**Case Study Submission Part-1**

## **Title: DISTRICT SURAKSHA**

|  |  |  |
| --- | --- | --- |
| RollNo | Name | Contribution in case study |
| CB.EN. U4CSE19105 | A V V L N BALRAM | RECEIVING DEPARTMENT |
| CB.EN. U4CSE19137 | P.S.V. AKASH | VERIFICATION DEPARTMENT |
| **CB.EN. U4CSE19154** | **T SAI JAYANTH** | **EXECUTION DEPARTMENT** |

**Description of the case study:**

As the time grows, we need to develop our standards there one used to be a box for complaints earlier. So, why don’t we transform that into digital. We are introducing citizens problem reporting center where three departments are available Namely Receiving, Verification, Execution. Here Receiving section will receive the complaints, Verification section will verify the complaints, Execute section will execute the complaints which are verified by the verification department.

**Why is Networking required?**

Here we use networks in order to communicate between multiple departments. Also, execution department will communicate with various other offices in order to execute the verified problems. So, we need networking here for Receiving, Verification, Execution.

**List of Network performance parameters:**

Normally the performance of a network is used to measure the service quality of a network as perceived by the user. To measure the performance of a network there are different ways depending on design of the network.

|  |  |  |
| --- | --- | --- |
| Parameters | Definition | Formula |
| Bandwidth | The maximum amount of data transmitted over an internet connection in a given amount of time. It determines how rapidly the web server is able to upload the requested information.  Digital Devices: bps(bytes per second)  Analog Devices: cps(cycles per second)  It is a potential measurement of a link. | Expressed as [bits](https://web.archive.org/web/20190816003233/https:/whatis.techtarget.com/definition/bit-binary-digit) per second ([bps](https://web.archive.org/web/20190816003233/https:/searchnetworking.techtarget.com/definition/bits-per-second)), modern network links have greater capacity, which is typically measured in millions of bits per second ([megabits per second](https://web.archive.org/web/20190816003233/https:/searchnetworking.techtarget.com/definition/Mbps), or Mbps) or billions of bits per second ([gigabits per second](https://web.archive.org/web/20190816003233/https:/whatis.techtarget.com/definition/Gbps-billions-of-bits-per-second), or Gbps). |
| Throughput | The number of messages successfully transmitted per unit time. The maximum throughput of a network may be consequently higher than the actual throughput achieved in every day consumption.  It is an actual measurement of how fast we can send the data. | R = I/T  R : Rate(Throughput)  I : Inventory  T : Time |
| Transmission time | The time required for transmission of a message depends on the size of the message and the bandwidth of the channel. | Transmission time=Message size / Bandwidth |
| Propagation Time | Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed. | Propagation time = Distance /Propagation speed |
| Processing Delay | Time taken by the processor to process the data packet is called processing delay. |  |
| Queuing Delay | Time spent by the data packet waiting in the queue before it is taken for execution is called queuing delay. |  |
| Packet Loss | Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Due to network congestion | Efficiency = 100% \* (transferred - retransmitted) / transferred  Network Loss = 100 - Efficiency |
| Latency | The time required to successfully send a packet across the network.  The total time taken for a complete message to arrive at the destination, starting with the time when the first bit of the message is sent out from the source and ending with the time when the last bit of the message is delivered at the destination. Here Latency is also known as ping rate and measured in milliseconds. | Latency = Propagation Time + Transmission Time + Queuing Time + Processing Delay.  Propagation Time = Distance / Propagation Speed |
| Jitter | Jitter is nothing but Packet delay Variance. The variation in the delay of received packets. It is considered as a problem when different packets of data face different delays in a network and the data at the receiver application is time sensitive i.e., audio or video data. It is measured in MilliSeconds(ms) | Latency=sum of all delays  To measure Jitter, we take the difference between samples, and then divide by the number of samples. |

1. Description of the text file:

In text file 6 columns are there namely Name, Phone\_no, District, Pincode, Area, Problem. Each column is important to validate problems.

1. List of operations completed with the file

V-VIEW

F - FILTER

U - UPDATE

M - MODIFY

1. **Server-Side** program with output

import socket

import os

from \_thread import \*

import pandas as pd

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

ThreadCount = 0

try:

connection.bind((socket.gethostname(), 4751))

except socket.error as e:

print(str(e))

print('WAITING FOR AN CONNECTION..')

connection.listen(5)

col\_names = ["Name","Phone\_No","District","Pincode","Area","Problem","Verification","Execution\_Status"]

filename = "problem.csv"

df = pd.read\_csv(filename)

df.columns = col\_names

def showData(df,column\_name,value):

g = df.groupby(column\_name)

return g.get\_group(value)

def getUserData(df,Phone\_No):

df = df.iloc[Phone\_No]

print(df)

return df

def modifyData(df, Phone\_No, new\_value):

for index in df.index:

if df.loc[index,"Phone\_No"] == Phone\_No :

df.loc[index,"Execution\_Status"] = new\_value

return df

def update(df):

df["Execution\_Status"] = df["Execution\_Status"].fillna("NotExecuted")

return df

def delete(df):

for index in df.index:

if df.loc[index,"Verification"] == "Not Verified":

df.drop([index],axis=0,inplace=True)

return df

def threaded\_client(connection):

while True:

data = connection.recv(4048).decode('utf-8')

if not data:

break

data = str(data)

print("From Connected User : "+data)

if data == "V":

show\_data = df.to\_string()

connection.send(show\_data.encode())

elif data.find("M") != -1:

split\_data = data.split()

show\_data = modifyData(df,split\_data[1],split\_data[2]).to\_string()

connection.send(show\_data.encode())

df.to\_csv("problem.csv",index=False)

elif data.find("U") != -1:

show\_data = update(df).to\_string()

connection.send(show\_data.encode())

df.to\_csv("problem.csv",index=False)

elif data.find("F") != -1:

show\_data = delete(df).to\_string()

connection.send(show\_data.encode())

df.to\_csv("Executedlist.csv",index=False)

connection.close()

while True:

clt, adr = connection.accept()

print(f"Connection established to {adr} established")

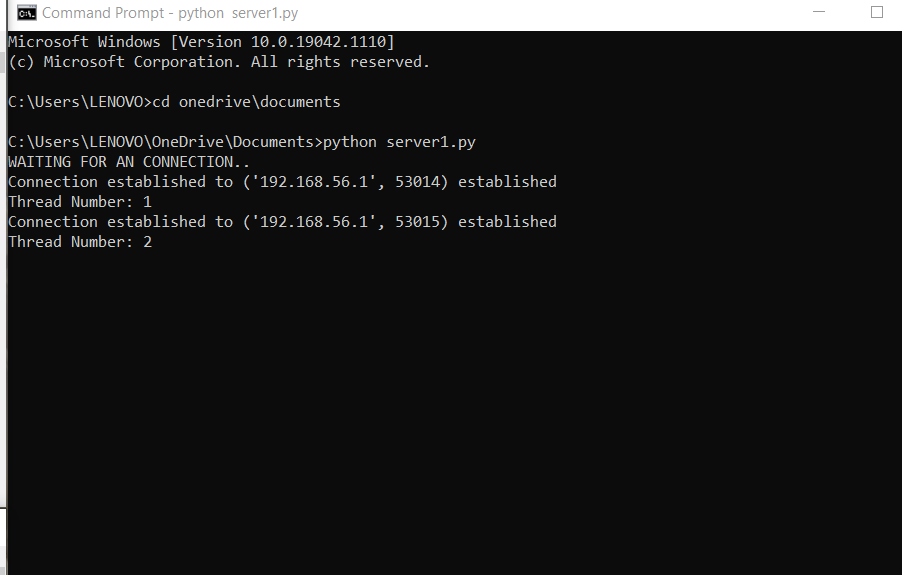
start\_new\_thread(threaded\_client, (clt,))

ThreadCount += 1

print('Thread Number: ' + str(ThreadCount))

connection.close()

**OUTPUT:**

****

1. **Client-Side** program with output for each operation:

import socket

def client\_program():

print("V - VIEW")

print("F - FILTER")

print("U - UPDATE")

print("M - MODIFY")

print("exit")

client\_socket = socket.socket()

host = socket.gethostname()

port = 4751

print('Waiting for connection response')

try:

client\_socket.connect((host,port))

except socket.error as e:

print(str(e))

result = input(" => ")

while result.lower().strip()!="exit":

client\_socket.send(result.encode())

data = client\_socket.recv(4048).decode()

print("RECEIVED FROM SERVER : \n"+ data)

result = input('=>')

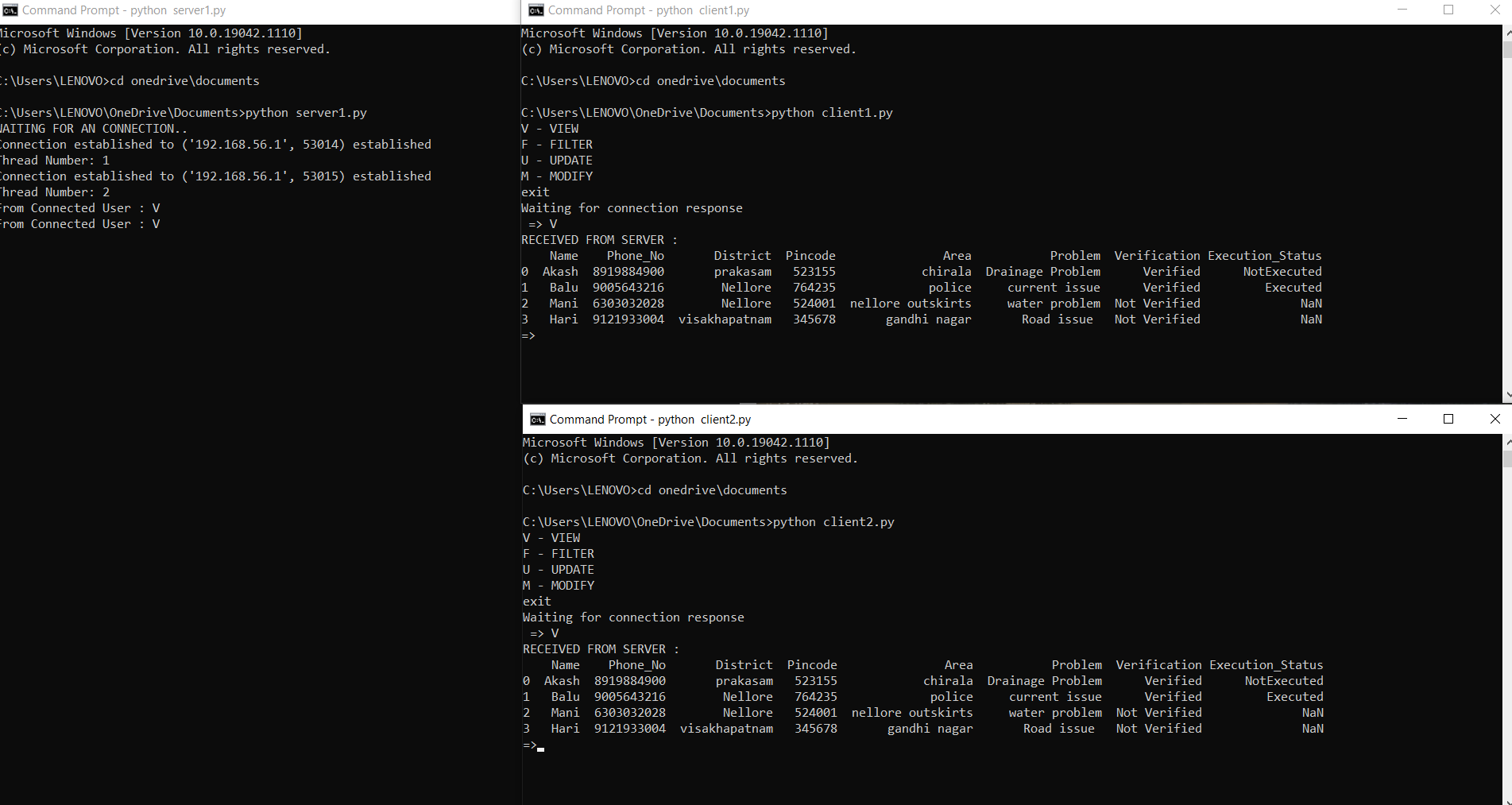
client\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

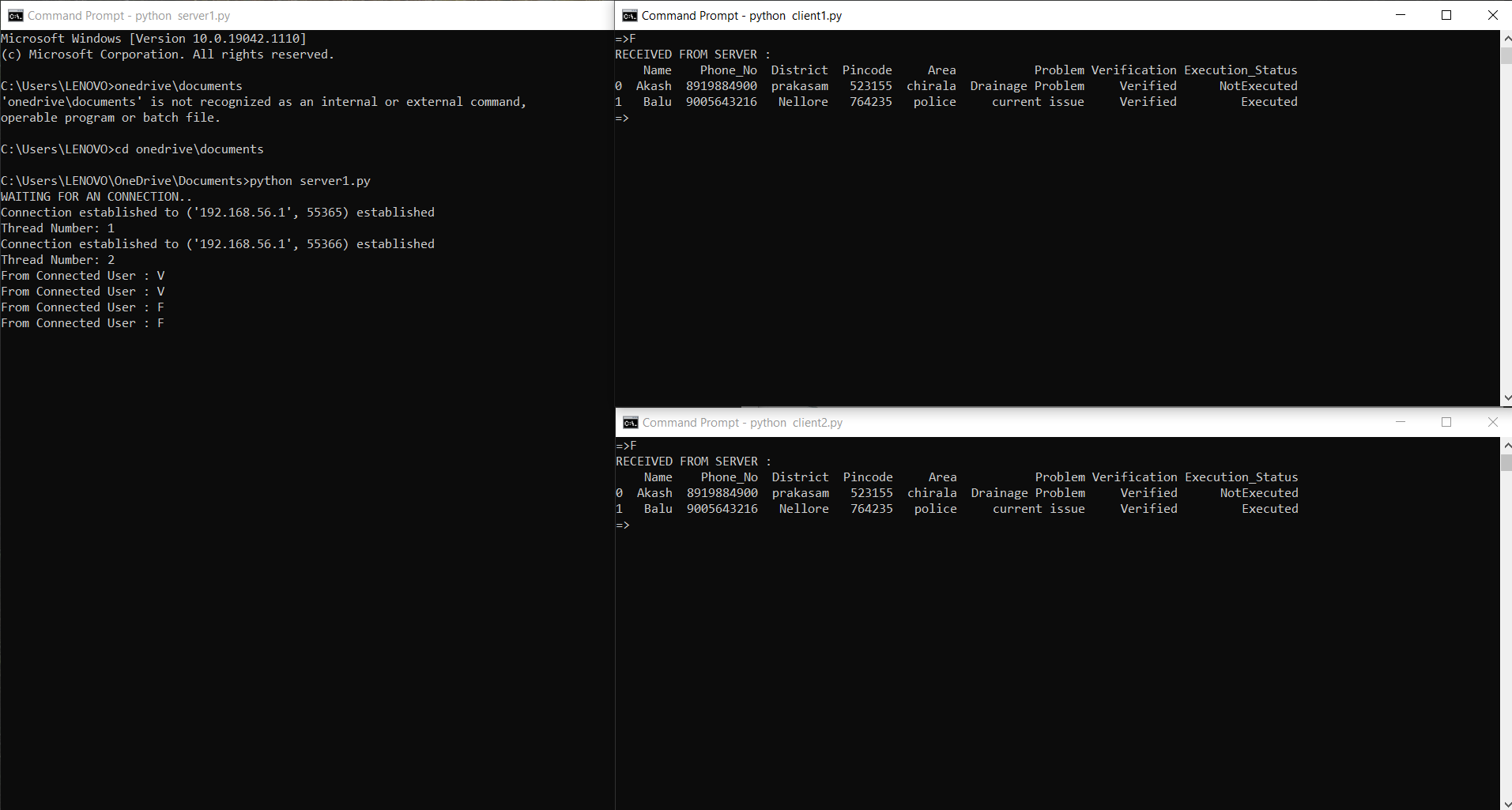
client\_program()

**OUTPUT:**

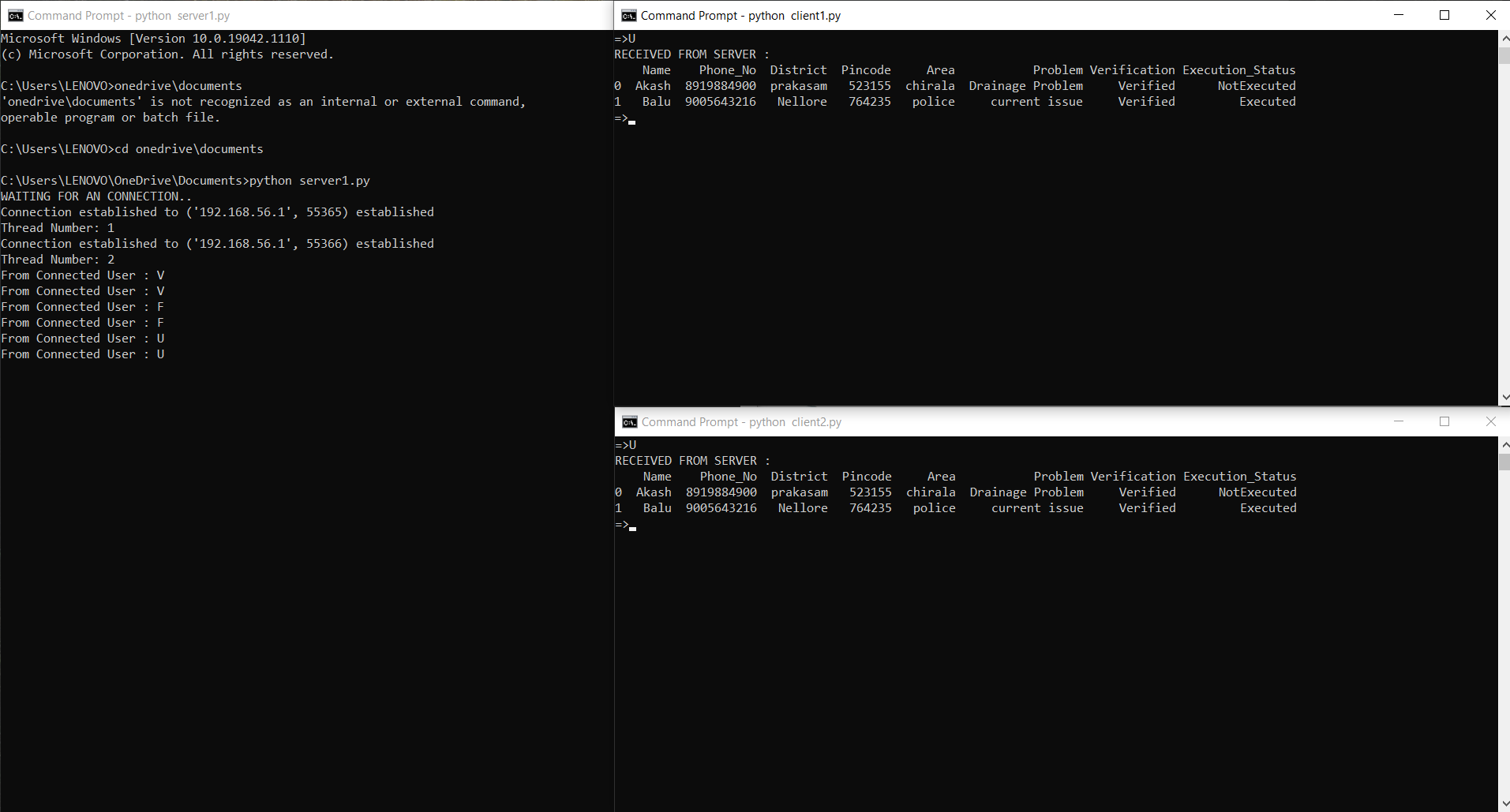
**VIEW:**

****

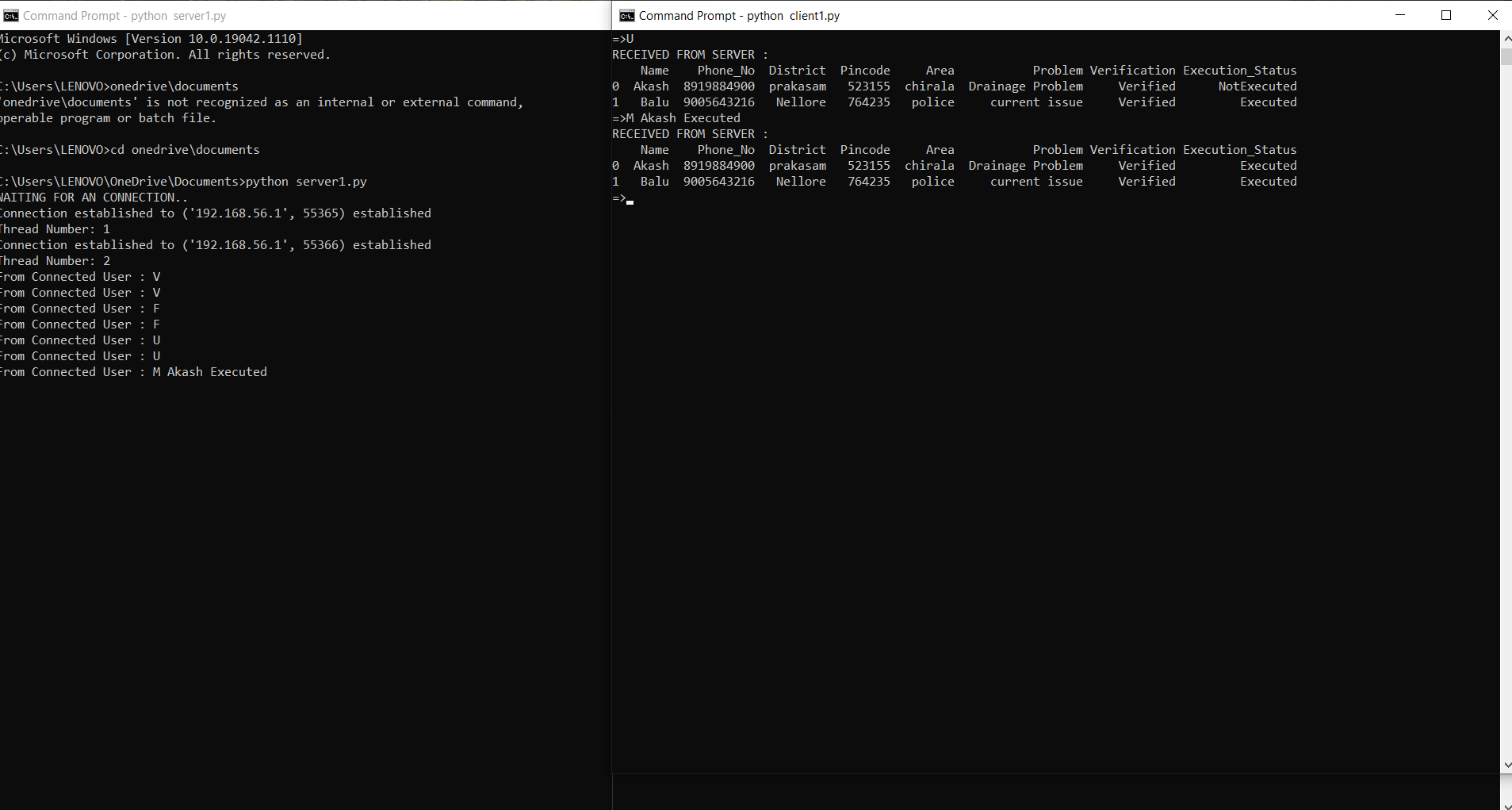
**FILTER:**

****

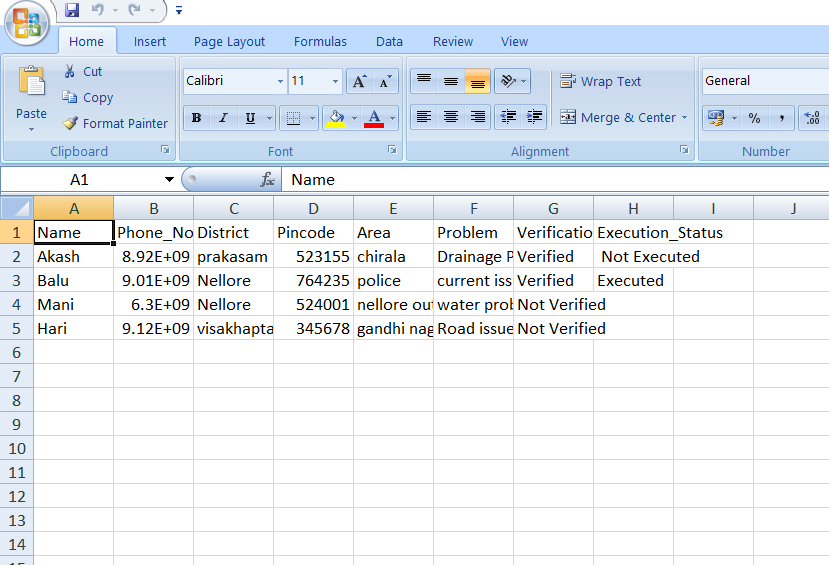
**UPDATE:**

****

**MODFIY:**

****

**CSV FILE:**

****