ACN Programming Assignment-2 : WWW

Group 13 **Extension 3**

Overview of coding parts

Part 1: Simple Web Client

File Name : *web_client.py*

Our simple web client is able to handle http and some https request. Client takes command line input from the user like server address, server port, file path and proxy address & proxy port (if configured) and send the get request to the server or proxy over TCP socket. If server port is 443 or server work over https then out client create SSL socket by wrapping on TCP socket and send the request on SSL socket. On receiving response from the server client parse the data and if there is external reference link inside file than client one by one fetch all the data from links.

For implementing we are using 3 libraries:

ssl	:	For https implementation
socket	:	For TCP socket implementation
re	:	For regex use

Part 2: Simple Web Proxy Server

File Name : *web_proxy.py*

Our simple web proxy server able to handle http request. When client sends a http get request, proxy server parse that request and fetch some useful details from it like server address and all. Then proxy creates new request and send it to server. On receiving response from server, proxy first parse and understand the response code and response message and then just simply forward the response to the client. For the communication all client-proxy and proxy-server connections are handle over new threads. Also it can handle and show the response code like 200, 404, 304, 301.

For implementing we are using 4 libraries:

time	:	For generating response time.
socket	:	For TCP socket implementation
threading	:	For concurrent communication

queue	: For message passing between 2 threads.

Part 3: Simple Web Server

File Name : *web_server.py*

Our simple web server is able to handle http requests. Server listen on port 80 (default http port) and when it get connection request from client it accept connection and start the communication over new thread, means our server is capable of multiple client / concurrency. Server parse the request coming from the client or proxy then it simply check for the file in server's local directory if the file is available then it set file data as body of response otherwise it set small html code for 404 Not Found and send the response to the client.

For implementing we are using 5 libraries:

threading:	For concurrency
socket :	For TCP socket implementation
os :	For file system works
time :	For generating response time
magic ;	For getting MIME extension / content-type of the file.

Part 4: Extension to simple proxy server

File Name : *web_proxy_extend.py*

Our extended web proxy is able to handle http requests. Main role of our extended proxy is to develop the Content Filtering and URL filtering system. Where proxy takes some list of blacklisted websites as part of URL filtering and takes list of blacklisted keywords as part of Content Filtering. It than listen on 8080 port and when client request to proxy, it first parse the request and then check the host if the host is in blacklisted website list, then proxy send response of 403 Forbidden / Access Denied. If website is not in list, then it creates a new thread for proxy-server communication and send the modified request to server. On receiving response from the server proxy parse the data and check if the body contains keyword from blacklisted keyword list or not if present then it shade that word by ***** and generate a modified response and send it to client otherwise send the response simply.

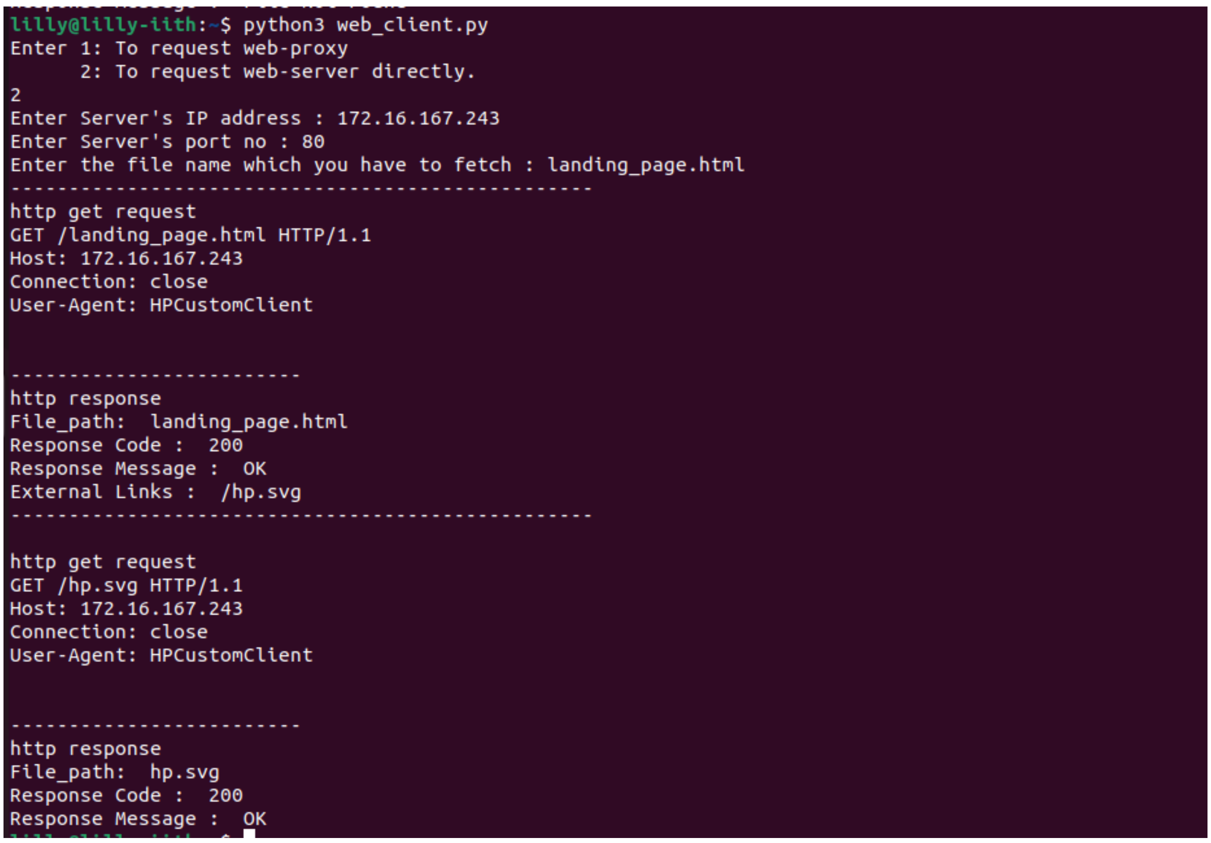
Other role of extended proxy is to generate web statistics in which proxy keeps track of users session and users requested and visited websites. Then at end it plot 2 pie charts one is for the no of times particular website is visited and another one is for the no of overall websites user request through proxy.

