Third Practical Exercise

Telecommunication Software

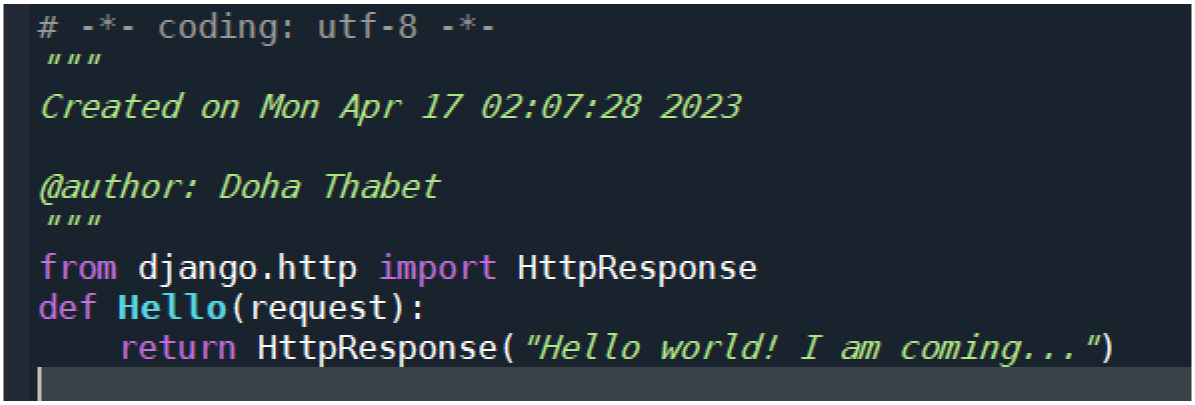
Doha Thabet Hassan Mahmoud
ID: 230AMB014

1. **Django's Hello World program:**

- Create a new Django project by running the following command in the terminal:

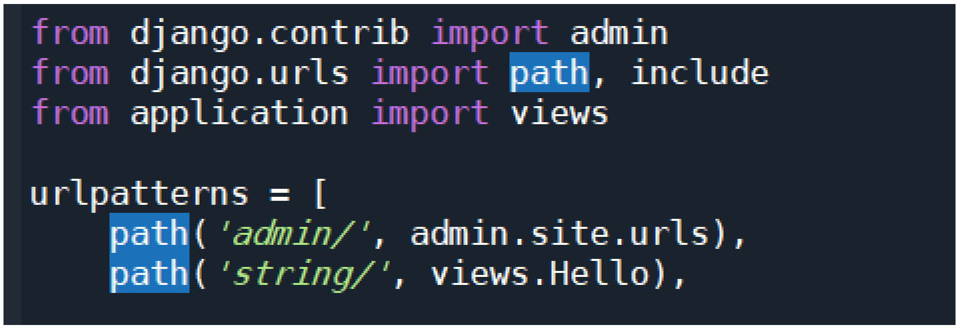
django-admin startproject <mysite>

- Create the first application.
- Add function to **views.py** to return a string:



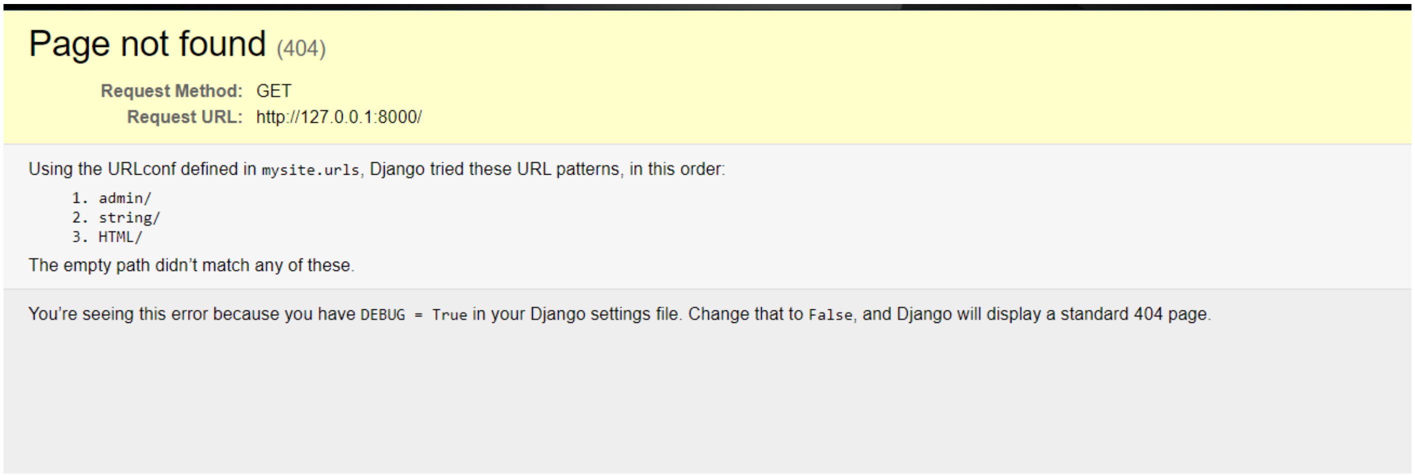
```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 17 02:07:28 2023

@author: Doha Thabet
"""
from django.http import HttpResponse
def Hello(request):
    return HttpResponse("Hello world! I am coming...")
```

- Add the URL patterns for the app in the **urls.py**:



```
from django.contrib import admin
from django.urls import path, include
from application import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('string/', views.Hello),
```

- The results:



Page not found (404)

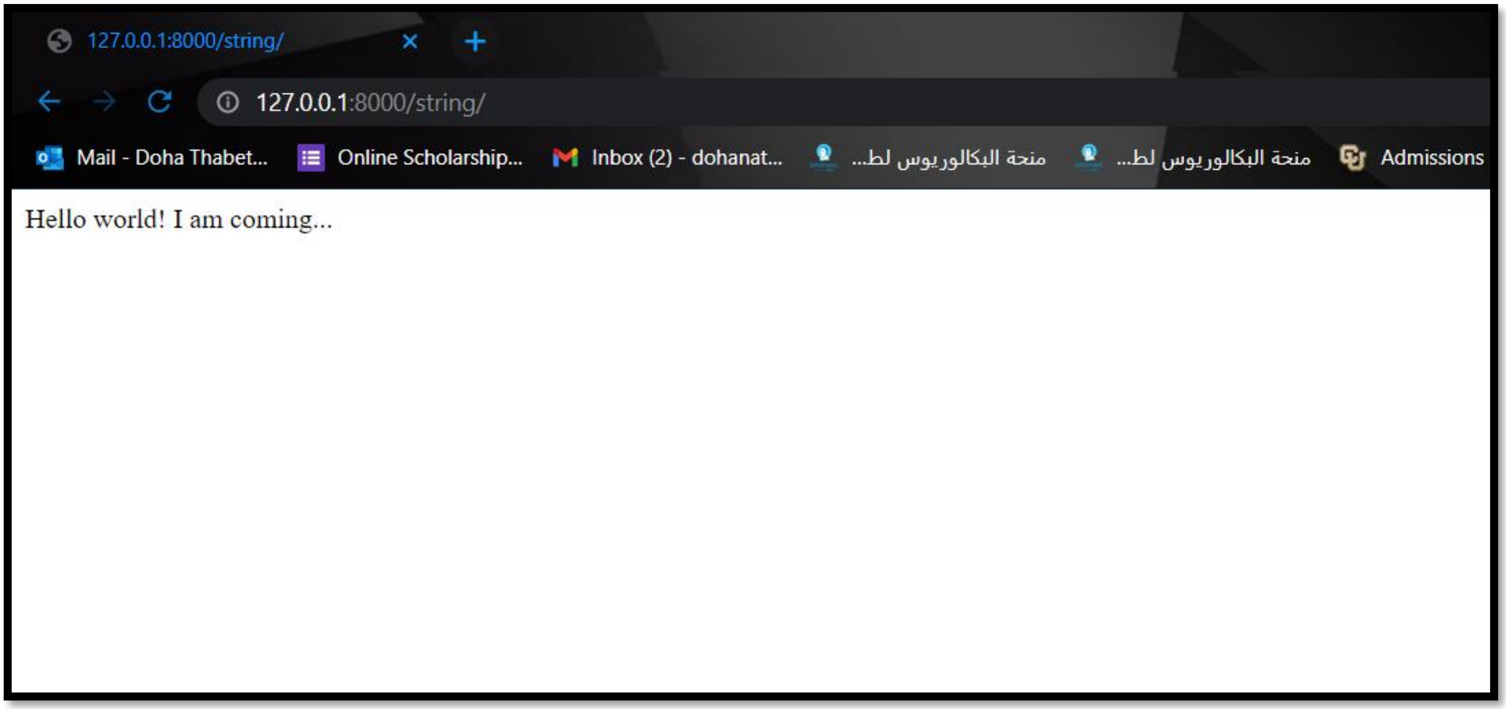
Request Method: GET
Request URL: http://127.0.0.1:8000/

Using the URLconf defined in `mysite.urls`, Django tried these URL patterns, in this order:

```
1. admin/
2. string/
3. HTML/
```

The empty path didn't match any of these.

