CSE351 – Computer Networks

Project Report

Supervised by:

Dr. Ayman Bahaa Eldin

Submitted by:

Mostafa Essam Mohamed Bayomy	18P9203
Basel Mohamed Ramadan	18P8121

Introduction:

This is a Simple Proxy Server code that can take requests from the client then processes the request and sends back a response message to the client.

The proxy Server receives a message from the client then forwards the client's request to the web server, then receives the response from the web server and forwards it to the client.

A few implementations are added to the proxy server:

1- Cashing: Cache functionality is added to store the web pages if it's the first time for them to be requested (If it's not found in the cache, it is stored).

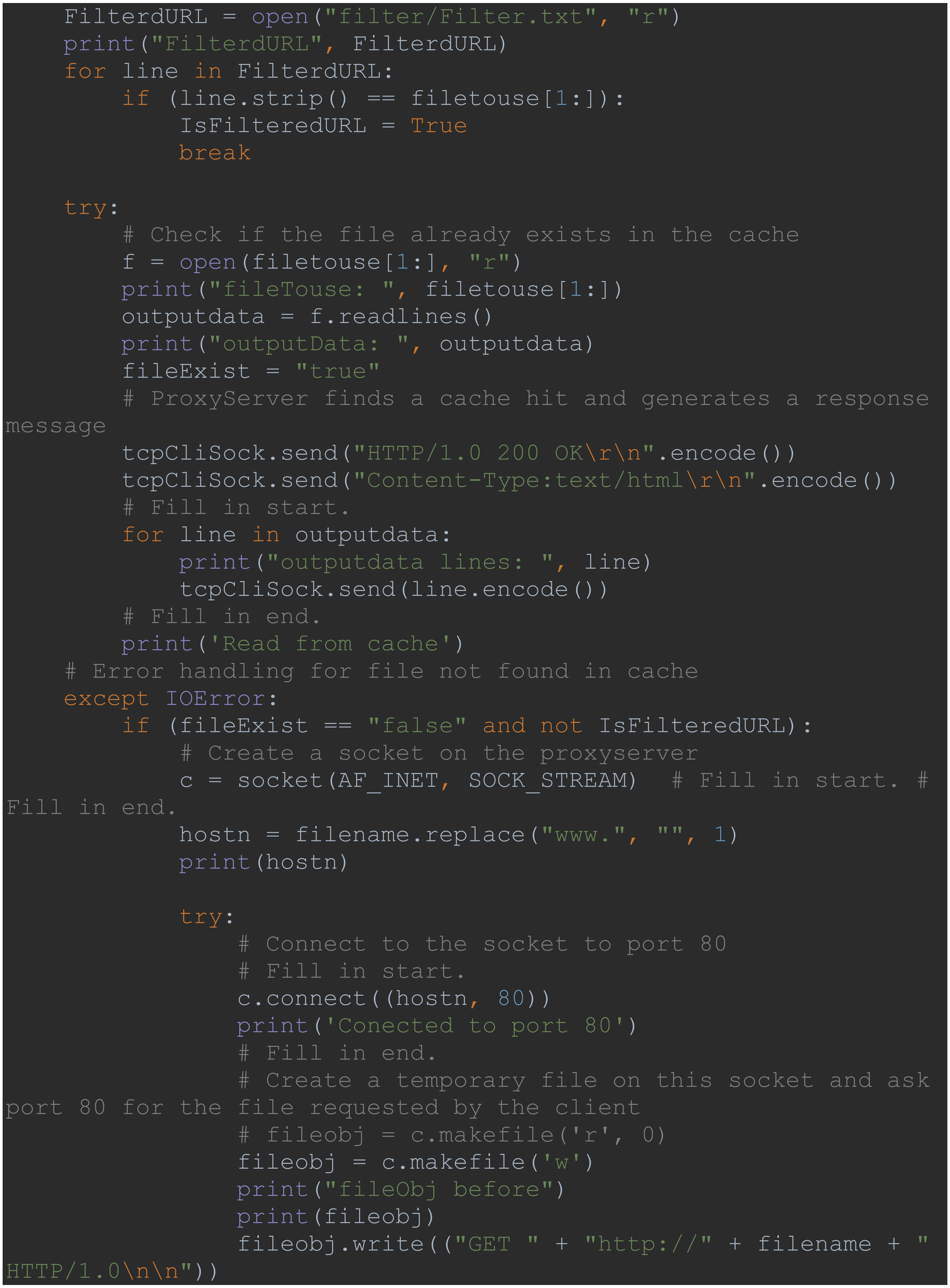
2- Error Handling: the "404 Not found" error is now handled and will appear to the user instead of an empty body.

3- URL Filter: a filter is implemented where when the server receives a request for a specific URL and that URL is listed in the filter data, then it returns a message that says "Access Denied".

