**Zefeng (Daniel) Wang**

Availability: May – December 2021

7 Wycombe Way, Princeton Junction, NJ

(609) 454-1717 | [dnl.wang@gmail.com](mailto:dnl.wang@gmail.com)

[linkedin.com/in/zefeng-daniel-wang](http://linkedin.com/in/zefeng-daniel-wang)| [github.com/zefwang](https://github.com/zefwang)

**EDUCATION**

**Northeastern University**, Boston, MA May 2022

**Khoury College of Computer Sciences** GPA: 3.75/4.00

*Candidate for Bachelor of Science in Computer Science & Business Administration (Finance)*

*Minor in Mathematics*

**Relevant Courses:** Algorithms and Data, Object Oriented Design, Linear Algebra, Computer Systems, Database Design

**Honors and Awards:** Dean’s List, International Scholar, Top 10/Best Rookie Award at Hack Beanpot 2019

**COMPUTER KNOWLEDGE**

**Languages:** Java, JavaScript (NodeJS, React), Go, SQL, Python, C, HTML, CSS

**Tools/Libraries:** Docker, Firebase, Google Cloud (Firestore, Functions, Storage), JUnit, Kubernetes, Unix, NumPy, pandas

**Miscellaneous:** Git, GitHub, JSON, npm, RESTful APIs, XML, YAML

**EXPERIENCE**

**Software Developer Co-op** *@ Intuit May 2020 – Present*

* Developing in Java and Go to improve internal metrics on the Observability team for QuickBooks Online product
* Creating API in Go by writing a YAML parser, implementing CRUD functions, and configuring an LRU cache
* Used Micrometer and Hystrix (Java Spring) to create a filter for unnecessary data to improve readability of metrics

**Backend Web Engineer** @ *BusRight, Inc*  *November 2019 - Present*

* Improving web API by adding new features, enforcing test-driven development, and increasing efficiency
* Developed password reset process by adding functionality to get info from database & send emails via SendGrid
* Enforced persistent notifications by writing endpoints to get/send notifications to collections in Cloud Firestore
* Added Cloud Functions for daily database backup and profile picture functions triggered on user deletion/creation

**Full-Stack Developer** @ *Sandbox at Northeastern*  *September 2019 - Present*

* Maintaining and improving SearchNEU, a React/Node website that facilitates searching for classes at NEU
* Implementing changes to ensure data on site is accurate, durable, and accessible for over 50,000 unique users

**Core Team – Sponsorship** @ *HackBeanpot Inc*  *May 2019 – Present*

* Engaged in agile environment to organize annual hackathon for approximately 200 Boston students
* Devised sponsorship packet and worked with dozens of companies to raise over $20,000 in financial backing
* Led team of three in revamping judging system for students’ projects to be more holistic and efficient
* Designed and implemented algorithm for scheduling to resolve shortcomings with prior technology

**Mobile Application Development Intern** @ *DoubleBridge Technologies*  *July – August 2019*

* Researched and presented use of Android Studio for mobile development in Java to team of three
* Developed application that performs CRUD operations on MySQL database using REST API and PHP framework

**PROJECTS**

**SearchNEU** (NodeJS, React, Typescript, Cheerio, PostgreSQL) *September 2019 - Present*

* Moving data to relational database (PostgreSQL) by creating data models using Sequelize ORM
* Developed toggle and dropdown components in React to improve loading times ~30x and enable filtered search
* Rewrote scraper classes that produce info from colleges’ sites with Cheerio implementation of jQuery
* Refactoring JavaScript backend and enforcing reliability using Jest framework and snapshot testing

**Emojify** (Google Cloud API, Twilio API, Node.JS) *November 2019*

* Built program that translates text message from English into emojis using Twilio’s Programmable SMS
* Trained dataset that classifies content using custom categories with Google’s AutoML Natural Language API
* Developed using Node to integrate Twilio API and send responses based on result of Python script

**CrimePot** (Python, Flask) *February 2019*

* Implemented Google Maps API with HTML/CSS/JavaScript to place markers representing crimes on map
* Performed HTTP requests using Flask microframework to retrieve data from CKAN action API
* Filtered through JSON data based on user parameters and then converted to GeoJSON in Python