

# MY DJANGO

- Django is Python Based Web-Framework.
- To install Django in Our Local Computer:

Go to Terminal > **pip install django** > {Downloading}

- To check Version of Django

Go to Terminal > **python -m django --version** > Enter

## *Let's Begging the Django Project*

Go to Terminal Write > **django-admin** > Run

- It shows some utilities which is you can use, and it is clear that Django are install Successfully in Our Local Computer.

## *To start our project*

Go to Terminal > **django-admin startproject {project\_name}** > Run

- It creates an automatic Directories and some codes.

- 1) project\_name is our python package.
- 2) init.py in one blank python file.
- 3) setting.py is python file consist of our python project settings.
- 4) urls.py this consist of URL declarations and mappings. (URL ke konse end point pe kya hota hai vo declare krte hai)
- 5) wsgi.py(web server getaway interface) no need to understand this file as a beginner. It uses to Deployment.
- 6) manage.py is our main file to write a code.

## **To Start the Development Server**

Go to Terminal > **python manage.py runserver**

- It shows the Development server link, copy it and open in Browser.
- We See like this In Screen..



The install worked successfully! Congratulations!

View [release notes](#) for Django 5.2

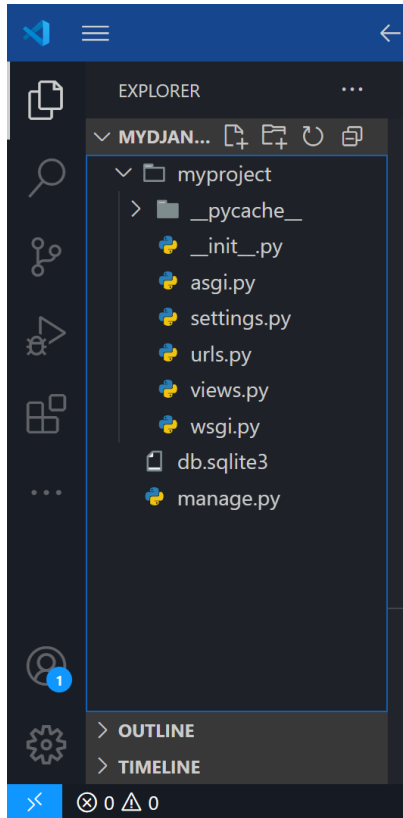
You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

**django**

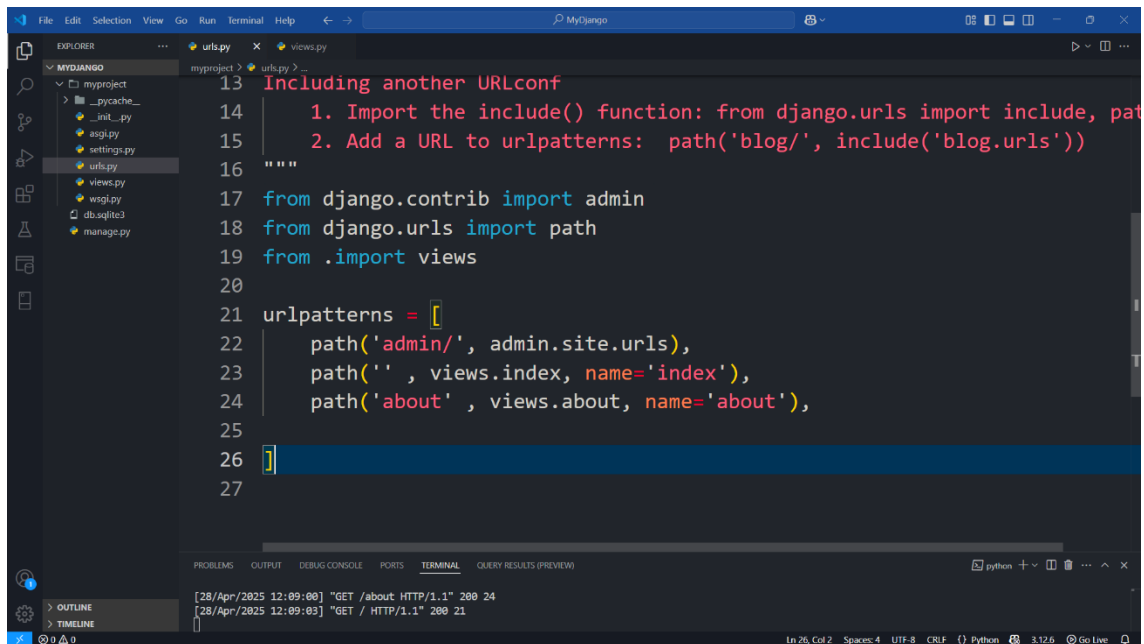
---

# Setting up Our Routes

1) Go to folder and create views.py file



2) Go to urls.py and import statements and set Path



### 3) In views.py

```
1 # i have created this file ##nikhillanje
2 from django.http import HttpResponse
3
4
5 def index (request):
6     return HttpResponse("Hello Nikhil")
7     #also
8     #return HttpResponse("<h1>Hello Nikhil</h1>")
9
10
11 def about (request):
12     return HttpResponse("Hello Nikhil from about ")
```

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.  
For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/>

4) Then Run the Server and see It Print Hello Nikhil also we can Print in `<h1>` tag

5) If we go to `/about` route then it will print Hello Nikhil from About.

**Likewise, we can Create Our Routes.....**

---

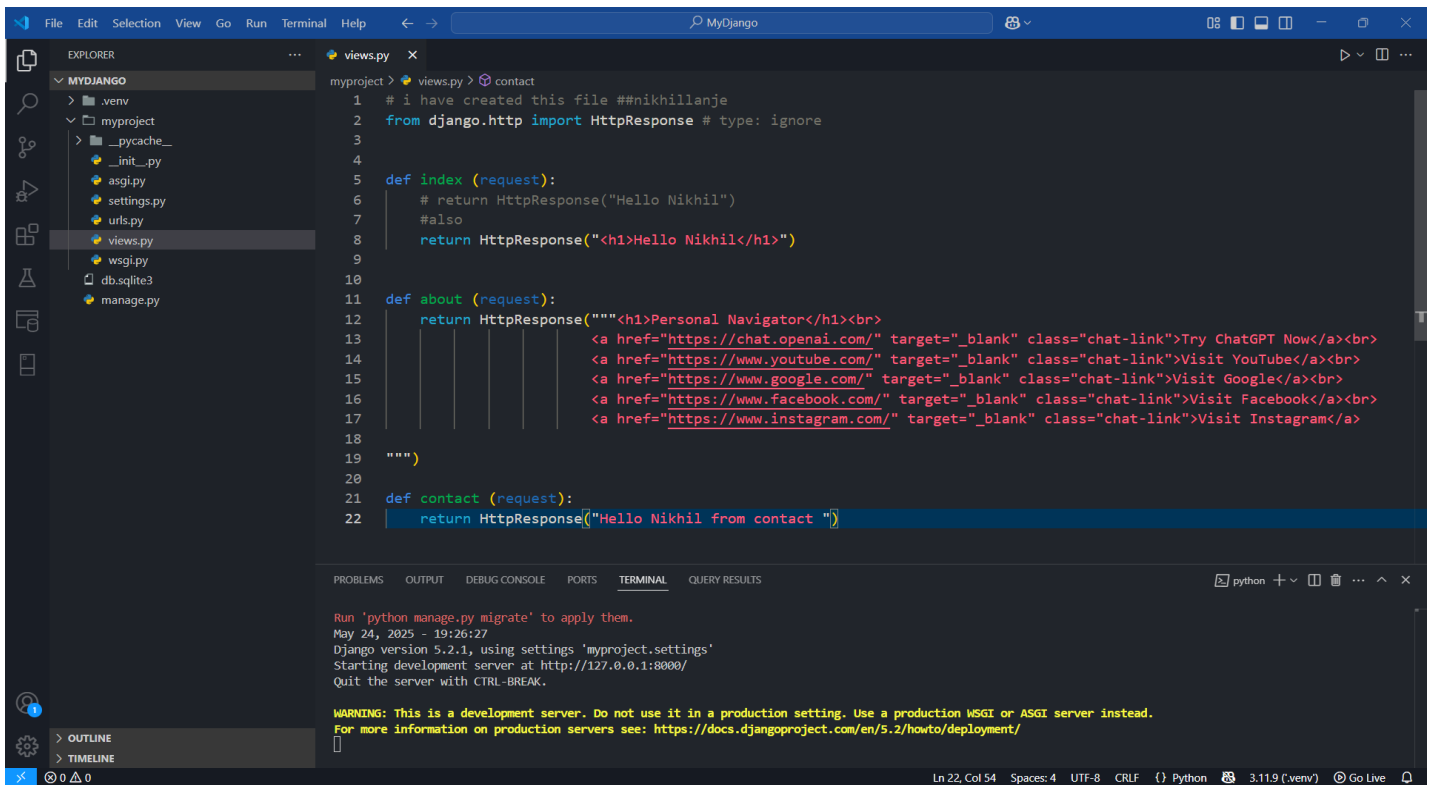
### **Exercise 1: - Personal Navigator**

Create a Personal Navigator Webs

Take Any 5 Webs and Crate a Navigator After Clicking , they redirect to that URL.

**Solution:** - 1) Go to views.py or Your any route and Create like this

2) Go to that route and Click Links.....

