

**Telecommunication Software**

Third Practical Exercise

Full name: **Mithin kumar Ananthula**

ID: **241AEM014**

**Riga 2024**

Table of Contents

Task13

Example 13

**Example 25**

**Example 36**

**Example 411**

Task218

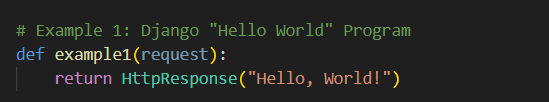
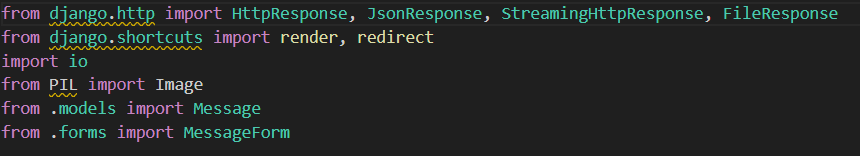
GitHub18

**Practical Exercise 3 (Python cloud full stack development)**

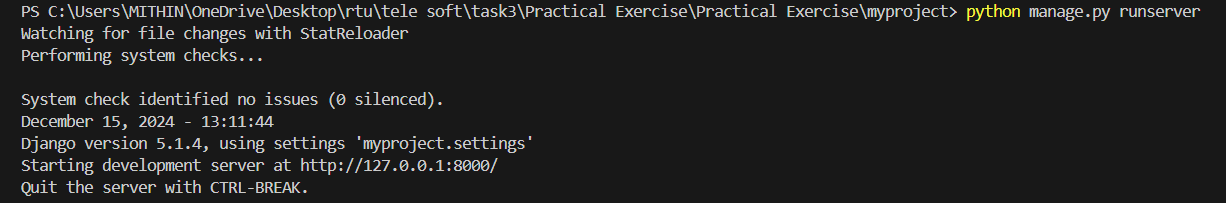
Task1:  
  
Example 1

Django's Hello World program (Tips: • This application inspection of a specific function, please return a string-related web page that can combine with CSS and JS to design your style.)

Code :

