INFO3180 Lab 2 (20 marks)
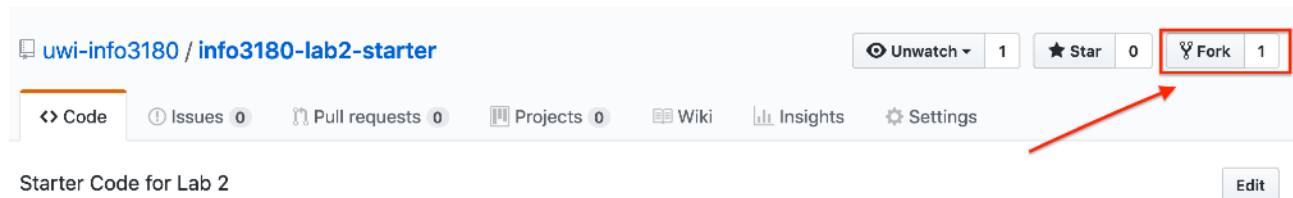
Due Date: Sunday, February 14, 2021 at 11:55pm
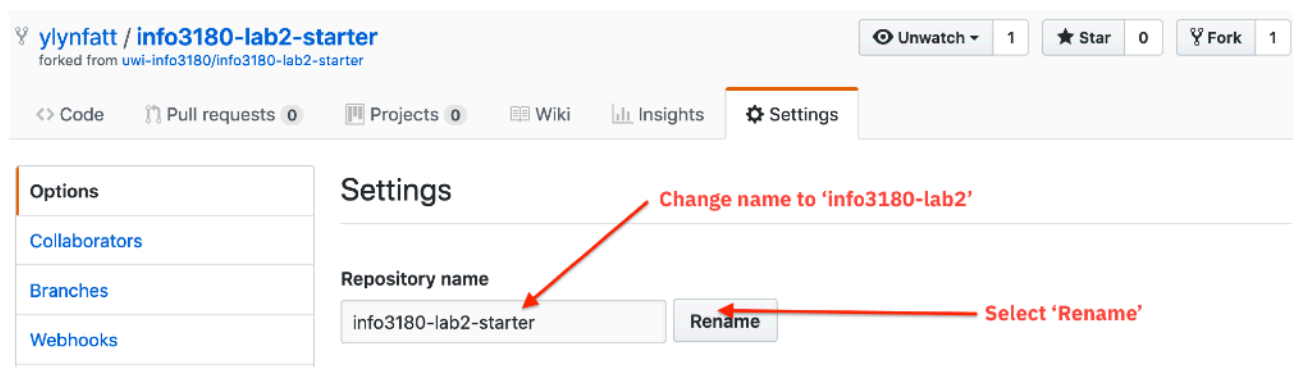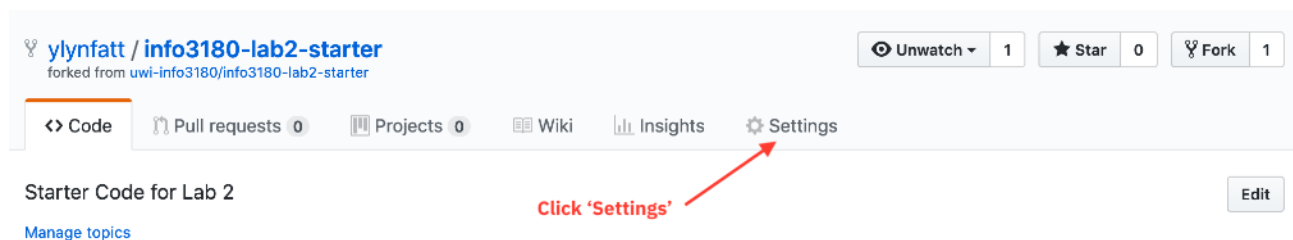
Initial Setup

Start with the starter code at: https://github.com/uwi-info3180/info3180-lab2-starter

Fork the repository to your own account

In github.com rename it by going to settings and changing the name to info3180-lab2

To start working on your code clone it from your newly forked repository for example:

```
git clone https://github.com/{yourusername}/info3180-lab2
```

Note: Ensure you change **{yourusername}** to your actual Github username.

Exercise 1 - Creating a new route in Flask

For this part of the lab, you will create a route in Flask that displays your "fake" profile information (See Figure 2). In a future lab/project this will need to be a database driven application but for now we just want to have a placeholder.
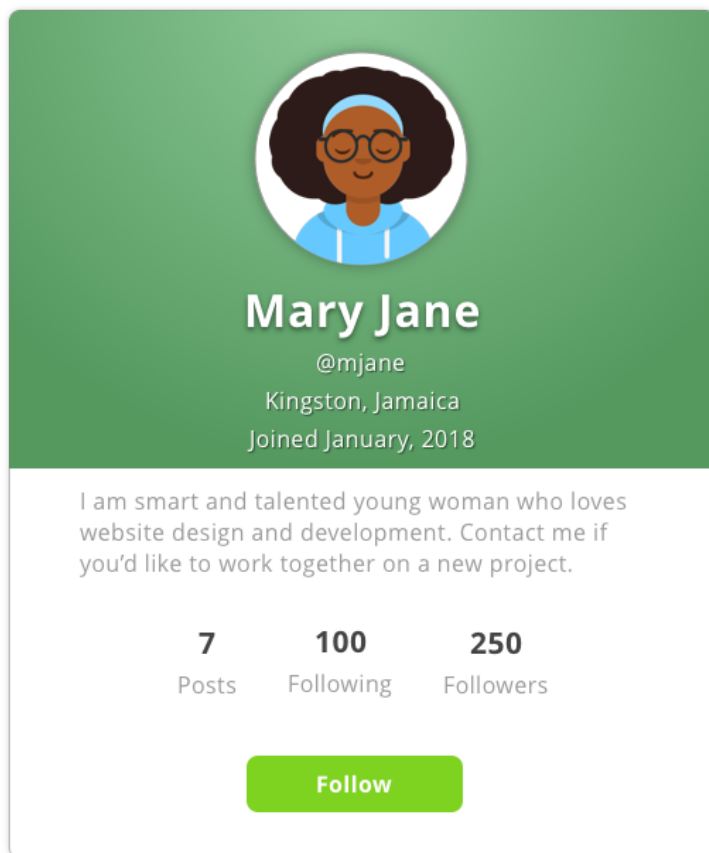

FIGURE 2: SCREENSHOT OF WHAT YOUR PROFILE PAGE SHOULD LOOK LIKE

Steps you'll need to take:

1. Create a new view function and route in your **app/views.py** file. Call your route "**/profile**" and your view function "**profile()**"

2. Create a new template in your **app/templates** directory, call it "**profile.html**" and customize it to look like the screenshot above (See Figure 2).

   - Your profile should have *a **photo***, *your **full name***, **username**, **location**, the **date you joined**, a **short bio** and the **number of posts**, **number of followers, number persons you are following** *and a **Button which says "Follow"***.

   - Use your own colours, image and information, but ensure you save your image in your '**app/static**' folder and reference it in your template using the **url_for()** Flask method. Also ensure that your template *extends* the **base.html** template and that your HTML code is put within the main block ie. **{% block main %} {% endblock %}**. You can add your CSS styles to the **app.css** file in the **static/css** directory.

3. Add another navigation link to your Navbar by editing the **header.html** template file. The link should be called '**Profile**' and should link to your **profile** view function using the **url_for()** Flask method.

4. Create a separate function (not a view function but instead a regular python function) in your **views.py** file called "**format_date_joined**" which when given a date as an argument returns the date formatted as **Month, Year (e.g. Feb, 2021)**.

5. Use the **format_date_joined()** function from the previous step in your **profile** view function to pass a specific date to your template

as the date you created your profile page and output it using the Jinja2 **`{{ variablename }}`** template syntax.

Hint:

Get the date using the Python **datetime** library (NOT Javascript) (this is needed in the fake profile) you might find this code helpful (For more information read: http://www.cyberciti.biz/faq/howto-get-current-date-time-in-python/)

```python
import datetime


now = datetime.datetime.now() # today's date
date_joined = datetime.date(2019, 2, 7) # a specific date
## Format the date to return only month and year date
print "Joined "  + date_joined.strftime("%B, %Y")
```

Testing the app

You'll need to activate your virtual environment and install the requirements then use '**python run.py**'.

```
$ python -m venv venv (you may need to use python3 instead)
$ source venv/bin/activate (or .\venv\Scripts\activate on Windows)
$ pip install -r requirements.txt
$ python run.py
```

You will see output similar to this

```
(venv)ylynfatt:~/workspace (master) $ python run.py
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 339-249-416
```

You can open your web browser and visit:

http://localhost:8080

Commit your code to your Github repository

Ensure that you add, commit and push your code (including any images used) to your Github repository.

```
git add .
git commit -m 'your commit message'
git push origin master
```

Submission

Submit your code via the "Lab 2 Submission" link on OurVLE. You are NOT required to push your code to Heroku. You should submit the following link:

1. Your Github repository URL for your Flask Exercise e.g. https://github.com/{yourusername}/info3180-lab2

Grading

1. Created route for Profile and its associated view function (2 marks)
2. Created Profile template file in the appropriate directory and render a template that extended the base template. (3 marks)

3. The Profile page should look similar to sample profile page seen in Figure 2 and should have all the appropriate information. (5 marks)

4. The image used on the profile page should be placed in the proper directory and referenced with the appropriate Flask function. (3 marks)

5. A separate function should be created to format the date the user joined. The format should be "Month, Year" (for example Feb, 2019). (2 marks)

6. The date should be passed to the template and printed using the appropriate Jinja2 template language syntax. (2 mark)

7. A link to the Profile page should be added to the Navbar. (1 mark)

8. The Flask app server should start without any errors and we should be able to view the profile page in a web browser. (2 marks)