

# Corbyn Kwan

[corbynkwan.com](http://corbynkwan.com) | [corbynkwan@gmail.com](mailto:corbynkwan@gmail.com) | [github.com/corbynkwan](https://github.com/corbynkwan) | [linkedin.com/in/corbyn-kwan](https://linkedin.com/in/corbyn-kwan) |

## EDUCATION

**University of British Columbia**

*BSc in Computer Science 3.6GPA*

Vancouver, BC

*Sep. 2018 – May 2022*

## TECHNICAL SKILLS

**Languages:** JavaScript, HTML/CSS, Java, Ruby, C/C++, SQL

**Technologies:** React.js, React Native, AWS, GCP, Node.js, Ruby On Rails, Express.js, PostgreSQL, Mocha/Chai

## EXPERIENCE

**Software Engineer Intern**

Apr 2020 – Sep 2020

*Credit Hero*

*Hong Kong*

- Created a popup that uses **Microsoft Azure's Javascript Text to Speech SDK** to read contracts to the user, and a playback display showing the remaining time left.
- Implemented active jobs that send automated SMS messages to users using **Ruby On Rails**.
- Used **AWS** to store data, and used **Dataiku's Data Science Studio (DSS)** to track spending habits of users to train the model optimisation of the default rate.
- Reformatted **Javascript** code to **Ruby On Rails** thus switching client-side web page rendering to the server, improving loading speed of webpages by 50%.
- Fixed bugs associated with page responsiveness on mobile devices by reformatting **CSS** layouts using **Flexbox**.

## PROJECTS

**Scheduling Pickup System** | *React Native, React, Node.js, Express.js, PostgreSQL* | 🏆🏆

Jan 2021

- Won **1st Place in BizHacks 2021** out of 33 teams.
- Built a scheduling pickup system to reduce the wait time of BestBuy customers when they pickup from a shop.
- Created a mobile app that tracks the **geolocation** of a user, and sends an **email alert** when the user reaches pickup location using **React Native, Node.js, Express.js, Google Maps API**, and **SendGrid API**.
- Built a website where users can select a timeslot to pickup their goods from a BestBuy location, and created an algorithm that recommends timeslots with not as many people using **React**, and **Node.js**.
- Created a **centralized database** where Best Buy staff can view the amount of time left before a customer's scheduled pickup, and the products to prepare to give them using **PostgreSQL**.

**Logistics System** | *React Native, React, Node.js, Express.js, PostgreSQL, VBA* |

Dec 2020 - Present

- Built a logistics system to automate excel accounting spreadsheets and order receipts for a local company.
- Created a mobile app that sends purchase orders directly to the company's database using **React Native, Node.js**, and **PostgreSQL**
- Created a user admin dashboard that can edit purchase orders, and create order tickets using **React**

**Email Campaign System** | *AWS, React, Node.js, Express.js, PostgreSQL* |

Jan 2021 - Present

- Building an email campaign system for **HSBC** that allows developers to upload an email template (MS docx file) with dynamic values to **AWS**.
- Once the template is uploaded, the system create an **API end-point** which can be triggered by the project team, to allow the developers to create an email campaign from **upstream APIs**.

**Find Your Neighbourhood** | *React.js, Node.js, Express.js, Python, PostgreSQL, GCP* | 🌐🌐

January 2021

- Built a full stack web application that recommends a neighbourhood to live in Vancouver based on how much you value price, safety, restaurants, and transportation.
- Created a responsive front-end using **React.js**, and **Redux**
- Used **Python** to develop a **web-scraper** and **PDF parser** as well as connected to various **APIs** to gather data on price, safety, restaurant quality, and ease of transportation, and stored this data in a **PostgreSQL** database.
- Deployed using **Google Cloud Platform**, and **Heroku**

**University Grades Website** | *TypeScript, Node.js, HTML/CSS* | 🌐

Sep 2020 – Dec 2020

- Built a web application using **TypeScript** and **HTML/CSS** that parses datasets about course information in UBC, and returns a set of courses that meet the user's criteria (i.e average > 80%)
- Developed an automated test suite for the front-end and back-end using **Mocha**, **Chai**, and **Karma**.
- Created a **REST API** that can parse a query from the web application, add a dataset, and remove a dataset