CS302 – Computer Networks Lab

Lab -2 Report

Ragul N S – 191CS146

Rakshith H R – 191CS148

1) Using TCP socket, implement HTTP server and client.

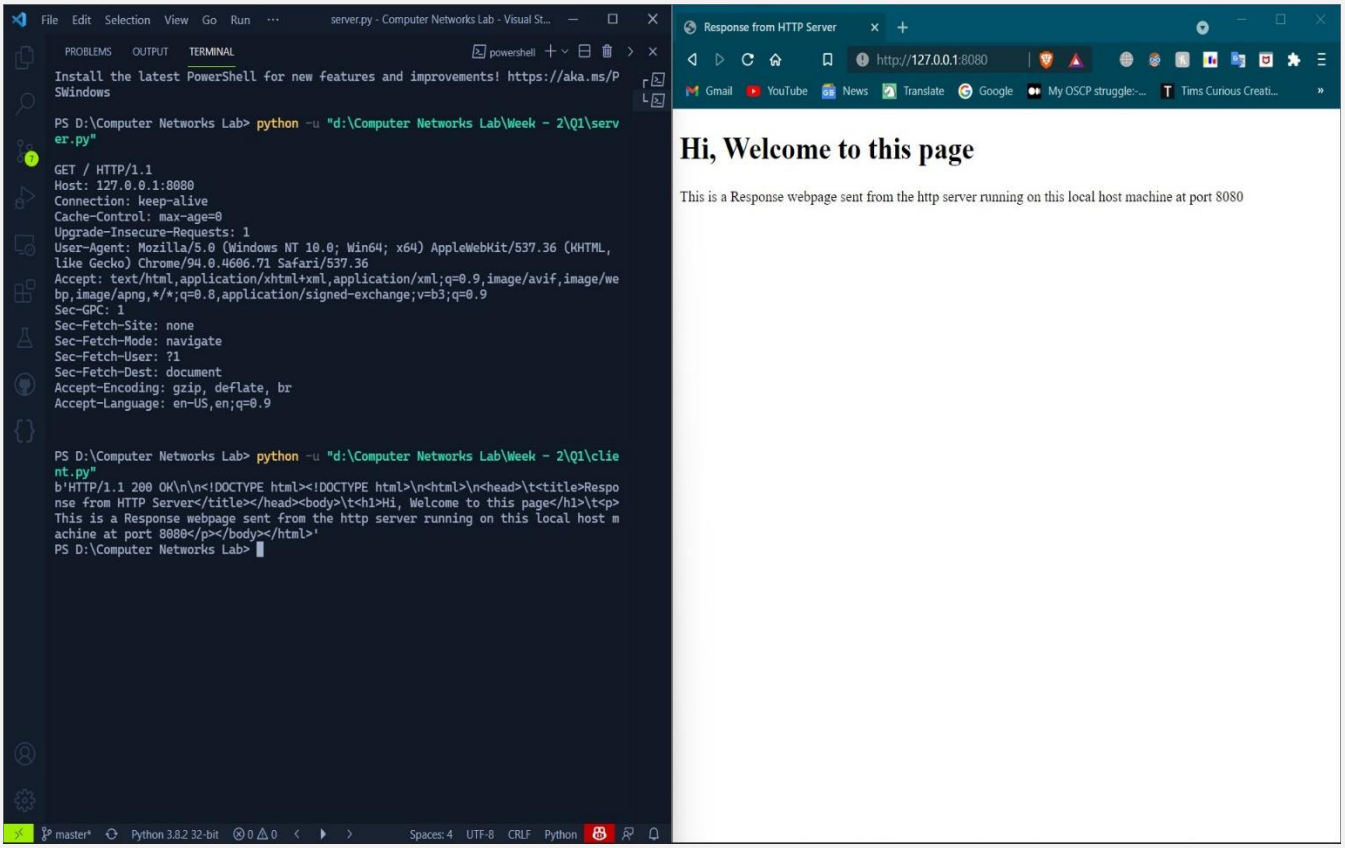
Solution: TCP protocol is used to implement the server and client. The server is waiting for a request to come from the client. Client program sends a request message, which is printed in the terminal where server is running and then the html response page is printed in the client terminal. A request can be sent from a browser also, then the browser will render the html response and show it on the screen

We have attached the server program, client program, screenshot of the output.

```
# A http server implemented using tcp protocol
# This server responds with a basic html webpage when a request is
 made to it

import socket

host = "127.0.0.1"
port = 8080
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSocket.bind((host, port))
serverSocket.listen(5)


def createWebpage():
    webPage = "HTTP/1.1 200 OK\n\n<!DOCTYPE html>"
    webPage += "<!DOCTYPE html>"
    webPage += "\n<html>"
    webPage += "\n<head>"
    webPage += "\t<title>Response from HTTP Server</title>"
    webPage += "</head>"
    webPage += "<body>"
    webPage += "\t<h1>Hi, Welcome to this page</h1>"
    webPage += "\t<p>This is a Response webpage sent from the http
 server running on this local host machine at port 8080</p>"
    webPage += "</body>"
    webPage += "</html>"
    return webPage.encode()
```

```
response = createWebpage()

while True:
    try:
        clientSocket, addr = serverSocket.accept()
        requestRecieved = clientSocket.recv(1024).decode()
        print(requestRecieved)
        clientSocket.send(response)
        clientSocket.close()
    except Exception as error:
        print(error)
```

```
# A client program which sends a http request to the http server v
ia a tcp connection
# The response then recieved is printed in the terminal

import socket

host = "127.0.0.1"
port = 8080
request = f"GET / HTTP/1.1\r\nHost: {host}:{port}\r\n\r\n".encode(
)
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.connect((host, port))
clientSocket.send(request)
response = clientSocket.recv(1024)
print(response)
```