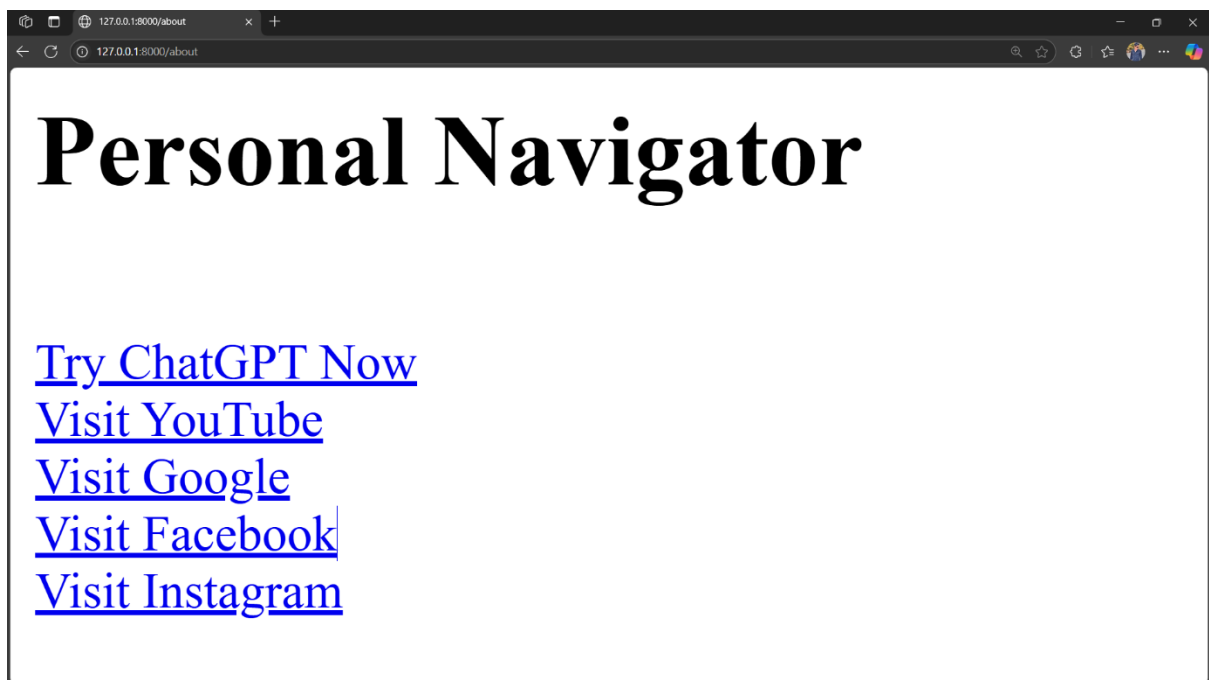


```
1 # i have created this file ##nikhillanje
2 from django.http import HttpResponse # type: ignore
3
4
5 def index (request):
6     # return HttpResponse("Hello Nikhil")
7     #also
8     return HttpResponse("<h1>Hello Nikhil</h1>")
9
10
11 def about (request):
12     return HttpResponse("""<h1>Personal Navigator</h1><br>
13     <a href="https://chat.openai.com/" target="_blank" class="chat-link">Try ChatGPT Now</a><br>
14     <a href="https://www.youtube.com/" target="_blank" class="chat-link">Visit YouTube</a><br>
15     <a href="https://www.google.com/" target="_blank" class="chat-link">Visit Google</a><br>
16     <a href="https://www.facebook.com/" target="_blank" class="chat-link">Visit Facebook</a><br>
17     <a href="https://www.instagram.com/" target="_blank" class="chat-link">Visit Instagram</a>
18     """)
19
20
21 def contact (request):
22     return HttpResponse("Hello Nikhil from contact ")
```

Run 'python manage.py migrate' to apply them.  
May 24, 2025 - 19:26:27  
Django version 5.2.1, using settings 'myproject.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.

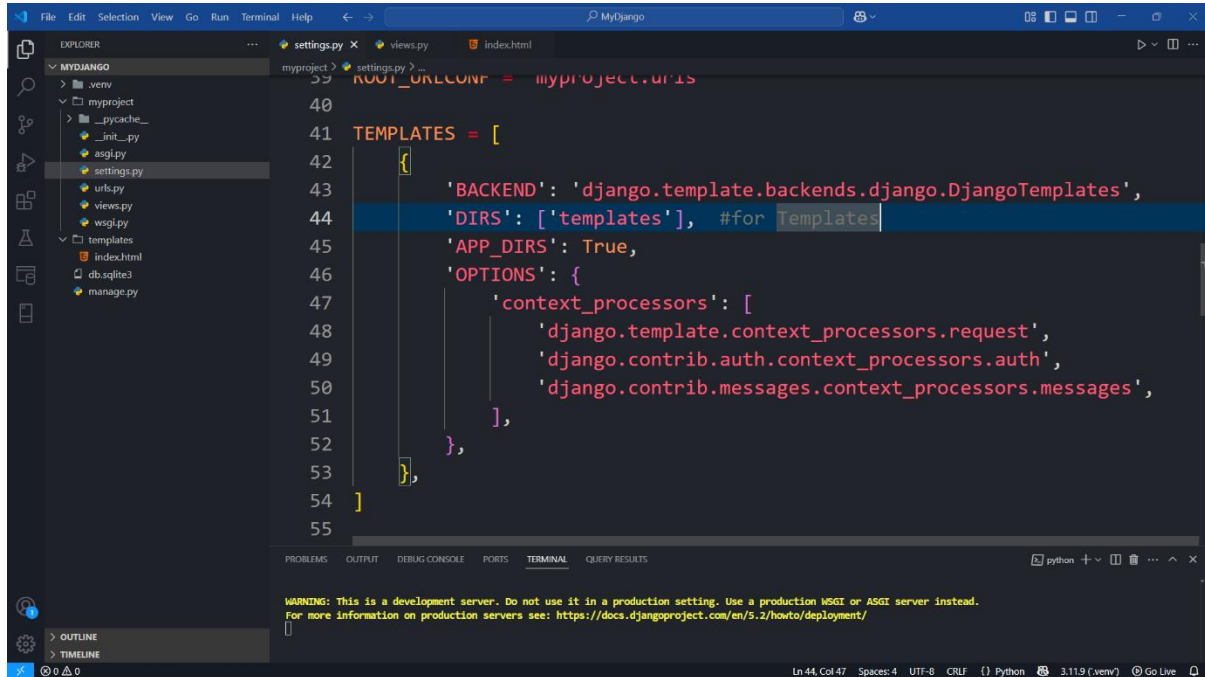
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.  
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/

## Output Is



# Django Templates

- 1) Go to Settings.py file and Find Templates and add **templates in 'DIRS'**

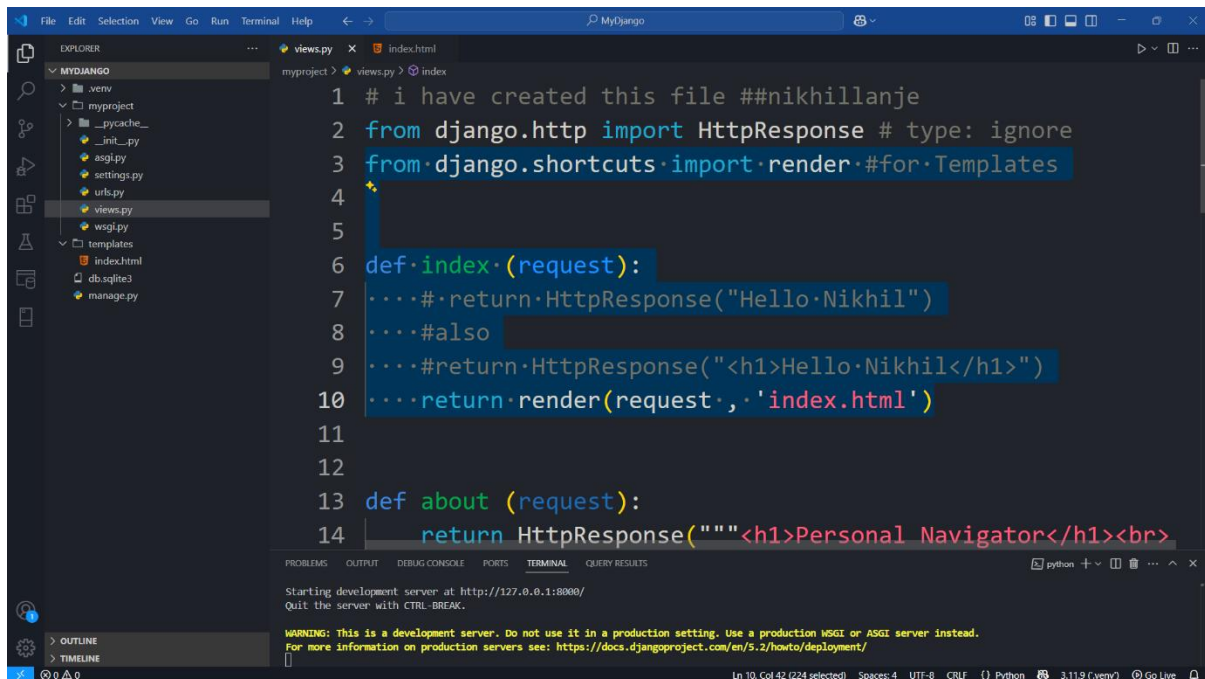


The screenshot shows the settings.py file in a Django project. The TEMPLATES configuration is being edited. The 'DIRS' list is updated to include 'templates'.

```
40 ROOT_URLCONF = 'myproject.urls'
41
42 TEMPLATES = [
43     {
44         'BACKEND': 'django.template.backends.django.DjangoTemplates',
45         'DIRS': ['templates'], #for Templates
46         'APP_DIRS': True,
47         'OPTIONS': {
48             'context_processors': [
49                 'django.template.context_processors.request',
50                 'django.contrib.auth.context_processors.auth',
51                 'django.contrib.messages.context_processors.messages',
52             ],
53         },
54     ],
55 ]
```

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.  
For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/>

