

Django Blog Project

Part 6 – Users and Registration

In this lab, you will be adding a way to interact with you users in your blog project. The end result will be the following:

- Users can log in and log out
- Users can add a comment only if they are logged in (the author of the comment will be filled in automatically based on which user made the comment)
- Users can edit only their own comments

Part 1: Set up user login system

1. `cd` to your project directory (the one that `manage.py` is in) and make a new app called `reg`:

```
django-admin.py startapp reg
```

This will make a new folder called `reg` with some files already created for you.

2. Edit your main `urls.py`. After the part that redirects all `blog/` urls to the `blog/urls.py` file, make a similar line that redirects all `reg/` urls to the `reg/urls.py` file
3. `cd` into the `reg` directory
4. The models file doesn't need to be edited because django makes a user model by default
5. Now edit the `views.py` file and add the code from “instructions/part_6_instructor_code/reg_view.py”. Read through it.
6. In `reg/urls.py`, add URL rules so that ‘`reg/login`’ will route to the `do_logout` function and ‘`reg/logout`’ will route to the `do_login` function.

7. Implement the code in the `do_login` where it says `#YOUR CODE HERE`
 - This `if` statement will be executed when you press the submit button to log in.
 - Check if the username and password are correct, and if they are, log in the user and refresh the page (i.e. redirect to the current page).
 - The submitted username is `request.POST['username']`. (Reminder: `request.POST` is a dictionary)
 - You can check if a user is authenticated using `authenticate(username , password)`. If the return value is `None`, then authentication failed. Otherwise, it worked.
 - You can log the user in by doing `login(request , user)`, where `user` is the returned value of the `authenticate` function.

8. Now that your view is done, you can now edit the templates.

```
cd to your templates directory
mkdir reg
cd reg
```

9. Create the template that is rendered by the `do_login` view function and add the HTML and template code found in `"instructions/part_6_instructor_code/login_template.html"`.

The code provided will show the user the login form every time. Instead, change this so if the user is *already* logged in, they will be shown a "you are already logged in" message, and a link to log out.

Hint: look at the context you are passing to this view.

10. Create the template that is rendered by the `do_logout` view function, and add the HTML and template code found in `"instructions/part_6_instructor_code/logout_template.html"`
11. Test to see if it's working. First, go to your admin page and add a few users. Then, go to `localhost:8000/reg/login`, and make sure you can log in and log out.

Part 2: Integrate users into your blog:

12. Open up your blog `views.py`

13. You need to make three changes to the `post_detail` function:

- Exclude the 'author' field from the comment form.
- Make sure the author is still put into the comment before it is saved.
Hint: the author is the user's username. You need to find out how to get this.
- Pass the `'request.user'` object to the template by adding it to the context with the name 'user'. This will be used for making decisions in `post_detail.html`

14. Edit the view function that is called to edit a comment so that it returns a `HttpResponseForbidden` object with the message "you do not have permission to edit this comment" if the current user is not allowed to edit the comment that they are trying to edit. Only the author is allowed to edit his/her own comment.

15. Finally, edit the `post_detail.html` template so that only logged-in users can see the form to add a comment, and so that the "Edit Comment" link only appears on comments that the currently logged in user is authorized to edit.

16. Check that this works properly when you are logged in or logged out! You have made an awesome website using django!

PUSH TO GITHUB AND HEROKU.

TO BE IN THE CONTEST,
READ THE RULES.
(in "instructions/contest.pdf" if its not there, pull)