CS 5060 : ADVANCED COMPUTER NETWORKS

PROGRAMMING ASSIGNMENT # 2

Submitted By : Group 2

Anirudh Joshi cs23mtech11002

Apurba Saha cs23mtech11003

Kaustubh Gaikwad sm23mtech14003

Course Instructor : Prof. Bheemarjuna Reddy Tamma

MTech Sem-I, July-Dec 2023

Computer Science & Engineering, Department

Indian Institute of Technology, Hyderabad

3 november 2023

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 plagiarised 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.

Anirudh Joshi <cs23mtech11002>

Apurba Saha <cs23mtech11003>

Kaustubh Gaikwad <sm23mtech14003>

Date: 03.11.23

Signature: A.J

:A.P

:K.G

TASK-1

SIMPLE WEB CLIENT

- **Functionality** : our web client is able to connect to any website like cse.iith.ac.in or our own web server made in task-3 . It opens non persistent tcp connection to the website then it makes http get request to get the base html file then it uses the beautiful SOAP library of python to parse and retrieve the references objects. The referenced objects to be parsed are images ,icons and scripts.it also makes use of the PIL library of python to display the images as they are downloaded. Furthermore we also save the received object data like images,scripts,icons in separate files on our PC.

- **How to give input arguments to client program :**
  For client programs we have two choices connecting directly to the server or connecting via proxy both have different ways of giving input arguments. To connect directly to the server open the terminal where Client.py is stored and enter "python3 Client.py hostname port_no path" and press enter and then select choice 0 when asked. For example, to connect to cse.iith.ac.in directly : " python3 Client.py cse.iith.ac.in 443 / " Enter.
  : enter 0 (choice) and Enter
  Before connecting Client via proxy make sure proxy server is running
  To connect to cse.iith.ac.in via Proxy : " python3 Client.py " Enter
  : Enter choice 1 and press enter
  : Enter Proxy IP = "127.0.0.1" Enter
  : Enter port on which proxy is listening = "15000" Enter
  : Enter server hostname or IP = " cse.iith.ac.in " Enter
  : Enter port no of server = " 443 " Enter
  : Enter the file path = " / " Enter

- **Flow path of client program :**
  If the user chooses 0 in step 2 then the client program directly connects to the web server, not implicating a proxy in between. And then it checks if the server port is 443 or not. If it is 443 then our client program connects using a ssl socket otherwise it connects to a web server using normal socket. However if the choice user selects is 1 then our client program connects to the web server via our own webproxy made as part of task 2. And to do so it connects to the proxy using a normal socket, and then the proxy then connects to the web server and returns the response back to the client.

- **Outputs of client program:**
  Our client program first displays the html base file and also saves all the image,icon,scripts and finally we also output the latency ,so that we can compare the speedy web client program done as part of our extension.

TASK-2

SIMPLE WEB PROXY

- **Functionality:** our web proxy perform functionality of web server and also some functionality of our web client basically it receive the http get requests from the client and forward to the web server then it receive the response from the web server then it checks the status code of http response message and than it create http response header based on the status code ,appends the http response data to the headers and sends it to the client. Our web proxy can be connected to any web browser(client) and also it can connect to any web server.

- **How to give input arguments to proxy program :**
  The server host name and the server port to which proxy needs to connect to are extracted from the http get request received from the client itself so there is no explicit argument we need to pass to proxy. Just make sure to bind the proxy to correct ip as the ip changes depending upon the network to which we are connected to. And this ip can be found using the ifconfig command in linux.

- **Flow path of proxy program :**
  In proxy there is always one socket open to serve incoming tcp connection requests, it makes a separate new thread for every new client . It receives the http get request from the client and extracts server host name and server port number from it. If the port number is 443 then it connects using an ssl socket to the server otherwise using a normal socket. Then it waits for the response from the server, checks the status code of the response and sends back the appropriate http response to the client.

  Outputs of proxy program:

Because we were not suppose to do catching, we are not saving any object as files on the proxy side.output of the proxy is simply the response sent back to the client.

TASK-3

SIMPLE WEB SERVER

- **Functionality:** our server is always in listening state to the incoming client requests, making a separate thread for each client. It extracts the file path from the http get request, searches the file path in the directory and returns back the file if present otherwise it returns 404 not found http response message. Also our server can be connected using any web browser or our own web client program.

- **How to give input arguments to server program :**
  There are no input arguments to the server program.

- **Flow path of server program :**
  Once the server receives the tcp connection request it opens a new thread for that client, upon receiving the http get request it extracts the file path and then reads the file data. It makes the appropriate http response header, appends file data to the header and sends back the response client.

- **Outputs of server program:**
  Output of the server program is simply a requested file by the client, sent back as response.

