Muhammad Zaman

(226) 751-5104 | mhzaman.cs@gmail.com | www.linkedin.com/in/muhammad-hamza-zaman/ | github.com/mhzaman-cs

Technical Skills

Languages: JavaScript, Java, TypeScript, Python, C, C++, C#, SQL
Technologies: Pandas, TensorFlow, MongoDB, Express, React, Angular, Node, Spring Boot, ASP.NET Core

Education

University of Waterloo Waterloo, CA
Bachelor of Computer Science, Minor in Statistics *Expected Graduation: 2025*

Experience

Achievers Inc May 2022 – Aug. 2022
Software Developer Intern *Toronto, ON (Remote)*

- Developed a **React** component library with **25+** standardized components to ease the development process
- Migrated documentation from Gatsby to **Storybook.js** and configured it manually to employ SCSS, SVGs aliases using **Webpack5** and also configured storybook internals like the docs, canvases, and controls
- Ensured the library met the **WCAG 2.1** Accessibility standards by adding ARIA labels with proper tab indexing to components, and testing their accessibility with JAWS and NVDA screen readers
- Met and exceeded the **80%** coverage threshold of **JavaScript** unit testing by writing **50+** React unit tests to ensure that the components and their attributes are working as expected

SPARK July 2021 – April 2022
Software Developer *Fremont, CA (Remote)*

- 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 students
- Authenticated **200+** student accounts to access quizzes using **ASP.NET Core** with **C#** to validate credentials
- Set up automated emails which sent **2000+** emails to student accounts confirming registration, quiz results, etc.
- Developed a front-end educational platform using **React** and **Bootstrap** to teach **500+** children
- Reduced load time by **17%** by code-splitting, utilizing CDNs, minifying code and removing unnecessary plugins
- Standardized design outputs with a mobile-first approach which increased mobile user's satisfaction rates by **29%**

CrowdDoing May 2020 – Aug. 2020
Data Scientist Intern *San Francisco, CA (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 **Beautiful Soup**
- Processed **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 into nutrient categories
- 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

Forex | *MongoDB, Express, React, Node, Firebase, Tailwind CSS, JWT* December 2021

- Full-stack **MERN** app featuring a discussion form based on **Firebase** that allows for P2P Currency Exchange
- Uses a **MongoDB** database to store login credentials with an **Express.js** server to handle user authentication
- Utilizes Fixer.io's API to display exchange rates for **170** currencies as recommendations on the dashboard
- Keeps user authenticated in order to retain access to the API and chat by using **JWT** signature verification
- **React**-based front-end is focused heavily on **UX/UI** by using frameworks like **Tailwind CSS** and **PostCSS**

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 cities
- Cleaned and structured multiple provided and external data sets with **1000000+** entries and used that data to come to statically significant conclusions about congestions 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**

Bank Account Manager | *C++* July 2022

- Created an optimized banking system in **C++** which uses data structures, such as Hashmaps and BSTs to efficiently store account information and allow for time-efficient access to it
- Enforced multiple user types which have different authorization levels allowing for specialized user tasks