- 2) In views.py import statement and render the Templates



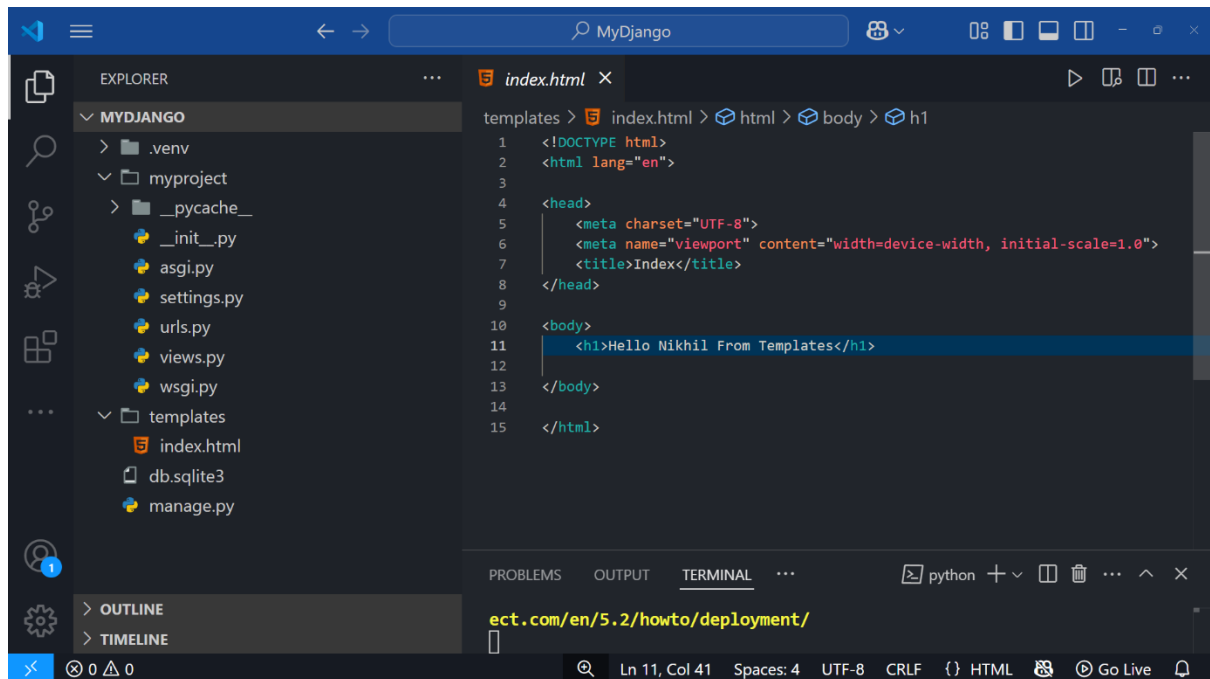
The screenshot shows the views.py file in a Django project. The index view is being edited to use the render function to render the 'index.html' template.

```
1 # i have created this file ##nikhillanje
2 from django.http import HttpResponse # type: ignore
3 from django.shortcuts import render #for Templates
4
5
6 def index(request):
7     ...#return HttpResponse("Hello Nikhil")
8     ...#also
9     ...#return HttpResponse("<h1>Hello Nikhil</h1>")
10    ...return render(request, 'index.html')
11
12
13 def about(request):
14    return HttpResponse("<<h1>Personal Navigator</h1><br>")
```

Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.  
For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/>

- 3) Also Create templates Directory in that Directory Where the manage.py is Present (ie. templates Directory and manage.py is in Same Directory).
- 4) Create index.html file in templates Directory.



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for 'MYDJANGO'. The 'templates' directory is expanded, showing 'index.html'. The main editor window displays the content of 'index.html', which is an HTML file with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Index</title>
8 </head>
9
10 <body>
11   <h1>Hello Nikhil From Templates</h1>
12
13 </body>
14
15 </html>
```

The bottom status bar indicates the file is at Line 11, Column 41, with 4 spaces, UTF-8 encoding, and CRLF line endings. The terminal panel at the bottom shows a snippet of text: 'ect.com/en/5.2/howto/deployment/'.

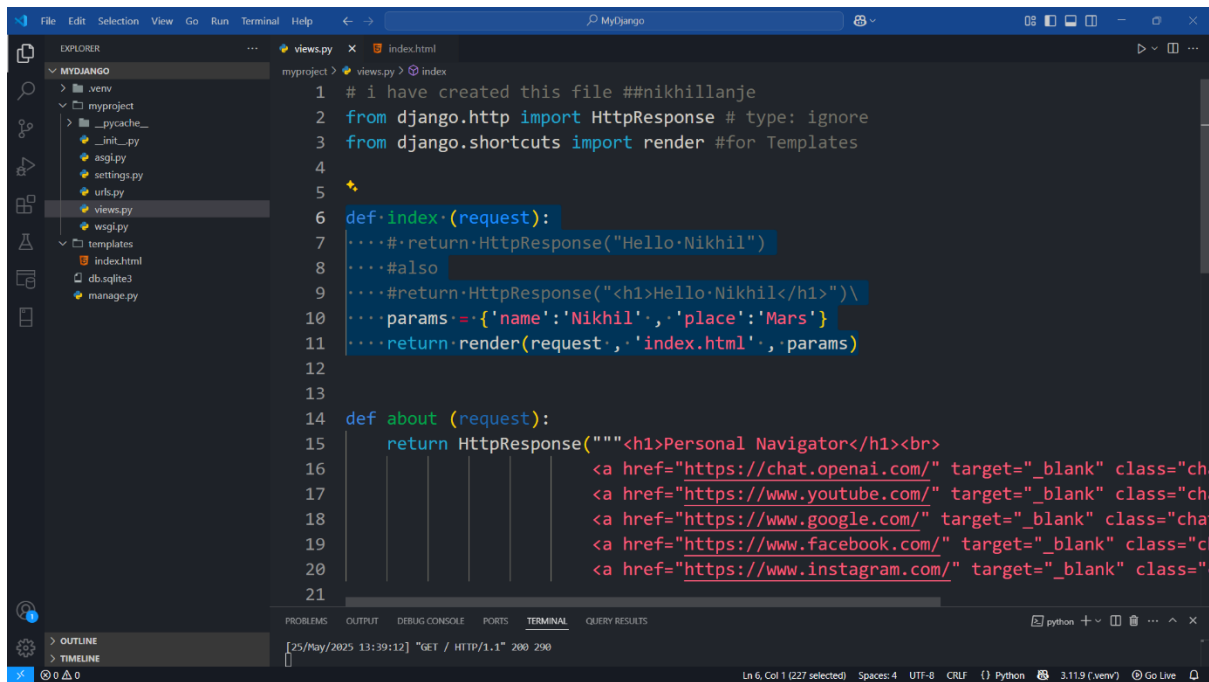
- 5) Run the server and see our Output.....



Also We Can Pass The Variables From our Python code to Templates.

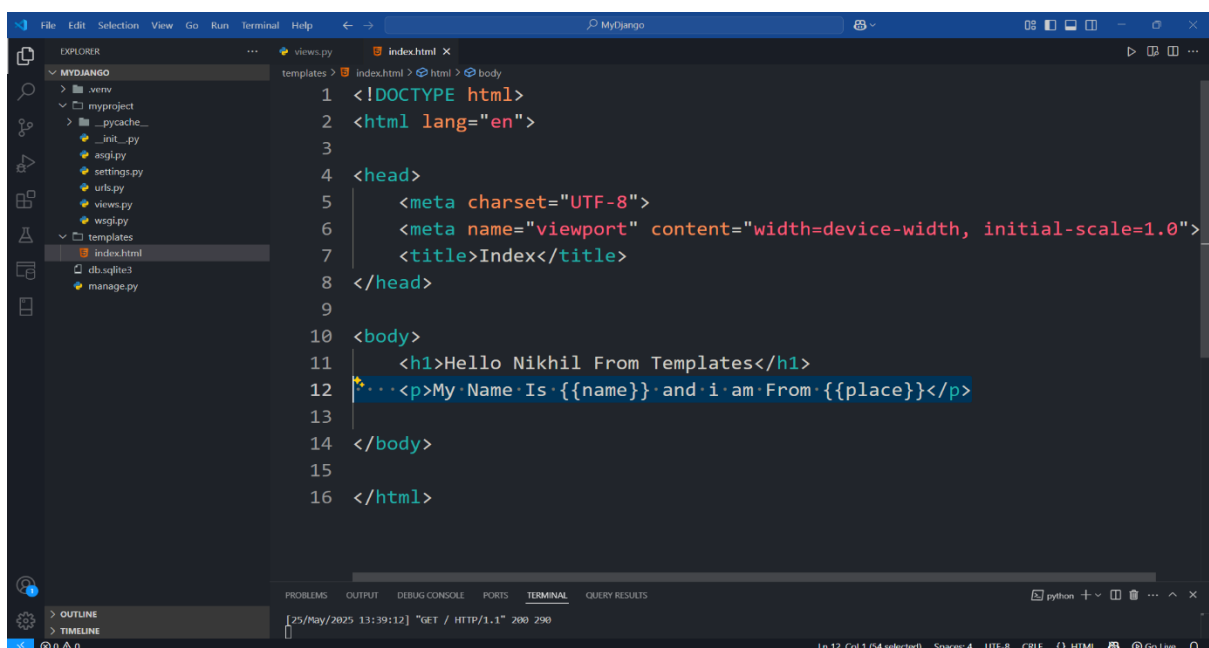
Lets See.....

- 1) Go to that route (Function) and pass variables in **params** and **pass params in return Statement also**