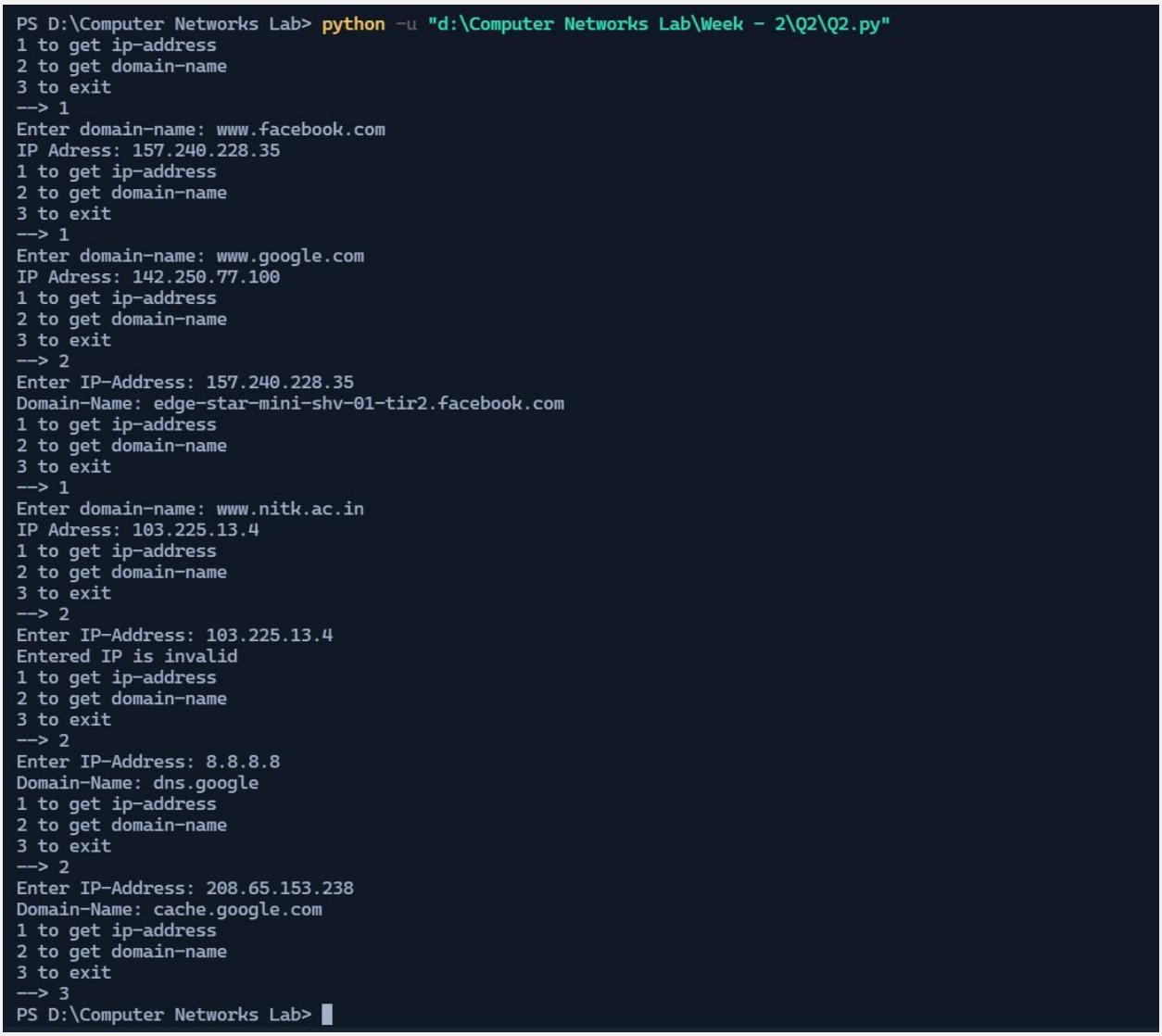


2) Write a program to translate a Domain name or hostname to its IP address and vice versa

Solution: The function gethostbyName() and gethostbyAddress() can be used to achieve this. These functions are part of socket library in python.

[see next page]

```python
# Python program to convert domain-name to ip-
address and viceversa

import socket

while True:
    option = int(
        input("1 to get ip-address\n2 to get domain-
name\n3 to exit\n--> "))
    if option == 1:
        domainName = input("Enter domain-name: ")
        try:
            ipAddress = socket.gethostbyname(domainName)
            print(f'IP Adress: {ipAddress}')
        except:
            print(
                "The domain-
name you entered is invalid.\nPlease follow this structure: <text>
.<text>.<text>")
    elif option == 2:
        ipAddress = input("Enter IP-Address: ")
        try:
            infoFromDns = socket.gethostbyaddr(ipAddress)
            domainName = infoFromDns[0]
            print(f'Domain-Name: {domainName}')
        except:
            print("Entered IP is invalid")
    else:
        break
```



In this screenshot you can see that we have given different inputs and got the corresponding outputs for each input.

3) Develop a program to view the data of top 50 movies in IMDB. (Movie name,actors, IMDB ratings)

