Yijiashun(Elijah) Qi

Personal Website
Github: github.com/elijahqi

Email: elijahqi@umich.edu
Mobile: +1-734-450-5998

Education

- **University of Michigan** Ann Arbor, MI
  Bachelor of Science - Computer Science and Data Science; GPA: 4.0/4.0 *Sep 2021 - May 2024(expected)*
  Honors/Awards: Three terms University Honors, James B. Angell Scholar (Two consecutive A terms)
- **The Hong Kong Polytechnic University** Hong Kong
  Bachelor of Business Administration - Financial Services; *Sep 2019 - May 2021*
  Honors/Awards: Two years Dean's List (Awarded to top 10% of the whole Business School)

Technical Skills

- **Programming language**: C++, Python, HTML/CSS, JavaScript, SQL, R
- **Frameworks/Technologies**: React, Bootstrap, jQuery, Flask, Jinja, Hadoop, Node.js
- **Tools**: Git, Linux

Professional Experience

- **School of Information, University of Michigan** Ann Arbor, MI
  Research Assistant with Professor Jinseok Kim *May 2022 – Present*
  - Developed a program to read and parse all JSON files, including those in subfolders, using the author's name and paper, and assigning unique IDs to each article and author.
  - Implemented Union find to merge authors with names that had a similarity of greater than 85% as determined by difflib.SequenceMatcher.
  - Enhanced a Random Forest model by adding an "ethnicity" feature through the creation of a transformer object and modification of the distance estimator, resulting in an improvement in model precision from 77.5% to 85%.
- **The Pacific Securities** Shanghai, China
  Automobile Research Team Intern *July 2021 – Aug 2021*
  - Conducted market research and analysis of the Geely Auto, including market trends, competitor analysis, and consumer behavior.
  - Created a financial news dataset by writing a web crawler to gather data from financial news articles using Requests, Beautifulsoup, and Selenium. Utilized Pandas to clean and prepare the data for analysis.

Projects

- **Insta485 Web Project**
  - Created an Instagram-like web application using React for the front end, Python with Flask for the back end, and SQLite for the database.
  - Implemented features such as creating, updating, and deleting users, posts, comments, likes, and infinite scrolling using event listeners and browser routing.
  - Deployed the website using an EC2 instance on Amazon Web Services.
  - Solved the issue of automatic updates not occurring for likes and comments by refactoring the code to use object-oriented programming and implementing shared state management.
- **Wikipedia Search Engine**
  - Built a scalable search engine with a pipeline of MapReduce programs, a Flask-based RestAPI index server, and a React-based front-end search interface.
  - Implemented a MapReduce framework with distributed processing on a cluster of computers in Python, utilizing TCP for task communication and UDP for fault tolerance.
  - Improved efficiency using the subprocess.Popen class to run the reduce executable in a new process.
  - Developed the search interface using React as a client-side dynamic page, sending queries to the index server and using the response to update the page.
- **To-Do List**
  - Created a Node.js application using the Express framework and implemented CRUD functionality for to-do list items
  - Used MongoDB with the Mongoose library to store data and applied error handling and debugging techniques
  - Implemented a custom list feature that allows users to create their own to-do lists, using the EJS template engine to render HTML pages and the body-parser library to parse HTTP request bodies.