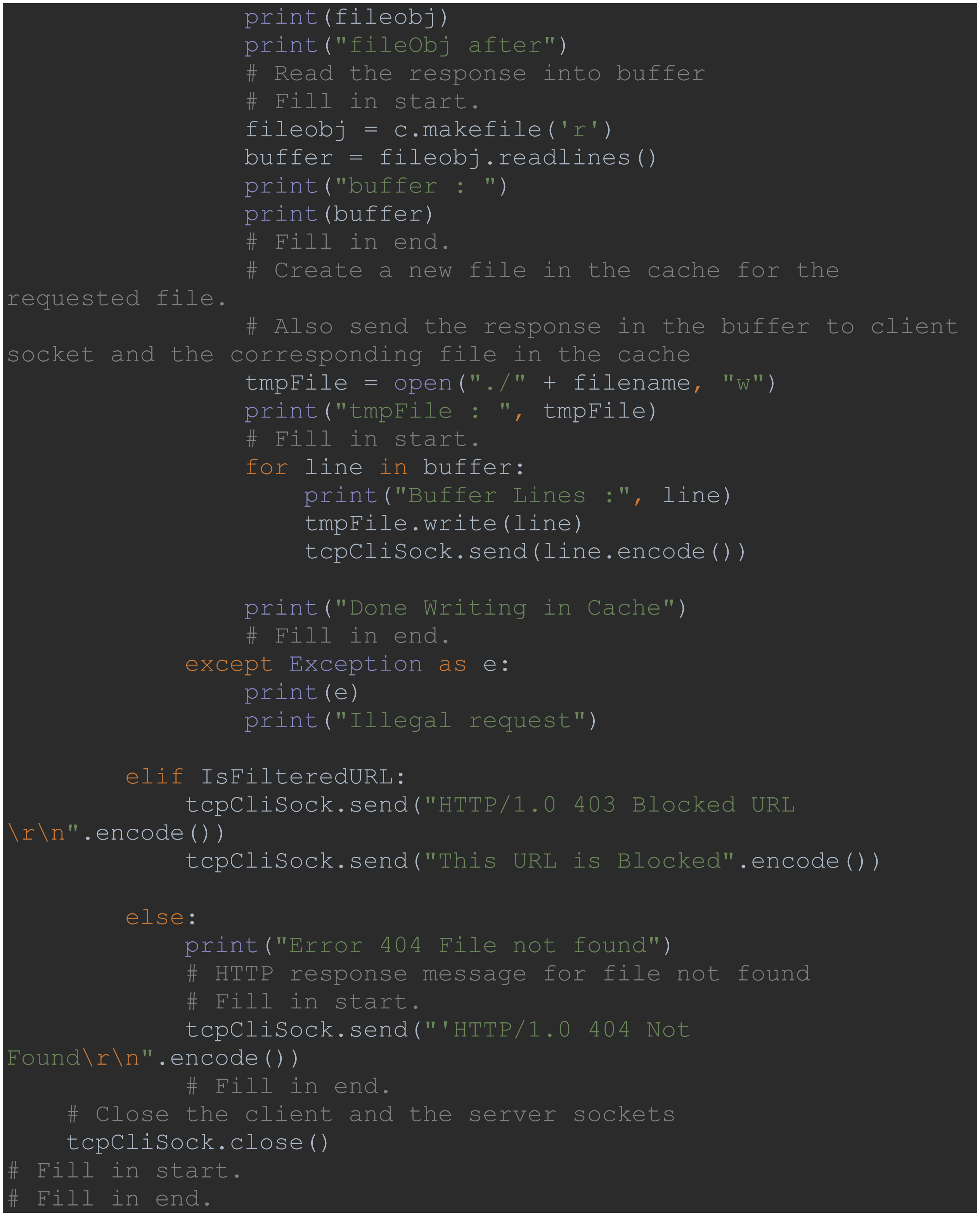
GitHub link: https://github.com/3ssam21/Proxy-Server

Project Code:

```
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)
print("*** Socket successfully created ***")

# Fill in start.
tcpSerProt = 8888
tcpSerSock.bind(('', tcpSerProt))
print("socket binded to %s" % (tcpSerProt))

# put the socket into listening mode
tcpSerSock.listen(5)
print("-socket is listening-")
# Fill in end.

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).decode()  # Fill in start. #
Fill in end.
    print(message)

    # Extract the filename from the given message
    print(message.split()[1])
    filename = message.split()[1].partition("/")[2]
    print(filename)
    fileExist = "false"
    filetouse = "/" + filename
    print(filetouse)

    # Implementing the URL Filter to block Sites
    IsFilteredURL = False
```



