Mobassera Zaman
+1 438 924 1770 | mobassera.jemi@gmail.com
GitHub: https://github.com/mobasserazaman | Portfolio: https://mobasserazaman.github.io/portfolio24/

EDUCATION & QUALIFICATIONS

Bachelor of Science in Software Engineering
McGill University, Canada **September 2017 – June 2021**
Relevant Courses: Algorithms and Data Structures, Algorithm Design, Operating Systems, Concurrent Programming, Computer Networks, Applied Machine Learning, Basics of Web Development, Introduction to C++, Discrete Structures

Udemy React - The Complete Guide 2024
This course includes. React Router and Redux **2023-2024**

SKILLS

Technical/Computer:

- **Programming Languages:** JavaScript, Java, TypeScript, Python, C++, HTML, CSS3, JSON, SQL
- **JavaScript Frameworks and Libraries**: React.js, Redux, Express.js, Node.js
- **CSS Frameworks:** TailwindCSS, Bootstrap
- **Java Framework:** Spring Boot
- **Operating Systems:** Linux Ubuntu, MacOS
- **Version Control**: Git, GitHub
- **Build Automation and Containerization**: Docker, Maven

Behavioral: Committed, Avid learner, Passionate, Optimistic, Team Player

WORK EXPERIENCE

Software Engineer
HubSpot, Canada **April 2022 - November 2022**

- Designed and developed user facing features for HubSpot Sales Chrome Extension and Outlook Add-In
- Extensively used React components and hooks, Redux, React Router, TypeScript, JavaScript, JSX, CSS, HTML for development
- Wrote unit tests as well as Selenium tests for automated UI testing for quality assurance
- Used GitHub for version control, code reviews and collaboration
- Investigated issues reported by customers and deployed fixes to production
- Attended meetings with product designers and coworkers to understand requirements of new features, report progress and roadblocks
- Gained exposure to Chrome extension development, Chrome APIs and Office APIs

PROJECTS

Recipe Sharing App **2024**

- Built backend REST API for creating, deleting, fetching, saving recipes etc using the Node.js web application framework Express.js
- Used Mongoose ODM to manage, store and retrieve data from MongoDB database
- UI is built using React, React Router for navigation, Hooks for state management and side effects, CSS for styling etc
- Features include creating new recipes, user authentication using JWT, saving recipes, viewing recipes shared by other users

Weather app **2024**

- Basic web app built with React.js
- Uses OpenWeatherMap API to fetch data
- Functionality - search weather by location (city)

Portfolio using React **2023**

- Developed a portfolio with interactive UI using React.js, React Router, JavaScript and Tailwind CSS.
- Hosted the site using GitHub Pages.

McGill School of Computer Science Website **Fall 2020**

- Developed a complete website with over 70% of the code base built from scratch.
- Designed a responsive and interactive front-end using HTML5, CSS and JS as a minimum.
- Set up a working back-end using PHP to support the dynamic update of contents and MySQL to build an on-server database, and later to connect to it.
- Utilized XAMPP to create a local web server on the computer to fast-track development and make local testing of the codes feasible.
- Used SSH to transfer files to the McGill SOCS web server.
- Collaborated with teammates to choose technology stack, divide tasks, design, and document the project.

Simplified Link State Routing Protocol with Java Socket Programming **Winter 2020**

- Designed a pure user-space program which simulates the major functionalities of a routing device running a simplified Link State Routing protocol.
- Implemented a router class and mapped the ‘Process IP’ and ‘Process Port’ of each program instance to a ‘simulated IP address’ attribute used to identify the router in the simulated network space.
- Added a Link State Database which is a map from the router's IP address to the link state description which is originated by the corresponding router and Link State Advertisement structures that summarize the latest link state information of the router.
- Developed a command line-based terminal interface for the router to allow the user to input the following commands: attach, start, connect, disconnect, detect, neighbors, quit. Some of these commands trigger Link, State Database synchronization and update.
- Built a weighted graph representing the topology of the network and used Dijkstra’s algorithm to find the shortest path between routers.
- Implemented a message class that defines the message format transmission among routers.

Logistic Regression and Naive Bayes using Python **Winter 2020**

- Implemented two classification techniques in Python —logistic regression and naive Bayes from scratch on four distinct datasets and compared the accuracy of both classification techniques on the four datasets.
- Cleaned the datasets by removing any data with missing or malformed features and used one-hot encoding for categorical variables.
- Computed basic statistics such as distributions of the positive vs. negative classes, distributions of some of the numerical features, correlations between the features etcetera to get a better understanding of the data.
- Implemented a script to run k-fold cross validation.
- Compared the accuracy of naive Bayes and logistic regression on the four datasets.
- Compared the accuracy of the two models as a function of the size of the dataset (by controlling the training size).