Mar 25

Ex:

D:\Django_20MAR_7PM>django-admin startproject applevelurlsproject

D:\Django_20MAR_7PM>cd applevelurlsproject

D:\Django_20MAR_7PM\applevelurlsproject>py manage.py startapp testapp

-->Add app in settings.py

views.py

```
from django.http import HttpResponse
def exams_view(request):
    return HttpResponse('<h1>Exams View</h1>')
def attendance_view(request):
    return HttpResponse('<h1>Attendance View</h1>')
def fees_view(request):
    return HttpResponse('<h1>Fees View</h1>')
```

- **urls.py-->Application level**

```
from django.urls import path
from . import views
urlpatterns = [
    path('exams/', views.exams_view),
    path('attendance/', views.attendance_view),
    path('fees/', views.fees_view),
]
```

Create another project:

D:\Django_20MAR_7PM>django-admin startproject sunnyproject

-->Copy testapp from applevelproject and paste it in current project, then include app level urls in project level urls.

project level urls

path('testapp/', include('testapp.urls')),

start server send request:

http://127.0.0.1:8000/testapp/exams/

http://127.0.0.1:8000/testapp/fees/

http://127.0.0.1:8000/testapp/attendance/

Chapter-3

Django Templates & Static Files

-->It is not recommended to write HTML code inside python script(views.py file) because:

1.It reduces readability because of python code mixed with HTML code.

2.No separation of roles. Python developers has to concentrate on both python code and HTML code.

3.It does not promotes re-usability of code.

-->We can overcome these problems by separating HTML code into a separate html file. This html file is nothing but template.

-->From the python file(views.py) we can use these templates based on our requirement.

-->We have to write templates at project level only, we can use these templates in multiple applications.

Python stuff:

pathlib-->module name

Path-->class name

pathlib module provides various classes representing file system paths based on different OS.

```
from pathlib import Path

print(__file__)#It will returns the name of the file:test.py

fpath = Path(__file__)

print(type(fpath))#<class 'pathlib.WindowsPath'>

complete_path = fpath.resolve()

print(complete_path)#D:\Mahesh_Classes\test.py

print(Path(__file__).resolve().parent)#D:\Mahesh_Classes

print(Path(__file__).resolve().parent.parent)#D:
```

Note:

The main advantage of this aapproach is we are not required to hard code system specific paths(locations) in python script.

MVC design pattern/architecture:

M-->Model(Business logic)

V-->View(Presentation Logic)

C-->Controller(C-ordination)

MVT design pattern:

M-->Model(Database)

V-->View(Business logic-->Python file)

T-->Template(Presentation Layer)

Steps to develop Template Based Application:

D:\Django_20MAR_7PM>django-admin startproject templateproject

D:\Django_20MAR_7PM>cd templateproject

D:\Django_20MAR_7PM\templateproject>py manage.py startapp testapp

-->Add app in settings.py

-->Create a 'templates' folder inside main project folder.

-->In that templates folder create a separate folder named with testapp to hold that particular application specific templates.

-->Add templates folder to settings.py file so that django can aware of our templtes.

```
TEMPLATES = [
                    'DIRS':
[D:\Django_20MAR_7PM\templateproject\templates],
]
```

-->It is not recommended to hard code system specific location in settings.py file.

To overcome this problem, we can generate templates directory path programmatically as.

```
BASE_DIR = Path(__file__).resolve().parent.parent
TEMPLATE_DIR = BASE_DIR/'templates'
```

-->Specify TEMPLATE_DIR inside settings.py

```
'DIRS': [TEMPLATE_DIR],
```

-->Create html file inside templateproject/templates/testapp folder. This html file is nothing but Template.

wish.html

```
<body>
```

```
    <h1>Welcome to Django Template Demo</h1>
    <h2>Second hero of Django in MVT:Templates</h2>
</body>
```

-->Define function based view inside views.py

views.py

```
def wish(request):
    return render(request,'testapp/wish.html')
```

Define url-pattern:

urls.py

```
path('test/',views.wish)
```

start server send request:http://127.0.0.1:8000/test/