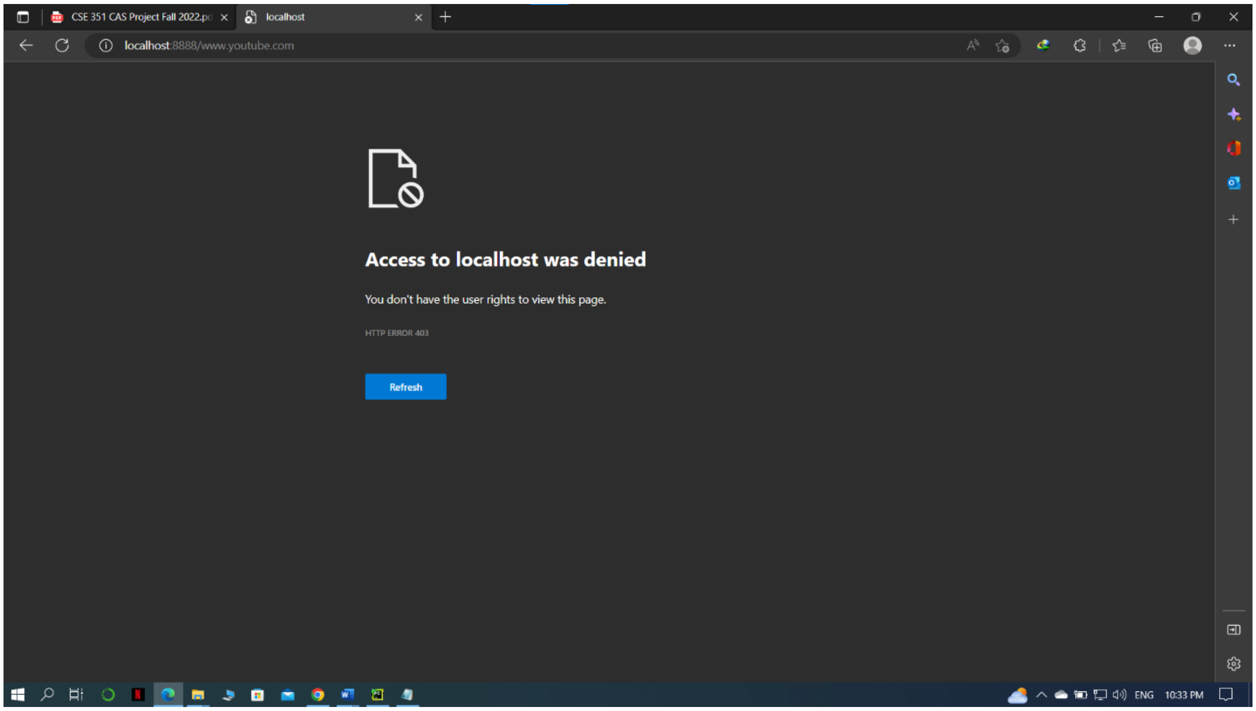
```
    FilterdURL = open("filter/Filter.txt", "r")
    print("FilterdURL", FilterdURL)
    for line in FilterdURL:
        if (line.strip() == filetouse[1:]):
            IsFilteredURL = True
            break

    try:
        # Check if the file already exists in the cache
        f = open(filetouse[1:], "r")
        print("fileTouse: ", filetouse[1:])
        outputdata = f.readlines()
        print("outputData: ", outputdata)
        fileExist = "true"
        # ProxyServer finds a cache hit and generates a response
message
        tcpCliSock.send("HTTP/1.0 200 OK\r\n".encode())
        tcpCliSock.send("Content-Type:text/html\r\n".encode())
        # Fill in start.
        for line in outputdata:
            print("outputdata lines: ", line)
            tcpCliSock.send(line.encode())
        # Fill in end.
        print('Read from cache')
    # Error handling for file not found in cache
    except IOError:
        if (fileExist == "false" and not IsFilteredURL):
            # Create a socket on the proxyserver
            c = socket(AF_INET, SOCK_STREAM)  # Fill in start. #
Fill in end.
            hostn = filename.replace("www.", "", 1)
            print(hostn)

            try:
                # Connect to the socket to port 80
                # Fill in start.
                c.connect((hostn, 80))
                print('Conected to port 80')
                # Fill in end.
                # Create a temporary file on this socket and ask
port 80 for the file requested by the client
                # fileobj = c.makefile('r', 0)
                fileobj = c.makefile('w')
                print("fileObj before")
                print(fileobj)
                fileobj.write(("GET " + "http://" + filename + "
HTTP/1.0\n\n"))
```



```
                print(fileobj)
                print("fileObj after")
                # Read the response into buffer
                # Fill in start.
                fileobj = c.makefile('r')
                buffer = fileobj.readlines()
                print("buffer : ")
                print(buffer)
                # Fill in end.
                # 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
                tmpFile = open("./" + filename, "w")
                print("tmpFile : ", tmpFile)
                # Fill in start.
                for line in buffer:
                    print("Buffer Lines :", line)
                    tmpFile.write(line)
                    tcpCliSock.send(line.encode())

                print("Done Writing in Cache")
                # Fill in end.
            except Exception as e:
                print(e)
                print("Illegal request")

        elif IsFilteredURL:
            tcpCliSock.send("HTTP/1.0 403 Blocked URL
\r\n".encode())
            tcpCliSock.send("This URL is Blocked".encode())

        else:
            print("Error 404 File not found")
            # HTTP response message for file not found
            # Fill in start.
            tcpCliSock.send("'HTTP/1.0 404 Not
Found\r\n".encode())
            # Fill in end.
    # Close the client and the server sockets
    tcpCliSock.close()
# Fill in start.
# Fill in end.
```

Some Screenshots:

- Entering a blocked URL (www.youtube.com)