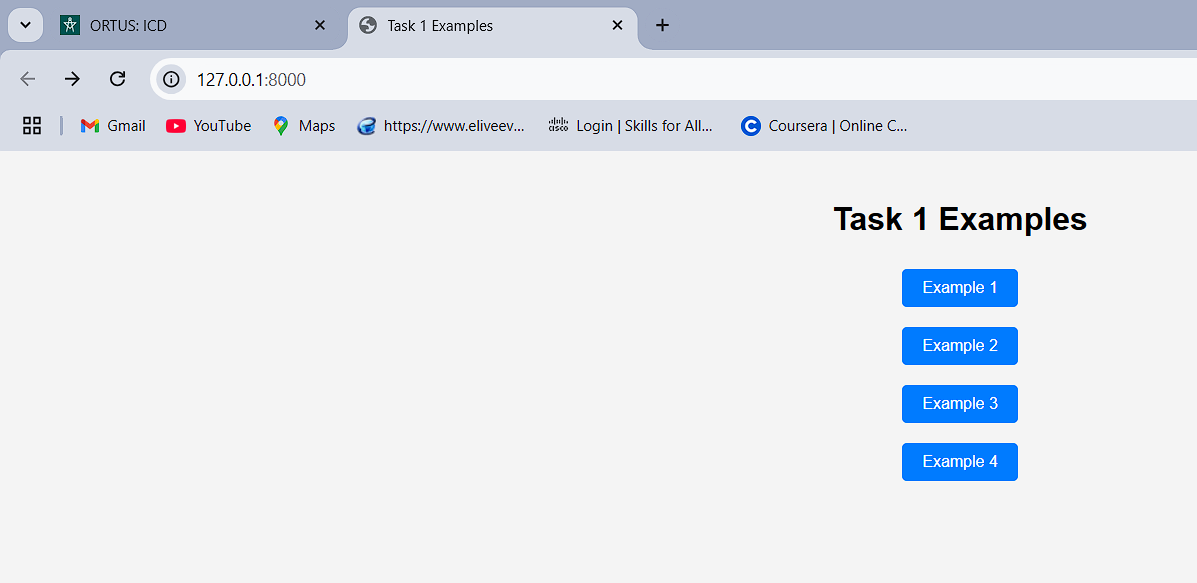


Command : python manage.py runserver

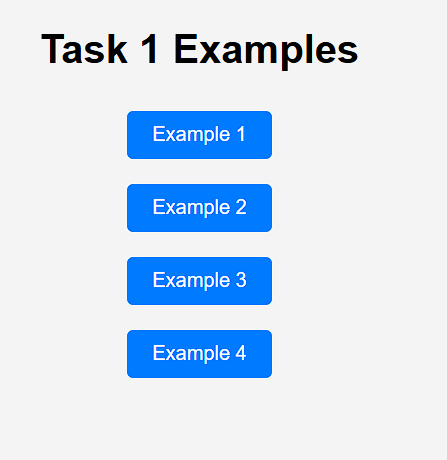


As in the about image you can see the server link at <http://127.0.0.1:8000/>

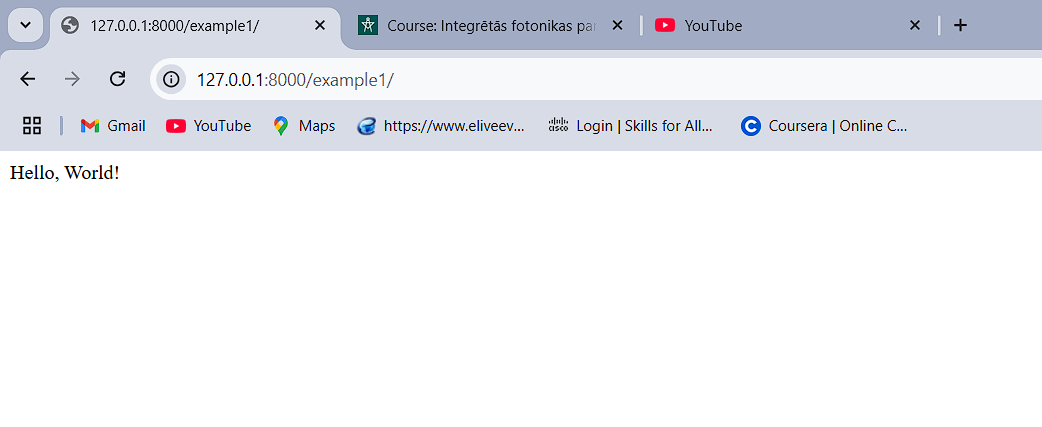
Copy the link and paste in any browser



Click on example 1



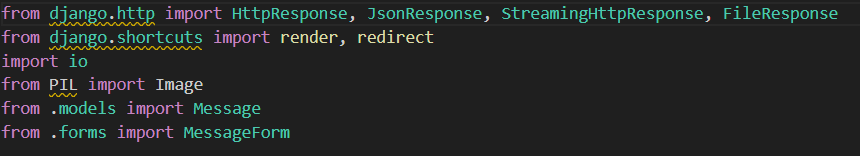
Example 1 of Django's “Hello World” program

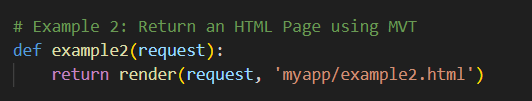


### Example 2

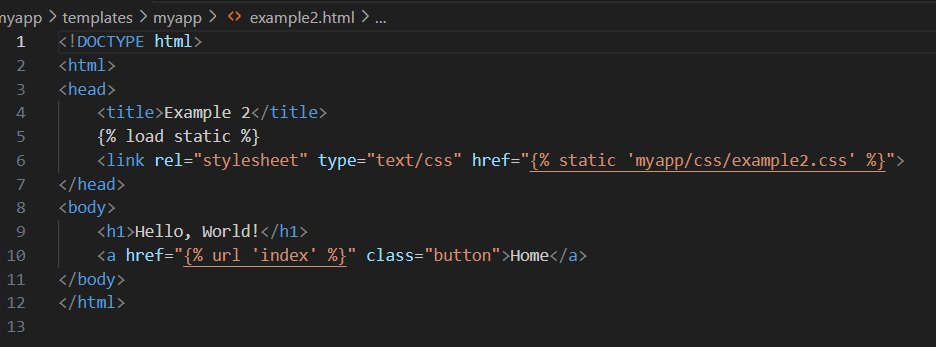
Improve example 1 and return an HTML page, not a string; please use MVT (Model View Template) to design the pattern. You are welcome to use multiple decorations on your HTML page.

Code :

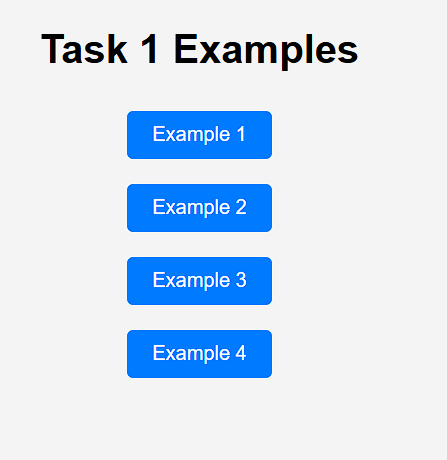




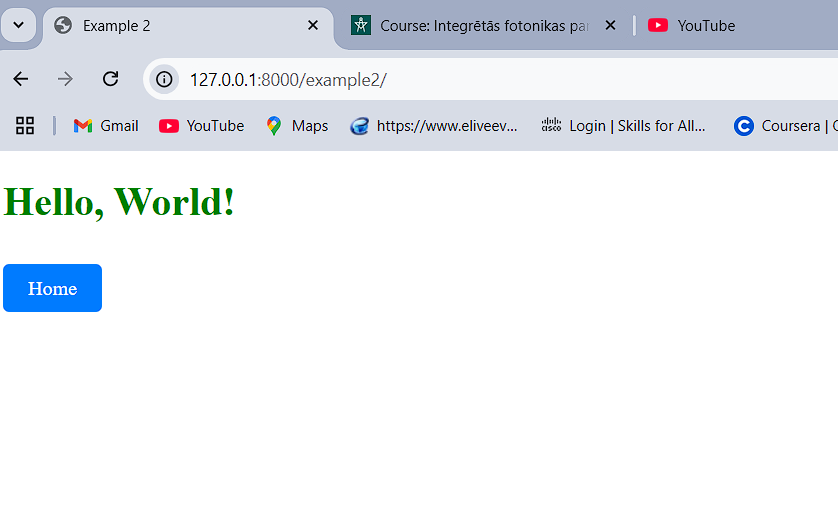
**HTML**



same as in the above link you can click on example 2



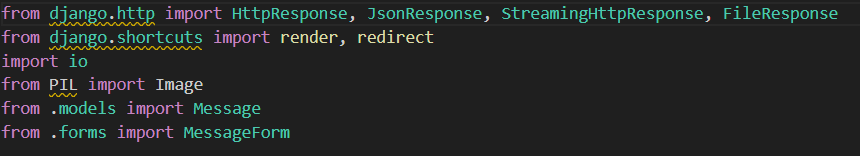
MVT (Model View Template) to design the pattern.

