COMP 8347: Internet Applications and Distributed Systems

SUMMER 2021 LAB #8

PART 1: Work with Forms

Update your app so that users are able to search for courses of a given *length*. They should also be able to see a list of possible choices for length and select a particular *length*. The app should then display all courses that are of that *length*. Create a new file *forms.py* in your *myapp* dir.

1. Create new form class.

The form should have 2 fields:

i) *name*: The user can enter a text input specifying their first name. This field may be left blank when submitting the form.

ii) *length*. The user to selects one value of *length* from a list of displayed choices. The field should be displayed using a 'checkbox'.

Add the following lines to your *forms.py* file to create the new form class.

```
from django import forms

class SearchForm(forms.Form):
    LENGTH_CHOICES = [
        (8, '8 Weeks'),
        (10, '10 Weeks'),
        (12, '12 Weeks'),
        (14, '14 Weeks'),
    ]
    name = forms.CharField(max_length=100, required=False)
    length =   forms.TypedChoiceField(widget=forms.RadioSelect,
                            choices = LENGTH_CHOICES, coerced=int)
```

2. Define a view function **findcourses(request)** as follows: Add the following lines to views.py

```
from django.http import HttpResponse
from .forms import SearchForm
```

Add the following function in *views.py*

```
def findcourses(request):
    # breakpoint()
    if request.method == 'POST':
        form = SearchForm(request.POST)
        if form.is_valid():
            name = form.cleaned_data['name']
            length = form.cleaned_data['length']
            topics = Topic.objects.filter(length=length)
            courselist = []
            for top in topics:
                courselist = courselist + list(top.courses.all())
            return render(request, 'myapp/results.html',
{'courselist':courselist, 'name':name})
        else:
            return HttpResponse('Invalid data')
    else:
```

```
form = SearchForm()
return render(request, 'myapp/findcourses.html', {'form':form})
```

3. Update *myapp/urls.py:* Add the necessary *path* to *myapp/urls.py* file so this *findcourses* view function is executed if the user accesses the url *myapp/findcourses* under your website.

4. Create template files:

a. Create a new file *findcourses.html* in dir *mysiteS21/myapp/templates/myapp*

b. Edit *findcourses.html* as follows:

```
{% extends 'myapp/base.html' %}

{% block body_block %}
    <form action="{% url 'myapp:findcourses' %}" method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <input type="submit" name="submit1" value="Find Courses" />
    </form>

{% endblock %}
```

c. Create a new template file *results.html* in dir *mysiteS21/myapp/templates/myapp* This file should only contain a single line as follows: `{% extends 'myapp/base.html' %}`

d. Update *results.html* The file should display the text 'Hello *name'* and then the list of courses in *courselist*, with a suitable message. Here *name* and *courselist* are the items passed to the template from the view function **`findcourses(request)`**.

PART 2: Update Forms

1. Modify SearchForm class. Modify the **SearchForm** class created in **PART 1:step 1a** as follows:

a. Change the label for the name field to 'Student Name'.

b. Make the *length* field optional and change its label to 'Preferred course duration:'

c. Add a new required field *max_price* that accepts an integer value.

The *label* for this field should be 'Maximum Price'.

Users should not be allowed to enter a value less than 0.

2. Update *findcourses* view.

a. Modify the *findcourses* function from **PART 1:step 2** to send *name*, *length* and *courselist* to the template file *results.html*. Here *name* and *length* are values obtained from the form fields and *courselist* is a list of courses containing:

i. courses whose price is less than or equal to *max_price* and of the selected *length* (if a length has been selected).
ii. all courses in the database with a price less than or equal to *max_price* (if no length is selected).

c. Update the template *results.html* to display *name*, *category* and *courselist* as appropriate.
d. Update *base.html* to add a link to **url** *myapp/findcourses/* in addition to the main (index) and about page.