**OPEN SOURCE LAB**

**Roll No: 18bit170 & 17bce118**

**Project Title: Server-Client chatbot**

**Description:**

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server. The server-client chatbot made here is made using socket programming and runs on two command prompts, one acting as the server and the other acting as a client.

**Server chatbot code:**

use threads;

use IO::Socket::INET;

$| = 1;

$port = 5000; # Define you port

$socket = IO::Socket::INET->new(

LocalHost => '127.0.0.1', # Define you server's ip. Now its set to localhost.

LocalPort => $port,

Proto => 'tcp',

Listen => 5,

Reuse => 1,

) or die "Coudn't open socket port $port \n" unless $socket;

print "Socket started @ $port \n";

sub handle\_connection {

$request = $\_[0];

$peer\_address = $request->peerhost();

$peer\_port = $request->peerport();

print "New Client Connection from IP : $peer\_address, PORT : $peer\_port\n ";

while(1){

$request->recv($receiveData,1024);

if($receiveData =~ "connectionclose"){

last;

}

print "$peer\_address saying : ".$receiveData;

print "your msg:";

$yourMsg = <STDIN>;

if($yourMsg){

$request->send($yourMsg);

}

}

close($request);

threads->exit();

}

while (my $request = $socket->accept) {

async(\&handle\_connection, $request)->detach;

}

**Client chatbot code:**

use IO::Socket::INET;

# auto-flush on socket

$| = 1;

# create a connecting socket

my $socket = new IO::Socket::INET (

PeerHost => '127.0.0.1',

PeerPort => '5000',

Proto => 'tcp',

);

die "cannot connect to the server $!\n" unless $socket;

print "connected to the server\n";

print "your msg :";

$yourMsg = <STDIN>;

$socket->send($yourMsg);

while(1){

$socket->recv($response,1024);

if($response){

print "Webserver saying :".$response;

}

print "your msg :";

$msg = <STDIN>;

if($msg){

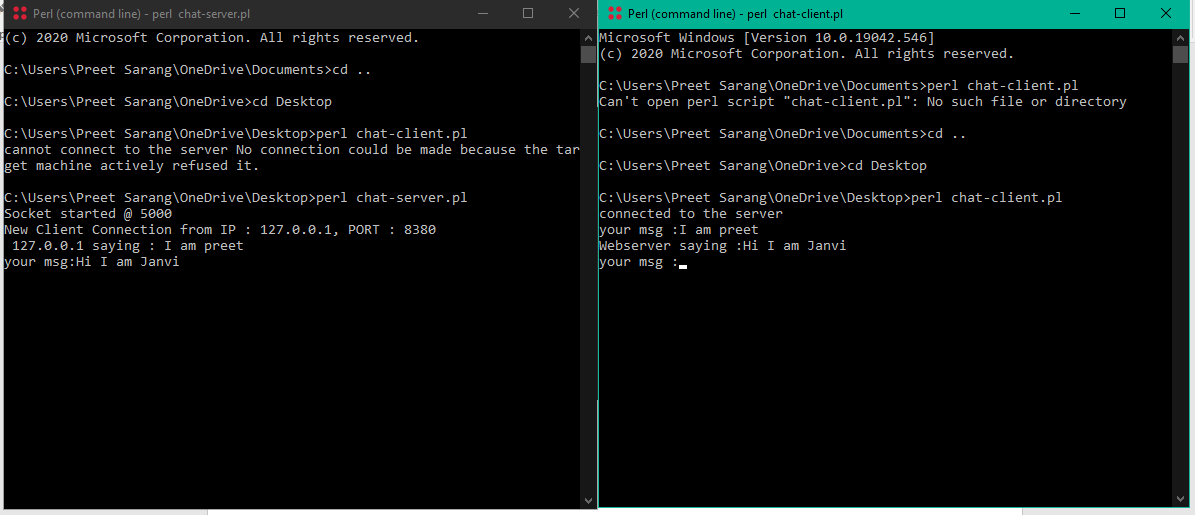
$socket->send($msg);

}

}

$socket->close();

**Output:**

****