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
  - CS 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 (APM), our primary SaaS product for commercial and residential Property Managers

- Work in a small team with a Product Manager, UX Designer, and other Software Engineers employing agile practices and close interaction with customers to discover and develop new product features
- Rapidly develop and deliver robust and performant web solutions using Ruby on Rails, MySQL, and JavaScript
- Quickly implemented a highly requested data reporting feature on an internal parallel job scheduling framework to prevent one of our larger customers from churning and to help pivot our product to align with the newest company goal of catering to larger customers
- Researched, designed, and developed an improved maintenance workflow for customers using React and Ruby on Rails to allow larger customers to manage their own in-house maintenance teams using AppFolio Property Manager
- Implemented a self-help chat bot in APM for our annual Hack Day using Twilio and Wit.ai to aid customers performing complicated tasks

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 the DealerSync web platform, from web clients to back-end and storage

- Worked on the entire application stack of the core Dealership Management Software (DMS) Platform: front-end (HTML, CSS, JavaScript), middle-tier (.NET MVC), and backend (SQL Server)
- Improved speed and memory efficiency of the email template engine by re-implementing it using a performance-oriented token replacement library, leading to a decrease in overall CPU time and memory spent performing email templating operations and creating a more robust template engine for client utilization
- Added key customer-requested features to the core platform's financial applications, leading to increased client site traffic and conversion rates and helping DealerSync acquire and keep new clients
- Implemented Dependency Injection across key pieces of the DealerSync codebase, promoting consistent and maintainable code

Projects

HaaS: Helper as a Service **www.haas.club**

HaaS is an intelligent personal assistant that aims to take care of complex tasks for users such as tracking and updating you 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
- 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 **www.fitbruin.com**

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

- Parses nutrition information from UCLA dining website, organizes data for easier use, inputs data into a MySQL database, and runs queries on information needed for webpages to display to users
- Implemented the backend using Python (Django) and a MySQL database running on an Amazon Web Services EC-2 micro-instance, served using Apache Server, and implemented the frontend in HTML, CSS, and JavaScript to provide an easy-to-use interface
- 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