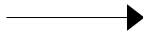


# ACN Programming Assignment-2: WWW

## PART-1: A Simple Web Client



This Python script allows the end user to communicate with the desired destination server and fetch the required resources and associated resources, and provides options to view or save the content as specified by the user, dumped at the server's end. The Local Client is designed to make just GET requests that to with just HTTP 1.0. It can be useful for web scraping, resource downloading, or analyzing web pages.

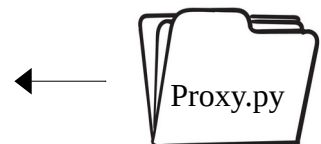
The client takes the following inputs:

- 1) Server IP/ Domain (Mandatory)
- 2) Server Port (Mandatory)
- 3) Proxy IP (optional)
- 4) Proxy Port (optional)

It is assumed that if the user specifies Proxy entries then it must want to go to destination via proxy server. Based on the input of the user the program generates HTTP/1.0 GET request packet and sends to the destined server. While receiving the corresponding response it first check the status of the response message to verify that whether its requested was served or not by the server. If served it separates the content from the headers and parse the response if it contains html kind of file otherwise it stores what it receives. While parsing if it finds links to the other resources, it in very similar fashion as mentioned above make non-persistent connection with the destined server to fetch the required stuff. At the end the web client stores whatever it recieved in a valid extension with an option to open that in decoded or raw version in browser or command line interface respectively.

## PART-2: A Simple Web Proxy Server

This multithreaded python program acts as a simple proxy server i.e. which just performs Proxying (Forwarding the request) and not caching the content. In other words it just an intermediary thing that takes clients requests and passes it to the destined location and vice versa. When a client request arrives at the proxy server it first separates the headers and the data content. Then it analyze the headers via some split operation on the headers to detect what is the dedstination host, its port and the path.

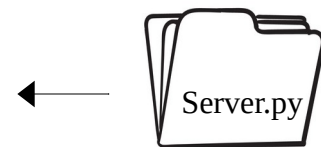


Now, using the extracted values of destiantion host, port and path it will modify the header as the path of the request line that arrived at the proxy is the complete URL and not just path. So this modification of header will convert the URL to path in the request line. After it modifies the HTTP

request it sends it to the destination server and waits for the response in loop. Once it start receiving the response it immediately sends them to the requesting client until no more response is received.

### **PART-3: A Simple Web Server**

This multithreaded Python script is listening for the client's request on a particular port. It is designed to server only GET request so if any user request other then the GET arrives at the local web server the it won't be appreciated by the server. Whenever it receives a GET request it parse that and checks that whether it has resources mentioned in that request. If so that it formulates the response packet with the required HTTP headers fields and throws to the requesting client. If not then it sends the status code corresponding to resource not found along with status phrase.



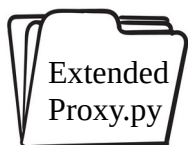
It is important to note that there will be a separate TCP connection in a separate thread for each HTTP request/response pair.

### **PART-4: Extensions to Simple Proxy Server or web client/web server**

#### **What's our Extension?**

English to Hindi Web Page Translation and web usage stats at web proxy

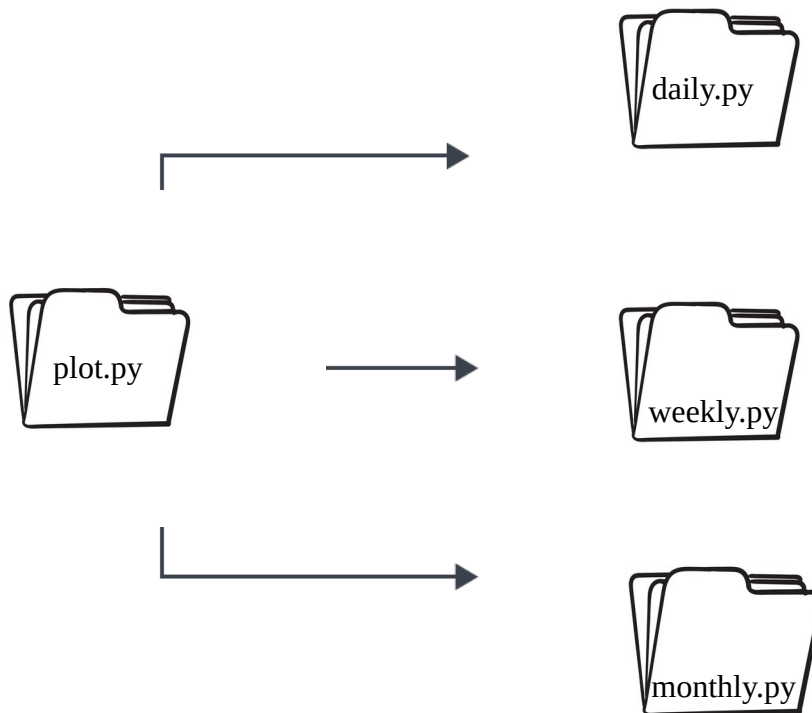
#### **English to Hindi Web Page Translation**



The Extended proxy server is a simple web proxy server with more enhanced feature of translating the web page's content from english to hindi i.e. if any request that goes through this server then the english text in its response must be translated to hindi and then should be passed on to the client. The only difference here is that first we should accumulate the data instead of sending it directly when we have a packet as translating the data from one language to another impacts the content length. So whenever a request arrives similar to the simple proxy server first we modify the request and directs it to the destination. But now we will club all the packets from the destination if the response contains text/html then we must convert the text included in that file to hindi. So we parse all the tag that contain text additionally taking care of tagless text. After translation we again modify the response header to have the length of the modified data. Its an important step otherwise client may receive incomplete content or may indefinitely buffer.

It should be noted that only the content that contains text should be translated and other types of content must be directly forwarded to the client without any translation. Here also client connections are handled concurrently using threads, allowing multiple clients to be served simultaneously.

## Web usage stats at web proxy



The code runs three external Python scripts to generate statistics, combines the resulting images vertically into a single chart, and offers the user the choice to open and view the combined chart in a web browser or exit the program.

### **daily.py plots:**

- **Website with the Highest Count on Each Day (Bar Chart):** This is a bar chart that displays the daily counts for websites. It shows the highest count for each day, and the bars are annotated with the corresponding website names.
- **Daily Count (Line Chart):** The second plot is a line chart showing the daily total counts. It represents the cumulative count of visits for each day.

### **weekly.py plots:**

- **Stacked Bar Chart of Visits by Server IPs:** The first plot is a stacked bar chart that displays weekly statistics of website visits categorized by different server IPs. Each bar represents a week, and the heights of the bars are segmented to show the contribution of each server IP to the total count for that week.
- **Box Plot of Visits per Week:** The second plot is a box plot that visualizes the distribution of visit counts per week. It provides insights into the spread and central tendencies of the weekly data.

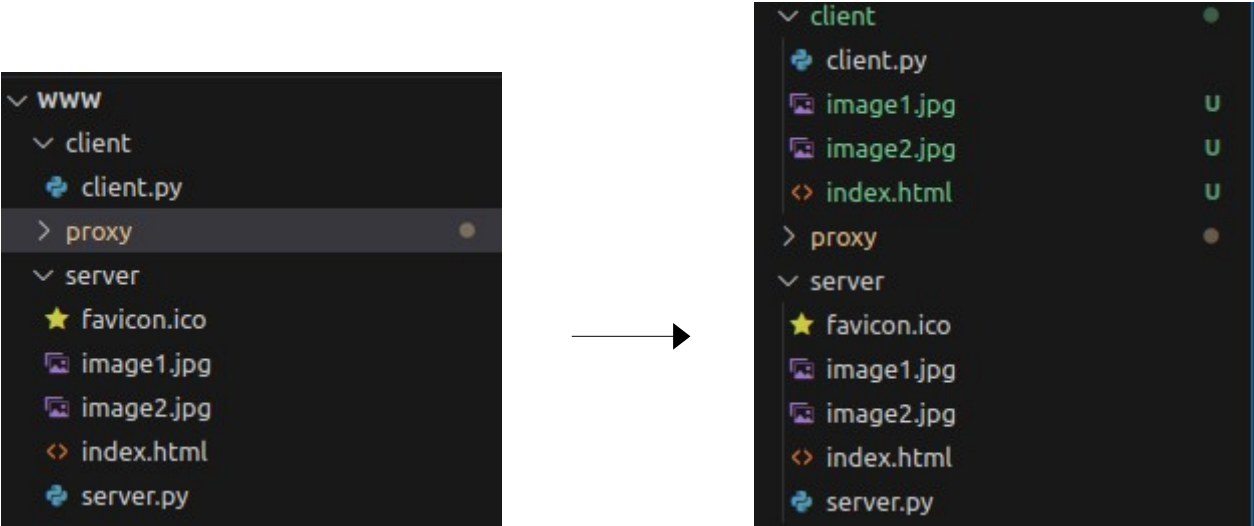
### **monthly.py plots**

- **Pie Chart - Number of Users per Month:** This is a pie chart that illustrates the distribution of the number of unique users visiting a website for each month. Each slice of the pie represents a month, and the percentage of users for each month is labeled on the chart. The

pie chart provides a visual summary of the distribution of users across different months. In summary, the code generates a pie chart that presents the distribution of unique users per month, highlighting the proportion of users for each month in the dataset.

