Homework 7: Deployment

Due date: March 5, 2020 at 11:59pm

This homework is the fourth (and final) in a series of homeworks in which you will build an increasingly sophisticated nano-blogging site.

For this assignment, you will deploy your application to the cloud.

The learning goals for this assignment are to:

- Demonstrate mastery of the learning goals from the previous homeworks, including both technical features of web applications and the development process.
- Gain familiarity with a cloud deployment platform.
- Gain basic familiarity with database systems administration.

Specification

This section describes the enhancements you will make to your social network application. To begin your assignment, recall that you should start by copying over your project from the last homework over to the hw7 folder (see Turning in your work for the directory structure).

- Deploy the social network application on either AWS EC2 or Heroku. If you wish to deploy to another cloud-based platform, you must get permission from the course instructor.

Requirements

Your application must also follow these requirements:

- You must meet all requirements specified in the previous assignments.
- You must store your data in a relational database that is not SQLite.
- If you deploy on Heroku, you must store your images in Amazon S3.
- If you deploy on AWS, you may not use the development server. (You must deploy with Apache HTTP Server or something similar.)

- You may not store passwords or access keys in your application code (or in your repos).

- Cite all external resources used and any additional notes you would like to convey to your grader in the `README.md` file.

Assignment hints

1. If you are deploying on EC2, the recommended method is to run Django under Apache HTTP Server. Django documentation is:
    - https://docs.djangoproject.com/en/3.0/howto/deployment/wsgi/modwsgi/

2. If you are deploying on Heroku, the tutorial is:
    - https://devcenter.heroku.com/articles/getting-started-with-python

3. Data files (pictures) created in Heroku will disappear when your server becomes inactive. To store your pictures permanently, you should put your images AWS S3. The Heroku guide is here:
    - https://devcenter.heroku.com/articles/s3-upload-python

4. You should not include passwords and other access keys in your application code and you should not put them in your GitHub (or Heroku) repositories.
    - On EC2, put your access keys in files that will not be stored on GitHub. Be sure such files are listed in your `.gitignore` file.
    - On Heroku, set environment variables for passwords and access keys (as is shown in the S3 documentation).
    - You may commit, to your repo, files that describe the structure of files that contain access keys, as was shown in the course example with `config.ini.sample`.
    - You do not have to put the `SECRET_KEY` in `settings.py` into a separate file (though you may).

5. You may leave `DEBUG=True` in `settings.py`
    - It's easier for us to find problems this way. (We're not deploying for commercial use.)

Specification fulfillment

Your submission must follow all specifications and requirements introduced in the previous sections.

Turning in your work

You must provide the URL for your deployed application in the `hw7/README.md` file in your GitHub student repository. Specifically, in your `hw7/README.md` file you must have the following line:

```
The URL to access my homework solution is http…
```

(Replace the `http…` part with the URL where your solution is deployed.)

If your deployment code is not being developed in your student repo (e.g., because you're using a Heroku repo), you must copy your code into your student repo and then submit it. The directory structure will look somewhat like this:

```
[YOUR-ANDREW-ID]/hw7/
    |-- README.md
    |-- config.ini.sample
    |-- webapps/
        |-- settings.py
        |-- urls.py
    |-- socialnetwork/
        |-- static/
        |-- templates/
        |-- forms.py
        |-- models.py
        |-- views.py
    |-- manage.py
```