```
1 # i have created this file ##nikhillanje
2 from django.http import HttpResponse # type: ignore
3 from django.shortcuts import render #for Templates
4
5
6 def index(request):
7     ...#return HttpResponse("Hello-Nikhil")
8     ...#also
9     ...#return HttpResponse("<h1>Hello-Nikhil</h1>")\
10     ...params = {'name': 'Nikhil', 'place': 'Mars'}
11     ...return render(request, 'index.html', params)
12
13
14 def about(request):
15     return HttpResponse("<h1>Personal Navigator</h1><br>
16     <a href='https://chat.openai.com/' target='_blank' class='cha
17     <a href='https://www.youtube.com/' target='_blank' class='cha
18     <a href='https://www.google.com/' target='_blank' class='cha
19     <a href='https://www.facebook.com/' target='_blank' class='cl
20     <a href='https://www.instagram.com/' target='_blank' class='
21
```

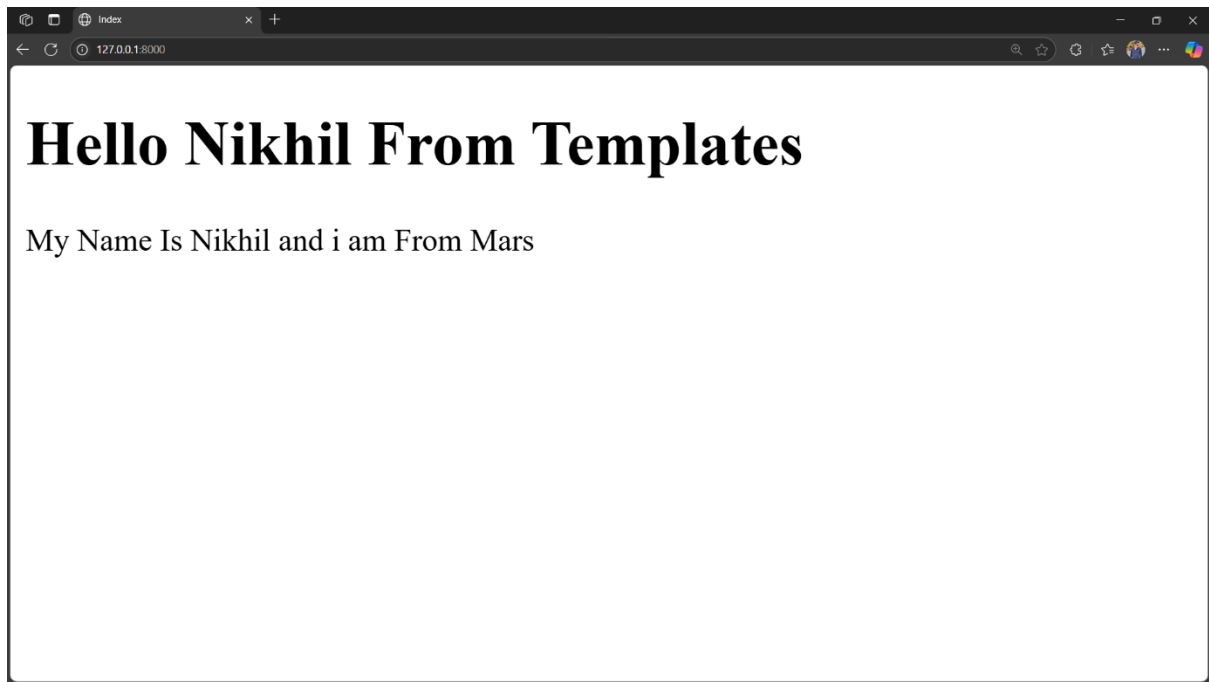
- 2) Go to Our html file and and Write in **{{variable\_name}}** to Access this Variable in html file



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Index</title>
8 </head>
9
10 <body>
11     <h1>Hello Nikhil From Templates</h1>
12     ...<p>My Name Is {{name}} and i am From {{place}}</p>
13
14 </body>
15
16 </html>
```



### 3) Run the Server and See Output

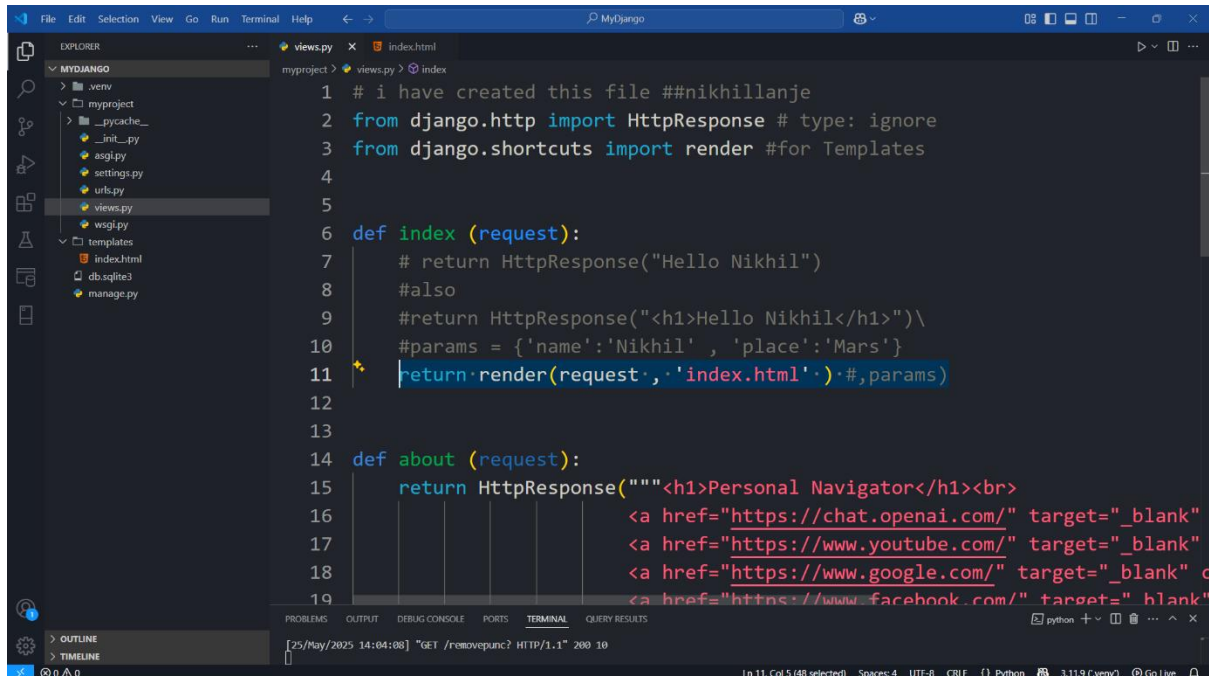


**#DONE**

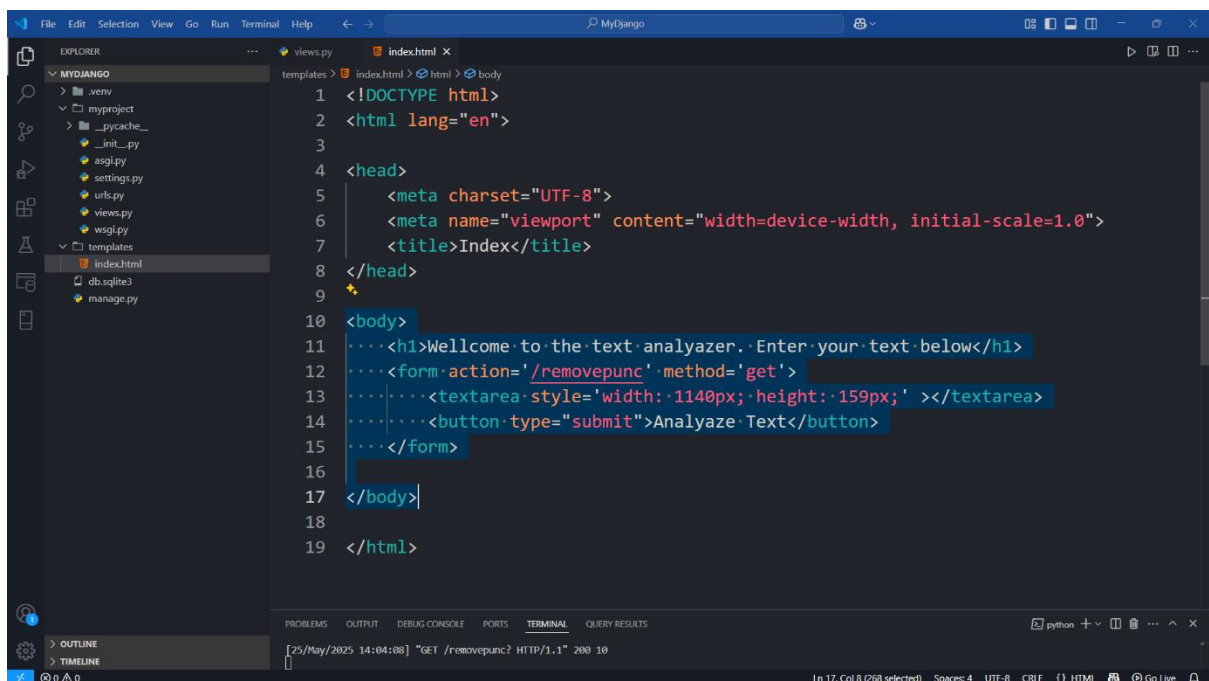
Now Let's Create Our Project.....!

# Creating Homepage of our TextUtils Website

- 1) In our Index route We render index.html , in that index page We Create one form....

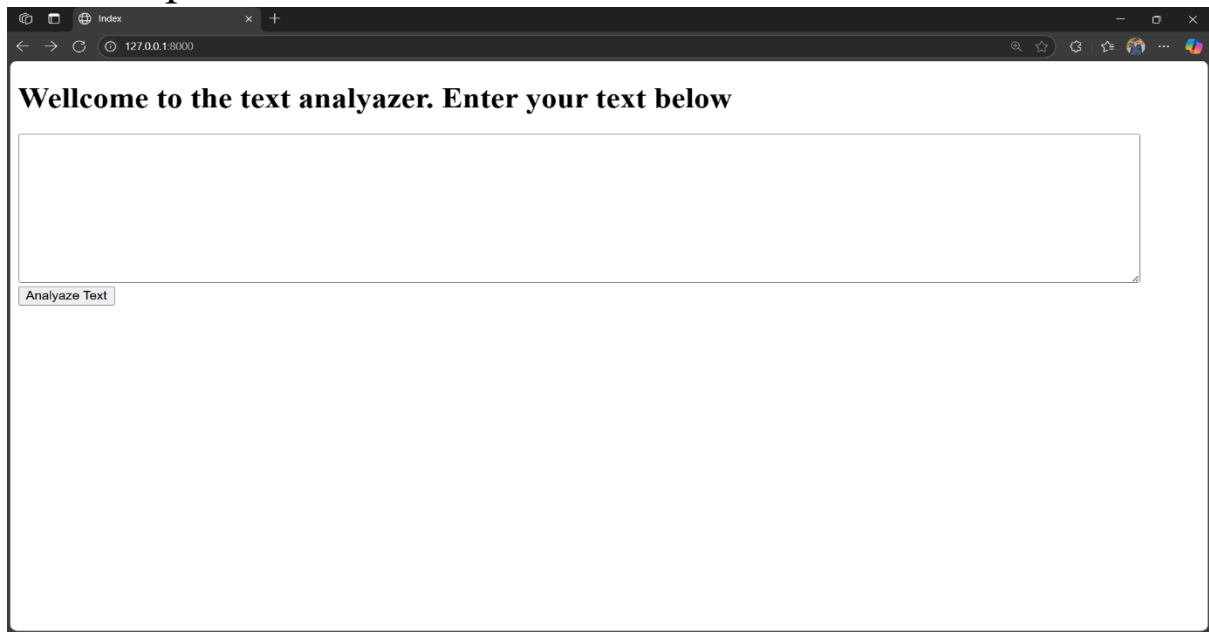


```
1 # i have created this file ##nikhillanje
2 from django.http import HttpResponse # type: ignore
3 from django.shortcuts import render #for Templates
4
5
6 def index (request):
7     # return HttpResponse("Hello Nikhil")
8     #also
9     #return HttpResponse("<h1>Hello Nikhil</h1>")\
10    #params = {'name': 'Nikhil' , 'place': 'Mars'}
11    return render(request, 'index.html', #,params)
12
13
14 def about (request):
15     return HttpResponse("""<h1>Personal Navigator</h1><br>
16     <a href="https://chat.openai.com/" target="_blank"
17     <a href="https://www.youtube.com/" target="_blank"
18     <a href="https://www.google.com/" target="_blank" c
19     <a href="https://www.facebook.com/" target="_blank"
```



