Andre Hei Wang Law

4017 5600

COEN 366 – FL-X

Socket Programming + Wireshark Assignment 1

1. **“Socket\_programming\_assignment.pdf” Assignment**

HTML Code - “coen366.html”:

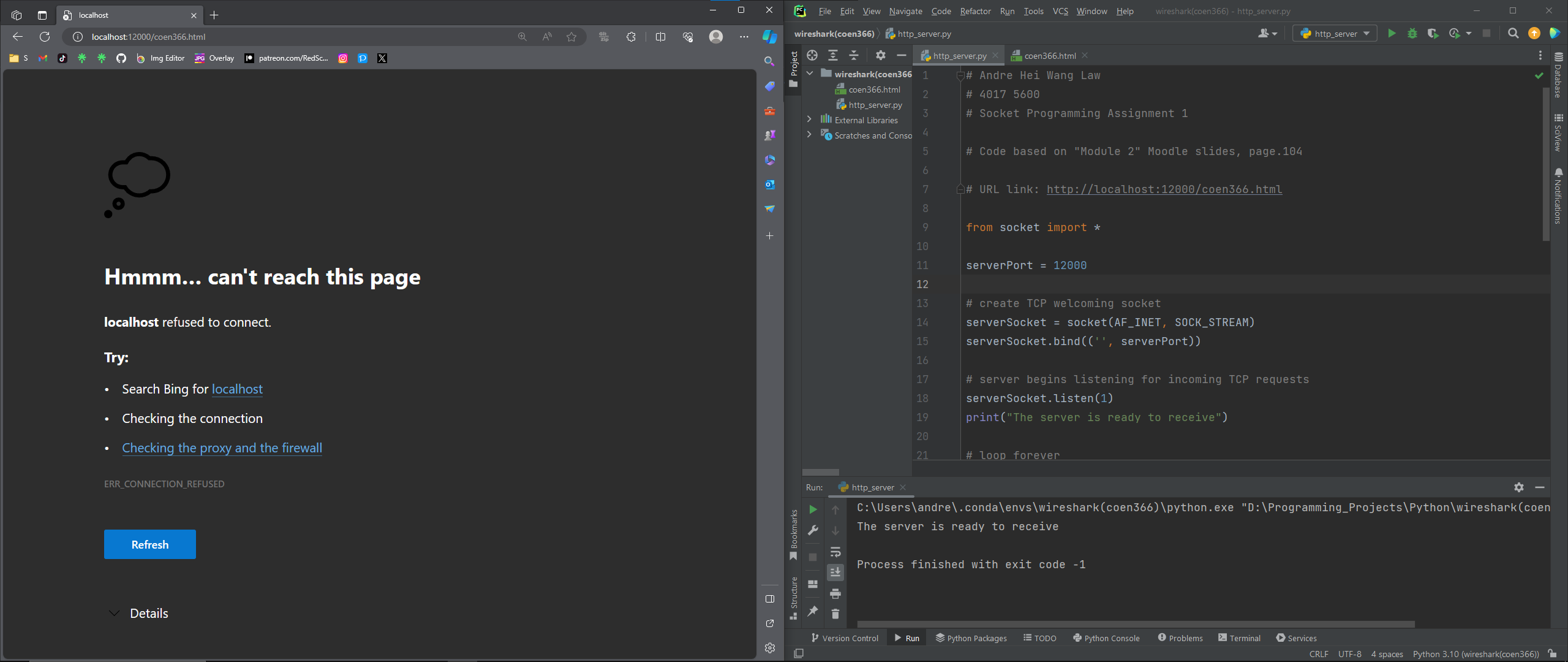
<!DOCTYPE html>  
<**html** lang="en">  
<**head**>  
 <**meta** charset="UTF-8">  
 <**meta** content="IE=edge" http-equiv="X-UA-Compatible">  
 <**meta** content="width=device-width, initialscale=1.0" name="viewport">  
 <**title**>Document</**title**>  
</**head**>  
<**body**>  
<**h1**>Welcome to COEN 366</**h1**>  
<**p**>Course info:</**p**>  
<**ul**>  
 <**li**>Given by Prof. Chadi Assi</**li**>  
 <**li**>Prof's email: chadi.assi@concordia.ca</**li**>  
 <**hr**>  
 <**li**>There are three TAs for this course:<**br**/>  
 <**ul**>  
 <**li**>Shreya Khisa</**li**>  
 <**li**>Ali Amhaz</**li**>  
 <**li**>Y A Joarder</**li**>  
 </**ul**>  
 </**li**>  
 <**hr**>  
</**ul**>  
</**body**>  
</**html**>

Python Code - “http\_server.py”:

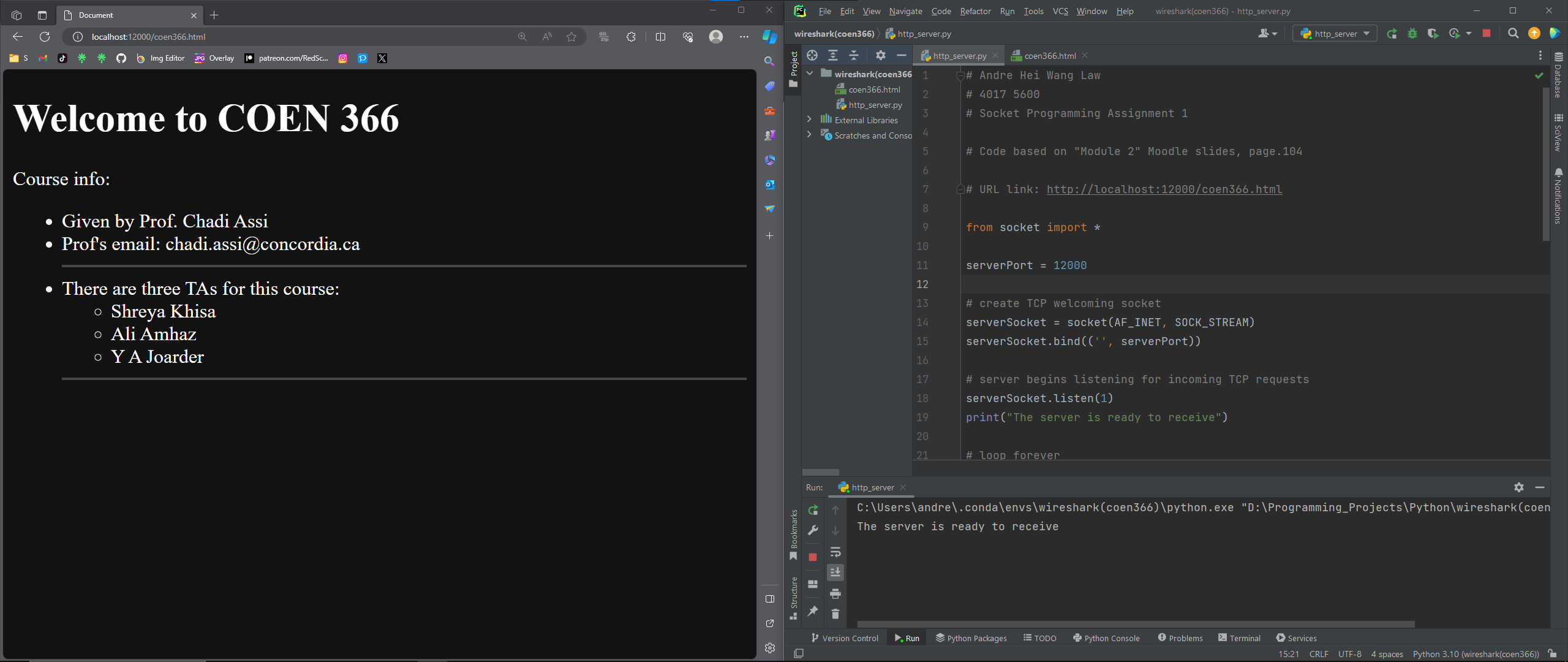
# Andre Hei Wang Law  
# 4017 5600  
# Socket Programming Assignment 1  
  
# Code based on "Module 2" Moodle slides, page.104  
  
# URL link: http://localhost:12000/coen366.html  
  
from socket import \*  
  
serverPort = 12000  
  
# create TCP welcoming socket  
serverSocket = socket(AF\_INET, SOCK\_STREAM)  
serverSocket.bind(('', serverPort))  
  
# server begins listening for incoming TCP requests  
serverSocket.listen(1)  
print("The server is ready to receive")  
  
# loop forever  
**while** True:  
 # server waits on accept() for incoming  
 # requests, new socket created on return  
 connectionSocket, addr = serverSocket.accept()  
  
 # read bytes from socket (but not address as in UDP)  
 request = connectionSocket.recv(1024).decode()  
  
 # well-formed request (return error otherwise)  
 **try**:  
 **with** open('coen366.html', 'r') **as** htmlFile:  
 htmlContent = htmlFile.read()  
 response = "HTTP/1.1 200 OK\r\n\r\n" + htmlContent  
 except FileNotFoundError: # error when file doesn't exist  
 response = "HTTP/1.1 404 Not Found\r\n\r\nFile not found"  
 except PermissionError: # error when permissions are set properly  
 response = "HTTP/1.1 403 Forbidden\r\n\r\nPermission denied"  
  
 # send HTTP response to the client (browser)  
 connectionSocket.send(response.encode())  
  
 # close the connection  
 connectionSocket.close()

Before Running the Python Code:

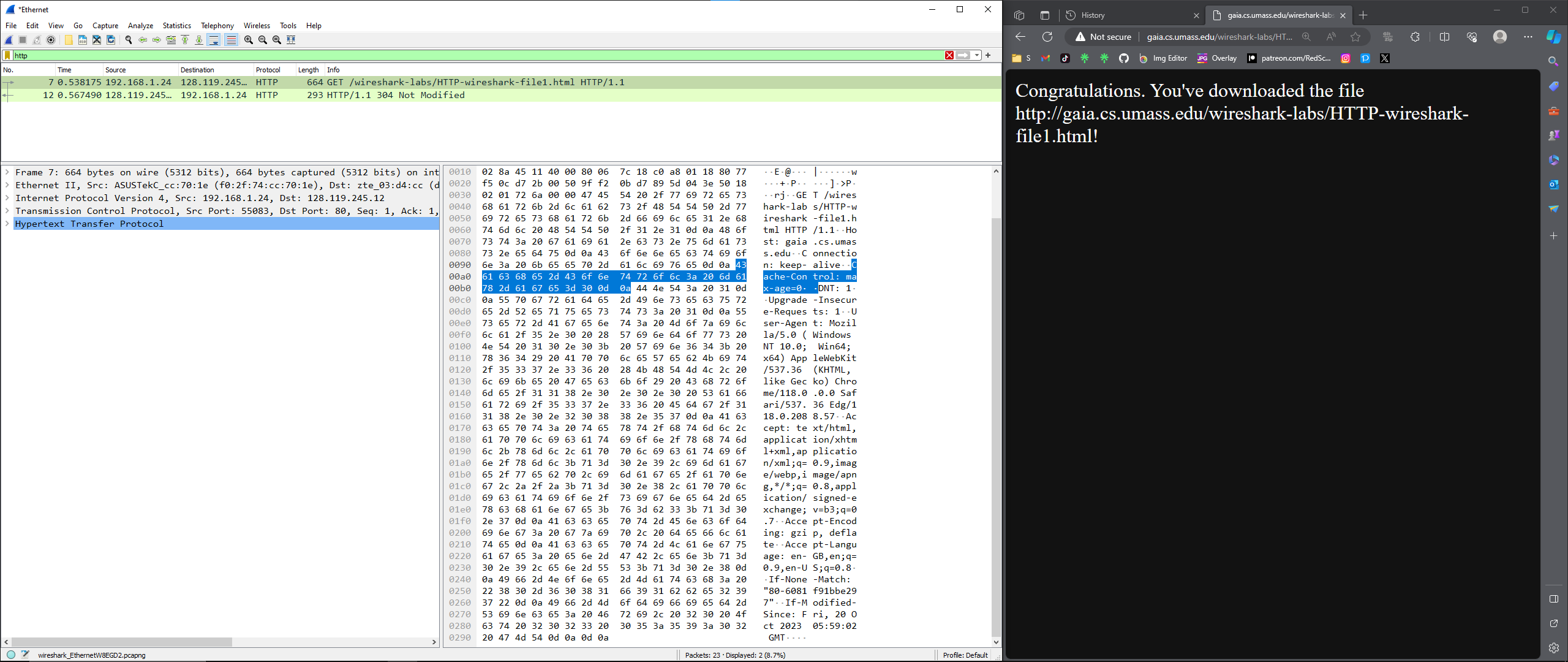
* Url: http://localhost:12000/coen366.html



After Running the Python Code:



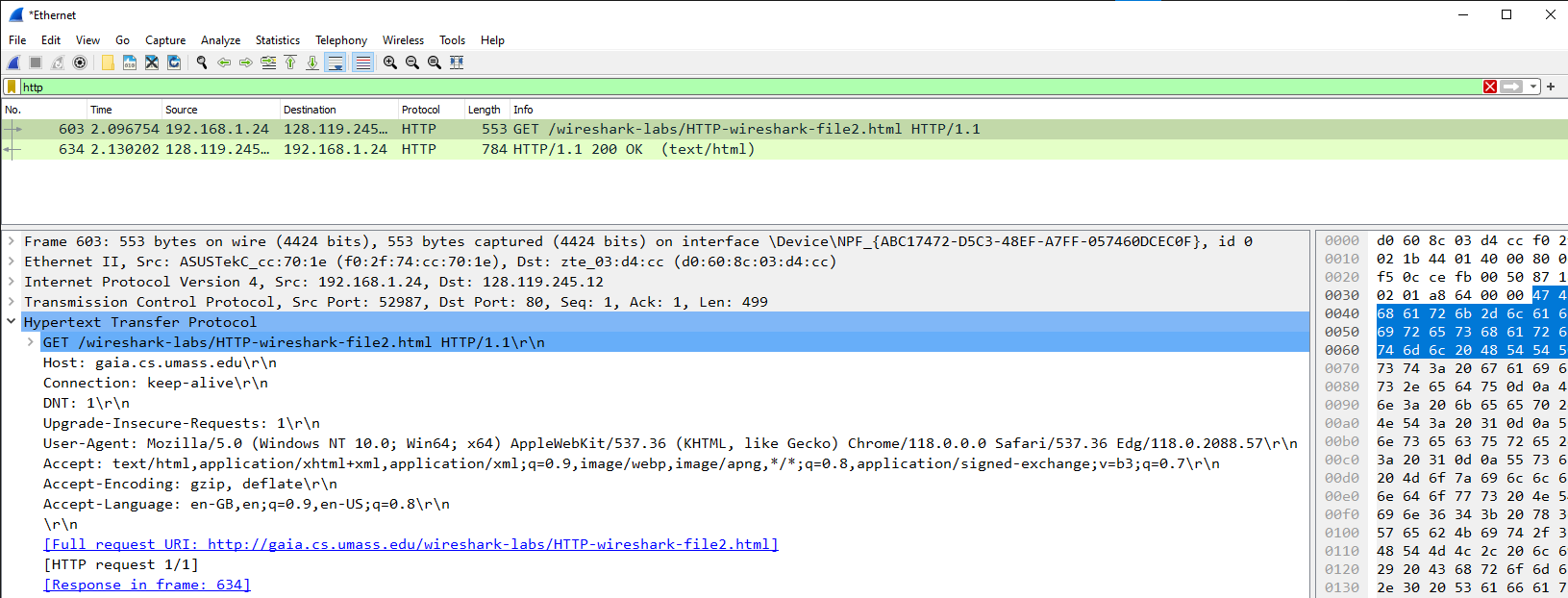
1. **“Wireshark\_assignment\_1.pdf” Assignment**

2.1 The Basic HTTP GET/response interaction 

The above image is the baseline output after waiting a minute and opening gaia website.

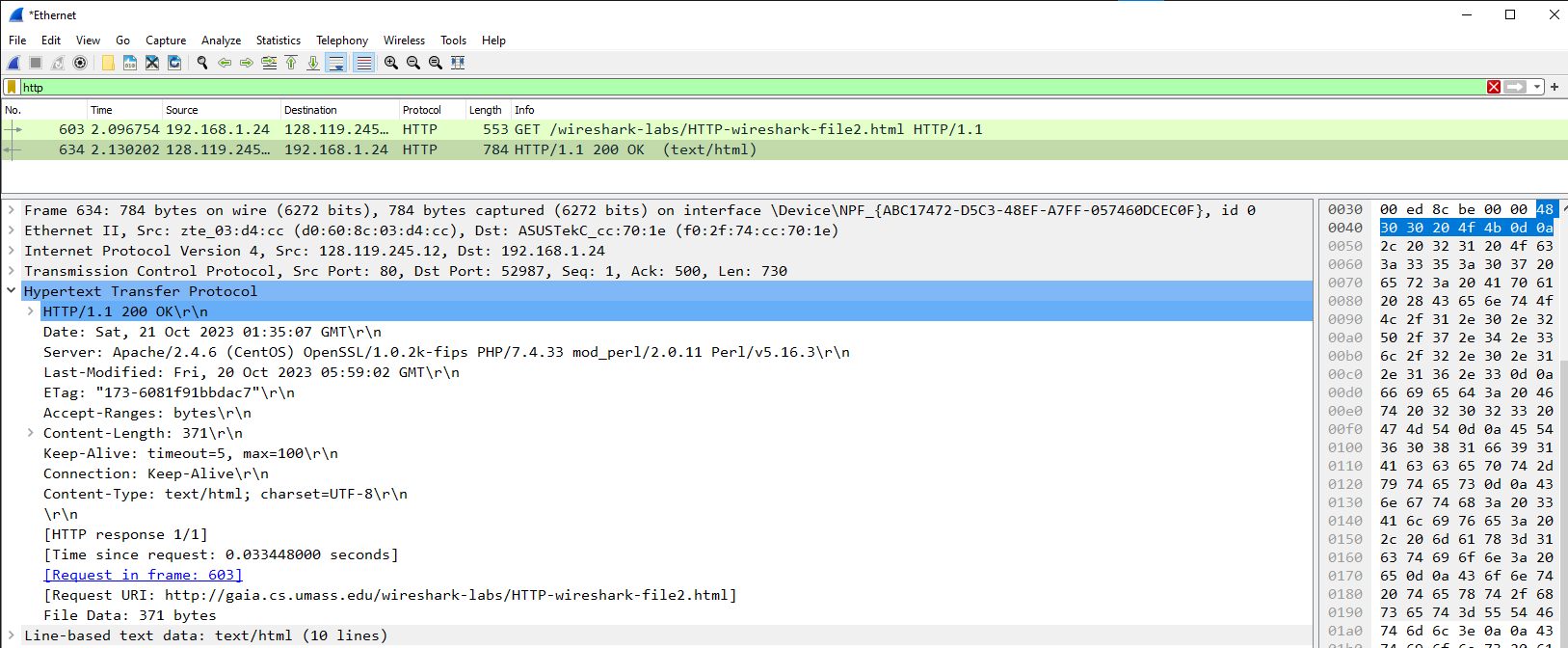
**1. Is your browser running HTTP ver. 1.0 or 1.1? What ver. of HTTP is the server running?**

From HTTP GET, Hypertext Transfer Protocal, my browser is running version 1.1.





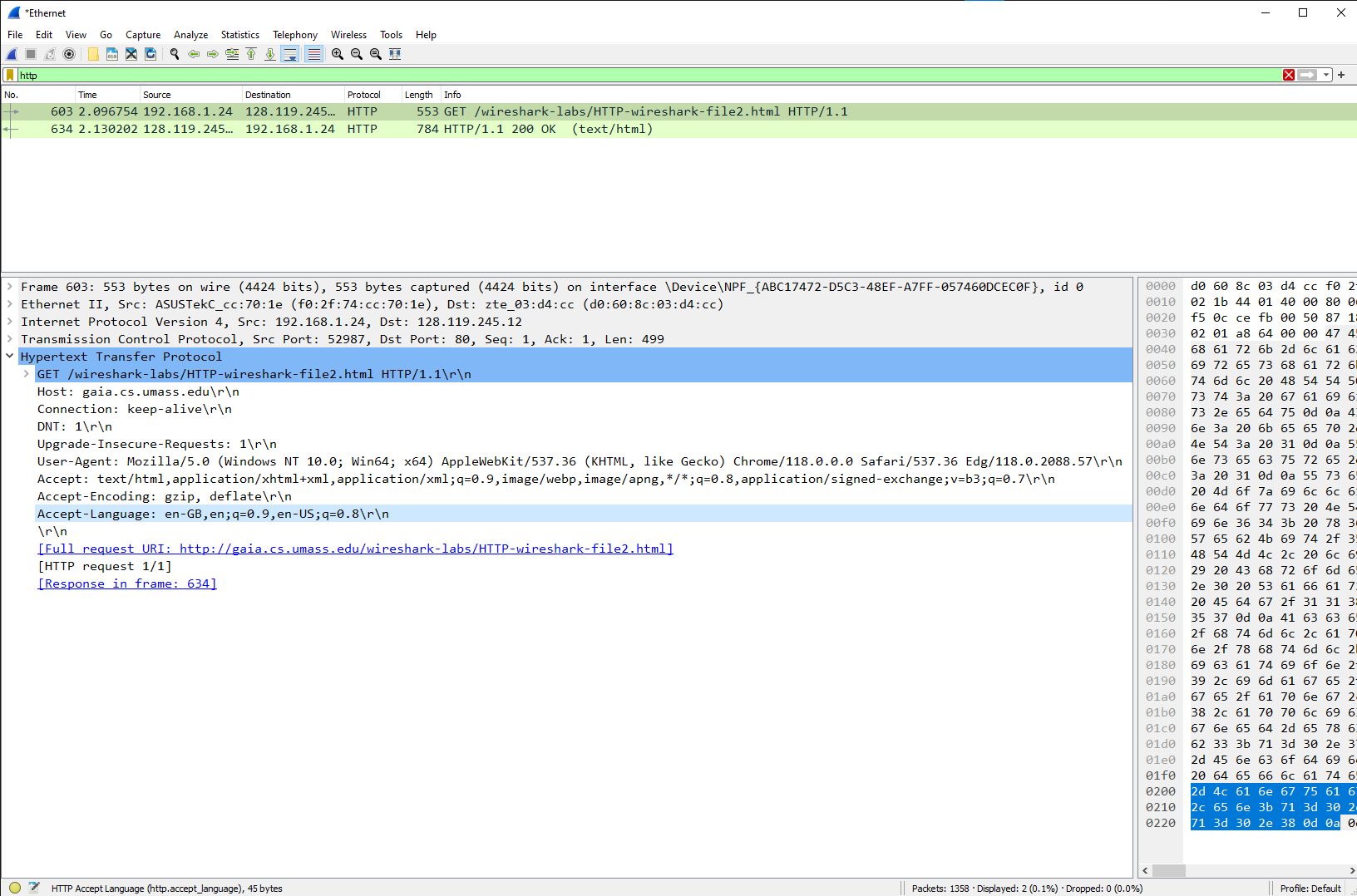
From HTTP response, Hypertext Transfer Protocal, the server is running version 1.1.





**2. What languages (if any) does your browser indicate that it can accept to the server?**

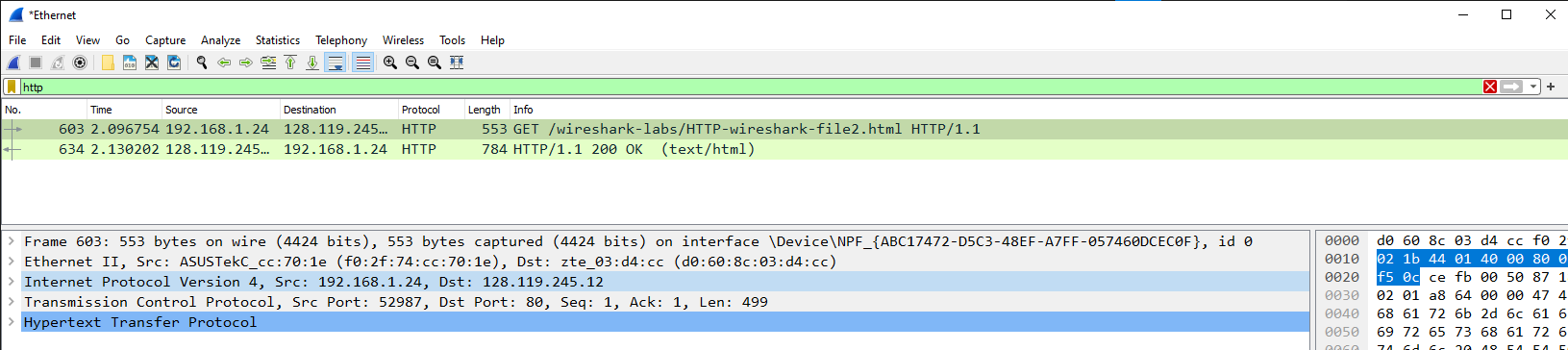
From HTTP GET, Hypertext Transfer Protocol, it accepts: en-GB,en;q=0.9,en-US;q=0.8\r\n





**3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?**

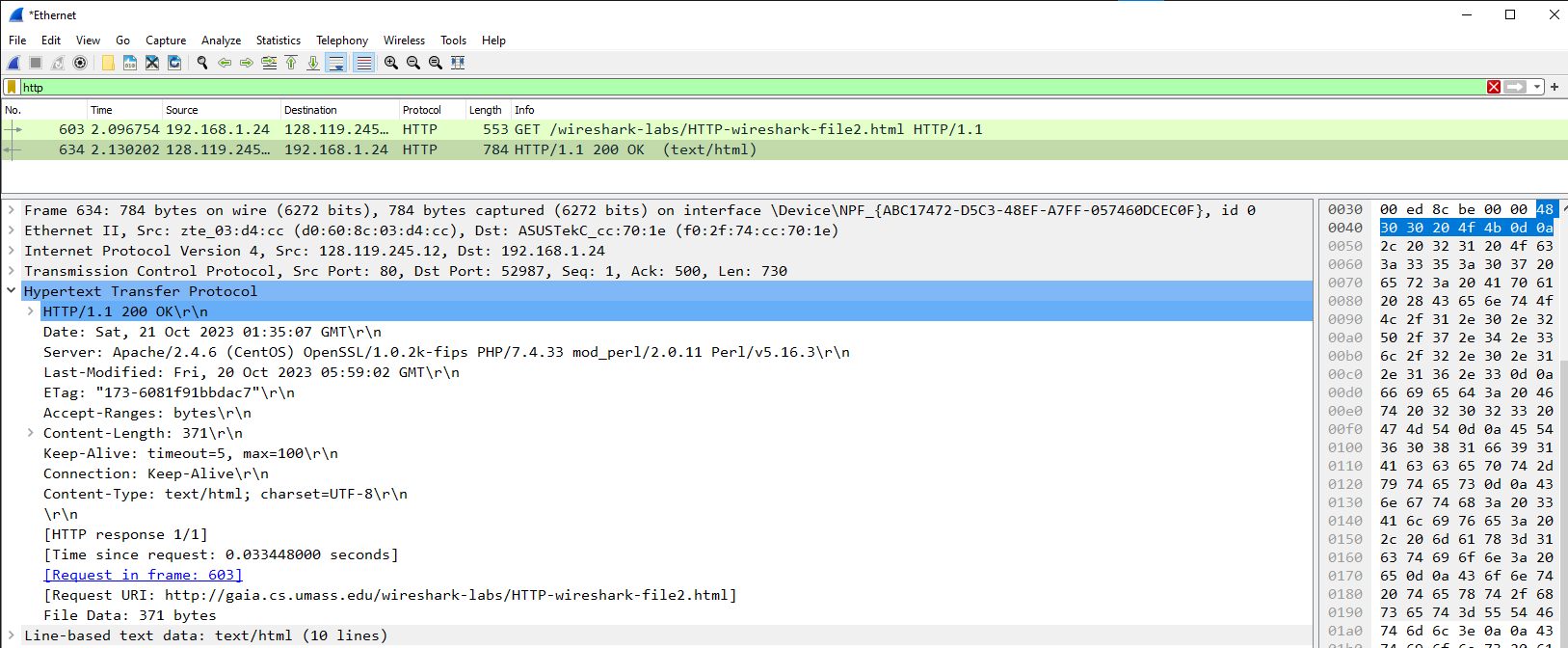
From HTTP GET, my computer’s IP is Src: 192.168.1.24 and gaia’s is Dst: 128.119.245.12





**4. What is the status code returned from the server to your browser?**

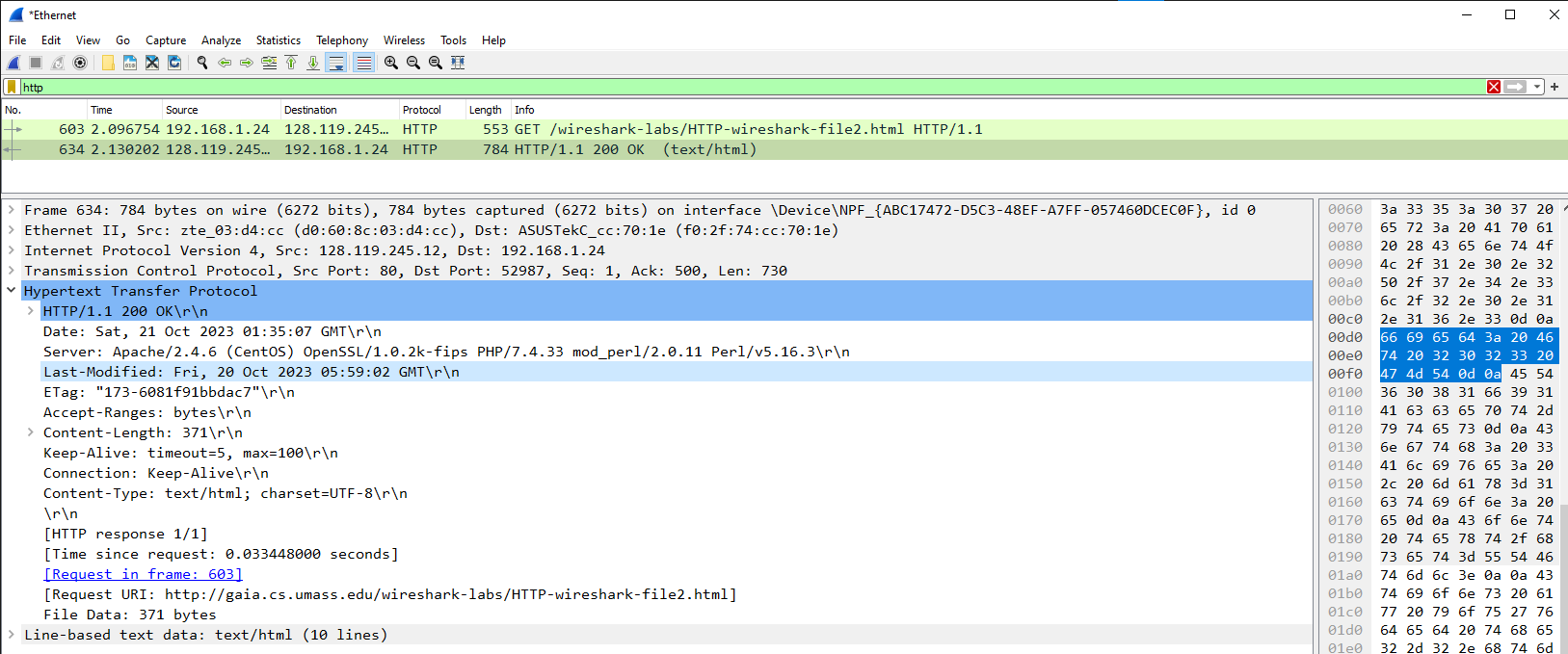
From HTTP response, Hypertext Transfer Protocol, the status code returned from the server to my browser is 200 OK.





**5. When was the HTML file that you are retrieving last modified at the server?**

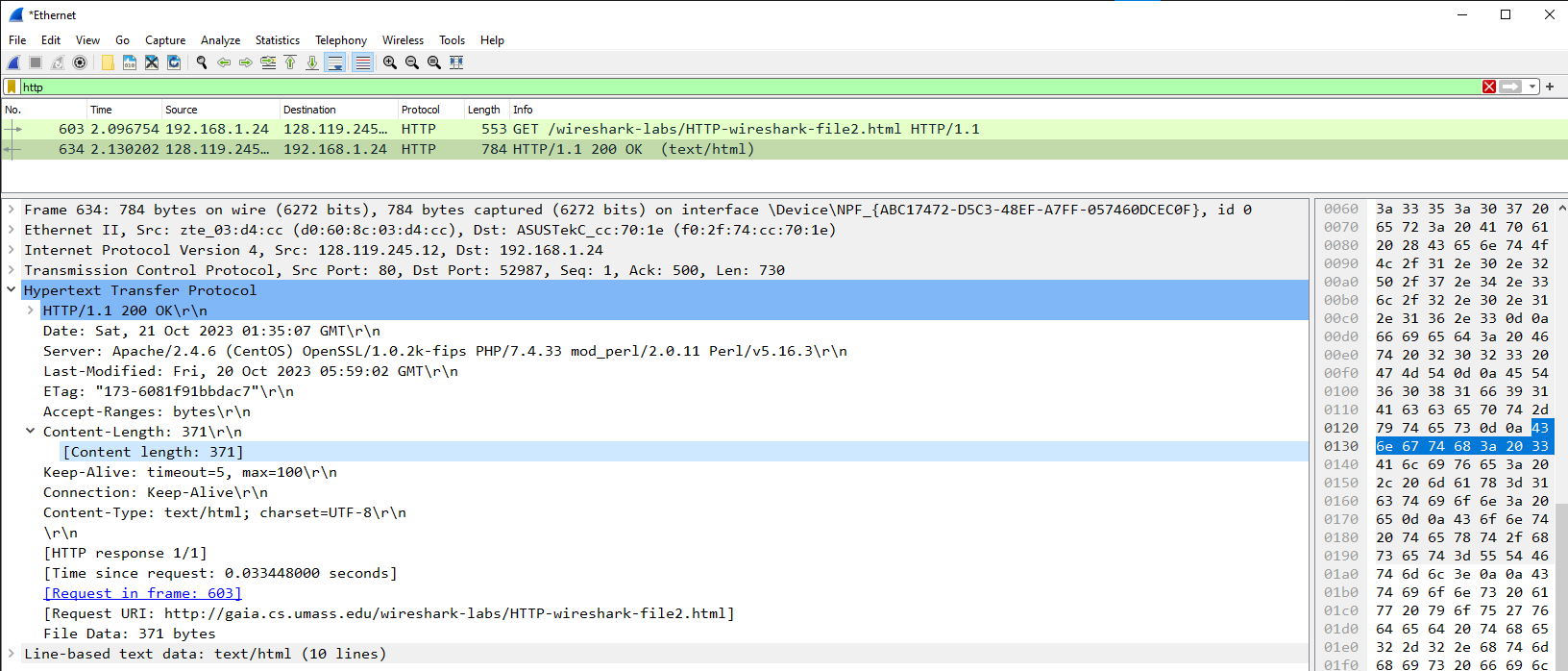
From HTTP response, Hypertext Transfer Protocol, the last modified time is Fri, 20 Oct 2023 05:59:02 GMT.





**6. How many bytes of content are being returned to your browser?**

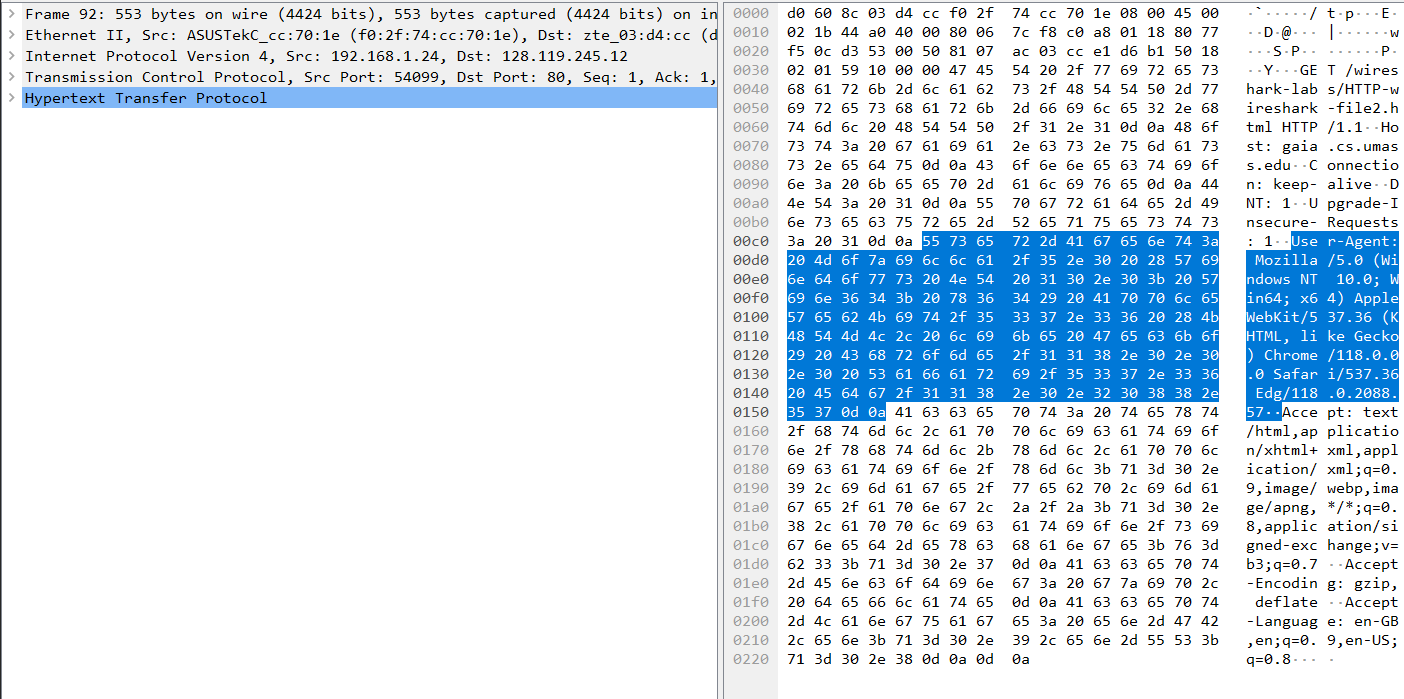
From HTTP response, Hypertext Transfer Protocol, the content length is 371 bytes.



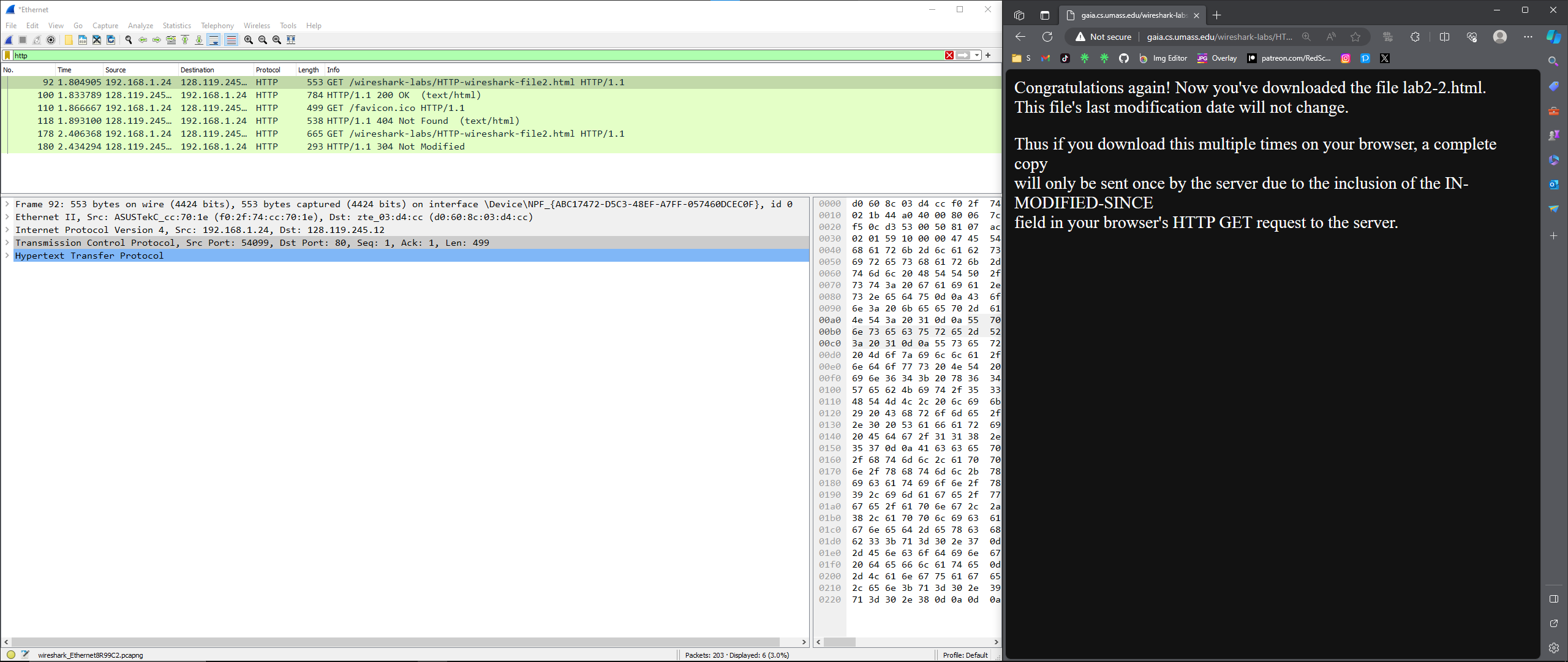


**7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.**

No, when inspecting the raw data in the packet content window, there are no additional headers within the data displayed in the packet-listing window.



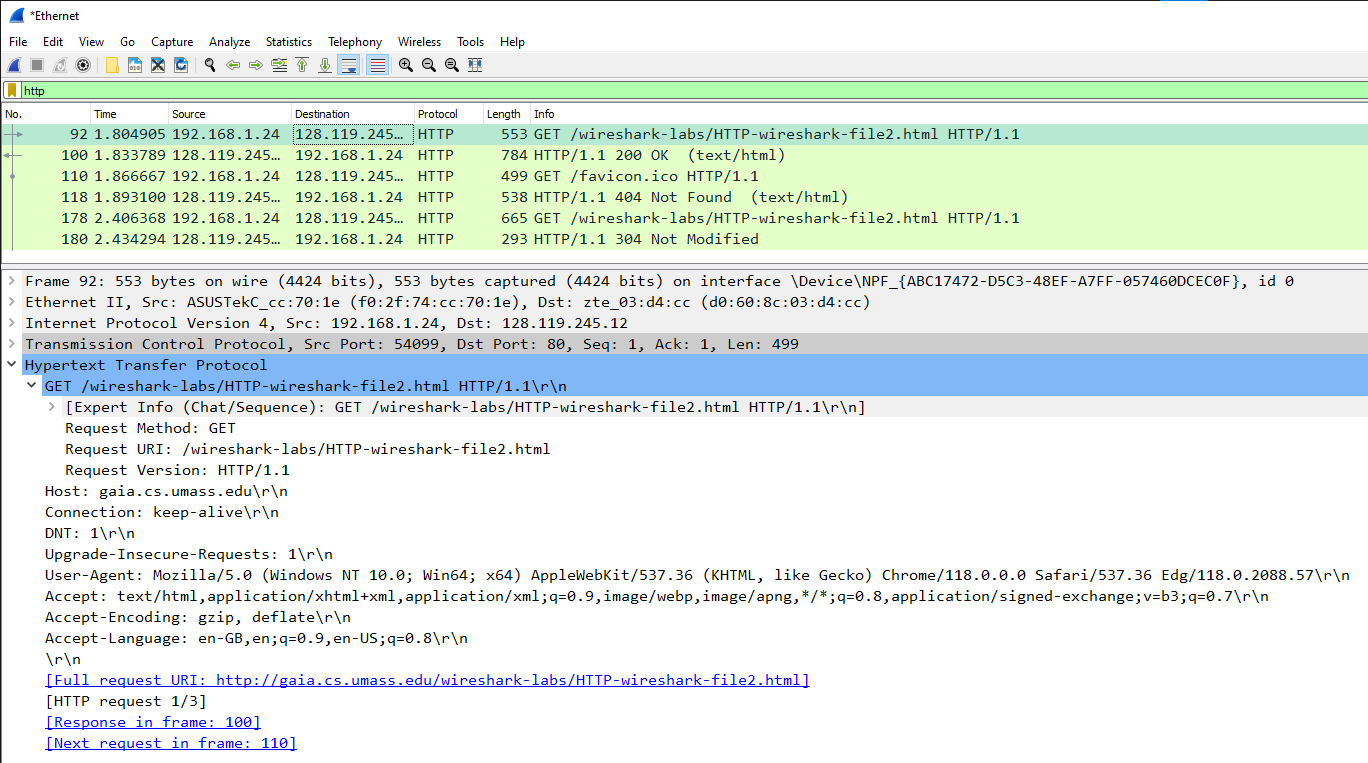
2.2 The HTTP CONDITIONAL GET/response interaction



The above image is the baseline output opening gaia website and reloading the website.

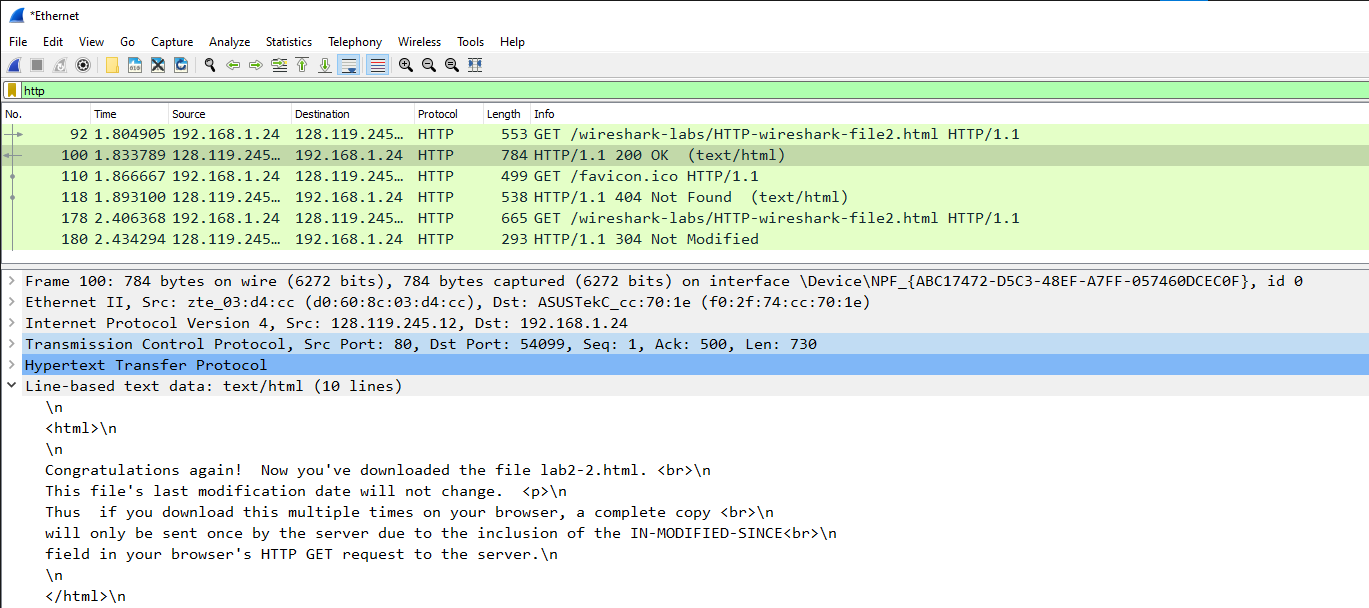
**8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?**

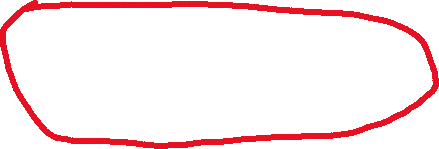
From the first HTTP GET, Hypertext Transfer Protocol, it doesn’t have any “IF-MODIFIED-SINCE” line.



**9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?**

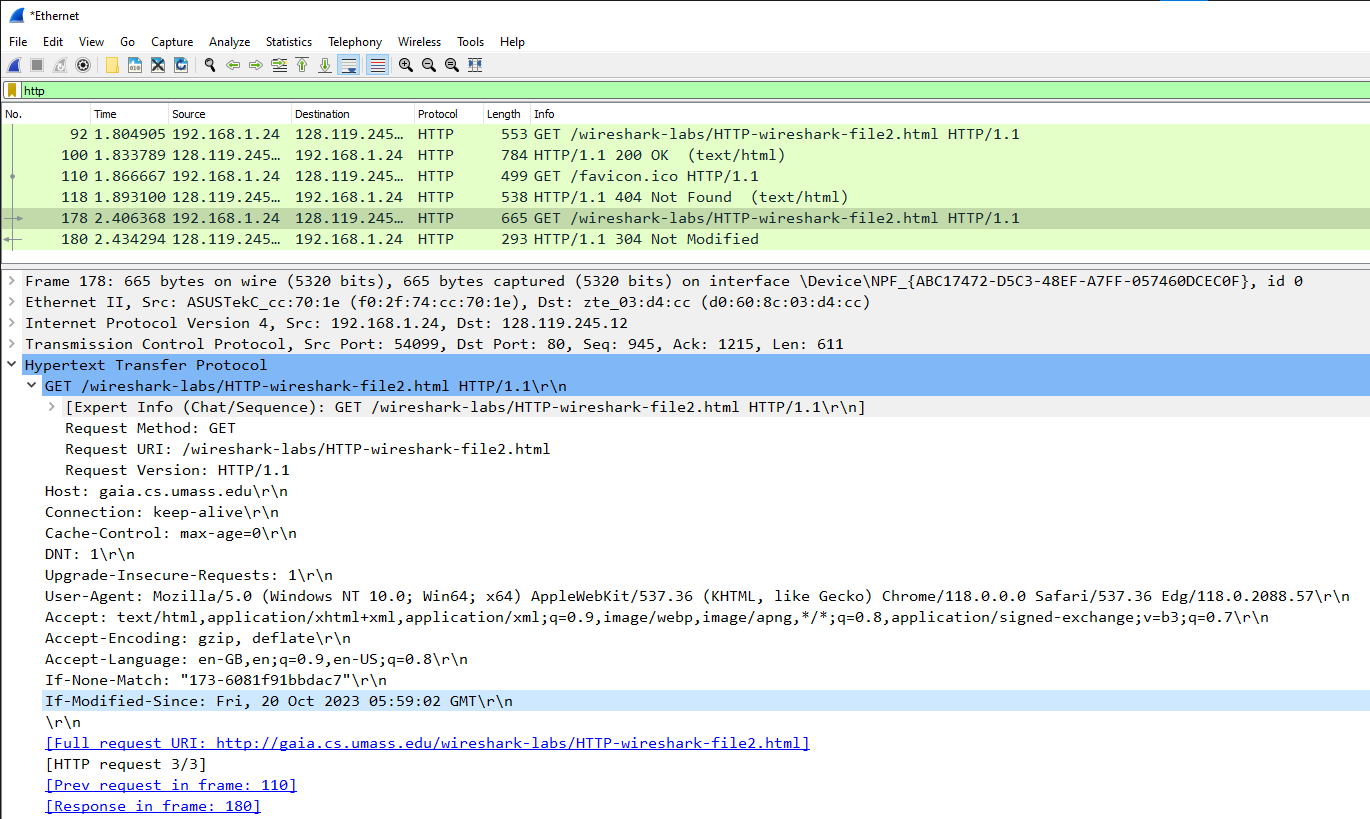
From the first HTTP response, Line-based text data, yes, the server did explicitly return the contents of the file. It is an HTML code.





