

Banarsidas Chandiwalala Institute of Information And Technology

Bachelor of Computer Applications

Django Web Developement

Programme Code- 218

Student Name	ALOK SHARMA
Enrolment No.	02211102021
Program	BCA
Batch/Semester	2021-24 / 4th Semester
Submitted To	
Faculty Name	Mr. Alok Mishra

BCIIT

BCA Sem IV

Lab Manual-1 Web Development through Django (BCA-218)

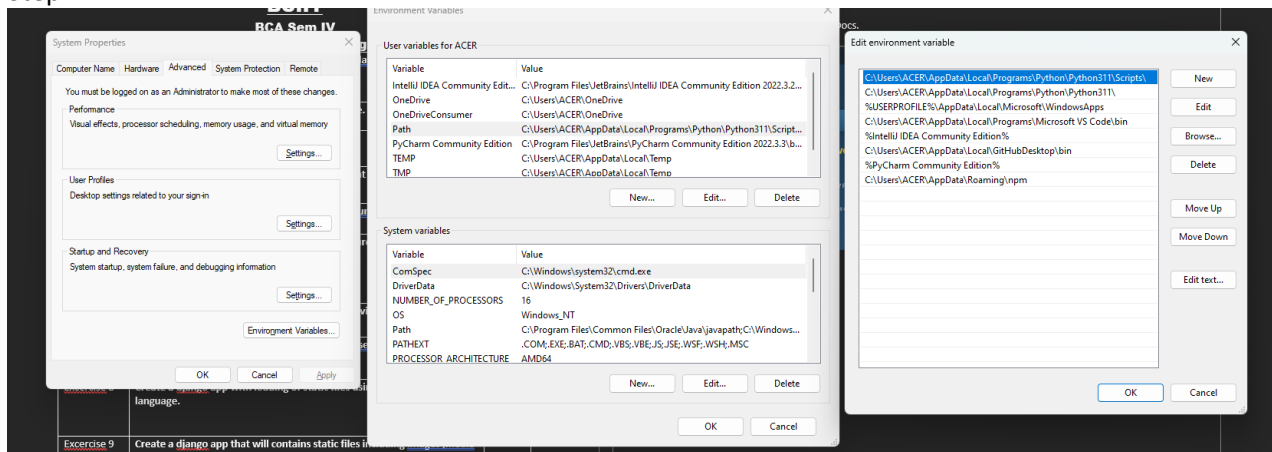
Excercise1	Installpythonincludinginstallationofpip,installationandsettingupvirtualenv vironment&Installationof Django.	status
Excercise2	Createanewdjangoprojectusingcommandline.	
Excercise3	Createanapplicationwith'Helloworld'.	
Excercise4	Createadjangoappthatwilldefineurlpatternatapplicationlevelinsteadof projectlevel.	
Excercise5	Createadjangoapplicationwithmultipleviewfunctions.	
Excercise6	Createadjangoapplicationusingtemplatefeatures Renderingatemplate Multiple ContextContextvariable Templateloadingincludetemplatetag.	
Excercise6	Createa django appwith dynamiccontentfromviewfunction.	
Excercise7	Createadjangoappwithdynamicgreettexonserver.	
Excercise8	Createadjangoappwithloadingofstaticfilesusingjinja2templatelanguage.	
Excercise9	Createadjangoappthatwillcontainsstaticfilesincludingimages,media&cssfiles.	
Excercise10	Createadjangoappwithadvancedtemplatefeatures.	
Excercise11	Apptoconnecttemplateswithmodelstoserver datadynamically.	
Excercise12	RenderingamodelinDjangoAdminInterface	
Excercise13	Writeadjangowebapptouseparametersfromviews.py	
Excercise14	Writeadjangowebappusingcontrolstatements.	

Excercise1: Install python including installation of pip, installation and setting up virtual environment & Installation of Django.