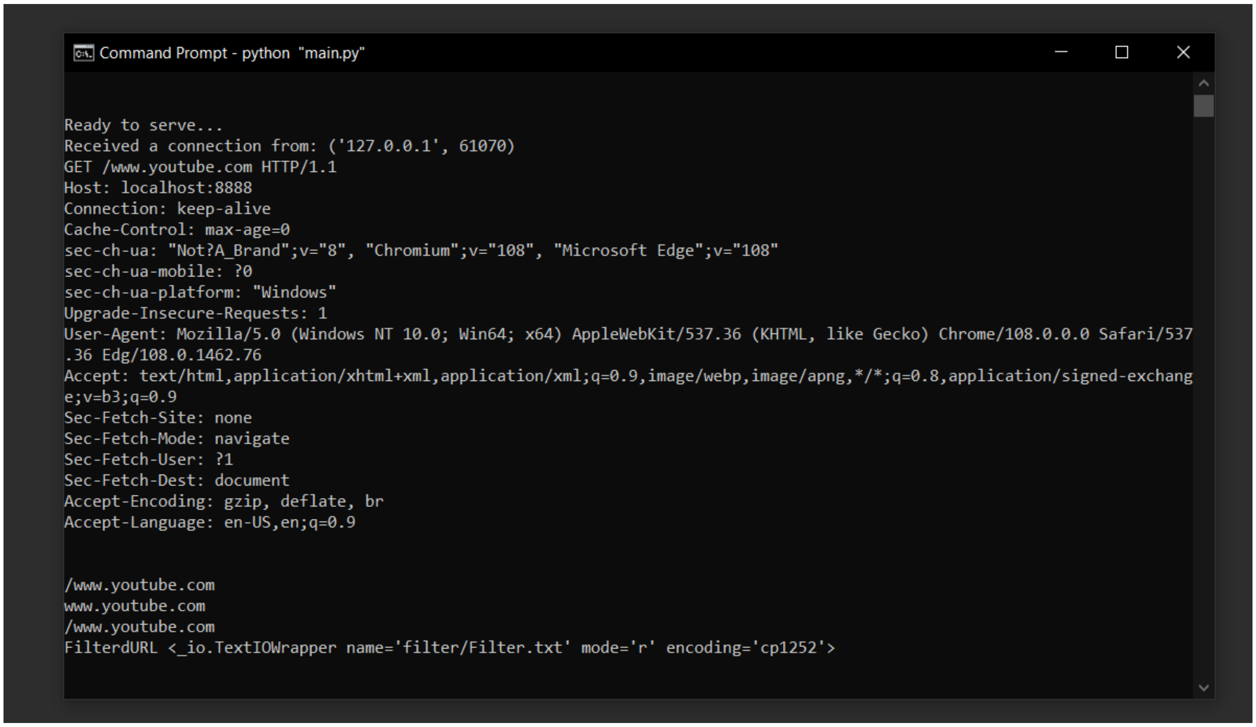


localhost:8888/www.youtube.com

Access to localhost was denied

You don't have the user rights to view this page.

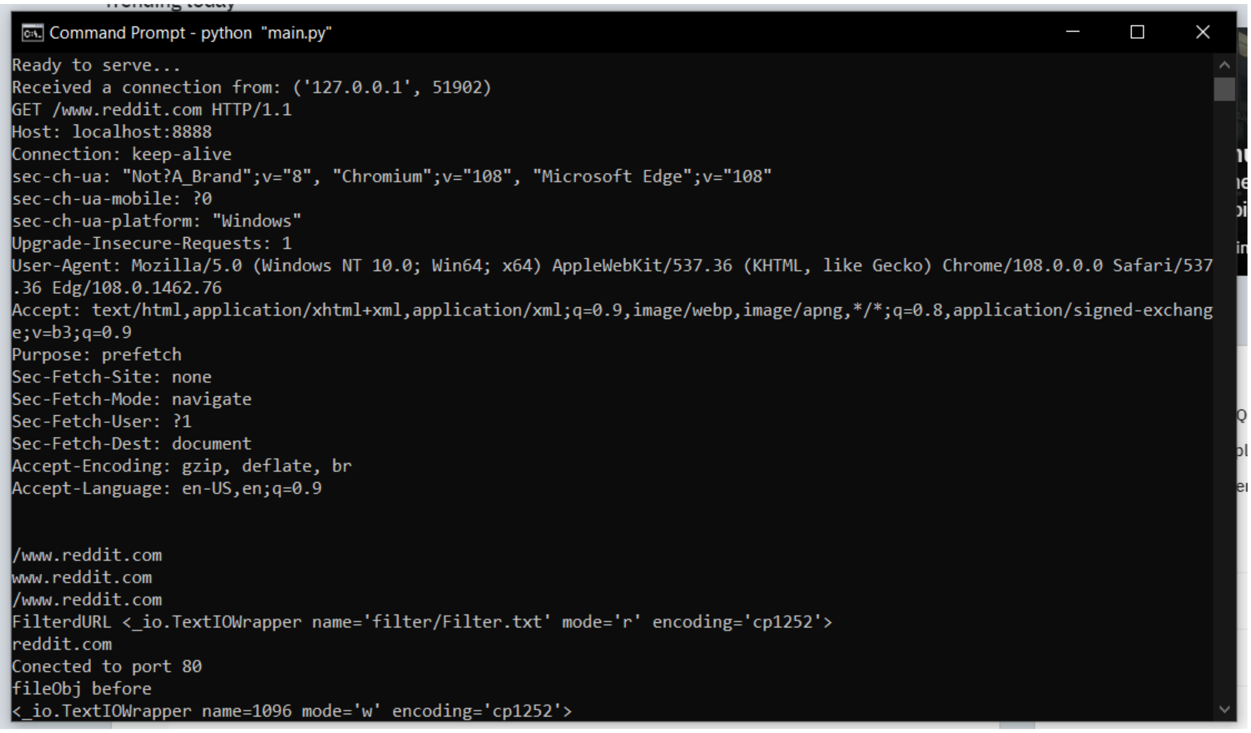
HTTP ERROR 403



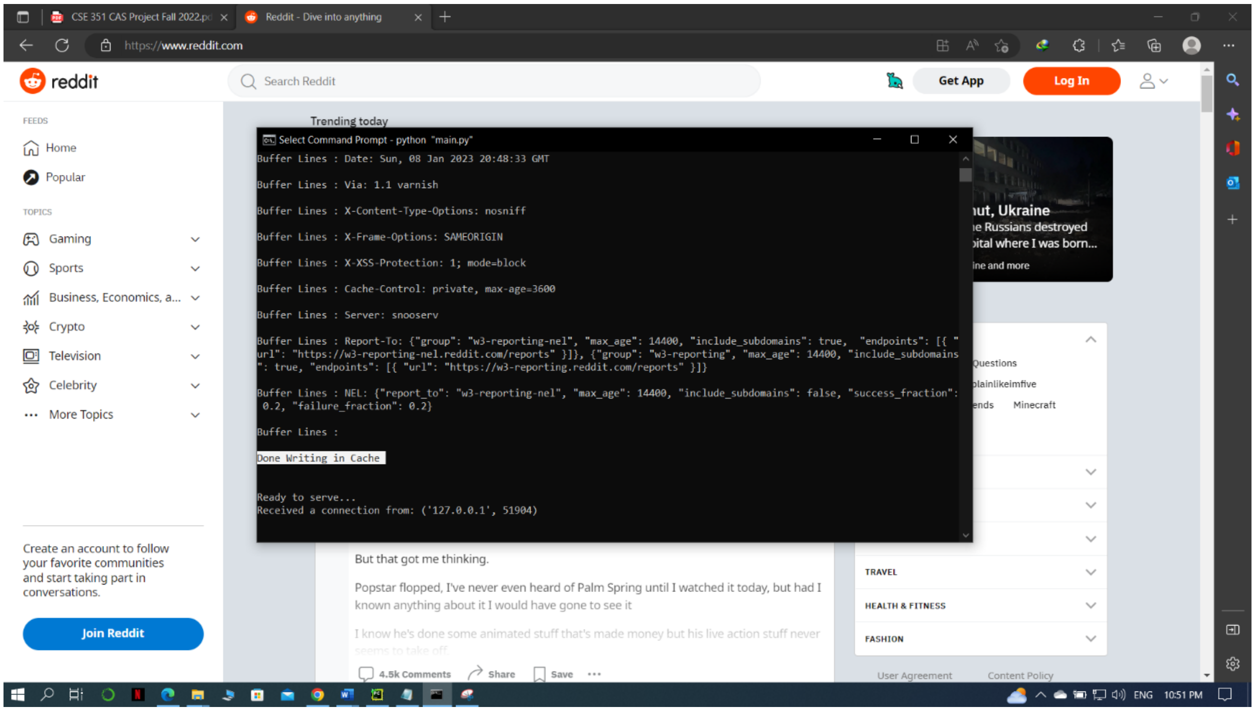
```
Ready to serve...
Received a connection from: ('127.0.0.1', 61070)
GET /www.youtube.com HTTP/1.1
Host: localhost:8888
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-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/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9


/www.youtube.com
www.youtube.com
/www.youtube.com
FilterdURL <_io.TextIOWrapper name='filter/Filter.txt' mode='r' encoding='cp1252'>
```

- Entering a new URL NOT stored in Cache

(www.reddit.com)



```
Ready to serve...
Received a connection from: ('127.0.0.1', 51902)
GET /www.reddit.com HTTP/1.1
Host: localhost:8888
Connection: keep-alive
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-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/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.9
Purpose: prefetch
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9


/www.reddit.com
www.reddit.com
/www.reddit.com
FilterdURL <_io.TextIOWrapper name='filter/Filter.txt' mode='r' encoding='cp1252'>
reddit.com
Conected to port 80
fileObj before
<_io.TextIOWrapper name=1096 mode='w' encoding='cp1252'>
```



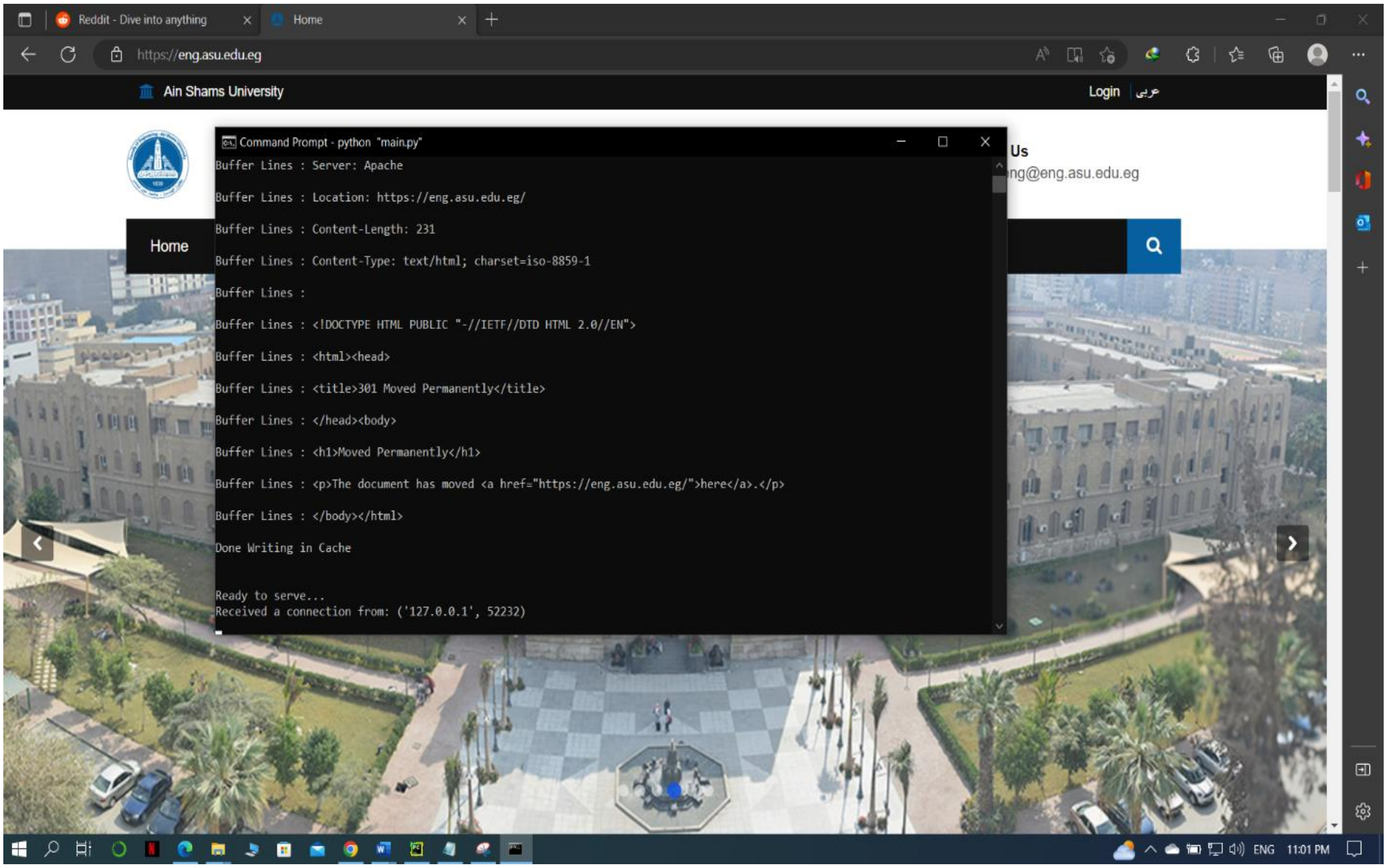
```
Buffer Lines : Date: Sun, 08 Jan 2023 20:48:33 GMT

Buffer Lines : Via: 1.1 varnish

Buffer Lines : X-Content-Type-Options: nosniff

Buffer Lines : X-Frame-Options: SAMEORIGIN

Buffer Lines : X-XSS-Protection: 1; mode=block

Buffer Lines : Cache-Control: private, max-age=3600

Buffer Lines : Server: snooserv

Buffer Lines : Report-To: {"group": "w3-reporting-nel", "max_age": 14400, "include_subdomains": true,  "endpoints": [{ "
url": "https://w3-reporting-nel.reddit.com/reports" }]}, {"group": "w3-reporting", "max_age": 14400, "include_subdomains
": true, "endpoints": [{ "url": "https://w3-reporting.reddit.com/reports" }]}

Buffer Lines : NEL: {"report_to": "w3-reporting-nel", "max_age": 14400, "include_subdomains": false, "success_fraction":
 0.2, "failure_fraction": 0.2}

Buffer Lines :

Done Writing in Cache


Ready to serve...
Received a connection from: ('127.0.0.1', 51904)
```

