Final Project

First off, let's take a second to congratulate you for making it this far! We know we've packed a lot of knowledge into a relatively short time! Kudos for rocking it!

Prompt

This is it, the moment you've been building for- time to build a Flask app that's all your own.

Got Ideas?

You are free to make a website or API about any topic you'd like as long as it meets all the requirements listed below. If you're having trouble coming up with a topic, here are a few to consider:

- Find a fun API and base it on that. Wrapping an API with added functionality is a valid use of Flask!
- Make an "About Me" site that's all about you.
- Make a portfolio website for one of your hobbies.
- Make a blog about a topic you find interesting.
- Make a tool you think would improve your life in some way (travel planning, financial advice, data aggregator, etc.)
- Feel free to share resources and inspiration with your classmates!

Deliverables

You must have a Flask site running locally on your own machine. You will work individually on this project, but feel free to share inspiration, resources, or cool APIs that you find with your classmates.

These requirements are **mandatory for graduation**. If you want to build something that doesn't

clearly meet these requirements, we can discuss this and make accomodations.

Requirements

Primary goal here is to learn how to use Flask and Python. Bare minimum requirements:

1. A working Flask application with at least 3 working routes.
2. Proper use of Python concepts where necessary (dictionaries/sets/tuples, classes, user or file I/O, etc.)
3. Data pulled in from at minimum 1 API.
4. Clearly commented code that any other coder could understand.

If you're building an API:

1. `GET` and `POST` routes.
2. Data stored either internally to the application (such as the `dog_api` we built with the list of Dogs), or, preferably, data stored in a CSV or `pickle` file.
3. A `README` file describing each of the routes and their usages. Include example API calls and responses copy-pasted from Postman.

If you're building a website:

1. 3 pages with HTML and CSS. These resources can be shared via templating, or unique for each page.
2. A `README` file describing the purpose of your website and description of decisions made while building it.

Bonus: Have extra time? Ask how to deploy a Flask site to a cloud service like Heroku! Or, try to follow the directions in this article.

Resources

Suggested Ways to Get Started

- **Begin with the end in mind.** Know where you want to go by planning ahead, so you don't waste time building things you don't need.
- **Read the docs** for whatever technologies or APIs you use. Most of the time there is a tutorial that you can follow! This isn't always the case, though learning to read documentation is crucial to your success as a developer.
- **Write pseudocode before you write actual code.** Thinking through the logic of something helps.

Useful Resources

- A List of Free APIs
- An Extremely Helpful Debugging Flowchart
- The Python Docs
- How to Use Keyword Args
- How to Use `*args` and `**kwargs`

- Using Chain and Other Itertools
- Python Sets Tutorial)
- Tuples Tutorial
- Writing a Great User Story
- Presenting Information Architecture (includes insight into wireframing)
- Heroku (platform for hosting your back end)

Evaluation

Your project will be evaluated based on the rubric below. **You must recieve a 2 or above to pass.**

Rubric

Score	Expectations
0	Incomplete.
1	Does not meet expectations.
2	Meets expectations — good job!
3	Exceeds expectations — fantastic!