Mohammed Wasif Uddin

 $Irving, TX \mid \underline{mwasifu@outlook.com} \mid (682)\text{-}716\text{-}6147$ $\text{https://www.mwasifu.com} \mid \underline{\text{https://github.com/mwasifu}} \mid \text{https://www.linkedin.com/in/mohammed-wasif-uddin-10b6b1201/}$

EXPERIENCE

Software Engineer, HM & RC; Dhaka, Bangladesh — Jan 2021 - June 2021

- Architected and developed a .Net 5 (Core) Windows database management application using WPF with C# and XAML following material UI to manage entries for over 600 properties.
- Introduced and incorporated new features like custom report generation and to-do list over the legacy system to increase revenue by up to 37%.

EDUCATION

• B.S in Computer Science
The University of Texas at Dallas

Jan 2016 - Dec 2020

- Richardson, TX

TECHNICAL SKILLS

- Frameworks/Libraries React.js, Vue.js, Node.js, Bootstrap, JQuery
- Programming Languages JavaScript, HTML, CSS, Java, C#, C++, Python, C, Ruby
- Database MongoDB (with Mongoose), SQL
- Version Control/Software Tools/IDEs VS Code, Visual Studio, IntelliJ IDEA, Git, Postman

SOFTWARE PROJECTS

Portfolio Website

- React Single Page Application with Bootstrap responsive across all screen sizes.
- Axios used to utilize OpenWeather API to fetch geolocation coordinates to display current location weather details (city, temperature, highs, lows, etc) in real time.
- Tensorflow.js speech model used to receive user voice commands to manipulate object movement on screen.
- State manipulation using useState hook to add and delete tasks on to-do list and useEffect hook to initialize tsparticle.js in the background when website initiates

Blog Website

- Vue.js with Vuetify used on the frontend with Node.js on the backend with MongoDB.
- Collaborated and developed REST APIs with backend engineer to perform CRUD operations on the database to manipulate users, blog posts, etc.

DragonRuby GTK Top Down RPG

 Collaborated with a team of 3 while leading in UI to develop a cross-platform Fire Emblem clone game in Ruby using the DragonRuby Game Toolkit