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 via 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()

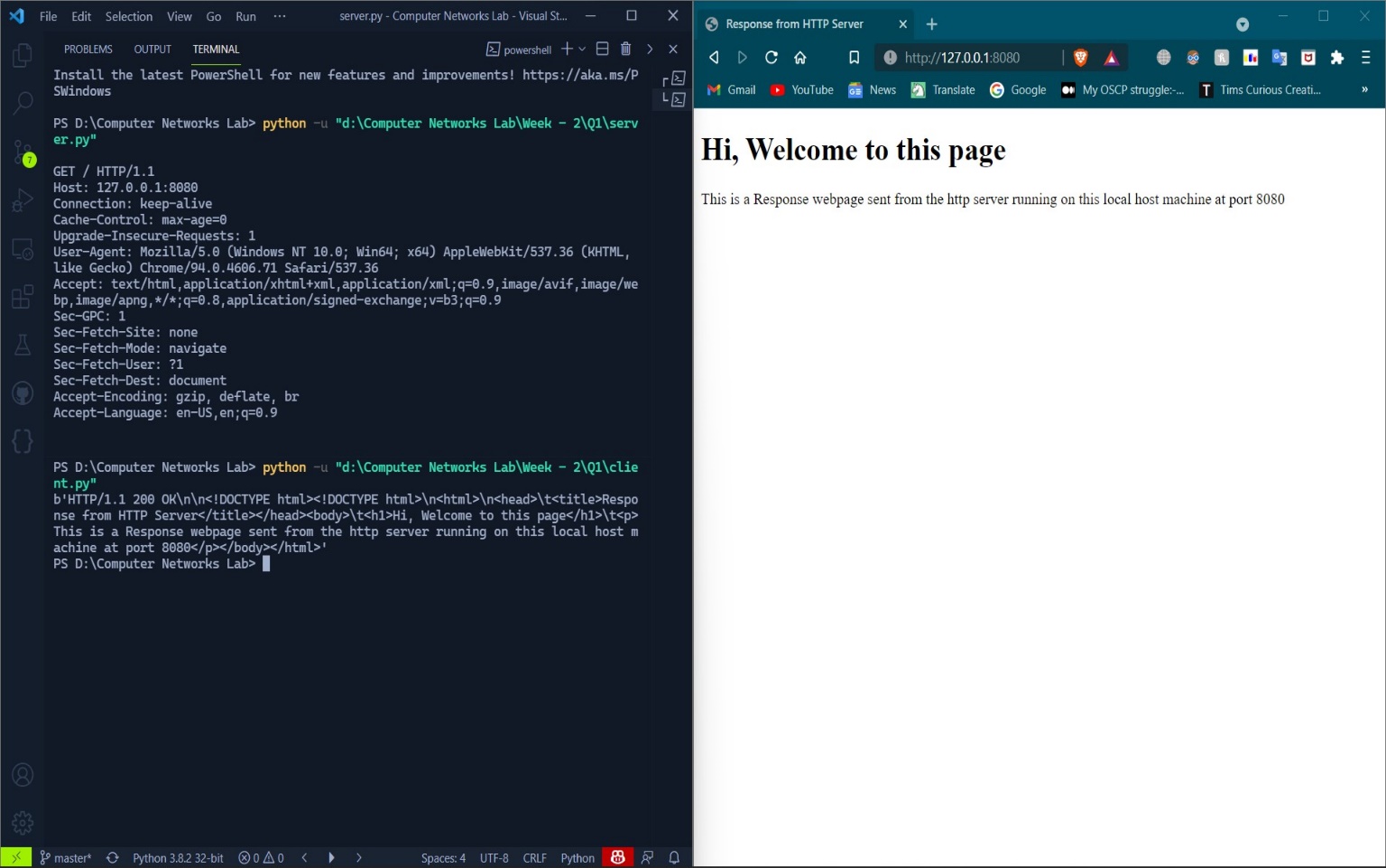
clientSocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