Some Possible Client – Server Interaction

(1) Local client communicates with local web server directly



```
d client
yug@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/Desktop/ACN/WWW/client$ python3 client.py
Enter Server IP:127.0.0.1
Enter Server Port:6789
Enter Proxy IP (else hit enter for direct communication):
Enter Proxy Port (else hit enter for direct communication):
Enter Path:index.html
files downloaded.
Do you want to open the file in browser(Y/n):y

yug@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/Desktop/ACN/WWW/server$ python3 server.py
Server listening on port 6789
Accepted connection from ('127.0.0.1', 57198)
Accepted connection from ('127.0.0.1', 57212)
Accepted connection from ('127.0.0.1', 43938)
```

hello there



**(2) Local client communicates with local web server via simple proxy**

```
yug@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/Desktop$
p/ACN/WWW/clients python3 client.py
Enter Server IP:127.0.0.1
Enter Server Port:6789
Enter Proxy IP (else hit enter for direct communication
):127.0.0.1
Enter Proxy Port (else hit enter for direct communication):12090
Enter Path:index.html
files downloaded
Do you want to open the file in browser(Y/n):
```

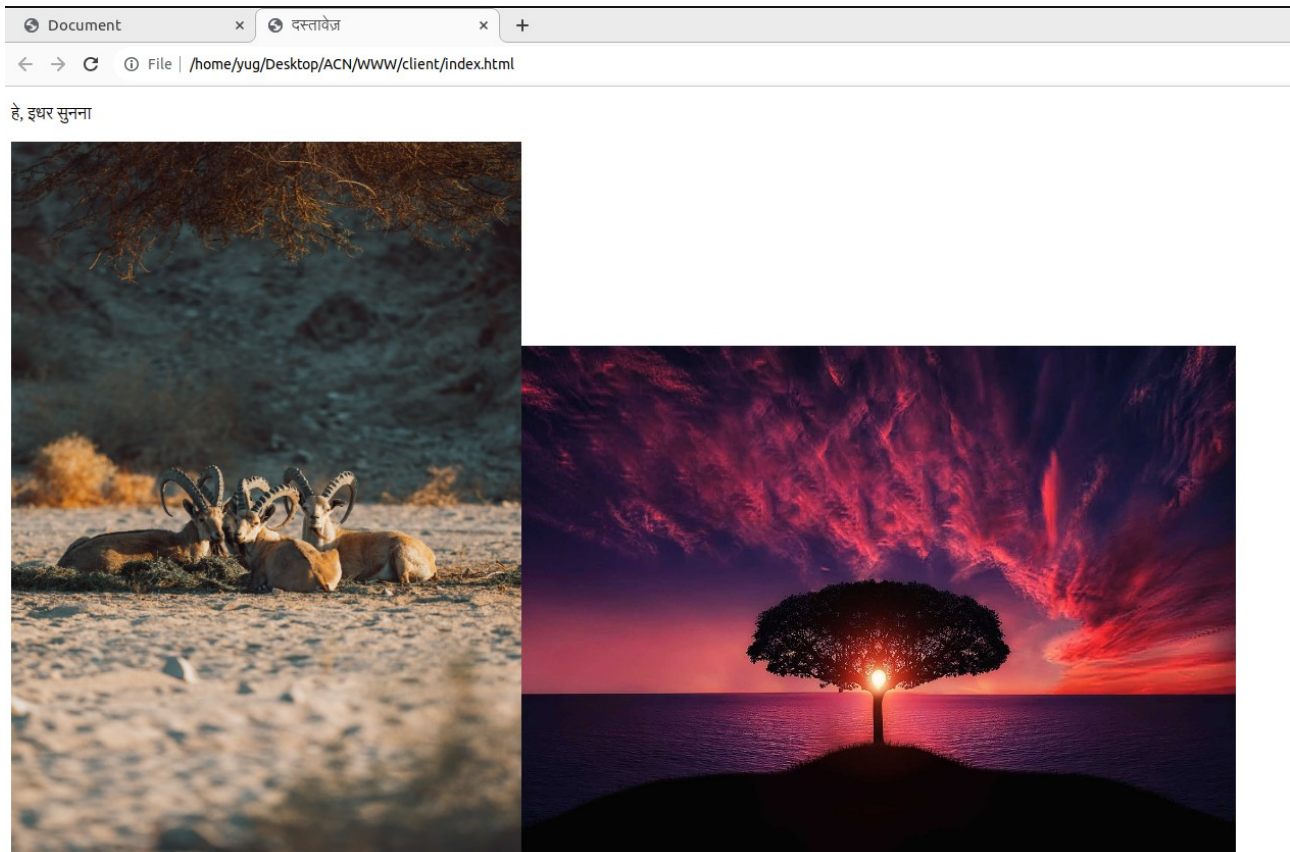
```
yug@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/Desktop
# ACN/www/client$ python3 client.py
Enter Server IP:127.0.0.1
Enter Server Port:6789
Enter Proxy IP (else hit enter for direct communication):127.0.0.1
Enter Proxy Port (else hit enter for direct communication):12000
Enter Path:index.html
files downloaded.
Do you want to open the file in browser(Y/n):n
Do you want to print the recieved raw content(Y/n):y
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Document</title>
</head>
<body>
<p>hello there</p>


</body>
</html>
```