For implementing we are using 7 libraries:

threading :	For concurrency
socket :	For TCP socket implementation
os :	For file system works
time :	For generating response time
queue :	For message passing between 2 threads
re :	For regex and string matching
matplotlib:	For plotting piechart as part of web statistics

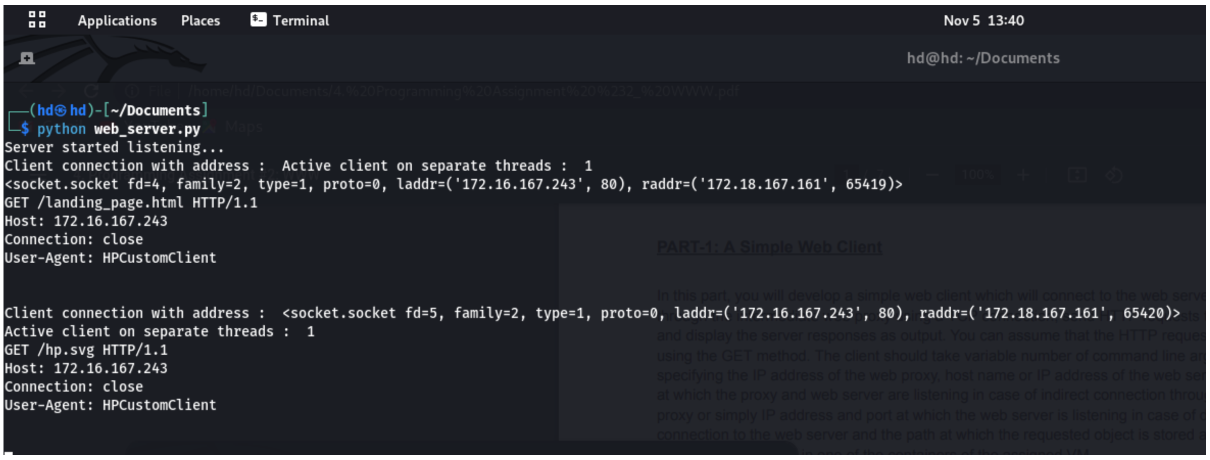
Note: All parsing is done using simple string modification and string methods. No prebuild libraries are used.

<u>**Output**</u>



```
lilly@lilly-iith:~$ python3 web_client.py
Enter 1: To request web-proxy
      2: To request web-server directly.
2
Enter Server's IP address : 172.16.167.243
Enter Server's port no : 80
Enter the file name which you have to fetch : landing_page.html
--------------------------------------------------
http get request
GET /landing_page.html HTTP/1.1
Host: 172.16.167.243
Connection: close
User-Agent: HPCustomClient


-------------------------
http response
File_path:  landing_page.html
Response Code :  200
Response Message :  OK
External Links :  /hp.svg
--------------------------------------------------

http get request
GET /hp.svg HTTP/1.1
Host: 172.16.167.243
Connection: close
User-Agent: HPCustomClient


-------------------------
http response
File_path:  hp.svg
Response Code :  200
Response Message :  OK
```

Image 1.1 : Simple Client Server Communication (Client Side)



```
┌──(hd㉿hd)-[~/Documents]
└─$ python web_server.py
Server started listening...
Client connection with address :  Active client on separate threads :  1
<socket.socket fd=4, family=2, type=1, proto=0, laddr=('172.16.167.243', 80), raddr=('172.18.167.161', 65419)>
GET /landing_page.html HTTP/1.1
Host: 172.16.167.243
Connection: close
User-Agent: HPCustomClient


Client connection with address :  <socket.socket fd=5, family=2, type=1, proto=0, laddr=('172.16.167.243', 80), raddr=('172.18.167.161', 65420)>
Active client on separate threads :  1
GET /hp.svg HTTP/1.1
Host: 172.16.167.243
Connection: close
User-Agent: HPCustomClient
```

Image 1.2 : Simple Client Server Communication (Server Side)