But that got me thinking.

Popstar flopped, I've never even heard of Palm Spring until I watched it today, but had I known anything about it I would have gone to see it

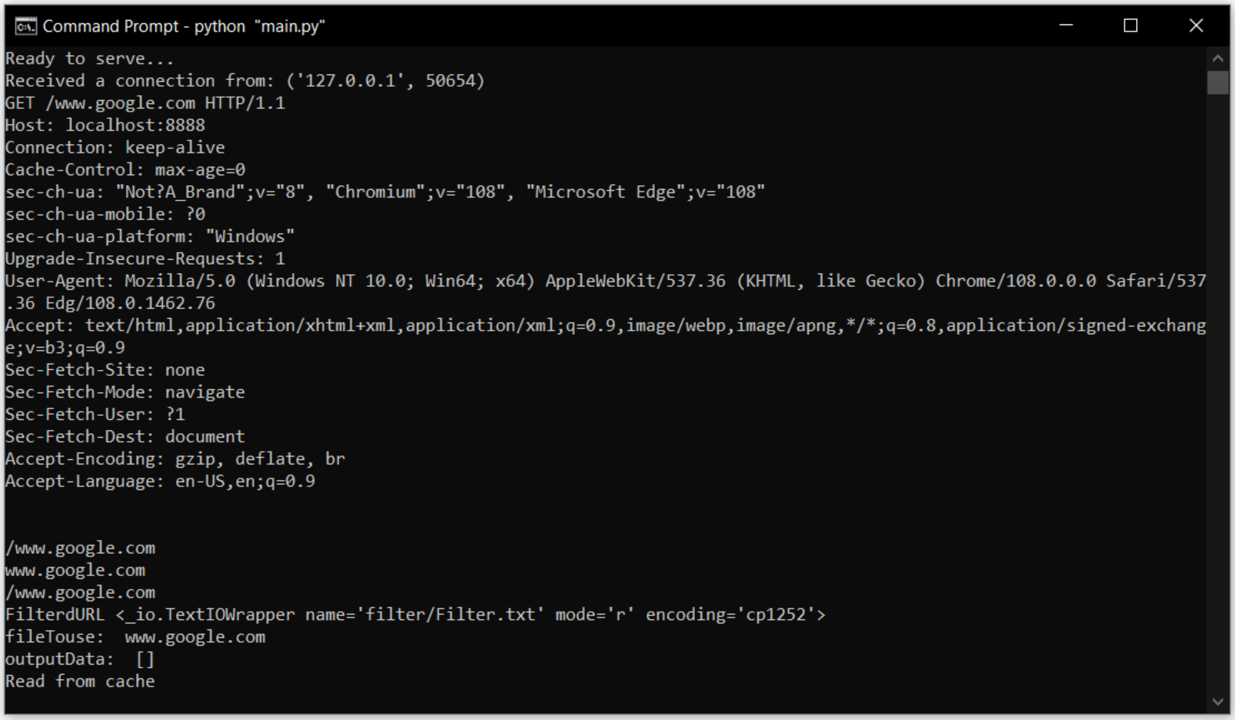
I know he's done some animated stuff that's made money but his live action stuff never

(eng.asu.edu.eg)



```
Buffer Lines : Server: Apache

Buffer Lines : Location: https://eng.asu.edu.eg/

Buffer Lines : Content-Length: 231

Buffer Lines : Content-Type: text/html; charset=iso-8859-1

Buffer Lines :

Buffer Lines : <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

Buffer Lines : <html><head>

Buffer Lines : <title>301 Moved Permanently</title>

Buffer Lines : </head><body>

Buffer Lines : <h1>Moved Permanently</h1>

Buffer Lines : <p>The document has moved <a href="https://eng.asu.edu.eg/">here</a>.</p>

Buffer Lines : </body></html>

Done Writing in Cache


Ready to serve...
Received a connection from: ('127.0.0.1', 52232)
```