**10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?**

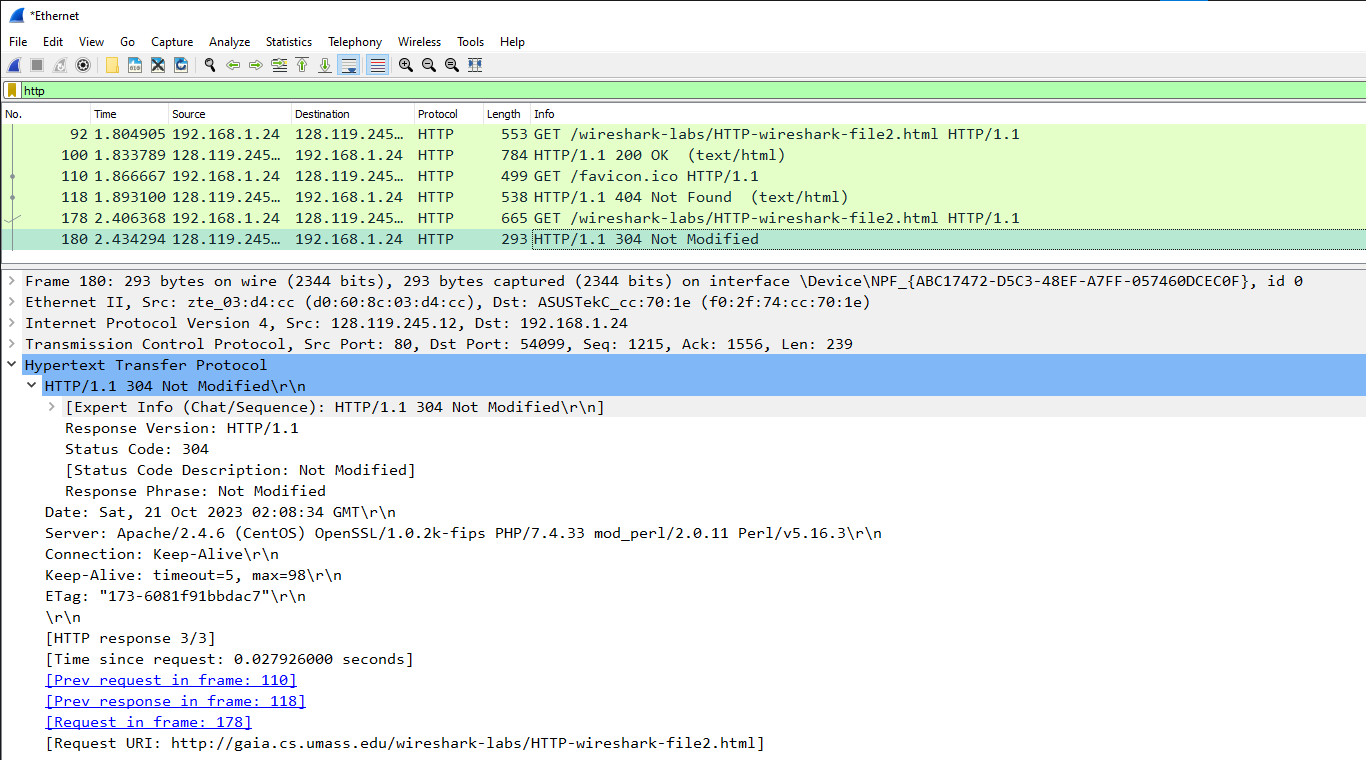
From the second HTTP GET, Hypertext Transfer Protocol, it has the “IF-MODIFIED-SINCE” line. The information that follows this header is the date last modified, which is Fri, 20 Oct 2023 05:29:02 GMT\r\n.





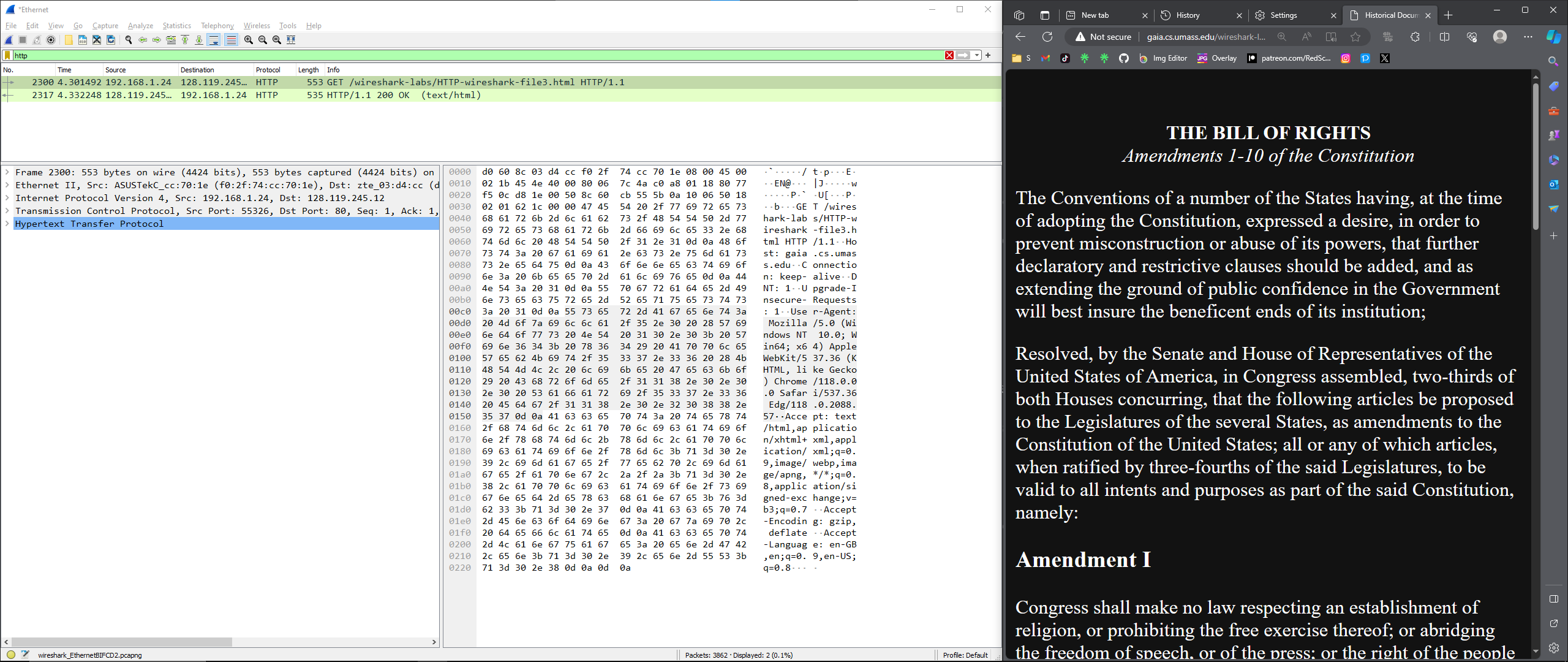
**11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.**

From the second HTTP response, Hypertext Transfer Protocol, the status is “304 Not Modified”. With no modification, no content is explicitly returned from the server (no Line-Based text data).





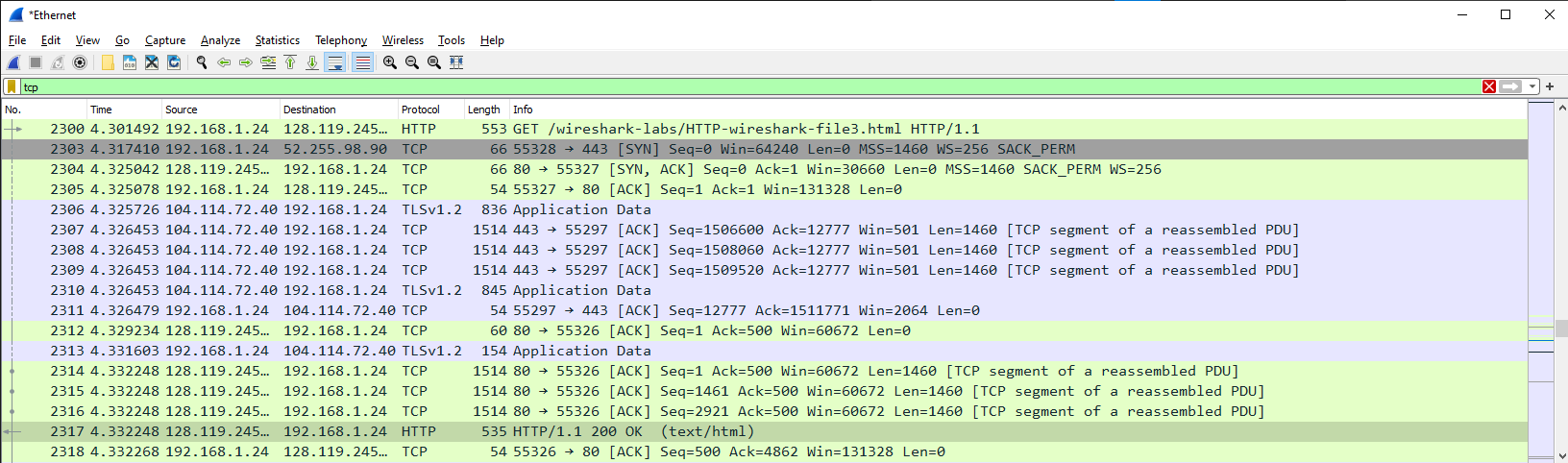
2.3 Retrieving Long Documents



The above image is the baseline output after running gaia website with a long HTML file.

**12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?**

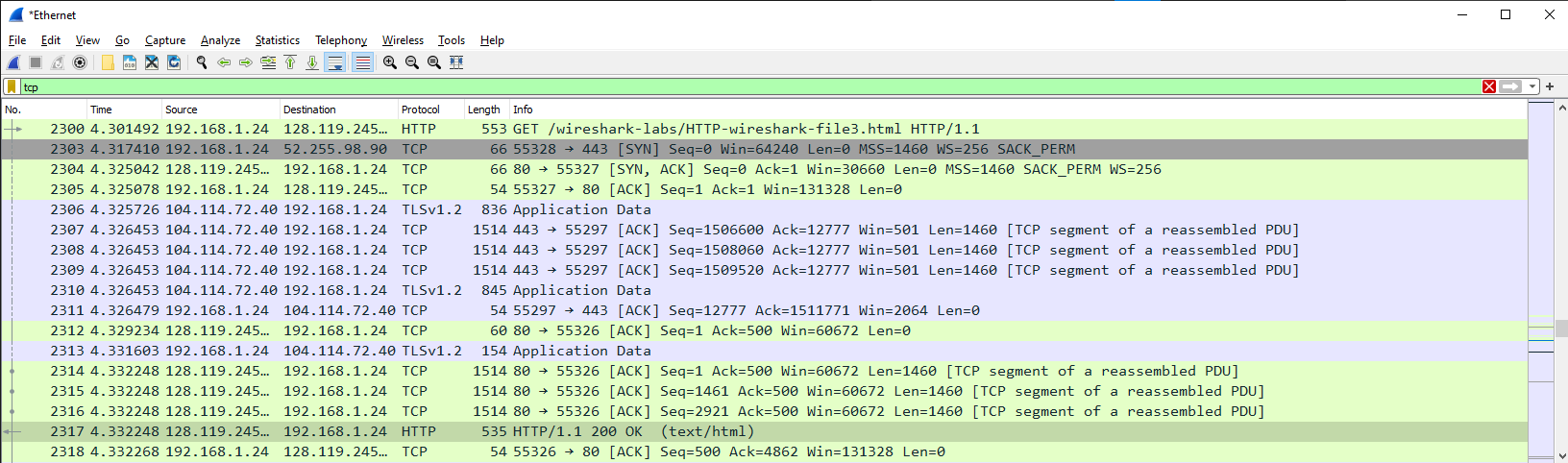
There are only one HTTP GET request. The packet number is No.2300.





**13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?**

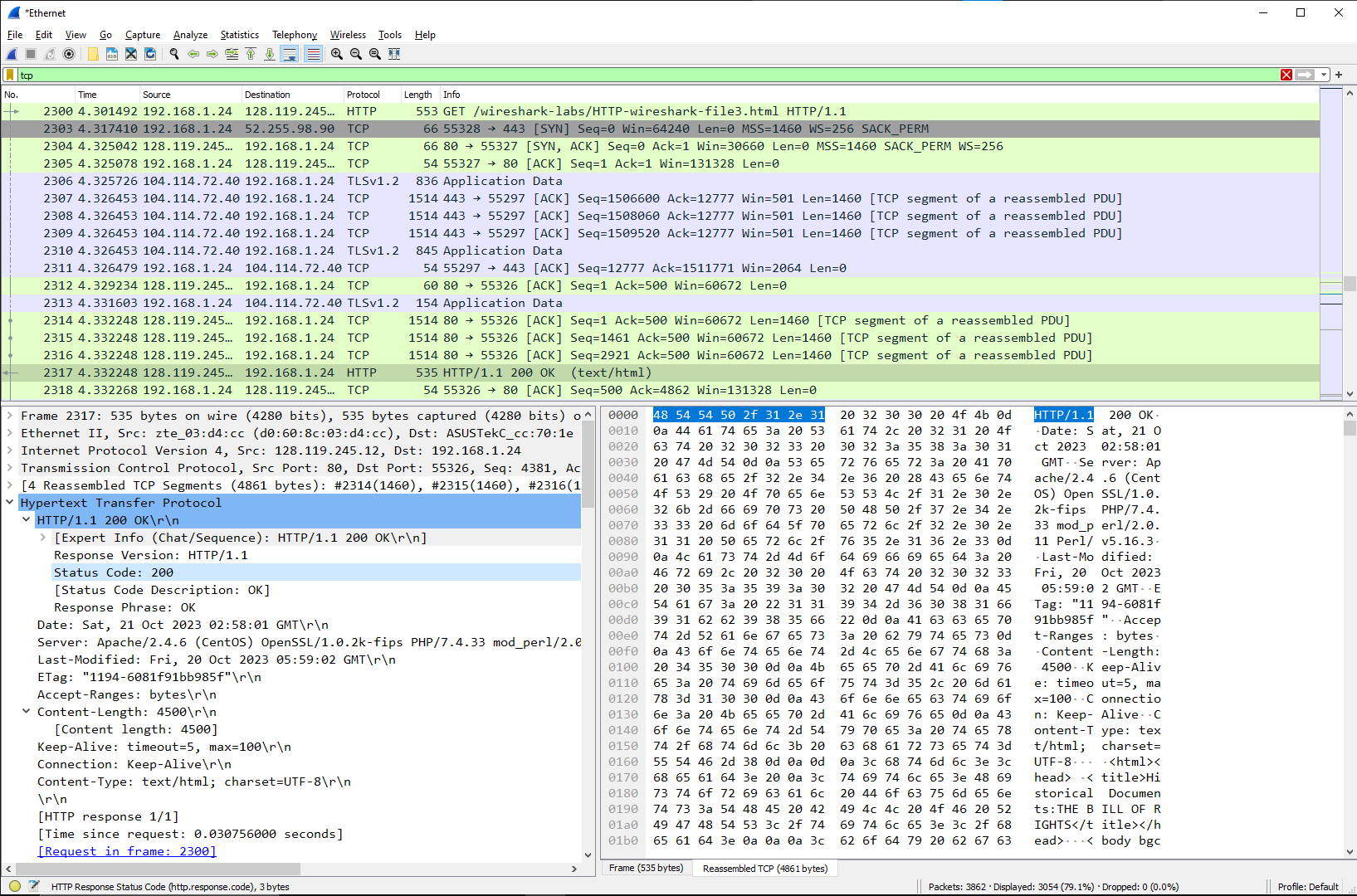
The packet number that contains the status code and phrase is No.2317.





**14. What is the status code and phrase in the response?**

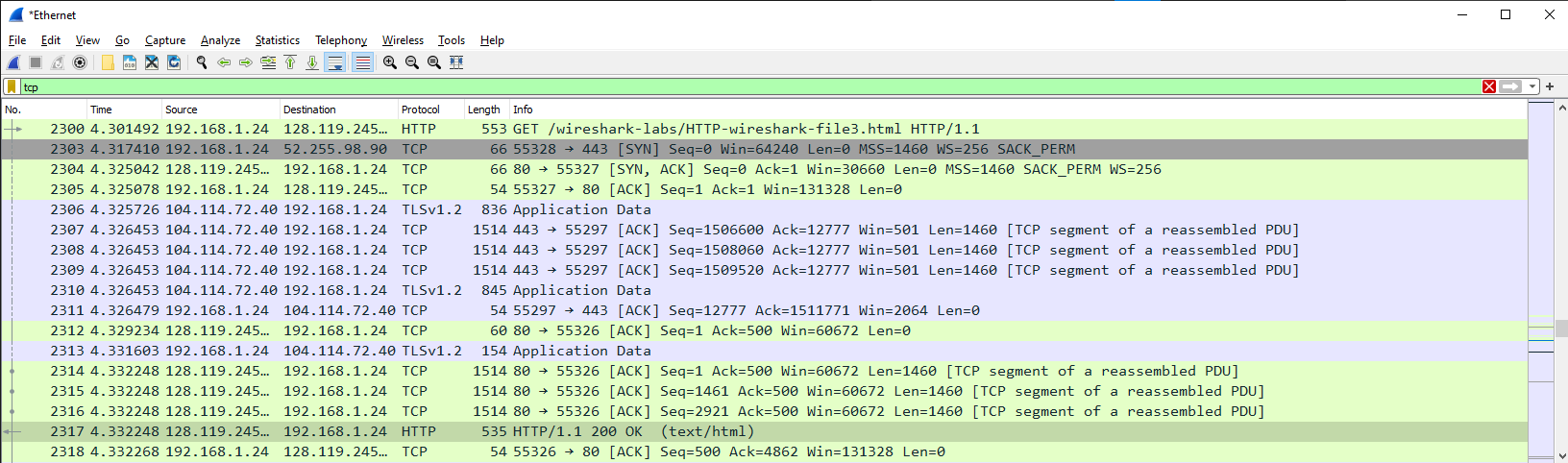
From HTTP response, Hypertext Transfer Protocol, the status code and phrase is 200 OK.





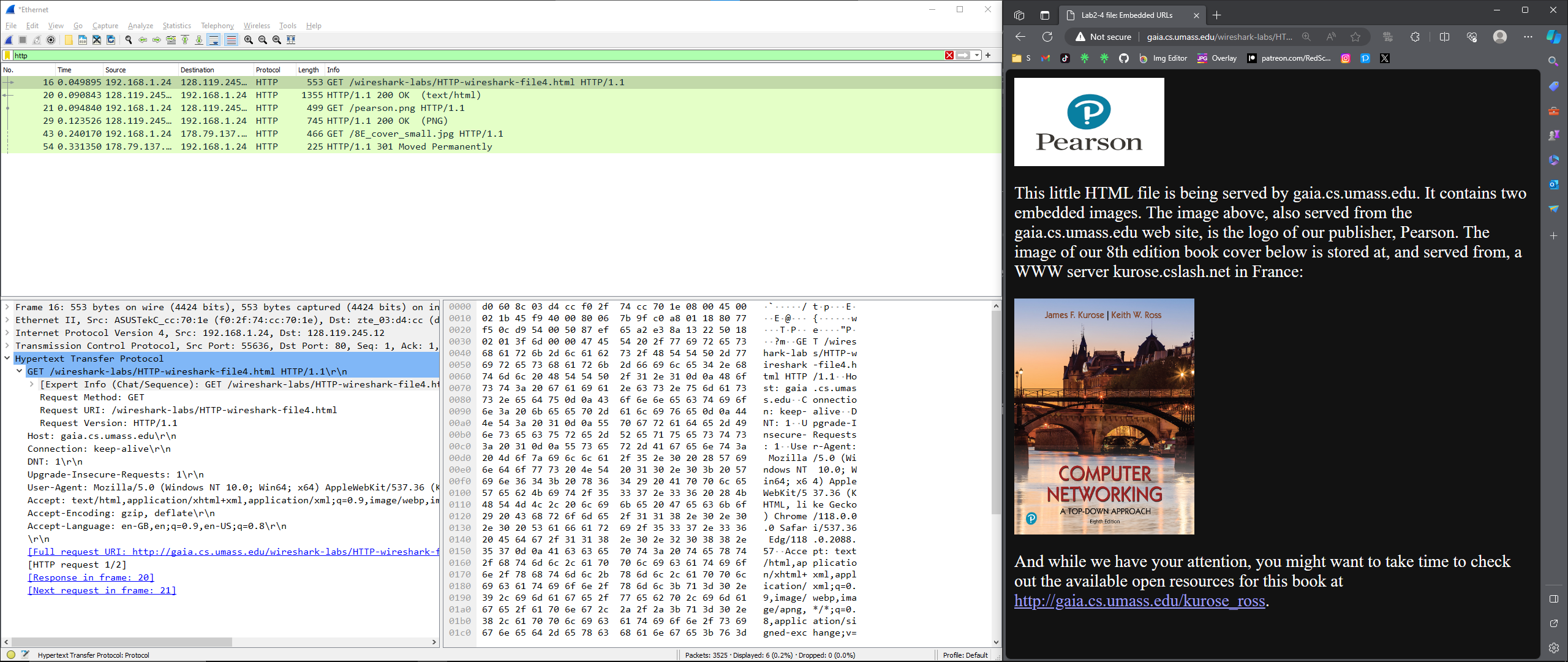
**15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?**

When filtering for TCP and looking for “TCP segment of a reassembled PDU”, we observe 6 of them. Their number are No.2307, No.2308, No.2309, No.2314, No.2315, No.2316.





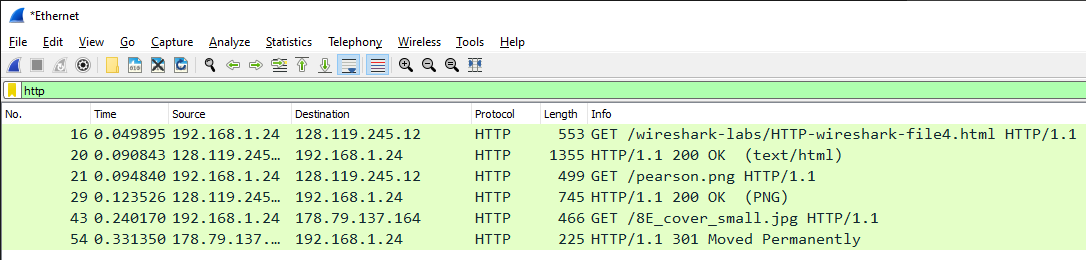
2.4 HTML Documents with Embedded Objects



The above image is the baseline output of opening gaia website with embedded object.

**16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?**

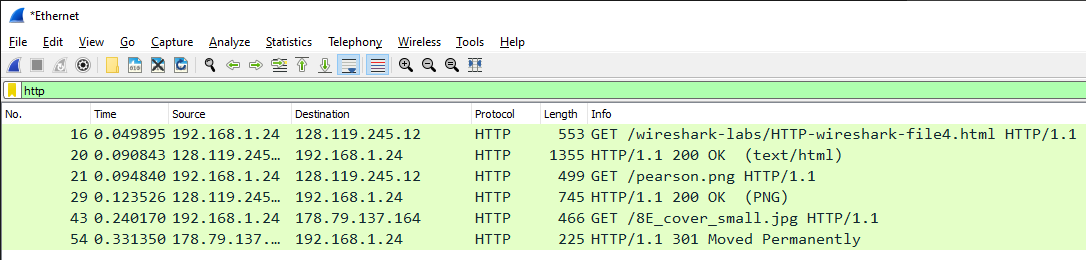
There were three HTTP GET requests. The first one is a GET request from my browser to gaia web server (destination 128.119.245.12). The second is to get “pearson.png” from gaia’s webstie logo publisher, Pearson (destination 128.119.245.12). The third GET request is to get “8E\_cover\_small.jpg” from Kurose.cslash.net server (destination 178.79.137.164).





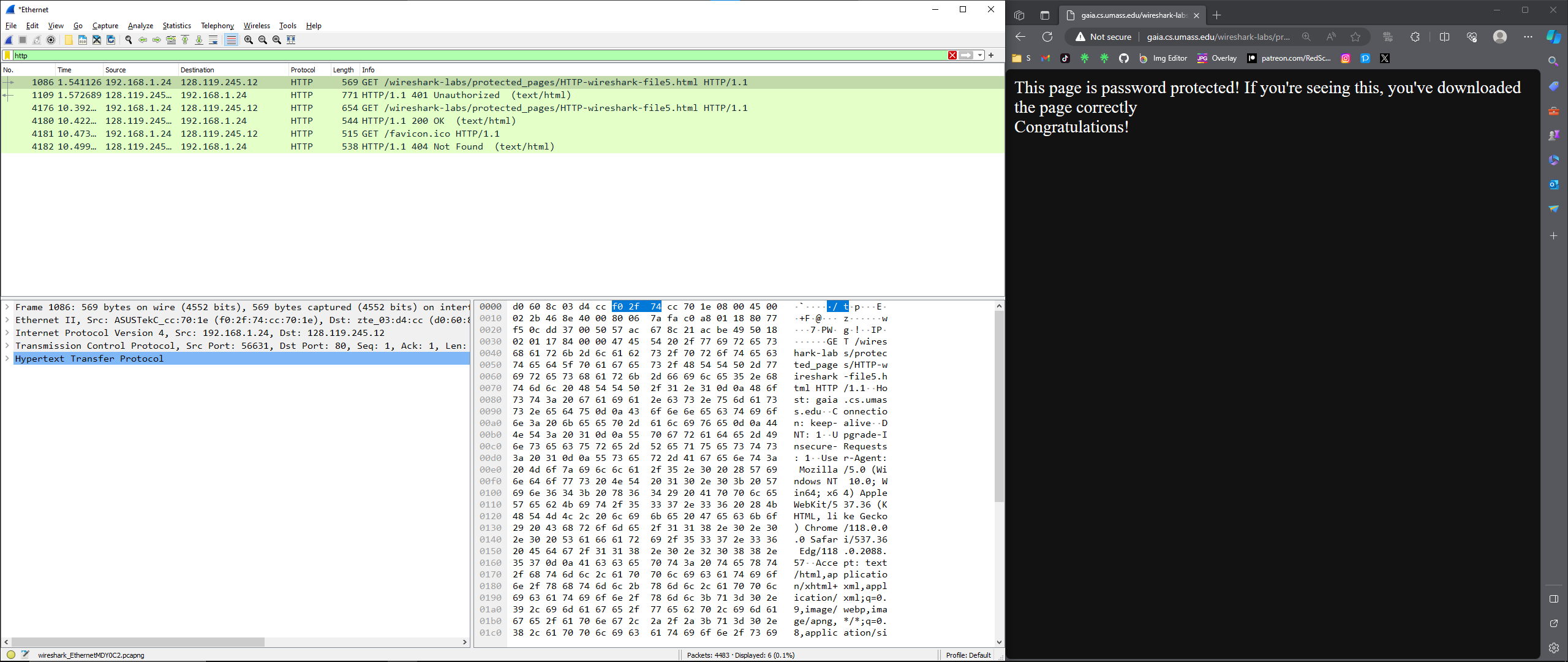
**17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.**

The images were downloaded serially, because the first image finished downloading before the third HTTP GET request. Mainly, observe packet 29 “200 OK (PNG)” comes before packet 43 which is the third HTTP GET request.





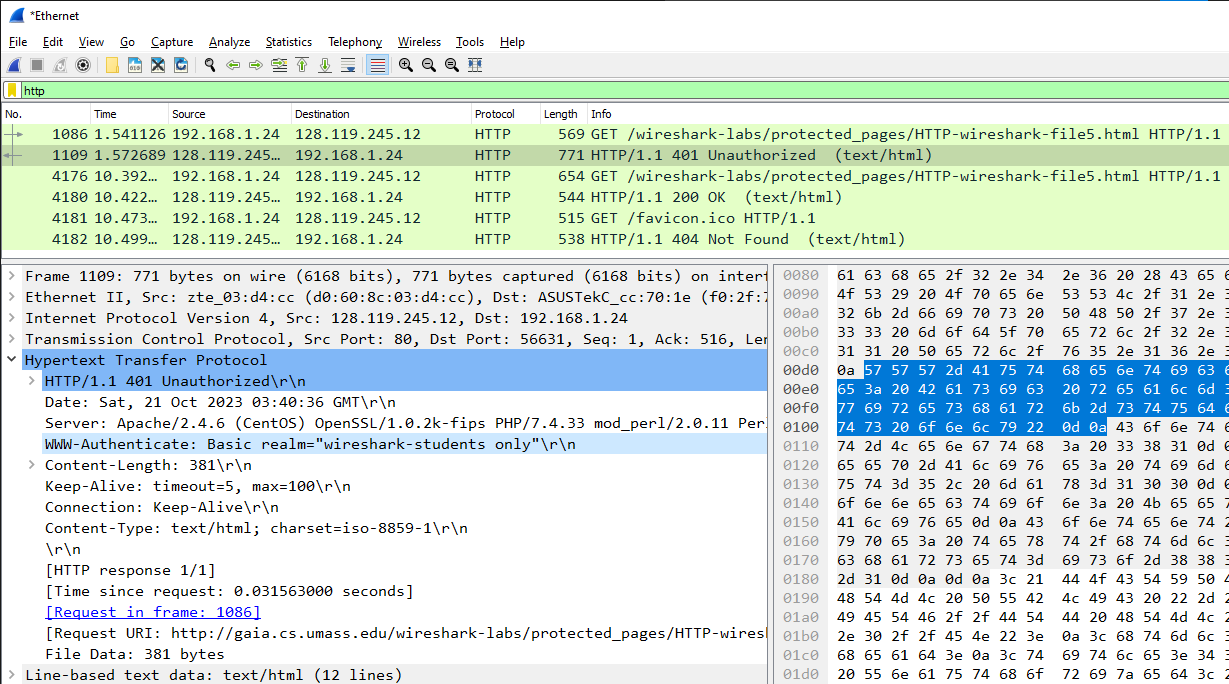
2.5 HTTP Authentication



The above image is the baseline output of opening gaia website with login authentication.

**18. What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?**

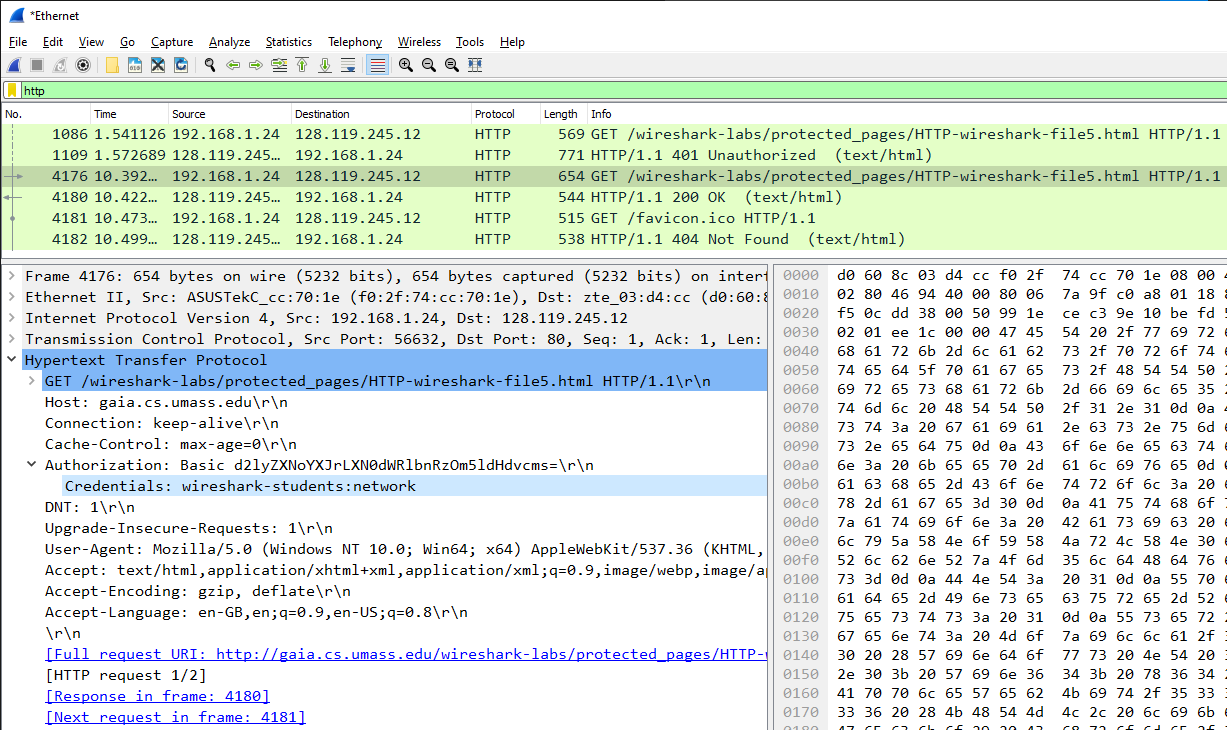
Packet 1109 is the response to the HTTP GET request packet 1086. It says “401 Unauthorized”.





**19. When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?**

When sending the second HTTP GET requests ( packet 4176), “Authorization: Basic” is included with a string of Base64 format that represents the username and password.





1. **Concepts Learned from this Lab:**

Socket programming lab allowed me to understand more about TCP connection and how to create a simple web server. The Wireshark lab provided hands-on experience with HTTP protocol aspects, including GET/response interactions, conditional GET requests, handling long documents, and the use of embedded objects. It also illustrated potential security issues with basic authentication and the need for more secure web access methods.