SI 364 Fall 2017 | Final Project

Full credit points: 3000
Maximum possible points: 3100
For the final project, you will build another Flask application of your own!

To submit

Submit a link to a GitHub repository containing all of the requirements to the **Final Project** assignment on Canvas. We haven't provided a repository to fork this time — you should create your own.

That repository should contain all of the following requirements for full credit, including the README.md file with specifications of what your project does, anything that needs to be installed for it, and how to run it!

If you have any private data that needs to be included for the project that you do not want to put on GitHub, you can attach it in a file to a comment on your final project submission on Canvas. You should note in the README that it is necessary!

Requirements for the final project

Total 2500 points:

- Get some data from another source (an API, BeautifulSoup)
- At least 4 view functions (not counting error handling)
- At least 2 error handling view functions (404 and whatever other real error you want)
- At least 3 models (database tables)
- At least 1 one-to-many relationship
- At least 1 many-to-many relationship with association table
- At least 2 get_or_create functions to deal with entering data into a database
- At least 1 form using WTForms
- At least 2 dynamic links, which can be covered by
    - a href tags that send data that is processed by the end URL
    - using url_for
    - using redirect
- *(200 points of the 2500)* **Use at least one flask extension so that it works:**
    - Could be Flask email
    - Or using template extensions with base templates, using template inheritance (http://flask.pocoo.org/docs/0.12/patterns/templateinheritance/)

    - Or any other you find — in docs, online, etc (http://flask.pocoo.org/extensions/) — of equivalent complexity to email (e.g. writing only one line at the top of a Python file will not fulfill this requirement, but doing a little setup, like setting up email. to make something happen will)
    - The Bootstrap extension will *not* count for this, however.
- **IMPORTANT: Include a README.md file in the GitHub repository that explains what your application does and what is important to know about running it:**
    - What name database must be created
    - Any special command that must be run and how specifically to run the program, e.g. **tell us in the README what to run like:** python filename.py runserver …
    - Any dependencies of the project that we have to pip install — make sure to specify ALL of them clearly (you can do this in a requirements.txt file!)

.

- NOTE: **Do not repeat code** from lecture, section, or earlier assignments to fulfill requirements, though you may include it as an addition. If there is a specific reason you want to, you can explain it in your HW5 questions, and we will approve or not (if we do not, we'll offer comments about what changes you could make)!

Various amounts of points for:

- *(200 points)* User authentication
    - This has to make sense in the application — can't just be showing a name, has to involve data you can see or use while logged in that you otherwise can't.
- *(150 points)* Uploading a file (an image or text file) AND deploying your application to the internet
    - Note that you'll have to save the files to a database table to do this.
- *(200 points total)* Adding a (good) test suite — at least 8 good test methods (non-trivial tests), which can be run with a test command (there's a chapter in the Grinberg book about testing in Flask, and you can also apply previous course knowledge, if you want to do this)
- *(250 points)* Adding in AJAX requests that work, or using a translation module to translate into other languages from English (this is quite challenging — we haven't learned this in class! https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-xv-ajax)

NOTE:
If you do *all* of those things correctly, you will get more than 3000 points! The maximum amount of points you can get for the final project to count toward your score is 3100. Up till that point, we will count points for everything that is correct and works per the specifications above.

You are *not* being graded on anything involving design in this course. Though you may certainly use CSS or any styling you know if you plan to add this to your portfolio, it is neither expected nor a bonus in this class!

You can learn about design of the front end of the app in SI 339, and could of course study this and try it out on your own if you want!

Suggestions for coming up with an idea

- Make an app with social media data, like the Twitter app for HW4 — except with real Twitter data, and more relationships (rewriting the get_or_creates you wrote for HW4 will not count, but there are a lot of relationships you could build with Twitter: mentions, for example)
- Make an app representing language in Facebook posts (we have some instructions for getting data from Facebook in an easier way than you did in 106, if you took that course that we'll provide)
- Make an app representing data about different kinds of music (but way more than we've seen in the Songs examples so far, including that it could get data from elsewhere, e.g. an API)
- Anything else! What data do you want to play around with and display?