INFO3180 - Lab 4

File Uploads

Due:

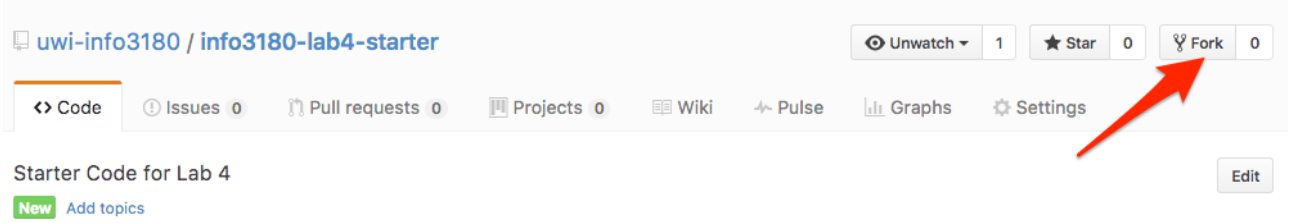
Note:

Because all Python and all the relevant development tools are already installed on C9.io, it is recommended that you use C9.io for this lab, you will need to sign up for an account. At a later date you can setup a similar environment on your computer

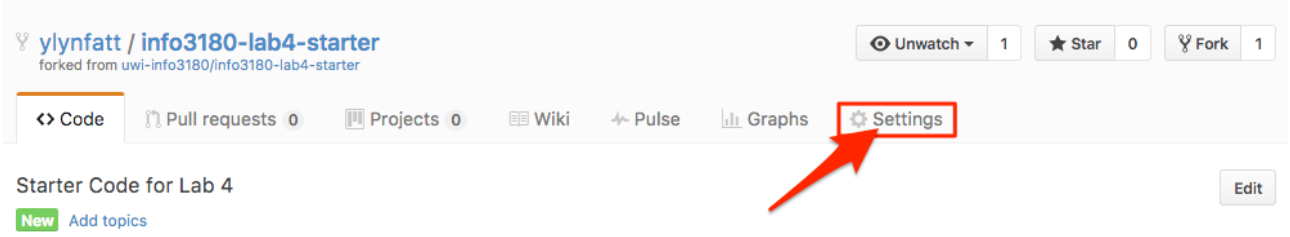
Exercise 1 - File Uploads

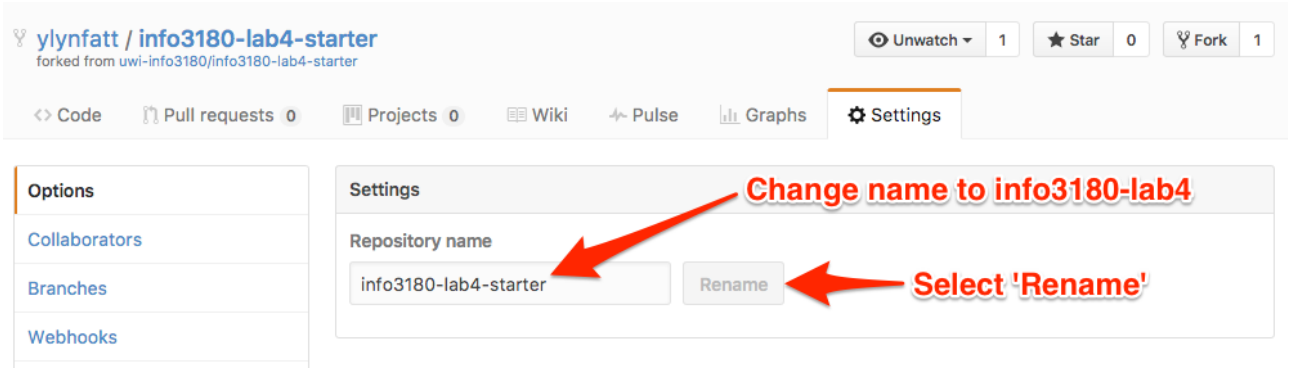
Start with the example at: https://github.com/uwi-info3180/info3180-lab4-starter

Fork the repository to your own account



In github.com rename it by going to settings and changing the name to info3180-lab4





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

If you are working in Cloud9 then create a new workspace and paste the repository URL (e.g. https://github.com/{yourusername}/info3180-lab4) in the"Clone from Git or Mecurial URL" field on the workspace creation page as you have done in previous labs.

Or if you are working locally on your own computer (ie. NOT in Cloud9) :

```
git clone https://github.com/{yourusername}/info3180-lab4
```

Step 1: Setting up your upload folder

Note: Ensure you create your virtual environment using **`virtualenv`** as you have done in previous labs and activate it. Also ensure you install the necessary libraries for the application from your **`requirements.txt`** file by using the **`pip install`** command.

1. Add a configuration option for your **`UPLOAD_FOLDER`** to point to your "**`./app/static/uploads`**" folder in your **`__init__.py`** .
2. Alter the **`add_file`** method so that the **`file_folder`** variable points to your upload folder configuration option.
3. Test that your file uploads work by starting the development server for your Flask app, log in and go to the **`/add-file`** route. Upload a file and then check your uploads directory to see if it is there.

Note: The username and password to use to login for this exercise is in your **`__init__.py`** file as configuration options.

4. Commit your code to your repository.

Step 2: Listing your uploaded files

Here's an example python script for iterating over some files in a specific directory

```
import os
rootdir = os.getcwd()
print rootdir
for subdir, dirs, files in os.walk(rootdir + '/some/folder'):
    for file in files:
        print os.path.join(subdir, file)
```

1. Use the code from the example above to help you create a method in your app which iterates over the contents of the **`static/uploads`** folder. **Hint:** Yes, you will need to make some modifications to the code above.
2. Create a route called "**`/filelisting`**" and render a template which lists the files uploaded in the **`static/uploads`** folder as an HTML list (ie. using an unordered or ordered list)
3. Commit your code.
4. Ensure that the **`/filelisting`** route is only accessible to users who are logged in. **Hint:** Take a look at the **`add_file`** method.
5. Commit your code.
6. Update the route called "**`/filelisting`**" which displays uploaded **`jpgs`** but lists the other files by name only (Bonus)

Deploy to Heroku and Submit

Step 1 - If you didn't do so before, add, commit and push your code to your Github repository

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

Step 2 - Deploy the application to Heroku

Now deploy your application to Heroku. If you haven't already done so, ensure that you sign up for an account on the Heroku website (https://heroku.com) and then do the following. **Note:** You will also need the Heroku CLI. This should already be installed on Cloud9. If you are instead working on your local machine, then you must install the CLI tool. See instructions at https://devcenter.heroku.com/articles/heroku-cli .

```
heroku login
heroku apps:create
git push heroku master
```

Submission

Submit your code via the "Lab 4 Submission" link on OurVLE. You should submit the following links:

1. Your Github repository URL for your Flask app e.g. https://github.com/{yourusername}/info3180-lab4
7. Your URL for your Heroku app e.g. https://{yourappname}.herokuapp.com