# **JAINISH MEHTA**

► 519-781-4847

igainishmehta09@gmail.com

in LinkedIn GitHub Website

## SKILLS 🍄

#### Languages/ Frameworks/DBs:

C++, C, Java, Python, Javascript / Typescript (intermediate), Matlab, R, HTML, CSS, Selenium, MySQL

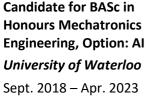
- Other: Git, Docker, GTest, AWS, Node.js, JUnit, RMarkdown, Latex, Postman, Jenkins, Micro Focus UFT, Jira, Linux Ubuntu
- Data Analysis
- SolidWorks, AutoCAD, GD&T
- Knowledge of PLC and Microprocessors
- Rapid Prototyping, Laser-Cutting, Machining
- 1000+ volunteer hours over 5.5 years, solidifying my leadership, problem

   solving & project management skills

## AWARDS ¥

- 3<sup>rd</sup> Place, Canada Chemistry Contest (Alberta/NWT)
- 2<sup>nd</sup> Place, 2018 Interdisciplinary Science Competition
- 2<sup>nd</sup> place team in Canada, CyberPatriot X Security Competition (placed top 2% worldwide)

## **EDUCATION**



#### RELEVANT EXPERIENCES

## Full Stack Software Developer (Cybersecurity), Red Canari Inc.

 Created a web application and chrome extension using HTML, CSS, and React.JS. Used AWS for backend services and creating RESTful APIs; specifically used AWS Lambda, AWS Cognito for user authentication, and the AWS Amplify framework. Serverless application served as email proxy to mask original emails with shadowed ones.

#### III DevOps/Quality Services Software Engineer, Manulife Financial

Jan – Apr 2020

Aug 2020-present

- Used *Java* and *JUnit* to develop new features, enhancements in test reports to improve user experience, and health checks for ALM and DevTest. Setup *Jenkins* pipelines for existing projects.
- Integrated *Perfecto* platform within existing *Selenium* framework (TestNG and Cucumber) to fully automate an efficient mobile app and mobile browser testing process.
- Used *Postman* to parse information on Perfecto statistics and *SQL* to query these statistics in order to visualize it on Grafana.

## Robotics/Software Engineer, Mission Control Space Services Inc. May – Aug 2019

- Developed functions and unit and integration tests in C++ (primarily) and Typescript to enhance a *path planning algorithm*; these include creating one for cubic spline interpolation, using XML and data parsing to classify terrain classes, and getting ROS nodes to subscribe LiDAR point cloud data.
- Used linear algebra to analyse sensor data from an IMU and stereo camera and then implemented a graph in C++, through nodes, in order to develop a *calibration routine* that determines the tilt of the stereo camera mounted on a rover.
- Formulated system design documentation for the navigation & control subsystems of a lunar rover, including **sensor fusion research** and modelling camera specifications, such as field-of-views.

## Waterloop Mechanical & Embedded Software Team

Jan 2019 – Present

- Modelled parts of the braking system and battery enclosure for the Hyperloop using Solidworks.
- Worked with team members to read and interpret message outputs from RobotEq motor controller.
   Used C to create an Arduino program to interpret signals for a rotary encoder.

## SIDE PROJECTS **Q**

### **Movie Recommendation System**

May 2020

• Created a simple movie recommendation program using **Python** and skicit-learn library. Used vector analysis based on select features to find similarities between two objects.

### **SmartVision: Object Recognition (Mostly Frontend)**

Sept 2019

Used Swift to develop the user interface for a prototype iOS app which aimed to help impaired
individuals distinguish their surroundings. Aided in incorporating a CoreML model for Apple's Vision
framework to enable computer vision functionality.

#### Thrifter: Finding an Optimal Match Between Images

Feb 2019

- Worked with a team to design parts of frontend and backend for a web application that incentivizes
  users to thrift shop by finding a match between a database of business-side clothing and an
  uploaded user's image; used HTML, CSS, and Javascript, specifically jQuery to make API requests.
- Utilized Google Cloud Platform Vision API for *computer vision* functionality, a MongoDB database to store images, and *K-nearest neighbours algorithm* to find an optimal match between two images.

#### **Aktiv: Typing Robot**

Oct - Nov 2018

- Automated a typing Lego Mindstorms EV3 robot using *C++ and RobotC*, which takes text files as user input. A computer keyboard was mapped using hash tables.
- Employed PID controllers and rack and pinion models to increase robot's typing accuracy.