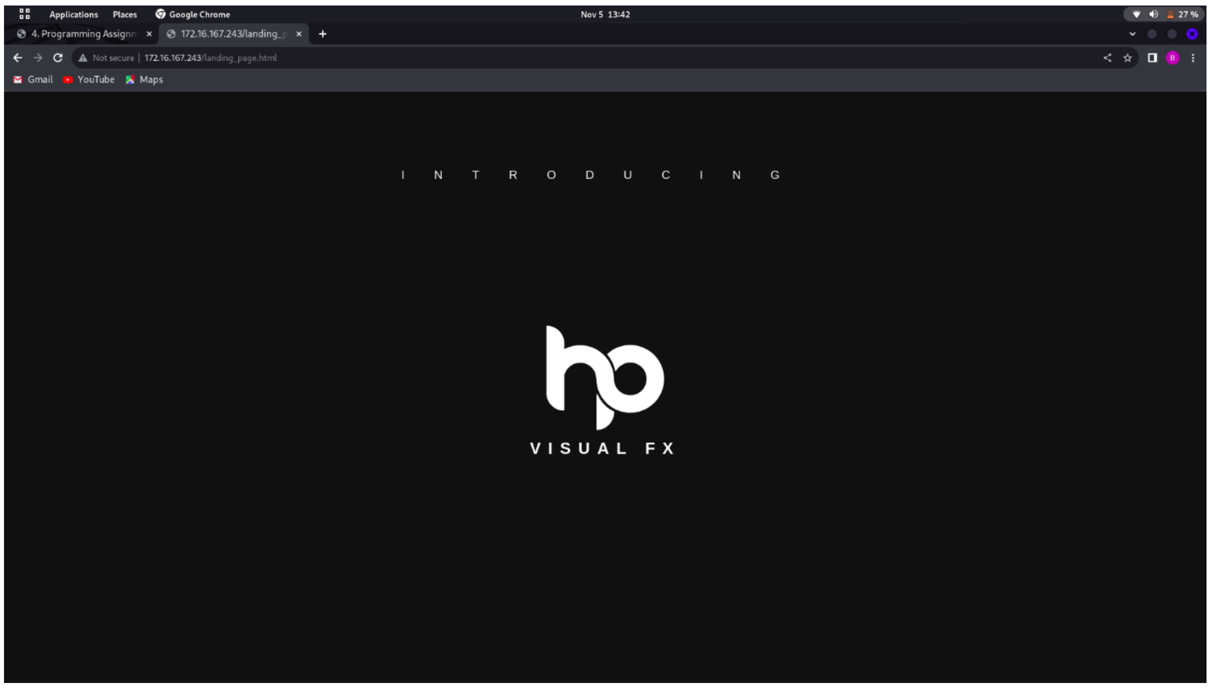
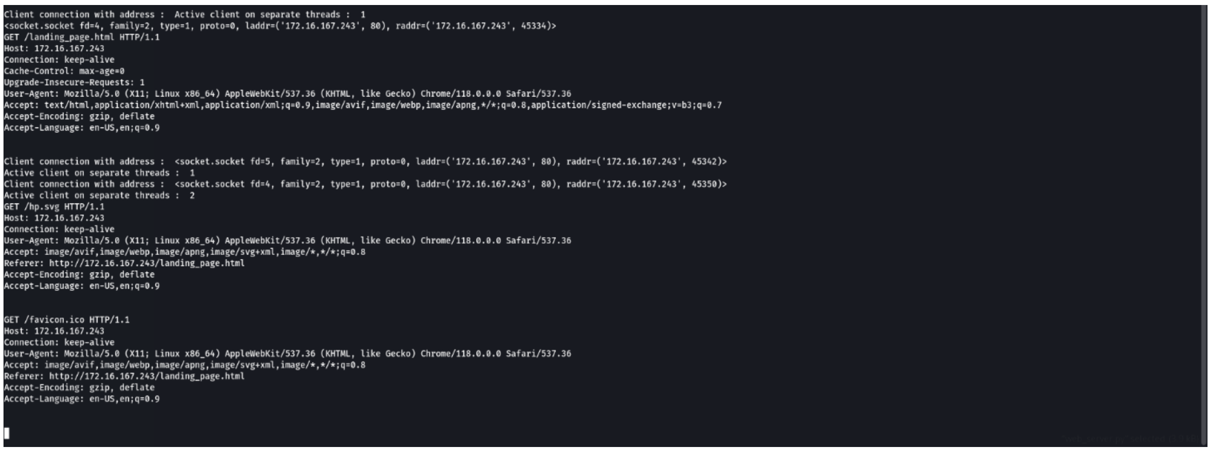
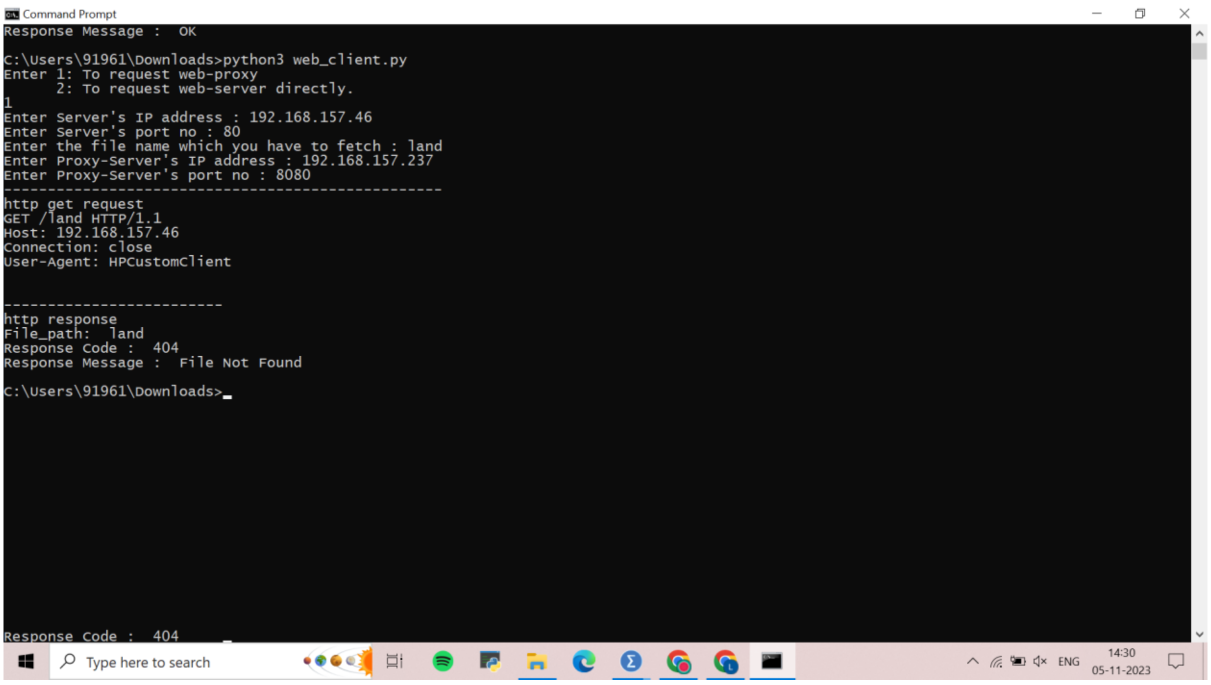


