****

**Riga Technical University**

**Telecommunications Software (RAE411).**

**Third Practical Exercise.**

**29th of May 2023**

**Zeyad Mohamed Nashaat Abdelghany.**

**230AMB013.**

**Spring 2023.**

1. **Introduction:**

This report discusses the tasks needed to be made in third practical exercise. There are four examples which will be done using Django. First example is to create a Django’s hello world program to return a webpage with a string in it. The second example is to return an HTML page using Django. Third example discusses the idea of cloud message and how to accept and send message from a cloud to user on the web. Last example examines the 10 different HTML responses by using Django.

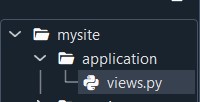
1. **Example 1:**

In this example we need to return a webpage with a string in it.

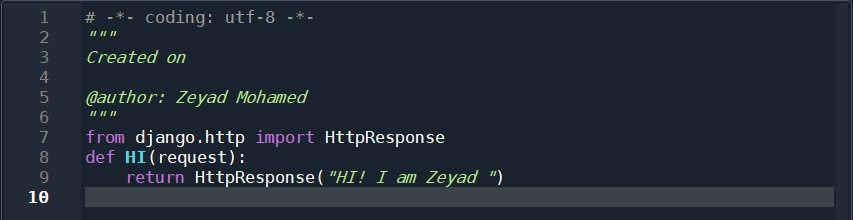
1. Create a new Django project by running this command in the terminal:

django-admin startproject <mysite>

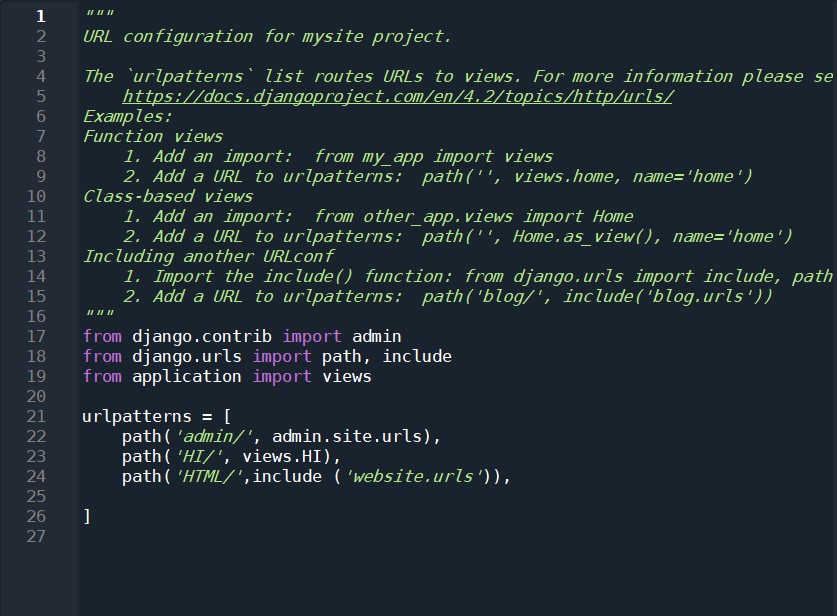
1. Create a folder for this example (application):



1. Create views.py and put the code in it:



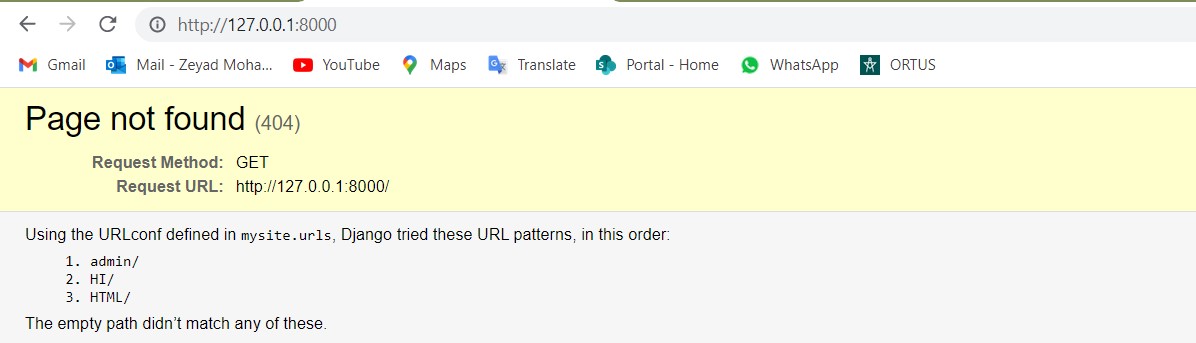
1. Add URL for the application in the main file urls.py:

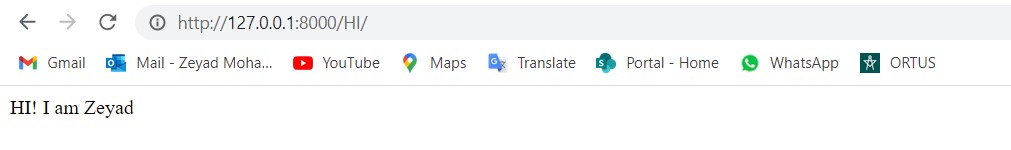


1. Run the program by writing this line in the terminal:

python manage.py runserver

1. Results should be as follows:

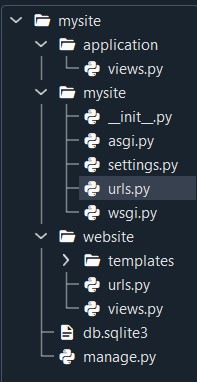




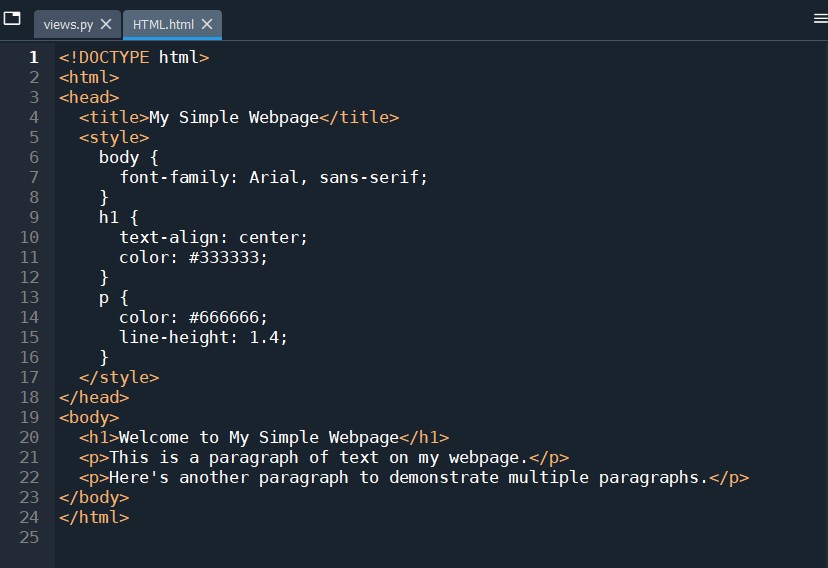
1. **Example 2:**

In this example, we will update the previous one to be able to return HTML page instead of a string.

