CSE 351: COMPUTER NETWORKS

Team

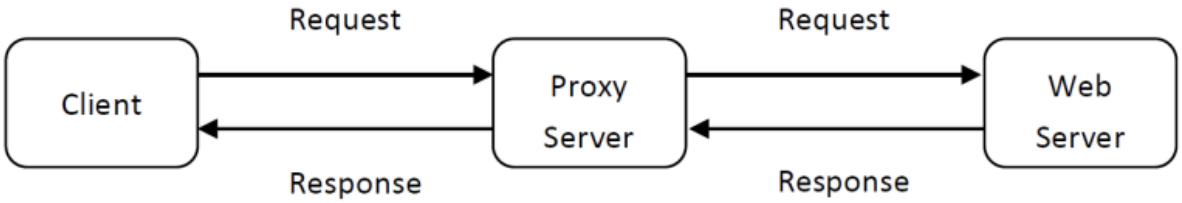
Alia Tamer Fathi - 19P3093

Omar Mohamed Youssef - 19P4953

Kareem Ahmed Azab - 19P5097

Contents

1.0 System Design



The client requests the objects via the proxy server. The proxy server will forward the client's request to the web server. The web server will then generate a response message and deliver it to the proxy server, which in turn sends it to the client.

Moreover, there is Error Handling for when the web page requested is not found.

The Proxy Server is also a Caching Server, so when the proxy gets a request, it checks if the requested object is cached, and if yes, it returns the object from the cache, without contacting the server. If the object is not cached, the proxy retrieves the object from the server, returns it to the client and caches a copy for future requests.

Finally, the server has URL Filtering so when the Proxy receives a request for a specific URL, it looks up the local data base and if the URL is listed there, then it returns an error page with a message that says "This URL is blocked". This local database is a simple text file where each URL is written in a separate line.

2.0 Proxy Server Code

2.1 Server Initialization

```python
from socket import *
import sys

if len(sys.argv) <= 1:
    print(
        'Usage : "python ProxyServer.py server_ip"\n[server_ip : It is the IP Address Of Proxy Server'
    )
    sys.exit(2)
# Create a server socket, bind it to a port and start listening
tcpSerSock = socket(AF_INET, SOCK_STREAM)
# Fill in start.
tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind((sys.argv[1], 5050))
tcpSerSock.listen(10)
# Fill in end.

```

2.2 Server waiting for message and reading blocked sites from file

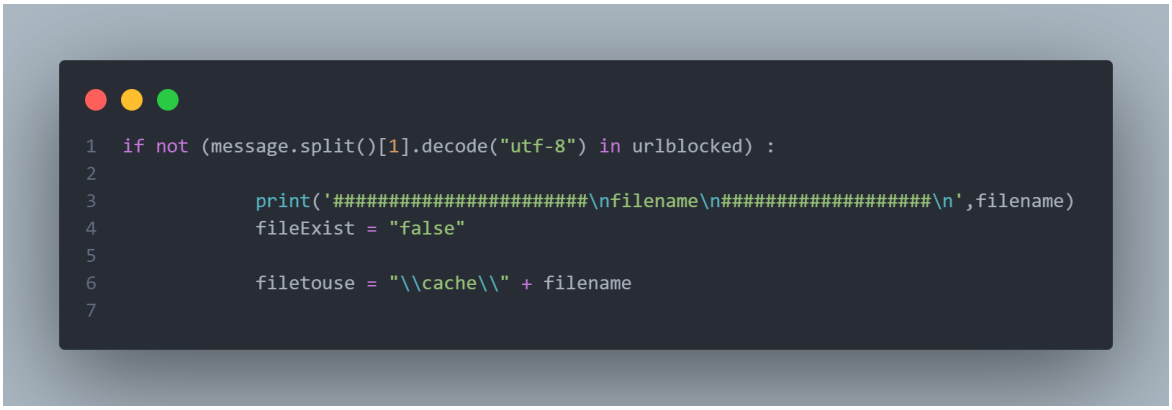
```python
while 1:
    # Start receiving data from the client
    print("\n\nReady to serve...")
    tcpCliSock, addr = tcpSerSock.accept()
    print("Received a connection from:", addr)
    message = tcpCliSock.recv(1024)  # Fill in start. # Fill in end.
    ##############################
    urlfile = open("urlfilter.txt", "r")
    urlblocked = urlfile.readlines()
    urlfile.close()
    ##############################
```

