

<b>CS471 - Web Technologies (Laboratory)</b>		<b>Lab Week 4</b>
		<b>Django and MVT Architecture</b>

Description.... here

### Pre-lab Preparation:

1. Do this
2. Do this

### Lab Activities:

**Part 1: Define patterns globally (core/urls.py) and specifically for each app/module**

**Task 1: what is here**

**Step 1:** what to do step-wise

**Part 2: Define action functions in views.py to handle requests**

**Task 1: what is here**

**Step 1:** what to do step-wise

**Part 3: Create a simple HTML template**

**Task 1: what is here**

**Step 1:** what to do step-wise

**Part 3: Create a simple HTML template that processes a context**

**Task 1: what is here**

**Step 1:** what to do step-wise

## Part 1: Define patterns globally (core/urls.py) and specifically for each app/module

### Task 1: what is here

#### Step 1: what to do step-wise

صفحة الـ urls في الـ pages

```
project1 > urls.py
1 """
2 URL configuration for project1 project.
3
4 The 'urlpatterns' list routes URLs to views. For more information please see:
5     https://docs.djangoproject.com/en/5.1/topics/http/urls/
6 Examples:
7 Function views
8     1. Add an import: from my_app import views
9     2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', include('pages.urls')),
23 ]
24
```

```
pages > urls.py > ...
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.usermodel, name='usermodel'),
6     path('book', views.bookmodel, name='bookmodel'),
7 ]
```

★ Specifically

★ Globally

Task 1: Left Pic - explain where you define the main routing configuration for the whole Django Project. it acts as the entry point for all URL requests.

Right Pic - each app in django can have its own urls.py file, which defines the routes specific to that app. this modular structure makes it easier to manage large Project.

Step: Left Pic - 1- Open core/urls.py 2- import required modules 3- Define the URL patterns 4- Ensure urls.py is included in settings.py file.

Right Pic - 1- Create or open the urls.py in each app 2- import required modules 3- Define the URL patterns 4- Link app specific URLs to the global urls.py.

## Part 2: Define action functions in views.py to handle requests

### Task 1: what is here

#### Step 1: what to do step-wise

```
EXPLORER
PROJECT1
  pages
    > __pycache__
    > migrations
    > __init__.py
    > admin.py
    > apps.py
    > models.py
    > tests.py
    > urls.py
    > views.py
  project1
    > __pycache__
    > __init__.py
    > asgi.py
    > settings.py

pages > views.py > bookmodel
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 # Create your views here.
4
5 def usermodel(request):
6     x={'name': 'ahmed', 'age': 25}
7     return render(request, 'pages/index.html', x)
8     #return HttpResponse('Hello')
9
10 def bookmodel(request):
11     return HttpResponse('about page')
```

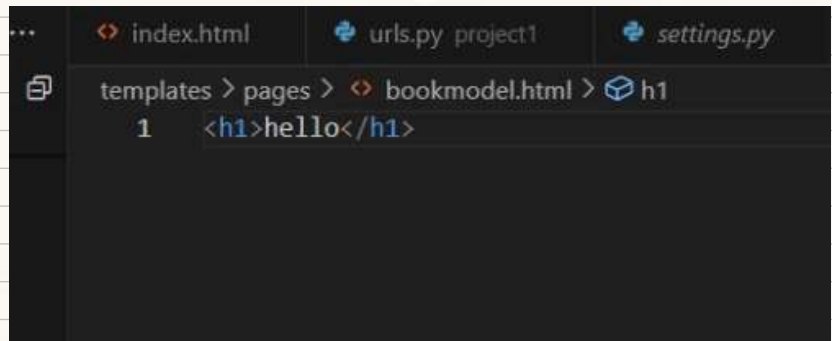
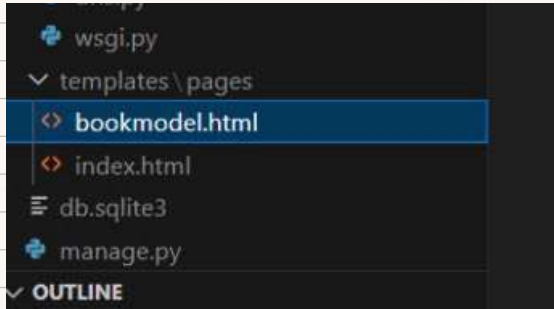
**Task 1:** This is a basic action function in Django responsible for taking the HTTP request and returning the HTTP response to the client.

**Step 1:** 1- Import required libraries 2- Define the Action Function 3- Return HTTP Response

## Part 3: Create a simple HTML template

**Task 1: what is here**

**Step 1:** what to do step-wise



**Task 1:** Django uses templates to dynamically render HTML pages. Templates allow you to your Python code from the Presentation Layer (HTML).

**Step 1:** 1- Set up the template directory 2- create a simple HTML template 3- Run Server and view the template

## Part 3: Create a simple HTML template that processes a context

**Task 1: what is here**

**Step 1:** what to do step-wise

**Task 1:** System allows you to pass context from Views to templates. This task will guide you through the creation of a basic HTML template that dynamically processes context passed from a Django view.

**Step 1:** 1- Set up the template directory 2- Create a simple HTML template 3- modify the View function 4- define a URL pattern for the view 5- Run the Django server