Image 1.3 : Simple Client Server Communication test (Client as Chrome Browser)



```
Client connection with address :  Active client on separate threads :  1
<socket.socket fd=4, family=2, type=1, proto=0, laddr=('172.16.167.243', 80), raddr=('172.16.167.243', 45334)>
GET /landing_page.html HTTP/1.1
Host: 172.16.167.243
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


Client connection with address :  <socket.socket fd=5, family=2, type=1, proto=0, laddr=('172.16.167.243', 80), raddr=('172.16.167.243', 45342)>
Active client on separate threads :  1
Client connection with address :  <socket.socket fd=4, family=2, type=1, proto=0, laddr=('172.16.167.243', 80), raddr=('172.16.167.243', 45350)>
Active client on separate threads :  2
GET /hp.svg HTTP/1.1
Host: 172.16.167.243
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://172.16.167.243/landing_page.html
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


GET /favicon.ico HTTP/1.1
Host: 172.16.167.243
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://172.16.167.243/landing_page.html
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

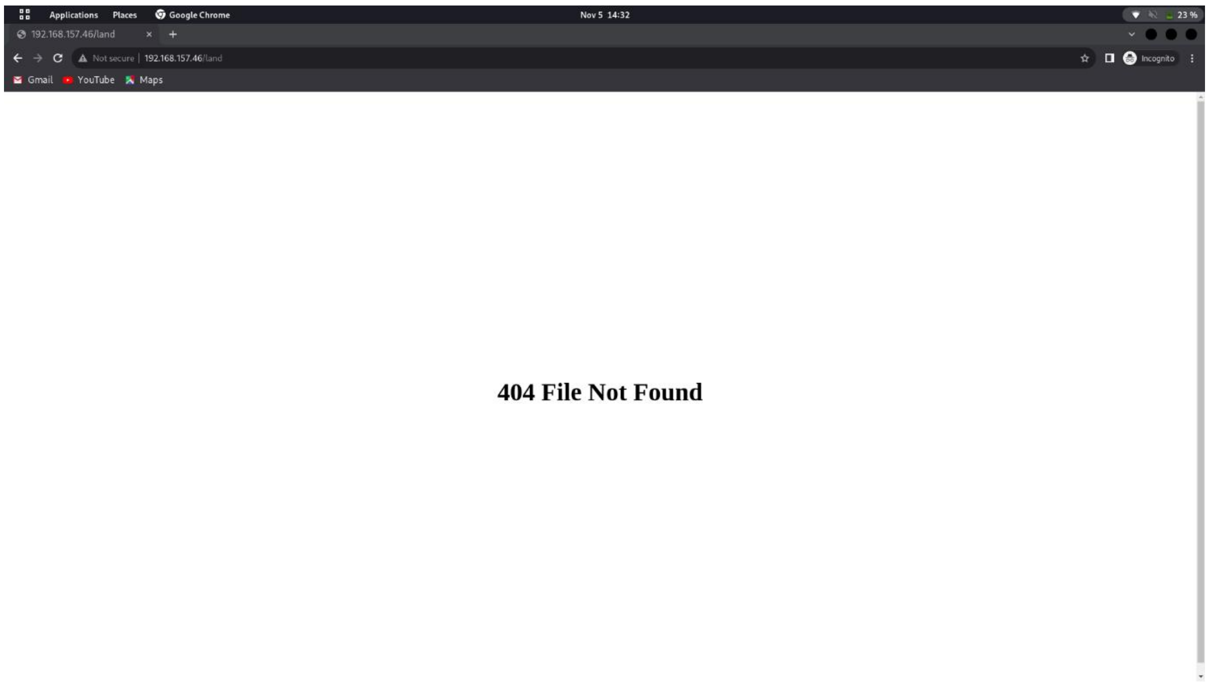
Image 1.4 : Simple Client Server Communication test (Server Side)



```
Response Message :  OK

C:\Users\91961\Downloads>python3 web_client.py
Enter 1: To request web-proxy
      2: To request web-server directly.
1
Enter Server's IP address : 192.168.157.46
Enter Server's port no : 80
Enter the file name which you have to fetch : land
Enter Proxy-Server's IP address : 192.168.157.237
Enter Proxy-Server's port no : 8080
--------------------------------------------------
http get request
GET /land HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient


-------------------------
http response
File_path:  land
Response Code :  404
Response Message :  File Not Found

C:\Users\91961\Downloads>_
```

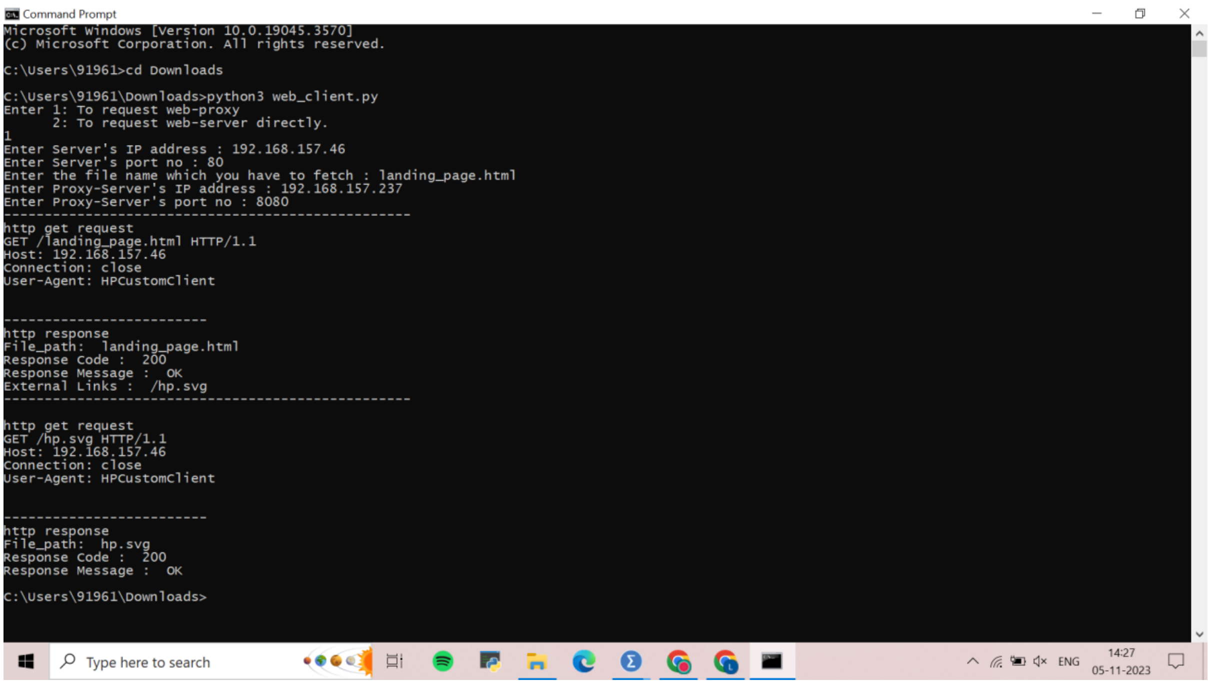
```
Response Code :  404
```

Image 1.5 : Simple Client Server Communication File Not Found (Client Side)



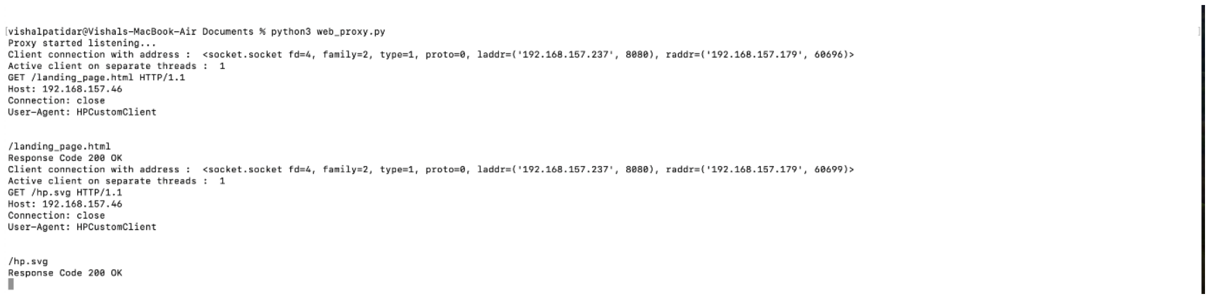
404 File Not Found

Image 1.6 : Simple Client Server Communication File Not Found (Client as Chrome Browser)



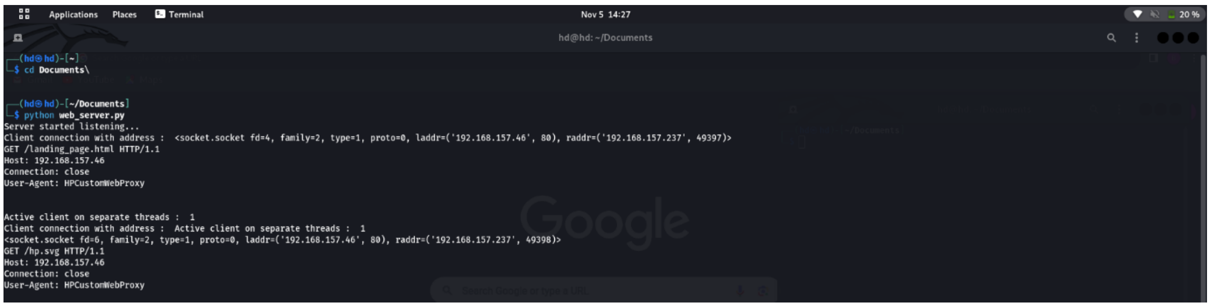
```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91961>cd Downloads