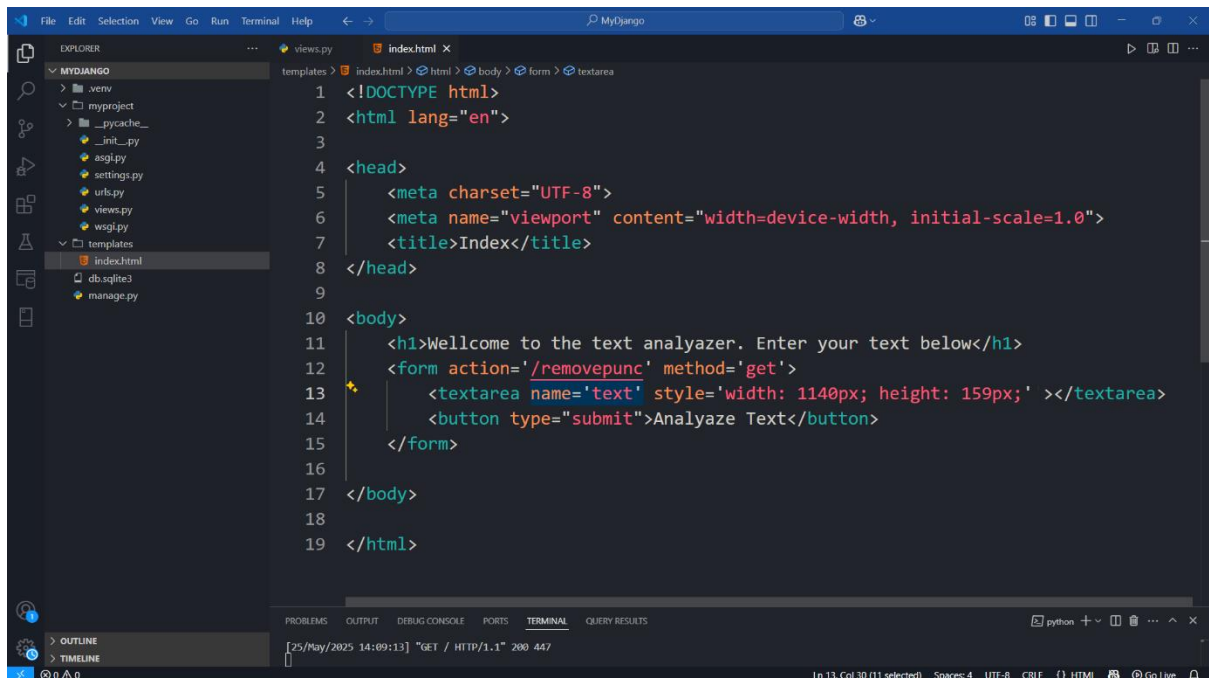
```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Index</title>
8 </head>
9
10 <body>
11     <h1>Wellcome to the text analyzer. Enter your text below</h1>
12     <form action="/removepunc" method="get">
13         <textarea style="width: 1140px; height: 159px;" ></textarea>
14         <button type="submit">Analyaze Text</button>
15     </form>
16
17 </body>
18
19 </html>
```

## 2) Our Output is

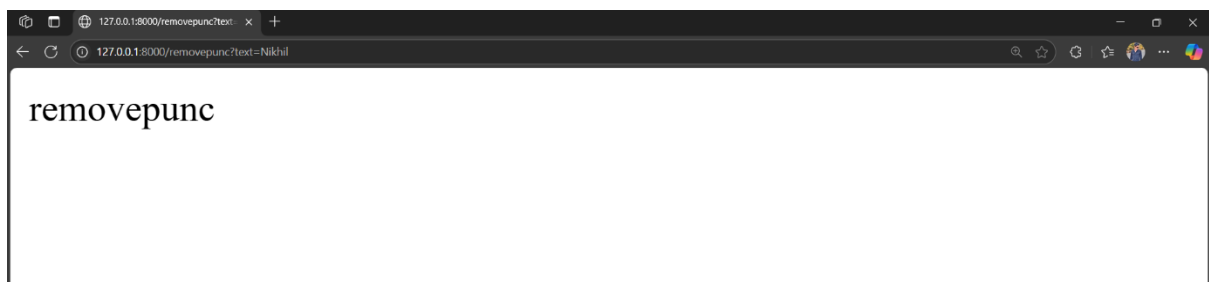


3) After clicking the Analyzer Text Button it go to the /removepunc Route....

4) Give a name to our textarea



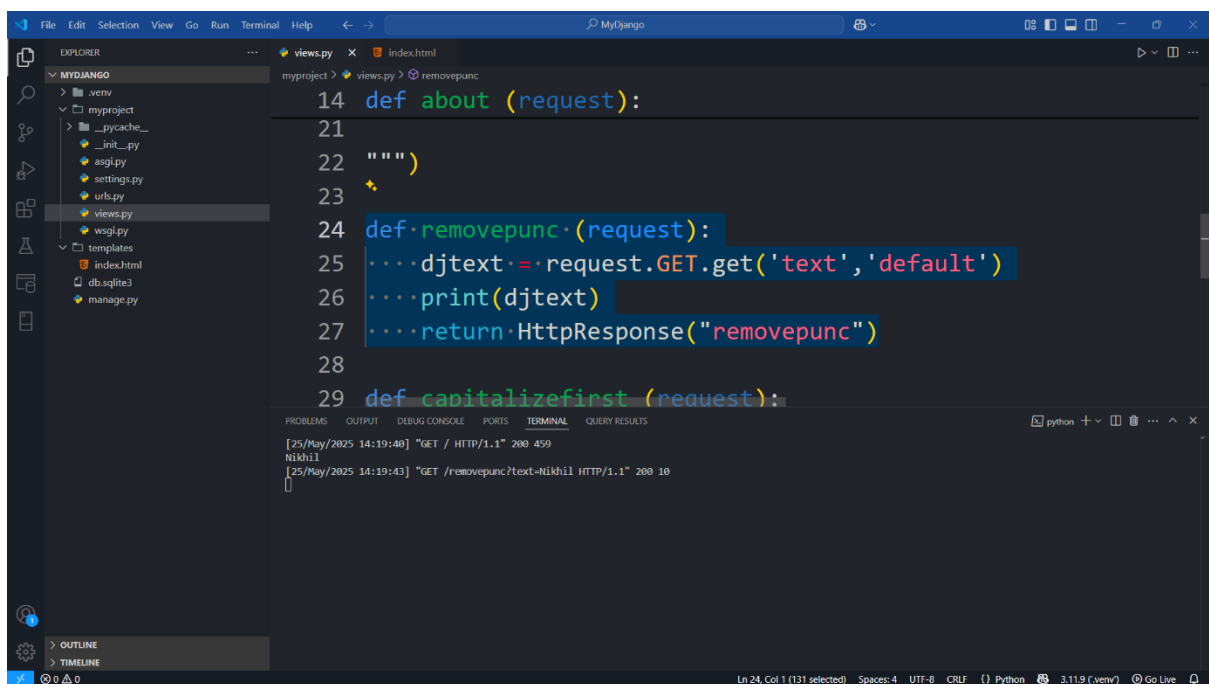
5) We see that that text pass to our server url



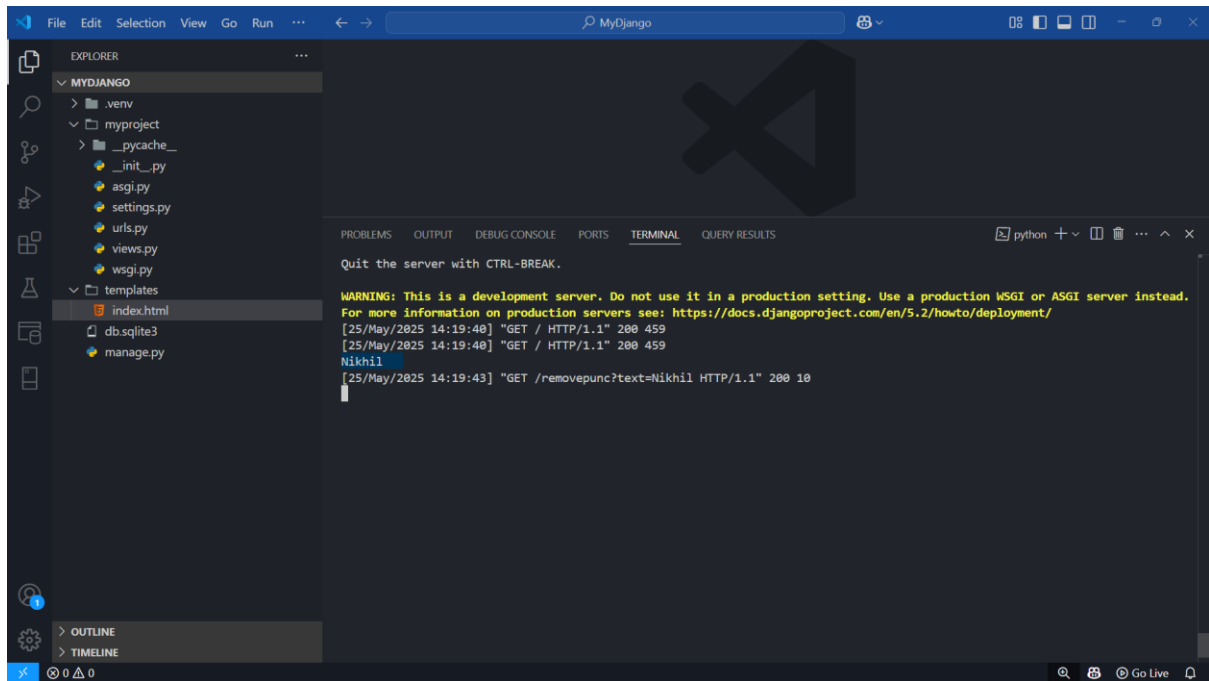
6) We can access this data into /removepunc Route..

