Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

A Web Proxy Server

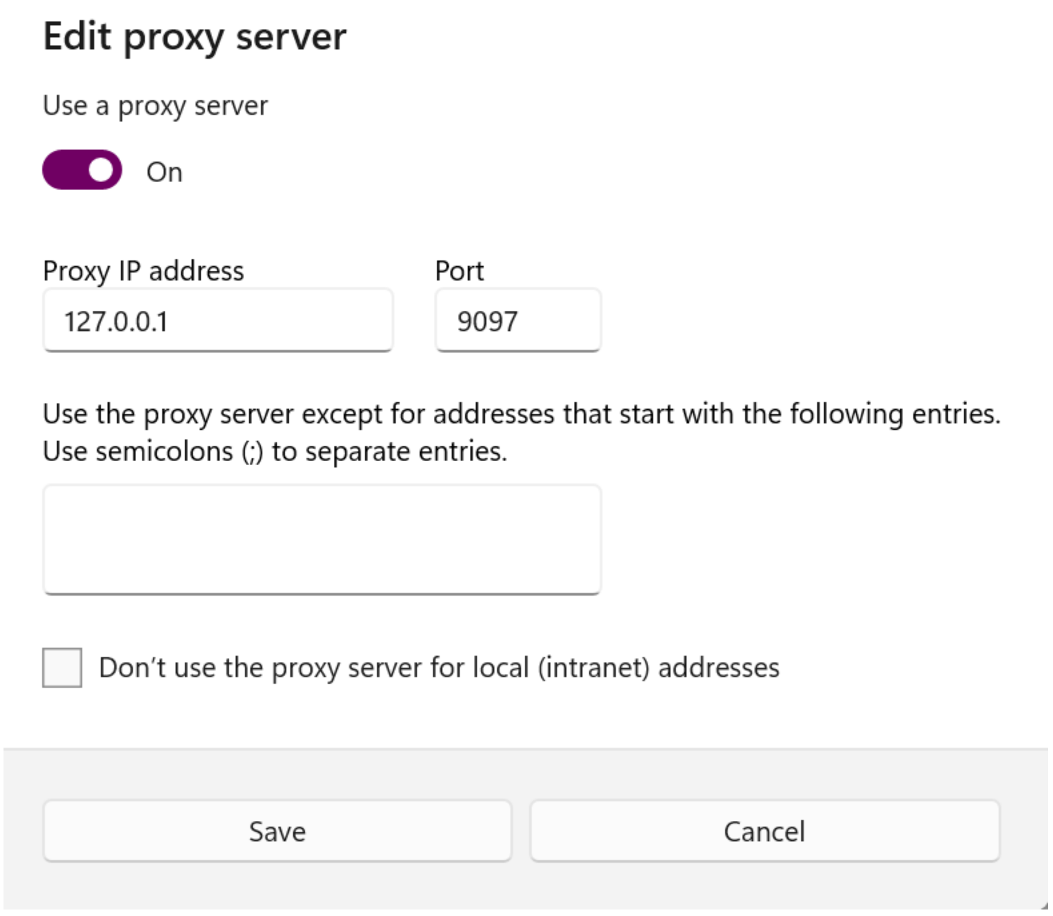
Advanced Computer Networks – CSU33032

Aislinn Addison-Smyth

Contents

1. Proxy Instructions

To run and test our web proxy server we must set up our proxy server in proxy settings. Here we choose a Proxy IP address and port number for the web proxy to work from. I have attached a screen clipping below in figure 1.1 of my setting alterations for a proxy server. I set the Proxy IP address to 127.0.0.1, also known as the localhost so I can test my server from my own machine. I set the port number to 9097, this is a random pot number which I will test my code from.


Edit proxy server

Use a proxy server

On

Proxy IP address: 127.0.0.1

Port: 9097

Use the proxy server except for addresses that start with the following entries.
Use semicolons (;) to separate entries.

☐ Don’t use the proxy server for local (intranet) addresses

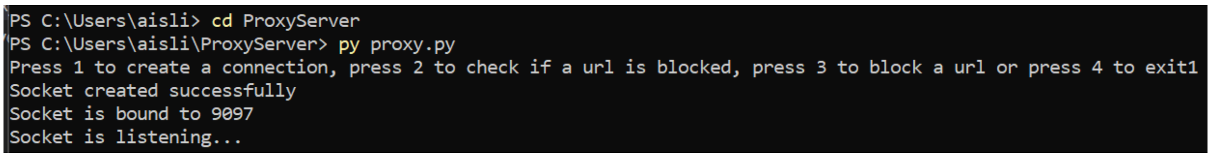
Figure 1.1: Setting alteration for a proxy server

2. Code Implementation

2.1 Responding to HTTP/HTTPS Requests and WebSocket connections

We were given five requirements that our program should be able to do: Respond to HTTP and HTTPS requests, handle WebSocket connections, block selected URLS, cache HTTPS requests, and our server should be threaded. A Web Proxy server which responds to HTTP/HTTPS requests uses the Transmission Control Protocol (TCP).

