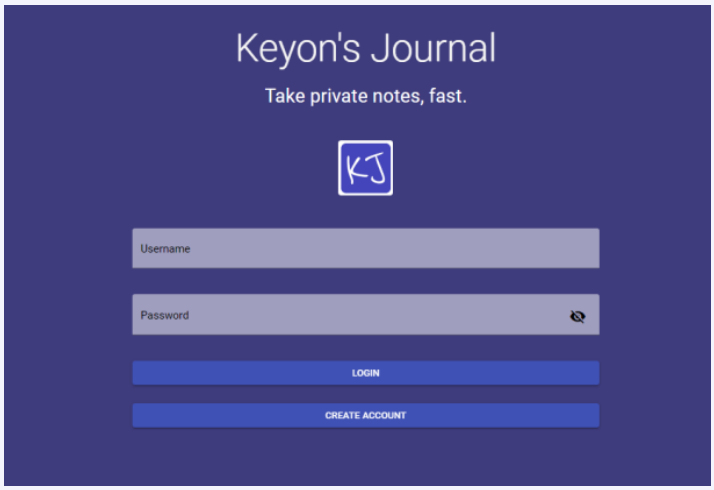
SOFTWARE PORTFOLIO

Keyon Jerome

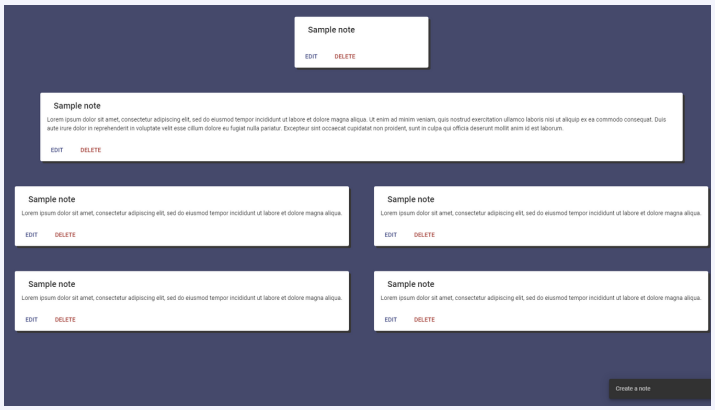


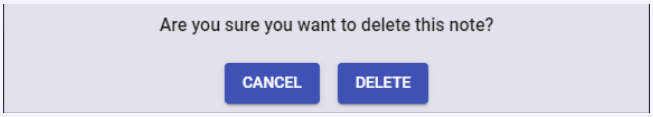
KEYON'S JOURNAL

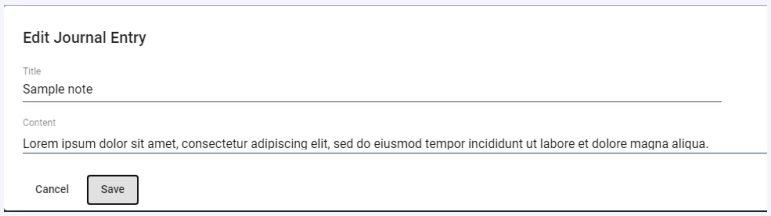
ANGULAR | MYSQL | PHP

At STEP Software, I was given the opportunity to shadow software developers in their Agile environment while learning **Angular, SQL/MYSQL, PHP, and C#** in the .NET framework. Over the course of 2 weeks (after learning Angular for the first two), I created Keyon's Journal; an Angular web journalling app complete with login, password resets, routing and keyboard shortcuts.

On the frontend, I used **Bootstrap 4** (Angular Bootstrap) for quick development and a mobile-first, responsive layout. The app consists of two main pages, with pop-up modals to handle account creation and note creation. In the bottom left, a Material Snackbar is used as a floating overlay to create sticky notes regardless of where you are on the page.







The backend was manually hosted on a remote **phpMyAdmin MYSQL** database. HTTP requests are sent to the server backend and the PHP code makes SQL calls to the database, altering it and returning the relevant information—users previous posts, with all their metadata, or upon a delete request, notification that the information was properly deleted. For example, upon user creation, the user's password was hashed on the front-end and sent to the phpmyAdmin backend.

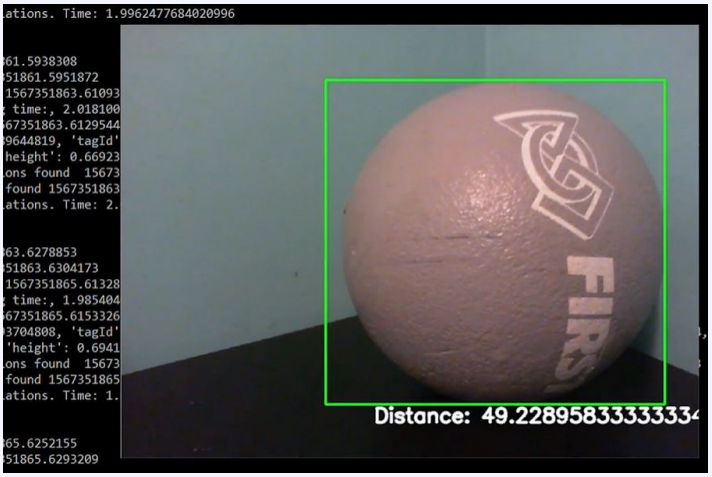
keyon.io
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca

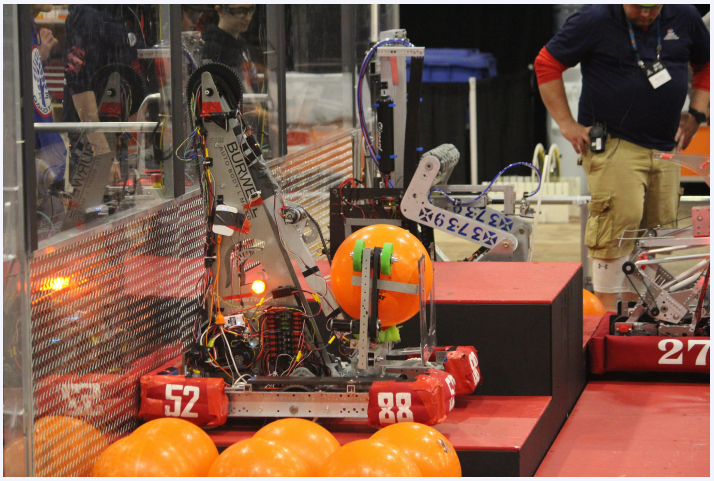
FRC COMPUTER VISION & ROBOT CODE

PYTHON | JAVA | MICROSOFT AZURE

For the past two years in FIRST Robotics Competition, I've worked on vision tracking (computer vision) for our team's robots, as well as the robot's main code. My computer vision programs, running on embedded systems such as the **Jetson TX1** and **Raspberry Pi**, use video cameras and lights to detect field objects in real-time.

By building a library of labelled images of each game field object, I was able to use **Microsoft's CustomVision.AI** on the **Azure** platform to develop a machine-learning model that could detect game objects. The machine-learning model exported to **Tensorflow** in **Python**. From there, I used computer vision techniques and frameworks such as **OpenCV** to find out where the game object was in 3D space, relative to the robot's camera.





Finally, the result of the real-time image processing and calculations is sent over to the robot's control system via ethernet. From there, the robot's **Java** code is written to take in the result and use **PID** motion control to drive itself to the target.

Winner of the BOS Innovations 2019 Raspberry Pi Programming Contest

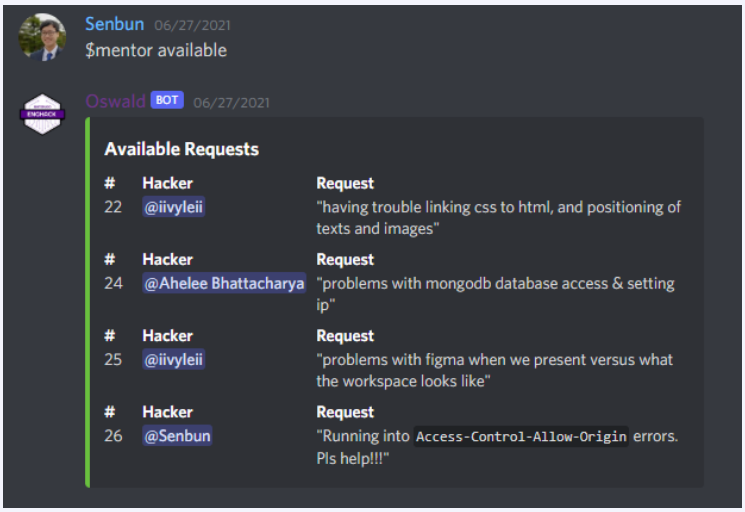
keyon.io
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca

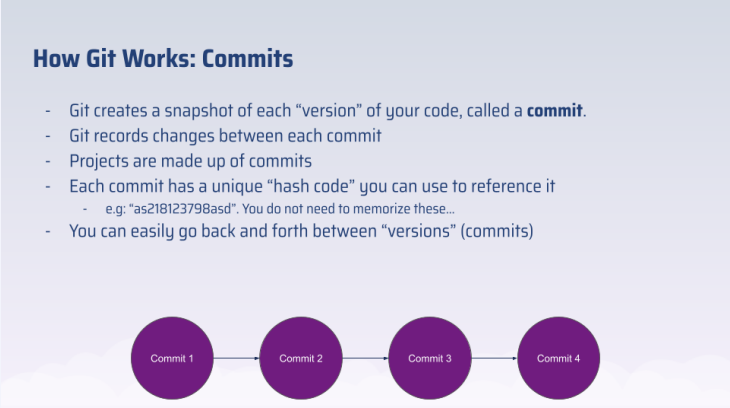
ENGHACK: DISCORD BOT, GIT WORKSHOP, AND WEBSITE

REACT | GIT | MJML | NODE.JS

This year, I volunteered with EngHack — Waterloo Engineering's annual hackathon as a developer and workshop lead. I co-hosted their **Intro to Git and GitHub** workshop, created a mentor-hacker pairing request system for their **Discord bot**, designed and coded their email templates with **MJML** and **Figma**, and coded much of their website with **React**.

