Cardiff University
School of Computer Science & Informatics

Optional Independent Further Work

Flask 5:
Admin panel

Objectives

The objective of these exercises is to provide an extension to the ***main*** Flask labs 1 - 4, by demonstrating how to:

- incorporate a basic administration panel to provide site management tools.

General Comments

- **NB:** As for all labs, you should do all the work with your virtual environment **activated**.
- This exercise is optional and is independent further work.
- A repository with possible solutions to this exercise is found in the `code` dir of this repository.
- Review the **RELIMINARIES** section in Flask 1 instructions - these are relevant to this lab too.

Useful addresses and resources	
Code for this exercise is available at:	https://git.cardiff.ac.uk/scmne/flask-labs
Flask-Admin documentation: Flask-Admin Getting Started	https://flask-admin.readthedocs.io/en/latest/ https://flask-admin.readthedocs.io/en/latest/introduction/#getting-started
Also see the 'Useful Resources' section in Flask 1 lab.	

ADMIN PANEL

An administration panel will provide a GUI that allows novice administrators to manage the website through a simple interface. We will be using **Flask-Admin** [1] to implement the administration panel, so you need to add it to your project dependencies.

1. To implement basic database manipulation functionality, we can add the following code to the end of our **__init__.py** file:

```
...
from flask_admin import Admin
from flask_admin.contrib.sqla import ModelView
from blog.models import User, Post
admin = Admin(app,name='Admin panel',template_mode='bootstrap3')
admin.add_view(ModelView(User, db.session))
admin.add_view(ModelView(Post, db.session))
```

   Visit `http://127.0.0.1:5000/admin`, and you should be able to see the home page of the Admin backend site:

2. At the moment, anybody can see the Admin Panel without needing to log in.

   We can improve the security of this by making it so that only users with administrative rights can access these pages.

   (a) First, we need to update our **User()** class in **models.py** by adding the following field to identify whether a user is indeed an administrator:

```
...
is_admin = db.Column(db.Boolean, nullable=False, default=False)
...
```

   (b) We must update the **user** table in our db to reflect this change, i.e. we need to add **is_admin** column (attribute) to the table, and then we must tell the system that a particular user is an **admin** user.

   This can be done by changing **is_admin** value in that user's record from to **0** to **1**, using a db GUI (e.g. phpMyAdmin or MySQL Workbench). Alternatively, you can use the following MySQL commands:

```
UPDATE user
SET is_admin = 1
WHERE username = 'janedoe';
```

(1) https://flask-admin.readthedocs.io/en/latest/

(c) We need to extend the functionality of the built-in `ModelView` by modifying the function `is_accessible()`. To do this we create a new file called `views.py` containing the following code:

```
from flask_admin.contrib.sqla import ModelView
import flask_login as login
from blog.models import User

class AdminView(ModelView):
  def is_accessible(self):
    if login.current_user.is_authenticated:
      if login.current_user.get_id():
        user = User.query.get(login.current_user.get_id())
        return user.is_admin
    return False
```

This code checks that the user is logged in and then verifies that the `is_admin` field reads a `True`.

(d) We must now modify our code in `__init__.py` file to use this new class:

```
...
from flask_admin import Admin
from blog.views import AdminView
from blog.models import User, Post
admin = Admin(app, name='Admin panel', template_mode='bootstrap3')
admin.add_view(AdminView(User, db.session))
admin.add_view(AdminView(Post, db.session))
```

(e) Finally, we can improve the '*feedback*' of the site by incorporating some logic into the admin home screen. Create a file in the following directory of your project `templates/admin/index.html`:

```
{% extends 'admin/master.html' %}
{% block body %}
{% if current_user.is_admin %}
  Welcome to the admin panel {{ current_user.username }}!
{% else %}
  Hello, please login as an admin before accessing this panel  |
  <a href="{{ url_for('login') }}">Login </a>
{% endif %}
{% endblock %}
```

[NB] This is of course a very simple/basic implementation of Admin Panel, and requires further work to make it more sophisticated.