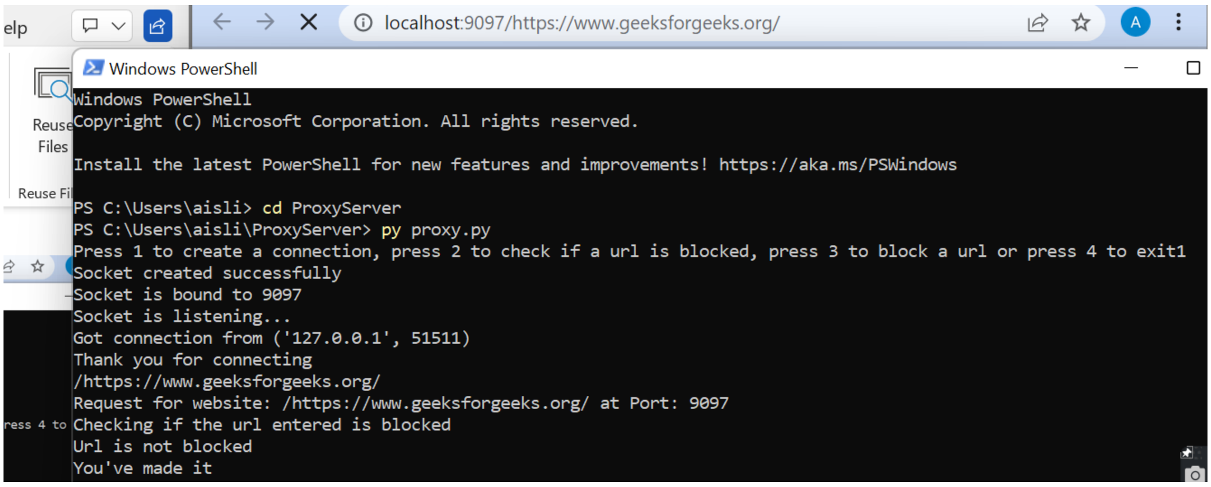
I used PowerShell as my management console, my code was implemented to be able to take user input.



```
PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit1
Socket created successfully
Socket is bound to 9097
Socket is listening...
```

Figure 1.2: User input options

In figure 1.2, we can see the different options we are given from the program, I have selected 1, this option creates our socket which connects to port 9097 and listens for any interactions on this port number.



```
localhost:9097/https://www.geeksforgeeks.org/
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit1
Socket created successfully
Socket is bound to 9097
Socket is listening...
Got connection from ('127.0.0.1', 51511)
Thank you for connecting
/https://www.geeksforgeeks.org/
Request for website: /https://www.geeksforgeeks.org/ at Port: 9097
Checking if the url entered is blocked
Url is not blocked
You've made it
```

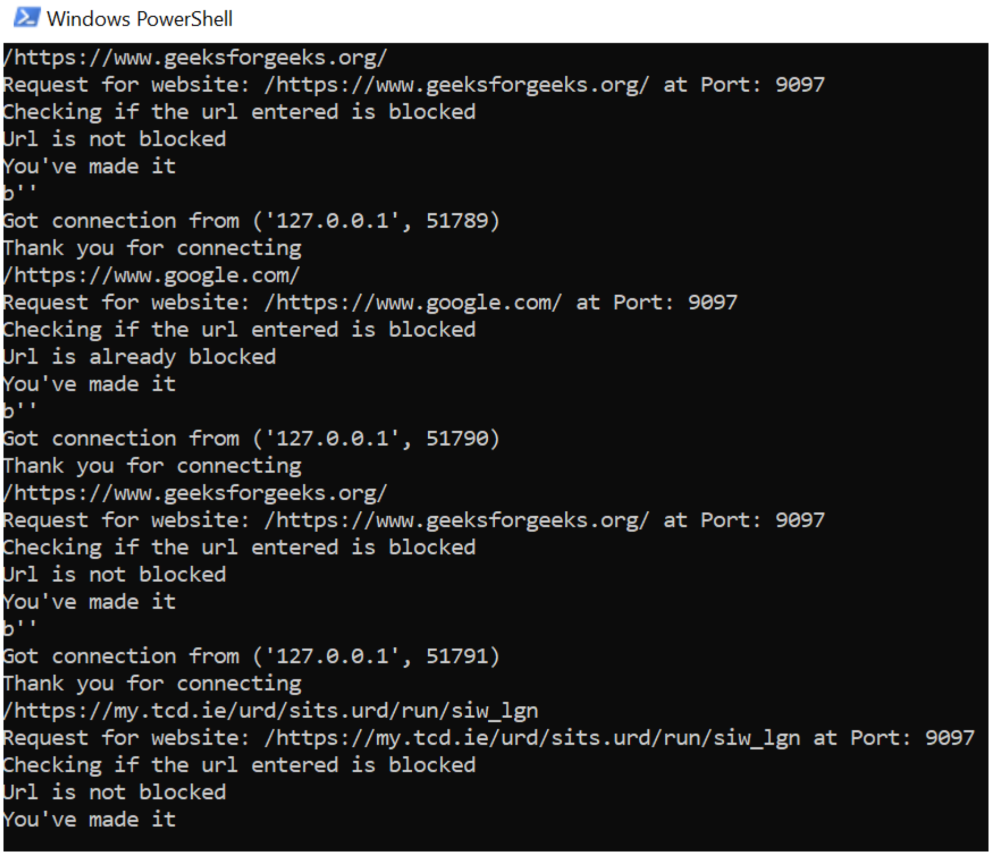
Figure 1.3: Request for https://www.geeksforgeeks.org/on localhost port number 9097.

