

**Computer Network Lab
CS359**

Tanishq Malu

Lab:3

1901CS63

Lab Topic : Socket Programming in Python

Question Number : FileName

Question1 : server.py

Question2 : multicient.py

Additional Files

create_config.py : It is used to create config file in json format, which is later used in configuration multicient server.

sample.json : Config file created by the create_config.py

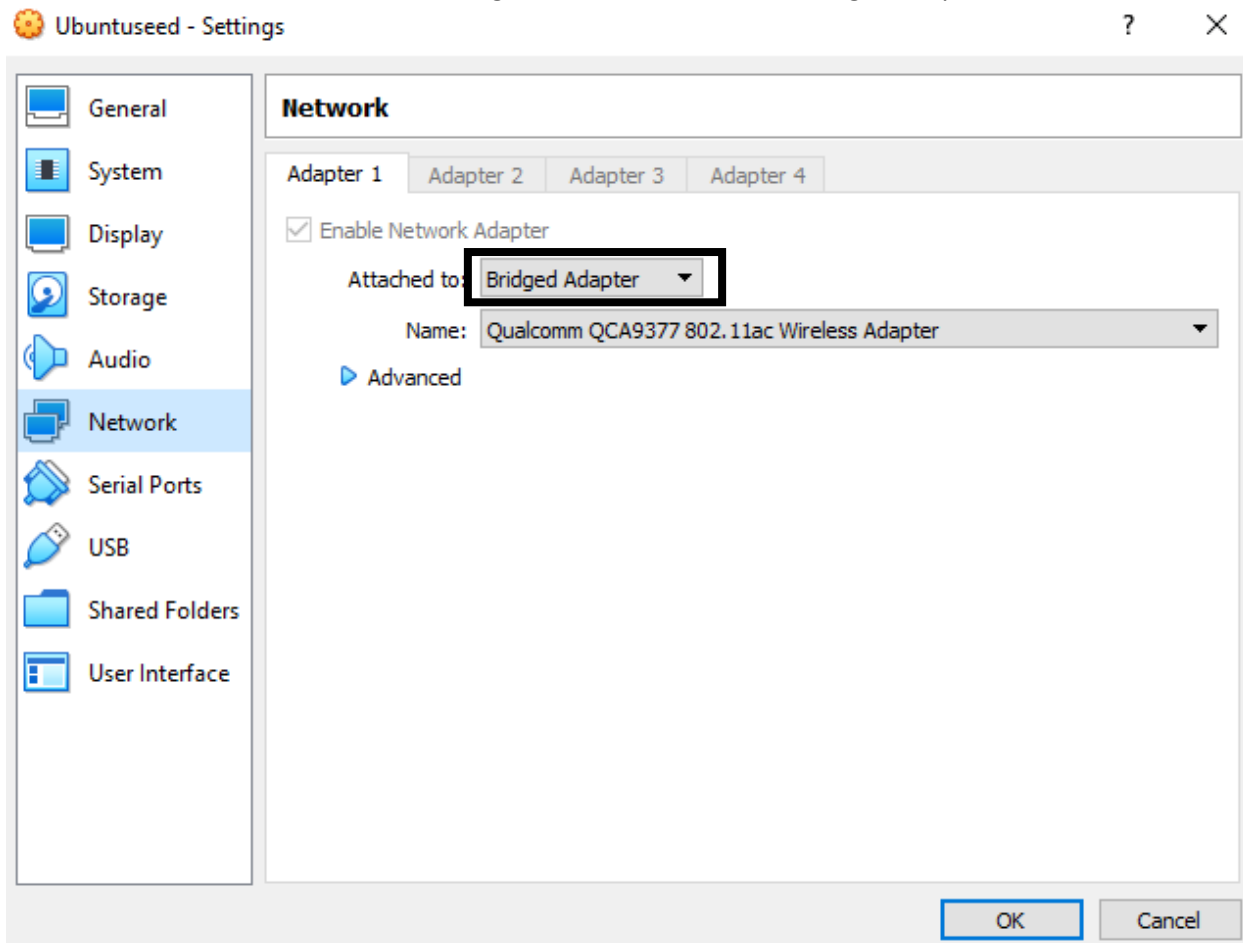
Socket.txt : Files which client can request