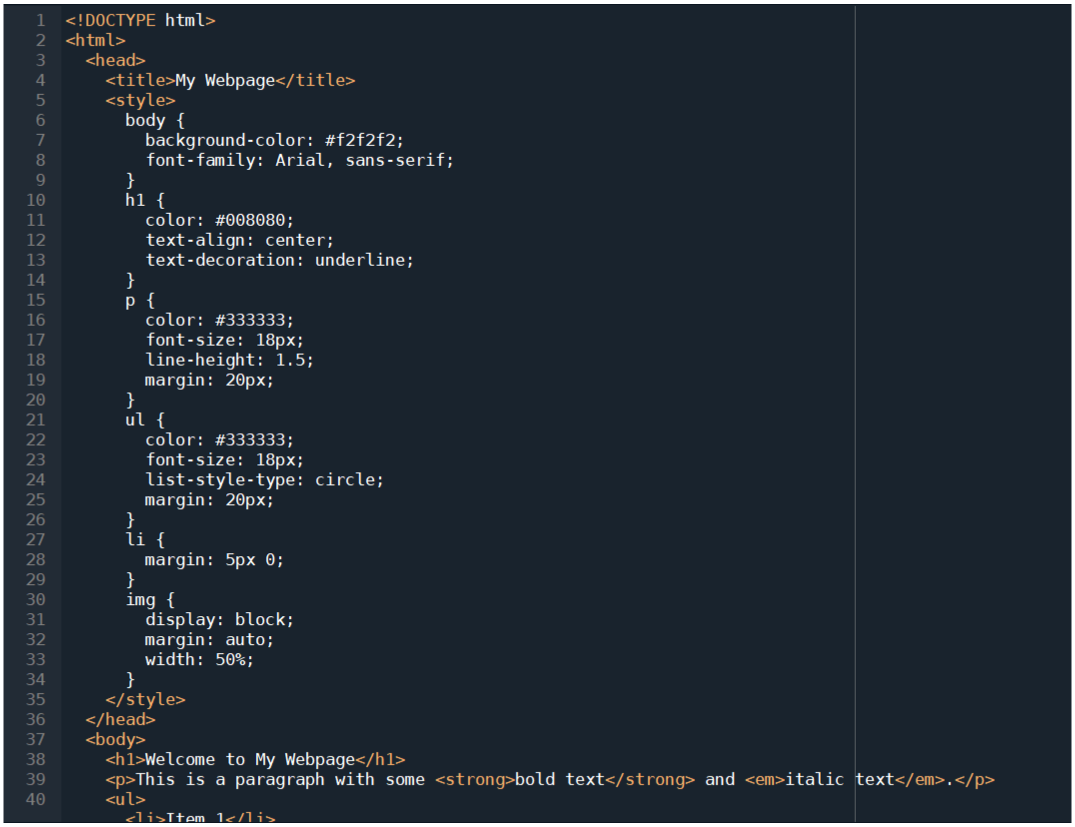
You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



Hello world! I am coming...

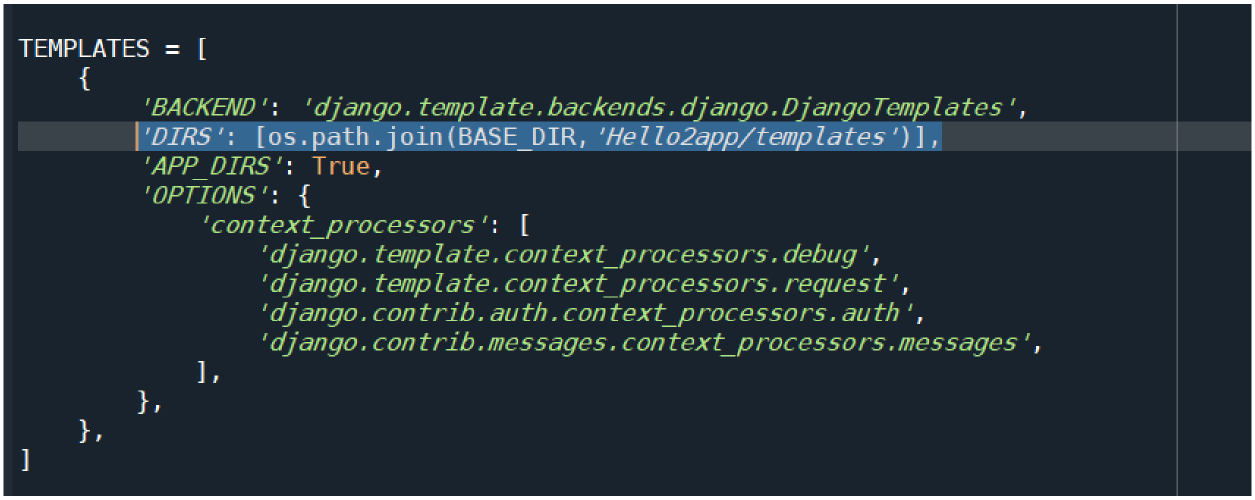
2. Improve example 1 and return an HTML page, not a string :

- Add the HTML code to the application Templates (hello.html)



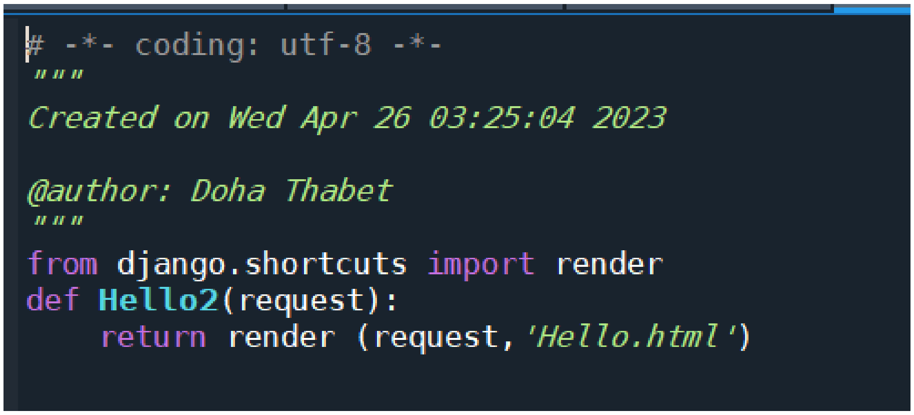
```
<!DOCTYPE html>
<html>
  <head>
    <title>My Webpage</title>
    <style>
      body {
        background-color: #f2f2f2;
        font-family: Arial, sans-serif;
      }
      h1 {
        color: #008080;
        text-align: center;
        text-decoration: underline;
      }
      p {
        color: #333333;
        font-size: 18px;
        line-height: 1.5;
        margin: 20px;
      }
      ul {
        color: #333333;
        font-size: 18px;
        list-style-type: circle;
        margin: 20px;
      }
      li {
        margin: 5px 0;
      }
      img {
        display: block;
        margin: auto;
        width: 50%;
      }
    </style>
  </head>
  <body>
    <h1>Welcome to My Webpage</h1>
    <p>This is a paragraph with some <strong>bold text</strong> and <em>italic text</em>.</p>
    <ul>
      <li>Item 1</li>
```

- Modify the project **settings.py** file to add the application templates:



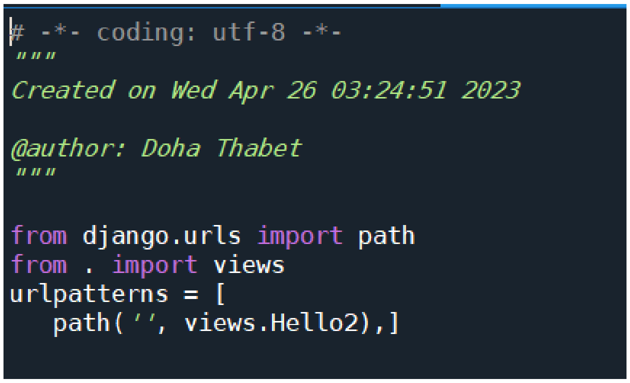
```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'Hello2app/templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

- Add function to **views.py** file to return an HTML Page:



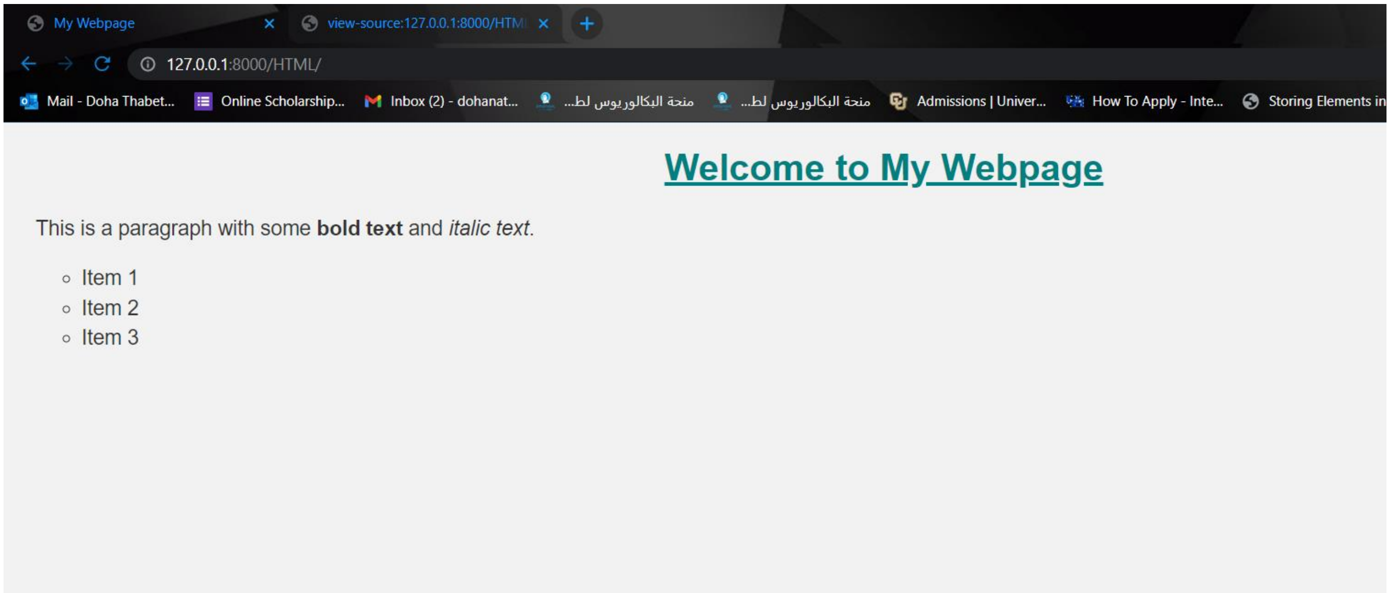
```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 26 03:25:04 2023

@author: Doha Thabet
"""
from django.shortcuts import render
def Hello2(request):
    return render (request, 'Hello.html')
```

- Add the URL patterns for the app in the **urls.py:**



