Karan Kajla

◆ (909)-576-8198 ◆ kkajla12@ucla.edu ◆ kkajla12.github.io ◆ github.com/kkajla12 ◆

Education

- **University of California, Los Angeles –** *B.S. Computer Science*, June 2016
  - Cumulative GPA: 3.17/4.0, Upper-Division CS GPA: 3.46/4.0
  - Coursework: Algorithms, Data Structures, Operating Systems, Networks, Databases, Data Mining, Programming Languages, Scalable Internet Services, Artificial Intelligence, Web Applications, Software Engineering, Parallel and Distributed Computing

Skills

- **Languages:** Ruby, C++, C, Java, JavaScript, SQL
- **Technologies & Tools:** Rails, Node.js, React, MySQL, MongoDB, Git

Experience

Software Engineer – AppFolio, Inc. **August 2016 – Present**

Full-stack engineer developing and maintaining AppFolio Property Manager, AppFolio's primary SaaS product for commercial and residential Property Managers

- Implemented a highly requested data reporting feature for one of our largest customers using an internal parallel job scheduling framework.
- Added in-app emailing to key property maintenance pages, resulting in a 4x increase in emails sent and tracked by our system.
- Researched, designed, and developed a new property maintenance module for our platform using React and Ruby on Rails to allow customers to streamline management of their in-house maintenance teams. This new service was adopted by 5% of our customers in less than one month.
- Implemented a self-help chatbot using Twilio and Wit.ai for our annual Hack Day to aid customers in performing complicated tasks.
- Helped train new hire software engineers through code reviews and pair programming sessions.

Software Engineer Intern – DealerSync, Inc. **June 2015 – September 2015**

Full-stack engineer at an early stage startup, part of a lean 4 engineer team responsible for designing, developing, and maintaining DealerSync's core automotive dealership management platform

- Improved speed and memory efficiency of the email templating engine by re-implementing it using a more performance-oriented C# string replacement library which decreased overall CPU and memory usage and made the template engine more robust.
- Implemented highly requested features for the platform's financial modules, leading to increased online loan signing.
- Implemented Dependency Injection across key parts of the codebase, promoting consistency and maintainability.

Projects

HaaS: Helper as a Service **haasavocado.herokuapp.com**

HaaS is an intelligent personal assistant that aims to take care of complex tasks for users such as tracking and updating users on in-transit packages or suggesting meals to make given a list of ingredients on hand

- Integrated with Facebook Messenger to provide a rich and robust chat interface.
- Integrated Facebook's Wit.ai service to perform Natural Language Processing on messages sent by the user to understand the intent and context of each message.
- Architected and implemented a REST API using Node.js to manage users, service user requests derived from messages received through Messenger, and reply to those messages with the appropriate information.

Arpeggio: A Scalable Peer-to-Peer Musical Instrument Rental Marketplace **Web Application**

An online marketplace developed for a Scalable Internet Services course where users can rent out musical equipment and search for equipment to rent based on location and several other product filters (Airbnb for musical instruments)

- Worked in a team of four students using weekly sprints to incrementally build the application consisting of a JavaScript front-end, Ruby on Rails middle-tier, and MySQL back-end in 10 weeks.
- Implemented a cart-based shopping experience and a peer-to-peer payment system using Braintree.
- Scaled the application using strategies such as caching, database sharding, and query optimization then compared the effects of 'scaling-up' versus 'scaling-out' as the topic of a course paper.

Fit Bruin Calorie Counter **Web Application**

A web application that utilizes UCLA's dining hall nutrition fact information and allows users to create personal profiles which have a diet plan tailored to their needs and vital statistics so they can track their daily nutrition

- Wrote an HTML scraper to get nutrition information from UCLA's dining website, structured the data, and stored and queried it from a MySQL database.
- Implemented and hosted the backend using Python (Django), MySQL, Amazon Web Services EC-2 micro-instance, and Apache Server. Created an interactive, responsive, and easy-to-use user interface using HTML, CSS, and JavaScript.
- Fit Bruin was featured in an article in UCLA's campus newspaper, the Daily Bruin, and has over 6,000 visits and 11,000 page views.