



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 6 – Django Templates

Marks: 2

Submission: On BrightSpace, submit *views.py* file under myappF23 and all the templates (.html) files inside the above new created dir. Read the instructions below.

PART 1: Create basic templates:

1. Templates directory:

Check if a dir named *templates* has been created inside your *myappF23* dir. If it does not exist, create a new dir *DistanceEd/myappF23/templates*. Inside *myappF23/templates* create a new dir called *myappF23* to store your template files. So, your templates will be in this dir hierarchy ***DistanceEd/myappF23/templates/myappF23***.

2. Update index view.

a. Import *render* by adding following line to *views.py*.

```
from django.shortcuts import render # it may be already in your views.py file
```

b. Update *index* view in *views.py* as follows:

```
def index(request):  
    category_list = Category.objects.all().order_by('id')[:10]  
    return render(request, 'myapp/index0.html', {'category_list':  
category_list})
```

c. If you are passing any extra context variables to the template, let the grader know during the evaluation.



3. *index0.html*

a. Create a new file *index0.html* in dir *../templates/myappF23*

b. Edit *index0.html* as follows:

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>E-Learning WebApp!</title>
</head>
<body>
  {% if category_list %}
    <h2>List of Categories</h2>
    <ol>
      {% for category in category_list %}
        <li> {{ category.name}} </li>
      {% endfor %}
    </ol>
  {% else %}
    <strong>There are no available categories!</strong>
  {% endif %}
</body>
</html>
```

c. Go to the url for your app to view the *index0* page.

d. At the bottom of the *index0* page, add a link to go to the ‘About’ page by adding the following line:

```
<p><a href="/myappF23/about">About page</a></p>
```

e. Change the above hardcoded url to the about page so that it uses **url namespace**.



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

4. *about0.html*

- Create a new file *about0.html* in dir *../templates/myappF23*. In addition to displaying the text for the about page, it should have a link to go back to the main (*index0.html*) page. Try using both **hardcoded url** and **url namespace** for this.
- Update the about view function in *views.py* so that it uses the *about0.html* template.
- Do you need to pass any extra context variables to the template? YES NO
- If yes, what variable(s) are you passing?

5. *category_detail0.html*

- Create a new file *detail0.html* in dir *../templates/myappF23*
 - detail0.html* will be rendered if you go to the url: ~~—~~**myappF23/<category_no>**
 - It should display the *name*, and *category* for the topic whose **id=category_no**. It should also display the list of courses for that category.
 - The *detail0* page should also have a link to go back to the main (*index0.html*) page (as in *about0.html*). Use **url namespace** for this.
- Update the detail view function in *views.py* so that it uses the *detail0.html* template.
- Do you need to pass any extra context variables to the template? YES NO
- If yes, what variable(s) are you passing?
- Update *index0.html* so that you can click on a **Category Name** from the listed category to go to the detail page for that category.

6. **Your turn:** Now create a new view function that retrieves an instructor name, the course(s) he/she teaches and the students who took those courses. Pass the instructor's name, courses, and students to a template using context variables. Design a suitable url pattern for the view function. The name of the function, template, and context variables, and the url pattern are of your choice.