By requesting another website consecutively, we can see that I have coded a threaded server, it can handle multiple requests at the same time Although we did not choose the option to check if a url is blocked, once you choose to create a connection, my requests function will check with the blockedUrls function to check whether to give access to the url or not.

2.2 Threaded Server

The code does not break if you attempt to send through multiple requests from your browser. It allows access and displays the browser in the management console. We can see that for geeksforgeeks, this website was not blocked, however, google was a blocked url.

I created a text document which contains all blocked urls, I will demonstrate the blocked urls later in this document.



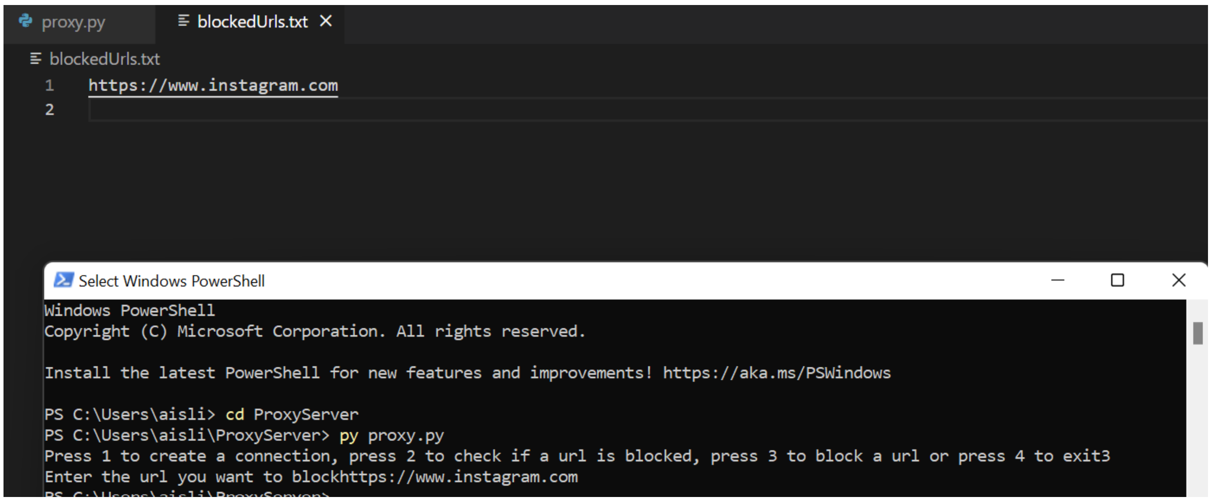
Windows PowerShell

```
/https://www.geeksforgeeks.org/
Request for website: /https://www.geeksforgeeks.org/ at Port: 9097
Checking if the url entered is blocked
Url is not blocked
You've made it
b''
Got connection from ('127.0.0.1', 51789)
Thank you for connecting
/https://www.google.com/
Request for website: /https://www.google.com/ at Port: 9097
Checking if the url entered is blocked
Url is already blocked
You've made it
b''
Got connection from ('127.0.0.1', 51790)
Thank you for connecting
/https://www.geeksforgeeks.org/
Request for website: /https://www.geeksforgeeks.org/ at Port: 9097
Checking if the url entered is blocked
Url is not blocked
You've made it
b''
Got connection from ('127.0.0.1', 51791)
Thank you for connecting
/https://my.tcd.ie/urd/sits.urd/run/siw_lgn
Request for website: /https://my.tcd.ie/urd/sits.urd/run/siw_lgn at Port: 9097
Checking if the url entered is blocked
Url is not blocked
You've made it
```

Figure 1.4: Demonstration of a threaded server with unblocked and blocked urls.

2.3 Blocked URLS

In figure 1.5, I selected option 3 which was to block an inputted url, for demonstration purposes I cleared my blockedUrls from my text document. I inputted Instagram as the website to be blocked and we can see from the screenshot that it was successfully written in.