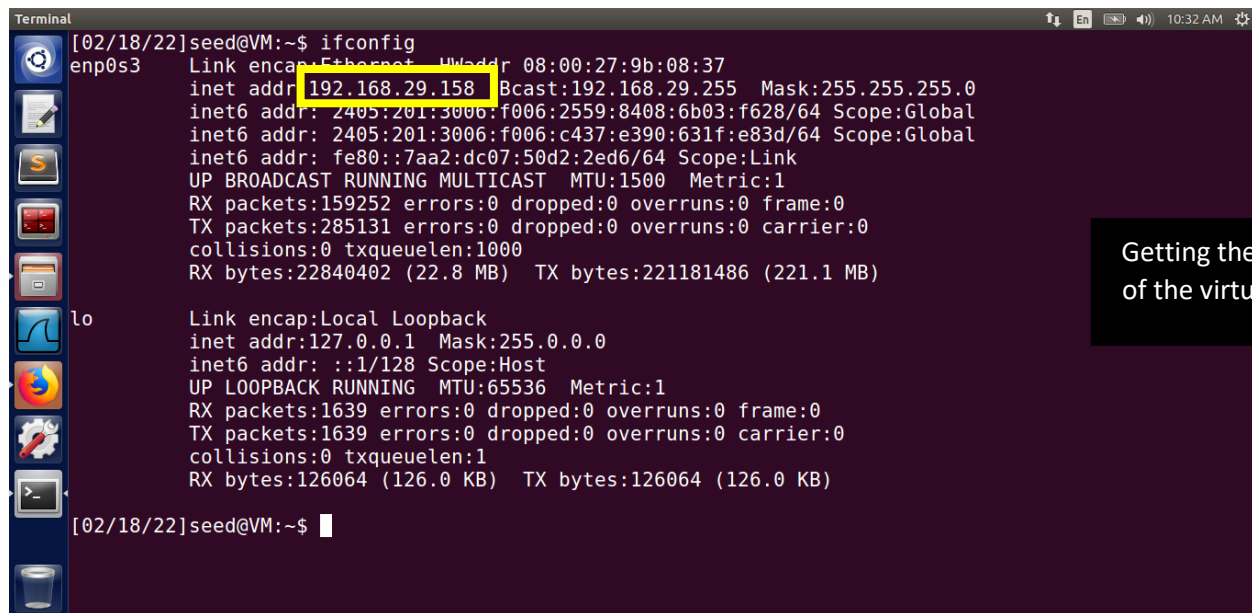
hello.html : Files which client can request

(Please turn over for detailed report)

Our basic aim is to create a server-client model using socket programming in python. For this purpose, I'll be running my server code on my ubuntu virtual machine and host my server on it. Simultaneously, I'll try to send requests as a client from my windows host.

So, First let us find the ip address on which we will host the server, i.e. IP address of our virtual machine. For this purpose, minor settings should be made in your VM so that your original host computer consider it as an another network. Network settings of VM should be set as bridged adapter to work this out.





```

[02/18/22]seed@VM:~$ ifconfig
enp0s3: Link encap:Ethernet HWaddr 08:00:27:9b:08:37
      inet addr: 192.168.29.158 Bcast:192.168.29.255 Mask:255.255.255.0
      inet6 addr: 2405:201:3006:f006:2559:8408:6b03:f628/64 Scope:Global
      inet6 addr: 2405:201:3006:f006:c437:e390:631f:e83d/64 Scope:Global
      inet6 addr: fe80::7aa2:dc07:50d2:2ed6/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:159252 errors:0 dropped:0 overruns:0 frame:0
      TX packets:285131 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:22840402 (22.8 MB)  TX bytes:221181486 (221.1 MB)

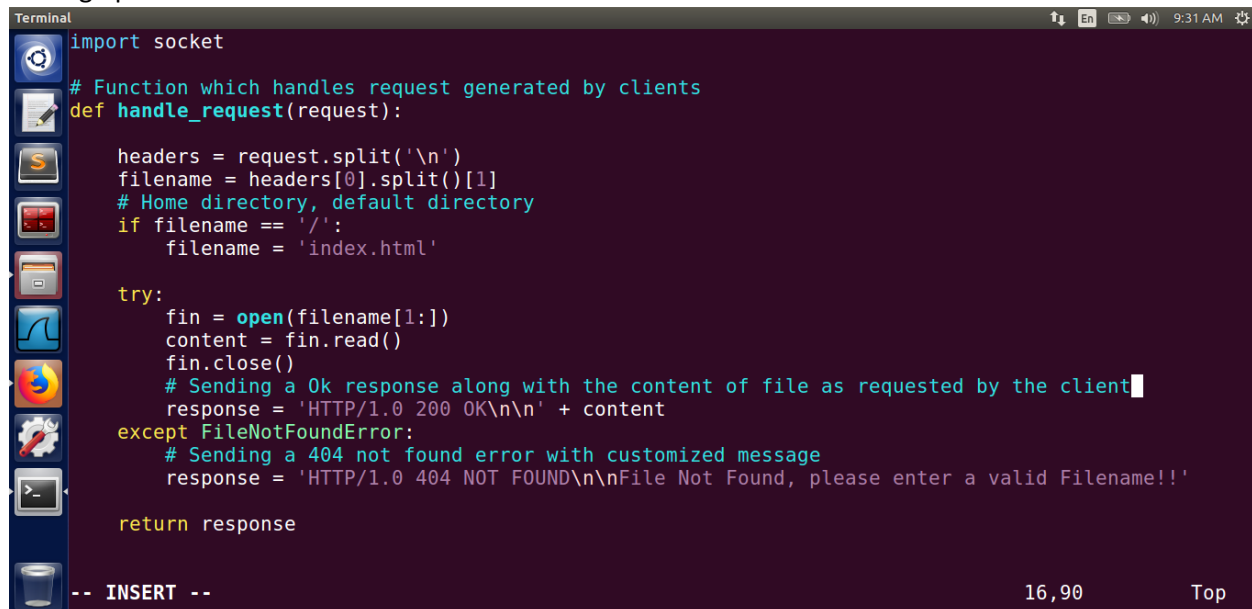
lo:    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      inet6 addr: ::1/128 Scope:Host
      UP LOOPBACK RUNNING  MTU:65536  Metric:1
      RX packets:1639 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1639 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1
      RX bytes:126064 (126.0 KB)  TX bytes:126064 (126.0 KB)

[02/18/22]seed@VM:~$

```

Question-1:

Setting up the Server code:



```

import socket

# Function which handles request generated by clients
def handle_request(request):

    headers = request.split('\n')
    filename = headers[0].split()[1]
    # Home directory, default directory
    if filename == '/':
        filename = 'index.html'

    try:
        fin = open(filename[1:])
        content = fin.read()
        fin.close()
        # Sending a OK response along with the content of file as requested by the client
        response = 'HTTP/1.0 200 OK\n\n' + content
    except FileNotFoundError:
        # Sending a 404 not found error with customized message
        response = 'HTTP/1.0 404 NOT FOUND\n\nFile Not Found, please enter a valid Filename!!'

    return response

-- INSERT --

```

16,90

Top

```

Terminal
# Define socket host and port
SERVER_HOST = "192.168.29.158" # Ip address of the host
SERVER_PORT = 8888 # Port used for communication between server and client

# Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print(server_socket, ' Listening on port %s ...' % SERVER_PORT)

while True:
    # Wait for client connections
    client_connection, client_address = server_socket.accept()
    # Get the client request
    request = client_connection.recv(1024).decode()
    print("Client is connected with Ip:", client_address)
    # Return an HTTP response
    response = handle_request(request)
    client_connection.sendall(response.encode())
    print("Client has disconnected with Ip:", client_address)
    # Close connection
    client_connection.close()

-- INSERT --
48,1 92%

```

Using port number 8888. Usually port number greater than 1024 are not reserved and can be used for our experiment

