Karan Kajla

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

Education
- **University of California, Los Angeles –** *B.S. Computer Science*, expected June 2016
  - Cumulative GPA: 3.1/4.0, Upper-Division CS GPA: 3.5/4.0
  - CS Coursework: Data Structures, Digital Logic Design, Linux/GNU Tools, Operating Systems, Algorithms, Data Mining, Networks, Computer Architecture, Digital Logic Design Lab, Systems Modeling, Programming Languages, Scalable Internet Services, Artificial Intelligence, Database Systems, Prototyping Programming Languages, Web Applications
- **Eleanor Roosevelt High School –** Graduated June 2012
  - Cumulative GPA: 4.59/5.00 (weighted) 3.89/4.00 (un-weighted) **|** Class Rank: 4/753

Skills
- **Languages:** C++, C, C#, Python, Java, JavaScript, Ruby, SQL
- **Web Technologies:** jQuery, HTML, CSS, Django, Node.js (Express), .NET, Rails, MySQL, MongoDB
- **Software/Tools:** MS Visual Studio, Eclipse, Android Studio, Atom, Git

Experience

Software Engineer Intern, DealerSync, Inc. (www.dealersync.com)   **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 several key client-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 the entire DealerSync platform, promoting consistent style and more maintainable code across the codebase and laying the groundwork for the implementation of shared caching (Redis) as the platform began to scale

Projects

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

A mobile-friendly online marketplace developed as a project 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
- Worked in a team of four students using AGILE methodologies (weekly sprints) to 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 widely used scaling strategies such as caching, database sharding, and query optimization and compared the effects of scaling-up versus scaling-out as the topic of a course paper

Hitch: The Conversation Starter   **Android Application**

A mobile application that serves as a forum environment where users can create a profile to exchange advice on how to continue a texting conversation in which they have reached a dead-end or cannot think of what further to say
- Wrote the backend API in JavaScript, using the Express 4 framework (Node.js), which serves data requests with JSON responses and stores posted user data and uploaded files/meta-data in a MongoDB instance on an Amazon Web Services EC-2 micro-instance
- Wrote and designed the Android application in Java and XML using the Android SDK (API 19/20) and Android Studio

Guitar Starter Tutorial Website   **www.theguitarstarter.com**

An informational website that explains the in-and-outs of several important pieces of guitar equipment (electric guitars, amplifiers, effects pedals, etc.) in an understandable format in order to give a valuable resource for guitarists to learn about integral guitar equipment before they invest money in it
- Designed the static website using the Express 4.x web framework (Node.js), HTML, CSS, and JavaScript so that it presented a fluid, responsive, and easy-to-use environment for users on all computing platforms.

Fit Bruin Calorie Counter   **www.fitbruin.com**

An easy-to-use, mobile-friendly 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, the Django web framework, and a MySQL database running on an Amazon Web Services EC-2 micro-instance managed using Apache Server
- Wrote the frontend in HTML, CSS (Twitter Bootstrap), and JavaScript (jQuery/AJAX) to provide a smooth user interface which is easier to use than manually navigating UCLA's provided nutrition information data
- Fit Bruin was featured in an article in UCLA's campus newspaper, the Daily Bruin, and currently has 6,000+ visits and 11,000+ page views