C:\Users\91961\Downloads>python3 web_client.py
Enter 1: To request web-proxy
      2: To request web-server directly.
1
Enter Server's IP address : 192.168.157.46
Enter Server's port no : 80
Enter the file name which you have to fetch : landing_page.html
Enter Proxy-Server's IP address : 192.168.157.237
Enter Proxy-Server's port no : 8080
--------------------------------------------------
http get request
GET /landing_page.html HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient

-------------------------
http response
File_path:  landing_page.html
Response Code :  200
Response Message :  OK
External Links :  /hp.svg
--------------------------------------------------

http get request
GET /hp.svg HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient

-------------------------
http response
File_path:  hp.svg
Response Code :  200
Response Message :  OK

C:\Users\91961\Downloads>
```

Image 1.7 : Simple Client – Proxy - Server Communication (Client side)



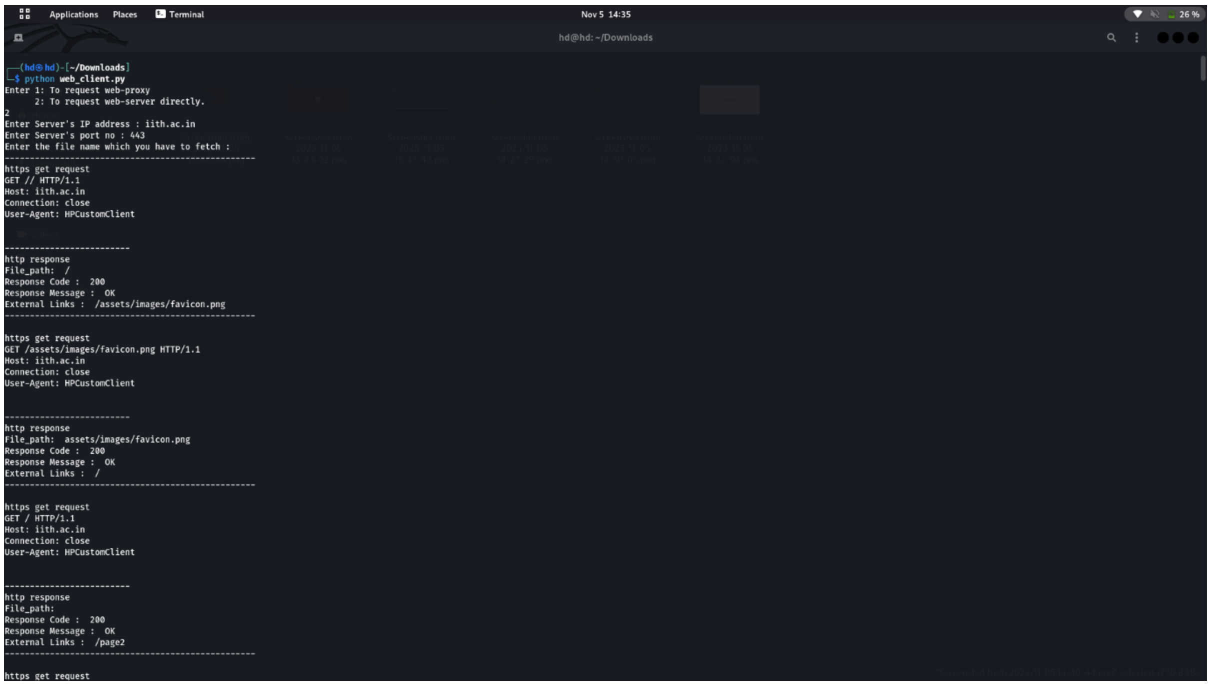
```
vishalpatidar@Vishals-MacBook-Air Documents % python3 web_proxy.py
Proxy started listening...
Client connection with address :  <socket.socket fd=4, family=2, type=1, proto=0, laddr=('192.168.157.237', 8080), raddr=('192.168.157.179', 60696)>
Active client on separate threads :  1
GET /landing_page.html HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient


/landing_page.html
Response Code 200 OK
Client connection with address :  <socket.socket fd=4, family=2, type=1, proto=0, laddr=('192.168.157.237', 8080), raddr=('192.168.157.179', 60699)>
Active client on separate threads :  1
GET /hp.svg HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient


/hp.svg
Response Code 200 OK
```

Image 1.8 : Simple Client – Proxy - Server Communication (Proxy side)



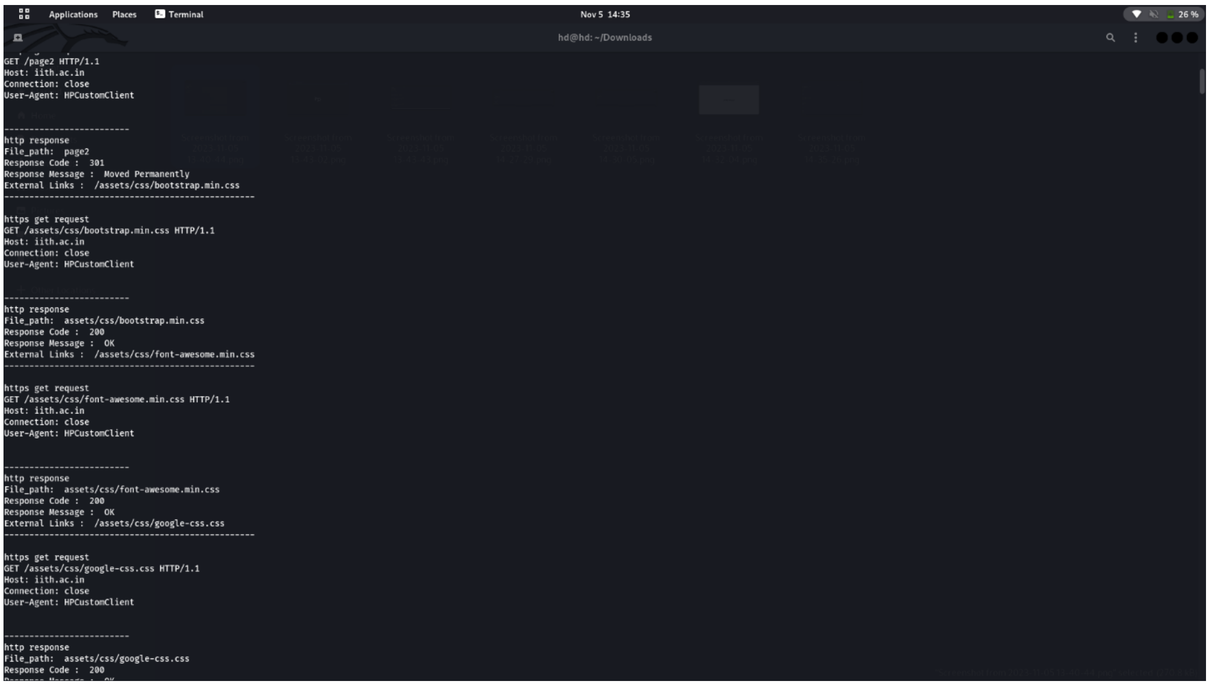
```
┌──(hd㉿hd)-[~]
└─$ cd Documents\

┌──(hd㉿hd)-[~/Documents]
└─$ python web_server.py
Server started listening...
Client connection with address :  <socket.socket fd=4, family=2, type=1, proto=0, laddr=('192.168.157.46', 80), raddr=('192.168.157.237', 49397)>
GET /landing_page.html HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomWebProxy


Active client on separate threads :  1
Client connection with address :  Active client on separate threads :  1
<socket.socket fd=6, family=2, type=1, proto=0, laddr=('192.168.157.46', 80), raddr=('192.168.157.237', 49398)>
GET /hp.svg HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomWebProxy
```

Image 1.9 : Simple Client – Proxy - Server Communication (Server side)



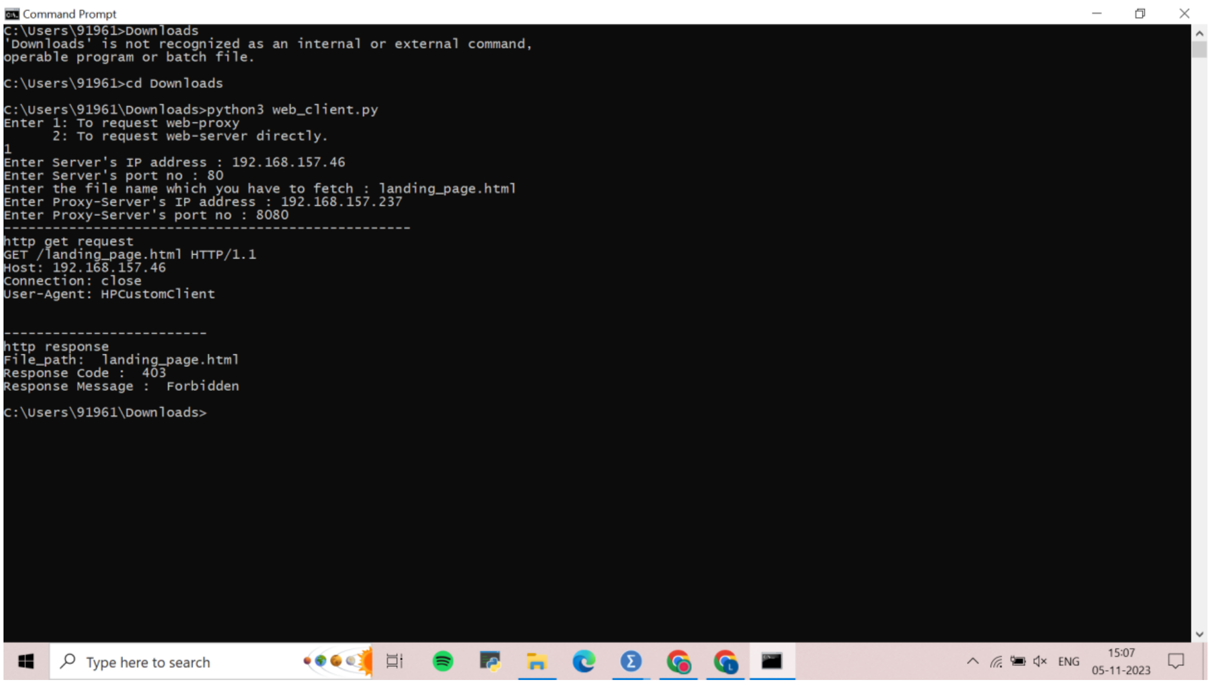
```
┌──(hd㉿hd)-[~/Downloads]
└─$ python web_client.py
Enter 1: To request web-proxy
      2: To request web-server directly.
2
Enter Server's IP address : iith.ac.in
Enter Server's port no : 443
Enter the file name which you have to fetch :
------------------------------------------------
https get request
GET // HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  /
Response Code :  200
Response Message :  OK
External Links :  /assets/images/favicon.png
------------------------------------------------