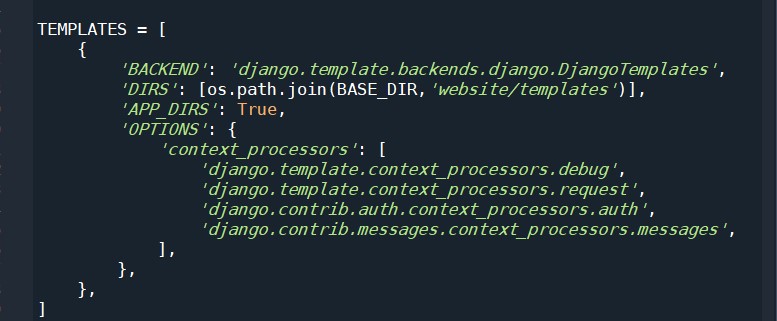
1. Create a folder for this example (website):



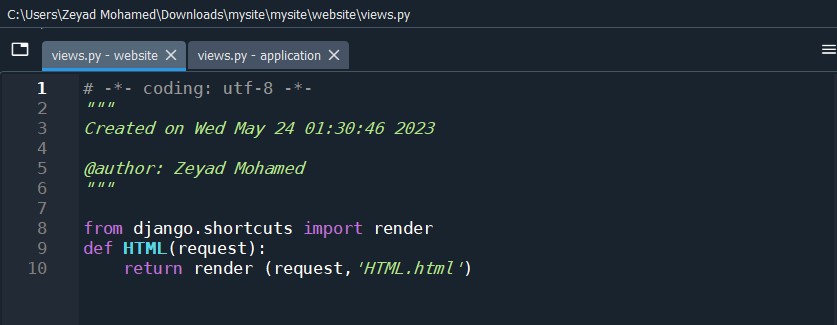
1. Add a template folder to hold the HTML code for the page:



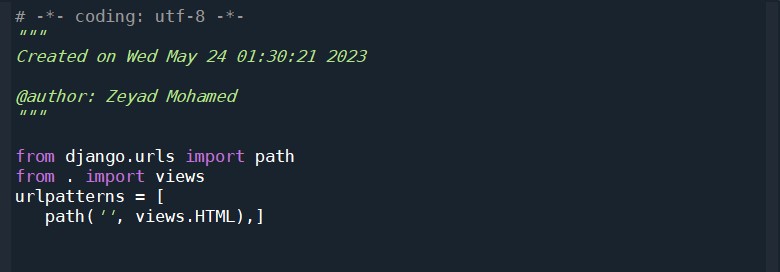
1. Edit the settings.py in the main project to hold the direction of the added template:



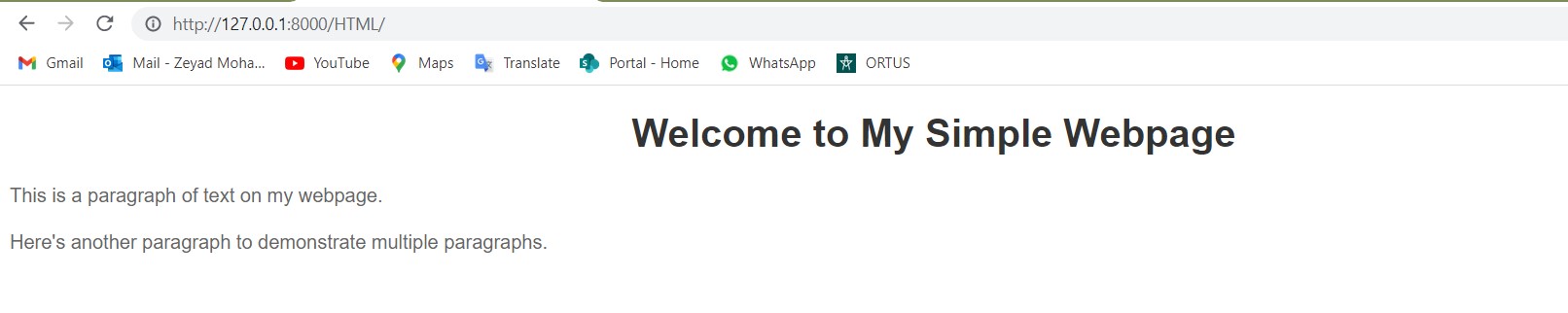
1. Add the function in views.py to call the template:



1. Add the URL for urls.py:



1. The results should be:



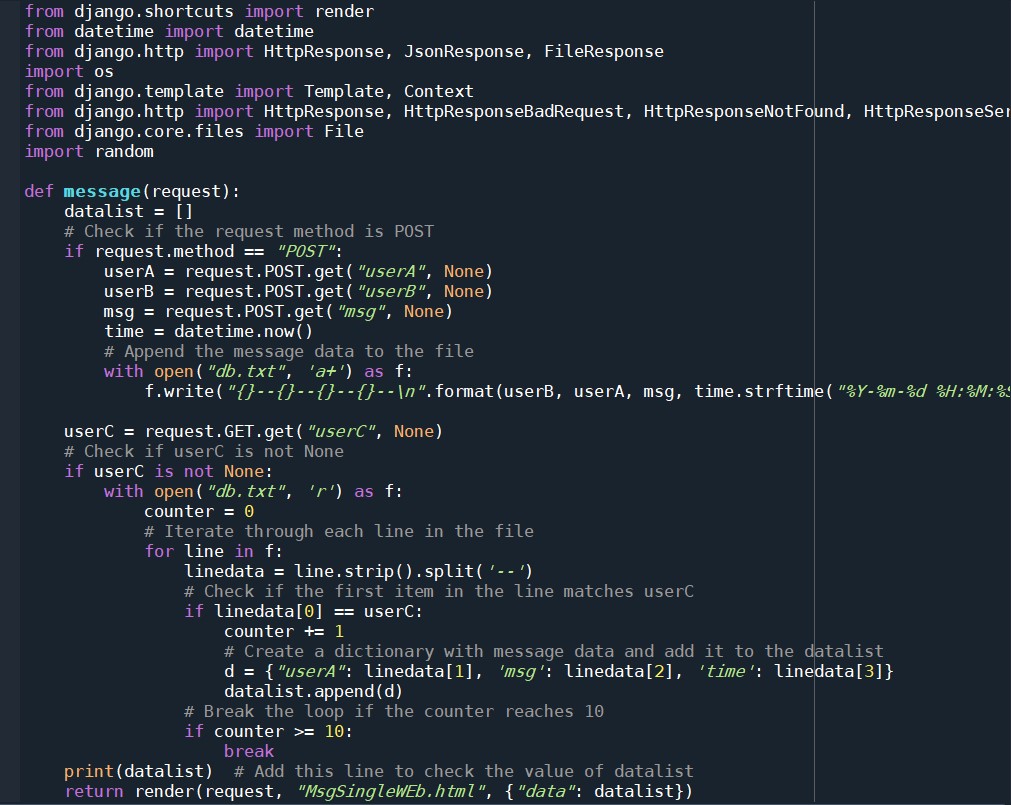
1. **Example 3:**

This example is to make a message system where you can submit a message with your name to a certain receiver. Also, the receiver could see up to 20 messages sent to him by showing the message with the name of the sender as well. The steps to create such project:

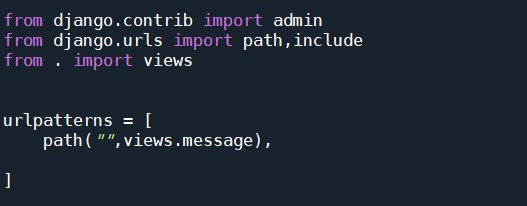
1. Create a new Django project through the terminal:

django-admin startproject <C\_Message>

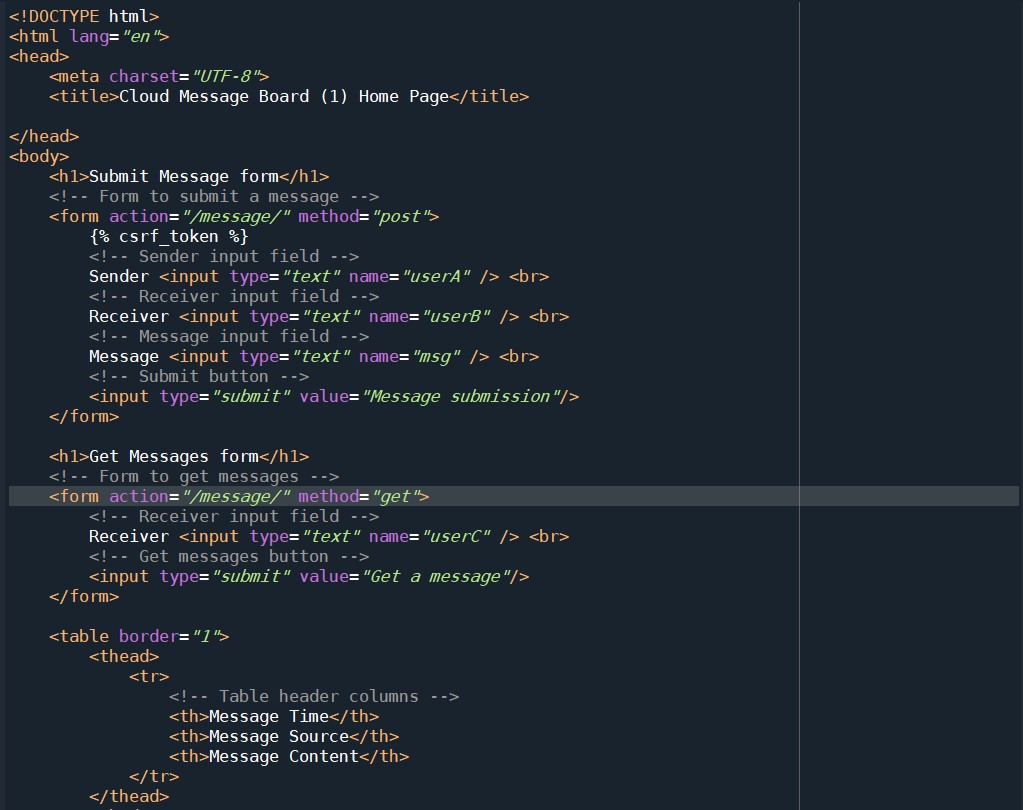
1. Add functions to views.py:



1. Add the needed URLs to urls.py:

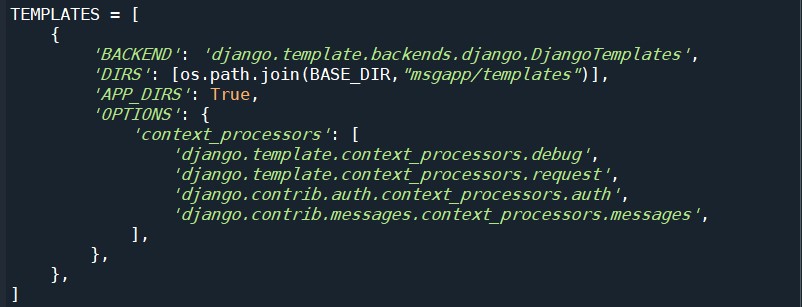


1. Create an HTML page to be the view for the user to hold the messages sending and receiving actions:

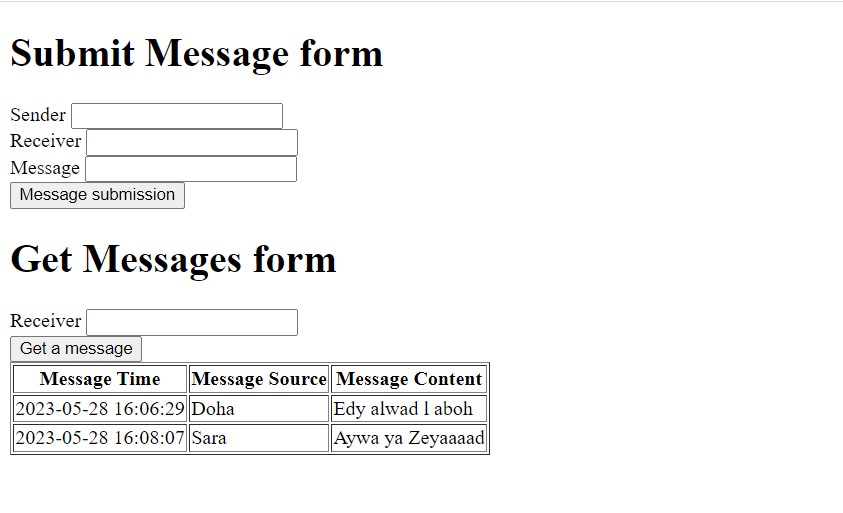


****

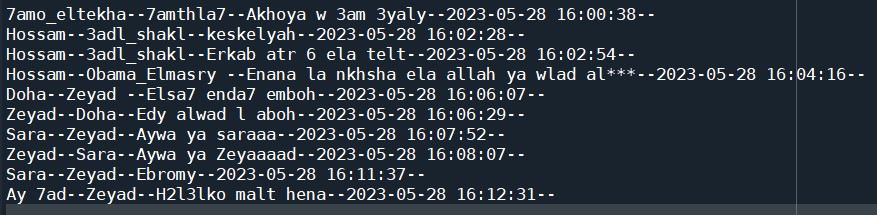
1. Adjust templates part in settings.py:



1. For the working results:



1. For the data base of the messages:



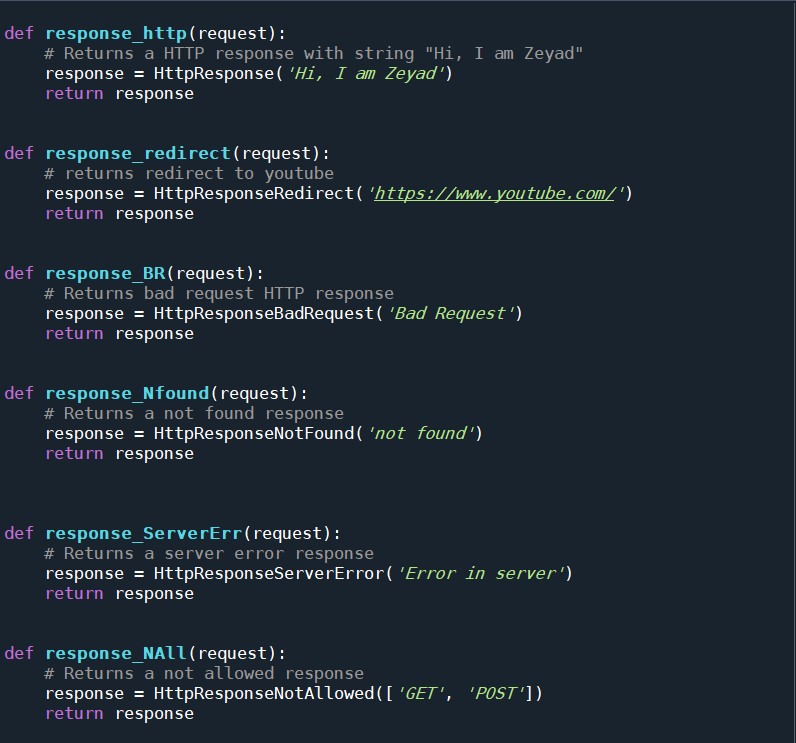
1. **Example 4:**

In this section, 10 responses for Django are tried. The 10 responses will be explained with given screenshots and discussion.

1. **Http response:** this response returns a simple Http page with a string in it “Hi, I am Zeyad”.
2. **Redirect Response:** this response returns a redirect page to any website, for example: YouTube.
3. **Bad Request response:** this response returns an HTTP page with Bad request.
4. **Not Found response:** this response returns Not found HTTP response.
5. **Server Error response:** this response returns server error HTTP response.
6. **Not Allowed response:** this response returns not allowed HTTP response.
7. **JSON response:** this response returns a JSON file with data “{‘Z’:’ Hi’}”.
8. **Stream response:** this response returns a stream HTTP response. For example, it can return a stream of 10 random alphabets.
9. **File response:** this response returns a file as HTTP response. For example, you can return pdf file as response.
10. **Video response:** this response returns a video as HTTP response.
11. **Permanent redirect response:** this response returns a permanent redirect link to YouTube.

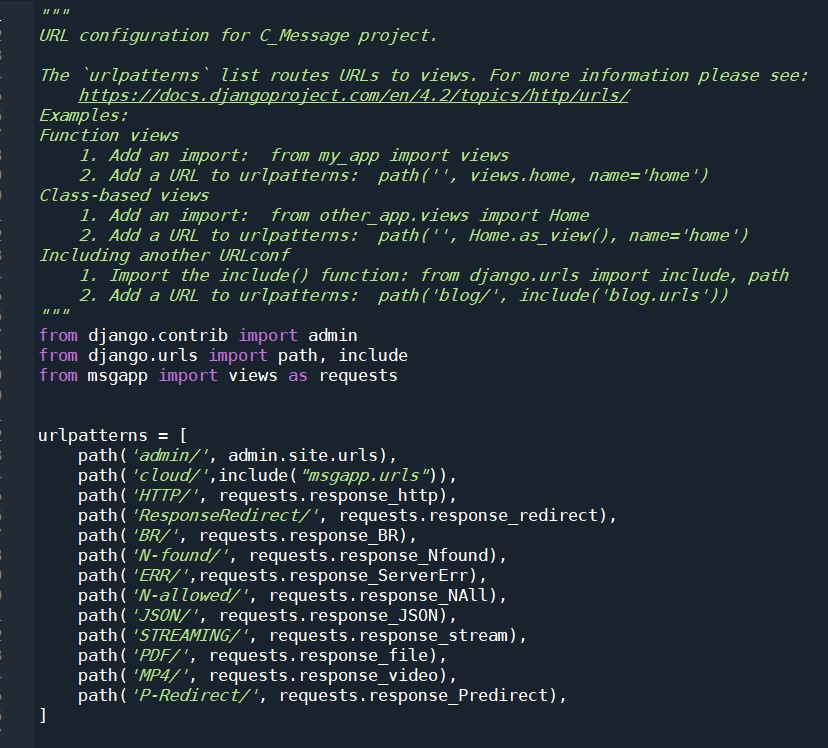
For the steps to make such responses:

1. Add the returning functions to views.py:

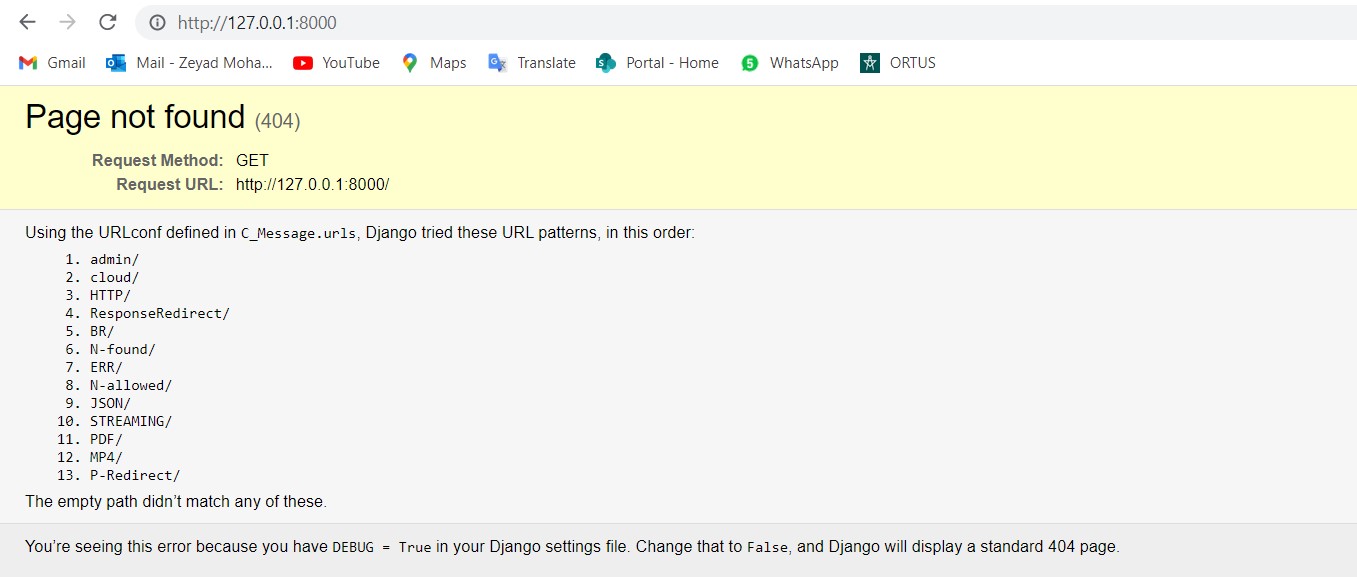




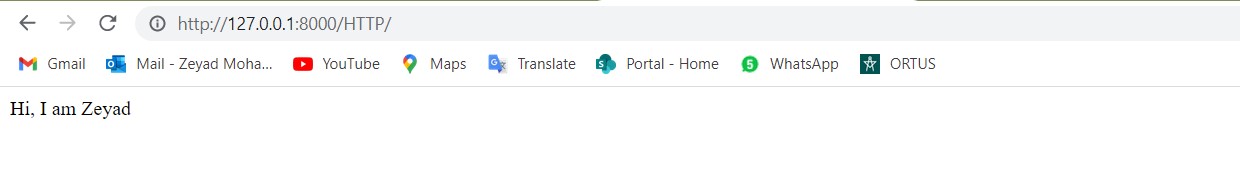
1. Adjust urls.py file:



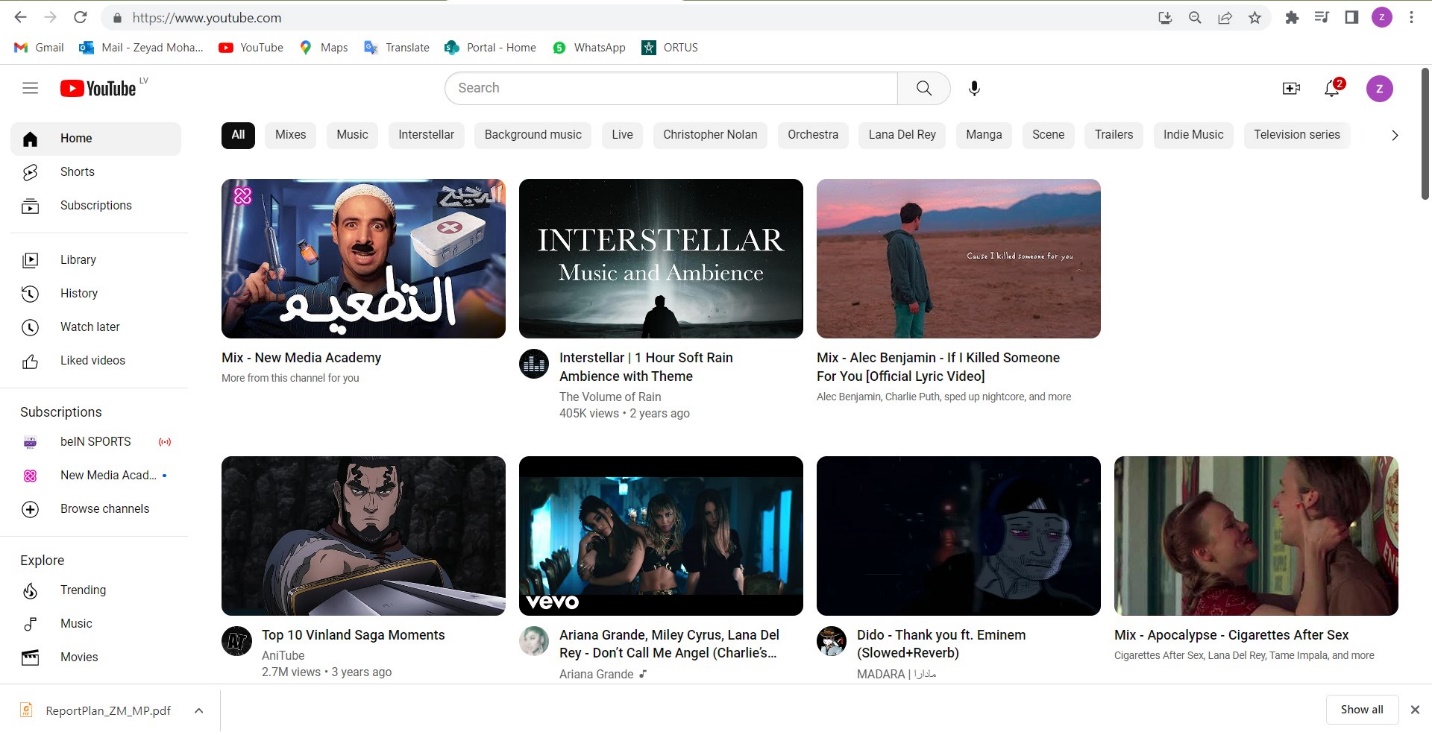
1. Results:



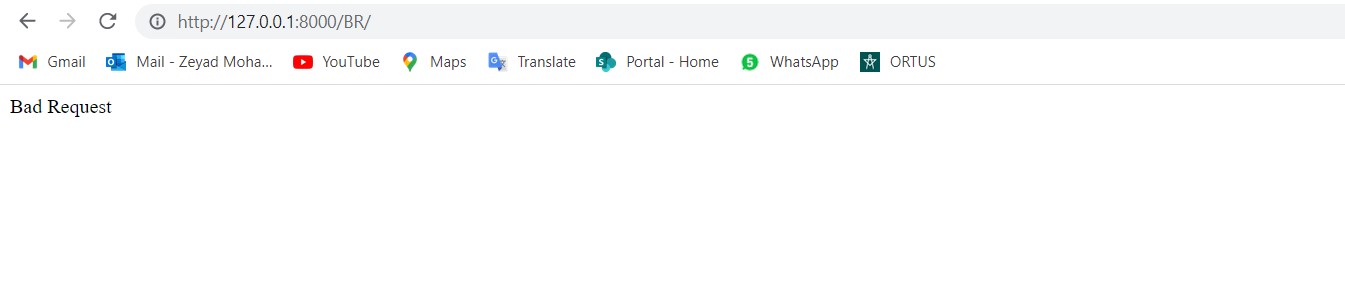
1. HTTP response:



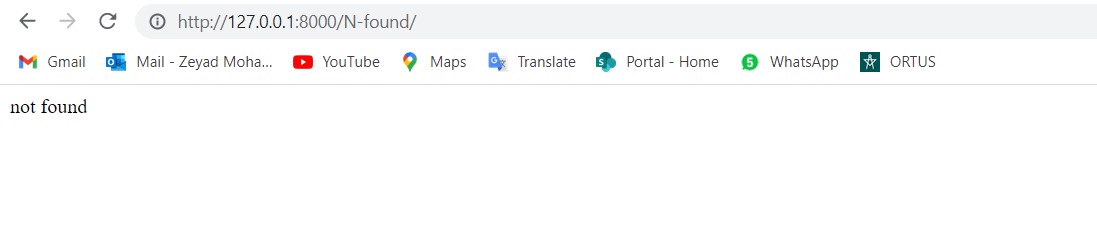
1. Redirect Response:



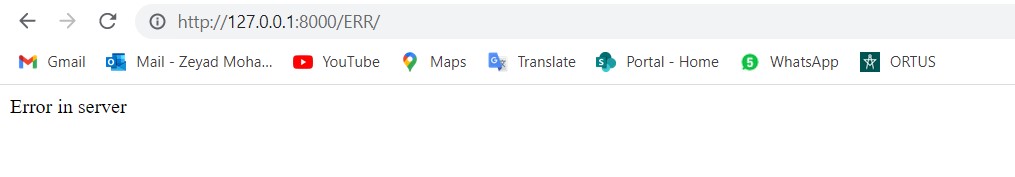
1. Bad Request response:



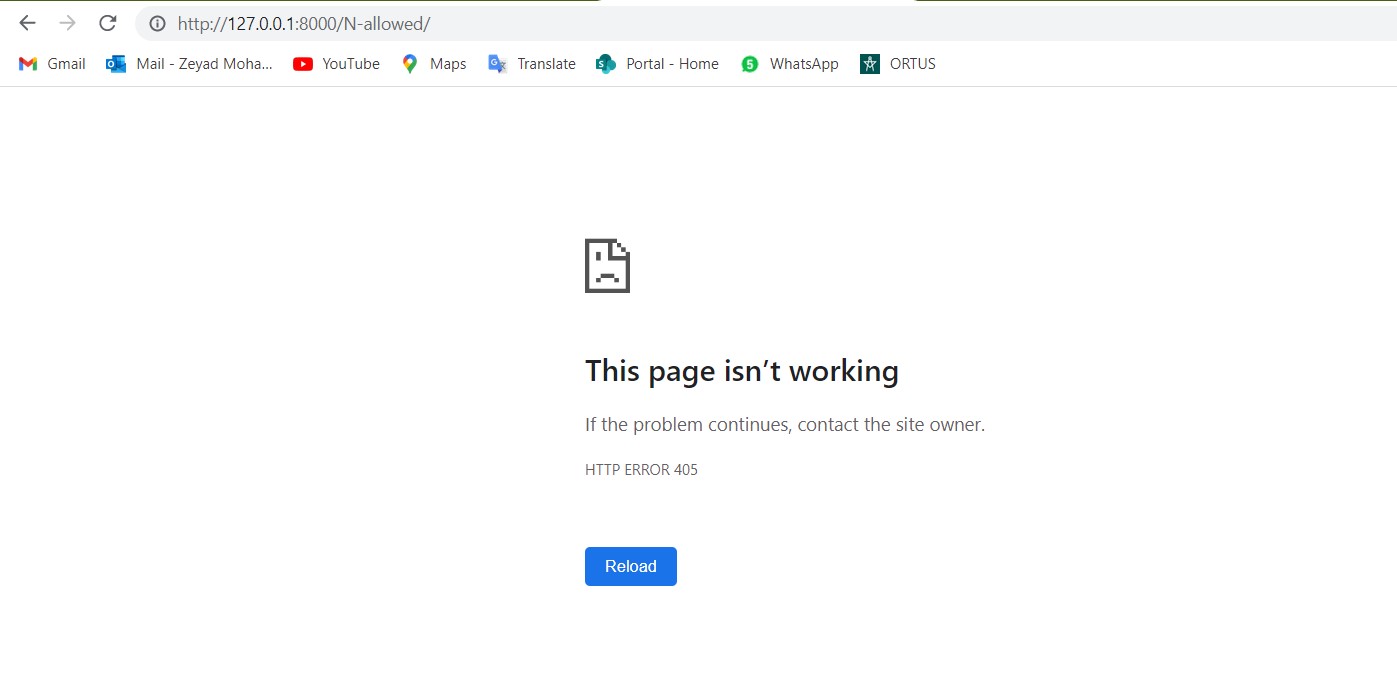
1. Not Found response:



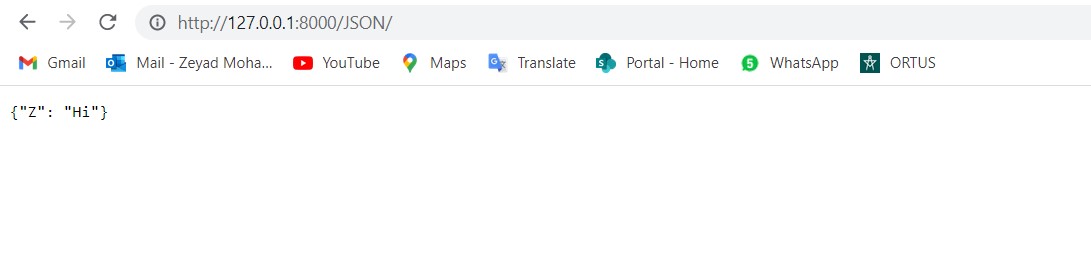
1. Server Error response:



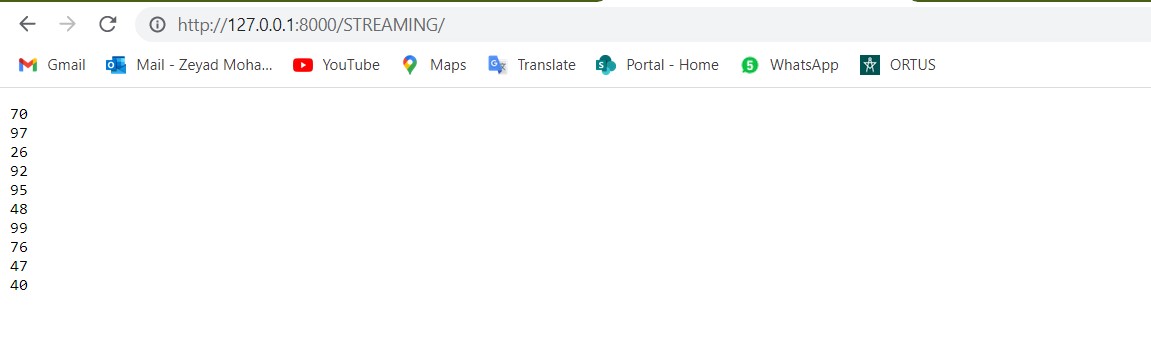
1. Not Allowed response:



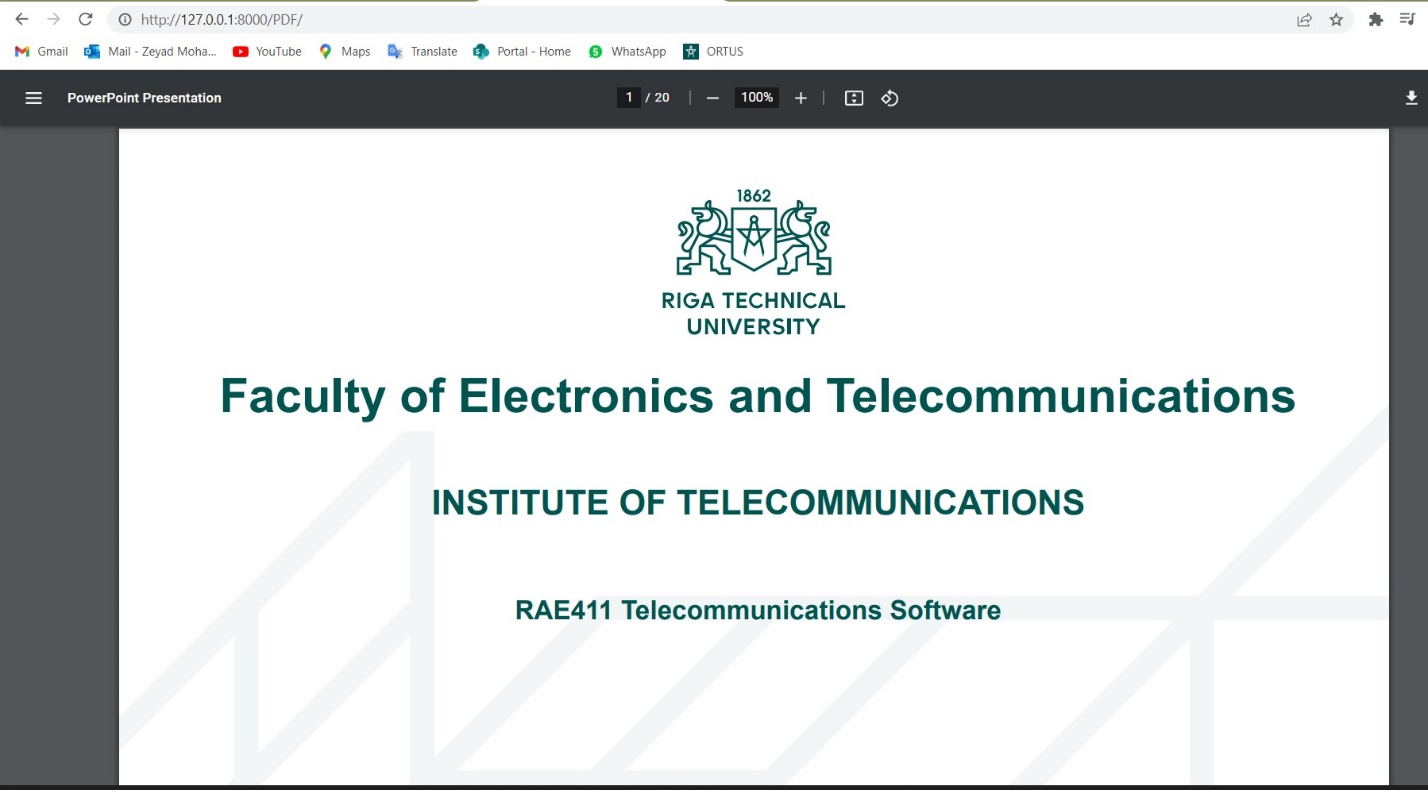
1. JSON response:



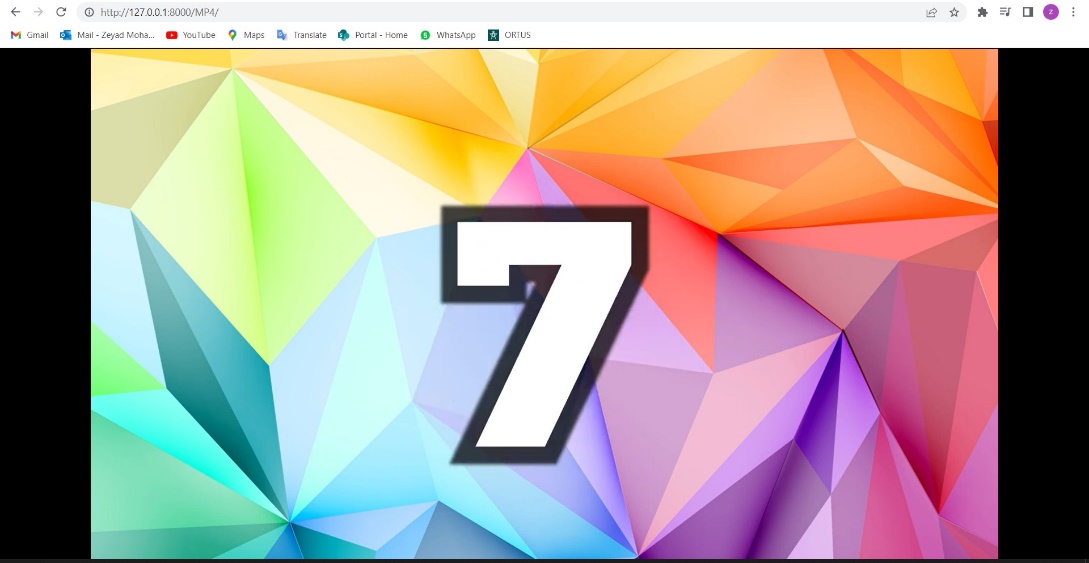
1. Stream response:



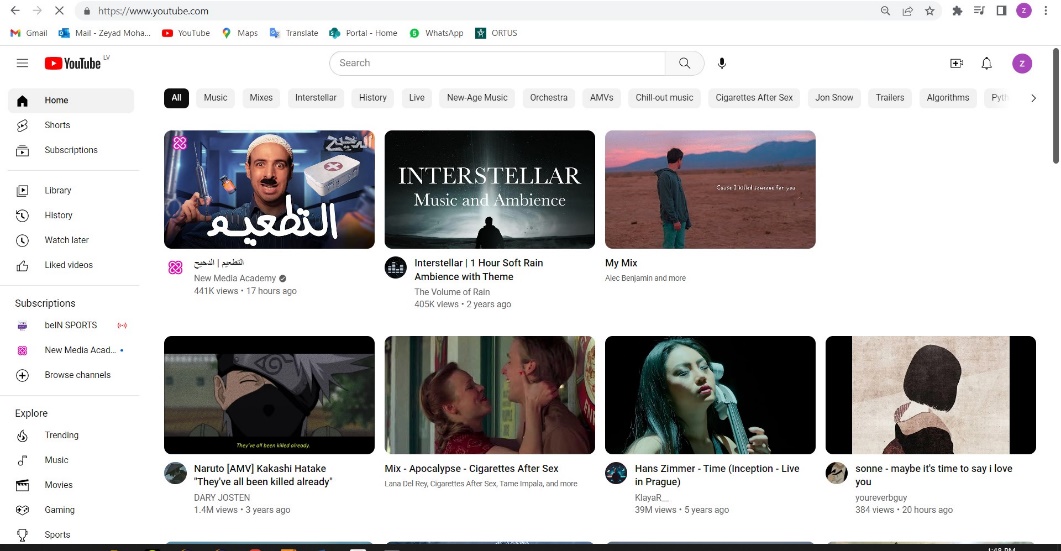
1. File response:

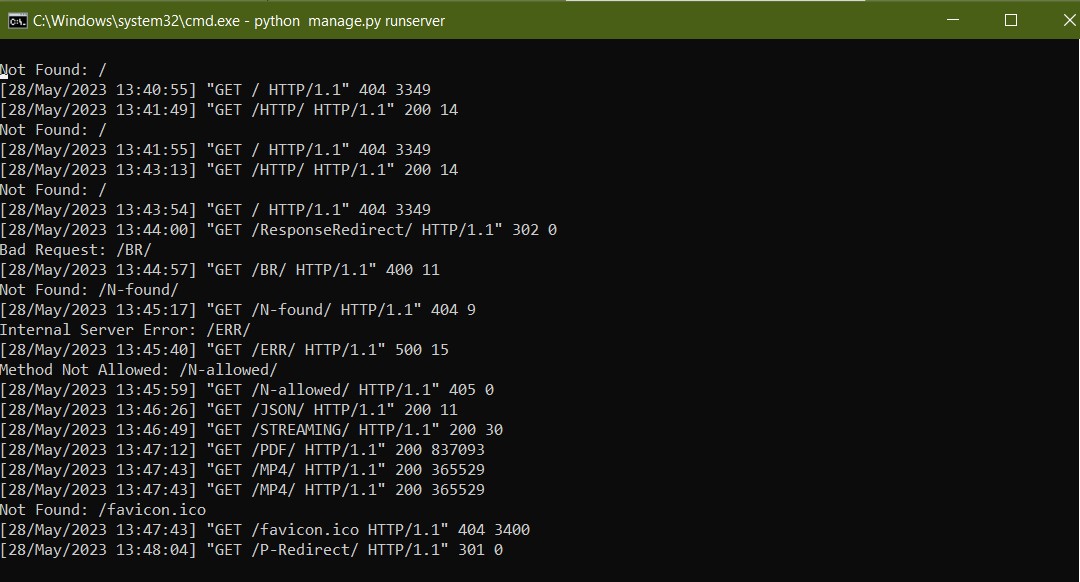


1. Video response:



1. Permanent redirect response:



For all the responses in the terminal: