SYED WAJIH RIZVI

syedwajihrizvi2000@gmail.com | https://syedwajihrizvi.com | https://github.com/syedwajihrizvi
University of Waterloo | B.A.Sc. in Mechanical Engineering with Minor in Computing | Class of 2023

PROFESSIONAL SUMMARY

Motivated and experienced software developer with over 8 years of programming experience. Professional experience in several aspects including automation testing, APIs, robotics, and multiprocessing. Looking to make an impact for a company seeking a developer with a comprehensive tech stack including front-end and back-end technologies.

WORK EXPERIENCE

Ford Motor Company | Lead Software Automation Engineer *2023-Present | Kanata, ON*

- Maintained and developed the test automation of multiple components through functional, integration, and hardware tests
- Maintained a component code coverage of over **90%** and a branch coverage of over **95%** on all product code
- Provided **100% rainy day** coverage across all testing requirements
- Reduced code duplication to **under 2%** across all managed repositories using SonarQube analysis Refactored code to reduce cognitive complexity via analysis of SonarQube
- Interacted with **Linux** via **UbuntuVM** from a Window's Host to simulate a vehicle environment to execute tests
- Developed hundreds of test cases using **Python's Slash Testing Framework** that adhered to software release requirements
- Worked with **ara.Com** library to utilize proxies in test cases as well as mock skeletons to perform several test scenarios
- Debugged failing test cases using **Python Logs** as well as **DLT logs** and worked with Developers to fix issues
- Configured **Jenkins Continuous Integration** pipelines to run test suites for functional, integration, and hardware tests
- Managed test case information and tracked results with **TestRails and** followed appropriate defect lifecycle using **JIRA**
- Refactored existing code and developed clean Python code to increase **SonarQube** quality and reduce bugs

Ford Motor Company | Embedded Bluetooth Software Automation Engineer *2021-2023 | Waterloo, ON*

- Developed test cases that covered various Bluetooth protocols using **Python's Slash Testing Framework**
- Programmed unit tests in Python to mock several API calls and ensure a minimum **90%** code coverage
- Optimized Jenkin nodes to improve Sanity run time by nearly **28%**
- Developed APIs and CLI tools that provide several text analysis methods for log files
- Debugged automation scripts through log files, breakpoints, and GUI inspection
- Utilized **Appium** to conduct automation tests and analyze android applications
- Performed version management using Git Bash and contributed to company's Continuous Integration Pipelines

Solace | Software Engineer in Test *2020-2021 | Kanata,ON*

- Developed functional test cases using **TCL** to verify several API Endpoints
- Utilized **SVN** to perform version management
- Replaced API functionality to work with **JSON** instead of XML
- Tested API Endpoints using Postman to verify data
- Developed and refactored front end code using **Java**

ADP Canada | Process Consultant *2019 | Toronto, ON*

- Analyzed **100s** of operating procedures and developed SOPs to document them
- Utilized metrics such as time, cost, and usage to identify bottlenecks in processes
- Optimized analysis time using Excel to properly determine improvement opportunities
- Worked with **RPA** team to develop algorithm to determine cost savings via automation
- Identified process improvements that could save the company upwards of **$500, 000** annually
- Participated in **SCRUM** meetings with managers and directors to update them on automation

RECENT PROJECTS

GameCom | *Typescript, NodeJS, MongoDB* *2025*

- Built website providing detailed information on **millions** of video games using React TypeScript, Node.js, and MongoDB
- Developed reusable, optimized React components using TypeScript and ChakraUI, ensuring high performance and scalability
- Configured several robust API endpoints using Express.js to handle **CRUD operations**, reducing response time by **over 70%**
- Connected backend to MongoDB, optimized database queries with Mongoose, and utilized asynchronous queries for speed

- Utilized React Query to implement **infinite queries**, cache API responses, and reduce network usage by **90%**
- Designed and implemented fluid, responsive animations using Framer Motion, enhancing user engagement
- Developed custom query hooks for efficient data fetching from IGDB, resulting in real-time updates
- Implemented **JWT-based authentication** to secure restricted endpoints, preventing unauthorized access
- Built fully responsive React components compatible with various devices and screen sizes
- Successfully launched and scaled the website using Heroku with and **version control** via Git, achieving **99.9% uptime**
- Utilized promises for server calls to implement asynchronous and non-blocking code Built website providing detailed information on **millions** of video games using React TypeScript, Node.js, and MongoDB
- Developed reusable, optimized React components using TypeScript and ChakraUI, ensuring high performance and scalability Configured several robust API endpoints using Express.js to handle **CRUD operations**, reducing response time by **over 70%**
- Utilized React Query to implement **infinite queries**, cache API responses, and reduce network usage by **90%**
- Designed and implemented fluid, responsive animations using Framer Motion, enhancing user engagement
- Developed custom query hooks for efficient data fetching from IGDB, resulting in real-time updates
- Implemented **JWT-based authentication** to secure restricted endpoints, preventing unauthorized access
- Built fully responsive React components compatible with various devices and screen sizes
- Utilized promises for server calls to implement asynchronous and non-blocking code

CScape | *Typescript, NodeJS, MongoDB, Python* *2024*

- Built a website that provides information on **thousands of cities worldwide**, enabling users to plan customized tourist trips
- Integrated **Google Places API** to fetch data on destinations like restaurants, hotels, amusement parks, and more
- Developed reusable, highly performant React components using Material-UI, ensuring a seamless user experience
- Implemented the **OpenWeather API** to display live weather data and 7-day forecasts for any city
- Utilized **Zustand** to manage global state efficiently, preventing prop drilling and improving component scalability
- Integrated **OpenAI's API** to dynamically generate city descriptions, enhancing user engagement with personalized content
- Built a robust backend API using Express.js for **CRUD operations**, providing real-time data updates to users
- Designed a scalable MongoDB schema with optimized queries using Mongoose for fast and efficient data retrieval
- Implemented **JWT-based authentication** to secure user login and restrict access to certain CRUD operations
- Used **Google's Distance Matrix API** for an optimized **Traveling Salesman algorithm**, reducing ravel time and distance
- Built hooks with React Query for efficient server calls, caching, and network optimization, resulting in faster load times
- Developed a responsive layout compatible with mobile, tablet, and desktop devices, ensuring accessibility across platforms

Autonomous Mecanum Wheeled Robot | Python, Linux, Raspberry PI, Excel *2023*

- Utilized Raspberry Pi and Python to program autonomous robot that could navigate any terrain using omni-directional wheels
- Developed a seamless Continuous Integration Pipeline and managed version control using GitHub
- Implemented multiprocessing for motor control, sensor collection, and localization
- Programmed several classes in an Object-Oriented Paradigm to aid in scalability and organization of code
- Developed Plant Model that could predict wheel velocities for a robot given location and destination using an odometric matrix Localized lidar data using the concept of moving average to filter out noise and outliers in order to accurately determine position Collected data points to develop interpolant equations that converted duty cycle to RPM for input into motor velocity functions Imported and programmed code that worked with several python packages including rplidar, pyexcel, and numpy