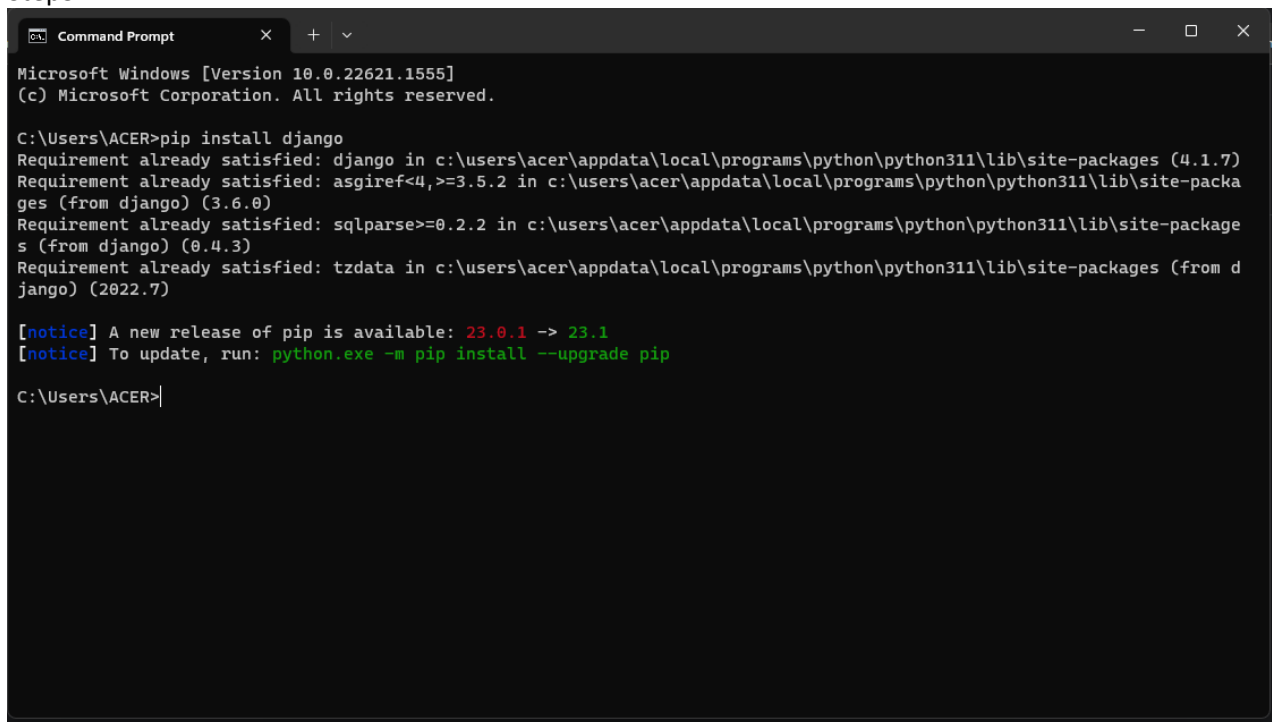
Step 1: Install Python from official Docs.



Step2:



Step3:



Excercise2. Createa new django project using command line.

```
Command Prompt
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>pip install django
Requirement already satisfied: django in c:\users\acer\appdata\local\programs\python\python311\lib\site-packages (4.1.7)
Requirement already satisfied: asgiref<4,>=3.5.2 in c:\users\acer\appdata\local\programs\python\python311\lib\site-packages (from django) (3.6.0)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\acer\appdata\local\programs\python\python311\lib\site-packages (from django) (0.4.3)
Requirement already satisfied: tzdata in c:\users\acer\appdata\local\programs\python\python311\lib\site-packages (from django) (2022.7)

[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\ACER>django-admin startproject myproject|
```

Excercise3, 4. Createanapplicationwith‘Helloworld’ and Createadjangoappthatwilldefineurlpatternatapplicationlevelinsteadof projectlevel.

Step1: urls.py:

```
contact.html  views.py  base.html  index.html  urls.py ...\exp1 M X  urls.py ...\home

Experiments > exp1 > exp1 > urls.py
1  """exp1 URL Configuration
2
3  The `urlpatterns` list routes URLs to views. For more information please see:
4  |   https://docs.djangoproject.com/en/4.1/topics/http/urls/
5  |   Examples:
6  |   Function views
7  |       1. Add an import:  from my_app import views
8  |       2. Add a URL to urlpatterns:  path('', views.home, name='home')
9  |   Class-based views
10 |       1. Add an import:  from other_app.views import Home
11 |       2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
12 |   Including another URLconf
13 |       1. Import the include() function: from django.urls import include, path
14 |       2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15 |   """
16  from django.contrib import admin
17  from django.urls import path, include
18
19  urlpatterns = [
20      path('admin/', admin.site.urls),
21      path('', include('home.urls')),
22      path('hello', include('hello.urls')),          # Add this line
23  ]
24
25
```