TASK-4

SPEEDY WEB CLIENT (EXTENSION 5 PART 4)

- **Functionality:** it has all the functionality which our simple web client had to offer ,additionally in speedy web client to get the referenced objects from the server we make multiple parallel tcp connections using ThreadPoolExecutor() functionality of concurrent.futures library of python. Our speedy web client can also connect to any web server.

- **How to give input arguments to speedy web client program :**
  For speedy web client programs we have two choices: connecting directly to the server or connecting via proxy both have different ways of giving input arguments. To connect directly to the server open the terminal where Client.py is stored and enter "python3 SpeedyWebClient .py hostname port_no path" and press enter and then select choice 0 when asked. For example, to connect to cse.iith.ac.in directly : " python3 SpeedyWebClient.py cse.iith.ac.in 443 / " Enter.

  : enter 0 (choice) and Enter

  Before connecting Client via proxy make sure proxy server is running
  To connect to cse.iith.ac.in via Proxy : " python3 SpeedyWebClient.py " Enter

  : Enter choice 1 and press enter

  : Enter Proxy IP = "127.0.0.1" Enter

  : Enter port on which proxy is listening = "15000" Enter

  : Enter server hostname or IP = " cse.iith.ac.in " Enter

  : Enter port no of server = " 443 " Enter

  : Enter the file path = " / " Enter

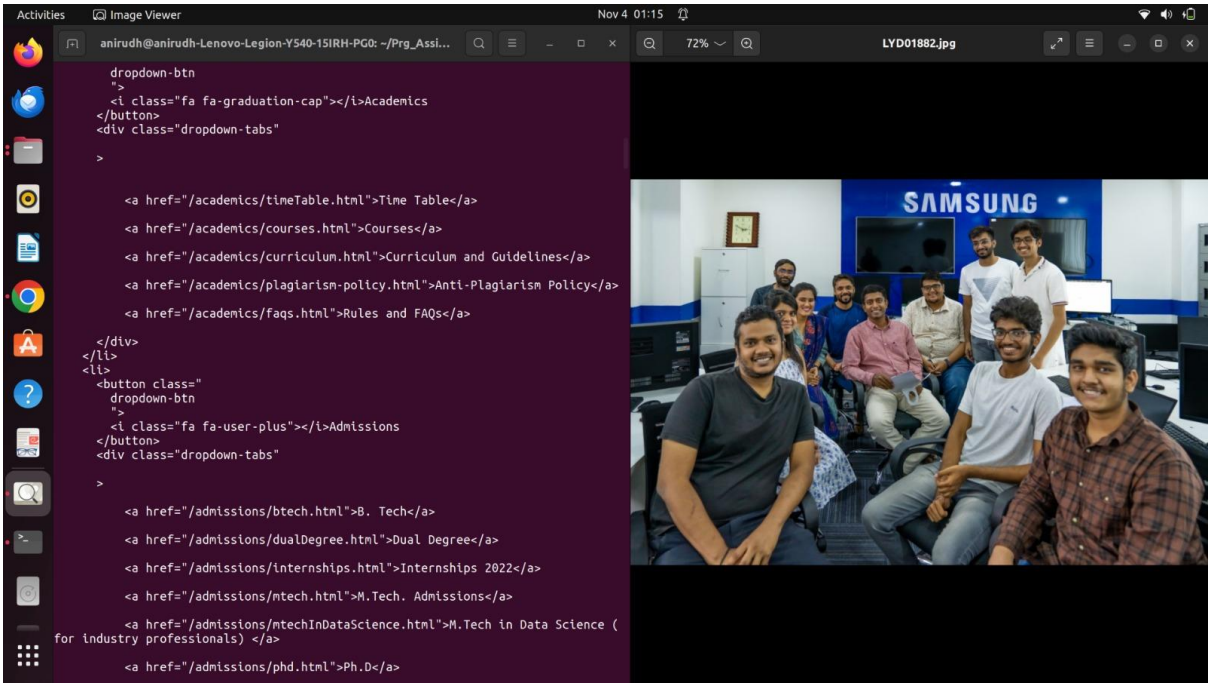
- **Flow path of speedy web client program :**
  If the user chooses 0 in step 2 then the speedy web client program directly connects to the web server, not implicating a proxy in between. And then it checks if the server port is 443 or not. If it is 443 then our client program connects using a ssl socket otherwise it connects to a web server using normal socket. However if the choice user selects is 1 then our speedy web client program connects to the web server via our own webproxy made as part of task 2. And to do so it connects to the proxy using a normal socket, and then the proxy then connects to the web server and returns the response back to the speedy web client. Once the speedy web client receives the base html file it parses html file for icons, scripts and images and stores all their respective urls in a list. This list is given as the one of the input arguments of the executor object of ThreadPoolExecutor(). The target function of the ThreadPoolExecutor() function is the getObject() function which takes server hostname, server port no and complete urls as arguments and sends the http get request to server. Because the ThreadPoolExecutor() makes multiple instances of getObject() function and in each instance we make a new tcp connection so multiple parallel connections are made in this way. We can also fix the max number of parallel tcp connections.

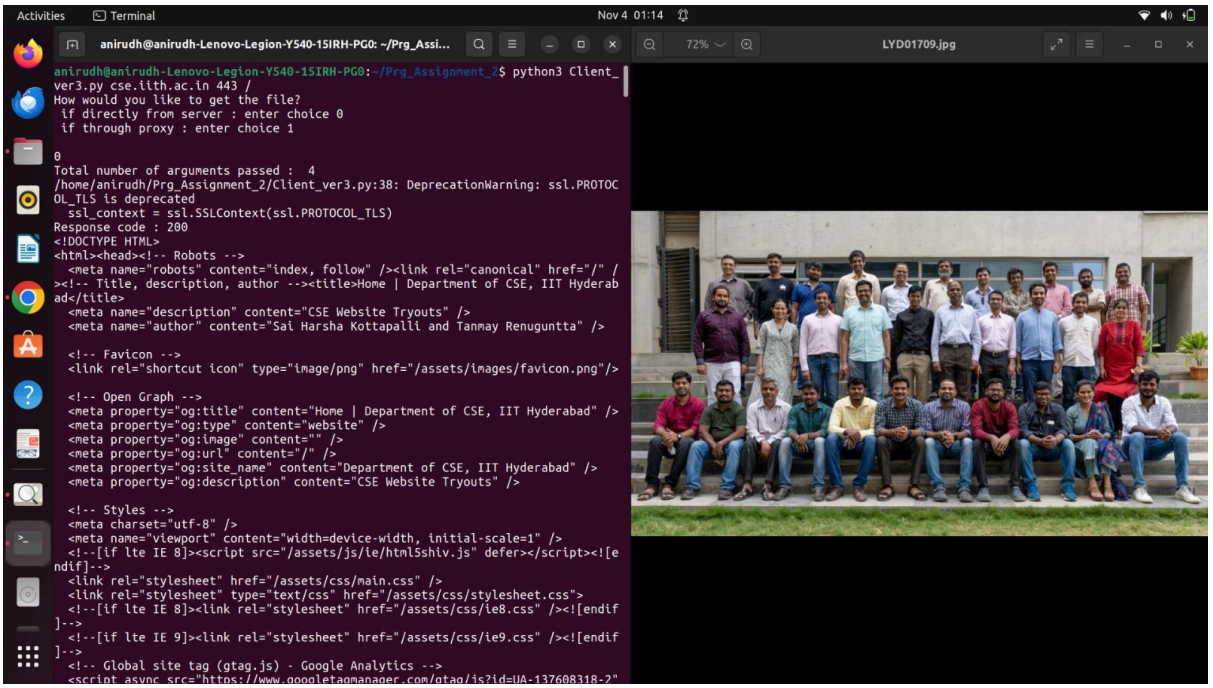
- **Outputs of speedy web client program:**
  Output of the speedy web client program is that it saves all the retrieved objects from the server, and it also outputs the latency to retrieve those objects.

{demonstrations from the next page}

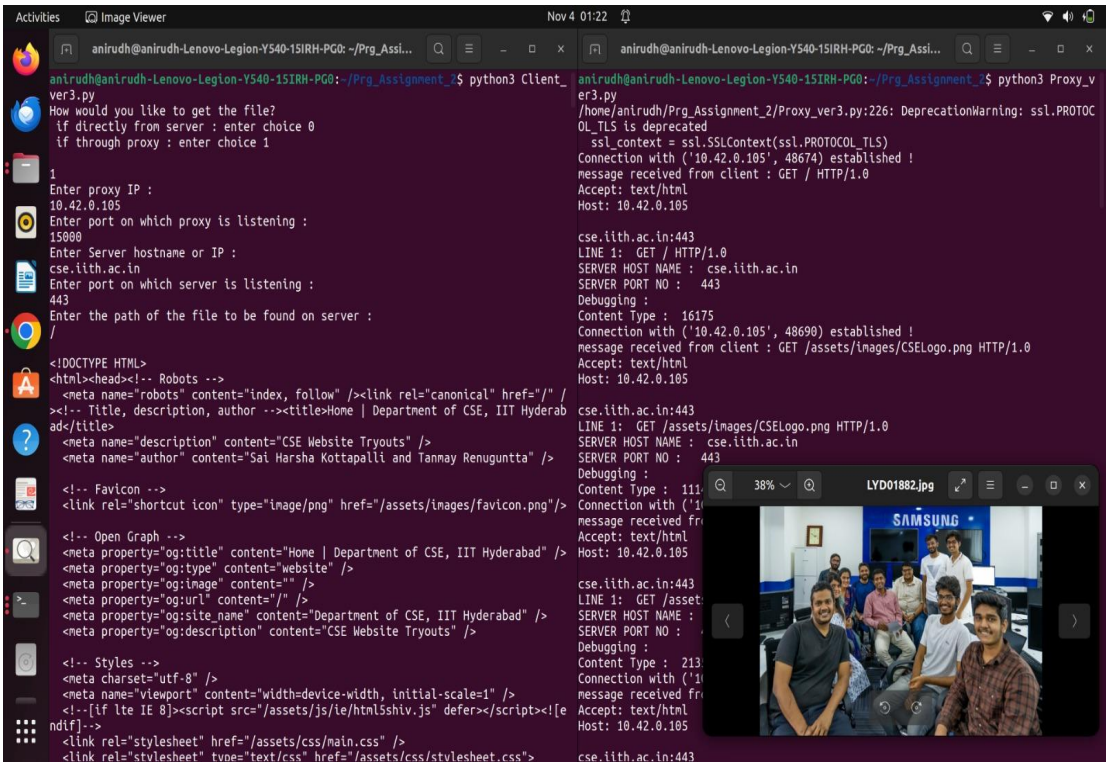
Demonstrations:

CLIENT<->WEB SERVER (CONNECTING TO CSE.IITH):

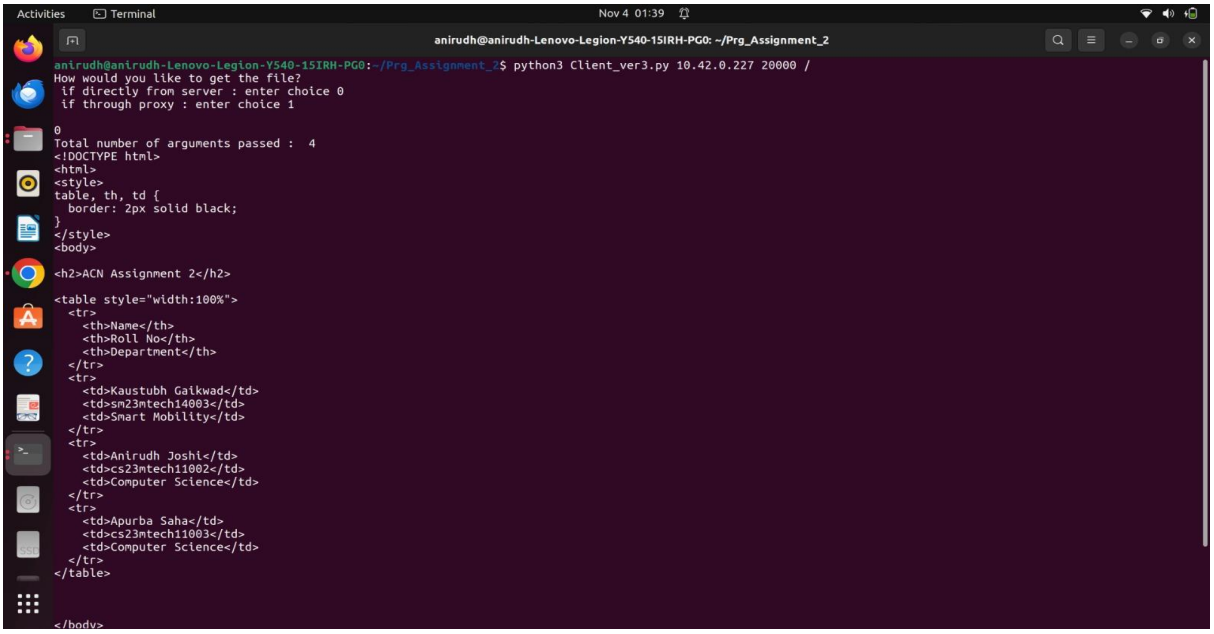




CLIENT<->PROXY<->WEB SERVER (CONNECTING TO CSE.IITH):



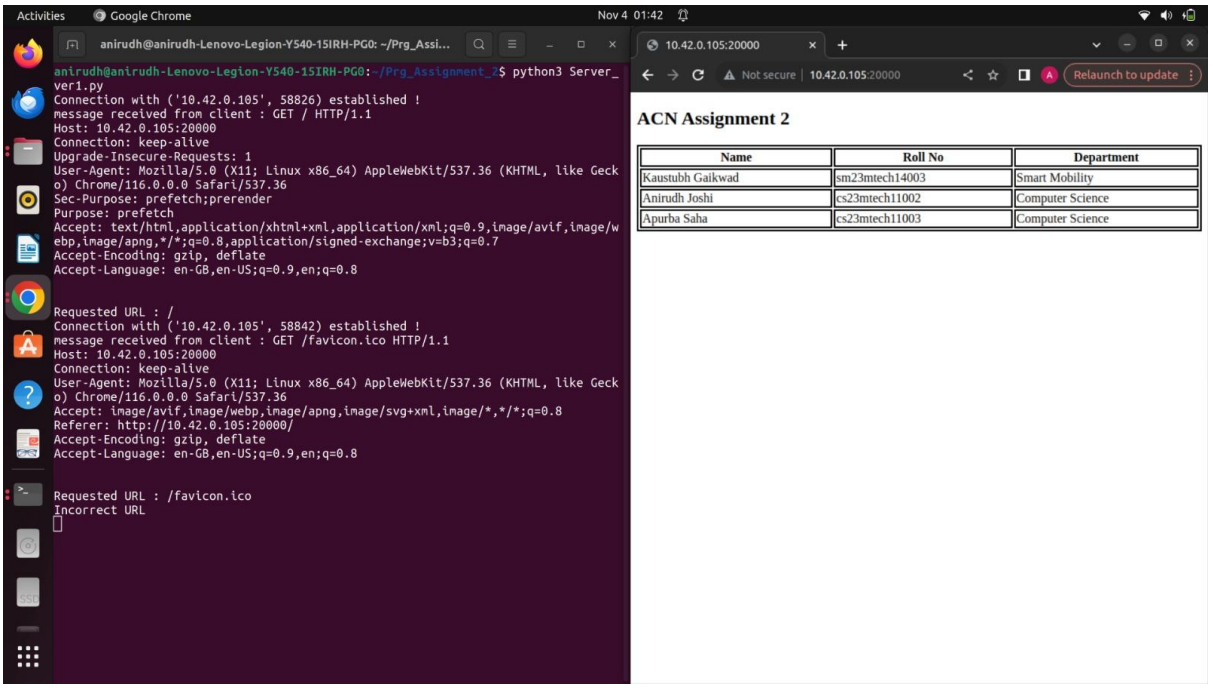
CLIENT<->OUR OWN WEB SERVER (ON DIFFERENT MACHINE):



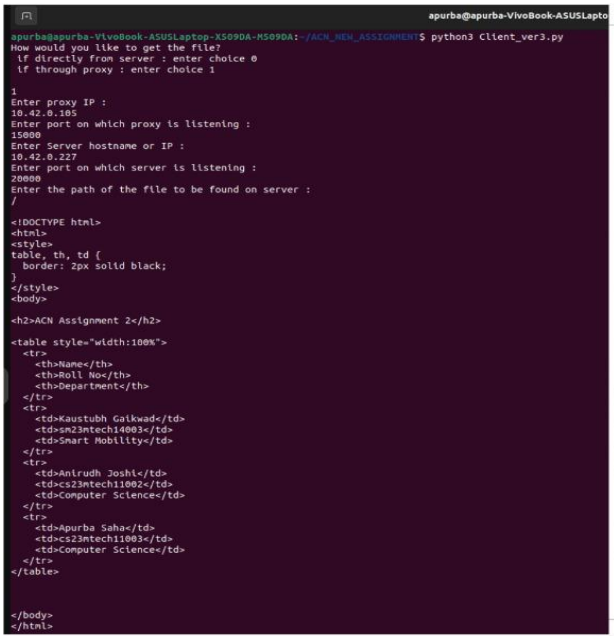
C:\WINDOWS\py.exe

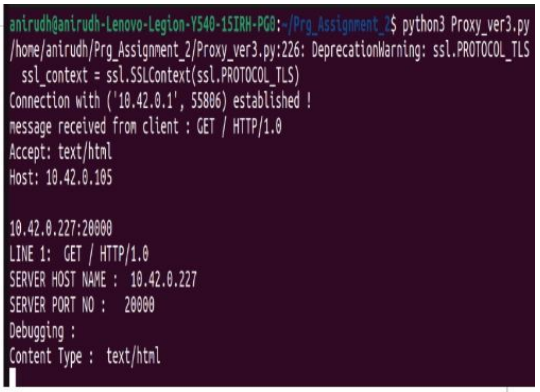
```
Connection with ('10.42.0.105', 43052) established !
message received from client : GET / HTTP/1.0
Accept: text/html
Host: 10.42.0.227


Requested URL : /
```

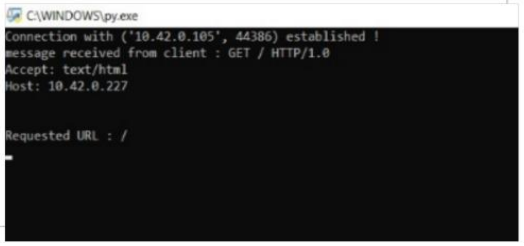
BROWSER TO OUR OWN SERVER (ON DIFFERENT MACHINE):



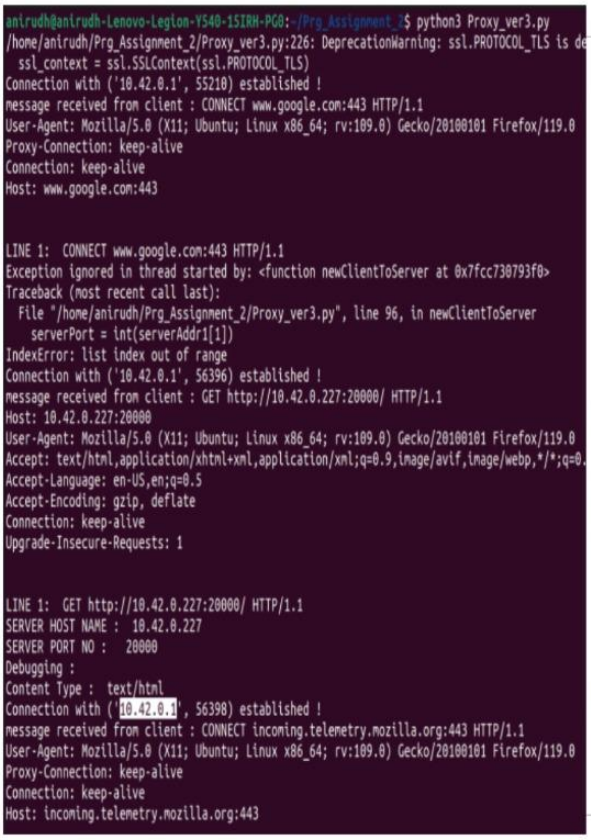
CLIENT PROXY OUR OWN WEB SERVER (ON DIFFERENT MACHINE):

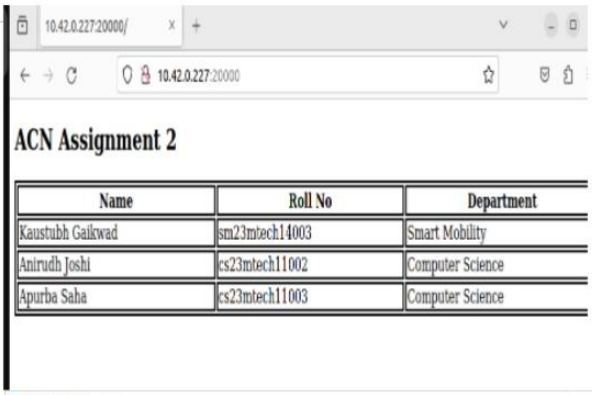


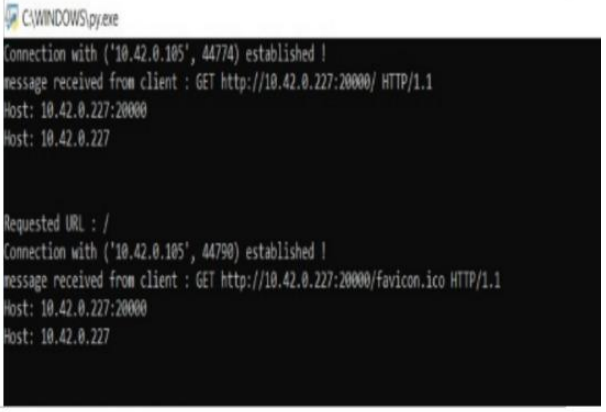




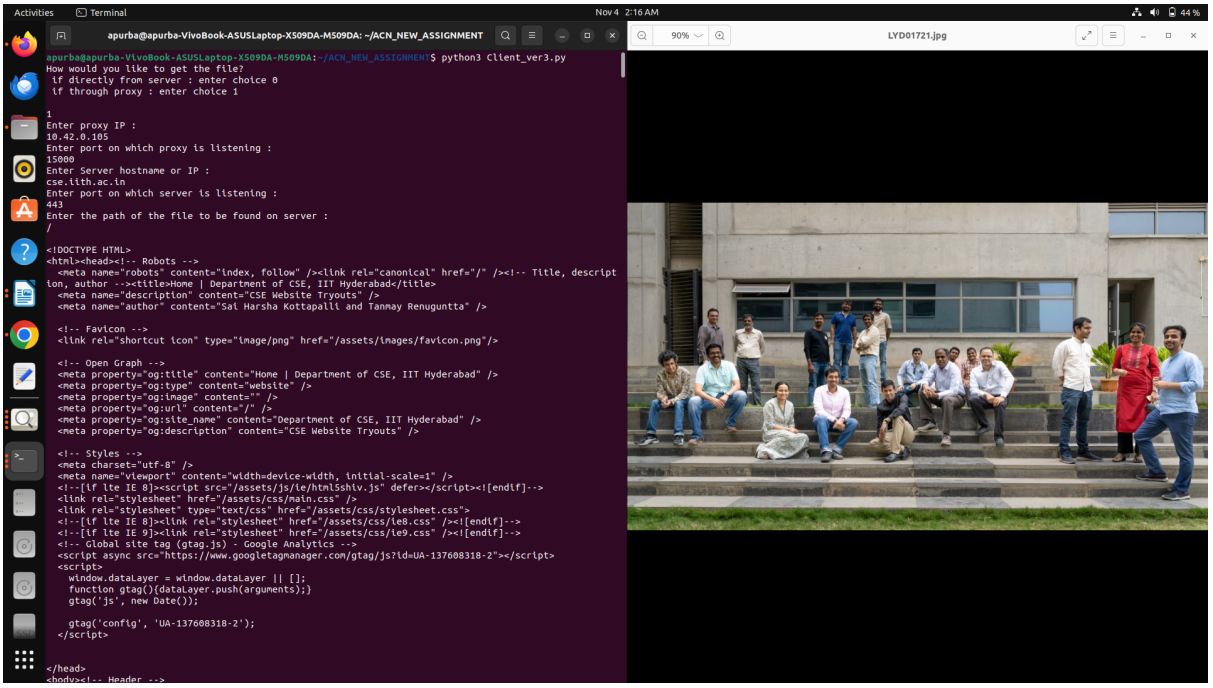
BROWSER<-> PROXY<-> OUR OWN WEB SERVER (ON DIFFERENT MACHINE):





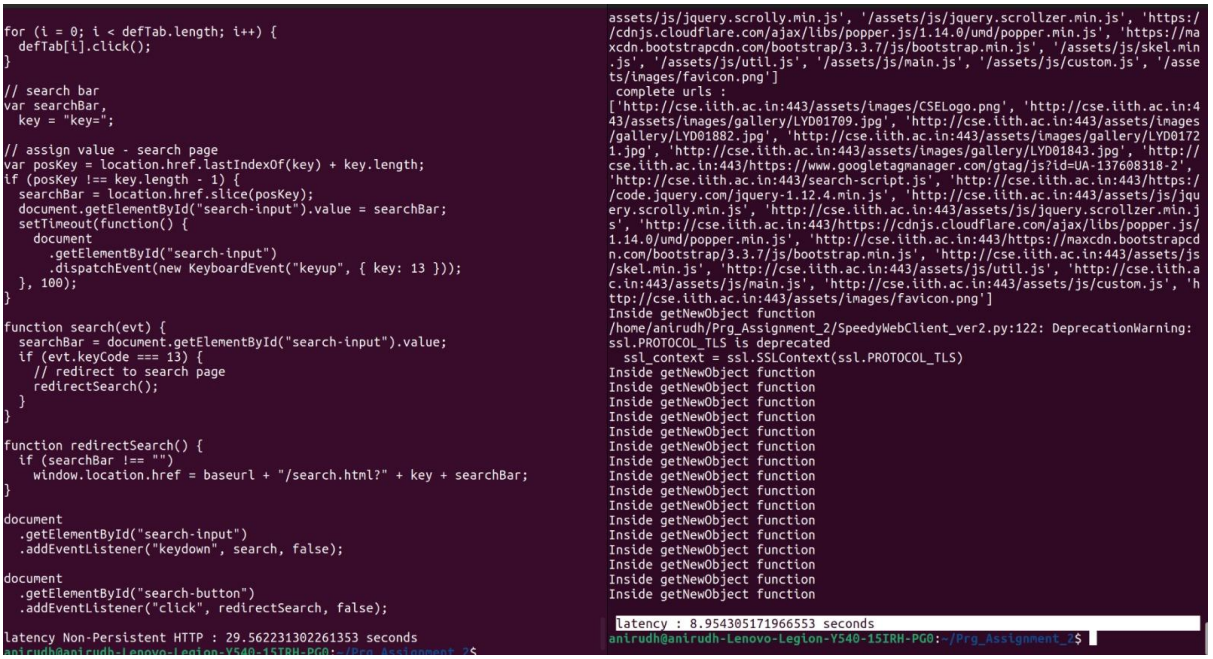


BROWSER<->PROXY<->WEB SERVER (ON DIFFERENT MACHINE):




```
anirudh@anirudh-Lenovo-Legion-Y540-15IRH-PG0:~/Prg_Assignment_2$ python3 Proxy_ver3.py
/home/anirudh/Prg_Assignment_2/Proxy_ver3.py:226: DeprecationWarning: ssl.PROTOCOL_TLS is deprecat
  ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS)
Connection with ('10.42.0.1', 36226) established !
message received from client : GET / HTTP/1.0
Accept: text/html
Host: 10.42.0.105

cse.iith.ac.in:443
LINE 1:  GET / HTTP/1.0
SERVER HOST NAME :  cse.iith.ac.in
SERVER PORT NO :  443
Debugging :
Content Type :  16175
Connection with ('10.42.0.1', 36234) established !
message received from client : GET /assets/images/CSELogo.png HTTP/1.0
Accept: text/html
Host: 10.42.0.105

cse.iith.ac.in:443
LINE 1:  GET /assets/images/CSELogo.png HTTP/1.0
SERVER HOST NAME :  cse.iith.ac.in
SERVER PORT NO :  443
Debugging :
Content Type :  11149
Connection with ('10.42.0.1', 36250) established !
message received from client : GET /assets/images/gallery/LYD01709.jpg HTTP/1.0
Accept: text/html
Host: 10.42.0.105

cse.iith.ac.in:443
LINE 1:  GET /assets/images/gallery/LYD01709.jpg HTTP/1.0
SERVER HOST NAME :  cse.iith.ac.in
SERVER PORT NO :  443
Debugging :
Content Type :  213528
Connection with ('10.42.0.1', 36258) established !
message received from client : GET /assets/images/gallery/LYD01882.jpg HTTP/1.0
Accept: text/html
Host: 10.42.0.105

cse.iith.ac.in:443
LINE 1:  GET /assets/images/gallery/LYD01882.jpg HTTP/1.0
SERVER HOST NAME :  cse.iith.ac.in
```

SPEEDY WEB CLIENT VS NORMAL WEB CLIENT (LATENCY COMPARISON):



```
for (i = 0; i < defTab.length; i++) {
  defTab[i].click();
}

// search bar
var searchBar,
  key = "key=";

// assign value - search page
var posKey = location.href.lastIndexOf(key) + key.length;
if (posKey !== key.length - 1) {
  searchBar = location.href.slice(posKey);
  document.getElementById("search-input").value = searchBar;
  setTimeout(function() {
    document
      .getElementById("search-input")
      .dispatchEvent(new KeyboardEvent("keyup", { key: 13 }));
  }, 100);
}

function search(evt) {
  searchBar = document.getElementById("search-input").value;
  if (evt.keyCode === 13) {
    // redirect to search page
    redirectSearch();
  }
}

function redirectSearch() {
  if (searchBar !== "")