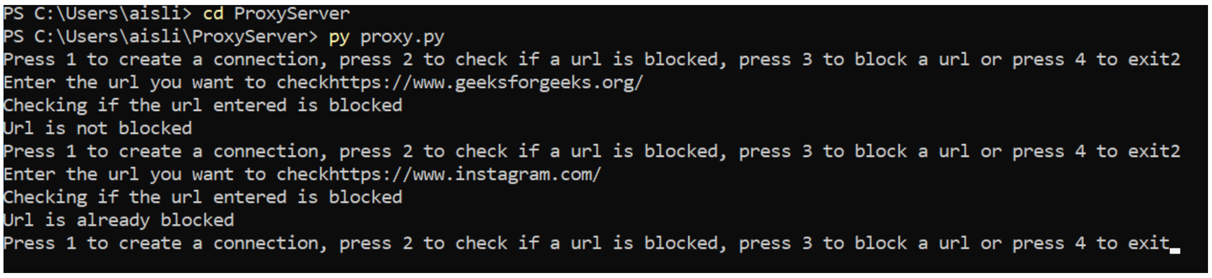


```
https://www.instagram.com
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit3
Enter the url you want to blockhttps://www.instagram.com
```

Figure 1.5: An inputted url being blocked.

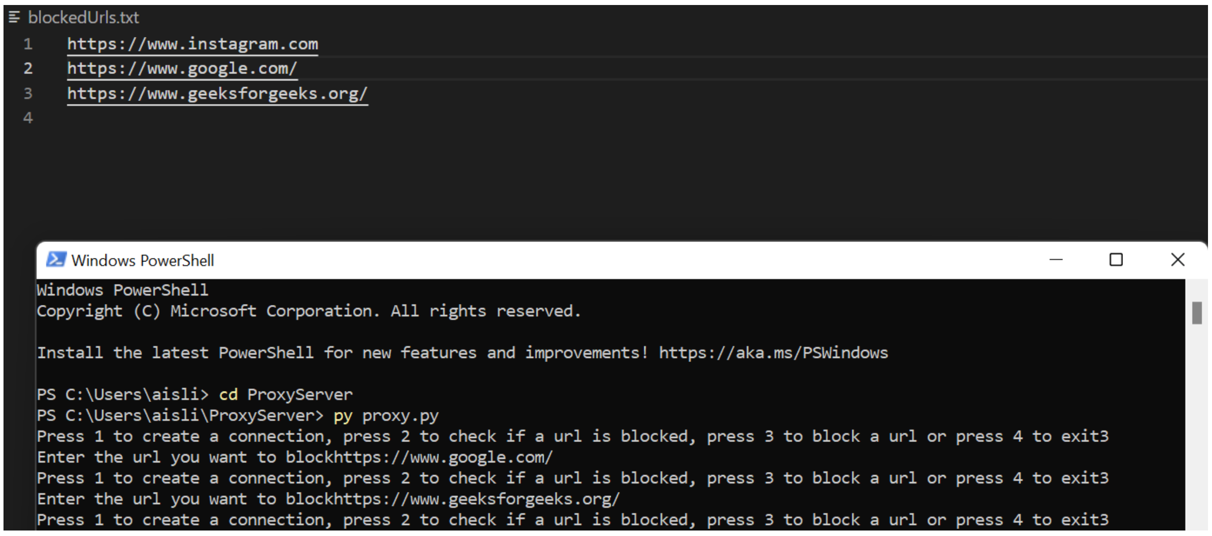
Now that we are aware of https://www.instagram.com/ is blocked, we can test this by selecting option 2, checking if a url is blocked, see figure 1.6. I used geeksforgeeks to show that if the site is not in in the text document it will return a ‘Url is not blocked’ message. We can see once we input Instagram, it shows us that the Url is already blocked.



```
PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit2
Enter the url you want to checkhttps://www.geeksforgeeks.org/
Checking if the url entered is blocked
Url is not blocked
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit2
Enter the url you want to checkhttps://www.instagram.com/
Checking if the url entered is blocked
Url is already blocked
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit
```

Figure 1.6: Testing different URLs which are blocked and unblocked.

In Figure 1.7 we can see a further demonstration of multiple websites being blocked to the same file.



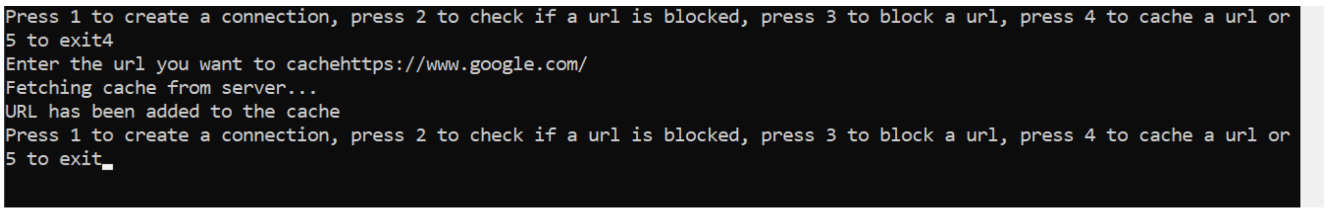
```
blockedUrls.txt
https://www.instagram.com
https://www.google.com/
https://www.geeksforgeeks.org/
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit3
Enter the url you want to blockhttps://www.google.com/
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit3
Enter the url you want to blockhttps://www.geeksforgeeks.org/
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url or press 4 to exit3
```

Figure 1.7: Multiple websites being blocked.