****

#### Example 3

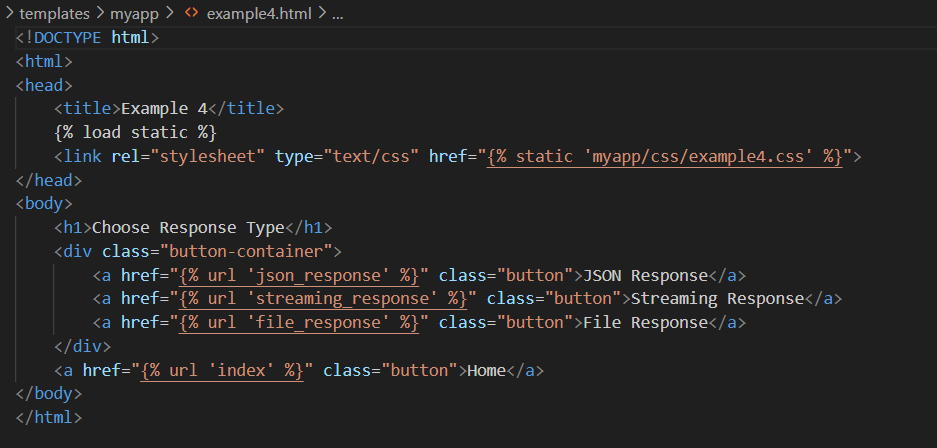
Cloud message board. Basic function definition: 1, Submit message function: Users can set their name as A, specify any name B and leave a message to B, record it as msg, and the message will be saved in the cloud. 2, Get message function: input name A, and the cloud will return the 20 latest message records.

Code:

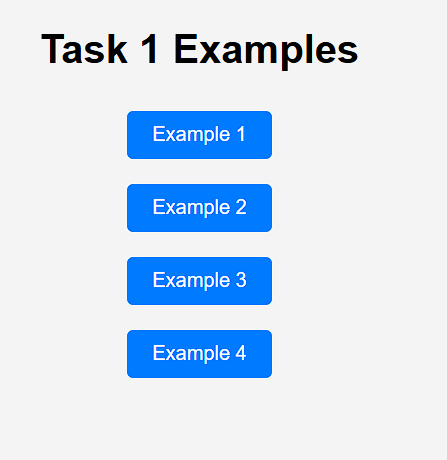




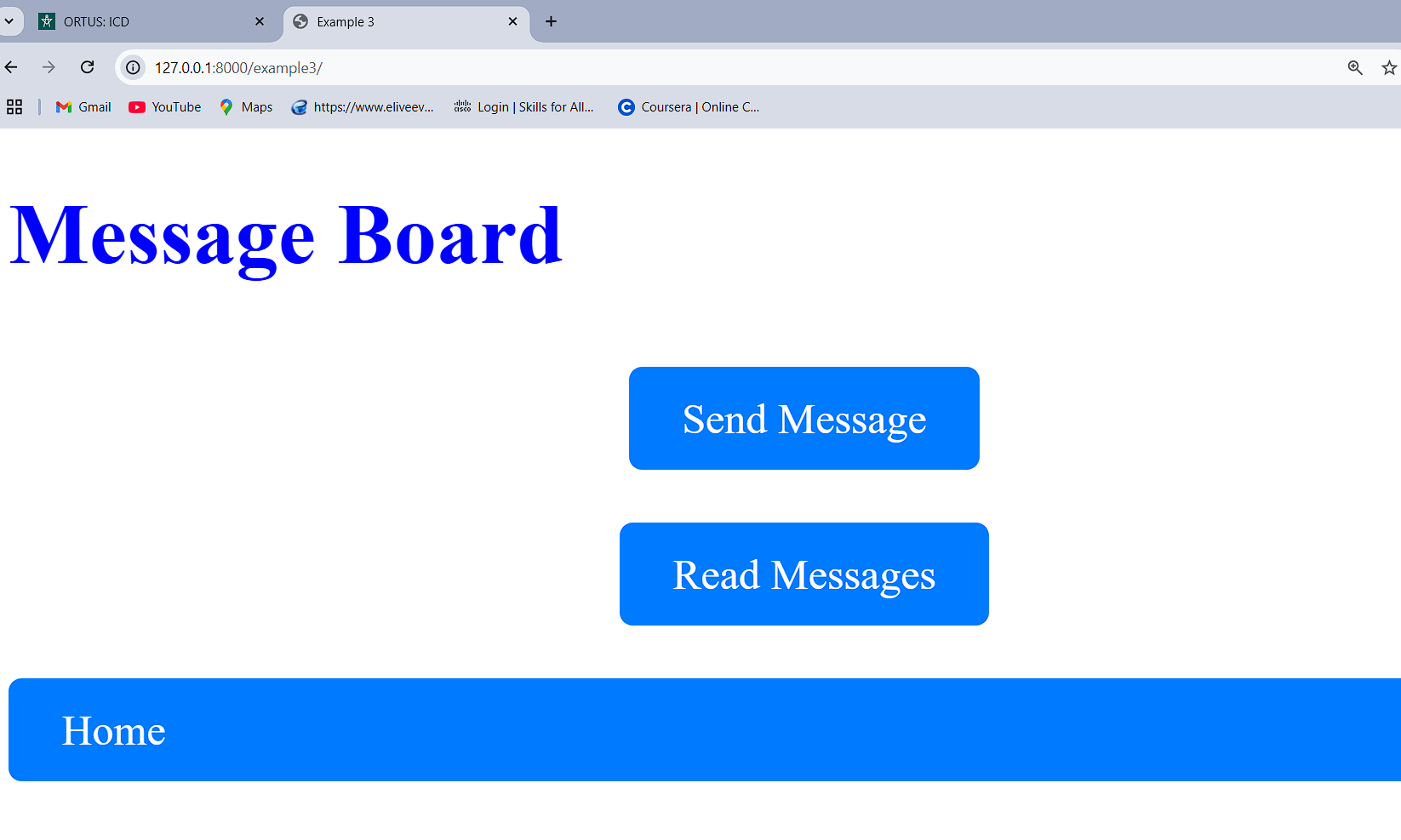
**HTML**



same as in the above link you can click on example 3



It will direct you to the message board were you can choose to send message or read message.

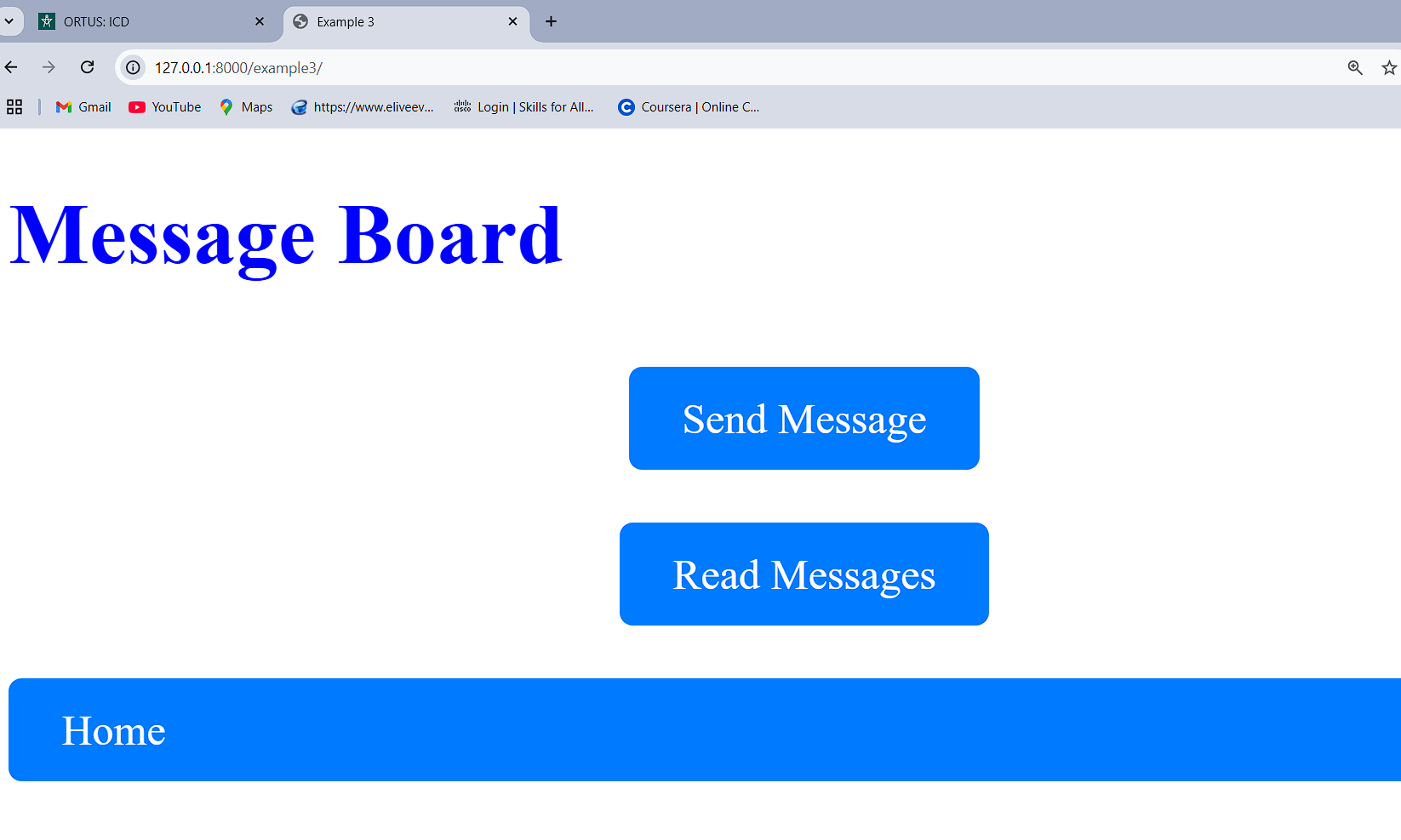
****

**Click on send message**

It will direct to the send message page whereas sender I Names as ‘a’, receiver as ‘b’ and you can type the message you want to sent and click the post message button. I will send the message.

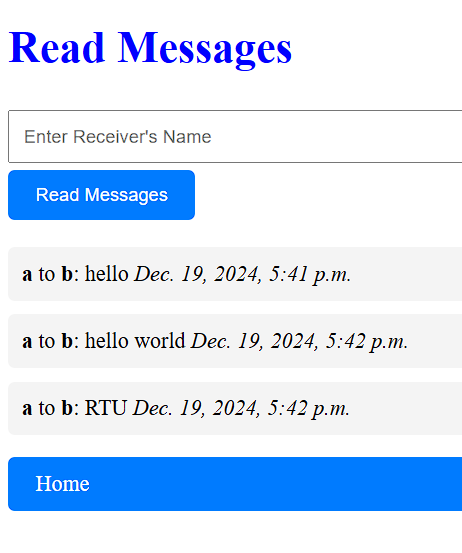


Next you can press Home or Back which will direct you to the message board

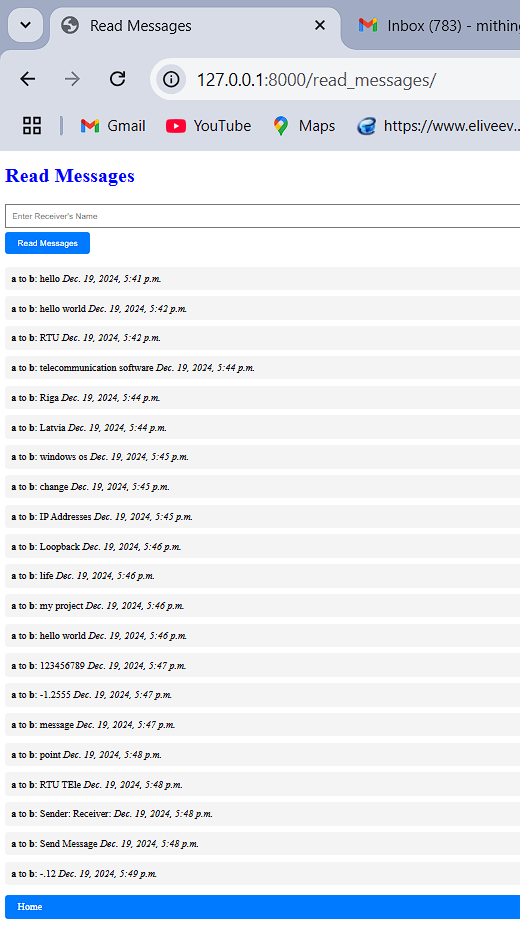
****

Click on read messages

As the receiver name is “b” in the message we sent type “b” and click on read messages



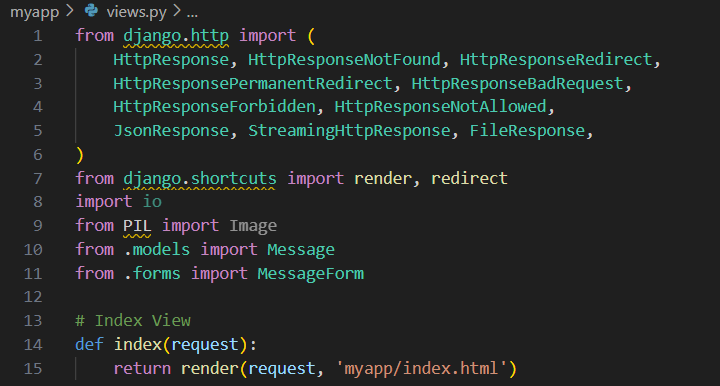
You can see the messages received for “b”

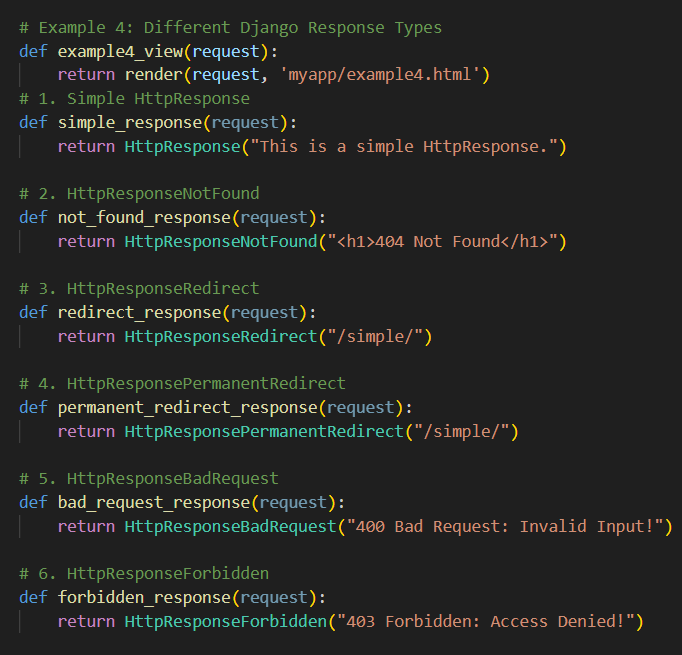
20 latest message records.  


##### Example 4

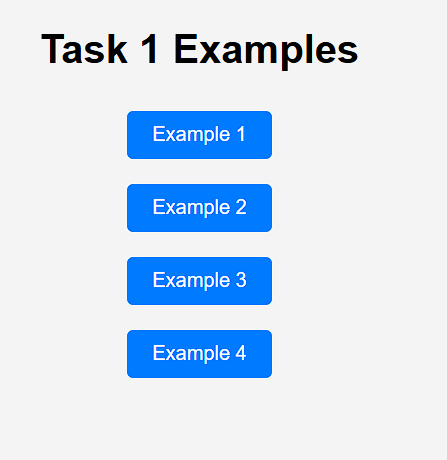
Please try to design Django's different response types, including HttpResponse class and subclasses (10 in total), JsonResponse class, StreamingHttpResponse, and FileResponse class. Please show those response files (Strings, Graphs, Videos......) on your screenshot.

Code:

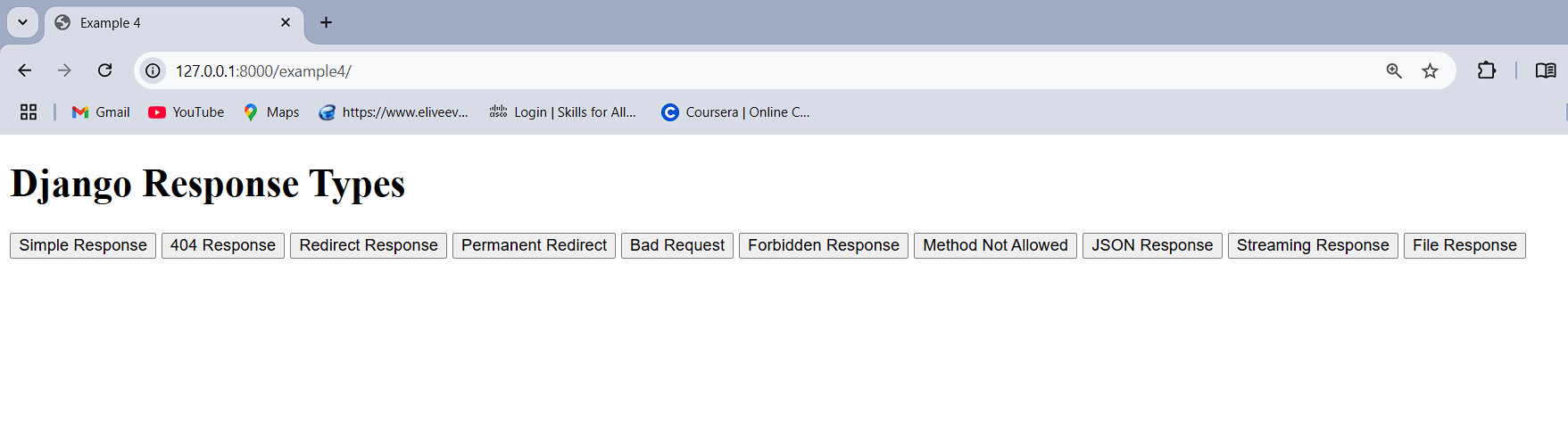








same as in the above link you can click on example 4

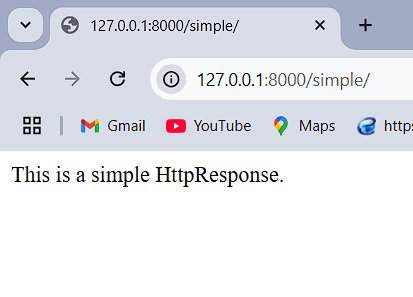


Where you can see the response types

Chick it one-by-one

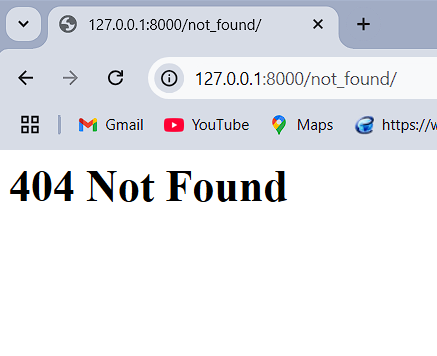
**Simple Response**

* **Description**: The most basic HTTP response. It can include plain text, HTML, or any other content.
* **Common Use Case**: Returning small bits of content directly to the user.

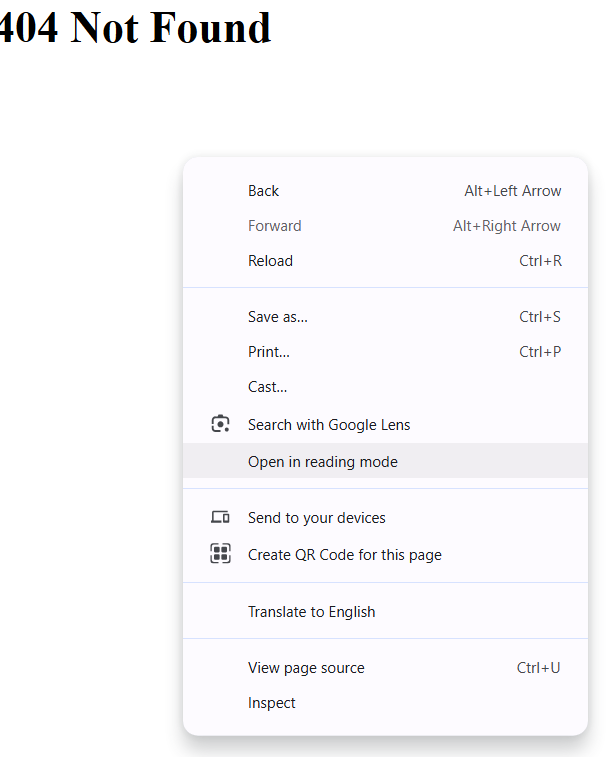


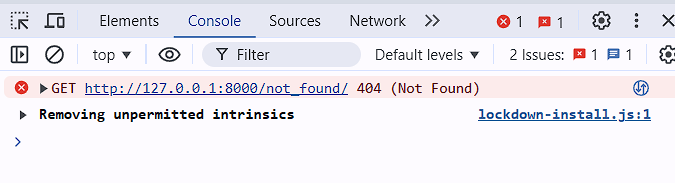
**404 Response**

* **Description**: Indicates the requested resource is not found. Often triggered when a URL doesn’t match any view.
* **Common Use Case**: Returning a custom "Page Not Found" message



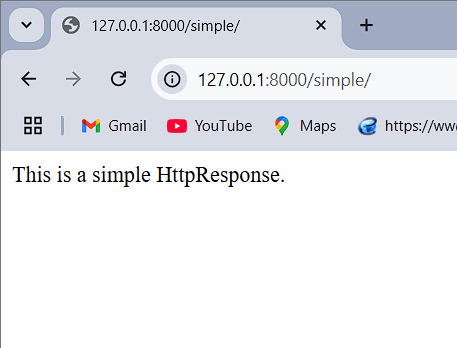
By clicking on the inspect you can see the out response indetail





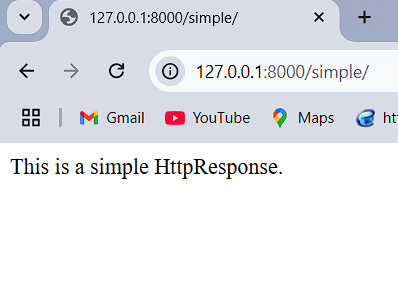
**Redirect Response**

* **Description**: Redirects the user to another URL (temporary redirect with HTTP status code 302).
* **Common Use Case**: Redirecting users after form submission or login.



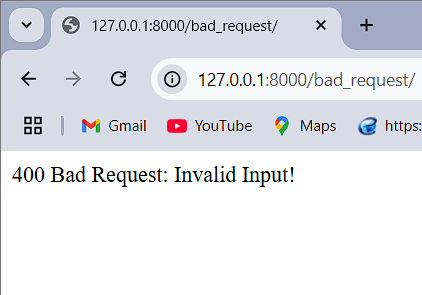
**Permanent Redirect**

* **Description**: Redirects permanently using HTTP status code 301. This informs browsers and search engines to update the URL.
* **Common Use Case**: Handling deprecated URLs or moved resources.



**Bad Request**

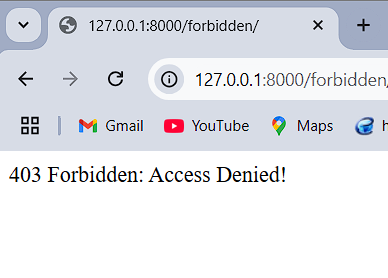
* **Description**: Used when the request is malformed or missing essential data (HTTP 400).
* **Common Use Case**: Validating user input or detecting security issues.

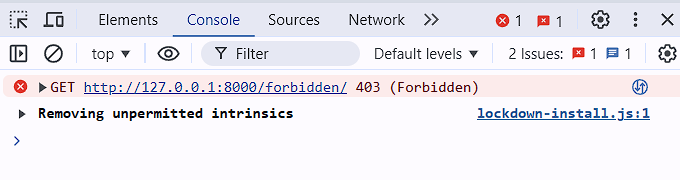




**Forbidden Response**

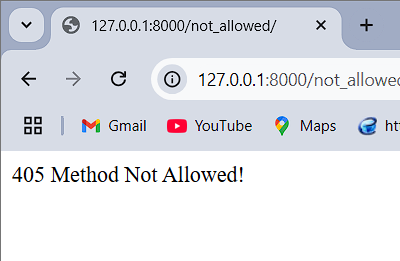
* **Description**: Used when access to a resource is denied (HTTP 403).
* **Common Use Case**: Restricting access based on permissions or roles.

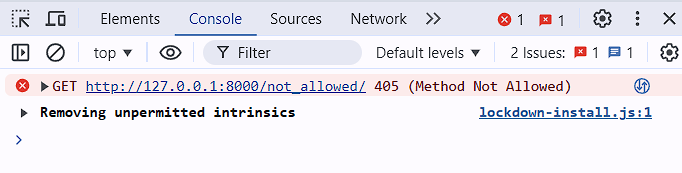




**Method Not Allowed**

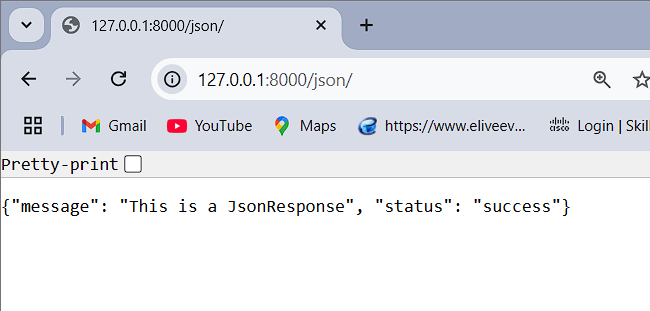
* **Description**: Indicates that the HTTP method used (e.g., POST, GET) is not allowed (HTTP 405).
* **Common Use Case**: Restricting certain endpoints to specific HTTP methods





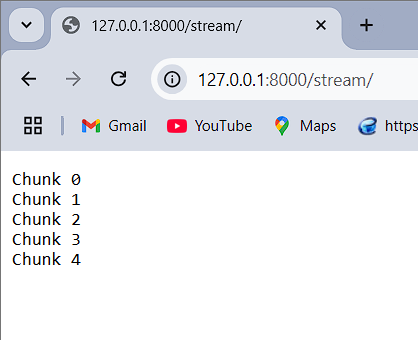
**JSON Response**

* **Description**: Returns data in JSON format, commonly used in APIs.
* **Common Use Case**: Serving data to frontend apps or external systems.