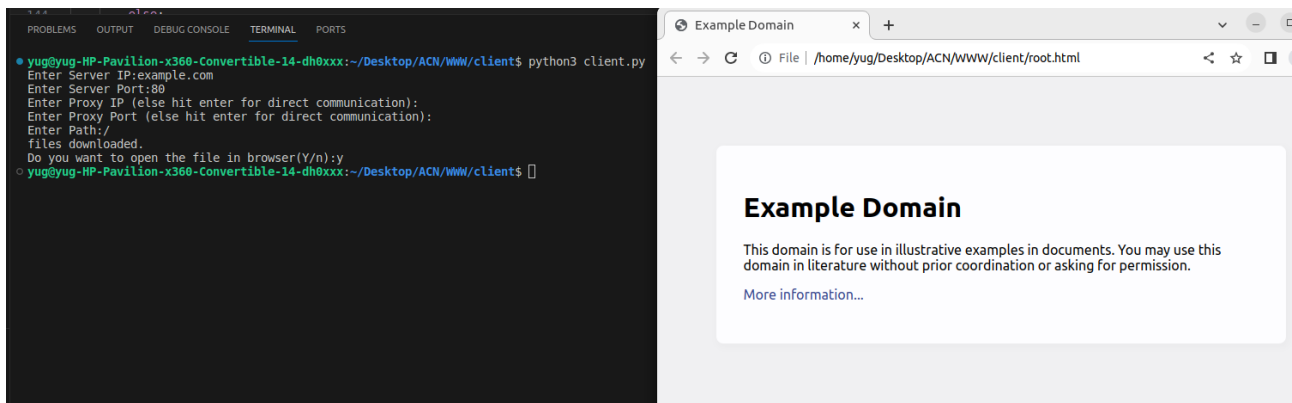
**(3) Local client communicates with local web server via extended proxy**

```
yug@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/Desktop
p/ACN/www/clients$ python3 client.py
Enter Server IP:127.0.0.1
Enter Server Port:6789
Enter Proxy IP (else hit enter for direct communication
):127.0.0.1
Enter Proxy Port (else hit enter for direct communication):13000
Enter Path:index.html
files downloaded.
Do you want to open the file in browser(Y/N):
```