Let's see How....

To Access This text Go to view.py > go to removepunc function and write the code.....



7) After Writing a code run Server Write some text And click Button ,  
**Then What Happen.....**  
That Text Print into the Terminal



## Now We Edit our code

1) For Our Project We need only one end point analyze

```
22 #endpoints
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('', views.index, name='index'),
26     path('about', views.about, name='about'),
27     path('analyze', views.analyze, name='analyze'),
28     # path('capitalizefirst', views.capitalizefirst, name='capitalizefirst'),
29     # path('newlineremove', views.newlineremove, name='newlineremove'),
30     # path('spacerremove', views.spacerremove, name='spacerremove'),
31     # path('charcount', views.charcount, name='charcount')
32 ]
33
```

2) We also create a analyze route and comment out other routes

```
def analyze (request):  
    djtext = request.GET.get('text','default')  
    print(djtext)  
    return HttpResponse("analyze")  
  
#Exercise Route
```

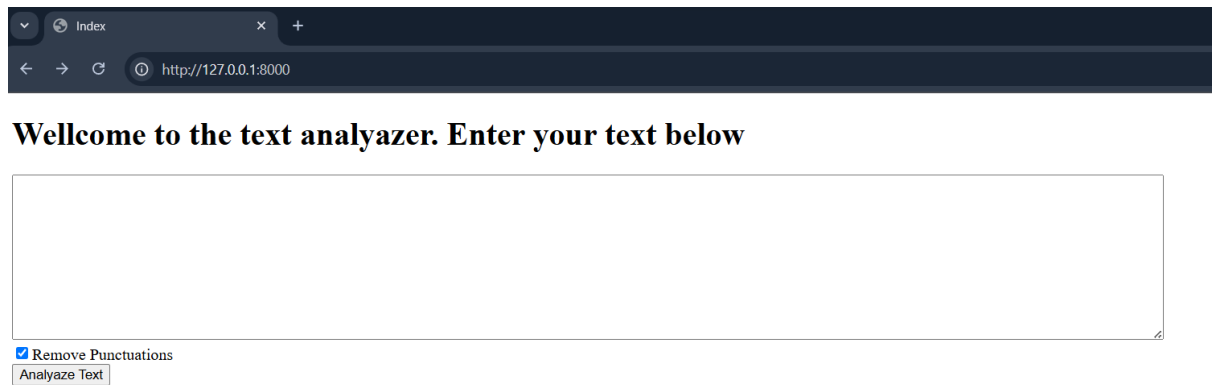
3) Our Index route is like this

```
def index (request):  
    # return HttpResponse("Hello Nikhil")  
    #also  
    #return HttpResponse("<h1>Hello Nikhil</h1>")\  
    #params = {'name': 'Nikhil' , 'place': 'Mars'}  
    return render(request , 'analyze.html' ) #,params)  
  
def analyze (request):  
    djtext = request.GET.get('text','default')  
    print(djtext)  
    return HttpResponse("analyze")  
  
#Exercise Route  
def about (request):  
    return HttpResponse("""<h1>Personal Navigator</h1><br>  
    <a href="https://chat.openai.com/" target="_blank" class="chat-link">Try ChatGPT Now</a><br>  
    <a href="https://www.youtube.com/" target="_blank" class="chat-link">Visit YouTube</a><br>  
    <a href="https://www.google.com/" target="_blank" class="chat-link">Visit Google</a><br>  
    <a href="https://www.facebook.com/" target="_blank" class="chat-link">Visit Facebook</a><br>  
    <a href="https://www.instagram.com/" target="_blank" class="chat-link">Visit Instagram</a>  
    """)
```

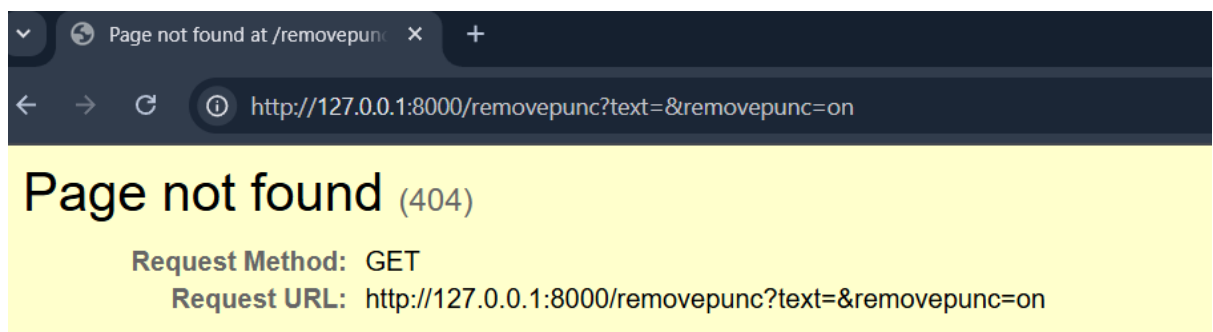
4) In Html filr add checkbox , and form Submit to analyze Route

```
EXPLORER  
... manage.py urls.py views.py analyze.html  
MYDJ...  
> .venv  
  myproject  
    __pycache__  
      _init_.py  
      asgi.py  
      settings.py  
      urls.py  
      views.py  
      wsgi.py  
  templates  
    analyze.html  
    db.sqlite3  
    manage.py  
  
templates > analyze.html > html  
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5     <meta charset="UTF-8">  
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7     <title>Index</title>  
8 </head>  
9  
10 <body>  
11     <h1>Wellcome to the text analyzer. Enter your text below</h1>  
12     <form action="/analyze" method='get'>  
13         <textarea name='text' style='width: 1140px; height: 159px;' ></textarea><br>  
14         <input type="checkbox" name="removepunc">Remove Punctuations <br>  
15         <button type="submit">Analyze Text</button>  
16     </form>  
17  
18 </body>  
19  
20 </html>
```

## 5) Now Run Code and See

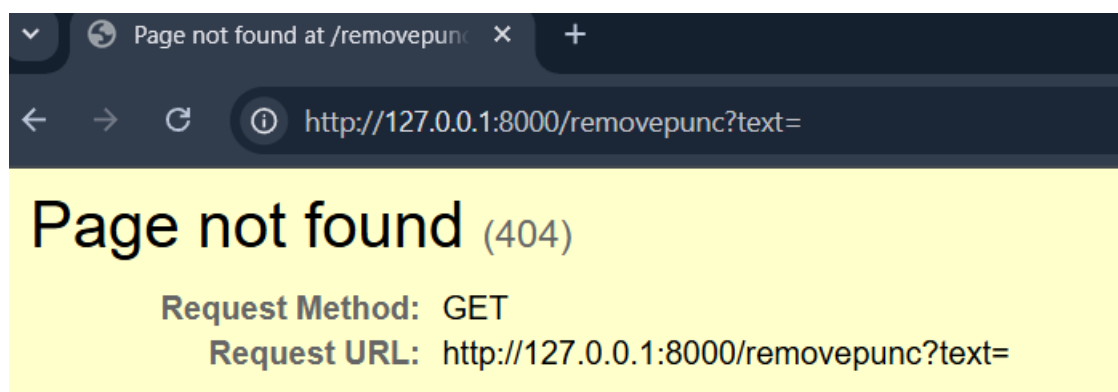


If We check the checkbox and Click analayze button then see in url



removepunc = on

6) removepunc is name of Checkbox and if we not check and click on Button then see



Now ,

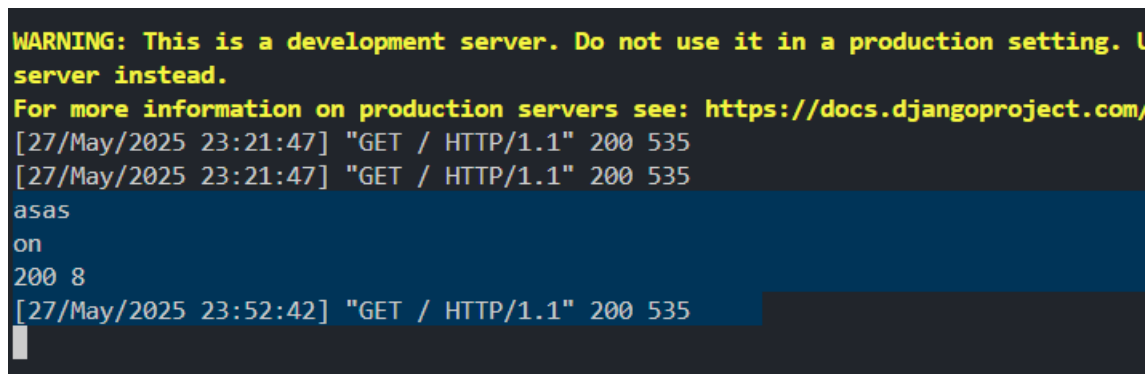
1) we go to analayze route



```
views.py 15
wsgi.py 16 def analyze (request):
templates 17     djtext = request.GET.get('text','default')
analyze.html 18     removepunc = request.GET.get('removepunc','default')
db.sqlite3 19     print(djtext)
manage.py 20     print(removepunc)
21     return HttpResponse("analyze")
```

2) Write Something in box and check box and click button

3) We see that analayse is print and in Terminal



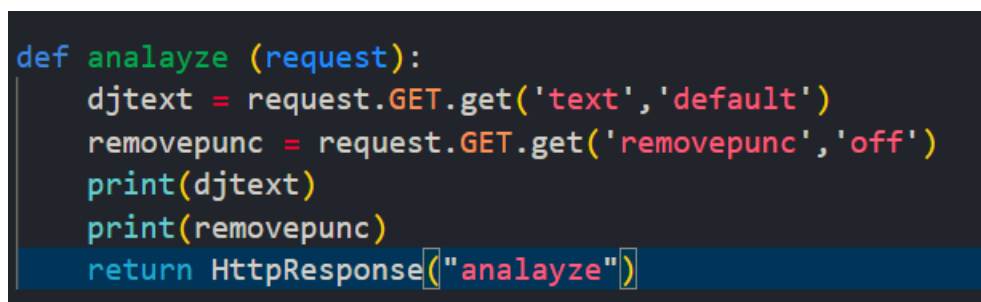
```
WARNING: This is a development server. Do not use it in a production setting. Use
server instead.
For more information on production servers see: https://docs.djangoproject.com/
[27/May/2025 23:21:47] "GET / HTTP/1.1" 200 535
[27/May/2025 23:21:47] "GET / HTTP/1.1" 200 535
asas
on
200 8
[27/May/2025 23:52:42] "GET / HTTP/1.1" 200 535
```

This Print in Terminal.