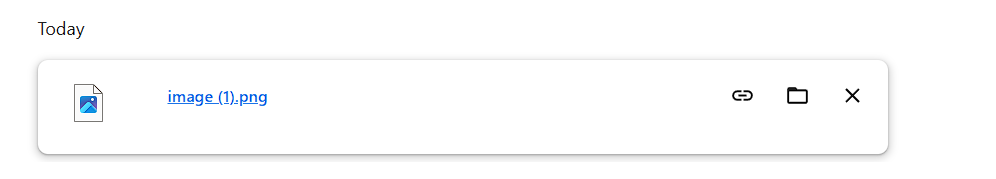
**Streaming Response**

* **Description**: Streams content to the client incrementally instead of sending it all at once.
* **Common Use Case**: Streaming large files, logs, or dynamic content.



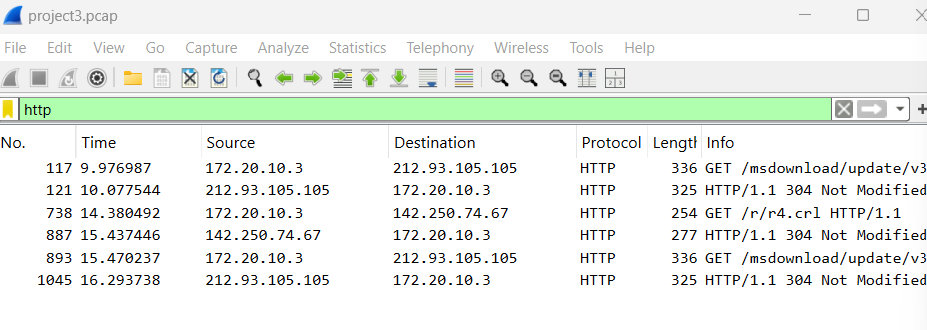
**File Response**

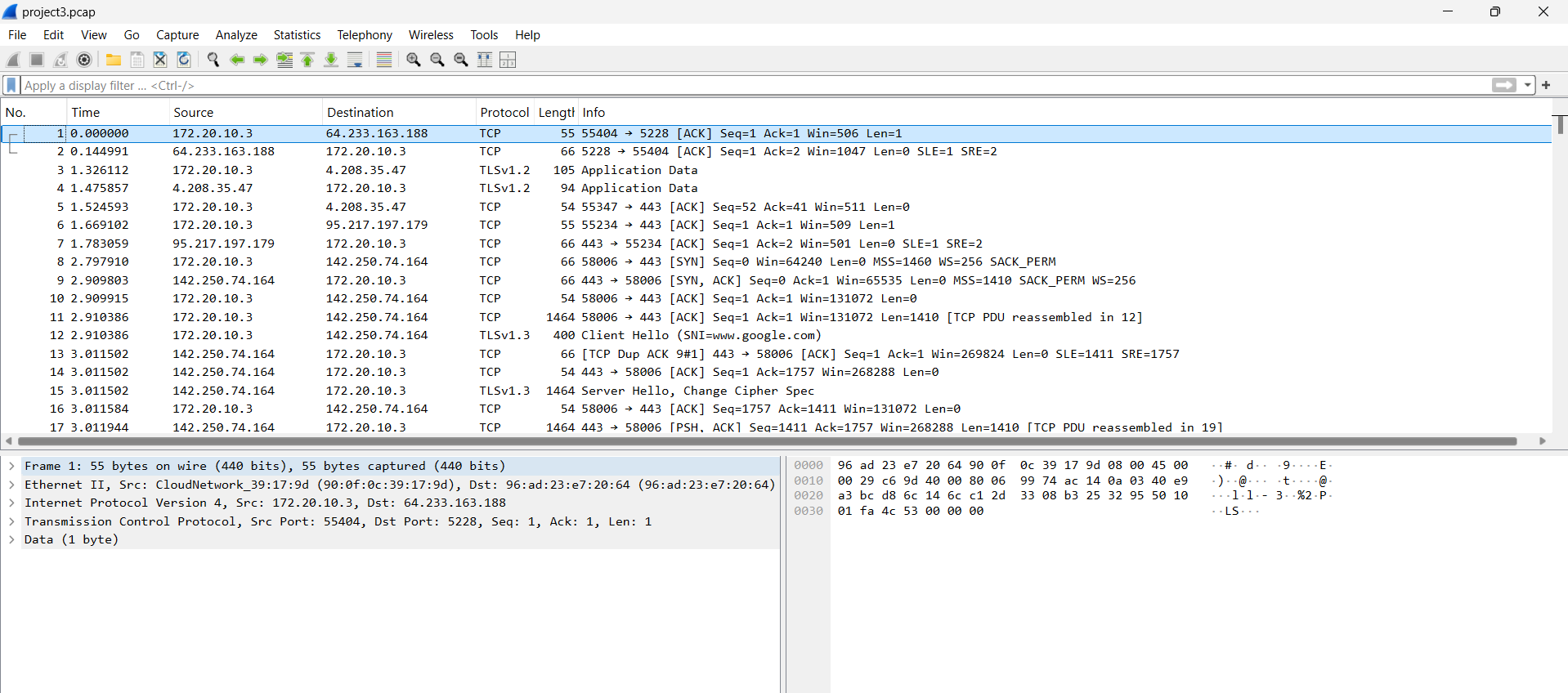
* **Description**: Sends a file to the client for download or viewing.
* **Common Use Case**: Serving static or dynamic files like PDFs, images, or reports.



**Task 2: Network Traffic Capture and Analysis**   
I have tried to capture the file with wire shark but the capture is unsuccessful due to my laptop drawback.

I tried to capture it with wifi but no result. But also I am senting the file of wireskark of wifi capture.





GitHub : <https://github.com/Mithin-RTU/Telecommunication_software_practical_3>