

JACK PAN

COMPUTER SCIENCE STUDENT

✉ xj2pan@uwaterloo.ca

☎ (647) 685 1256

🌐 github.com/xcjackpan

🌐 xcjackpan.me

WORK EXPERIENCE

Software Developer, D2L

May – Aug 2019

- Improved NLP algorithm to generate questions from text, increasing rate of acceptable questions by **~60%**
- Led a technical team in the end-to-end design and development of a browser-based PDF annotator
- Implemented major frontend features using **React** and **Redux** in **Typescript**
- Built crucial backend functionalities for a **Mongo** user database with **Mongoose** using **Express** and **Node.js**

Software Developer, D.A. Consulting

May – Aug 2018

- Developed full-stack web app to manage permission entries for Perforce using **React**, **Node.js**, and **Java**
- Designed and built plugins in **Java** for Jira and Confluence as per client request

AWARDS

PennApps XX Award Winner, 2019

Best AR/VR Hack

Hack the 6ix Finalist, 2018

Top 10 of 65 teams

Term Dean's Honour List, Fall 2018, Winter 2019

>87% term average

EDUCATION

Candidate for Bachelor of Computer Science

University of Waterloo

Anticipated graduation in 2022

SKILLS

Languages:

Javascript, Typescript, C++, C, Java, Kotlin, Python, Ruby

Frameworks:

React, Redux, Express, Mongoose, Rails

Technologies:

Node.js, MongoDB, Docker, React Native, Git

PROJECTS

UWClassWatch

React, Node.js, Express, MongoDB, Mongoose

- Web app for UW students with **1500+ users** and highly positive feedback
- Emails users when a class they are "watching" has a spot open for enrolment

claire | Finalist at Hack the 6ix

React, Node.js, Express, Firebase

- Chatbot to reduce loneliness in seniors
- Analyses conversation for sentiment to display metrics on mental wellbeing

CharityChain

React, Firebase, Solidity

- Online platform for charities to receive and spend donation money
- Charity expenses are tracked in an Ethereum blockchain and analytics displayed to users

JAKE

C++

- NLP algorithm to extract key phrases from a body of text
- Modified version of RAKE algorithm implemented from scratch