

Final Project

Due Dec 16 by 11:59pm **Points** 100

Final Projects

Your final project must be written using the Flask application framework. It can be a project based around almost anything as long as it meets the following criteria in terms of the use of web2py.

For a C

For a grade of a C you can create a blogging engine, by following the [Miguel Grinberg Flask Tutorial](http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world) (<http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world>)

- parts 1 -- 12 (you can skip 11 on email)
- This tutorial leads you step by step through building a microblog application using flask and associated libraries.

For A or B projects you can choose your own adventure. You will create a full stack web based application. The application does not have to be original, you can choose a part of an app that you use and like or you can do something of your own devising. The more real-world and useful it is the better. Applications that simply meet the requirements outlined below with no particular point to them will be docked a half a letter grade.

For a B

For a grade of a B your project must incorporate the following.

- You must have at least two controllers (Think python functions with `@app.route` decorators)
- You must have at least two views (Think templates)
- You must make use of some persistent data - something must be stored and accessed from the database. Using the DBAPI , SQLAlchemy or some other Python DB API. No more than one table is required.
- You must make use of one web api as part of your application. Or you may define your own API for your app to use on one or more of the pages.
- You must have some type of form processing so that your application is interactive.
- Your program must use the bootstrap templates, and you must make an attempt to make it look aesthetically nice.

For an A

To earn an A on the project you will need to go further.

- You must have at least three controllers (Think python functions with @app.route decorators)
- You must have at least three views (Think templates)
- Your app should have user session management. (anything to do with session management does not count for the above controller/view requirement or the below database requirements) I recommend you look at the Flask-Login plugin.
- You must use records or SQLAlchemy and create a database with 3-4 tables. These tables should incorporate a one to many relationship in some way, or they may have a many to many relationship. These tables do not include any tables that you create for user management.
- You should incorporate the WTForms module into your application for processing form data. -- You can teach yourself this from the tutorial.
- You must make use of at least 1 web API for some aspect of the project that you use in your web pages. Or you may define your own API for your app to use on one or more of the pages.
- Your program must use the bootstrap templates, and you must make an attempt to make it look aesthetically nice.

You must provide a README that describes to the TAs how they should access and use your application. This will aid them in grading.

The application you choose does not have to be an original idea, you may choose to implement your own version of some app on your phone or a web app that you have used and simply want to understand better by implementing yourself.

You may work in teams of 2 on this project, unless you are doing the blogging application for a C then you should work alone.

for the B and A grades the points assigned will be in the middle of the range some points will be assigned for usability and for the look and feel of the program.

Groups of 3

A couple of groups have asked if they can do a group of 3, and I will allow that. If that is the case you should plan to have an additional database table and an additional page and controller as a part of your application.

Draft : 11:30 2019-11-20