# CS 430P/530: Internet, Web, and Cloud Systems
# Final Project

**Overview**

You (and your lab partner if working in pairs) will take the App Engine web application developed from your homework assignments and adapt it to be a front-end that utilizes APIs of your choice from those on Google Cloud (Vision, Text-to-Speech, Natural Language Processing, Speech-to-Text, Video Intelligence, Translate, Knowledge Graph, Firebase, Maps) to external APIs of your choice (e.g. Maps, Spotify, Nutritionix, Yelp, LiFX). Other API ideas can be found at https://rapidapi.com . For graduate students, a minimum of 2 API integrations per student must be implemented for a passing grade. For undergraduates, a minimum of 1 API integration per student must be implemented for a passing grade. Integrations will be graded on the level of functionality that has been leveraged from the API.

**Submission**

**Part 1: Code submission, URL posting (Tuesday of Finals Week @ 11:30pm)**

Code should be stored in your course's Bitbucket repository under a directory called `final` and submitted to D2L as a zipped folder. In addition to your code, include a text file (`final/url.txt`) that contains the URL that your application is running on. In order to grade your project, your instance must be left running in AppEngine until a week after finals week is over. If you need an additional coupon, request it now.

After submitting your code and URL to Bitbucket and D2L, visit http://cs410c-wuchang-201515.appspot.com/

**Part 2: Demo (Wednesday of Finals Week @ 12:30pm-2:20pm)**

During the final exam slot for the course, you will demo the functionality of your application via a 1-2 minute demo. Projects done in pairs can be done as a 2-4 minute demo. We'll be buying a round of Bubble Tea for our favorite sites.

**Part 3: Screencast (Saturday at end of Finals Week @ 11:30pm)**

Via a narrated screencast of no longer than 15 minutes per person, you and your partner will show the following. (Please do these in the exact order specified)

- Checking out the application source code from your final project Bitbucket repository
- Setting up the Google Cloud project to run your application (i.e. directions someone would need to run your project, pausing the screencast if necessary to wait for deployment)
- Demo the application running within your Google Cloud project that steps through all of the functionality your application supports
- Perform a source code walk-through that explains how each feature has been implemented. For partners working in a group, each partner should narrate the code they have written as credit will be assigned based on individual contributions.

- Walk-through the git history to show the time of initial commit and the total number of commits per-partner. Note, to get a particular author's commits, you can perform a:
    - `git log --author="wuchang@pdx.edu" | less`

Screencast software (CaptureSpace Lite) and submissions are to be done via PSU's Media Space (https://media.pdx.edu) at the following channel:
https://media.pdx.edu/upload/media/catid/145620232
Please ensure you are able to upload to this channel now.

**Grading rubric**

The project will be graded based upon the following rubric:

- Code and URL submitted to course Bitbucket repository as specified
- Application builds and runs on Google Cloud Platform
- Application functions and basic usage works as intended and does not crash
- Screencast submitted to MediaSpace channel
- Screencast adhering to the time limit with the walkthrough done as specified
- Quantity and quality of code that has been added
- Quality and creativity of APIs integration into application including its ability to accept and process user input (e.g. hard-coded usage of APIs is not recommended)
- Code clarity and cleanness (i.e. no unused remains)
- Code readability and modularity
- Code documented with docstrings and comments
- Quality of screencast and its walkthrough
- git repository activity (commits, commit messages, tags)

**Final Project Ideas**

- Reviews with translations for different languages
- Displaying nutrition information using Nutritionix API
- Analyze (short or parts of) videos with Google Video Intelligence API
- Recommend a shop based on a review
- Sentiment analysis of user reviews
- Vision API to detect locations or objects in images uploaded
- Speech to Text translation – Identify a song based on lyrics with the help of Knowledge Graph
- Text-to-speech application for the blind
- Animal Analysis – Enter an image of an animal into the Vision API and use the knowledge graph API to return information about it (like the giraffe from Slack lab)
- CaptionBot