Muhammad Zaman

(226) 751-5104 | mh2zaman@waterloo.ca | https://www.linkedin.com/in/muhammad-hamza-zaman/ |
https://github.com/mhzaman-cs | **Full Resume: muhammadzaman.tech/resume.pdf**

Technical Skills

Languages: JavaScript, Java, TypeScript, Python, HTML/CSS, C, C++, C#, SQL (MySQL, MSSQL), R
Frameworks and Tools: React.js, Node.js, Material-UI, Bootstrap, Firebase, Tailwind CSS, REST API, Git/GitHub
Awards and Scholarships: PPG Canada Scholarship (2021), University of Waterloo President's Scholarship (2021)

Education

University of Waterloo — Waterloo, CA
Bachelor of Computer Science, Minor in Statistics — *Expected Graduation: 2024*

Experience

Achievers Inc — May 2022 – Aug. 2022
Software Developer Intern — *Toronto, ON*

- Developed a **React component library** with **25+ standardized components** on the **UI Systems Team**
- Ensured the library met the **WCAG 2.1 Accessibility standards** by adding **ARIA labels** to components
- Migrated external dependencies such as **Material-UI** out of the library by rewriting components which **decreased load time by 8%** and made it easier for the architecture team to stage and deploy the library to production

SPARK — July 2021 – June 2022
Software Developer — *Remote*

- Developed a user-friendly website using **React** and **Bootstrap** to educate **500+ children** in developing countries
- **Reduced load-time by 17%** by code-splitting, utilizing CDNs, minifying code and removing unnecessary plugins
- Created a multiple choice quiz section using **ASP.NET Core** as the back-end and an **SQL database** for storing the questions which were used in **800+ quizzes** taken by **200+** students

CrowdDoing — May 2020 – Aug. 2020
Data Scientist Intern — *Remote*

- Collected **data of 65+ herbs** from different sources including the **National Library of Medicine** using **data crawling** techniques through **Python** with libraries such as **Scrapy and BeautifulSoup**, in order to provide data to the analytics team with the product's potential benefits, safety concerns, reactions, etc.
- Cleaned **35+** unstructured data sets through libraries such as **Pandas** and **NumPy** for standardization
- Applied cluster analysis techniques such as **K-means clustering** to classify **100+** items by nutrients and ingredients into categories for the **recommendation engine** so that only items within clusters are recommended
- Assisted in the development of a **recommender system** to suggest items based on user interaction with clusters
- Constructed the end-to-end system using **TensorFlow** for covering the entire stack from serving with **ScaNN** for retrieval, through ranking with **TF ranking**, to post ranking, while leveraging **multitask learning** in the process

Projects

Citadel Data Open | *Python, Plotly, Seaborn, Pandas* — March 2022

- Wrote a report in a team of 2 about how investments in businesses and education affect traffic in major American cities and provided recommendations on how congestion can be reduced through investments in these areas
- Cleaned and structured multiple provided and external data sets with **1000000+** entries and used that data to come to **statically significant** conclusions about congestion in New York, NY, Austin, TX, and Washington, DC
- Graphs are generated using the **Python** libraries **Plotly** and **Seaborn**, and the data is organized using **Pandas**

Stockify | *Python, Plotly, Seaborn, Pandas* — May 2021

- Created a stock visualizer capable of displaying an interactive graph of any of the **S&P 500** stocks with different comparisons such as Opening value, Daily High, Daily Low Price, Closing value, or Volume traded compared against the date from 2013-2018 based on a **Kaggle** data set
- The graphs are generated using the **Python** library **Plotly** and the data is organized using **Pandas**

Amazon Reviews Scraper | *Python, Scrapy* — July 2021

- Leveraged **Python** and **Scrapy** to scrape product reviews from different products based on **ASIN** number
- Utilized Scrapy's built-in boilerplate and implemented the scraper components such as the **HTML** parser for scraping content on pages and the initiator which loops through the different products using **OOP principles**
- Added cool down and opened tabs in-browser to prevent the program from getting caught in Amazon's CAPTCHA