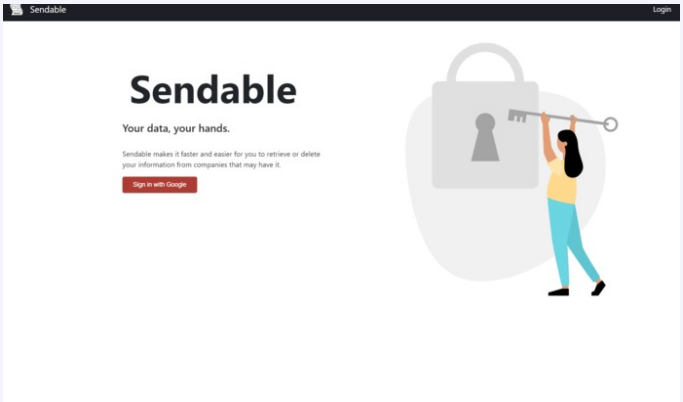
Discord bot: With **Node.js** and **Discord.js** (although written in TypeScript for cleaner code), I created a ticketing system where hackers could create their own requests for mentorship, and mentors could access, view, and claim hackers' quests for help with their projects. Once a mentor claimed a hacker's ticket, the mentor and hacker would both be notified of their match and the mentor would reach out to the hacker for assistance.





Git workshop: I co-hosted EngHack's **Intro to Git and GitHub** workshop, giving a simple presentation on Git theory (why we need it and how to use it) as well as its integrations with GitHub, and how first-time coders could use it to better their projects. You can find the slides *here* and the recording *here*.

keyon.io
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca



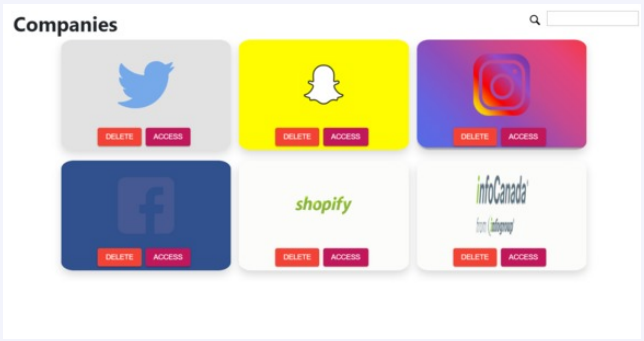
SENDABLE

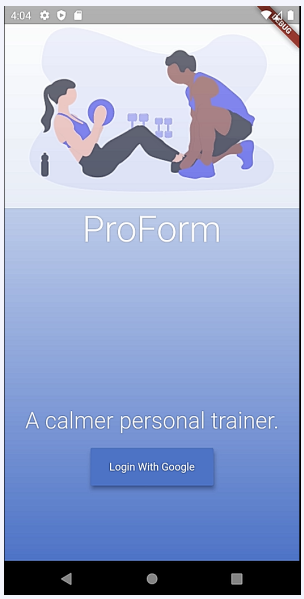
ANGULAR | FIREBASE

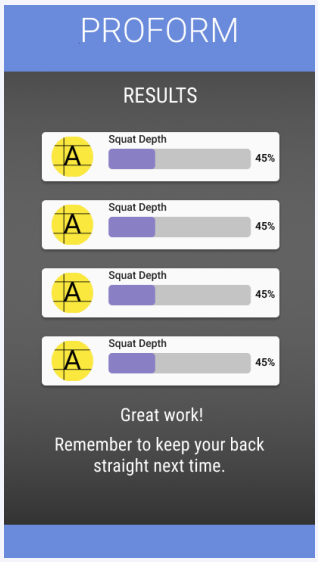
Developed in 36 hours at CitizenHacks, this **Angular** web app allows you to create data access law requests to send to any Canadian company. Per the **Personal Information Protection and Electronic Documents Act** (PIPEDA), every Canadian company is required to give you access to the information they keep on you, as well as give you the option to delete it, upon request.

However, would you know where to go to send this request? How to write it?

Sendable automates all of this in a simple **Angular web app**; it auto-generates email templates to send to the company of your choice, or if the company already provides you with options to access/delete your data, directs you straight to where it's done.







PROFORM

FLUTTER | TENSORFLOW

Developed in 36 hours at Hack the North, this **Flutter** app uses **Tensorflow's Posenet** to correct form in the user's squats and workouts. The user then records a short sample of them completing a workout, and after the video is analyzed via Posenet and compared with an ideal model, the results are published to the user via several progress bars.

keyon.io
linkedin.com/in/keyonjerome
github.com/keyonjerome
keyon.jerome@uwaterloo.ca