

SUSANNA SOUV

ssouv@berkeley.edu
GITHUB: susannasouv

Sacramento, CA 95842
(916) 844-6317

EDUCATION

Sophomore, University of California, Berkeley
Currently completing B.A. Computer Science

Expected graduation: Spring 2017

PROJECTS COMPLETED AND CONTRIBUTED TO

Listed below are non-academic projects; can be found on <http://challengepost.com/ssouv> and github.com/susannasouv. Please e-mail to see projects for coursework.

- **Wanderlust Tourguidr:** Android app with Flask server and MongoDB database. App captures image of landmark with device camera and is sent back to server to be matched with images in database (using OpenCV) and to retrieve metadata on landmark to return to user.
- **Twitter Reviews:** Utilized Flask and the Twitter API using Python wrappers (developed by github.com/tweepy) to create a web app that goes through the "tweets" of inputted username and outputs a positive or negative "rating".

EXTRA CURRICULARS

Lab assistant, Computer Science 61A Academic Intern Program.

- Assisted teaching assistants with class sections and office hours. Tasks included debugging students code, guiding students through homework and projects, and re-explaining course concepts. (Summer 2014, Fall 2014)

Junior mentor, Computer Science Mentors.

- Teaching small organized section to supplement and tutor students taking Computer Science 61A. (Spring 2015, in progress)

SKILLS

Languages (by recent use)	Python, L ^A T _E X, Java, C, Scheme; (in progress) HTML, CSS, Javascript
Operating Systems	Windows, UNIX/Linux; (in progress) OS X, Android, iOS
Software	Sublime Text, Vim, Github, Eclipse

RELEVANT COURSEWORK

University of California, Berkeley Bolded courses indicate completion; others are in progress.

- **Computer science 61A:** The Structure and Interpretation of Computer Programs: The course was an introduction to programming and computer science along with Python and Scheme syntax. Course included an introduction to object-oriented programming as well as declarative programming.
- **Computer science 61BL:** Data Structures and Programming Methodology: Fundamental dynamic data structures. Algorithms for sorting and searching. Introduction to the Java programming language. Elementary principles of software engineering such as test-driven development.
- **Computer science 61C:** Machine Structures: The internal organization and operation of digital computers. Machine architecture, support for high-level languages (logic, arithmetic, instruction sequencing) and operating systems (I/O, interrupts, memory management, process switching). Elements of computer logic design. Tradeoffs involved in fundamental architectural design decisions.
- Computer science 98: Snap! Below the Line: Snap! is the visual programming language used by CS10 at UC Berkeley and around the world. Students are taught the fundamentals of Javascript and Github, how to navigate, contribute to, and collaborate on a large code base such as the Snap! source code.
- Computer science 70: Discrete Mathematics and Probability Theory