

# Matthew Jeng

2520 Hillegass Ave. Berkeley, CA 94704

mjeng@berkeley.edu

(973) 908-5890

<https://github.com/mjeng>

## EDUCATION

### University of California, Berkeley

B.S. Electrical Engineering and Computer Science

Expected Graduation: May 2020

GPA: **3.83 / 4.00**

Coursework: Algorithms, Databases (IP), Computer Security (IP), Artificial Intelligence, Computer Architecture, Data Structures

Organizations: CodeBase (software consulting), Theta Tau (engineering fraternity), Eta Kappa Nu / HKN (EECS honor society)

Leadership: Western Region Conference Organizer (Theta Tau), Industrial Relations Chair (Eta Kappa Nu)

### Online Coursework

TeamTreeHouse: Swift/iOS (50 hours), Ruby/Rails/Sinatra (30 hours), Spring/Maven (20 hours), Internet Architecture (6 hours)

Coursera: Bitcoin and Cryptocurrency Technologies (Princeton University), Machine Learning (Stanford University)

Udemy: Java Apache Spark

## TECHNICAL SKILLS/TOOLS

*Proficient*: Python, Java, C

*Familiar*: HTML/CSS, SQL, Swift, Matlab

*Learning*: Ruby

Git, Pandas/NumPy

Spark, Flask, Spring, Twilio, BeautifulSoup

Rails, Docker, Postgres

## WORK EXPERIENCE

### Full Stack Developer (React.js, Javascript, MySQL) – BetterCloud

August 2018 – present

- Integrating BetterCloud's platform for creating/executing IT workflows into ServiceNow, another IT management tool
- Enabling high throughput data ingestion by conforming to a serverless architecture using Google Cloud Platform
- Using ServiceNow's GlideRecord API to query/modify the database that contains BetterCloud IT workflows

### Application and Systems Engineer (Java, SQL) – Prudential

June 2018 – August 2018

- Developed Prudential's LINK application, a unified dashboard that brings together all of Prudential's financial services
- Used Apache Spark cluster computing to optimize a batch process that updates over a million user profiles at once
- Migrated project code that sends HTTP requests to updated service API endpoints, and wrote accompanying tests

### Machine Learning Engineer (Python/NumPy) – Polymorph

January 2018 – May 2018

- Worked with Polymorph ad-tech to build ML models around their unified ad auctions to help maximize their revenue
- Used several approaches including logistic regression with SGD, random forests, and field aware factorization
- Worked with over 200GB of user-ad interaction data on AWS to extract demographic trends and ad click through rate

### CS 61A Undergraduate Instructor (Python, Scheme, SQL) – UC Berkeley

June 2017 – May 2018

- Taught a lab and discussion of ~35 students (per semester) as an official staff member for a core Berkeley CS class
- Took part in weekly meetings to adjust the course to fit student needs and to improve it for future semesters
- Guest lectured 400 students on the structure of Scheme, one of the three major languages used in the class

## PERSONAL PROJECTS

### Daily Comic Service (Python, HTML/CSS) – <https://daily-xkcd.herokuapp.com>

February 2018 – June 2018

- Developed a full-stack application that sends comics scraped from <https://xkcd.com> daily to users through text
- Uses Twilio API to send texts, BeautifulSoup for webscraping, Google Sheets API (db), and hosted through Heroku
- Deployed an accompanying HTML/CSS website served by Flask that sends form data to the backend/database
- Scales well, with a database soft cap of 500,000 users and batched API requests to prevent rate limiting

### Jukebox Control System (Python, C++)

February 2017 – May 2017

- Constructed a Bluetooth/AUX enabled jukebox that could also play music from a customizable preloaded playlist
- Programmed the music functionality on the Raspberry Pi and patterns into the jukebox's lights using an Arduino
- Acted as the project lead, coming up with the overall design then compartmentalizing the work and assigning teams

### Bay Area Maps (Java)

March 2017 – April 2017

- Developed a small-scale version of Google Maps for the Bay Area, including zoom and route-mapping functionalities
- Parsed XML data files to retrieve location information used to autocomplete user input and tag specified locations
- Rendered map image tiles efficiently in response to user queries, minimizing lag for zoom in/out functions