https get request
GET /assets/images/favicon.png HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  assets/images/favicon.png
Response Code :  200
Response Message :  OK
External Links :  /
------------------------------------------------

https get request
GET / HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:
Response Code :  200
Response Message :  OK
External Links :  /page2
------------------------------------------------

https get request
```

Image 1.10 (a) : Simple Client Working on https sites like iith.ac.in



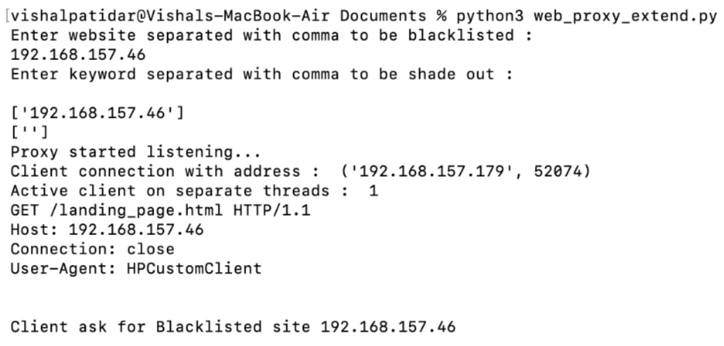
```
GET /page2 HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  page2
Response Code :  301
Response Message :  Moved Permanently
External Links :  /assets/css/bootstrap.min.css
------------------------------------------------

https get request
GET /assets/css/bootstrap.min.css HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  assets/css/bootstrap.min.css
Response Code :  200
Response Message :  OK
External Links :  /assets/css/font-awesome.min.css
------------------------------------------------

https get request
GET /assets/css/font-awesome.min.css HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  assets/css/font-awesome.min.css
Response Code :  200
Response Message :  OK
External Links :  /assets/css/google-css.css
------------------------------------------------

https get request
GET /assets/css/google-css.css HTTP/1.1
Host: iith.ac.in
Connection: close
User-Agent: HPCustomClient


------------------------
http response
File_path:  assets/css/google-css.css
Response Code :  200
Response Message :  OK
```

Image 1.10 (b) : Simple Client Working on https sites like iith.ac.in



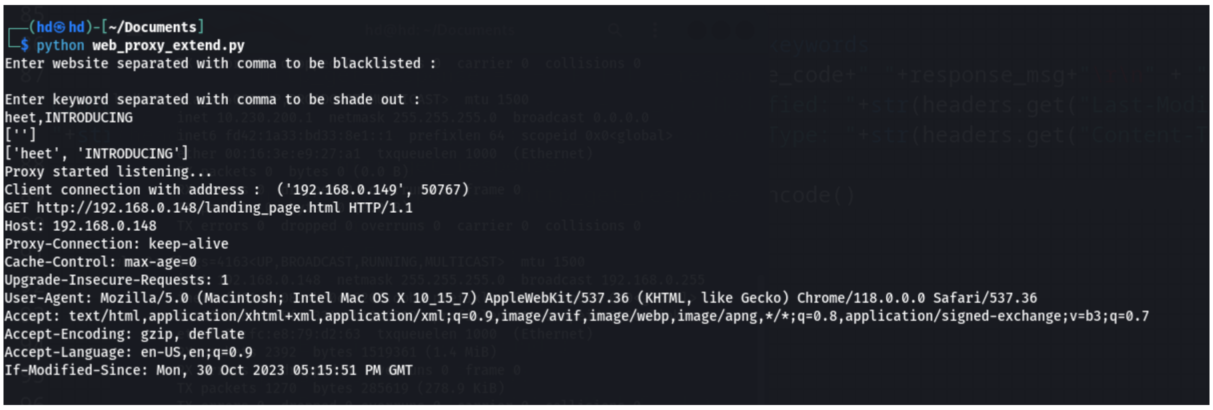
```
C:\Users\91961>Downloads
'Downloads' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\91961>cd Downloads

C:\Users\91961\Downloads>python3 web_client.py
Enter 1: To request web-proxy
      2: To request web-server directly.
1
Enter Server's IP address : 192.168.157.46
Enter Server's port no : 80
Enter the file name which you have to fetch : landing_page.html
Enter Proxy-Server's IP address : 192.168.157.237
Enter Proxy-Server's port no : 8080
------------------------------------------------------
http get request
GET /landing_page.html HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient


-------------------------
http response
File_path:  landing_page.html
Response Code :  403
Response Message :  Forbidden

C:\Users\91961\Downloads>
```

Image 1.11 : URL Filtering as part of Extended Proxy server (Client Side)



```
vishalpatidar@Vishals-MacBook-Air Documents % python3 web_proxy_extend.py
Enter website separated with comma to be blacklisted :
192.168.157.46
Enter keyword separated with comma to be shade out :

['192.168.157.46']
['']
Proxy started listening...
Client connection with address :  ('192.168.157.179', 52074)
Active client on separate threads :  1
GET /landing_page.html HTTP/1.1
Host: 192.168.157.46
Connection: close
User-Agent: HPCustomClient


Client ask for Blacklisted site 192.168.157.46
```

Image 1.12 : URL Filtering as part of Extended Proxy server (Proxy Side)



