PES University

UE21CS252B – Computer Networks

CN-MINI PROJECT 1

Session: Jan-May 2023

Semester & Section: 4th, I

Sl No.	Name of the Student	SRN
1.	Shweta Dash	PES2UG21CS519
2.	Smriti Sugur	PES2UG21CS531

Title:

Online Chat Room Using Python Socket Programming

Abstract:

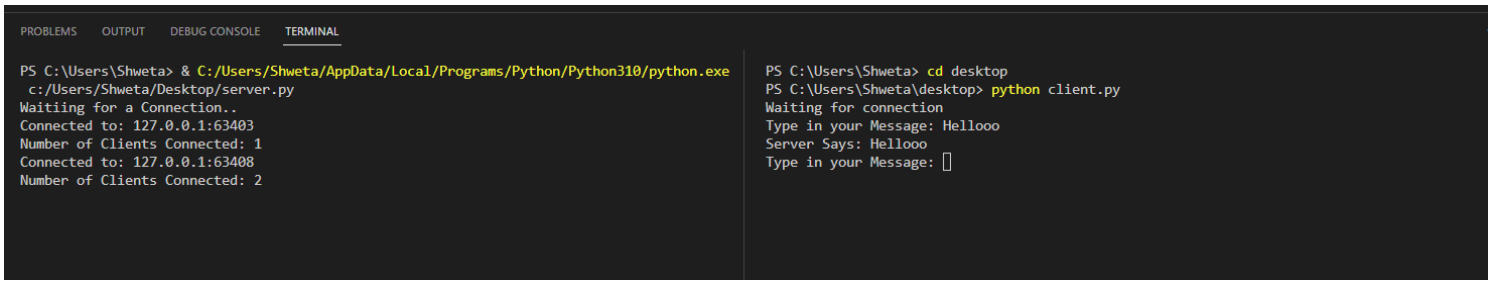
This project aims to replicate a chatroom server and allow multiple clients to connect to it using a client-side script. It is mainly coded using Python socket programming.

Initially, a server code is scripted using three Python modules- socket, os and _thread. The socket module provides functions to initialize a server-side socket, and bind a host and port to the socket. Meanwhile, the os and _thread modules are useful for the connection of multiple clients, by providing functions for multi-threading.

For the client-side code, socket module is used to create a client socket. A host ip address and port number are assigned to the client, and these are assigned to the socket using the bind() function. Next, a connection is set up using connect() of the socket library which establishes a connection with the server using the host and port that is provided. An input option is provided to the client so that it has the capability to send data back to the server, and alongside that, the recv() function is used to receive data from Server Side.

The chatroom has been tested with multiple clients which have been connected to the server.

Output:



Code:

1. server.py

```
import socket
import os
from _thread import * # _thread for python3

ServerSocket = socket.socket() #creates a socket
host = '127.0.0.1' #declaration of host and port
port = 1233
ThreadCount = 0 #initial thread count
try:
    ServerSocket.bind((host, port)) #binding host and
port to socket
except socket.error as e:
    print(str(e))
```

```
print('Waitiing for a Connection..')
ServerSocket.listen(5)


def threaded_client(connection): #connects to each
client at diff addresses
    connection.send(str.encode('Welcome to the
Server'))
    while True:
        data = connection.recv(2048) #to get data from
a client
        reply = 'Server Says: ' + data.decode('utf-8')
        if not data:
            break
        connection.sendall(str.encode(reply))
    connection.close()

while True: #to run the server until we stop it
    Client, address = ServerSocket.accept()
    print('Connected to: ' + address[0] + ':' +
str(address[1]))
    start_new_thread(threaded_client, (Client, ))
#assigns new thread to every client that connects
    ThreadCount += 1
    print('Number of Clients Connected: ' +
str(ThreadCount))
ServerSocket.close()
```

2. client.py

```
import socket

ClientSocket = socket.socket()
host = '127.0.0.1'
port = 1233

print('Waiting for connection')
try:
    ClientSocket.connect((host, port))
except socket.error as e:
    print(str(e))

Response = ClientSocket.recv(1024)
while True:
    Input = input('Type in your Message: ')
    ClientSocket.send(str.encode(Input))
    Response = ClientSocket.recv(1024)
    print(Response.decode('utf-8'))

ClientSocket.close()
```