Final Assignment – ENGI 1006

Due: May 15th, 2020

Guidelines:

This homework will be submitted using a Git repository. The Git repo should contain a folder called uni-final, all lowercase. A link to the Git repo should be submitted to Courseworks.

Flask

Up to this point, we've been primarily focusing on utilizing Python to manipulate data using various classes and libraries. Python is also useful for a variety of other applications, including creating a back end for web applications. A backend is essentially a series of routes that direct both users and data to and from various pages on a website. We can create such a backend using *Flask*, a Python microframework.

A basic Flask server has the ability to "serve" different webpages, depending on what a user has requested. This is also often referred to as the *client-server model* where a client (i.e. a user), requests information that is displayed by a server. In this assignment, we'll be creating a basic server with Flask that will allow a user to traverse through the pages of a simple website.

Getting Started

Although web applications are usually developed with IDEs such as Visual Studios, we will be using Spyder to remain consistent with the rest of the course. The skeleton code contains a folder called "final". **Rename this folder to uni-final.** Open the app.py file provided in the skeleton code. Read the comments in the code to make sure you get what's happening. More information about Flask will also be provided in recitation. Upon running the app.py file, you should get a message similar to this:

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

This indicates that the server is started on your local host. If you go to your favorite browser and enter the URL http://127.0.0.1:5000/ you should see "Hello World!" displayed.

HTML Files

The “Hello World!” displayed in the previous step was very basic HTML, but we can create more sophisticated webpages. Inside the “templates” folder of the skeleton code, there is a file called index.html. Replace the line ‘return “Hello World”’ with the line: ‘return render_template(“index.html”)’.

Flask, upon seeing this line, will search in the “templates” directory for the index.html file and render this file. Restart the app by running app.py again and notice that there is no difference! Modify the HTML file a bit to prove to yourself that Flask is indeed rendering the HTML file. **Note: you may have to press CTRL+C in the IPython console to stop the Flask app and then restart it, after saving index.html, to see the changes reflected on the webpage.**

As an exercise (not to be graded) try following the format of the “/” (home) route, and create a “/1006” route, that loads an HTML file which displays the words: “1006 homepage”. You can access this page by running the Flask app and going to the URL http://127.0.0.1:5000/1006.

Git Repository

At this point, you should initialize a Git Repository. More information will be provided about Git during recitation. Add all current files and folders to the Git repo and commit with the message “First commit.” As you work on this assignment, you should periodically make commits with relevant commit messages. For instance, after finishing the main page, I might make a commit with the message “Finished main page.” **At least 3 git commits (in addition to the first commit) must be made throughout the course of this assignment to show your progress with the website.**

The Assignment

Your goal in this assignment is to create a basic personal 1006 homepage using Flask. The webpage should have the following features:

1. A main page at the "/" route (the home route). This should include

- Your name

- A picture of yourself

- Your current grade

-Your major or intended major

- A link to **at least 2 other pages on your website (**these can optionally be an assignments and classes page – see info below!)

- A link that takes users to **a favorite (appropriate) website of yours**

Hint: You can create hyperlinks using Flask routes and/or the <a> href attribute.

OPTIONAL (Recommendations for the other 2 pages on your website – follow these optional guidelines or think of something unique for your other 2 pages; it is your choice!**):**

2. An assignments page. This could include

- A brief (1-2 sentence) summary for each homework assigned in 1006

- A few sentences explaining what your favorite assignment this semester was and why

3. A classes page. This could include

- A list (or an HTML table) of all the classes you're taking/have taken this semester

- A few sentences explaining what your favorite class taken so far has been

Note: The website does not have to be styled/designed at all. A few bullet points/plain text are fine and we will not be grading for style. However, if you want to add CSS/JavaScript etc. feel free to do so. You will need to create a "static" folder alongside the "templates" folder to store these files.