University of Windsor
School of Computer Science
https://cs.uwindsor.ca

Master of Applied Computing
COMP-8347 – Winter 2023
Internet Applications and Distributed Systems

Dr. Usama Mir – Usama.Mir@uwindsor.ca
Dr. Saja Al-Mamoori – sajak@uwindsor.ca

LAB 5 – Django Views

Marks: 2

Submission: You should submit 2 files for this lab, ***views.py*** and ***urls.py*** under your myappF23 dir.

In this lab, you'll practice creating view functions for your Django app ***myappF23*** *inside views.py* file**.** The views are ***index*** for the home page to display a list of categories, ***about*** for the about page and ***detail*** for displaying the details of a category. You'll also create suitable URL patterns for your views in *urls.py* file.

Before creating views and inside *models.py*, add a new model **Order** with fields:

- *course* (*ForeignKey(Course)*) that indicates the *course* that was ordered.
- *Student* (*ForeignKey(Student)*) that indicates the *student* that ordered the *course.*
- *order_status(IntegerField)*
  - choices of valid values = {0,1}. The default value is **1**. The values are interpreted as: `[(0, 'Order Confirmed'), (1, 'Order Cancelled')]`.
  - **HINT**: Use similar format as *status* field in ***Student*** model.
- *order_date*: (*DateField*) that indicates the date the *order_status* was last updated.

Register this model inside *admin.py*. Run the app and enter some data to create orders.

Create views in *views.py* under myappF23.

1. Create ***index*** view. This will be the view for the main landing page of your app. It should return the **`index`** page as a response. The page should show a list of the categories in your Distance-Ed webapp database.

a. Edit your *views.py* file as follows:

```
# Import necessary classes
from django.http import HttpResponse
from .models import Category, Course, Student, Instructor

# Create your views here.
def index(request):
```

Dr. Usama Mir – Usama.Mir@uwindsor.ca
Dr. Saja Al-Mamoori – sajak@uwindsor.ca

```
category_list = Category.objects.all().order_by('id')[:10]
response = HttpResponse()
heading1 = '<p>' + 'List of categories: ' + '</p>'
response.write(heading1)
for category in category_list:
    para = '<p>'+ str(category) + '</p>'
    response.write(para)
return response
```

Design url patterns in urls.py

a. Create *urls.py* under your `myappF23` dir and edit it as follows:

```
from django.urls import path
from myappF23 import views

app_name = 'myappF23'

urlpatterns = [
    path(r'', views.index, name='index'),
]
```

b. In your project's *urls.py* import the functions **include** and **path** from **`django.urls`** then add the following line to the list of paths (urls). This line will include all the urls of the app in the project.

```
path(r'myappF23/', include('myappF23.urls')),
```

Start your server and go to **`myapp`**. You should see a list of categories.

2. Update the *index* view function: update the view so the response **`index`** page displays a list of up to 5 *courses* as well, not necessarily related to a specific category. The *courses* should be sorted in descending order of price (i.e., most expensive first).

Dr. Usama Mir – Usama.Mir@uwindsor.ca
Dr. Saja Al-Mamoori – sajak@uwindsor.ca

3. Create ***about*** view: Define another function `about(request)` in your *views.py* file. When the user goes to url **`myapp/about`** it should display the following text: "**This is a Distance Education Website! Search our Categories to find all available Courses.**". Update *myyappF23/urls.py* with a suitable url pattern for the *about* page.

4. Create ***detail*** view: Define another view function `detail(request, category_no)` in your *views.py* file. When the user goes to url **`myapp/category_no`**, the function returns a **`detail`** page as a response. The page should display the category *name* and the list of *courses* for that category. Use *named groups* to capture the **`category_no`** from the url and pass it as an argument to `detail(request, category_no)`. Update *myyappF23/urls.py* with a suitable url pattern for the *detail* page.

5. If a user types a url **`myapp/category_no`** and that **`category_no`** does not exist, the function `detail(request, category_no)` should return the **`detail`** page with a **Page not found (404)** error.

    - HINT: To solve this part, import the function **`get_object_or_404()`** from `django.shortcuts`.