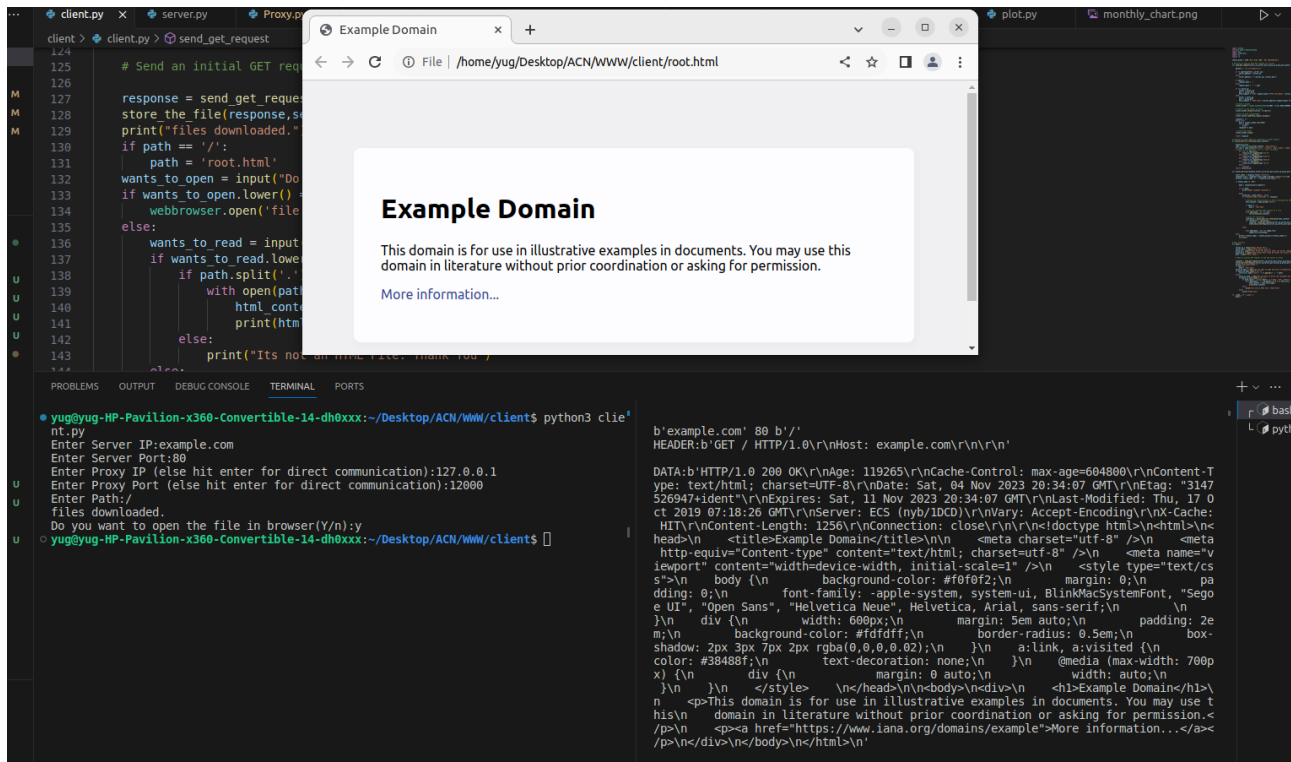




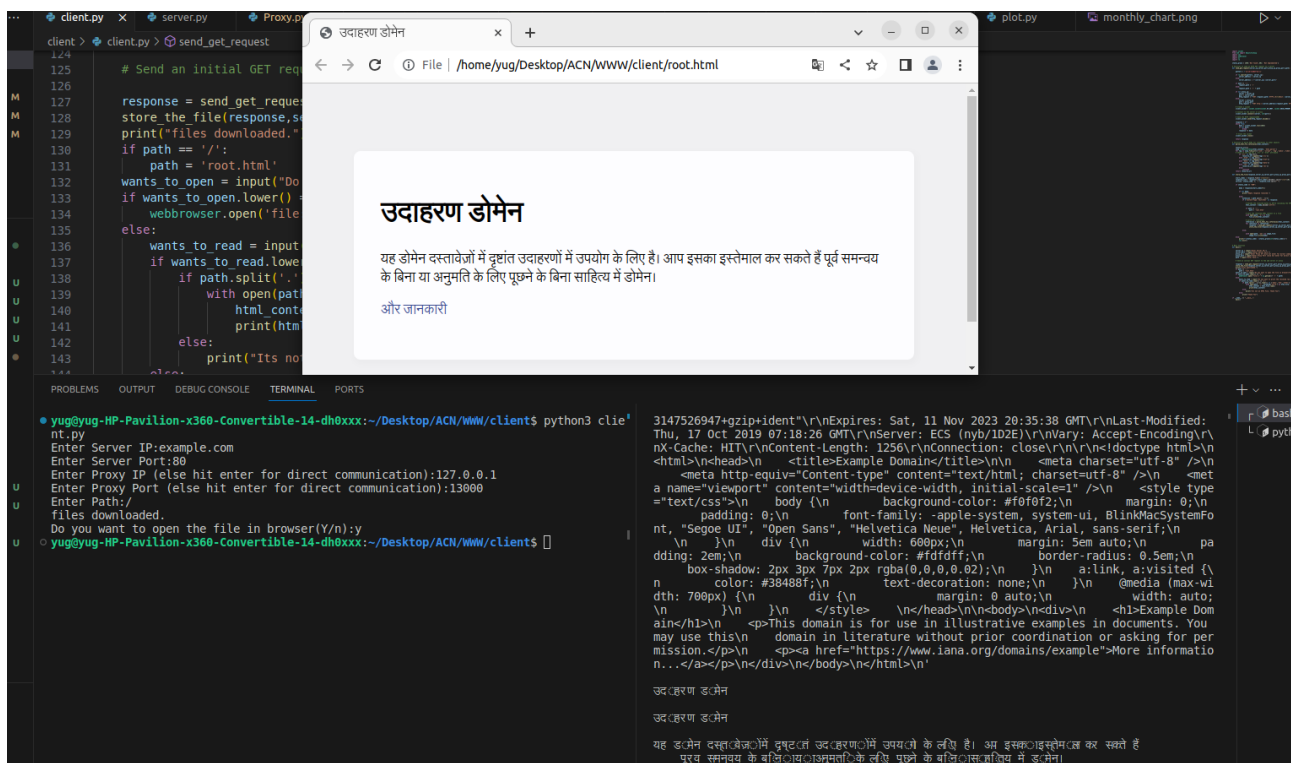
#### (4) Local client communicates with remote host (example.com) directly



## (5) Local client communicates with remote host (example.com) via simple proxy



## (6) Local client communicates with remote host (example.com) via Extended proxy



Activities Wireshark Nov 5 02:08 \*any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No.	Time	Time Stamp	Source	Destination	Protocol	Length	Info
64	3.154368127	20:37:40.850202816	127.0.0.1	127.0.0.1	HTTP	123	GET http://example.com/ HTTP/1.0
71	3.389504086	20:37:41.005330375	172.19.125.152	93.184.216.34	HTTP	105	GET / HTTP/1.0
73	3.623679025	20:37:41.319513714	93.184.216.34	172.19.125.152	HTTP	1678	HTTP/1.0 200 OK (text/html)
90	3.847868882	20:37:41.542902771	172.19.125.152	52.58.135.28	HTTP	359	GET /get?q=Example+Domain&langpair=en%7Chi HTTP/1.1
92	4.001788524	20:37:41.697623213	52.58.135.28	172.19.125.152	HTTP	452	HTTP/1.1 301 Moved Permanently (text/html)
132	5.106795927	20:37:42.802630616	172.19.125.152	18.185.183.197	HTTP	359	GET /get?q=Example+Domain&langpair=en%7Chi HTTP/1.1
139	5.249794089	20:37:42.945028778	18.185.183.197	172.19.125.152	HTTP	452	HTTP/1.1 301 Moved Permanently (text/html)
186	6.544429898	20:37:44.240264579	172.19.125.152	35.156.228.40	HTTP	593	GET /get?q=This+domain+is+for+use+in+illustrative+examples+in+documents.+You+may+use+this%0A++++doma HTTP/1.1
188	6.694816409	20:37:44.389851098	35.156.228.40	172.19.125.152	HTTP	596	HTTP/1.1 301 Moved Permanently (text/html)
228	7.593358176	20:37:45.289192865	172.19.125.152	52.58.135.28	HTTP	364	GET /get?q=More+information...&langpair=en%7Chi HTTP/1.1
230	7.747760466	20:37:45.443595155	52.58.135.28	172.19.125.152	HTTP	457	HTTP/1.1 301 Moved Permanently (text/html)
257	8.669662837	20:37:46.365696726	127.0.0.1	127.0.0.1	HTTP	1958	HTTP/1.0 200 OK (text/html)

Frame 71: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface any, id 0

- Linux cooked capture v1
- Internet Protocol Version 4, Src: 172.19.125.152, Dst: 93.184.216.34
- Transmission Control Protocol, Src Port: 53434, Dst Port: 80, Seq: 1, Ack: 1, Len: 37
- Hypertext Transfer Protocol
  - GET / HTTP/1.0\r\n
  - Host: example.com\r\n
  - \r\n
  - [Full request URI: http://example.com/]
  - [HTTP request 1/1]
  - [Response in frame: 73]

0000 00 04 00 01 00 06 5c 3a 45 11 bd 3b 00 00 08 00 .....: E-:....  
 0010 45 00 00 59 f3 3a 40 00 40 06 e7 dd ac 13 7d 98 E-Y:0 @-....  
 0020 5d 0b d0 22 d0 ba 00 50 93 d9 71 9e 4c 5d 8f 5d ].-.-P-q[L]  
 0030 80 18 01 f6 a7 34 00 00 01 01 08 0a 53 67 37 c3 .....4...Sg7  
 0040 77 83 00 4f 47 45 54 20 2f 20 48 54 54 50 2f 31 wOGET / HTTP/1  
 0050 2e 30 0d 0a 48 6f 73 74 3a 20 65 78 61 6d 70 6c .0-Host: exampl  
 0060 65 2e 63 6f 6d 6d 6a 6d 8a

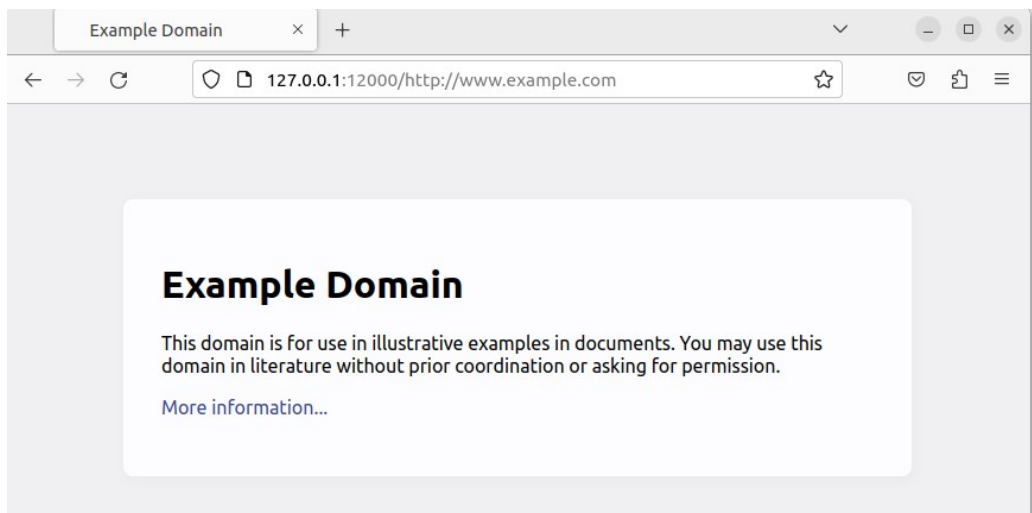
Hypertext Transfer Protocol: Protocol Packets: 271 · Displayed: 12 (4.4%) · Dropped: 0 (0.0%) Profile: Default

(7) Browser communicates with local server via simple proxy (explicit)





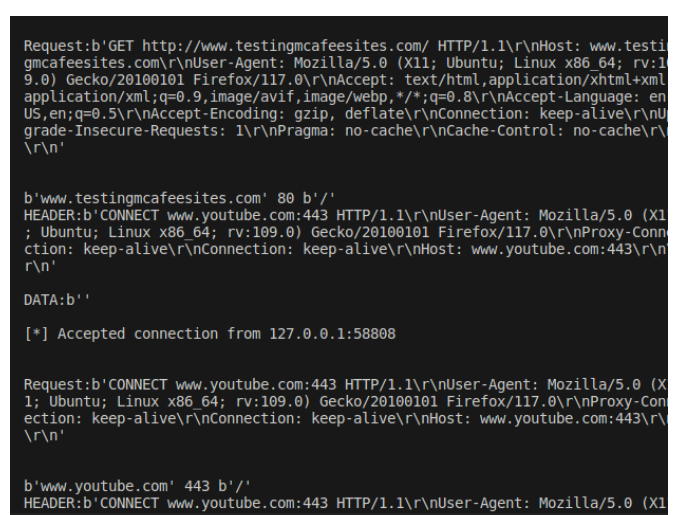
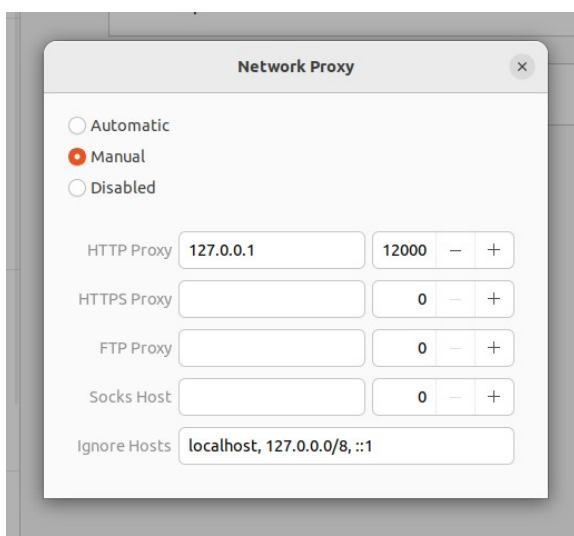
**(8) Browser communicates with remote host/server [example.com] via proxy (explicit)**





**(9) Browser communicates with remote host/server [example.com] via extended proxy (explicit)**



**(10) Browser communicates with remote host/server [example.com] via proxy (implicit)**



URL for testing. × +

← → ↻   www.testingmcafeesites.com

<http://www.testingmcafeesites.com/index.html>  
This is an index url which gives an overview of the different test urls available.

[http://www.testingmcafeesites.com/testcat\\_ac.html](http://www.testingmcafeesites.com/testcat_ac.html)  
-This is an example URL which should be categorized as an art/culture website with a minimal risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_al.html](http://www.testingmcafeesites.com/testcat_al.html)  
-This is an example URL which should be categorized as an alcohol website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_an.html](http://www.testingmcafeesites.com/testcat_an.html)  
-This is an example URL which should be categorized as an anonymizer website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_au.html](http://www.testingmcafeesites.com/testcat_au.html)  
-This is an example URL which should be categorized as an Anonymizing Utilities website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_be.html](http://www.testingmcafeesites.com/testcat_be.html)  
-This is an example URL which should be categorized as Browser Exploits website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_bl.html](http://www.testingmcafeesites.com/testcat_bl.html)  
-This is an example URL which should be categorized as a Blogs/Wiki website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_bu.html](http://www.testingmcafeesites.com/testcat_bu.html)  
-This is an example URL which should be categorized as a Business website with a low risk reputation score.


[http://www.testingmcafeesites.com/testcat\\_ch.html](http://www.testingmcafeesites.com/testcat_ch.html)  
-This is an example URL which should be categorized as a Chat website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_cm.html](http://www.testingmcafeesites.com/testcat_cm.html)  
-This is an example URL which should be categorized as a Public Information website with a low risk reputation score.

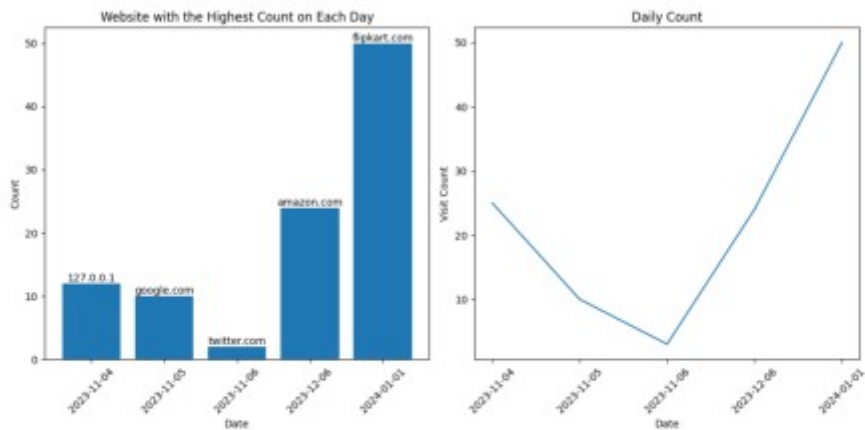
[http://www.testingmcafeesites.com/testcat\\_co.html](http://www.testingmcafeesites.com/testcat_co.html)  
-This is an example URL which should be categorized as a Controversial Opinions website with a low risk reputation score.

[http://www.testingmcafeesites.com/testcat\\_cn.html](http://www.testingmcafeesites.com/testcat_cn.html)

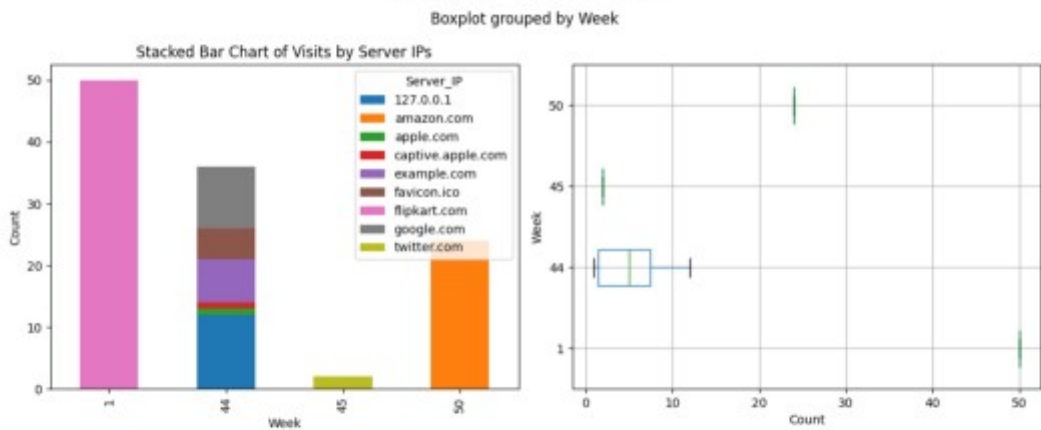
## (11) Web usage statistics

```
>  web_usage_data.csv
Client_IP,Server_IP,Date,Month,Week,Count
127.0.0.1,example.com,2023-11-04,11,44,5
127.0.0.2,example.com,2023-11-04,11,44,2
127.0.0.1,127.0.0.1,2023-11-04,11,44,12
127.0.0.1,google.com,2023-11-05,11,44,10
127.0.0.1,apple.com,2023-11-06,11,44,1
127.0.0.1,twitter.com,2023-11-06,11,45,2
127.0.0.1,amazon.com,2023-12-06,12,50,24
127.0.0.1,flipkart.com,2024-01-01,1,1,50
127.0.0.1,favicon.ico,2023-11-04,11,44,5
127.0.0.1,captive.apple.com,2023-11-04,11,44,1
```

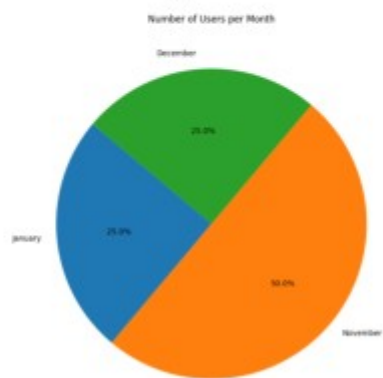
Daily Statistics



Weekly Statistics



Monthly Statistics



## **ANTI-PLAGIARISM Statement**

I certify that this assignment/report is my own work, based on my personal study and/or research and that I 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. I 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 I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. Additionally, I acknowledge that I may have used AI tools, such as language models (e.g., ChatGPT, Bard), for assistance in generating and refining my assignment, and I 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, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: Yug Patel, CS23MTECH14019  
Somya jain, CS23MTECH12011  
Anil Kumar, CS23MTECH13001

Date:05/11/2023

Signature: Yug Patel, CS23MTECH14019  
Somya jain, CS23MTECH12011  
Anil Kumar, CS23MTECH13001