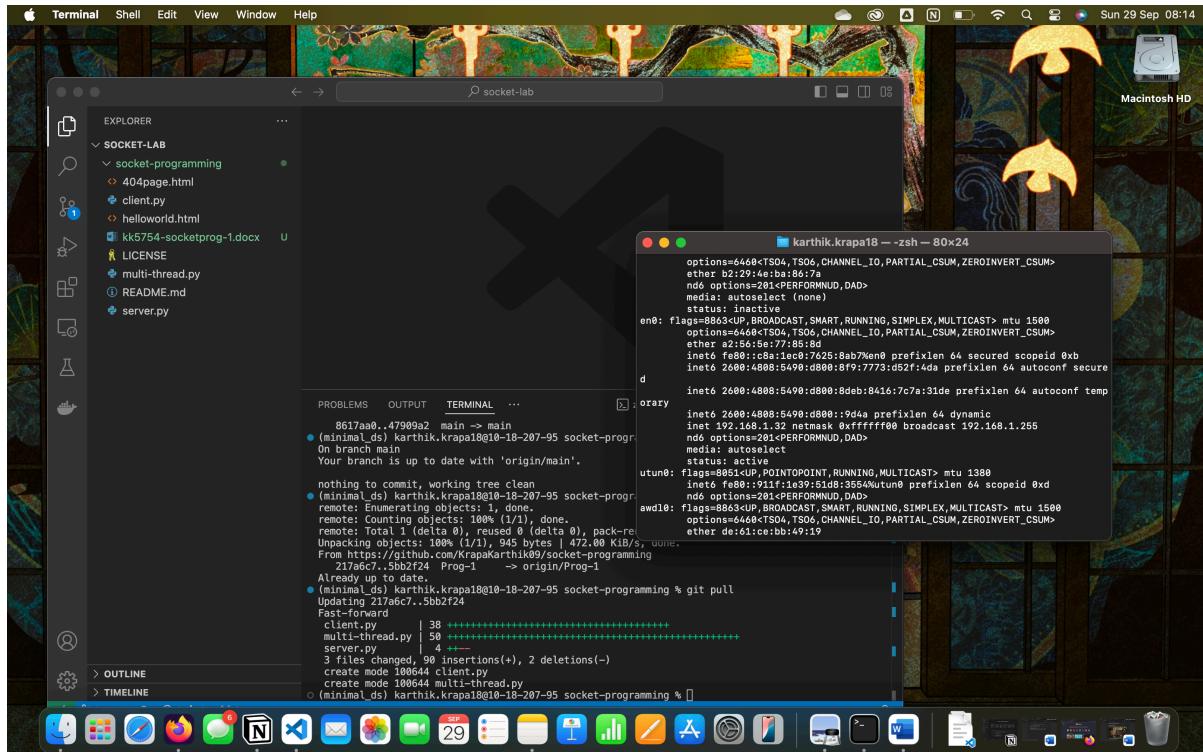


Computer Networking - Socket Programming Lab - 1

Identifying Lab Environment

The below image depicts my lab environment on my laptop, it uses MacOS.



Server.py File

Code with comment lines for explanation

```
from socket import *
import sys

def webServer(port):
    serverPort = port
    serverSocket = socket(AF_INET, SOCK_STREAM) #server socket - TCP byte stream
    serverSocket.bind(('',serverPort)) #binding port to server port
    serverSocket.listen(1) #server begins listening for incoming requests
    print('Server being set up...')

    while True:
        print ('Server is ready !')
        connectionSocket, addr = serverSocket.accept() #server waits on accept new
        socket returned on request as this is TCP
        print('Request accepted from:', addr)
        try:
```

```
message = connectionSocket.recv(1024).decode() #decodes the received
byte stream message
    filename = message.split()[1]
    f = open(filename[1:], 'r')
    outputdata = f.read()

    headerLine = 'HTTP/1.1 200 OK\r\n' #header line containing response
status code and status phrase
    connectionSocket.send(headerLine.encode()) #encoding to send back
    connectionSocket.send('\r\n'.encode()) #carriage return otherwise the
headers won't be displayed

    for i in range(0, len(outputdata)):
        connectionSocket.send(outputdata[i].encode())
    connectionSocket.send('\r\n'.encode())

    connectionSocket.close()

except IOError:
    errHeaderLine = 'HTTP/1.1 404 Not Found\r\n' #r for carriage return
and n for new line
    connectionSocket.send(errHeaderLine.encode())
    connectionSocket.send('\r\n'.encode())
    f2 = open('404page.html', 'r')
    outputdata_err = f2.read()

    for i in range(0, len(outputdata_err)):
        connectionSocket.send(outputdata_err[i].encode())
    connectionSocket.send('\r\n'.encode())
    connectionSocket.close()

serverSocket.close()
sys.exit()

if __name__ == '__main__':
    webServer(6879)
```

Output:

helloworld.html 200

Name: Krapa Karthik NYU ID: N12039854 NET ID: kk5754

The terminal window shows the following Python code for a socket server:

```
from socket import *
import sys
def webServer(port):
    serverPort = port
    serverSocket = socket(AF_INET, SOCK_STREAM) #server socket - TCP byte stream
    serverSocket.bind(('',serverPort)) #binding port to server port
    serverSocket.listen(1) #server begins listening for incoming requests
    print('Server being set up...')

while True:
    connectionSocket, addr = serverSocket.accept() #server waits to accept new socket returned on request as this is TCP
    print('Request accepted from:', addr)
    try:
        message = connectionSocket.recv(1024).decode() #decodes the received byte stream message
        filename = message.split()[1]
        f = open(filename[1:], 'r')
        outputdata = f.read()
        headerLine = 'HTTP/1.1 200 OK\r\n' #header line containing response status code and status phrase
        connectionSocket.send(headerLine.encode()) #encoding to send back
        connectionSocket.send(outputdata.encode()) #carriage return otherwise the headers won't be displayed
    except:
        connectionSocket.close()

serverSocket.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
serverSocket.bind(('',serverPort)) #binding port to server port
connectionSocket.send(headerLine.encode())
connectionSocket.send(outputdata.encode())
connectionSocket.close()
```

The terminal output shows the server starting and accepting a connection from '127.0.0.1' on port 61355.

The Firefox browser window displays the following response:

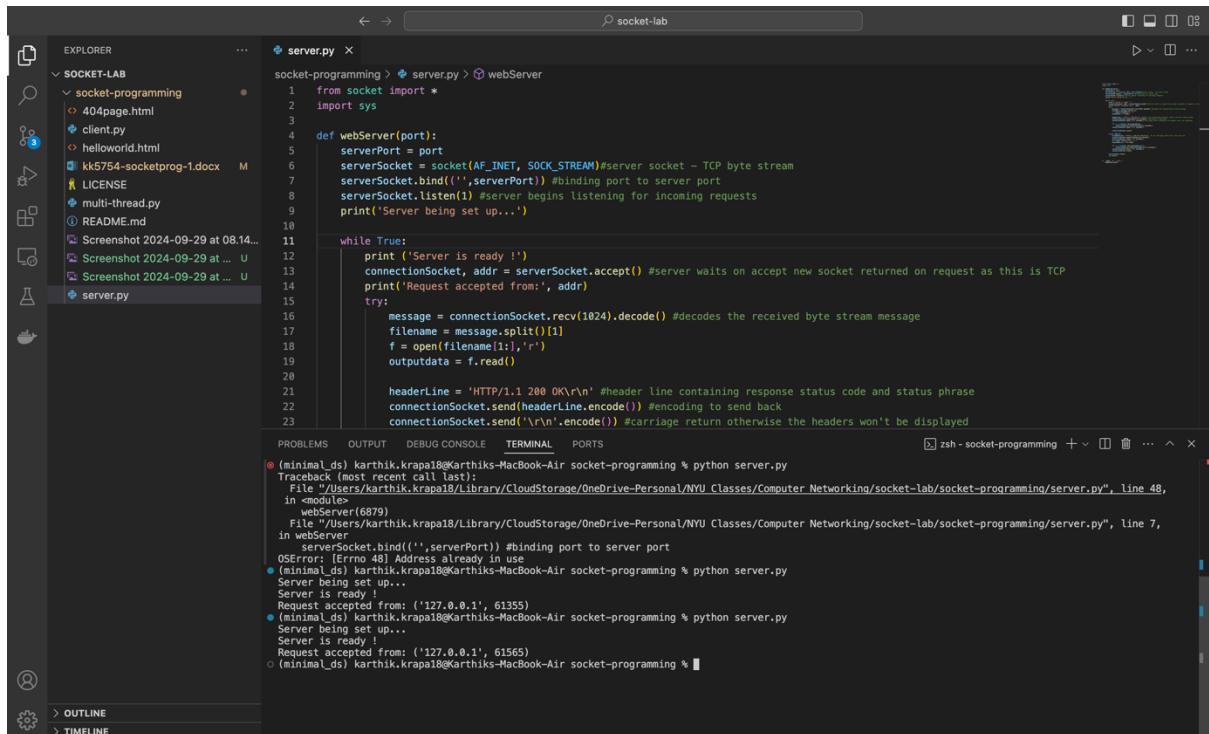
This page is for socket programming assignment. - Krapa Karthik NET ID: kk5754 NYU ID: 12039854

The Network tab of the developer tools shows the following requests:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	127.0.0.1:6879	helloworld.html	document	html	354 B	335 B	~3ms
0	GET	127.0.0.1:6879	favicon.ico	FaviconLoader.sys.mjs:175 (I...)	NS_ERROR_CONNECTION_R...	0 B	0 B	0 ms

404page.html 404

Name: Krapa Karthik NYU ID: N12039854 NET ID: kk5754

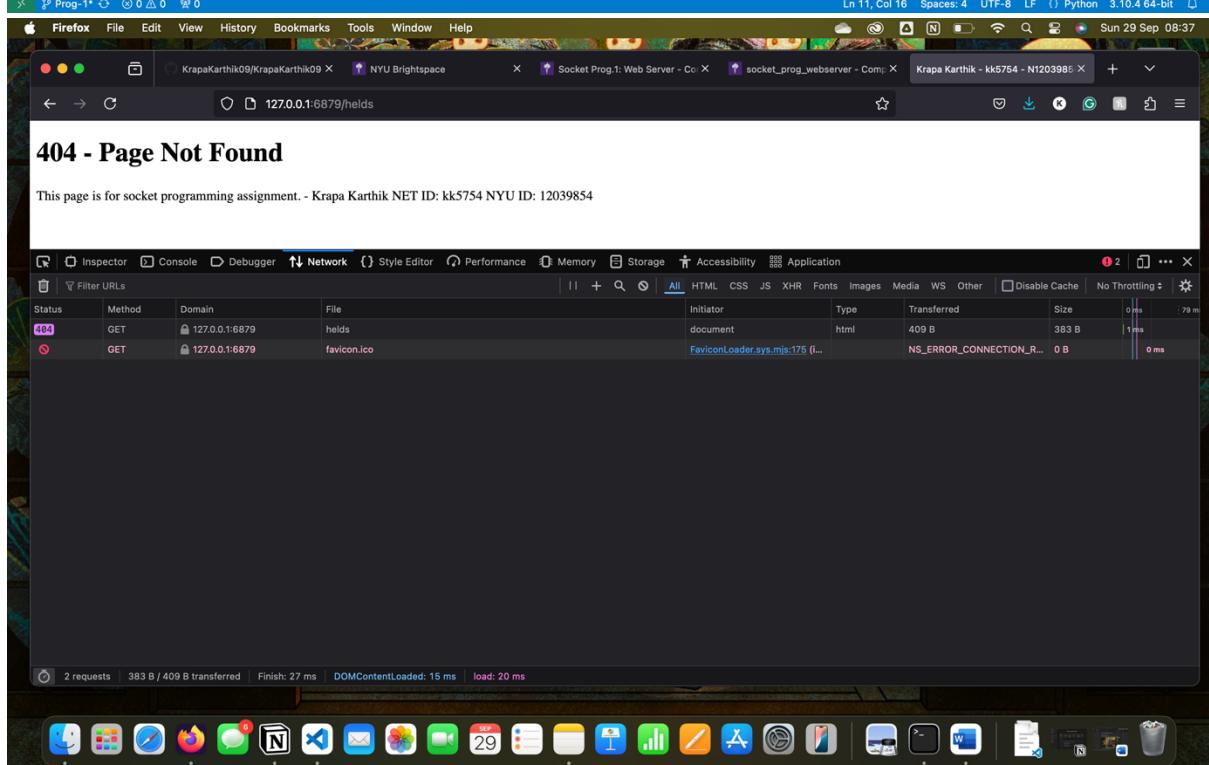


The screenshot shows a code editor with an open file named `server.py`. The code implements a simple TCP socket server. It binds to port 6879, waits for incoming connections, and reads the request header to determine the file to serve. The terminal below shows the server running and accepting requests from localhost:61555.

```
from socket import *
import sys
def webServer(port):
    serverPort = port
    serverSocket = socket(AF_INET, SOCK_STREAM) #server socket - TCP byte stream
    serverSocket.bind(('',serverPort)) #binding port to server port
    serverSocket.listen(1) #server begins listening for incoming requests
    print('Server being set up...')

    while True:
        connectionSocket, addr = serverSocket.accept() #server waits to accept new socket returned on request as this is TCP
        print('Request accepted from:', addr)
        try:
            message = connectionSocket.recv(1024).decode() #decodes the received byte stream message
            filename = message.split()[1]
            f = open(filename[1:],'r')
            outputdata = f.read()
            headerLine = 'HTTP/1.1 200 OK\r\n' #header line containing response status code and status phrase
            connectionSocket.send(headerLine.encode()) #encoding to send back
            connectionSocket.send((outputdata+'\r\n').encode()) #carriage return otherwise the headers won't be displayed
        except:
            connectionSocket.close()

(minimal_ds) karthik.krapa18@Karthiks-MacBook-Air socket-programming % python server.py
Traceback (most recent call last):
File "/Users/karthik.krapa18/Library/CloudStorage/OneDrive-Personal/NYU Classes/Computer Networking/socket_lab/socket_programming/server.py", line 48, in <module>
    webServer(6879)
  File "/Users/karthik.krapa18/Library/CloudStorage/OneDrive-Personal/NYU Classes/Computer Networking/socket_lab/socket_programming/server.py", line 7, in webServer
    serverSocket.bind(('',serverPort)) #binding port to server port
OSError: [Errno 48] Address already in use
(minimal_ds) karthik.krapa18@Karthiks-MacBook-Air socket-programming % python server.py
Server being set up...
Server is ready !
Request accepted from: ('127.0.0.1', 61555)
(minimal_ds) karthik.krapa18@Karthiks-MacBook-Air socket-programming %
Server is ready !
Request accepted from: ('127.0.0.1', 61565)
(minimal_ds) karthik.krapa18@Karthiks-MacBook-Air socket-programming %
```



The screenshot shows a Firefox browser window displaying a 404 - Page Not Found error page. The URL in the address bar is `127.0.0.1:6879/heads`. The developer tools Network tab shows two requests: one for `heads` (status 404) and one for `favicon.ico` (status 200).

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
404	GET	127.0.0.1:6879	heads	document	html	409 B	383 B	1 ms
	GET	127.0.0.1:6879	favicon.ico	FaviconLoader.sys.mjs:175 (I...)	NS_ERROR_CONNECTION_R...	0 B	0 B	0 ms

Optional Exercises

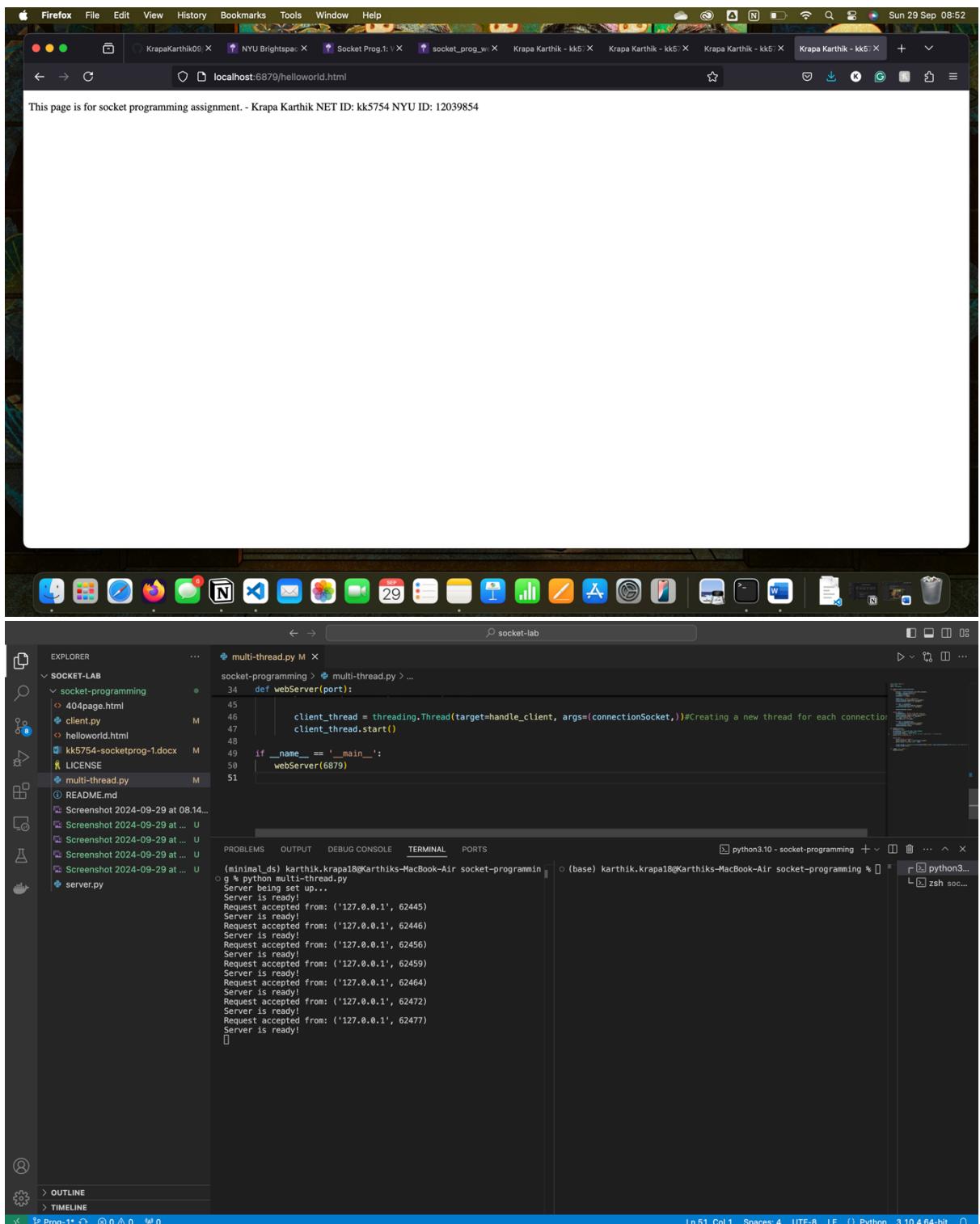
1. Multi-thread.py

```
1. from socket import *
2. import sys
3. import threading
4.
```

```
5. def handle_client(connectionSocket):
6.     try:
7.         message = connectionSocket.recv(1024).decode()
8.         filename = message.split()[1]
9.         f = open(filename[1:], 'r')
10.        outputdata = f.read()
11.
12.        headerLine = 'HTTP/1.1 200 OK\r\n'
13.        connectionSocket.send(headerLine.encode())
14.        connectionSocket.send('\r\n'.encode())
15.
16.        for char in outputdata:
17.            connectionSocket.send(char.encode())
18.        connectionSocket.send('\r\n'.encode())
19.
20.        connectionSocket.close()
21.
22.    except IOError:
23.        errHeaderLine = 'HTTP/1.1 404 Not Found\r\n'
24.        connectionSocket.send(errHeaderLine.encode())
25.        connectionSocket.send('\r\n'.encode())
26.        f2 = open('404page.html', 'r')
27.        outputdata_err = f2.read()
28.
29.        for char in outputdata_err:
30.            connectionSocket.send(char.encode())
31.        connectionSocket.send('\r\n'.encode())
32.        connectionSocket.close()
33.
34. def webServer(port):
35.     serverPort = port
36.     serverSocket = socket(AF_INET, SOCK_STREAM)
37.     serverSocket.bind(('', serverPort))
38.     serverSocket.listen(4) #this is for multiple connections - 4 connections
39.     print('Server being set up...')
40.
41.     while True:
42.         print('Server is ready!')
43.         connectionSocket, addr = serverSocket.accept()
44.         print('Request accepted from:', addr)
45.
46.         client_thread = threading.Thread(target=handle_client,
47.                                         args=(connectionSocket,))#Creating a new thread for each connection
48.         client_thread.start()
49.
50. if __name__ == '__main__':
51.     webServer(6879)
```

Output:

The below image indicates multi tabs open for indicating sessions with the server.



2. Client.py

```
52. from socket import *
53. import sys
54.
```

```
55. if (len(sys.argv) != 4): #For arguments to the script so that client can
   connect to server
56.     print('Incorrect number of arguments.')
57.     print('Help: To run python client.py <server_host> <server_port>
   <filename>')
58.     sys.exit()
59.
60. serverHost, serverPort, filename = sys.argv[1:] #server arguments
61. clientSocket = socket(AF_INET, SOCK_STREAM)
62. try:
63.     clientSocket.connect((serverHost, int(serverPort))) #client connecting
   to server
64. except:
65.     print('The server is currently inactive')
66.     clientSocket.close() #close connection if inactive
67.     sys.exit()
68. print('Connection OK.')
69.
70. httpRequest = 'GET /' + filename + ' HTTP/1.1\r\n\r\n' #HTTP Request
71. clientSocket.send(httpRequest.encode())
72. print('Request message sent.')
73.
74. print('Server HTTP Response:\r\n')
75.
76. data = ""
77. while True:
78.     clientSocket.settimeout(5)
79.     newData = clientSocket.recv(1024).decode()
80.     data += newData
81.     if (len(newData) == 0):
82.         break
83. print(data)
84.
85. print('Closing socket') #Closing socket
86. clientSocket.close()
```

Name: Krapa Karthik **NYU ID:** N12039854 **NET ID:** kk5754

Output:

The screenshot shows a terminal window with the following output:

```
(mininal_d) karpak.krapa1@Karthiks-MacBook-Air socket-programming % python server.py
Server being set up...
Server is ready
Request received from: ('127.0.0.1', 61921)
(mininal_d) karpak.krapa1@Karthiks-MacBook-Air socket-programming % (base) karpak.krapa1@Karthiks-MacBook-Air socket-programming % python client.py localhost 6879 helloworld.html
Request message sent.
Server HTTP Response:
HTTP/1.1 200 OK
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Krapa Karthik - kk5754 - N12039854</title>
</head>
<body>
    <p>This page is for socket programming assignment. - Krapa Karthik, NET ID: kk5754 NYU ID: 12039854</p>
</body>
</html>
```

The terminal also displays several screenshots and files related to the project.

Please refer to the code from my Github Repo Link :

<https://github.com/KrapaKarthik09/socket-programming/tree/Prog-1>