Lab 9 – Django Authentication and Session

Marks = 2

Submission =

- **On Brightspace. Submit views.py, urls.py and any templates you create in this lab or update it from the previous lab(s).**

Part 1: Work with login and logout

1. Implement a login page that authenticates users based on their credentials (username and password). In views.py, create view functions for user *login* and *logout*. Here is an initial version of the functions:

```
# Import necessary classes and models

from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required


# Create your views here.
def user_login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(username=username, password=password)
        if user:
            if user.is_active:
                login(request, user)
                return HttpResponseRedirect(reverse('myappF23:index'))
            else:
                return HttpResponse('Your account is disabled.')
        else:
            return HttpResponse('Invalid login details.')
    else:
        return render(request, 'myappF23/login.html')
```

```
@login_required
def user_logout(request):
    logout(request)
    return HttpResponseRedirect(reverse(('myappF23:index')))
```

2. Create *login.html* in the templates dir. It should ask for a username and password and pass the information to the **`user_login()`** view function given above.
3. Update *myappF23/urls.py* to include patterns for **`login()`** and **`logout()`** views.
4. Define a view function **`myaccount(request)`** in your *views.py* file. The user must be logged in to access this function. For a logged in user:

   a. If the user is a *Student*, return the following:

   - The first and last name of the student.
   - All the courses that have been ordered by the student.
   - All the courses the student is interested in.

   b. If the user is not a student, display a message: 'You are not a registered student!'.
5. Create *myaccount.html* and pass the information: full name, courses ordered, and courses interested in, as a context to be displayed in the template.
6. Update *myappF23/urls.py* with a suitable pattern for **`myaccount()`** view.

Part 2: work with sessions and cookies

1. Check cookies in user's bowser:
   a. Set a test cookie in some view(s) of your choice. Check if the test cookie worked in the subsequent view.
   b. After you verify the test cookie worked, i.e., the client's browser accepts cookies, delete that cookie. Its up to you to do that in the same view or in different views.

2. Set cookie in *about* view:

   a. In *index* view, check for a **cookie** 'user_visits' that indicates the number of times the about page has been visited so far.

b. If the cookie exists retrieve the value and increment by 1.

c. Store the updated information in a cookie ('user_visits') and set it to expire after 5 minutes.

d. Pass this information (i.e. number of visits) to the template. Update *index.html* template to show how many times the page has been visited.

e. Use '*Developer Tools*' of your browser (Chrome) to view your cookies.

3. Use session framework.

a. Update `user_login()` view to:

- Generate the date and time of the current login.
- Store this value as a session parameter (*last_login__info*).
- Set the session expiry to 1 hour.

b. When the index page is visited, check if *last_login_info* exists in session. If so, display this value. Otherwise, display the message "Your last login was more than one hour ago".

Is the value changing each time you visit the index page? Check what happens when:

i) you are logged in as an authenticated user
ii) you are **not** logged in.

c. Update `user_logout()` view to log out the current user by deleting the request session instead of calling the `logout()` function.

d. Make the user's session cookies to expire when the user's web browser is closed instead of one hour.

i) do you need to change anything in your view functions?

ii) what setting you use to do that?