2.3 Check If message is empty or not

```
if message:
        print('######################\nmessage\n###################\n',message)
        filename = message.split()[1].decode("utf-8").rpartition("/")[2]

```

2.4 Extract the filename from message and the cache location



```
if not (message.split()[1].decode("utf-8") in urlblocked) :

            print('######################\nfilename\n###################\n',filename)
            fileExist = "false"

            filetouse = "\\cache\\" + filename

```

2.5 in case file is found in the cache return response from the cache

```
try:
            # Check wether the file exist in the cache
            f = open(filetouse[1:], "rb")
            outputdata = f.readlines()
            print(f'######################\nfile exist at {filetouse}\n###################')
            print("Read from cache")
            fileExist = "true"
            # ProxyServer finds a cache hit and generates a response message
            tcpCliSock.send(b"HTTP/1.0 200 OK\r\n")
            tcpCliSock.send(b"Content-Type:text/html\r\n")
            # Fill in start.

            for line in outputdata:
                tcpCliSock.send(line)
            f.close()
```

2.6 Message Processing

2.6.0 In case file not found create connection

```
except IOError:
                try:
                    if fileExist == "false":
                        # Create a socket on the proxyserver
                        print('######################\nfile not found\n###################')

                        c = socket(AF_INET, SOCK_STREAM)  # Fill in start. # Fill in end.
                        hostn = message.split()[4].decode("utf-8")

                        # Connect to the socket to port 80
                        c.connect((hostn, 80))

```

2.6.1 Request the missing file from the link and download it

```
# Create a temporary file on this socket and ask port 80 for the file requested by the client
                        fileobjwrite = c.makefile("w", None)
                        # request
                        print('######################\ncaching....\n###################')
                        fileobjwrite.write(
                            "GET " + message.split()[1].decode("utf-8") + " HTTP/1.0\n\n"
                        )
                        fileobjwrite.close()
                        print('######################\ncaching complete\n###################')

```

2.6.2 Read from the new cached file and send a response to the client

```
                        fileobj = c.makefile("rb", None)
                        buff = fileobj.readlines()
                        # Create a new file in the cache for the requested file.
                        # Also send the response in the buffer to client socket and the corresponding file in the cache
                        print('######################\nsending response from cache\n###################')
                        File = open("./cache/" + filename, "wb+")
                        for line in buff:
                            File.write(line)
                            tcpCliSock.send(line)
                        File.close()
                        c.close()

```

2.7 In case something goes wrong

```
except:
    print("Illegal")
```

2.8 Closing connections or if the connection in the filter

```
        else:
            print("\n\n\n\nconnection blocked\n\n\n\n\n")
            tcpCliSock.close()

else:
    ...
    # HTTP response message for file not found
    tcpCliSock.send("HTTP/1.0 404 sendError\r\n")
    tcpCliSock.send("Content-Type:text/html\r\n")
    # Close the client and the server sockets
    tcpCliSock.close()
    print("socket closed")
```