```
┌──(hd㉿hd)-[~/Documents]
└─$ python web_proxy_extend.py
Enter website separated with comma to be blacklisted :

Enter keyword separated with comma to be shade out :
heet,INTRODUCING
['']
['heet', 'INTRODUCING']
Proxy started listening...
Client connection with address :  ('192.168.0.149', 50767)
GET http://192.168.0.148/landing_page.html HTTP/1.1
Host: 192.168.0.148
Proxy-Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
If-Modified-Since: Mon, 30 Oct 2023 05:15:51 PM GMT
```

Image 1.13 : Content Filtering as part of Extended Proxy server (Proxy Side)

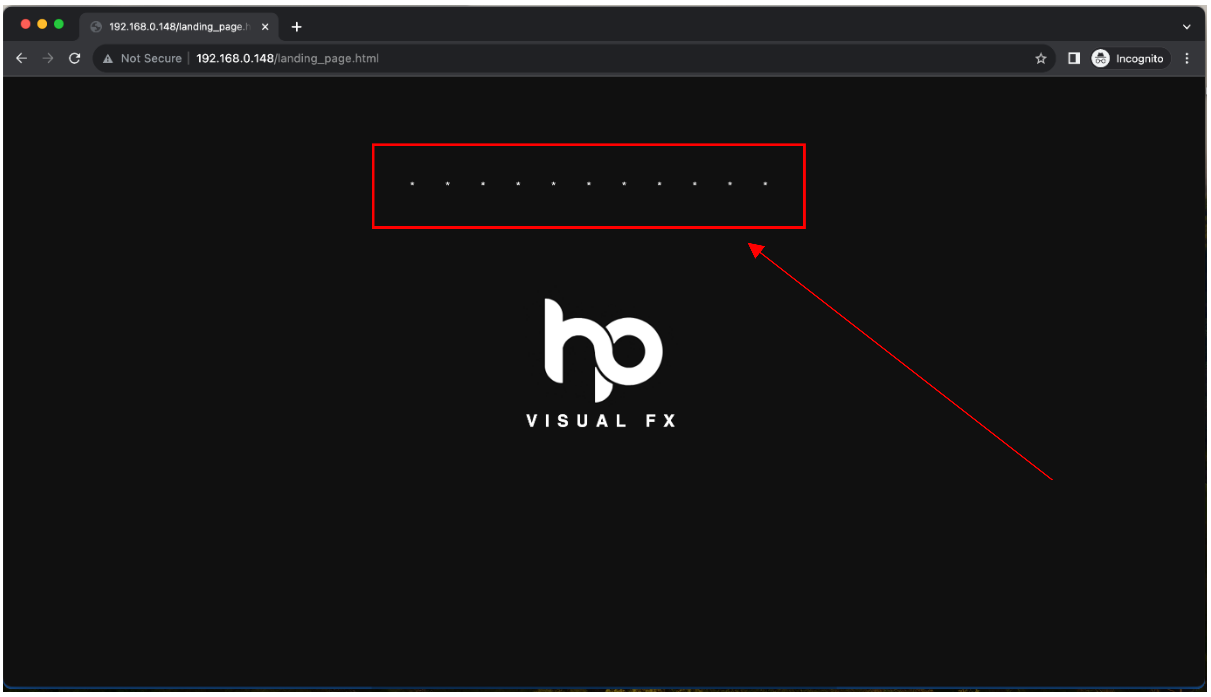

192.168.0.148/landing_page.h
Not Secure | 192.168.0.148/landing_page.html
Incognito
. . . .
VISUAL FX

Image 1.14 : Content Filtering as part of Extended Proxy server (Client Side as Browser)

You can see in above Image 1.14 content filtering is done and the keyword INTRODUCING is shaded by ********** compare it with Image

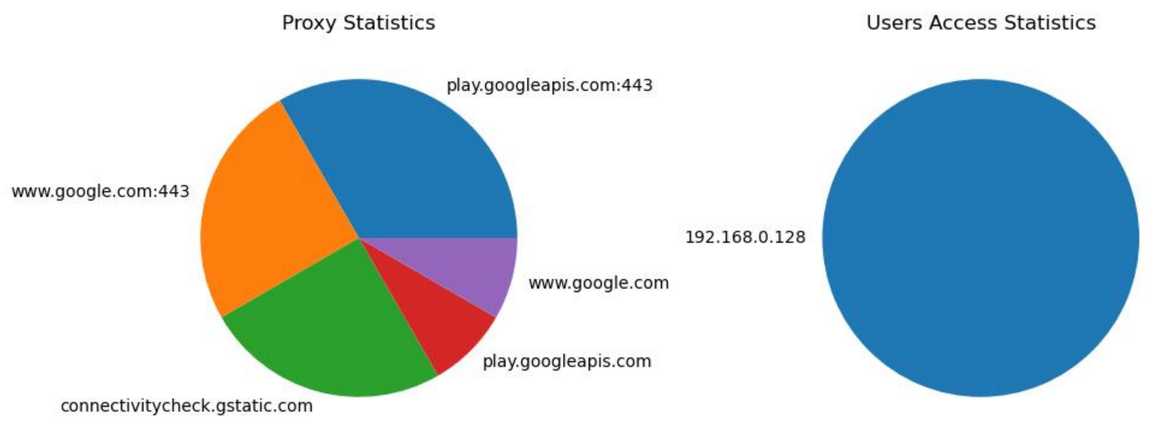


Image 1.15 : Web statistics plotting as part of Extended Proxy server

1. Piechart 1 shows website access proportion
2. Piechart 2 shows User's proxy usage proportion

Anti-Plagiarism Statement

We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. Additionally, we acknowledge that we may have used AI tools, such as language models (e.g., ChatGPT, Bard), for assistance in generating and refining my assignment, and we have made all reasonable efforts to ensure that such usage complies with the academic integrity policies set for the course. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, we understand our responsibility to report honour violations by other students if we become aware of it.

CS23MTECH11029
Date : 5/11/2023
PATEL HEETKUMAR D.

CS23MTECH11027
Date : 5/11/2023
BISWAS LILLY KUMARI

CS23MTECH14017
Date : 5/11/2023
VISHAL PATIDAR