Solution: We have used the library called BeautifulSoup for web scraping and displayed the details accordingly.

```
# Program to give the top 50 rated movies details from IMDB websit
e
# Do "pip install bs4"  for installing Beautifulsoup library
# Do "pip install lxml" for installing the lxml parser


from bs4 import BeautifulSoup
import requests
import re


def getData():
    url = "http://www.imdb.com/chart/top"
    response = requests.get(url)
    dataList = []
    soup = BeautifulSoup(response.text, 'lxml')
    movies = soup.select('td.titleColumn')
    ratings = soup.select('td.ratingColumn strong')
    crew = soup.select('td.titleColumn a')
    for i in range(0, 50):
        tempList = []
        tempList.append(' '.join(movies[i].get_text().split()))
        tempList.append(crew[i].attrs.get('title'))
        tempList.append(ratings[i].get_text())
        dataList.append(tempList)
    return dataList


dataList = getData()
for i in dataList:
    print(f'{i[0]} - {i[1]} - {i[2]}')
```

```
PS D:\Computer Networks Lab> python -u "d:\Computer Networks Lab\Week - 2\Q3\Q3.py"
1. The Shawshank Redemption (1994) - Frank Darabont (dir.), Tim Robbins, Morgan Freeman - 9.2
2. The Godfather (1972) - Francis Ford Coppola (dir.), Marlon Brando, Al Pacino - 9.1
3. The Godfather: Part II (1974) - Francis Ford Coppola (dir.), Al Pacino, Robert De Niro - 9.0
4. The Dark Knight (2008) - Christopher Nolan (dir.), Christian Bale, Heath Ledger - 9.0
5. 12 Angry Men (1957) - Sidney Lumet (dir.), Henry Fonda, Lee J. Cobb - 8.9
6. Schindler's List (1993) - Steven Spielberg (dir.), Liam Neeson, Ralph Fiennes - 8.9
7. The Lord of the Rings: The Return of the King (2003) - Peter Jackson (dir.), Elijah Wood, Viggo Mortensen - 8.9
8. Pulp Fiction (1994) - Quentin Tarantino (dir.), John Travolta, Uma Thurman - 8.8
9. Il buono, il brutto, il cattivo (1966) - Sergio Leone (dir.), Clint Eastwood, Eli Wallach - 8.8
10. The Lord of the Rings: The Fellowship of the Ring (2001) - Peter Jackson (dir.), Elijah Wood, Ian McKellen - 8.8
11. Fight Club (1999) - David Fincher (dir.), Brad Pitt, Edward Norton - 8.8
12. Forrest Gump (1994) - Robert Zemeckis (dir.), Tom Hanks, Robin Wright - 8.7
13. Inception (2010) - Christopher Nolan (dir.), Leonardo DiCaprio, Joseph Gordon-Levitt - 8.7
14. The Lord of the Rings: The Two Towers (2002) - Peter Jackson (dir.), Elijah Wood, Ian McKellen - 8.7
15. Star Wars: Episode V - The Empire Strikes Back (1980) - Irvin Kershner (dir.), Mark Hamill, Harrison Ford - 8.7
16. The Matrix (1999) - Lana Wachowski (dir.), Keanu Reeves, Laurence Fishburne - 8.6
17. Goodfellas (1990) - Martin Scorsese (dir.), Robert De Niro, Ray Liotta - 8.6
18. One Flew Over the Cuckoo's Nest (1975) - Milos Forman (dir.), Jack Nicholson, Louise Fletcher - 8.6
19. Shichinin no samurai (1954) - Akira Kurosawa (dir.), Toshirô Mifune, Takashi Shimura - 8.6
20. Se7en (1995) - David Fincher (dir.), Morgan Freeman, Brad Pitt - 8.6
21. The Silence of the Lambs (1991) - Jonathan Demme (dir.), Jodie Foster, Anthony Hopkins - 8.6
22. Cidade de Deus (2002) - Fernando Meirelles (dir.), Alexandre Rodrigues, Leandro Firmino - 8.6
23. La vita è bella (1997) - Roberto Benigni (dir.), Roberto Benigni, Nicoletta Braschi - 8.6
24. It's a Wonderful Life (1946) - Frank Capra (dir.), James Stewart, Donna Reed - 8.6
25. Star Wars (1977) - George Lucas (dir.), Mark Hamill, Harrison Ford - 8.6
26. Saving Private Ryan (1998) - Steven Spielberg (dir.), Tom Hanks, Matt Damon - 8.5
27. Interstellar (2014) - Christopher Nolan (dir.), Matthew McConaughey, Anne Hathaway - 8.5
28. Sen to Chihiro no kamikakushi (2001) - Hayao Miyazaki (dir.), Daveigh Chase, Suzanne Pleshette - 8.5
29. The Green Mile (1999) - Frank Darabont (dir.), Tom Hanks, Michael Clarke Duncan - 8.5
30. Gisaengchung (2019) - Bong Joon Ho (dir.), Kang-ho Song, Sun-kyun Lee - 8.5
31. Léon (1994) - Luc Besson (dir.), Jean Reno, Gary Oldman - 8.5
32. Seppuku (1962) - Masaki Kobayashi (dir.), Tatsuya Nakadai, Akira Ishihama - 8.5
33. The Pianist (2002) - Roman Polanski (dir.), Adrien Brody, Thomas Kretschmann - 8.5
34. The Usual Suspects (1995) - Bryan Singer (dir.), Kevin Spacey, Gabriel Byrne - 8.5
35. Terminator 2: Judgment Day (1991) - James Cameron (dir.), Arnold Schwarzenegger, Linda Hamilton - 8.5
36. Back to the Future (1985) - Robert Zemeckis (dir.), Michael J. Fox, Christopher Lloyd - 8.5
37. Psycho (1960) - Alfred Hitchcock (dir.), Anthony Perkins, Janet Leigh - 8.5
38. The Lion King (1994) - Roger Allers (dir.), Matthew Broderick, Jeremy Irons - 8.5
39. Modern Times (1936) - Charles Chaplin (dir.), Charles Chaplin, Paulette Goddard - 8.5
40. American History X (1998) - Tony Kaye (dir.), Edward Norton, Edward Furlong - 8.5
41. City Lights (1931) - Charles Chaplin (dir.), Charles Chaplin, Virginia Cherrill - 8.5
42. Hotaru no haka (1988) - Isao Takahata (dir.), Tsutomu Tatsumi, Ayano Shiraishi - 8.5
43. Whiplash (2014) - Damien Chazelle (dir.), Miles Teller, J.K. Simmons - 8.5
44. Gladiator (2000) - Ridley Scott (dir.), Russell Crowe, Joaquin Phoenix - 8.5
45. The Departed (2006) - Martin Scorsese (dir.), Leonardo DiCaprio, Matt Damon - 8.5
46. The Intouchables (2011) - Olivier Nakache (dir.), François Cluzet, Omar Sy - 8.5
47. The Prestige (2006) - Christopher Nolan (dir.), Christian Bale, Hugh Jackman - 8.5
```

You can see that we have listed the top 50 movies with their name, cast and rating according to IMDB website.

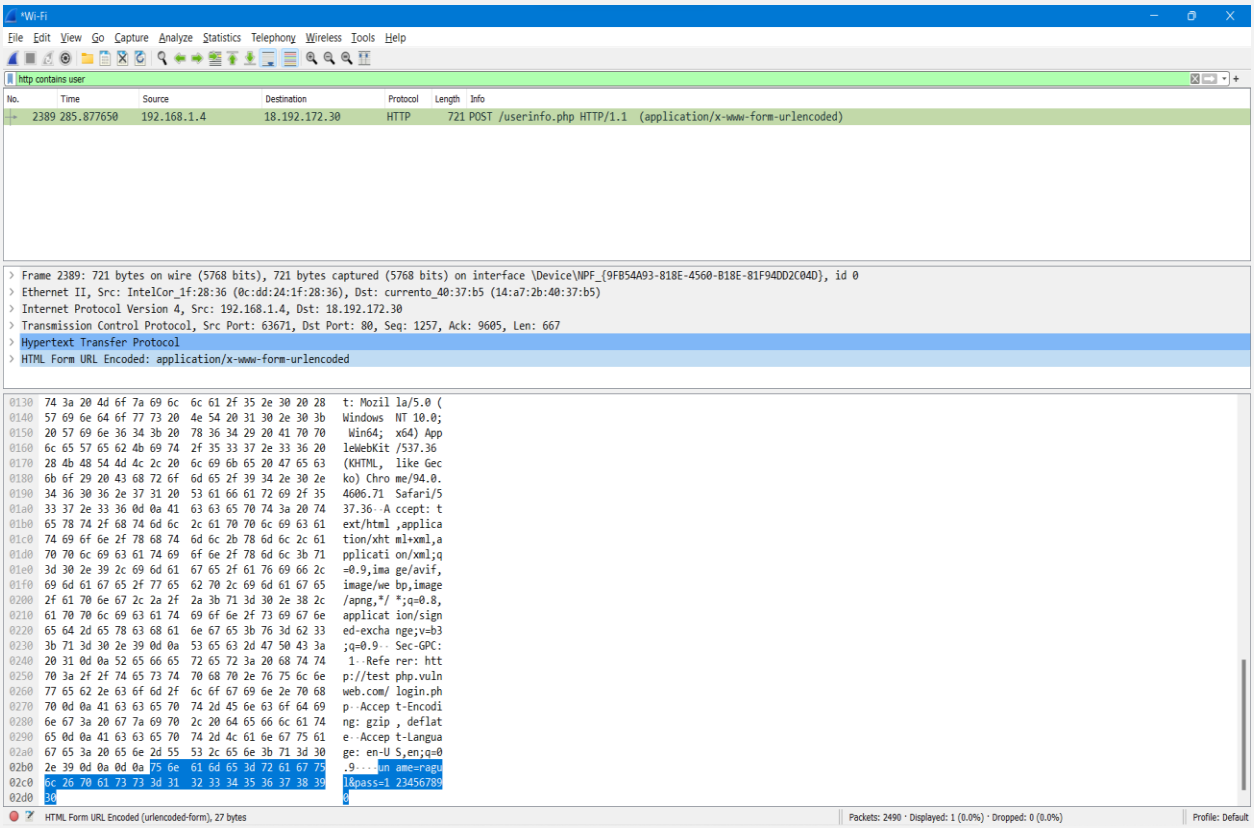
4) Write a program to display the details of an input URL (status code, headers, history, encoding, reason, cookies, elapsed, request)

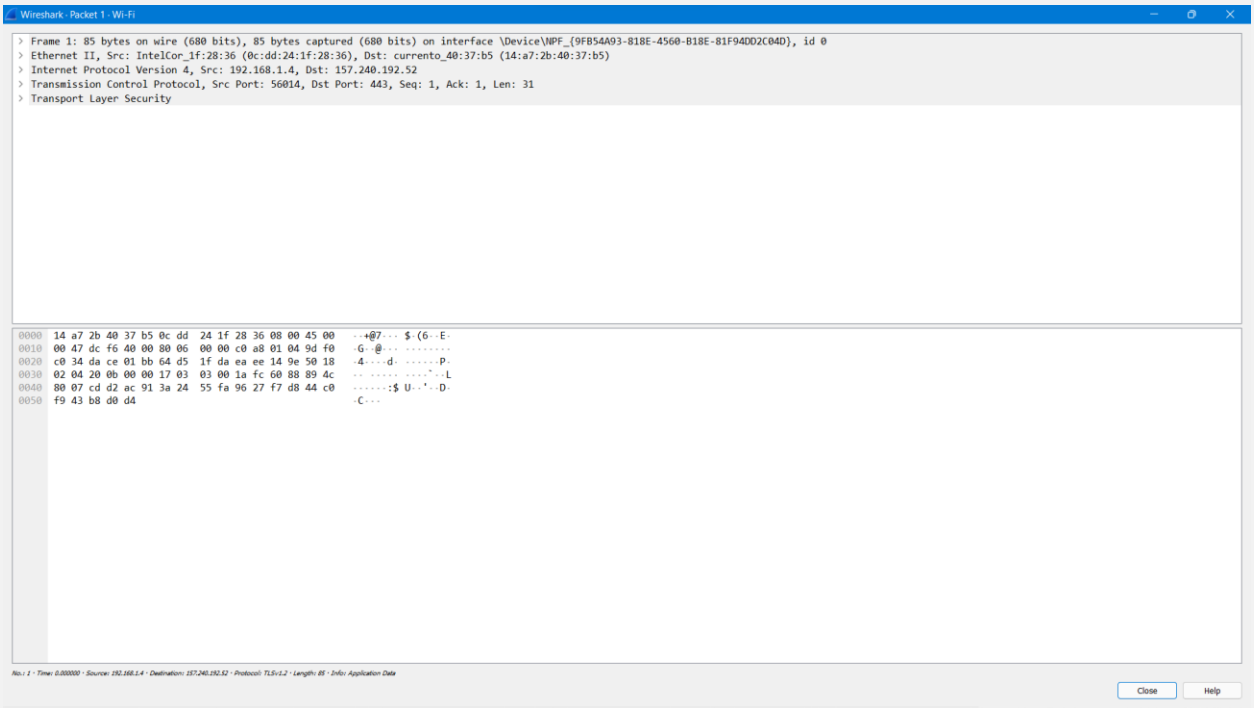
Solution: We have used the request library from python to program this question.

```
# Program to display the details of an input URL
# (status code, headers,history, encoding, reason, cookies, elapse
d, request)

import requests


def startProgram():
    url = input("Enter a vaild URL: ")
    try:
        response = requests.get(url)
        printOutput(response)
    except:
        print("Entered URL is not Valid")


def printOutput(response):
    print("\n<-- Details of the Input URL -->")
    print(f'\nStatus code: {response.status_code}')
    print(f'\nHistory: {response.history}')
    print(f'\nEncoding: {response.encoding}')
    print(f'\nReason: {response.reason}')
    print(f'\nElapsed: {response.encoding}')
    print(f'\nRequest: {response.request}')

    print(f'\nCookies: ')
    for i in response.cookies:
        print(i)

    print(f'\nHeaders:')
    for key, value in response.headers.items():
        print(f'{key}: {value}')
startProgram()
```

```
PS D:\Computer Networks Lab> python -u "d:\Computer Networks Lab\Week - 2\Q4\Q4.py"
Enter a vaild URL: https://iris.nitk.ac.in

<-- Details of the Input URL -->

Status code: 200

History: [<Response [301]>]

Encoding: utf-8

Reason: OK

Elapsed: utf-8

Request: <PreparedRequest [GET]>

Cookies:
<Cookie _akshay_IRIS_session=db8abdab8becb9dbf4ecc0b6ab488443 for .iris.nitk.ac.in/>

Headers:
Server: nginx
Date: Sat, 02 Oct 2021 19:07:16 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding, Accept, Origin
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-Permitted-Cross-Domain-Policies: none
Referrer-Policy: strict-origin-when-cross-origin
X-IRIS-SERVER-ID: N1
ETag: W/"6274daeeed17e3c2405bc26664e3084e"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: _akshay_IRIS_session=db8abdab8becb9dbf4ecc0b6ab488443; domain=iris.nitk.ac.in; path=/; secure; HttpOnly
X-Request-Id: 58625b8f-8dd4-4a6e-b9e7-0e12e5f9a603
X-Runtime: 0.033647
Strict-Transport-Security: max-age=63072000; includeSubDomains
X-Upstream-Server: 10.15.0.111:8540
Content-Encoding: gzip
PS D:\Computer Networks Lab>
```

5.Capture HTTP packets by visiting a HTTP Website, analyze the packets and significance of its various fields. Do the same for HTTPS packets and compare both





The main difference that can be observed from the http and the https packet is that the data in the http packet is plain text whereas in the https packet the data is encrypted by passing through one more protocol called Transport Layer Security(tls) protocol.

Ethernet Destination: This has the mac address of the destination device.

Ethernet source: This field has the mac address of the source device.

IPv4 Source Address: This field has the version 4 IP address of the source device.

IPv4 Destination Address: This field has the version 4 IP address of the destination device.

TCP Source Port: Port number for this tcp service on the source machine

TCP Destination Port: Port number for this tcp service on the destination machine

Https has one more protocol called tls. The tls for the packet we analyzed is http-over-tls.

The packets also have various other fields like window size, checksum, TCP segment length, Time to Live, fragment offset etc., with various other details of the packet.

Thank you