3.0 Proof Screenshots

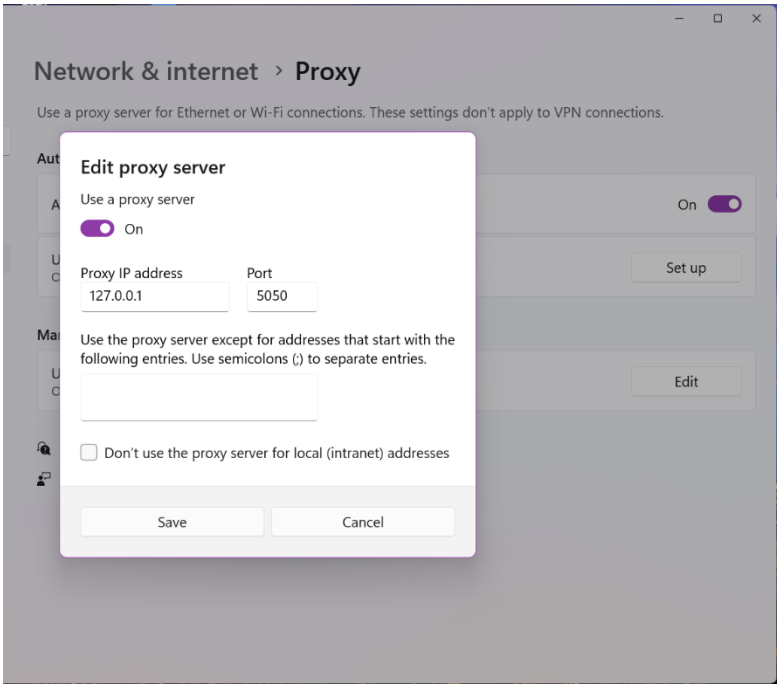


FIGURE 1: CONFIGURING THE PROXY CONNECTION ON CLIENT SIDE

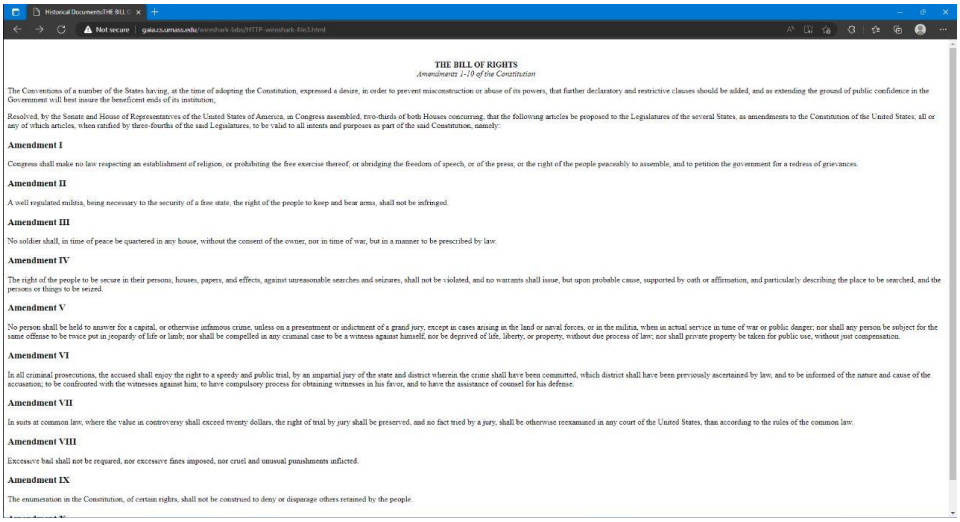


FIGURE 2: REQUESTING AN HTTP WEB PAGE IN MICROSOFT EDGE

FIGURE 3: GET MESSAGE IN TERMINAL AFTER REQUESTING THE HTTP WEB PAGE FOR THE FIRST TIME FROM THE REMOTE SERVER AND CACHING THIS PAGE

