

# Justin Chiu

SOFTWARE ENGINEER · COMPUTER SCIENCE MAJOR · STONY BROOK UNIVERSITY

☎ (516) 312-1066 | ✉ [justin.chiu@stonybrook.edu](mailto:justin.chiu@stonybrook.edu) | 🏠 [jtchiu.github.io](https://jtchiu.github.io) | 📺 [jtchiu](#) | 📺 [jtchiu](#)

## Technical Skills

**Programming** NodeJS · Python · Java · C  
**Web** Express · Flask · VueJS · NuxtJS · Angular · HTML · CSS · JavaScript · Bootstrap  
**Other** AWS/GCP/OpenStack · Git · MongoDB · Cassandra · MySQL · Elasticsearch · Logstash · Kibana · NGINX

## Work Experience

### FactSet Research Systems Inc.

*New York, NY*

SOFTWARE ENGINEERING INTERN - RESEARCH ENGINEERING TEAM

*May, 2018 - PRESENT*

- Built a full-stack web application dashboard using NuxtJS/VueJS and a RESTful NodeJS back-end hosted on an inhouse Heroku.
- Primarily focused on data visualization of Redis and Memcached servers to display cache performance and development tools.
- Saved hundreds of debugging hours and promoted caching service that improves speed by 200% to clients through Product Developers.
- Worked alongside Senior Software Engineers providing constructive feedback on design changes and features.
- Responsibilities consisted of UI/UX design, technology architecture design, JIRA Agile development, and collaborating with other teams.

### KEMP Technologies

*Melville, NY*

R&D INTERN - RESEARCH TEAM

*Apr. 2017 - Aug. 2017*

- Worked with a team of senior software developers, system integrators, and quality assurance engineers on the development of a Slack Bot, which will serve as the centerpiece for customer engagement.
- Moved a hard-coded communication platform to an NLP platform that calculates the score for user actions and outputs the top scoring intent.
- Centralized logging, (using Elasticsearch, Logstash, Kibana), for maintaining load balancers and other server instances.
- Visualized database queries using ChartJS as an additional feature to the Slack Bot.

## Education

### Stony Brook University

*Stony Brook, NY*

B.S. IN COMPUTER SCIENCE

*Aug. 2015 - Exp. Dec. 2018*

Coursework CLOUD COMPUTING · ANALYSIS OF ALGORITHMS · SYSTEM FUNDAMENTALS II · PRINCIPLES OF DATABASE SYSTEMS · NATURAL LANGUAGE PROCESSING

## Projects

### Twitter Clone

NODEJS · MONGODB · CASSANDRA · NGINX · POSTFIX · ANGULAR · TYPESCRIPT

*May, 2018*

- Full-stack web application that utilizes a wide range of technologies to meet quality of service requirements (tail latencies < 300ms).
- Configured NGINX as a load balancer to handle about 1000 simulated requests per second routed to a NodeJS RESTful API.
- Total compute allowance: 20 vCPUs, 20 GB of RAM, hosted on a cloud service of my choice, AWS/GCP/OpenStack.
- Supported functionality: login/logout, email verification, adding text/media tweets, search, and following. Majority of the information is stored on a sharded MongoDB cluster, while Cassandra stores media. Used monitoring tools such as Kibana, PM2, NGINX Plus, and top.

### REST-based Tic-tac-toe

PYTHON · MONGODB · NGINX · HTML · CSS · JAVASCRIPT

*Mar. 2018*

- Full-stack web application using Flask as a RESTful API and AJAX calls to handle client-side requests.
- Hosted web servers on Google Cloud virtual machines, configured NGINX as a load balancer, and optimized MongoDB queries by indexing collections to meet performance requirements of 200 requests per second.
- Supported features, such as, login authentication, email verification, user sessions, and match histories.

### Food Beyond

PYTHON · MONGODB · HTML · CSS · JAVASCRIPT

*Feb. 2017*

- Organizes the user's ingredients and lists recipes corresponding to the items in the user's inventory.
- Using the Raspberry Pi, SimpleCV, and UPC-database, the user can scan barcodes and update the their ingredients.
- Designed and implemented the front-end using HTML, CSS, Bootstrap, and jQuery.
- Utilized the Food2Fork API using Python Flask as a REST API.

### Feature-based Extraction Summarizer

PYTHON

*May, 2017*

- Implemented a feature-based extractive summarizer using Dragomir R. Radev's LexRank text summarization algorithm.
- Evaluated inverse document frequency values and calculated accuracy on our extractor based on the CNN/Daily Mail dataset.
- Learned how to make a cosine similarity matrix stochastic, aperiodic, and irreducible to interpret it as a Markov Chain.