Running the server:

```

[02/18/22]seed@VM:~$ vim server.py
[02/18/22]seed@VM:~$ python3 server.py
Listening on port 8888 ...

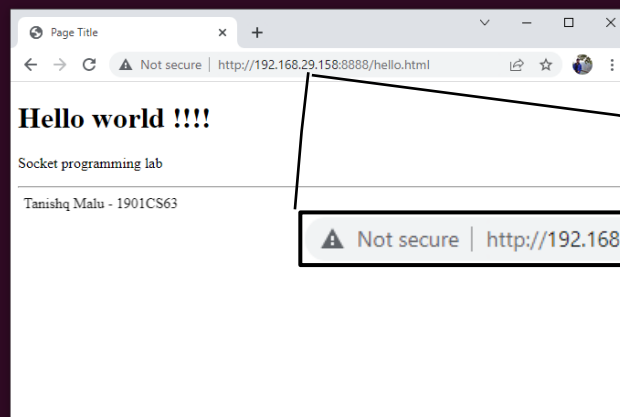
```

```

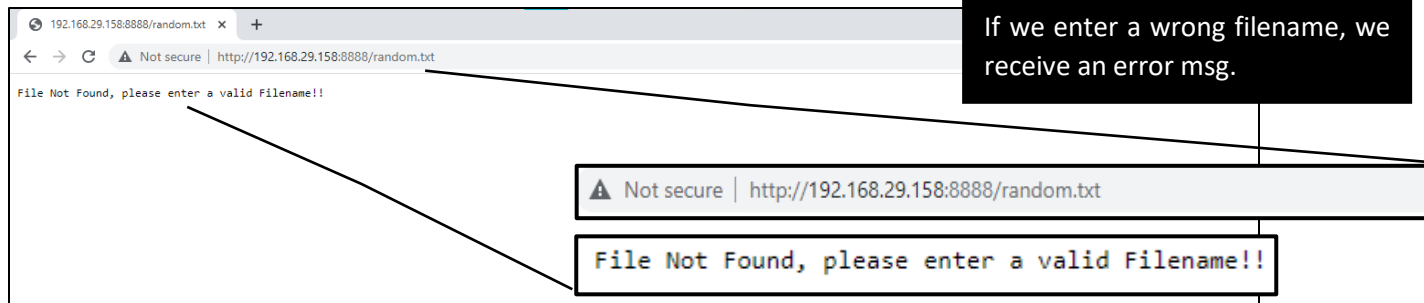
Terminal
[02/18/22]seed@VM:~$ vim server.py
[02/18/22]seed@VM:~$ python3 server.py
Listening on port 8888 ...
Client is connected with Ip: ('192.168.29.138', 64208)
Client has disconnected with Ip: ('192.168.29.138', 64208)

```

If we enter a valid filename, the file gets opened.



Not secure | http://192.168.29.158:8888/hello.html



Question-2:

```
Terminal
import socket
import threading
import _thread
import json
import time

# Reading config file
with open('sample.json', 'r') as openfile:
    # Reading from json file
    config = json.load(openfile)

# Default values, accessed from the config file

# Default filename
default_filename = config["defaultFile"]
# Default maximum concurrent connections allowed
max_concurrent_con = config["max_concurrent_thread"]
# Default IPs which are blocked
block_ip = config["blockIp"]

# Global variable to keep a count
# current number of concurrent connection
current_concurrent_con = 0

-- INSERT --
```

Setting up default values from config file.

```
Terminal
#Function to handle requests generated by clients
def handle_request(request,ip):
    global current_concurrent_con # accessing the global variable
    headers = request.split('\n')
    filename = headers[0].split()[1] # grabbing the filename from the header

    if filename == '/':
        filename = default_filename # redirecting to home directory/default

    try:
        fin = open(filename[1:])
        content = fin.read() # reading the contents of the file
        fin.close()
        # Generating an OK response for client
        response = 'HTTP/1.0 200 OK\n\n' + content
    except FileNotFoundError:
        #generating a 404 error with a customized msg
        response = 'HTTP/1.0 404 NOT FOUND\n\nFile Not Found!!'
    # Blocking the response, if ip is in blocked ip list
    if block_ip.count(ip)>0:
        response = 'HTTP/1.0 404 NOT FOUND\n\nYour Ip is blocked!!'

    if current_concurrent_con > max_concurrent_con:
        response = 'HTTP/1.0 404 NOT FOUND\n\nServer Overload, User limit exceeded !!!'

    -- INSERT --
```

Checking current IP with blocked IP

Checking number of concurrent clients requesting to server

```

Terminal
# Define socket host and port
SERVER_HOST = "192.168.29.158"
SERVER_PORT = 8081

# Create socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print(' Listening on port %s ...' % SERVER_PORT)

```

```

Terminal
# Function to handle multiple requests from different clients
def solve(client_connection, client_address):
    global current_concurrent_con # accessing the global variable
    ip = client_address[0] # Storing the ip address and port number of client
    port = client_address[1]
    # Increment the number of concurrent users
    current_concurrent_con = current_concurrent_con + 1
    print("The new connection was made from IP:", client_address)
    while True:
        request = client_connection.recv(1024).decode()
        response = handle_request(request, ip)
        client_connection.sendall(response.encode())
        print("The client has disconnected, Ip:", client_address)
        break
    # Making the thread sleep so that we can see the effect of max allowed u
    time.sleep(10)
    client_connection.close()
    # decrementing the concurrent users
    current_concurrent_con -= 1

-- INSERT --
83,1 82%

```

Making a thread sleep, so that we can see the effect of maximum allowed users

```

t
while True:
    try:
        # Wait for client connections
        client_connection, client_address = server_socket.accept()
        _thread.start_new_thread(solve, (client_connection, client_address,))
    except KeyboardInterrupt as e:
        print("Server is now closed")
        break
    except Exception as e:
        print("Error ", e)
# Close socket
server_socket.close()

-- INSERT --
94,1 Bot

```

Creating a new thread whenever a new client requests a server

Running the server code:

The terminal window shows the following commands and output:

```
[02/18/22]seed@VM:~$ vim create_config.py
[02/18/22]seed@VM:~$ vim multiclient.py
[02/18/22]seed@VM:~$ python3 multiclient.py
Listening on port 8081 ...
The new connection was made from IP: ('192.168.29.138', 63057)
The client has disconnected, Ip: ('192.168.29.138', 63057)
```

The web browser shows the following content:

This is the default file whose name is fetched from config file

In this server, you can open the following files:

1. hello.html
2. Socket.txt

Type any other filename to see what happens

Although a file name was not given but the server directed it to a default file mentioned in the config file

Not secure | <http://192.168.29.158:8081>

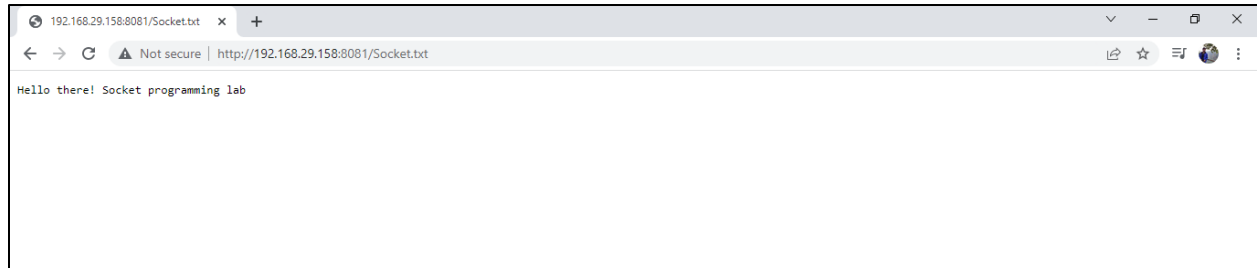
Opening different files in multiple tabs(less than max_concurrent_con)

The web browser shows the following content:

Hello world !!!!

Socket programming lab

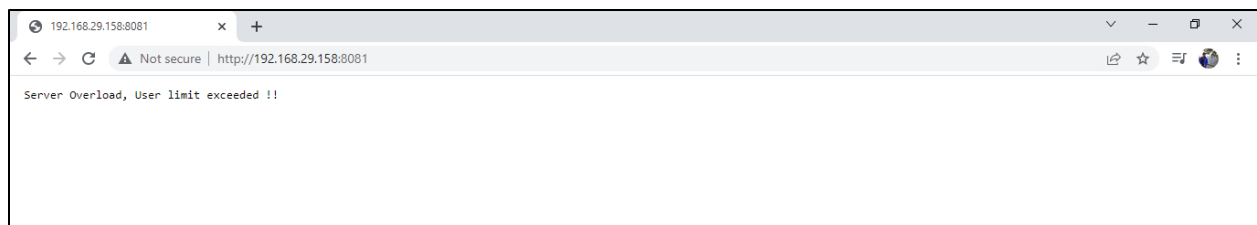
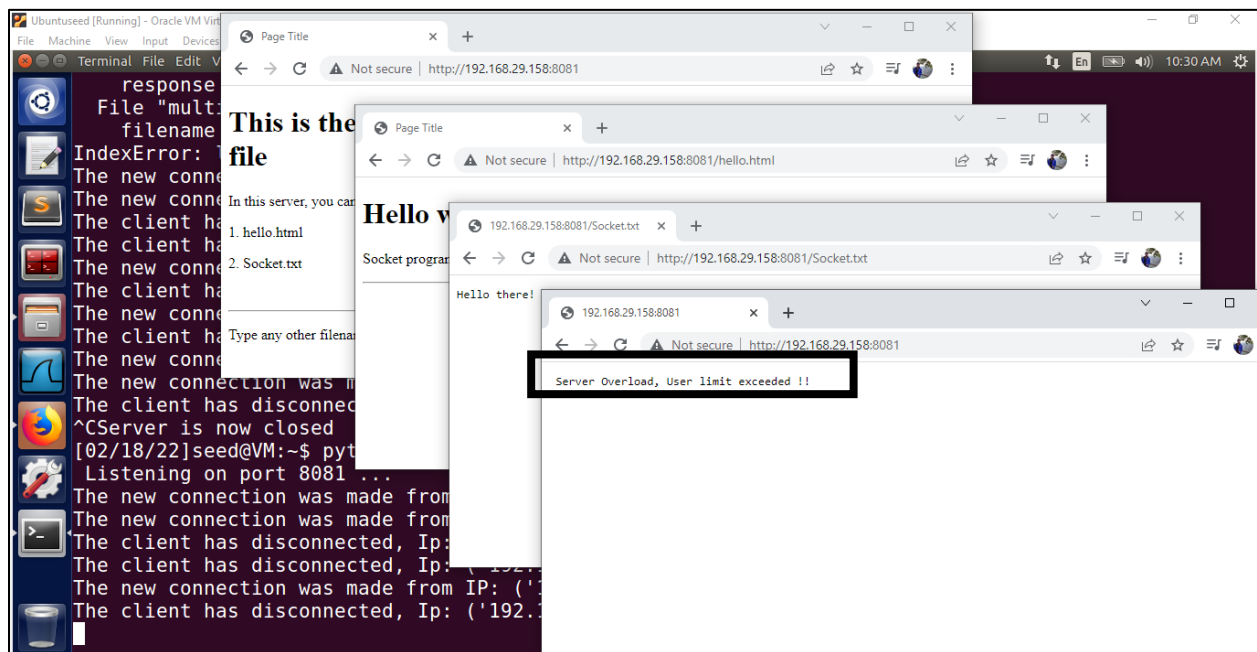
Tanishq Malu - 1901CS63



Max concurrent users allowed = 3 (in my config file)

So, when I try to make 4 concurrent requests

```
The new connection was made from IP: ('192.168.29.138', 63343)
The new connection was made from IP: ('192.168.29.138', 63342)
The client has disconnected, Ip: ('192.168.29.138', 63342)
The client has disconnected, Ip: ('192.168.29.138', 63343)
The new connection was made from IP: ('192.168.29.138', 63346)
The client has disconnected, Ip: ('192.168.29.138', 63346)
The new connection was made from IP: ('192.168.29.138', 63348)
The client has disconnected, Ip: ('192.168.29.138', 63348)
```



If I modify the config file and add my windows machine IP address in blocked IP, I will not be allowed to access the file.

Getting my ip address of windows machine:

```
wifi0: flags=4096<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.29.138 netmask 255.255.255.0 broadcast 192.168.29.255
    inet6 fe80::7c45:3e86:5ca:bb33 prefixlen 64 scopeid 0x0<global>
    inet6 2405:201:3006:f006:15cd:c740:f0d1:e787 prefixlen 128 scopeid 0x0<global>
    inet6 fe80::7c45:3e86:5ca:bb33 prefixlen 64 scopeid 0xfd<compat,link,site,host>
    ether b0:52:16:a0:87:af (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/Network_lab/lab-3$ ifconfig
```

Creating the config json file:

```
Terminal
import json

# a Python object (dict):
x = {
    "max_concurrent_thread": 3,
    "blockIp": ['192.168.29.138', '192.168.30.133'],
    "defaultFile": "/index.html"
}

print(x)
with open("sample.json", "w") as outfile:
    json.dump(x, outfile)

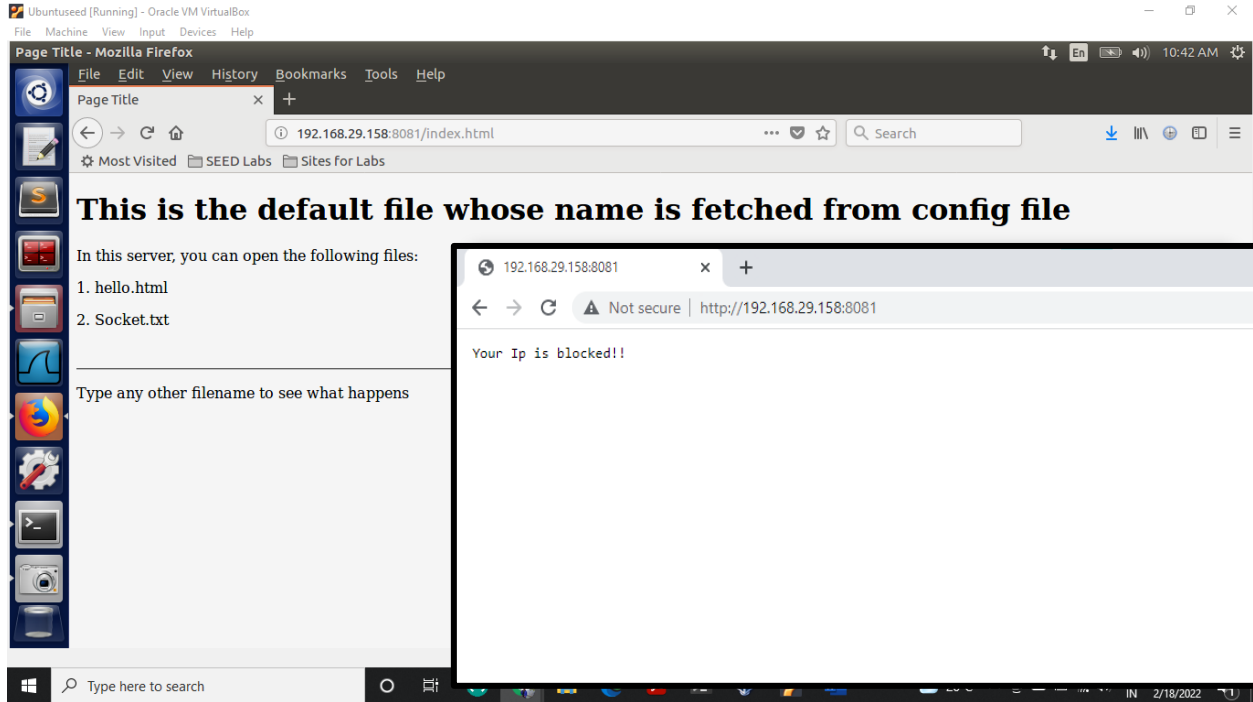
-- INSERT --
13,26 All
```

Now on running the server again, I can't access the server from my windows machine but can access it via virtual machine because its ip address is not mentioned in the blocked ip list.

```
[02/18/22]seed@VM:~$ vim create_config.py
[02/18/22]seed@VM:~$ python3 multiclient.py
Listening on port 8081 ...
The new connection was made from IP: ('192.168.29.158', 46330)
The client has disconnected, Ip: ('192.168.29.158', 46330)
The new connection was made from IP: ('192.168.29.158', 46332)
The client has disconnected, Ip: ('192.168.29.158', 46332)
The new connection was made from IP: ('192.168.29.158', 46334)
The client has disconnected, Ip: ('192.168.29.158', 46334)
The new connection was made from IP: ('192.168.29.138', 63814)
The new connection was made from IP: ('192.168.29.138', 63815)
The client has disconnected, Ip: ('192.168.29.138', 63815)
```

IP of Virtual machine client

IP of windows machine client



-----End Of Assignment-----