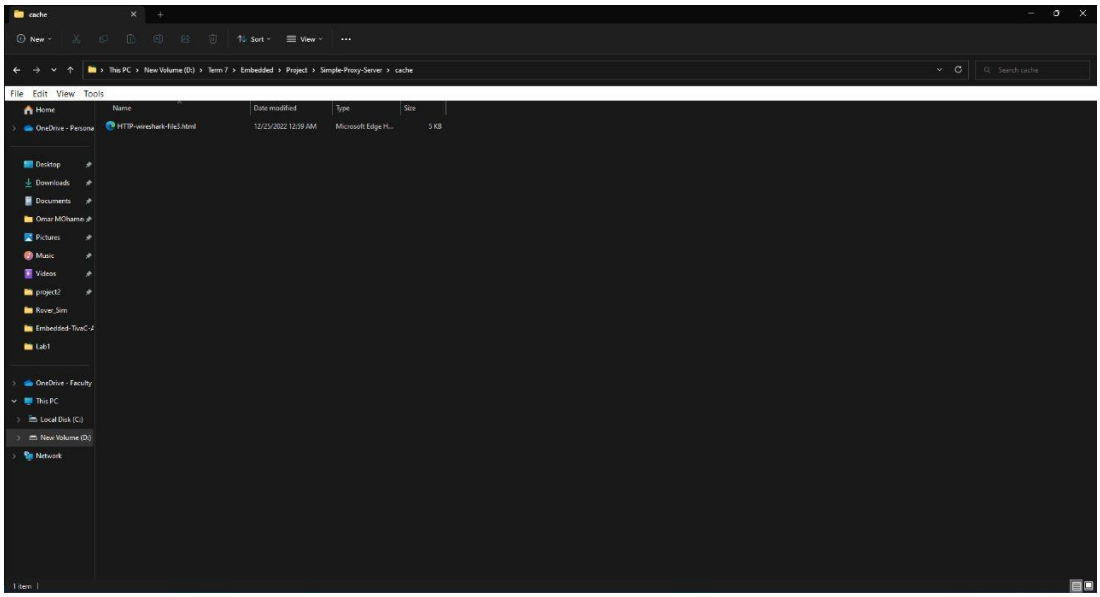
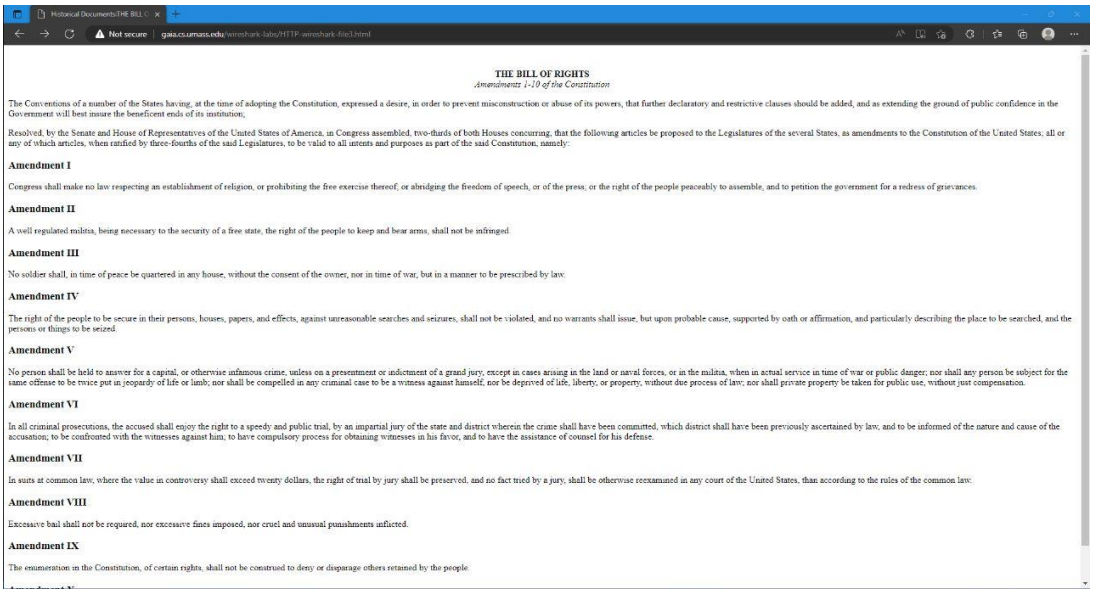


FIGURE 4: THE CACHE AFTER REQUESTING THE WEB PAGE AND CACHING IT



THE BILL OF RIGHTS

Amendments 1-10 of the Constitution

The Conventions of a number of the States having, at the time of adopting the Constitution, expressed a desire, in order to prevent misconstruction or abuse of its powers, that further declaratory and restrictive clauses should be added, and as extending the ground of public confidence in the Government will best insure the beneficent ends of its institution;

Resolved, by the Senate and House of Representatives of the United States of America, in Congress assembled, two-thirds of both Houses concurring, that the following articles be proposed to the Legislatures of the several States, as amendments to the Constitution of the United States; all or any of which articles, when ratified by three-fourths of the said Legislatures, to be valid to all intents and purposes as part of the said Constitution, namely:

Amendment I

Congress shall make no law respecting an establishment of religion, or prohibiting the free exercise thereof; or abridging the freedom of speech, or of the press; or the right of the people peaceably to assemble, and to petition the government for a redress of grievances.

Amendment II

A well regulated militia, being necessary to the security of a free state, the right of the people to keep and bear arms, shall not be infringed.

Amendment III

No soldier shall, in time of peace be quartered in any house, without the consent of the owner, nor in time of war, but in a manner to be prescribed by law.

Amendment IV

The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated, and no warrants shall issue, but upon probable cause, supported by oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.

Amendment V

No person shall be held to answer for a capital, or otherwise infamous crime, unless on a presentment or indictment of a grand jury, except in cases arising in the land or naval forces, or in the militia, when in actual service in time of war or public danger; nor shall any person be subject for the same offense to be twice put in jeopardy of life or limb; nor shall be compelled in any criminal case to be a witness against himself, nor be deprived of life, liberty, or property, without due process of law; nor shall private property be taken for public use, without just compensation.

Amendment VI

In all criminal prosecutions, the accused shall enjoy the right to a speedy and public trial, by an impartial jury of the state and district wherein the crime shall have been committed, which district shall have been previously ascertained by law, and to be informed of the nature and cause of the accusation; to be confronted with the witnesses against him; to have compulsory process for obtaining witnesses in his favor, and to have the assistance of counsel for his defense.

Amendment VII

In suits at common law, where the value in controversy shall exceed twenty dollars, the right of trial by jury shall be preserved, and no fact tried by a jury, shall be otherwise reexamined in any court of the United States, than according to the rules of the common law.

Amendment VIII

Excessive bail shall not be required, nor excessive fines imposed, nor cruel and unusual punishments inflicted.

Amendment IX

The enumeration in the Constitution, of certain rights, shall not be construed to deny or disparage others retained by the people.

FIGURE 5: REQUESTING THE SAME WEB PAGE FOR THE SECOND TIME

```
Windows PowerShell
Illegal


Ready to serve...
Received a connection from: ('127.0.0.1', 52471)
##########################
message
####################
 b'GET http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1\r\nHost: gaia.cs.umass.edu\r\nProxy-Connection: keep-alive\r\nCache-Contro
l: max-age=0\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.
0 Safari/537.36 Edg/108.0.1462.54\r\nAccept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.9\r\nAccept-Encoding: gzip, deflate\r\nAccept-Language: en-US,en;q=0.9\r\nIf-None-Match: "1194-5f08d6feac528"\r\nIf-Modified-Since: Sat, 24 Dec
2022 06:59:02 GMT\r\n\r\n'
##########################
filename
####################
 HTTP-wireshark-file3.html
##########################
file exist at \cache\HTTP-wireshark-file3.html
####################
Read from cache


Ready to serve...
Received a connection from: ('127.0.0.1', 52472)
##########################
message
####################
 b'CONNECT ytub.site:443 HTTP/1.1\r\nHost: ytub.site:443\r\nProxy-Connection: keep-alive\r\nUser-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.54\r\n\r\n'
##########################
filename
####################
 ytub.site:443
##########################
file not found
####################
Illegal
```

FIGURE 6: MESSAGE IN TERMINAL SHOWING THE SECOND REQUEST TO THE WEB PAGE AND BRINGING IT FROM THE CACHE (NOT THE REMOTE SERVER)

