**Experiment 4**

**Roll No.**

**Aim:** To implement a chat server using Socket Programming.

**Program:**

**Server Side:**

import socket

import select

import sys

from thread import \*

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

server.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

IP\_address = "200.198.15.15"

Port = 8081

server.bind((IP\_address, Port))

server.listen(100)

list\_of\_clients = []

def clientthread(conn, addr):

conn.send("Welcome to this chatroom!")

while True:

try:

message = conn.recv(2048)

if message:

print "<" + addr[0] + "> " + message

message\_to\_send = "<" + addr[0] + "> " + message

broadcast(message\_to\_send, conn)

else:

remove(conn)

except:

continue

def serverthread(conn, addr):

while True:

try:

message = "<Server> " + sys.stdin.readline()

sys.stdout.flush()

for clients in list\_of\_clients:

try:

clients.send(message)

except:

clients.close()

remove(clients)

print message

except:

continue

def broadcast(message, connection):

for clients in list\_of\_clients:

if clients!=connection:

try:

clients.send(message)

except:

clients.close()

remove(clients)

def remove(connection):

if connection in list\_of\_clients:

list\_of\_clients.remove(connection)

while True:

conn, addr = server.accept()

list\_of\_clients.append(conn)

print addr[0] + " connected"

start\_new\_thread(clientthread,(conn,addr))

start\_new\_thread(serverthread,(conn,addr))

conn.close()

server.close()

**Client Side:**

import socket

import select

import sys

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

IP\_address = "200.198.15.15"

Port = 8081

server.connect((IP\_address, Port))

while True:

sockets\_list = [sys.stdin, server]

read\_sockets,write\_socket, error\_socket = select.select(sockets\_list,[],[])

for socks in read\_sockets:

if socks == server:

message = socks.recv(2048)

print message

else:

message = sys.stdin.readline()

server.send(message)

sys.stdout.write("<You>")

sys.stdout.write(message)

sys.stdout.flush()

server.close()

**Output:**

**Server.py**

ss@starwars:~$ cd Desktop

ss@starwars:~/Desktop$ python2 chat\_server.py

200.169.2.210 connected

<200.169.2.210> hello

<200.169.2.210> this is computer Engineering

<200.169.2.210> students

195.168.2.9 connected

< 195.168.2.9> heyya

< 195.168.2.9> hope everyone is good

hello to all

<Server> hello to all

hope all are fine

<Server> hope all are fine

enjoyed talking to u all

<Server> enjoyed talking to u all

**Client1.py**

pp@sss:~$ cd Desktop

pp@sss:~/Desktop$ python2 client.py

Welcome to this chatroom!

hello

<You>hello

this is computer Engineering

<You>this is computer Engineering

students

<You>students

< 195.168.2.9> heyya

< 195.168.2.9>hope everyone is good

<Server> hello to all

<Server> hope all are fine

<Server> enjoyed talking to u all

**Client2.py**

anton@gidion:~$ cd Desktop

anton@gidion:~/Desktop$ python2 client.py

Welcome to this chatroom!

heyya

<You>heyya

hope everyone is good

<You>hope everyone is good

<Server> hello to all

<Server> hope all are fine

<Server> enjoyed talking to u all

**Conclusion:** Thus, we have successfully implemented a chat server using Socket Programming.