```
# -*- coding: utf-8 -*-
"""
Created on Wed Apr 26 03:24:51 2023

@author: Doha Thabet
"""

from django.urls import path
from . import views
urlpatterns = [
   path('', views.Hello2),]
```

- Results:



Welcome to My Webpage

This is a paragraph with some **bold text** and *italic text*.

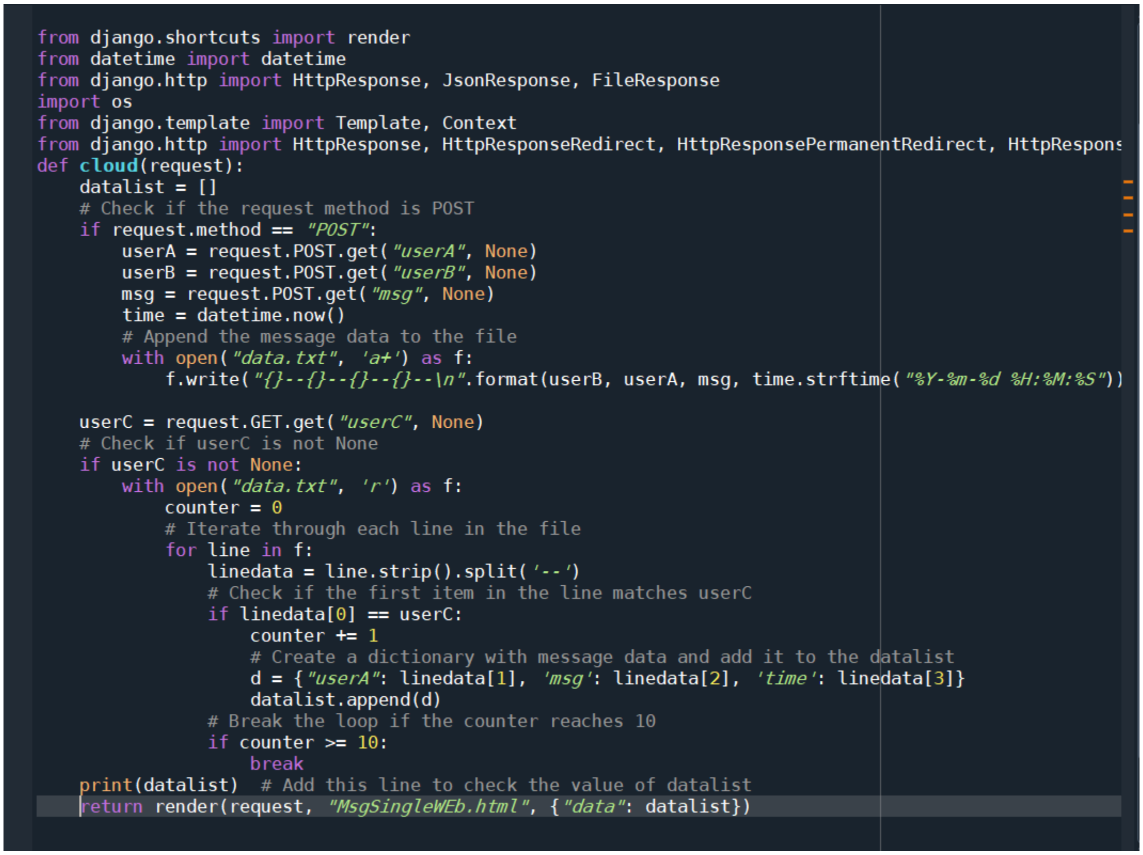
- Item 1
- Item 2
- Item 3

3. Cloud message board:

- **Create a new Django project by running the following command in the terminal:**

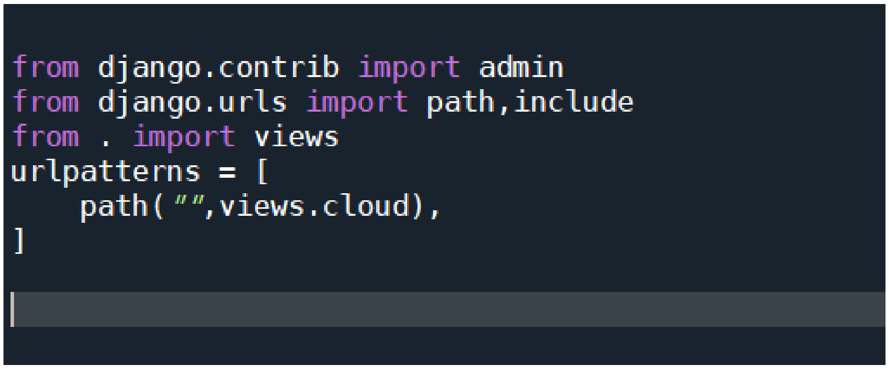
django-admin startproject <C_Message>

- Create **msgapp** application.
- Add function to **views.py** to submit and get the message:



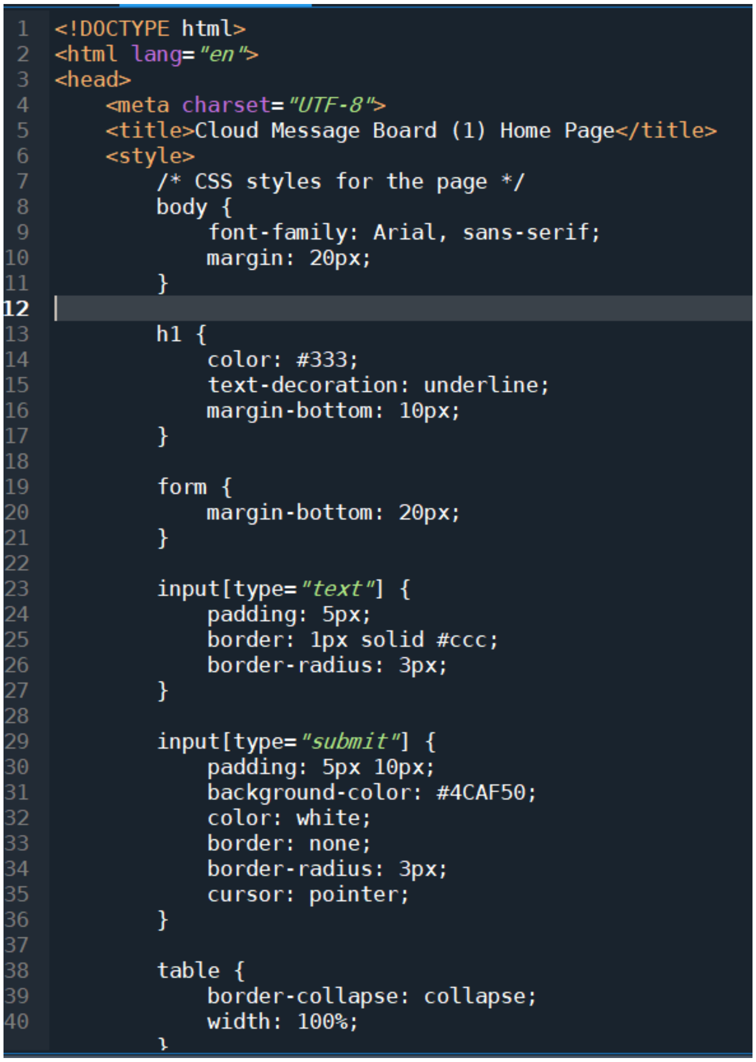
```python
from django.shortcuts import render
from datetime import datetime
from django.http import HttpResponse, JsonResponse, FileResponse
import os
from django.template import Template, Context
from django.http import HttpResponse, HttpResponseRedirect, HttpResponsePermanentRedirect, HttpRespons
def cloud(request):
    datalist = []
    # Check if the request method is POST
    if request.method == "POST":
        userA = request.POST.get("userA", None)
        userB = request.POST.get("userB", None)
        msg = request.POST.get("msg", None)
        time = datetime.now()
        # Append the message data to the file
        with open("data.txt", 'a+') as f:
            f.write("{}--{}--{}--{}--\n".format(userB, userA, msg, time.strftime("%Y-%m-%d %H:%M:%S"))

    userC = request.GET.get("userC", None)
    # Check if userC is not None
    if userC is not None:
        with open("data.txt", 'r') as f:
            counter = 0
            # Iterate through each line in the file
            for line in f:
                linedata = line.strip().split('--')
                # Check if the first item in the line matches userC
                if linedata[0] == userC:
                    counter += 1
                    # Create a dictionary with message data and add it to the datalist
                    d = {"userA": linedata[1], 'msg': linedata[2], 'time': linedata[3]}
                    datalist.append(d)
                # Break the loop if the counter reaches 10
                if counter >= 10:
                    break
    print(datalist)  # Add this line to check the value of datalist
    return render(request, "MsgSingleWEb.html", {"data": datalist})
```

- Add the URL patterns for the app in the **urls.py:**

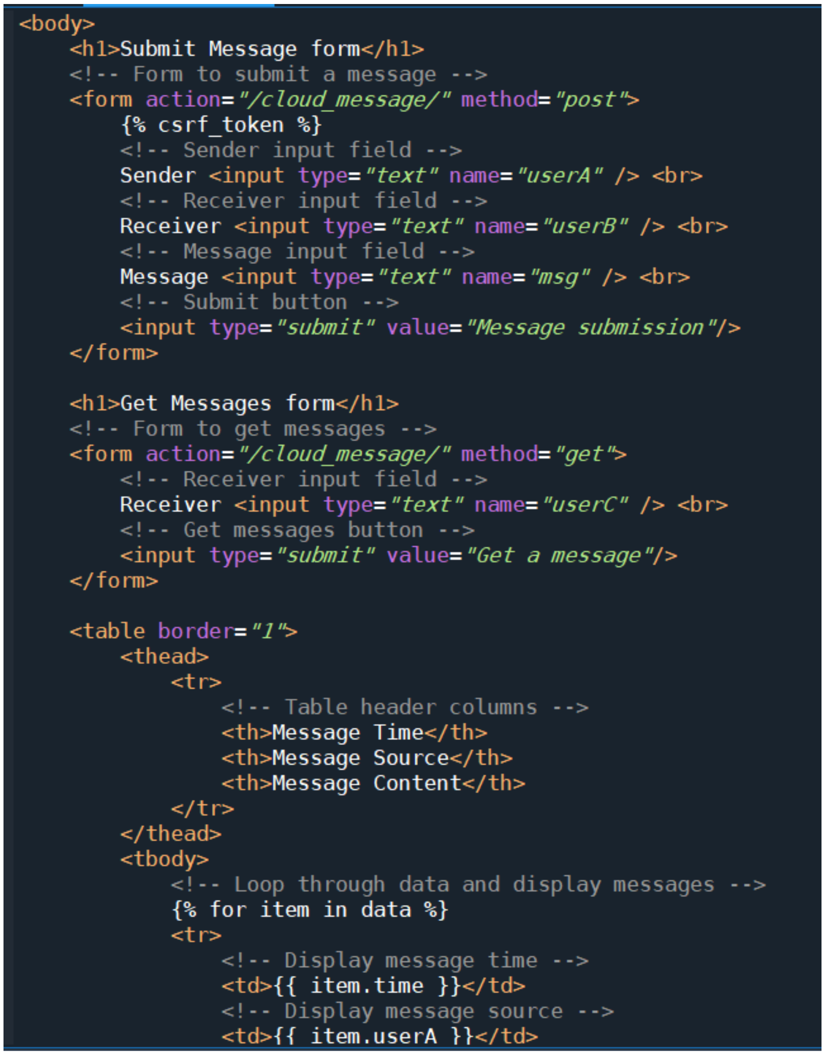