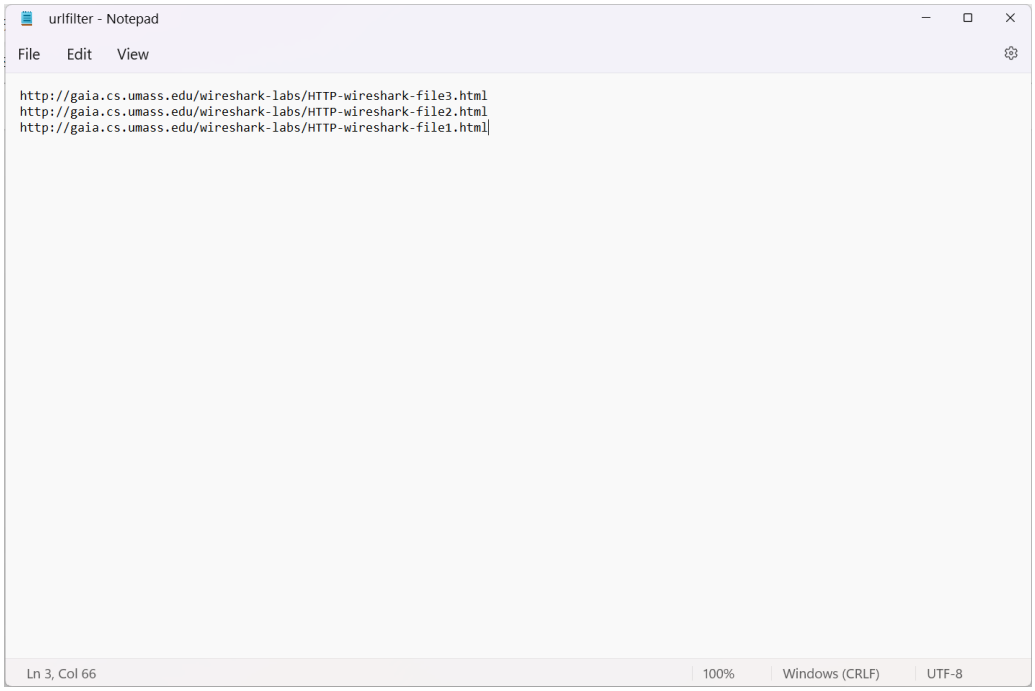


FIGURE 7: URL FILTER FILE CONTAINING THE LINKS TO BE BLOCKED

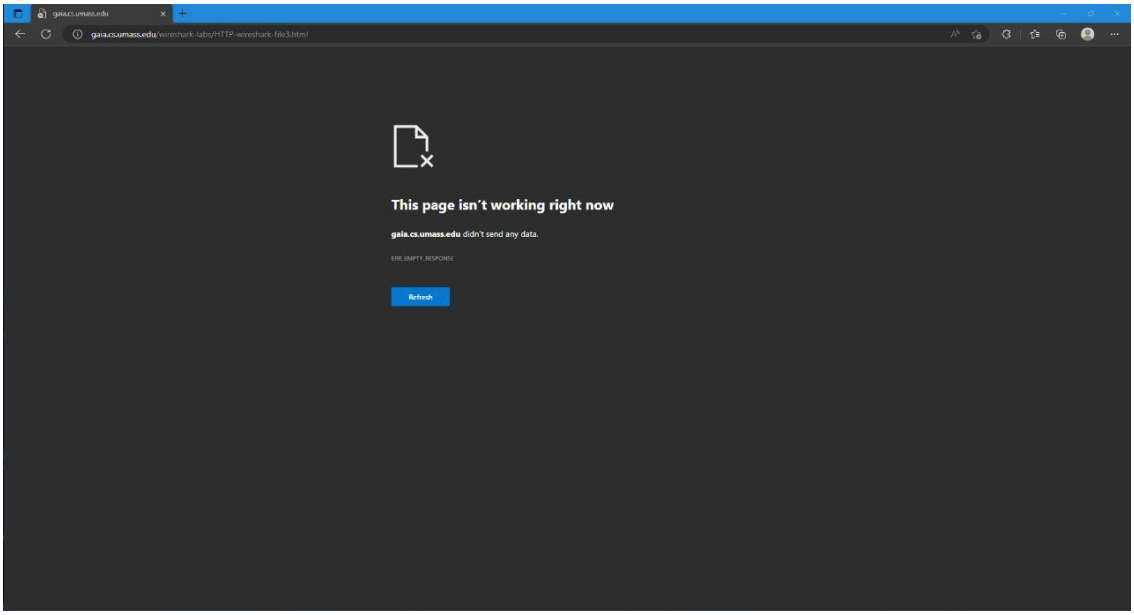


FIGURE 8: REQUESTING ONE OF THE BLOCKED WEB PAGES IN THE URL FILTER FILE

FIGURE 9: MESSAGE IN TERMINAL AFTER REQUESTING A BLOCKED WEB PAGE

4.0 Project GitHub Link

https://github.com/mourra950/Simple-Proxy-Server