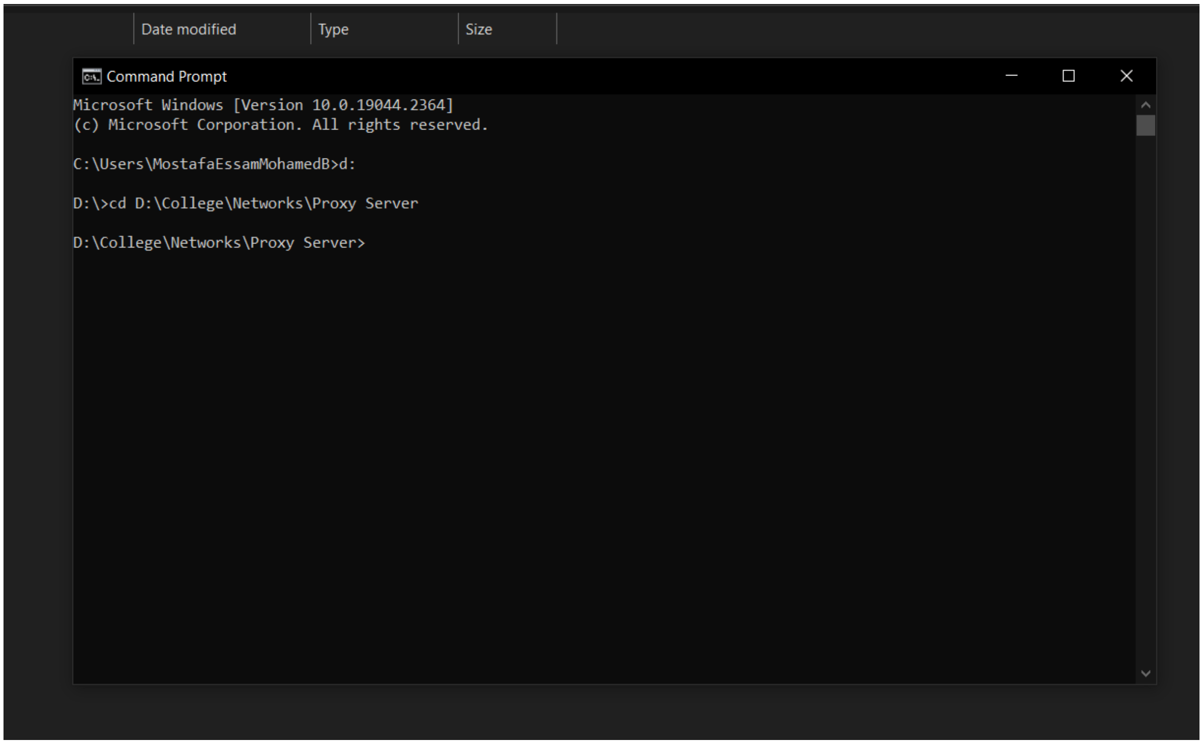
- Entering a URL stored in Cache



```
Ready to serve...
Received a connection from: ('127.0.0.1', 50654)
GET /www.google.com HTTP/1.1
Host: localhost:8888
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Not?A_Brand";v="8", "Chromium";v="108", "Microsoft Edge";v="108"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-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/108.0.1462.76
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9


/www.google.com
www.google.com
/www.google.com
FilterdURL <_io.TextIOWrapper name='filter/Filter.txt' mode='r' encoding='cp1252'>
fileTouse:  www.google.com
outputData:  []
Read from cache
```

How to run cmd:

To run the code on cmd you need to change the directory as follows



```
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MostafaEssamMohamedB>d:

D:\>cd D:\College\Networks\Proxy Server

D:\College\Networks\Proxy Server>
```

If your project is not in the default directory of cmd you need to change it.

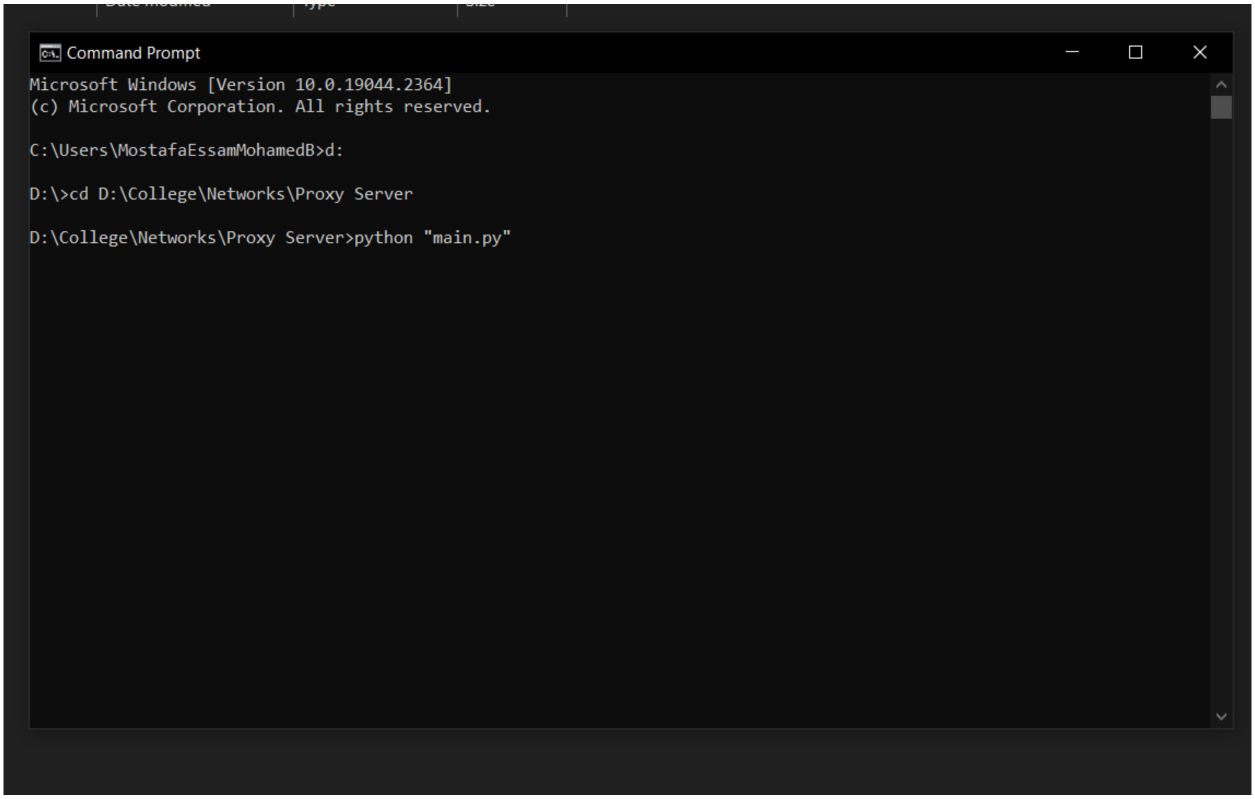
First enter “d:” to change from C Drive to D Drive or any other drive your project is in.

Then enter “cd “ followed by the actual path of the project, in this example(D:\College\Networks\Proxy Server).

This is done so that the cmd can read the “Filter.txt” file.

To run the project enter “python” followed by “ProjectName.py”, in this case it’s (python “main.py”).

Then hit enter and it will run.

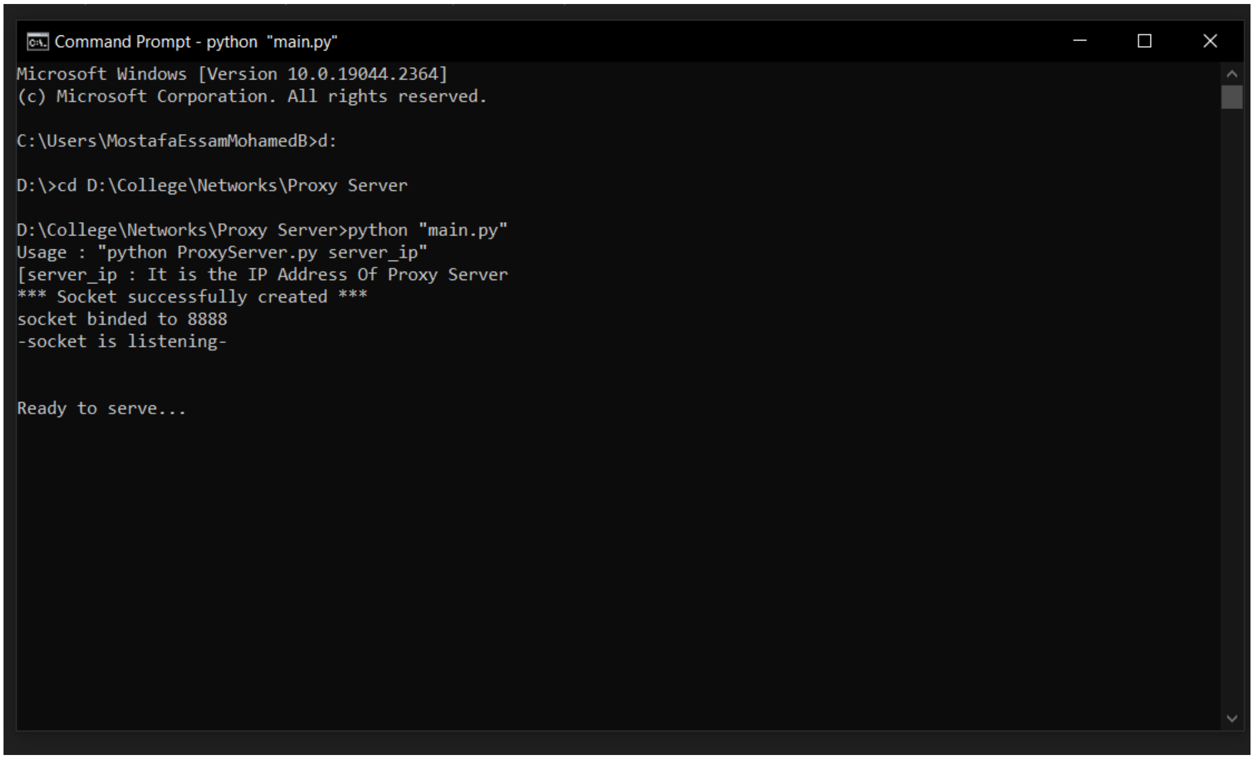


```
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MostafaEssamMohamedB>d:

D:\>cd D:\College\Networks\Proxy Server

D:\College\Networks\Proxy Server>python "main.py"
```

This is what it should look like when it runs

Then you go to your browser and enter the url in this format

http://localhost:8888/www.google.com



```
Microsoft Windows [Version 10.0.19044.2364]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MostafaEssamMohamedB>d:

D:\>cd D:\College\Networks\Proxy Server

D:\College\Networks\Proxy Server>python "main.py"
Usage : "python ProxyServer.py server_ip"
[server_ip : It is the IP Address Of Proxy Server
*** Socket successfully created ***
socket binded to 8888
-socket is listening-


Ready to serve...
```