```python
from django.contrib import admin
from django.urls import path,include
from . import views
urlpatterns = [
    path("",views.cloud),
]

```

- Add the HTML code to the **msgapp** Templates (MsgSingleWeb.html)

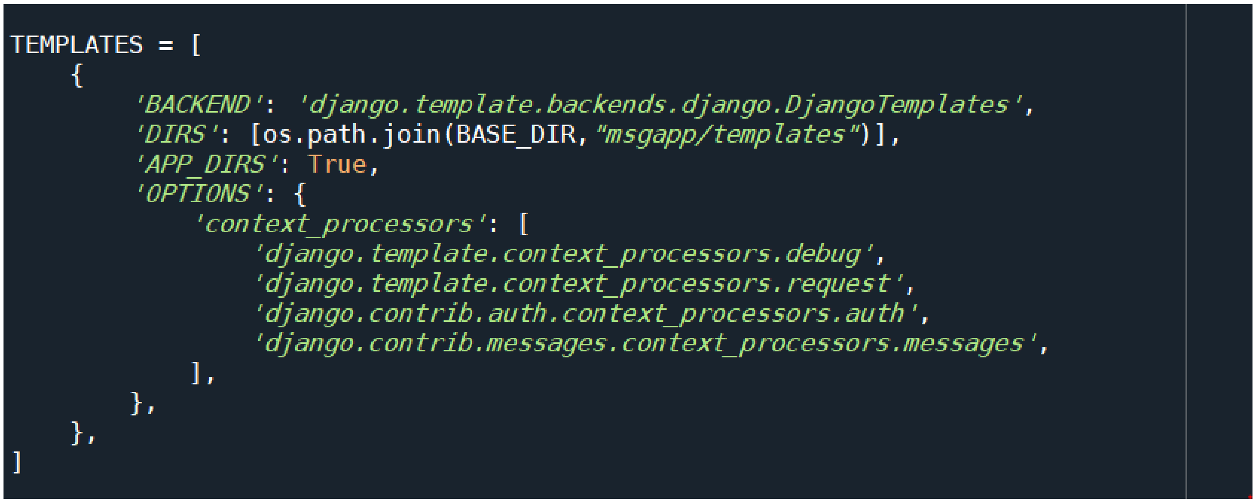


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cloud Message Board (1) Home Page</title>
    <style>
        /* CSS styles for the page */
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }

        h1 {
            color: #333;
            text-decoration: underline;
            margin-bottom: 10px;
        }

        form {
            margin-bottom: 20px;
        }

        input[type="text"] {
            padding: 5px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        input[type="submit"] {
            padding: 5px 10px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }

        table {
            border-collapse: collapse;
            width: 100%;
        }
```



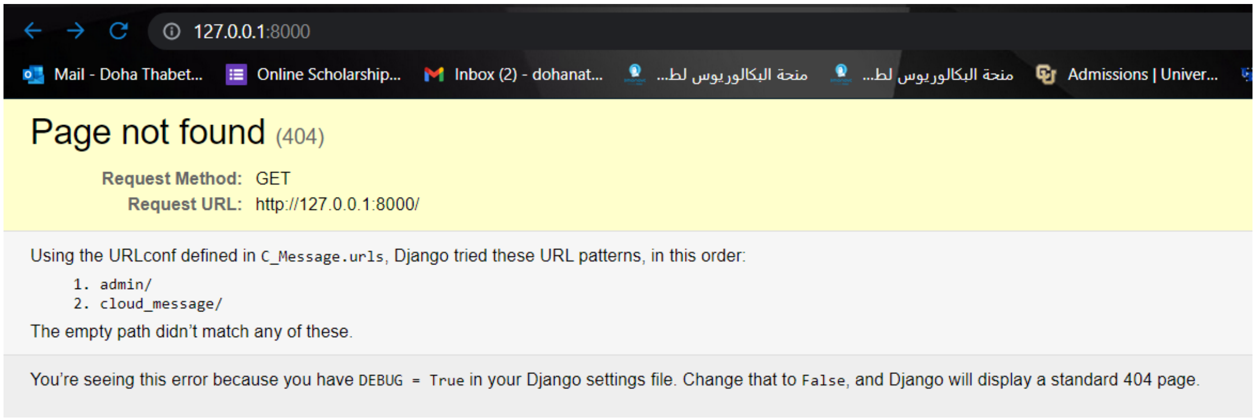
```html
<body>
    <h1>Submit Message form</h1>
    <!-- Form to submit a message -->
    <form action="/cloud_message/" method="post">
        {% csrf_token %}
        <!-- Sender input field -->
        Sender <input type="text" name="userA" /> <br>
        <!-- Receiver input field -->
        Receiver <input type="text" name="userB" /> <br>
        <!-- Message input field -->
        Message <input type="text" name="msg" /> <br>
        <!-- Submit button -->
        <input type="submit" value="Message submission"/>
    </form>

    <h1>Get Messages form</h1>
    <!-- Form to get messages -->
    <form action="/cloud_message/" method="get">
        <!-- Receiver input field -->
        Receiver <input type="text" name="userC" /> <br>
        <!-- Get messages button -->
        <input type="submit" value="Get a message"/>
    </form>

    <table border="1">
        <thead>
            <tr>
                <!-- Table header columns -->
                <th>Message Time</th>
                <th>Message Source</th>
                <th>Message Content</th>
            </tr>
        </thead>
        <tbody>
            <!-- Loop through data and display messages -->
            {% for item in data %}
            <tr>
                <!-- Display message time -->
                <td>{{ item.time }}</td>
                <!-- Display message source -->
                <td>{{ item.userA }}</td>
```

- Modify the project **settings.py** file to add the **msgapp** templates:



```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,"msgapp/templates")],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

- **Results:**



Page not found (404)

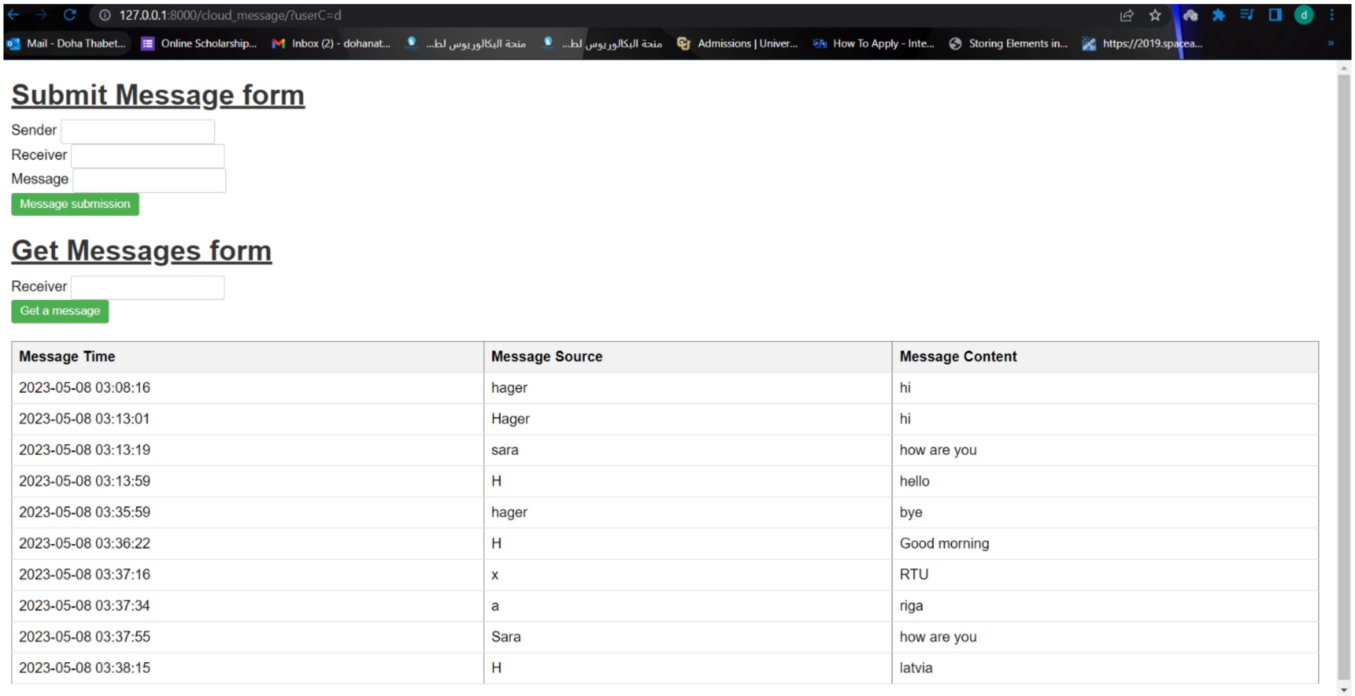
Request Method: GET
Request URL: http://127.0.0.1:8000/

Using the URLconf defined in `C_Message.urls`, Django tried these URL patterns, in this order:

1. `admin/`
2. `cloud_message/`

The empty path didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



Submit Message form

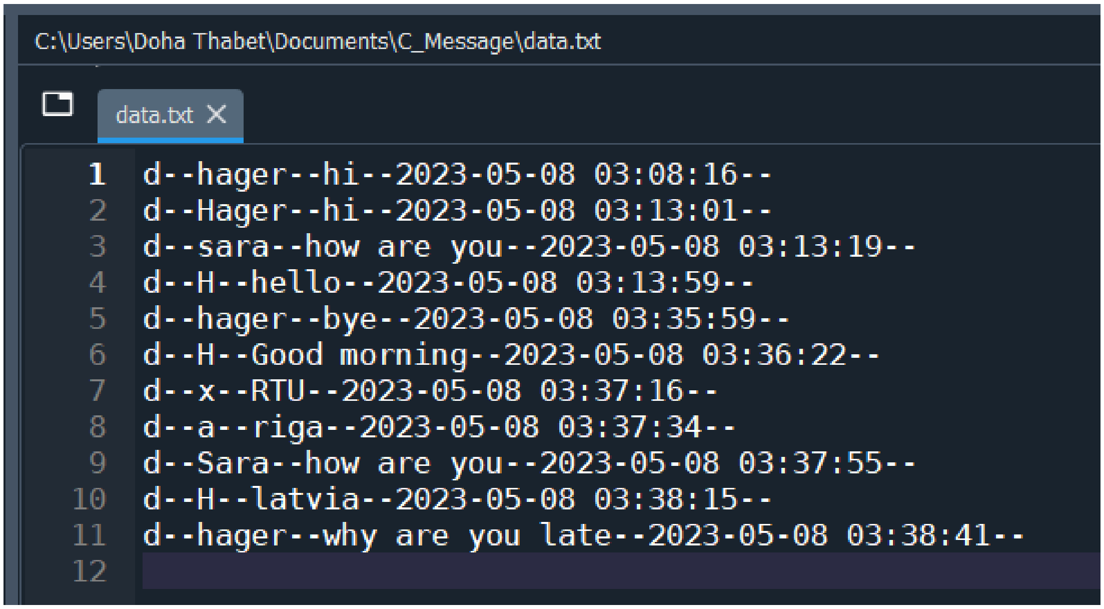
Sender
Receiver
Message
Message submission

Get Messages form

Receiver
Get a message

Message Time	Message Source	Message Content
2023-05-08 03:08:16	hager	hi
2023-05-08 03:13:01	Hager	hi
2023-05-08 03:13:19	sara	how are you
2023-05-08 03:13:59	H	hello
2023-05-08 03:35:59	hager	bye
2023-05-08 03:36:22	H	Good morning
2023-05-08 03:37:16	x	RTU
2023-05-08 03:37:34	a	riga
2023-05-08 03:37:55	Sara	how are you
2023-05-08 03:38:15	H	latvia

- Check the **data.txt** file:



```
d--hager--hi--2023-05-08 03:08:16--
d--Hager--hi--2023-05-08 03:13:01--
d--sara--how are you--2023-05-08 03:13:19--
d--H--hello--2023-05-08 03:13:59--
d--hager--bye--2023-05-08 03:35:59--
d--H--Good morning--2023-05-08 03:36:22--
d--x--RTU--2023-05-08 03:37:16--
d--a--riga--2023-05-08 03:37:34--
d--Sara--how are you--2023-05-08 03:37:55--
d--H--latvia--2023-05-08 03:38:15--
d--hager--why are you late--2023-05-08 03:38:41--

```

4. Django's different response types:

- Try 10 different responses type:

1- HttpResponse

2- HttpResponseBadRequest

3- HttpResponseNotFound

4- HttpResponseServerError

5- HttpResponseRedirect

6- HttpResponsePermanentRedirect

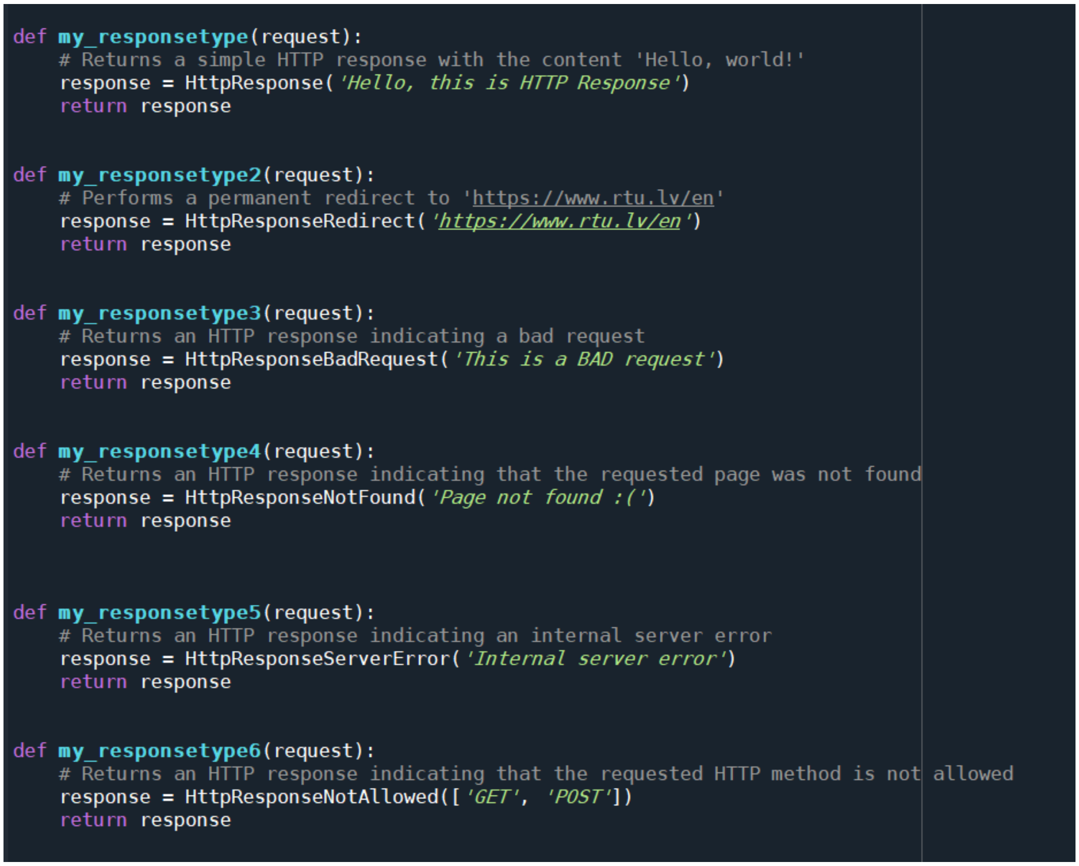
7- HttpResponseNotAllowed

8- JsonResponse

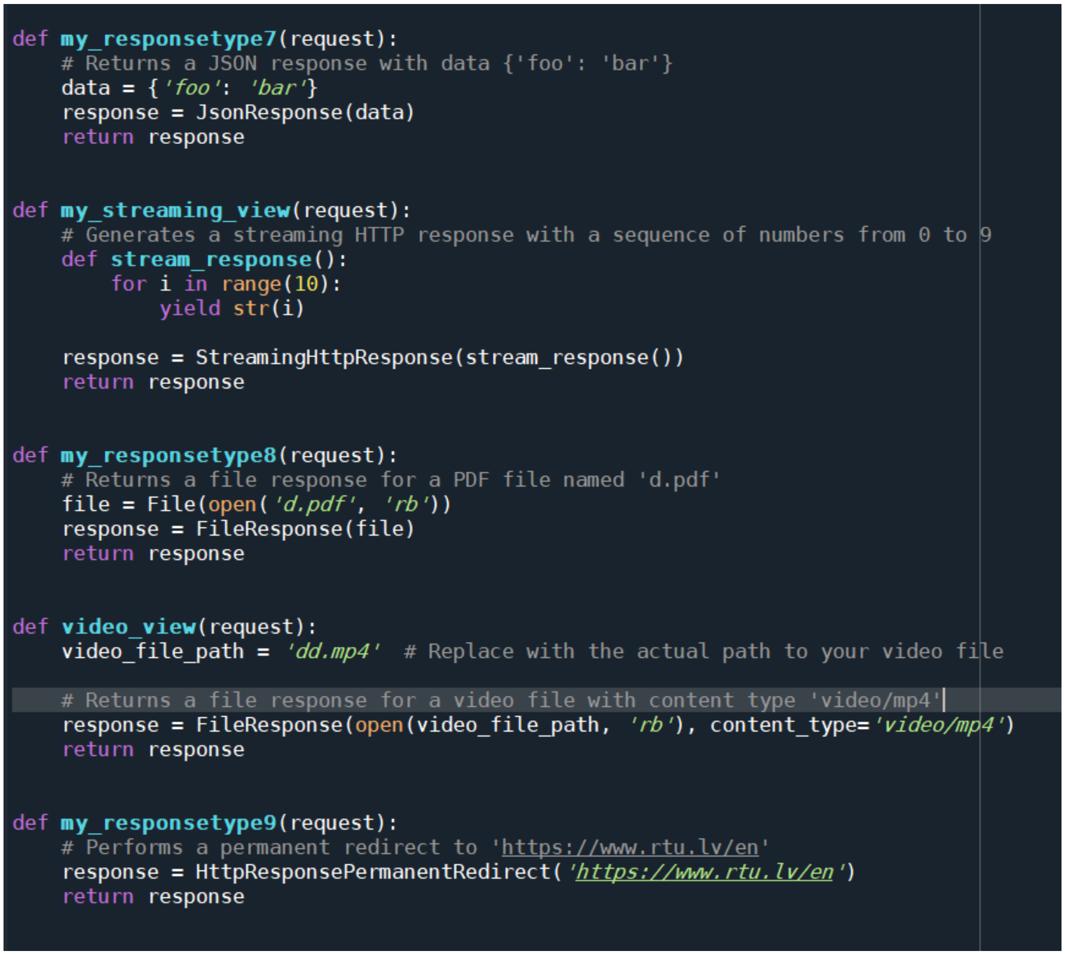
9- StreamingHttpResponse

10- FileResponse

- Add functions to **views.py** file:

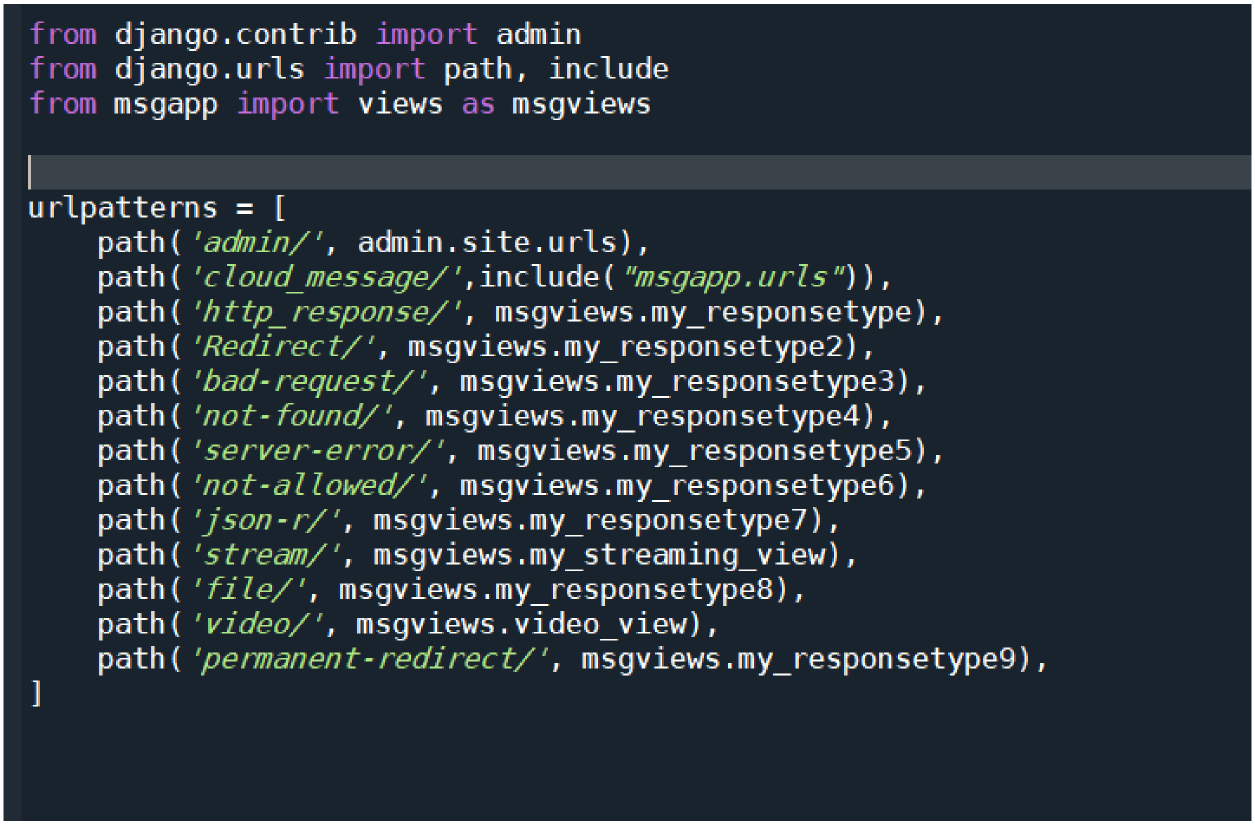


```python
def my_responsetype(request):
    # Returns a simple HTTP response with the content 'Hello, world!'
    response = HttpResponse('Hello, this is HTTP Response')
    return response


def my_responsetype2(request):
    # Performs a permanent redirect to 'https://www.rtu.lv/en'
    response = HttpResponseRedirect('https://www.rtu.lv/en')
    return response


def my_responsetype3(request):
    # Returns an HTTP response indicating a bad request
    response = HttpResponseBadRequest('This is a BAD request')
    return response


def my_responsetype4(request):
    # Returns an HTTP response indicating that the requested page was not found
    response = HttpResponseNotFound('Page not found :(')
    return response



def my_responsetype5(request):
    # Returns an HTTP response indicating an internal server error
    response = HttpResponseServerError('Internal server error')
    return response


def my_responsetype6(request):
    # Returns an HTTP response indicating that the requested HTTP method is not allowed
    response = HttpResponseNotAllowed(['GET', 'POST'])
    return response
```

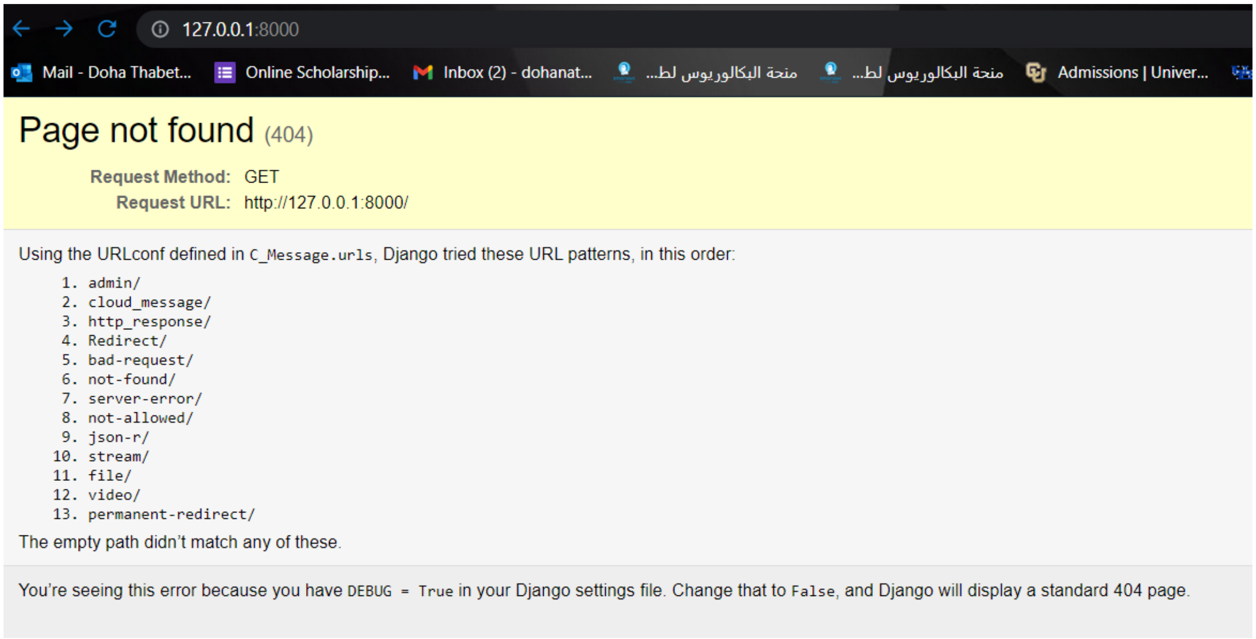


```python
def my_responsetype7(request):
    # Returns a JSON response with data {'foo': 'bar'}
    data = {'foo': 'bar'}
    response = JsonResponse(data)
    return response


def my_streaming_view(request):
    # Generates a streaming HTTP response with a sequence of numbers from 0 to 9
    def stream_response():
        for i in range(10):
            yield str(i)

    response = StreamingHttpResponse(stream_response())
    return response


def my_responsetype8(request):
    # Returns a file response for a PDF file named 'd.pdf'
    file = File(open('d.pdf', 'rb'))
    response = FileResponse(file)
    return response


def video_view(request):
    video_file_path = 'dd.mp4'  # Replace with the actual path to your video file

    # Returns a file response for a video file with content type 'video/mp4'
    response = FileResponse(open(video_file_path, 'rb'), content_type='video/mp4')
    return response


def my_responsetype9(request):
    # Performs a permanent redirect to 'https://www.rtu.lv/en'
    response = HttpResponsePermanentRedirect('https://www.rtu.lv/en')
    return response
```

- Add routing to **urls.py** file:

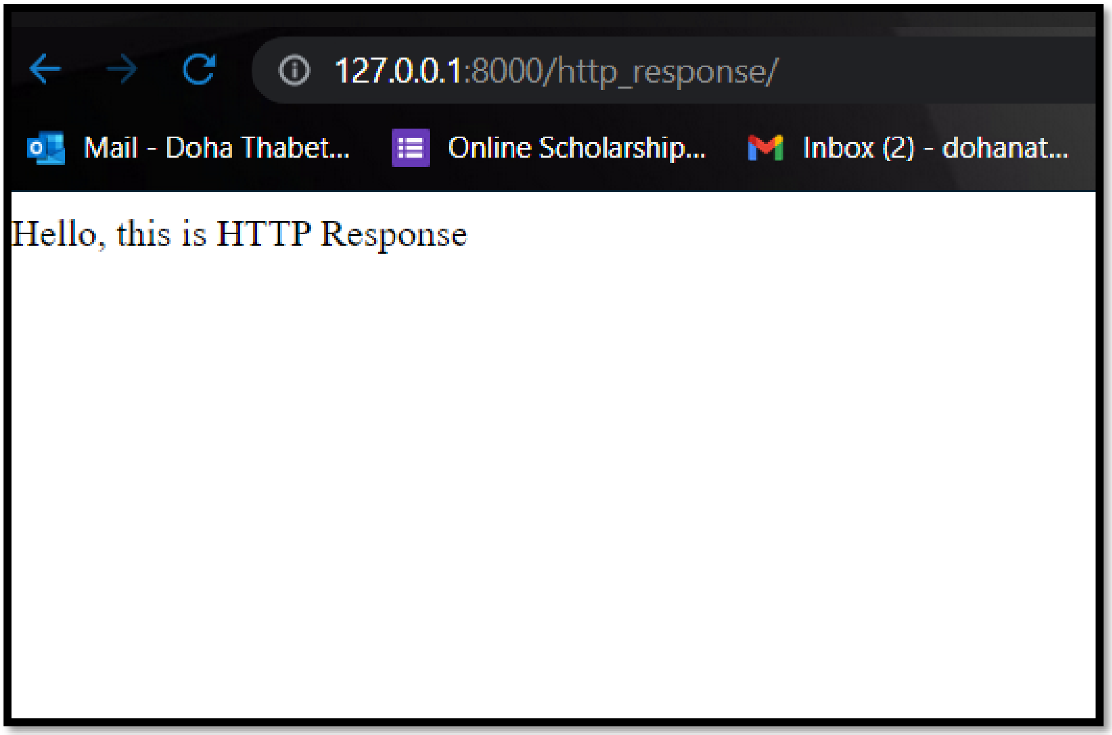


```
from django.contrib import admin
from django.urls import path, include
from msgapp import views as msgviews

urlpatterns = [
    path('admin/', admin.site.urls),
    path('cloud_message/',include("msgapp.urls")),
    path('http_response/', msgviews.my_responsetype),
    path('Redirect/', msgviews.my_responsetype2),
    path('bad-request/', msgviews.my_responsetype3),
    path('not-found/', msgviews.my_responsetype4),
    path('server-error/', msgviews.my_responsetype5),
    path('not-allowed/', msgviews.my_responsetype6),
    path('json-r/', msgviews.my_responsetype7),
    path('stream/', msgviews.my_streaming_view),
    path('file/', msgviews.my_responsetype8),
    path('video/', msgviews.video_view),
    path('permanent-redirect/', msgviews.my_responsetype9),
]
```



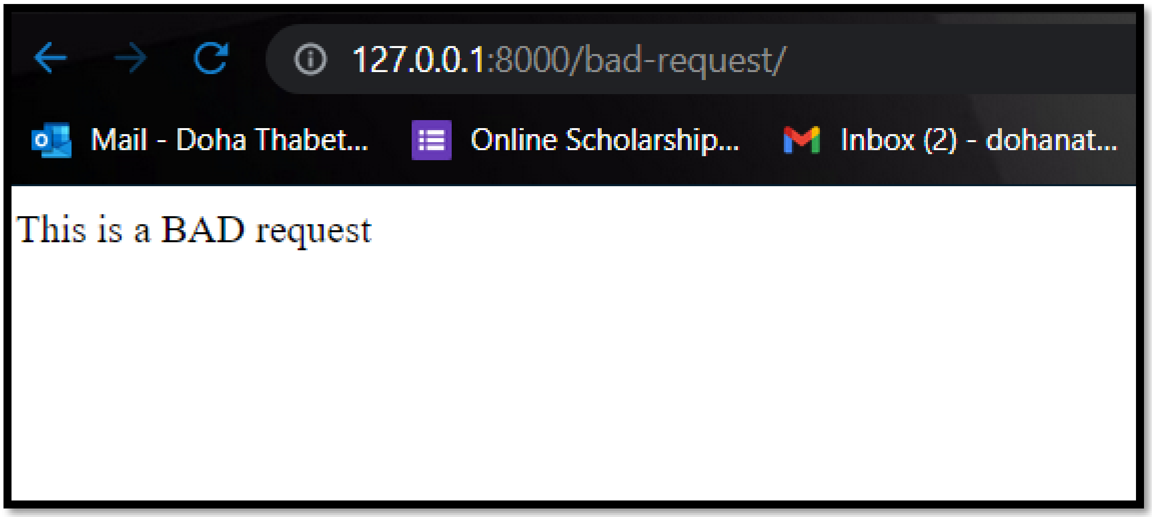
```
Page not found (404)

    Request Method:  GET
       Request URL:  http://127.0.0.1:8000/

Using the URLconf defined in C_Message.urls, Django tried these URL patterns, in this order:

    1. admin/
    2. cloud_message/
    3. http_response/
    4. Redirect/
    5. bad-request/
    6. not-found/
    7. server-error/
    8. not-allowed/
    9. json-r/
   10. stream/
   11. file/
   12. video/
   13. permanent-redirect/

The empty path didn't match any of these.

You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404 page.
```

1- **HttpResponse Results:**



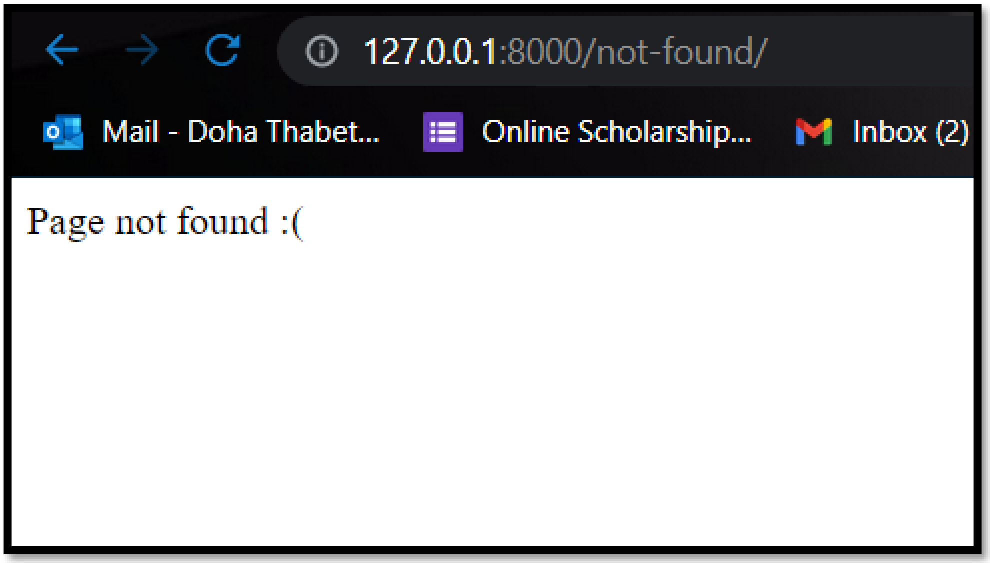
Hello, this is HTTP Response

2- **HttpResponseBadRequest Results:**



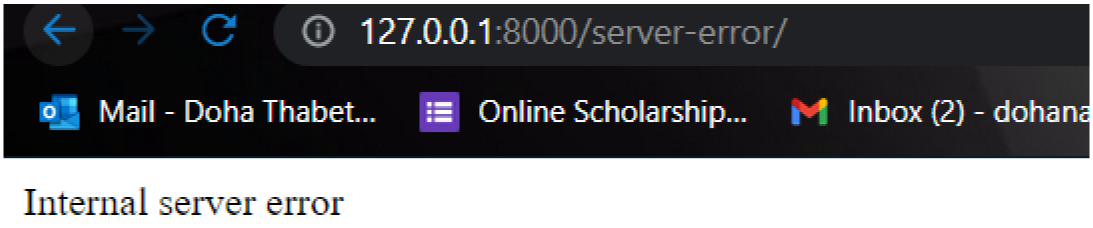
This is a BAD request

3- **HttpResponseNotFound Results:**



Page not found :(

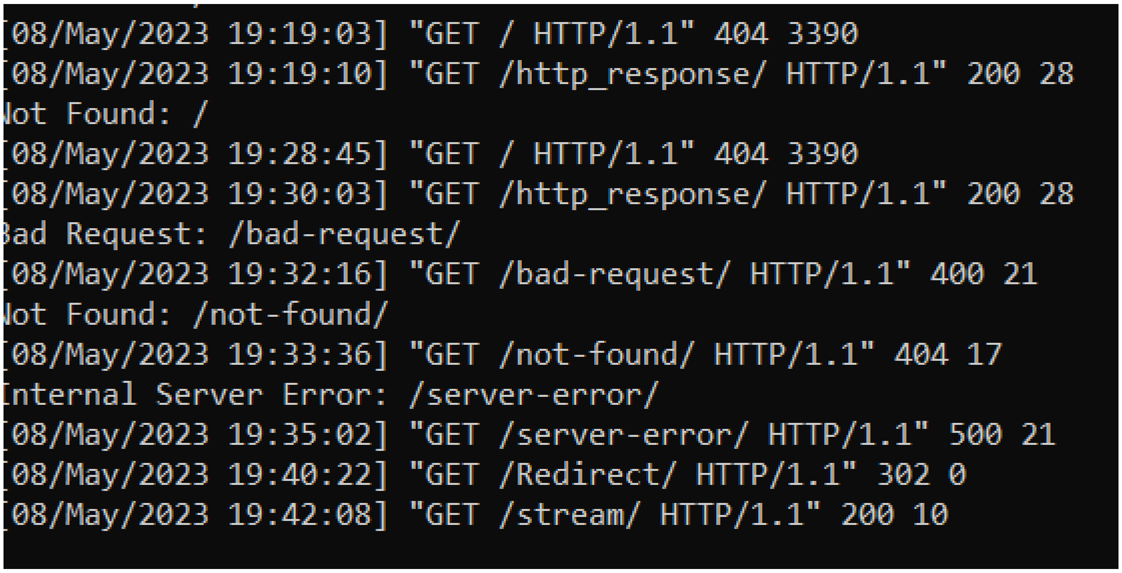
4- **HttpResponseServerError Results:**



Internal server error

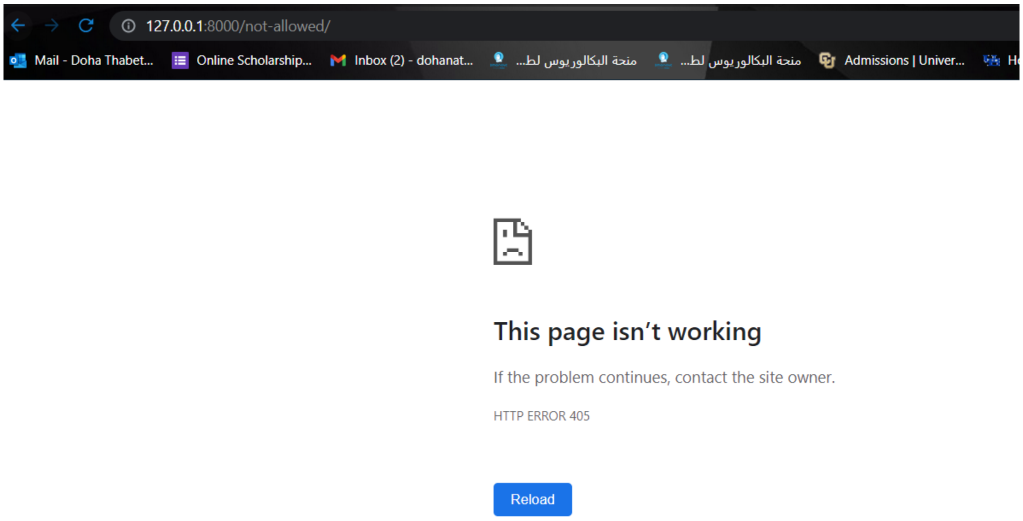
5- **HttpResponseRedirect Results: (The response will direct the page to the RTU website page)**

6- **HttpResponsePermanentRedirect (The response will direct the page to the RTU website page)**