2.4 Cached URLs

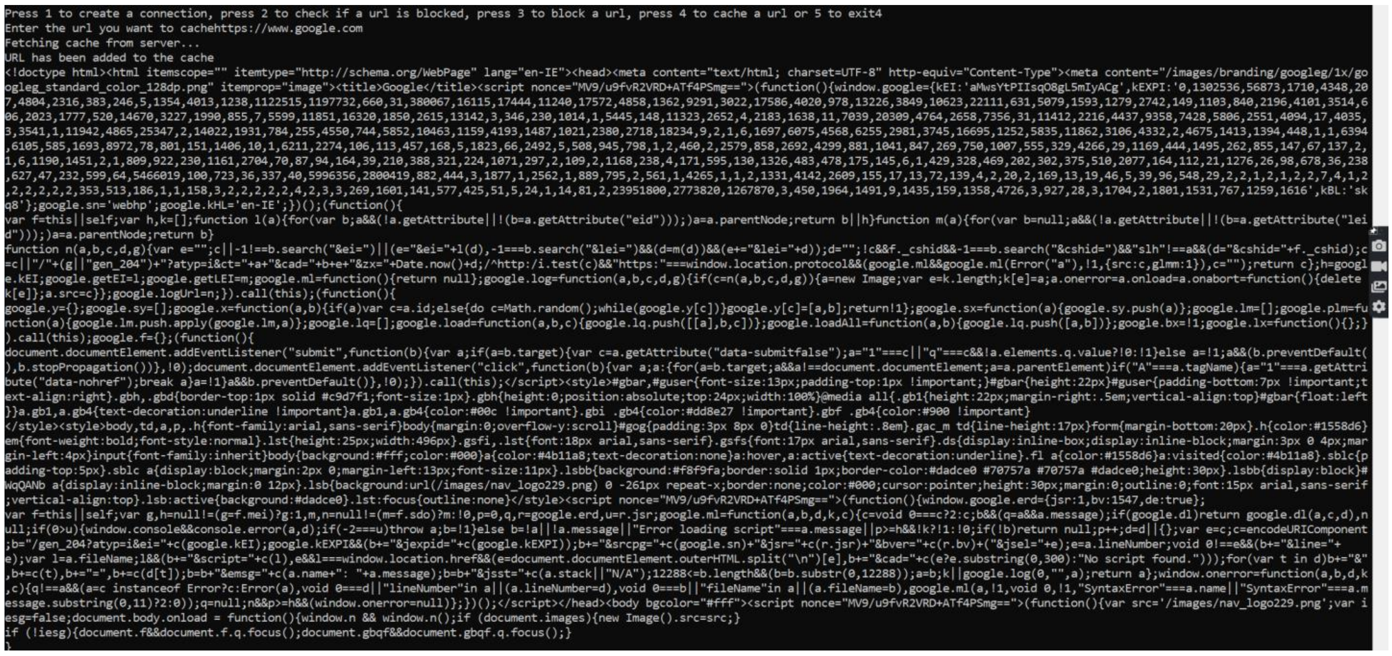
We also have an option to cache URLs (added at the end of the report, therefore in previous screenshots ‘press 4 to cache a url’ was not present). If we select ‘4’, we are given the option to cache our url, see figure 1.8.



```
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url, press 4 to cache a url or
5 to exit4
Enter the url you want to cachehttps://www.google.com/
Fetching cache from server...
URL has been added to the cache
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url, press 4 to cache a url or
5 to exit_
```

Figure 1.8: Caching a url from google.

If we print cache[url] we are met with the image from figure 1.9.