Step2: create a app in your Django project using below command line “py manage.py startapp hello”

Step3: create a urls.py in your hello app

```
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from hello import views
19
20 urlpatterns = [
21     path('', views.home, name='home'),
22 ]
23
```

Step4: Define home function in views.py

```
Experiments > exp1 > hello > views.py
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 # Create your views here.
5 def home(request):
6     return HttpResponse('Hello World')
```

Excercise 5, 6A, 6B, 7, 8, 9, 10, 11

Step1: create a Django home app in your Django project using below command

“py manage.py startapp home”

Step2: Go to views.py and create multiple views function

```
Experiments > exp1 > home > views.py
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from datetime import datetime
4 from home.models import User
5 # Create your views here.
6 def index(request):
7     name = {'Name': 'Alok Sharma'}
8     enroll = {'Enroll': '02211102021'}
9
10    now = datetime.now()
11    if now.hour < 12:
12        greeting = 'Good morning'
13    elif now.hour < 18:
14        greeting = 'Good afternoon'
15    else:
16        greeting = 'Good evening'
17    return render(request, 'index.html', {'greeting': greeting, 'Name': name, 'Enroll': enroll})
18
19
20 def dynamicData(request):
21     now = datetime.now()
22     return render(request, 'date.html', {'date': now})
23
24 def contact(request):
25     if request.method == 'POST':
26         name = request.POST.get('name')
27         email = request.POST.get('email')
28         phone = request.POST.get('phone')
29         content = request.POST.get('content')
30         print(name, email, phone, content)
31         user=User(name=name, email=email, phone=phone, content=content)
32         user.save()
33         return render(request, 'contact.html')
34
35 def getUser(request):
36     users = User.objects.all()
37     return render(request, 'user.html', {'users': users})
38
39 def getParam(request, slug):
40     return HttpResponse(f"Your slug is {slug}.")
```

Step4: For rendering template go to settings.py file and edit “TEMPLATES_DIR” to “BASE_DIR / ‘templates’ “

```
models.py x admin.py settings.py x urls.py views.py
Experiments > exp1 > exp1 > settings.py > ...
50     django.middleware.clickjacking.XFrameOptionsMiddleware,
51 ]
52
53 ROOT_URLCONF = 'exp1.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': [BASE_DIR / "templates"],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     ],
69 ]
```

Step5: create “base.html” and “index.html” inside the templates folder
And add at the top of the index.html {% extends ‘base.html’ %}

Base.html:

```
models.py admin.py settings.py base.html x urls.py user.html M # custom.css
Experiments > exp1 > templates > base.html
1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4      <head>
5          <meta charset="utf-8" />
6          <meta name="viewport" content="width=device-width, initial-scale=1" />
7          <title>{% block title %} {% endblock title %} | Exp1</title>
8          <link rel="stylesheet" href="{% static 'css/custom.css' %}" />
9          <link
10             href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
11             rel="stylesheet"
12             integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvVe+M/SXH301p5ILy+dN9+nJJOZ"
13             crossorigin="anonymous"
14         />
15      </head>
16      <body class="custom-css-gradientBg">
17          <nav class="navbar navbar-expand-lg fs-5 navbar-dark shadow-5-strong shadow p-3 rounded">
18              <div class="container-fluid">
19                  <a class="navbar-brand fs-4" href="/">
```

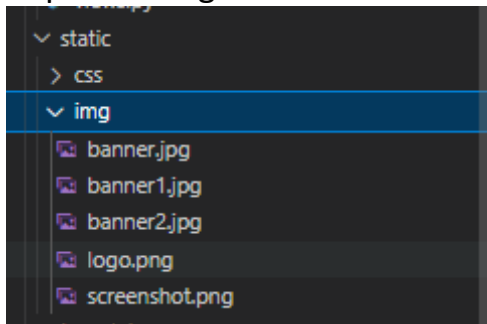
Index.html:

```
models.py admin.py settings.py index.html x urls.py user.html M # custom
Experiments > exp1 > templates > index.html
1  {% extends 'base.html' %} {% load static %}
2  {% block title %}home{% endblock title %} {% block body %}
3  <div class="text-light px-5 py-3 fs-5 d-flex justify-content-between">
4      <p>Enrollment No: {{Enroll.Enroll}}</p>
5      <p>{{ greeting }}, {{Name.Name}}</p>
6  </div>
7  <div id="carouselExampleIndicators" class="carousel slide container-fluid " data-bs-ride="true">
8      <div class="carousel-indicators">
9          <button
10             type="button"
11             data-bs-target="#carouselExampleIndicators"
12             data-bs-slide-to="0"
13             class="active"
14             aria-current="true"
15             aria-label="Slide 1"
16         ></button>
17          <button
18             type="button"
19             data-bs-target="#carouselExampleIndicators"
```

Step5: To add static file in webpage, firstly you need to add STATIC_DIR in settings.py

```
118
119     STATIC_URL = 'static/'
120
121     # Added manually
122     STATICFILES_DIRS = [
123         BASE_DIR / "static",
124     ]
125
126     # Default primary key field type
```

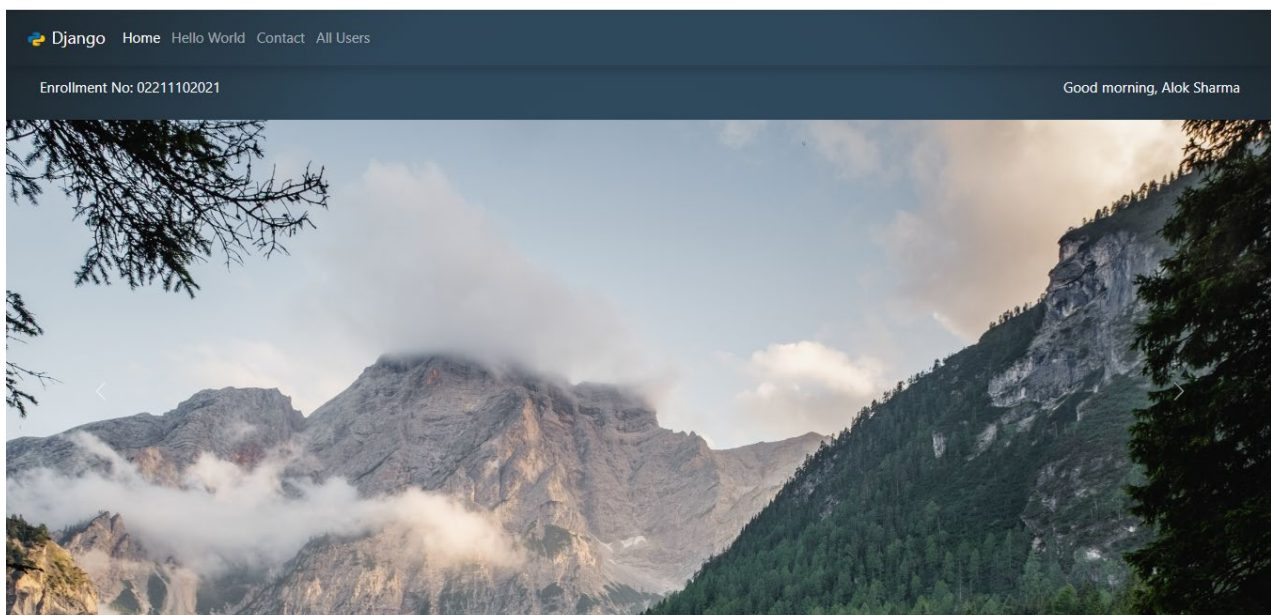
Step6: goto static directory and create a folder for called “img” and move all the required images inside the static folder



Step7: to display static images on your webpage use {% static load 'img/img_name' %}

```
<div class="carousel-item active">
  <img src={% static 'img/banner.jpg' %} class="d-block w-100" alt="bannerImage">
</div>
<div class="carousel-item">
```

Endresult:



Excercise12: Rendering a model in Django Admin Interface

Step1: Create a model inside the model.py

```
models.py X
home > models.py
1  from django.db import models
2
3
4
5  # Added manually
6  class User(models.Model):
7      sno = models.AutoField(primary_key=True)
8      name = models.CharField(max_length=255)
9      email = models.CharField(max_length=255)
10     phone = models.CharField(max_length=13)
11     content = models.TextField()
12     timeStamp = models.DateTimeField(auto_now_add=True, blank=True)
13
14     def __str__(self):
15         return self.name
16
```

Step2: register your model in “admin.py”

```
admin.py X
home > admin.py
1  from django.contrib import admin
2  from home.models import User, Emp, Product # Added manually
3  # Register your models here.
4  admin.site.register(User) # Added manually
```

Step3: Go to settings.py and include your apps.py class name inside INSTALLED_APPS

Eg:

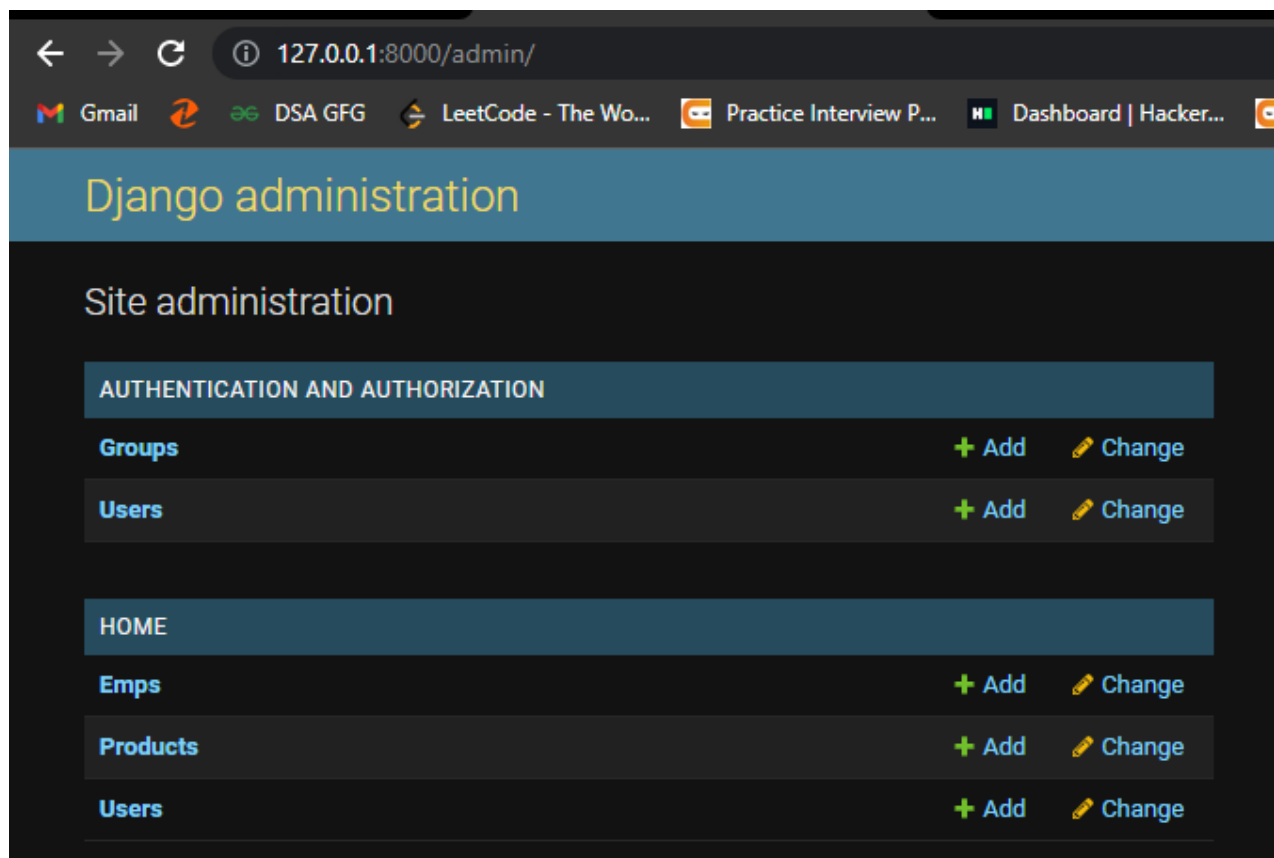
```
1  # Application definition
2
3  INSTALLED_APPS = [
4      'home.apps.HomeConfig', # Added manually
5      'django.contrib.admin',
6      'django.contrib.auth',
7      'django.contrib.contenttypes',
8      'django.contrib.sessions',
9      'django.contrib.messages',
10     'django.contrib.staticfiles',
11 ]
```


Step4: write "py manage.py makemigration" in terminal

Step5: write "py manage.py migrate" to apply all pending changes

Step6: Write "py manage.py createsuperuser" and then enter username, email and password

Step7: you'll see your model in Django Admin Panel:



Excercise13: Write a django web app to use parameters from views.py

Step1: To getData from URL define urls.py

Eg:

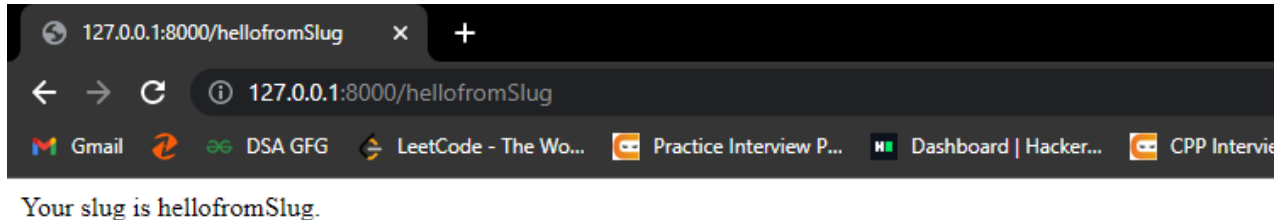
```
admin.py  urls.py  X
home > urls.py
14 | 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 | 
16 | from django.contrib import admin
17 | from django.urls import path
18 | from home import views
19 | 
20 | urlpatterns = [
21 |     path('', views.index, name='home'),
22 |     path('contact', views.contact, name='contact'),
23 |     path('users', views.getUser, name='allUsers'),
24 |     path('<slug:slug>', views.getParam, name='slug'), # Excercise13
25 |     path('emps', views.getEmp, name='allEmps'),
26 |     path('contentviews', views.dynamicData, name='content-views'),
27 | ]
28 |
```

Step2: Define views.py

Eg:

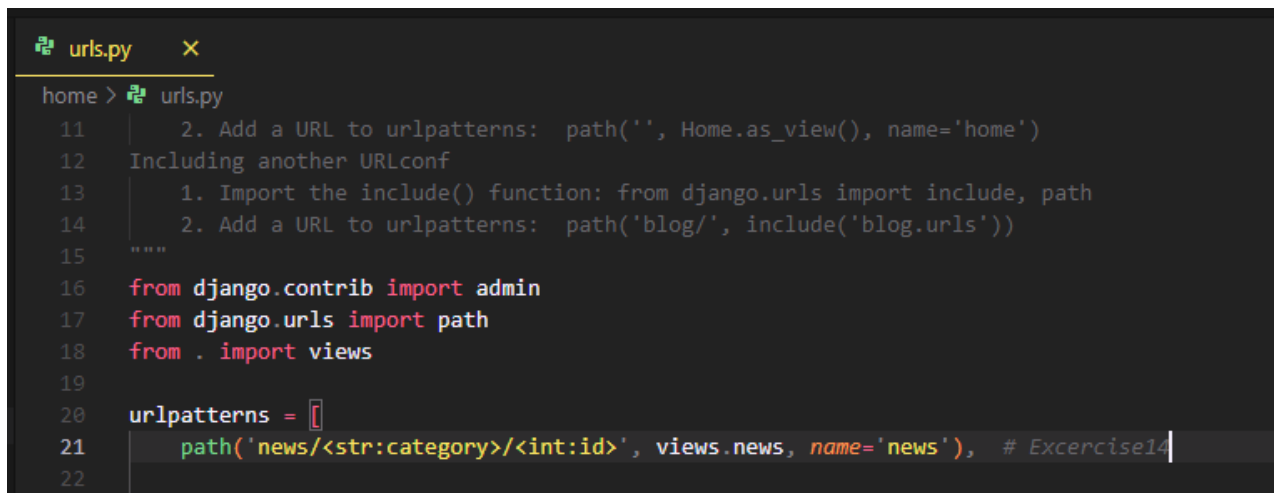
```
def getParam(request, slug):  
    return HttpResponse(f"Your slug is {slug}.")
```

Result:



Excercise14: **Write a Django web app using control statements.**

Step1: Define urls.py



Step2: Define Views.py

Eg:

```
23  
24 def news(request, id, category):  
25     if category == 'student':  
26         news = StudentNews.objects.filter(sno=id)  
27     elif category == 'faculty':  
28         news = FacultyNews.objects.filter(sno=id)  
29     elif category == 'events':  
30         news = EventsNews.objects.filter(sno=id)  
31     return render(request, 'news.html', {'news': news[0]})
```