```
[08/May/2023 19:19:03] "GET / HTTP/1.1" 404 3390
[08/May/2023 19:19:10] "GET /http_response/ HTTP/1.1" 200 28
lot Found: /
[08/May/2023 19:28:45] "GET / HTTP/1.1" 404 3390
[08/May/2023 19:30:03] "GET /http_response/ HTTP/1.1" 200 28
3ad Request: /bad-request/
[08/May/2023 19:32:16] "GET /bad-request/ HTTP/1.1" 400 21
lot Found: /not-found/
[08/May/2023 19:33:36] "GET /not-found/ HTTP/1.1" 404 17
nternal Server Error: /server-error/
[08/May/2023 19:35:02] "GET /server-error/ HTTP/1.1" 500 21
[08/May/2023 19:40:22] "GET /Redirect/ HTTP/1.1" 302 0
[08/May/2023 19:42:08] "GET /stream/ HTTP/1.1" 200 10
```

7- HttpResponseNotAllowed Results:

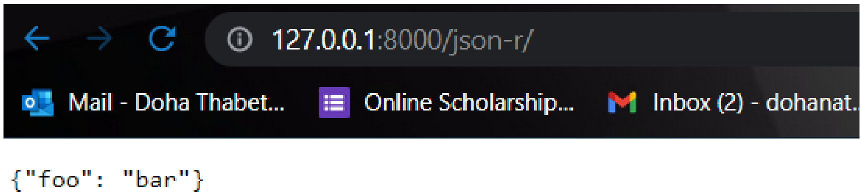


This page isn’t working

If the problem continues, contact the site owner.

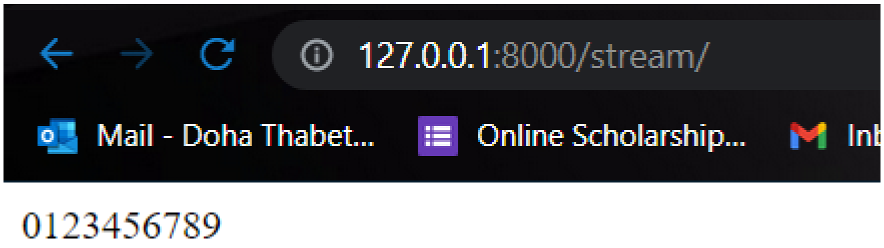
HTTP ERROR 405

8- JsonResponse Results:



```
{"foo": "bar"}
```

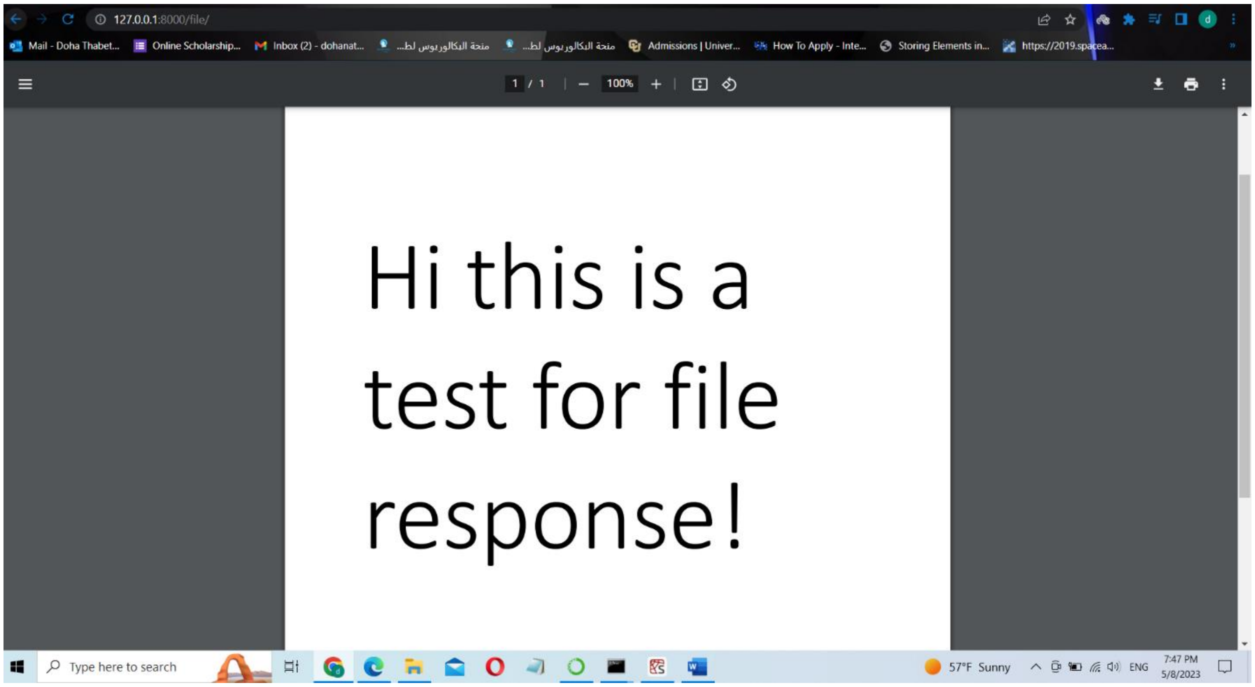
9- StreamingHttpResponse Results:



```
0123456789
```

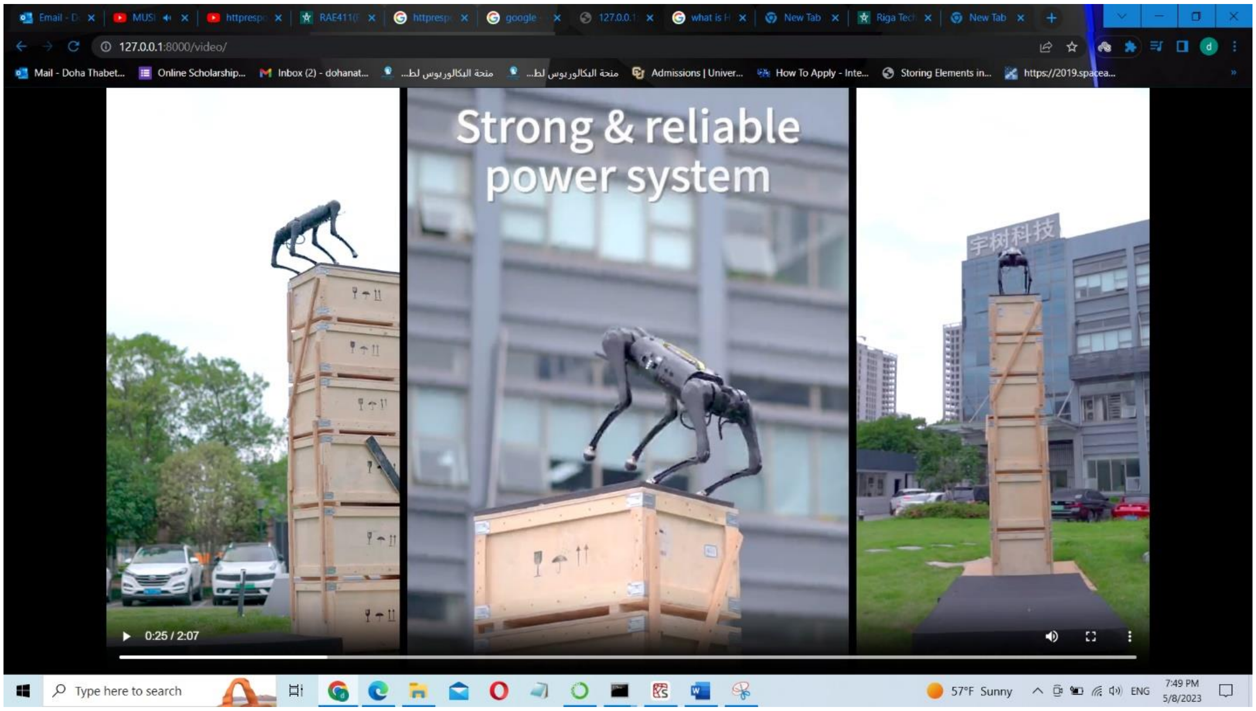
10- FileResponse Results :

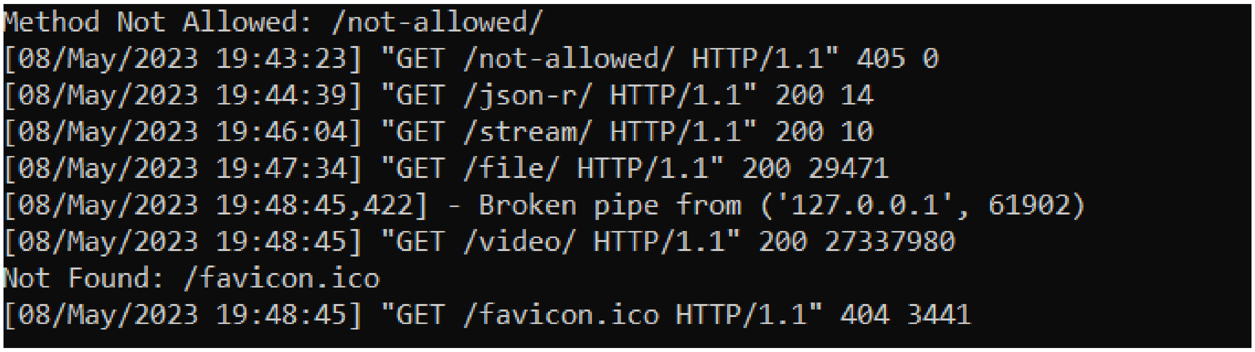
- **PDF file:**



Hi this is a test for file response!

- **Video file:**





```
Method Not Allowed: /not-allowed/
[08/May/2023 19:43:23] "GET /not-allowed/ HTTP/1.1" 405 0
[08/May/2023 19:44:39] "GET /json-r/ HTTP/1.1" 200 14
[08/May/2023 19:46:04] "GET /stream/ HTTP/1.1" 200 10
[08/May/2023 19:47:34] "GET /file/ HTTP/1.1" 200 29471
[08/May/2023 19:48:45,422] - Broken pipe from ('127.0.0.1', 61902)
[08/May/2023 19:48:45] "GET /video/ HTTP/1.1" 200 27337980
Not Found: /favicon.ico
[08/May/2023 19:48:45] "GET /favicon.ico HTTP/1.1" 404 3441
```