4) If We not Check the checkbox then it print  
Default and off in Terminal..

So We take Default value is off , when removepunc value id  
on then it is on otherwise it is off.

5) Take Default Value off



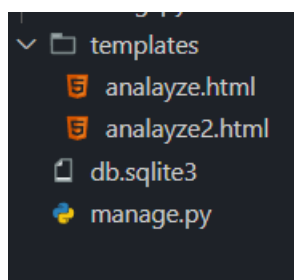
```
def analyze (request):
    djtext = request.GET.get('text','default')
    removepunc = request.GET.get('removepunc','off')
    print(djtext)
    print(removepunc)
    return HttpResponse("analyze")
```



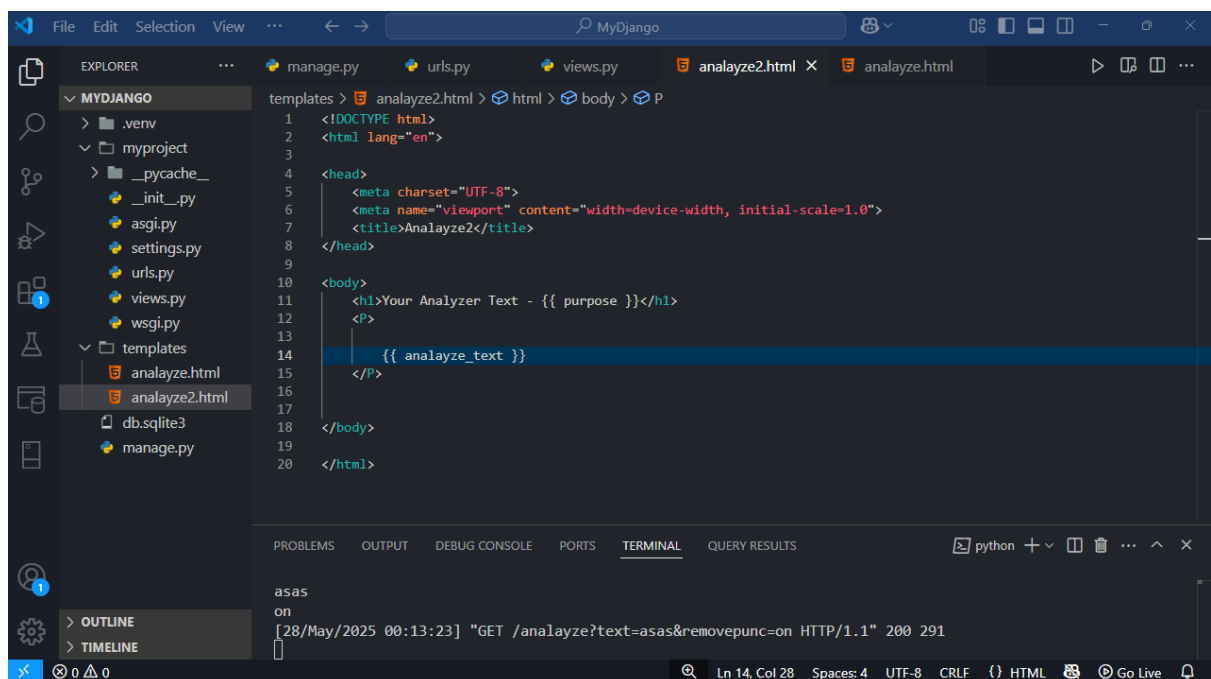
6) Now in analyze we render html file

```
def analyze (request):
    djtext = request.GET.get('text','default')
    removepunc = request.GET.get('removepunc','off')
    print(djtext)
    print(removepunc)
    analyze = djtext
    params = {'purpose':'Remove Punctuations' , 'analyze_text':analyze}
    # return HttpResponse("analyze")
    return render(request , 'analyze2.html' , params)
```

7) Create analyze2.html



8) And this looks like this

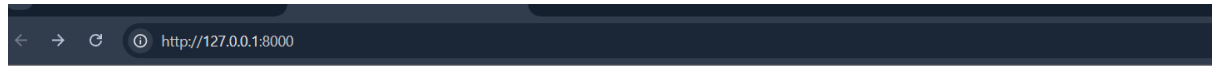


9) Output is

We enter anything in box and checkbox tick or not it print on analyze2.html file as it is.....

10) Remember the, In params there is No ' ' in analyze variable

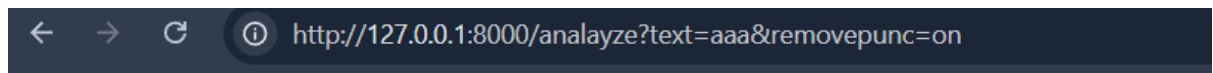
11) Output is



**Wellcome to the text analyazer. Enter your text below**

aaa

☒ Remove Punctuations  
Analyze Text

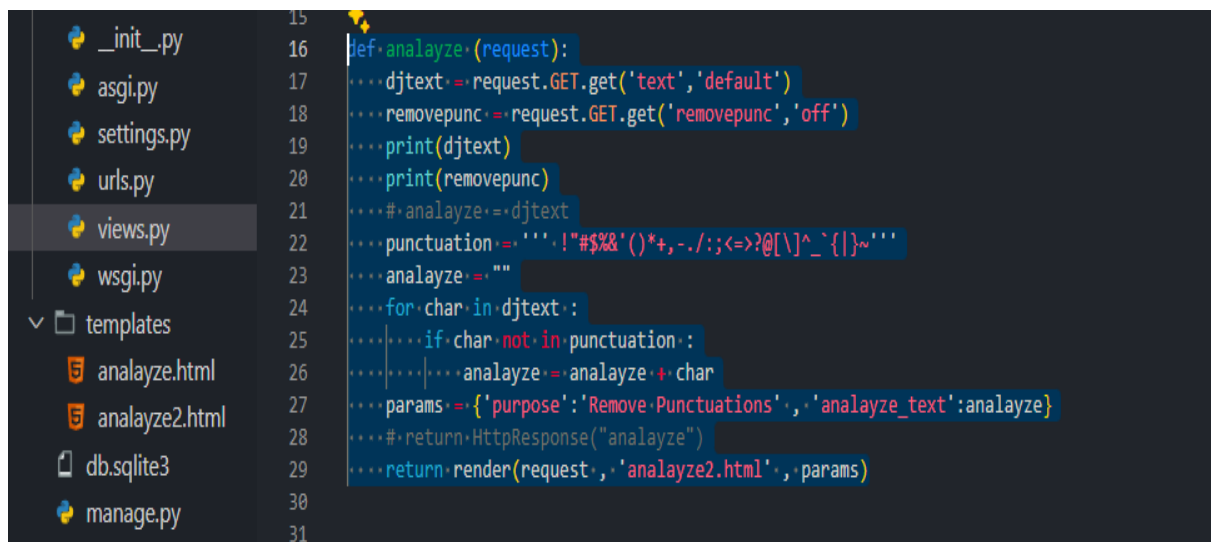


## Your Analyzer Text - Remove Punctuations

aaa

12) Now We Remove the Punctuations Using Pyhton Logic

Update our analyze Route



- 13) Write Anything with punctuation in boc checkboc check and click button
- 14) We see Correct Output.....
- 15) But our check box is not work Properly

What's why we edit our analayze Route

```
15
16 def analayze (request):
17     djtext = request.GET.get('text','default')
18     removepunc = request.GET.get('removepunc','off')
19     print(djtext)
20     print(removepunc)
21
22     if removepunc == 'on':
23
24         # analayze = djtext
25         punctuation = '' !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~''''
26         analayze = ""
27         for char in djtext :
28             if char not in punctuation :
29                 analayze = analayze + char
30         params = {'purpose':'Remove Punctuations' , 'analayze_text':analayze}
31         # return HttpResponse("analayze")
32         return render(request , 'analayze2.html' , params)
33     else :
34         return HttpResponse("Error , Please check the Box")
35
```

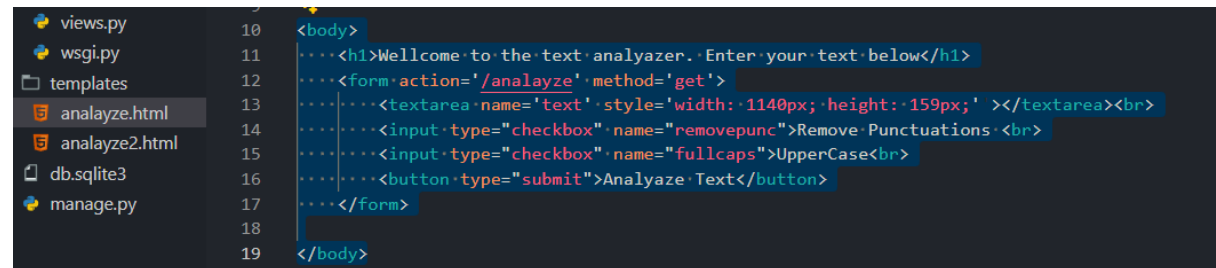
if removepunc = 'on' then only it work correctly otherwise it prints the "Error , Please check the Box"

And

**DONE**

## Similarly We add For Uppercase

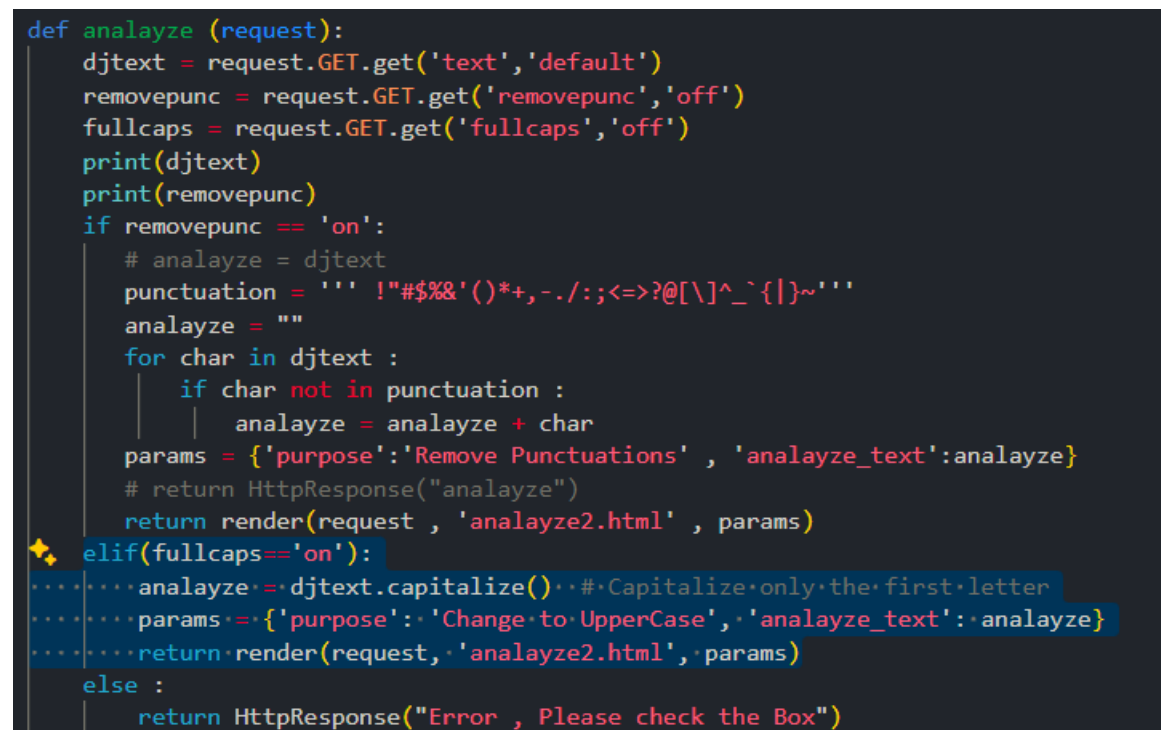
1) We go to analayze.html which is index file and add 1 more checkbox



```
10 <body>
11 <h1>Wellcome to the text analyzer. Enter your text below</h1>
12 <form action='/analayze' method='get'>
13 <input type='text' style='width:1140px; height:159px;' >/></textarea><br>
14 <input type="checkbox" name="removepunc">Remove Punctuations <br>
15 <input type="checkbox" name="fullcaps">UpperCase<br>
16 <button type="submit">Analyze Text</button>
17 </form>
18
19 </body>
```

2) No change in analayze2.html file

3) Change our Route like this



```
def analyze (request):
    djtext = request.GET.get('text','default')
    removepunc = request.GET.get('removepunc','off')
    fullcaps = request.GET.get('fullcaps','off')
    print(djtext)
    print(removepunc)
    if removepunc == 'on':
        # analayze = djtext
        punctuation = '!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~''''
        analayze = ""
        for char in djtext :
            if char not in punctuation :
                analayze = analayze + char
        params = {'purpose':'Remove Punctuations' , 'analayze_text':analayze}
        # return HttpResponse("analayze")
        return render(request , 'analayze2.html' , params)
    elif(fullcaps=='on'):
        analayze = djtext.capitalize() # Capitalize only the first letter
        params = {'purpose':'Change to UpperCase' , 'analayze_text':analayze}
        return render(request , 'analayze2.html' , params)
    else :
        return HttpResponse("Error , Please check the Box")
```

4) Add on elif statement and Logic for UpperCase...

5) Also fillcaps define upper.

**DONE**

Similarly We do for Remove New Line..

```

16
17 def analyze (request):
18     djtext = request.GET.get('text','default')
19     removepunc = request.GET.get('removepunc','off')
20     fullcaps = request.GET.get('fullcaps','off')
21     newlinere mover = request.GET.get('newlinere mover', 'off')
22     # print(djtext)
23     # print(removepunc)
24     if removepunc == 'on':
25         # analyze = djtext
26         punctuation = ' !"#%&'()*+,-./:;<=>?@[\\]^_`{|}~''''
27         analyze = ""
28         for char in djtext :
29             if char not in punctuation :
30                 analyze = analyze + char
31         params = {'purpose': 'Remove Punctuations' , 'analyze_text': analyze}
32         # return HttpResponse("analyze")
33         return render(request , 'analyze2.html' , params)
34     elif(fullcaps=='on'):
35         analyze = djtext.capitalize() # Capitalize only the first letter
36         params = {'purpose': 'Change to UpperCase', 'analyze_text': analyze}
37         return render(request, 'analyze2.html', params)
38     elif(newlinere mover=='on'):
39         analyze=""
40         for char in djtext :
41             if char != "\n":
42                 analyze = analyze + char.upper()
43
44         params = {'purpose': 'Remove New Lines', 'analyze_text': analyze}
45         return render(request, 'analyze2.html', params)
46     else :
47         return HttpResponse("Error , Please check the Box")

```

## Create This

- Create , Extra Space Remover
- Count Characters

```

elif(extraspace remover=="on"):
    analyzed = ""
    for index, char in enumerate(djtext):
        if not(djtext[index] == " " and djtext[index+1]==" "):
            analyzed = analyzed + char

    params = {'purpose': 'Removed NewLines', 'analyzed_text': analyzed}
    # Analyze the text
    return render(request, 'analyze.html', params)

```

**DONE....**

## Exercise 2: - Revamping TextUtils

Wellcome to the text analyzer. Enter your text below

- ☐ Remove Punctuations
  - ☐ UpperCase
  - ☐ New Line Remover
  - ☐ Extra Space Remover
- Analyze Text

In this output which option we are checked that only perform, if two option checked then that two only chek.....

**Solution:-** Update your route

```
59 def analyze(request):
60     djtext = request.GET.get('text', 'default')
61     removepunc = request.GET.get('removepunc', 'off')
62     fullcaps = request.GET.get('fullcaps', 'off')
63     newlineremover = request.GET.get('newlineremover', 'off')
64     extraspaceremover = request.GET.get('extraspaceremover', 'off')
65
66     operations_done = []
67     analyzed = djtext
68
69     if removepunc == 'on':
70         punctuation = '!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~'
71         analyzed = ''.join(char for char in analyzed if char not in punctuation)
72         operations_done.append('Removed Punctuations')
73
74     if fullcaps == 'on':
75         analyzed = analyzed.upper()
76         operations_done.append('Changed to Uppercase')
77
78     if newlineremover == 'on':
79         analyzed = analyzed.replace('\n', '.').replace('\r', '.')
80         operations_done.append('Removed New Lines')
81
82     if extraspaceremover == 'on':
83         analyzed = ''.join(analyzed.split())
84         operations_done.append('Removed Extra Spaces')
85
86     if not operations_done:
87         return HttpResponse("Error: Please select at least one operation.")
88
89     params = {
90         'purpose': ', '.join(operations_done),
91         'analyze_text': analyzed
92     }
93
94     return render(request, 'analyze2.html', params)
```

```
<body>
  <h1>Your Analyzer Text - {{ purpose }}</h1>
  <h2>Result:</h2>
  <P>
    {{ analyze_text }} <br>
    <a href="/">Go Back</a>
  </P>
</body>
```

❖ We See that our Next line Remover Work Automatically

It is Because of our html , html not print New line when we give input as New Line

That's why we use **<pre>** tag

```
<body>
  <div class="header">
    <h1>TextUtils - Text Analysis Result</h1>
  </div>/.header

  <div class="container">
    <h1>Text Analysis Summary</h1>
    <p>{{ purpose }}</p>

    <h2>Analyzed Output</h2>
    <p>
      <pre>{{ analyze_text }}</pre>
    </p>

    <a href="/">Go Back</a>
  </div>/.container
</body>
```

## Django CSRF Tokens & Post Request

- CSRF Means Cross Side Request Forgery.

It conforms that , Jo Request aa rhi hai vo apki hi hai...

Do we use `{% csrf_token %}` this with forms

```
<div class="container">
  <h2>Welcome to the text analyzer. Enter your text below:</h2>
  <form action='/analyze' method='post'> {% csrf_token %}|
    <textarea name='text' placeholder='Enter your text here... '></textarea><br>
    <input type="checkbox" name="removepunc"> Remove Punctuations<br>
    <input type="checkbox" name="fullcaps"> UpperCase<br>
    <input type="checkbox" name="newlineremover"> New Line Remover<br>
    <input type="checkbox" name="extraspaceremover"> Extra Space Remover<br>

    <button type="submit">Analyze Text</button>
  </form>
</div> /.container
```

- Also we Use POST

Because all the data which we can pass in text box it transfer through our url that's why our url some time not work properly so use POST request...

```
def analyze(request):
    ... djtext = request.POST.get('text', 'default')
    ... removepunc = request.POST.get('removepunc', 'off')
    ... fullcaps = request.POST.get('fullcaps', 'off')
    ... newlineremover = request.POST.get('newlineremover', 'off')
    ... extraspaceremover = request.POST.get('extraspaceremover', 'off')|
```

```
<div class="container">
  <h2>Welcome to the text analyzer. Enter your text below:</h2>
  <form action='/analyze' method='post'> {% csrf_token %}
    <textarea name='text' placeholder='Enter your text here... '></textarea><br>
    <input type="checkbox" name="removepunc"> Remove Punctuations<br>
    <input type="checkbox" name="fullcaps"> UpperCase<br>
    <input type="checkbox" name="newlineremover"> New Line Remover<br>
    <input type="checkbox" name="extraspaceremover"> Extra Space Remover<br>

    <button type="submit">Analyze Text</button>
  </form>
</div> /.container
```

Here, We Finish our Small Web app Completely.....