**NAME :- MANAN MILAN HEMANI**

**ID:- 191001001132**

**STREAM:- B.TECH CSE**

**BATCH:- BCS 3B GROUP A**

**SUBJECT:- COMPUTER NETWORKS ASSIGNMENT(AVIJIT SIR)**

**1) To Write a Socket program to implement echo**

**Ans)**

**Server Code (server.py)**

import socket

HOST = "127.0.0.1" # or use socket.gethostbyname(socket.gethostname())

PORT = 5050

ADDR = (HOST, PORT)

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.bind(ADDR)

print("[STARTING] Server is starting...")

s.listen()

conn, addr = s.accept()

with conn:

print(f"Connected by {addr}")

**Client Code (client.py)**

import socket

HOST = "127.0.0.1"

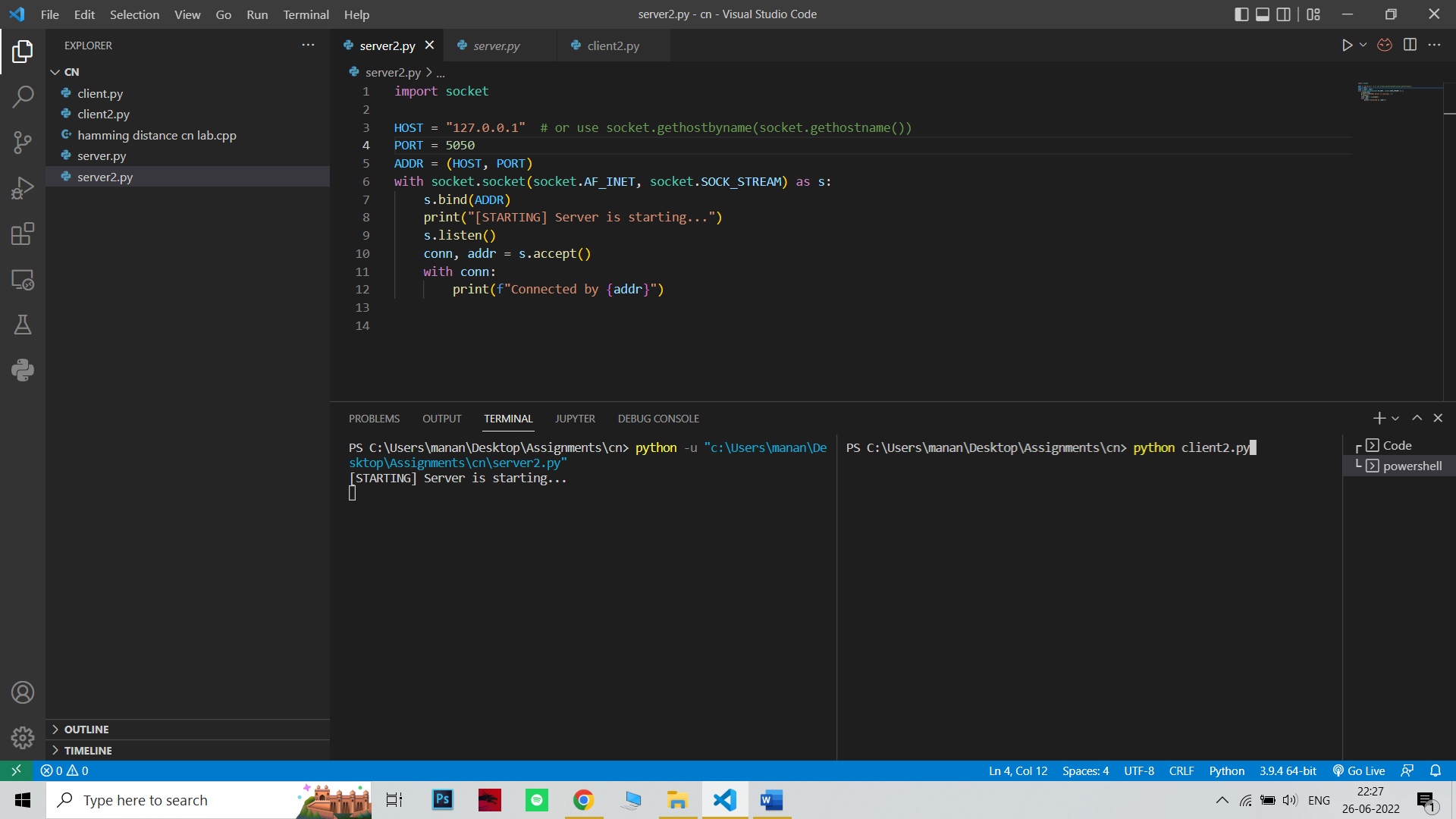
PORT = 5050

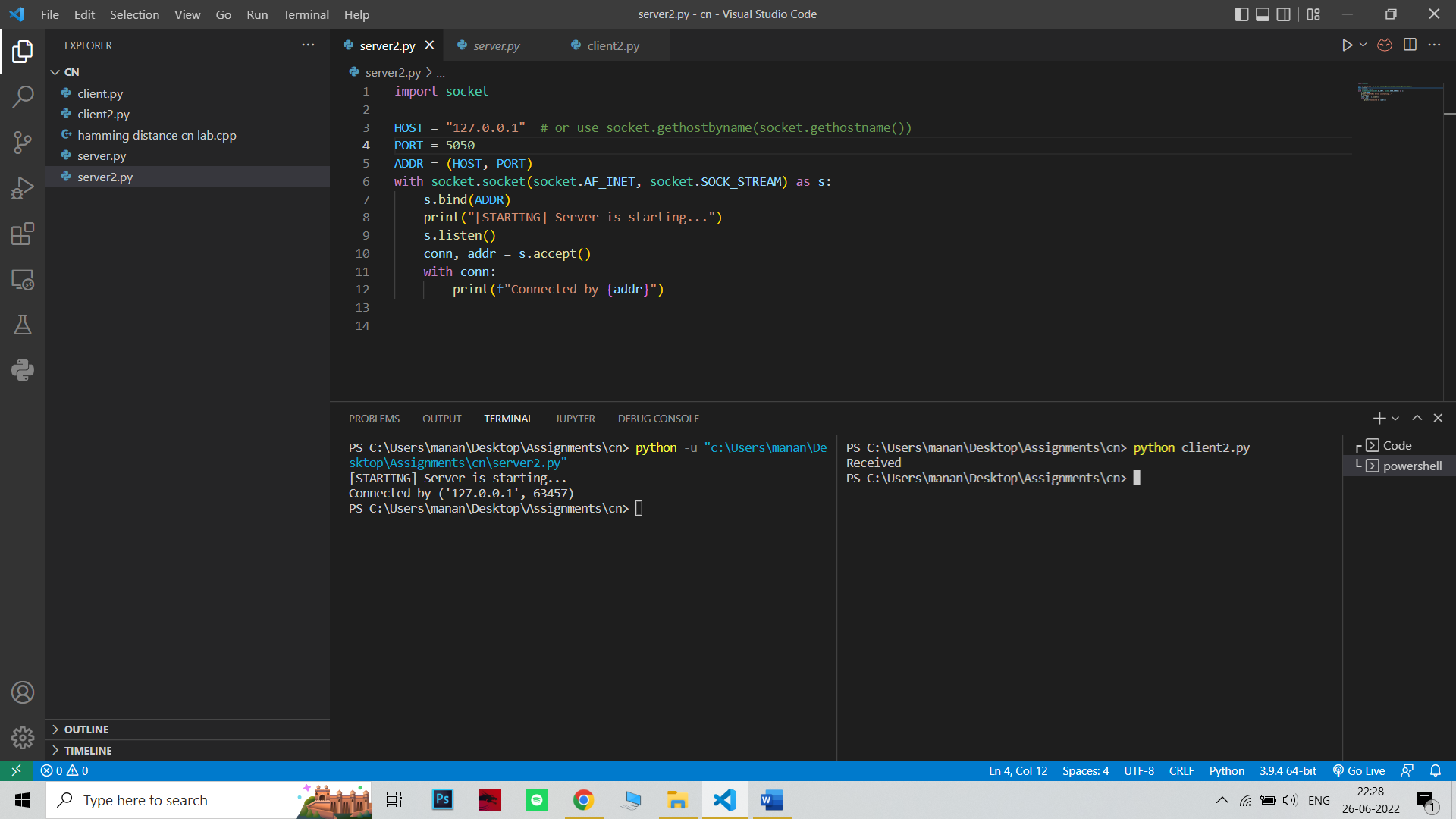
with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.connect((HOST, PORT))

print(f"Received ")

**OUTPUT**





**2) Calculating the hamming distance and return the appropriate data block and the codeword that has the minimum hamming distance**

**Ans)**

**Code:-**

#include <bits/stdc++.h>

using namespace std;

int hamming(string s1 , string s2)

{

int count=0;

for(int i=0;i<s1.size();i++)

{

if(s1[i]^s2[i])

count++;

}

return count;

}

string solve(vector<pair<string,string>> v, string s)

{

int mini=INT\_MAX;

string ans;

for(auto a:v)

{

int c=min(mini,hamming(a.second,s));

if(c!=mini)

{

mini=c;

ans=a.first;

} }

return ans;

}

int main()

{

vector<pair<string,string>> v={{"00","00000"},{"01","00111"},{"10","11001"},{"11","11110"}};

string s="00110";

cout<<"The codeword that has the minimum hamming distance is "<<solve(v,s); return 0;

}

**OUTPUT**

The codeword that has the minimum hamming distance is 01

**3) Implement the parity code. Consider the length of the data is ‘n’. Single bit will be parity bit. You can implement even or odd parity. You need to develop 2 functions sender (), receiver ()**

**Ans)**

**CODE: -**

#include <bits/stdc++.h>

using namespace std;

bool calc\_parity (string data) {

bool parity = true; //taking even parity as true

for(int i = 0; i<data.length(); i++) {

if(data[i] == '1') {

parity = !parity;

}

}

return parity;

}

string sender(string data) {

if(calc\_parity(data))

data += '0';

else

data += '1';

return data;

}

void receiver(string data) {

if(calc\_parity(data))

cout<<"Data is correct!";

else

cout<<"Data is incorrect!";

}

int main () {

string sendData = "", recdData = "";

cout<<"Enter data to send: ";

cin>>sendData;

cout<<"Data with adjusted parity: "<<sender(sendData)<<endl;

cout<<"\nEnter data received: ";

cin>>recdData;

receiver(recdData);

return 0;

}

**OUTPUT**