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 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]

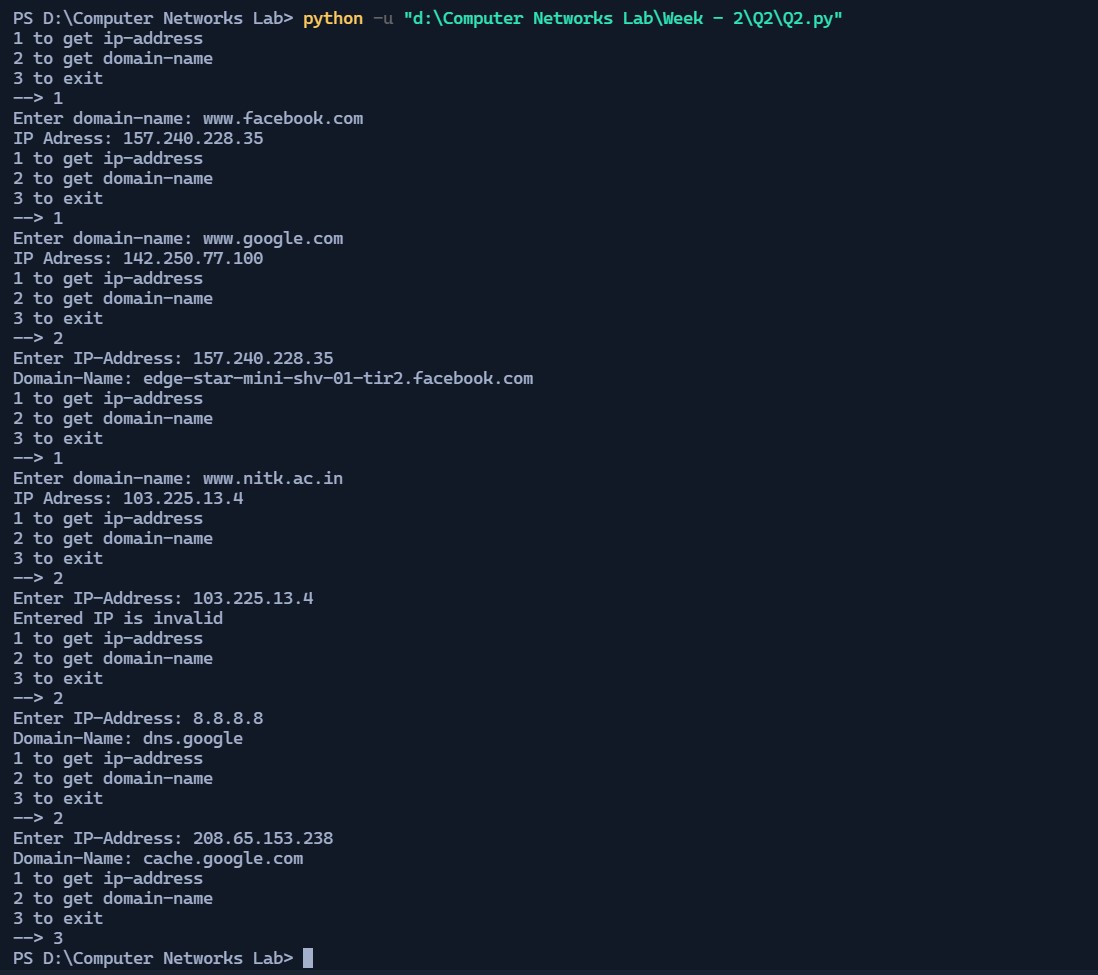
            print(f'Domain-Name: {domainName}')

        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 website*

*# 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())

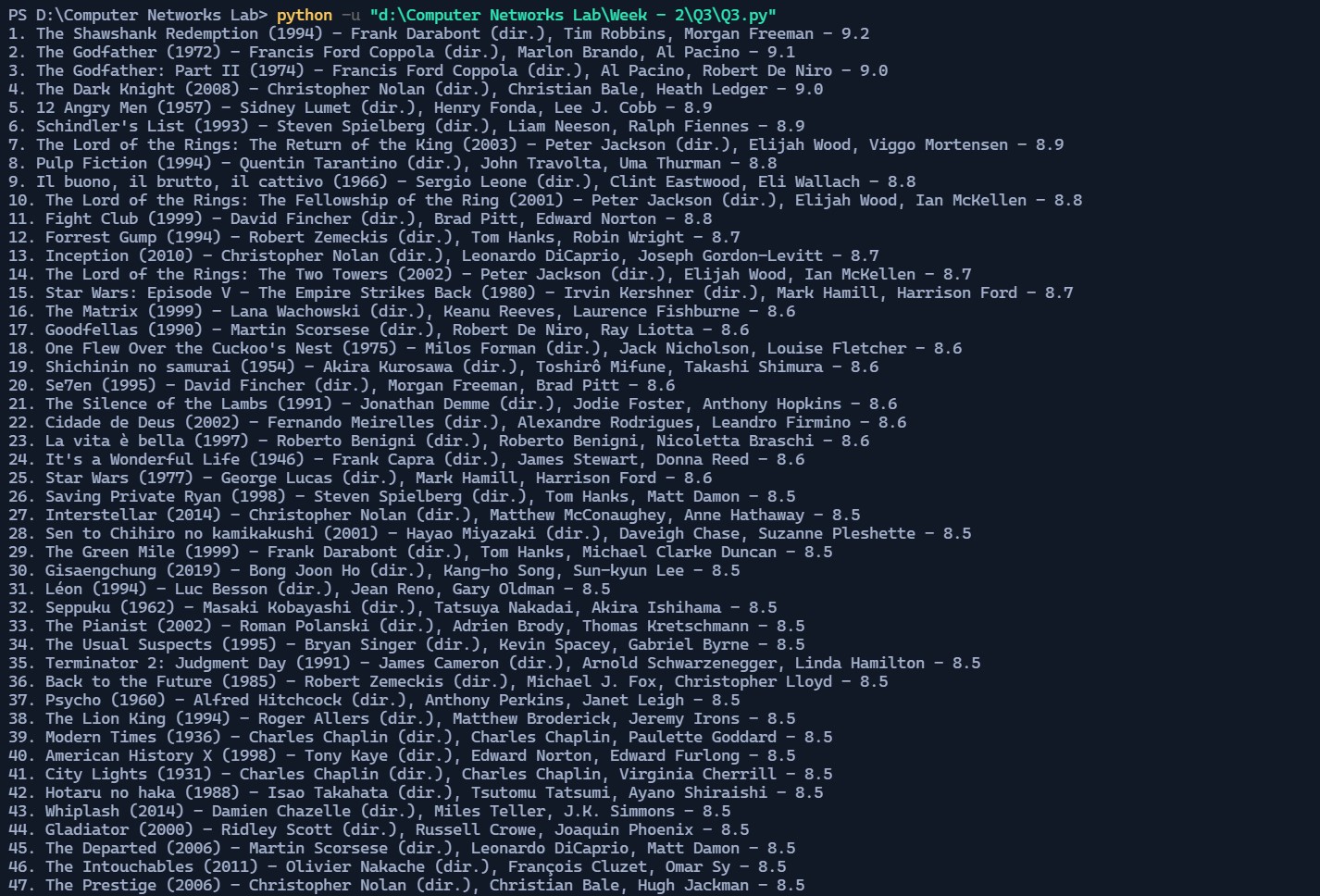
        dataList.append(tempList)

    return dataList

dataList = getData()

for i in dataList:

    print(f'{i[0]} - {i[1]} - {i[2]}')



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, elapsed, 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:

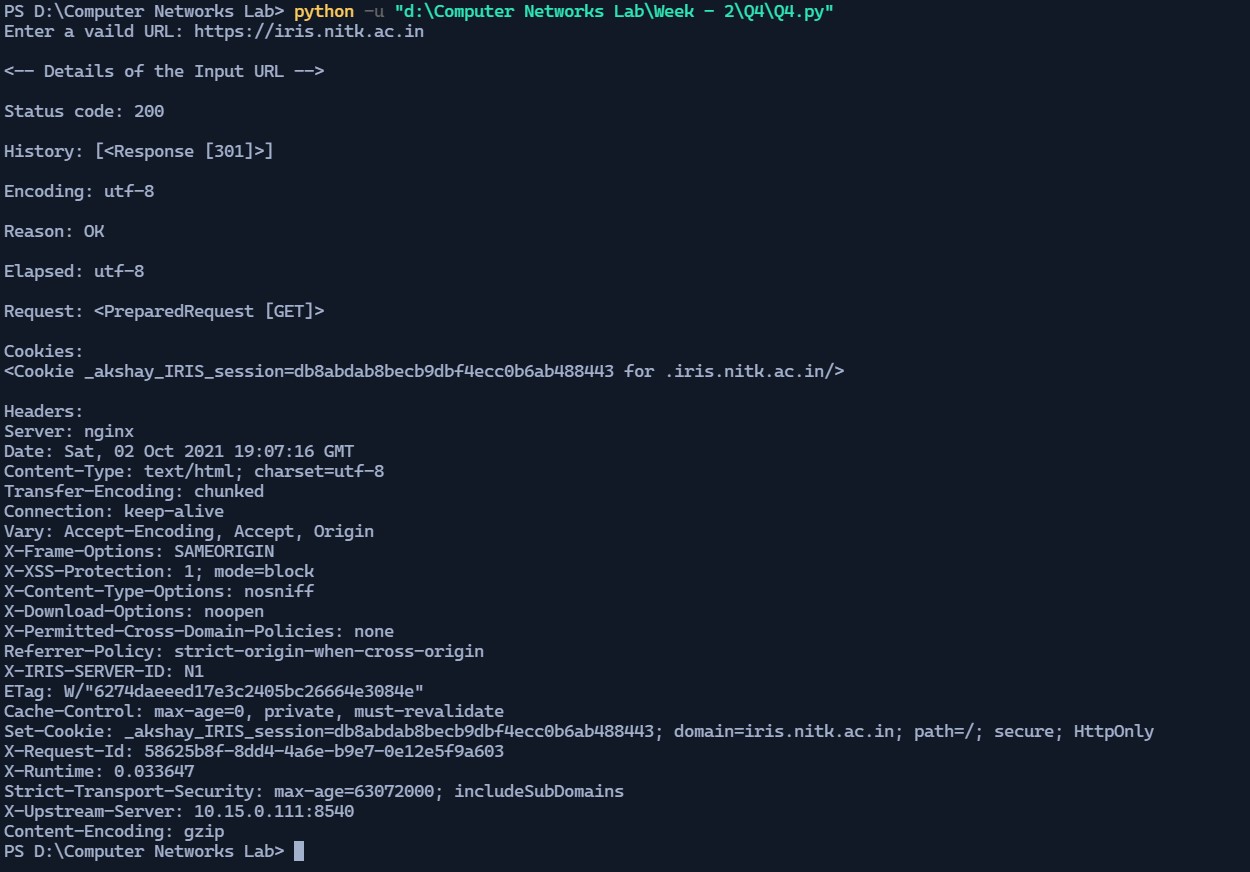
        print(i)

    print(f'\nHeaders:')

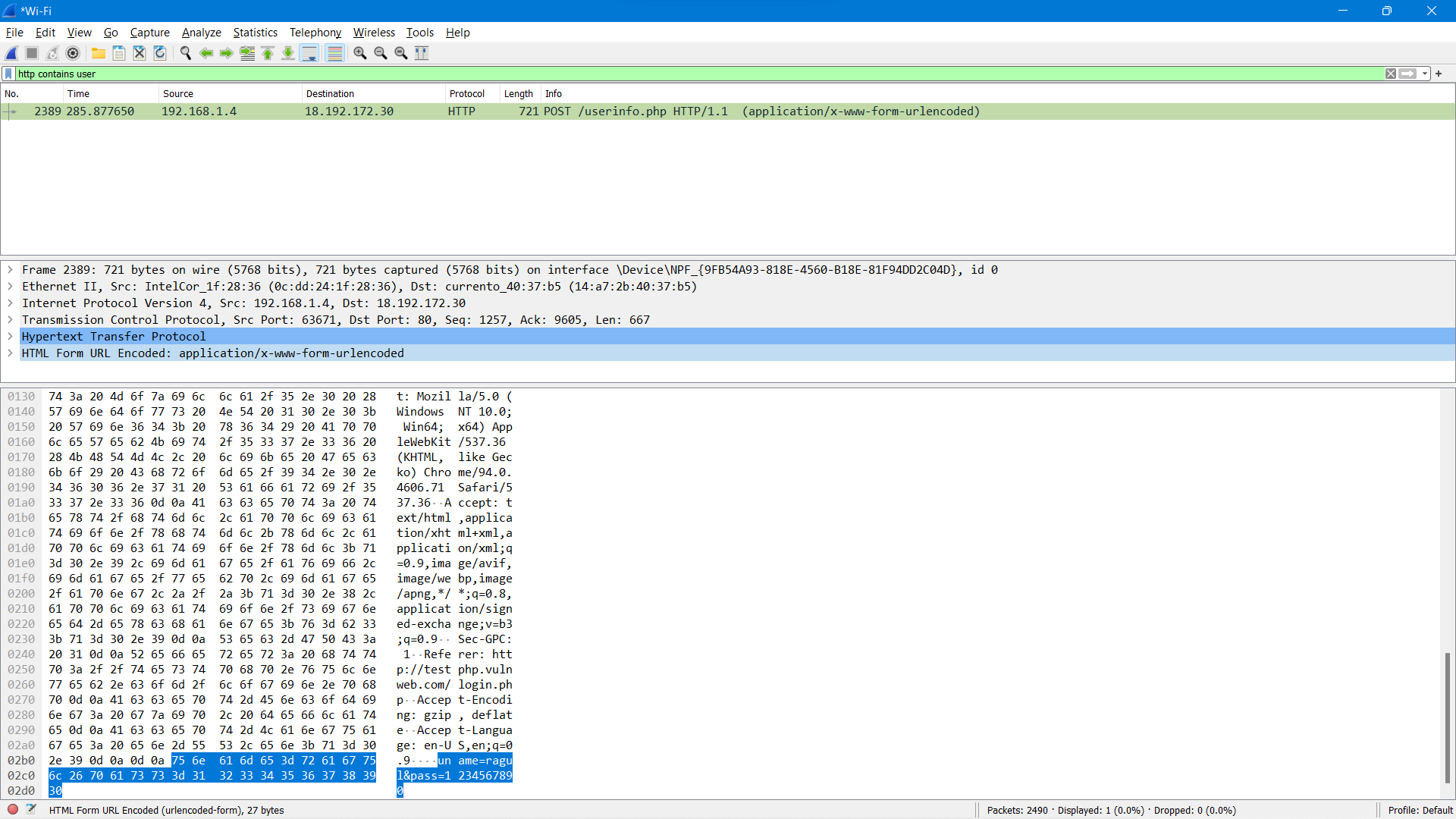
    for key, value in response.headers.items():

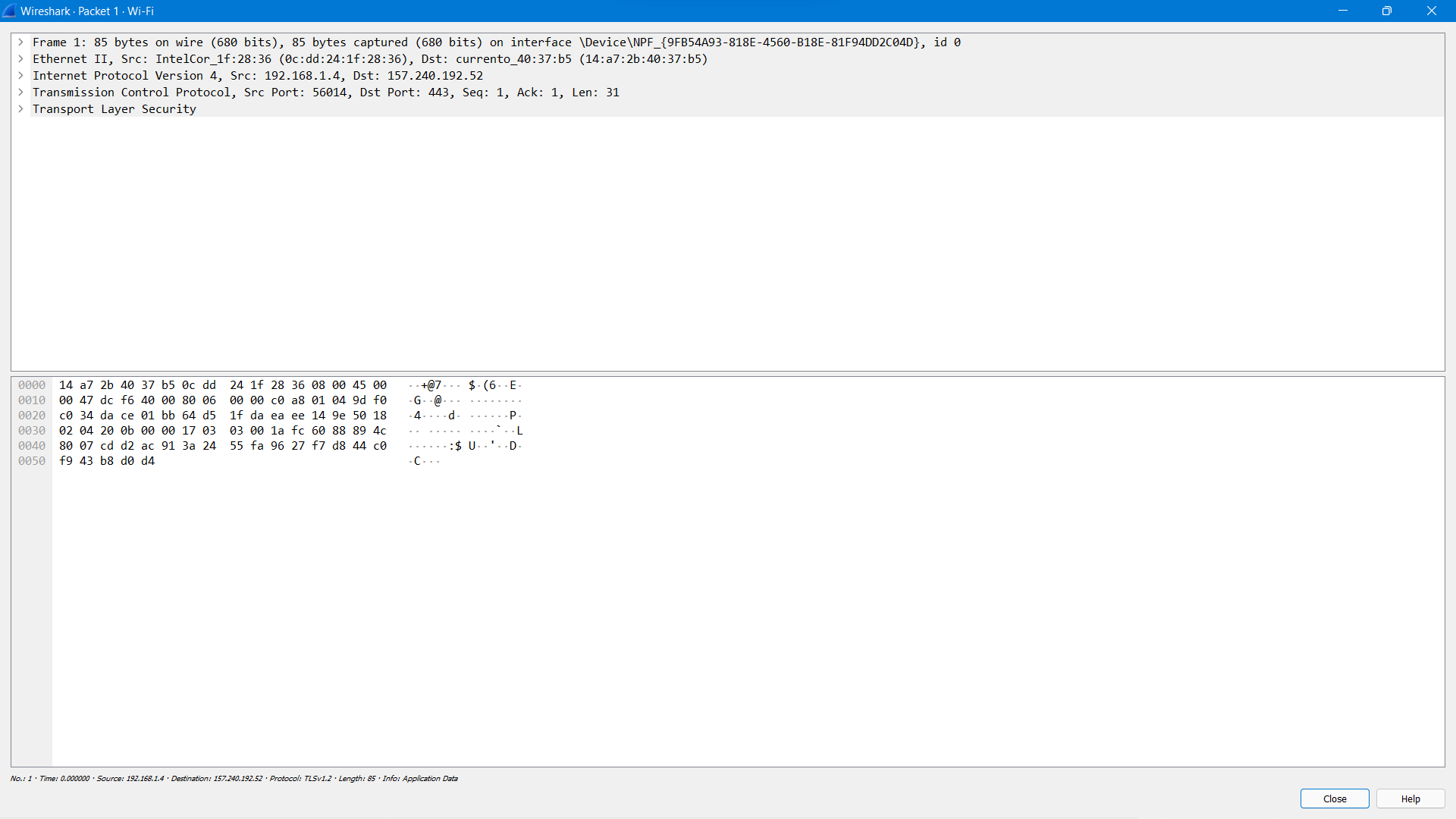
        print(f'{key}: {value}')

startProgram()



# 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\*\*\*