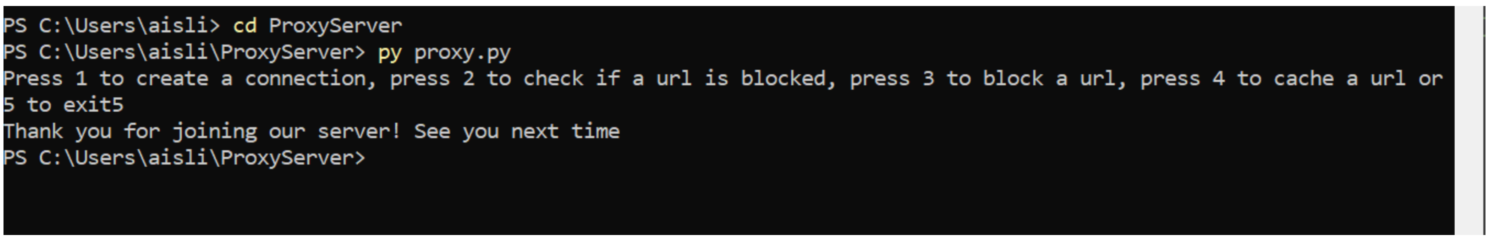


```
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url, press 4 to cache a url or 5 to exit4
Enter the url you want to cachehttps://www.google.com
Fetching cache from server...
URL has been added to the cache
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IE"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/go
ogleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="MV9/u9fvR2VRD+ATf4PSmg==">(function(){window.google={kEI:'aMwsYtPIIsqO8gL5mIyACg',kEXPI:'0,1302536,56873,1710,4348,20
7,4804,2316,383,246,5,1354,4013,1238,1122515,1197732,660,31,380067,16115,17444,11240,17572,4858,1362,9291,3022,17586,4020,978,13226,3849,10623,22111,631,5079,1593,1279,2742,149,1103,840,2196,4101,3514,6
06,2023,1777,520,14670,3227,1990,855,7,5599,11851,16320,1850,2615,13142,3,346,230,1014,1,5445,148,11323,2652,4,2183,1638,11,7039,20309,4764,2658,7356,31,11412,2216,4437,9358,7428,5806,2551,4094,17,4035,
3,3541,1,11942,4865,25347,2,14022,1931,784,255,4550,744,5852,10463,1159,4193,1487,1021,2380,2718,18234,9,2,1,6,1697,6075,4568,6255,2981,3745,16695,1252,5835,11862,3106,4332,2,4675,1413,1394,448,1,1,6394
,6105,585,1693,8972,78,801,151,1406,10,1,6211,2274,106,113,457,168,5,1823,66,2492,5,508,945,798,1,2,460,2,2579,858,2692,4299,881,1041,847,269,750,1007,555,329,4266,29,1169,444,1495,262,855,147,67,137,2,
1,6,1190,1451,2,1,809,922,230,1161,2704,70,87,94,164,39,210,388,321,224,1071,297,2,109,2,1168,238,4,171,595,130,1326,483,478,175,145,6,1,429,328,469,202,302,375,510,2077,164,112,21,1276,26,98,678,36,238
,627,47,232,599,64,5466019,100,723,36,337,40,5996356,2800419,882,444,3,1877,1,2562,1,889,795,2,561,1,4265,1,1,2,1331,4142,2609,155,17,13,72,139,4,2,20,2,169,13,19,46,5,39,96,548,29,2,2,1,2,1,2,2,7,4,1,2
,2,2,2,2,2,353,513,186,1,1,158,3,2,2,2,2,2,4,2,3,3,269,1601,141,577,425,51,5,24,1,14,81,2,23951800,2773820,1267870,3,450,1964,1491,9,1435,159,1358,4726,3,927,28,3,1704,2,1801,1531,767,1259,1616',kBL:'sk
q8'};google.sn='webhp';google.kHL='en-IE';})();(function(){
var f=this||self;var h,k=[];function l(a){for(var b;a&&(!a.getAttribute||!(b=a.getAttribute("eid")));)a=a.parentNode;return b||h}function m(a){for(var b=null;a&&(!a.getAttribute||!(b=a.getAttribute("lei
d")));)a=a.parentNode;return b}
function n(a,b,c,d,g){var e="";c||-1!==b.search("&ei=")||(e="&ei="+l(d),-1===b.search("&lei=")&&(d=m(d))&&(e+="&lei="+d));d="";!c&&f._cshid&&-1===b.search("&cshid=")&&"slh"!==a&&(d="&cshid="+f._cshid);c
=c||"/"+(g||"gen_204")+"?atyp=i&ct="+a+"&cad="+b+e+"&zx="+Date.now()+d;/^http:/i.test(c)&&"https:"===window.location.protocol&&(google.ml&&google.ml(Error("a"),!1,{src:c,glmm:1}),c="");return c};h=googl
e.kEI;google.getEI=l;google.getLEI=m;google.ml=function(){return null};google.log=function(a,b,c,d,g){if(c=n(a,b,c,d,g)){a=new Image;var e=k.length;k[e]=a;a.onerror=a.onload=a.onabort=function(){delete
k[e]};a.src=c}};google.logUrl=n;}).call(this);(function(){
google.y={};google.sy=[];google.x=function(a,b){if(a)var c=a.id;else{do c=Math.random();while(google.y[c])}google.y[c]=[a,b];return!1};google.sx=function(a){google.sy.push(a)};google.lm=[];google.plm=fu
nction(a){google.lm.push.apply(google.lm,a)};google.lq=[];google.load=function(a,b,c){google.lq.push([[a],b,c])};google.loadAll=function(a,b){google.lq.push([a,b])};google.bx=!1;google.lx=function(){};}
).call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a="1"===c||"q"===c&&!a.elements.q.value?!0:!1}else a=!1;a&&(b.preventDefault(
),b.stopPropagation())},!0);document.documentElement.addEventListener("click",function(b){var a;a:{for(a=b.target;a&&a!==document.documentElement;a=a.parentElement)if("A"===a.tagName){a="1"===a.getAttri
bute("data-nohref");break a}a=!1}a&&b.preventDefault()},!0);}).call(this);</script><style>#gbar,#guser{font-size:13px;padding-top:1px !important;}#gbar{height:22px}#guser{padding-bottom:7px !important;t
ext-align:right}.gbh,.gbd{border-top:1px solid #c9d7f1;font-size:1px}.gbh{height:0;position:absolute;top:24px;width:100%}@media all{.gb1{height:22px;margin-right:.5em;vertical-align:top}#gbar{float:left
}}a.gb1,a.gb4{text-decoration:underline !important}a.gb1,a.gb4{color:#00c !important}.gbi .gb4{color:#dd8e27 !important}.gbf .gb4{color:#900 !important}
</style><style>body,td,a,p,.h{font-family:arial,sans-serif}body{margin:0;overflow-y:scroll}#gog{padding:3px 8px 0}td{line-height:.8em}.gac_m td{line-height:17px}form{margin-bottom:20px}.h{color:#1558d6}
em{font-weight:bold;font-style:normal}.lst{height:25px;width:496px}.gsfi,.lst{font:18px arial,sans-serif}.gsfs{font:17px arial,sans-serif}.ds{display:inline-box;display:inline-block;margin:3px 0 4px;mar
gin-left:4px}input{font-family:inherit}body{background:#fff;color:#000}a{color:#4b11a8;text-decoration:none}a:hover,a:active{text-decoration:underline}.fl a{color:#1558d6}a:visited{color:#4b11a8}.sblc{p
adding-top:5px}.sblc a{display:block;margin:2px 0;margin-left:13px;font-size:11px}.lsbb{background:#f8f9fa;border:solid 1px;border-color:#dadce0 #70757a #70757a #dadce0;height:30px}.lsbb{display:block}#
WqQANb a{display:inline-block;margin:0 12px}.lsb{background:url(/images/nav_logo229.png) 0 -261px repeat-x;border:none;color:#000;cursor:pointer;height:30px;margin:0;outline:0;font:15px arial,sans-serif
;vertical-align:top}.lsb:active{background:#dadce0}.lst:focus{outline:none}</style><script nonce="MV9/u9fvR2VRD+ATf4PSmg==">(function(){window.google.erd={jsr:1,bv:1547,de:true};
var f=this||self;var g,h=null!=(g=f.mei)?g:1,m,n=null!=(m=f.sdo)?m:!0,p=0,q,r=google.erd,u=r.jsr;google.ml=function(a,b,d,k,c){c=void 0===c?2:c;b&&(q=a&&a.message);if(google.dl)return google.dl(a,c,d),n
ull;if(0>u){window.console&&console.error(a,d);if(-2===u)throw a;b=!1}else b=!a||!a.message||"Error loading script"===a.message||p>=h&&!k?!1:!0;if(!b)return null;p++;d=d||{};var e=c;c=encodeURIComponent
;b="/gen_204?atyp=i&ei="+c(google.kEI);google.kEXPI&&(b+="&jexpid="+c(google.kEXPI));b+="&srcpg="+c(google.sn)+"&jsr="+c(r.jsr)+"&bver="+c(r.bv)+("&jsel="+e);e=a.lineNumber;void 0!==e&&(b+="&line="+
e);var l=a.fileName;l&&(b+="&script="+c(l),e&&l===window.location.href&&(e=document.documentElement.outerHTML.split("\n")[e],b+="&cad="+c(e?e.substring(0,300):"No script found.")));for(var t in d)b+="&"
,b+=c(t),b+="=",b+=c(d[t]);b=b+"&emsg="+c(a.name+": "+a.message);b=b+"&jsst="+c(a.stack||"N/A");12288<=b.length&&(b=b.substr(0,12288));a=b;k||google.log(0,"",a);return a};window.onerror=function(a,b,d,k
,c){q!==a&&(a=c instanceof Error?c:Error(a),void 0===d||"lineNumber"in a||(a.lineNumber=d),void 0===b||"fileName"in a||(a.fileName=b),google.ml(a,!1,void 0,!1,"SyntaxError"===a.name||"SyntaxError"===a.m
essage.substring(0,11)?2:0));q=null;n&&p>=h&&(window.onerror=null)};})();</script></head><body bgcolor="#fff"><script nonce="MV9/u9fvR2VRD+ATf4PSmg==">(function(){var src='/images/nav_logo229.png';var i
esg=false;document.body.onload = function(){window.n && window.n();if (document.images){new Image().src=src;}
if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}
}
```

Figure 1.9: The printed cache

3. Program Termination

The user has the ability at any given time to terminate the program, there is an option as seen in figure 2.0 which allows a user to input '5' and automatically terminate the server.



```
PS C:\Users\aisli> cd ProxyServer
PS C:\Users\aisli\ProxyServer> py proxy.py
Press 1 to create a connection, press 2 to check if a url is blocked, press 3 to block a url, press 4 to cache a url or
5 to exit5
Thank you for joining our server! See you next time
PS C:\Users\aisli\ProxyServer>
```

Figure 2.0: Program terminating due to user input.

4. Code

```python
from functools import cache
import socket
import threading
from urllib import request  #
import requests
import time

#Program is a threaded server, can handle multiple requests
#Program can also handle websocket connections
#Snippets of code altered from https://www.geeksforgeeks.org/creating-a-proxy-
webserver-in-python-set-1/

port = 9097
host = '127.0.0.1'

cache = {}      #collection of cached urls

def main():     #main takes user input and sends the code to the designated
functions accordingly
    start = int(input('Press 1 to create a connection, press 2 to check if a url
is blocked, press 3 to block a url, press 4 to cache a url or 5 to exit'))
    if(start == 1):
            create_socket()
    elif(start == 2):
        blocked = (input('Enter the url you want to check'))
        blockedUrls(blocked)
    elif(start == 3):
        blocked = (input('Enter the url you want to block'))
        blockUrl(blocked)
    elif(start == 4):
        cacheUrl = (input('Enter the url you want to cache'))
        cacheCheck(cacheUrl)
    elif(start ==5):
        print("Thank you for joining our server! See you next time")
        exit()
    else:
        ("A fatal error has occured, please restart the program.")
```

```
def create_socket():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Next create a socket
object
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1) #Re-use the
socket
    print ("Socket created successfully")       #socket creation

    s.bind((host, port))                         #Bind to the assigned port
    print ("Socket is bound to %s" %(port))
    s.listen(10)                                #put the socket into listening
mode
    print ("Socket is listening...")

    while True:                                 #a forever loop until we interrupt
it or an error occurs
        connection, addr = s.accept()           #Establish connection with client
        print ('Got connection from', addr)
        threadListener = threading.Thread(target = requestRcvd(s,connection),
args=(connection,addr)) #Ready to create a new thread for any request made and run
requestReceived
        threadListener.daemon = True
        threadListener.start()                  #start threads

def getUrl(request):
        first_line = str(request).split('\n')[0] #parse the first line
        url = first_line.split(' ')[1]           #get url
        return url

def getPort(url):
        http_pos = url.find("://")              #find pos of ://
        if(http_pos == -1):
            temp=url

        else:
            temp = url[(http_pos+3):]           #get the rest of url

        port_pos = temp.find(":")               #find the port pos (if any)
        webserver_pos = temp.find("/")          #find end of web server
        if webserver_pos == -1:
            webserver_pos = len(temp)

        webserver = ""
        port = -1
        if(port_pos==-1 or webserver_pos < port_pos):

            port = 9097  #default port
            webserver = temp[:webserver_pos]
        else: #specific port
            port = int((temp[(port_pos+1):])[:webserver_pos-port_pos-1])
            webserver = temp[:port_pos]
        return port

def blockedUrls(blocked):   #checking if a url is blocked in txt document
    print("Checking if the url entered is blocked")
    lines = open("blockedUrls.txt").read().splitlines()         #opens reads the
txt document blockedUrls
    for line in lines:                                      #for loop locating if
a url is in the txt document print URL is already blocked
        if (str(line) in (blocked)):
            print("Url is already blocked")
            return True
        else:
            print("Url is not blocked")                     #else print that URL
is not blocked
            return False

    main()

def blockUrl(blocked):                          #function to block a specified
url
    lines = [blocked]                           #blocked being the inputted url from
user
    with open('blockedUrls.txt', 'a') as f:             #'a' meaning appending to
blockedUrl txt document
        for line in lines:
            f.write(line)                               #writing the new url to
the txt doc
            f.write('\n')
    main()

def getCache(url):                              #Function to get the cache
    print("Fetching cache from server...")
    response = requests.get(url)
    return response.text

def cacheCheck(url):                            #Function to add a URL to the cache if
it is not already present
    if url not in cache:
        cache[url] = getCache(url)
        print('URL has been added to the cache')
        print(cache[url])
        main()
```

```python
def requestRcvd(s,connection):                 #socket conncting to the browser -
data to be sent to and from browser.
    print("Thank you for connecting")
    s = socket.socket()                        #socket creation
    request = connection.recv(1024)

    url = getUrl(request)    #Find url
    print(str(url))
    port = getPort(url)         #getting port number

    print("Request for website: " + url + " at Port: " + str(port))
    blockedUrls(url)

    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('127.0.0.1', port))
    s.sendall(request)
    while 1:
        print("You've made it")

        data = connection.recv(1024)
        #data = cacheCheck(url)
        print(str(data))
        if (len(data) > 0):
            s.send(data)        #Send to browser
            print("Working")
            main()
        else:
            break

if __name__ == "__main__":
    main()
```

5. Demonstration Video

Demonstration Video attached in submission in the off chance it does not open in this document.

Advanced Computer Network.mp4 (Command Line)