and Express
- *Project Link (Administrator username: admin, password: admin)* *Github Repo*
- Designed and developed a campground sharing website based on Node, Express, and Bootstrap.
- Parse the user input location as latitude and longitude and display the location on the cluster map.
- Use Mongoose as ORM framework, utilize MongoDB to store data, and introduce Cloudinary as cloud storage to store upload images.
- Developed a middleware with passport package to implement authentication and authorization.
- Implemented Server side and client side validation to prevent unauthorized operation, database injection cross-site scripting and other security problems.