    window.location.href = baseurl + "/search.html?" + key + searchBar;
}

document
  .getElementById("search-input")
  .addEventListener("keydown", search, false);

document
  .getElementById("search-button")
  .addEventListener("click", redirectSearch, false);

latency Non-Persistent HTTP : 29.562231302261353 seconds
```

```
assets/js/jquery.scrolly.min.js', '/assets/js/jquery.scrollzer.min.js', 'https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js', 'https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js', '/assets/js/skel.min.js', '/assets/js/util.js', '/assets/js/main.js', '/assets/js/custom.js', '/assets/images/favicon.png']
 complete urls :
['http://cse.iith.ac.in:443/assets/images/CSELogo.png', 'http://cse.iith.ac.in:443/assets/images/gallery/LYD01709.jpg', 'http://cse.iith.ac.in:443/assets/images/gallery/LYD01882.jpg', 'http://cse.iith.ac.in:443/assets/images/gallery/LYD01721.jpg', 'http://cse.iith.ac.in:443/assets/images/gallery/LYD01843.jpg', 'http://cse.iith.ac.in:443/https://www.googletagmanager.com/gtag/js?id=UA-137608318-2', 'http://cse.iith.ac.in:443/search-script.js', 'http://cse.iith.ac.in:443/https://code.jquery.com/jquery-1.12.4.min.js', 'http://cse.iith.ac.in:443/assets/js/jquery.scrolly.min.js', 'http://cse.iith.ac.in:443/assets/js/jquery.scrollzer.min.js', 'http://cse.iith.ac.in:443/https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js', 'http://cse.iith.ac.in:443/https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js', 'http://cse.iith.ac.in:443/assets/js/skel.min.js', 'http://cse.iith.ac.in:443/assets/js/util.js', 'http://cse.iith.ac.in:443/assets/js/main.js', 'http://cse.iith.ac.in:443/assets/js/custom.js', 'http://cse.iith.ac.in:443/assets/images/favicon.png']
Inside getNewObject function
/home/anirudh/Prg_Assignment_2/SpeedyWebClient_ver2.py:122: DeprecationWarning: ssl.PROTOCOL_TLS is deprecated
  ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS)
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function
Inside getNewObject function

latency : 8.954305171966553 seconds
anirudh@anirudh-Lenovo-Legion-Y540-15IRH-PG0:~/Prg_Assignment_2$
```

References:

https://pythonprogramming.net/sockets-tutorial-python-3/.
https://pynative.com/python-timestamp/
https://pythontic.com/modules/socket/introduction.
https://www.beej.us/guide/bgnet/
https://en.wikipedia.org/wiki/Proxy_server

LLM References: