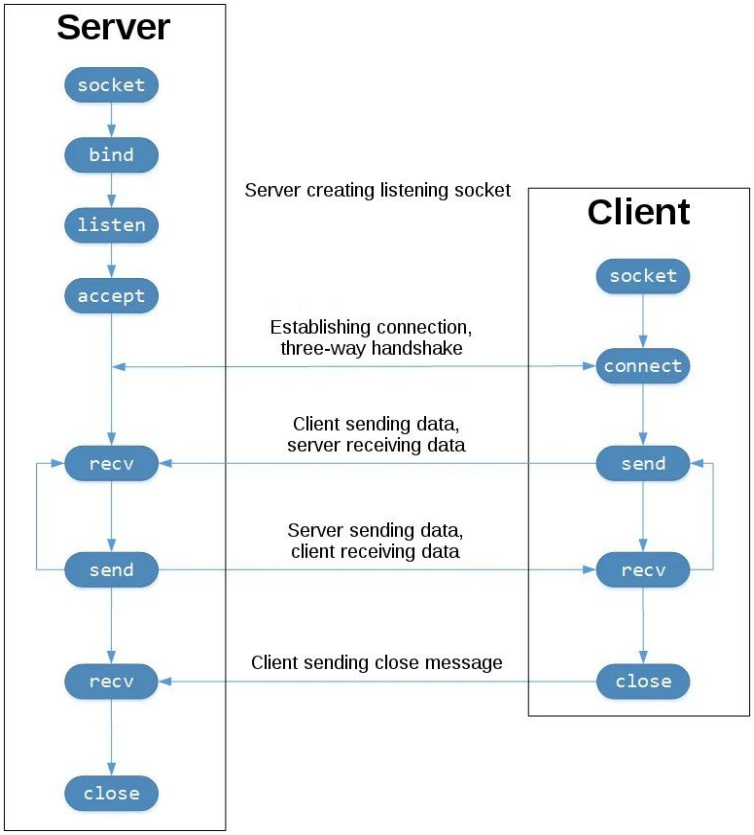
PRADNYA TOPALE
TE COMPS
2018130057

SOCKET PROGRAMMING

AIM: To implement socket programming and establish a connection between client and server.

THEORY: 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 another socket reaches out to the other to form a connection. Server forms the listener socket while the client reaches out to the server.



Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. The *socket* library provides specific classes for handling the common transports as well as a generic interface for handling the rest.

The primary socket API functions and methods in this module are:

- socket()
- bind()
- listen()
- accept()
- connect()
- connect_ex()
- send()
- recv()
- close()

IMPLEMENTATION: To write Internet servers, we use the socket function available in the socket module to create a socket object. A socket object is then used to call other functions to set up a socket server.

Now call bind(hostname, port) function to specify a *port* for your service on the given host.Next, call the *accept* method of the returned object. This method waits until a client connects to the port you specified, and then returns a *connection* object that represents the connection to that client.

Server.py

```
import socket                # Import socket module

s = socket.socket()          # Create a socket object
host = socket.gethostname()  # Get local machine name
port = 12345                 # Reserve a port for your service.
s.bind((host, port))         # Bind to the port

s.listen(5)                  # Now wait for client connection.
while True:
   c, addr = s.accept()      # Establish connection with client.
   print ("Got connection from", addr)
   c.send(bytes("Thank you for connecting","utf-8"))
   c.close()                 # Close the connection
```

The socket.connect(hostname, port) opens a TCP connection to *hostname* on the *port*. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file.

The following code is a very simple client that connects to a given host and port, reads any available data from the socket, and then exits

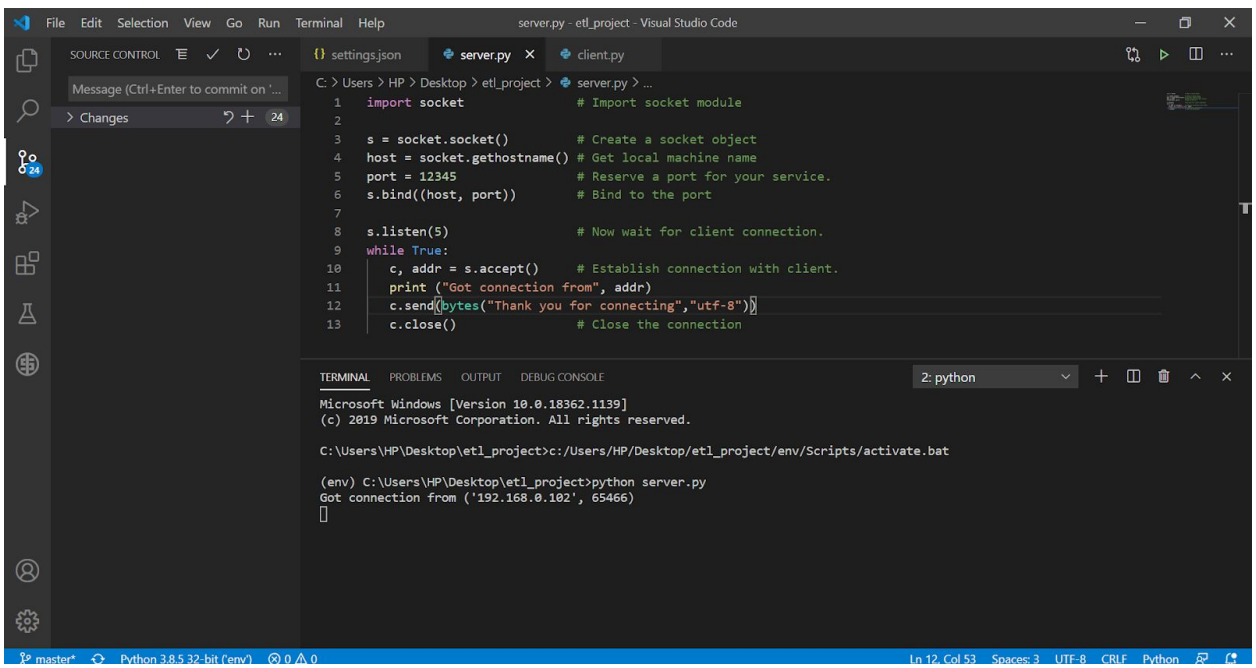
Client.py

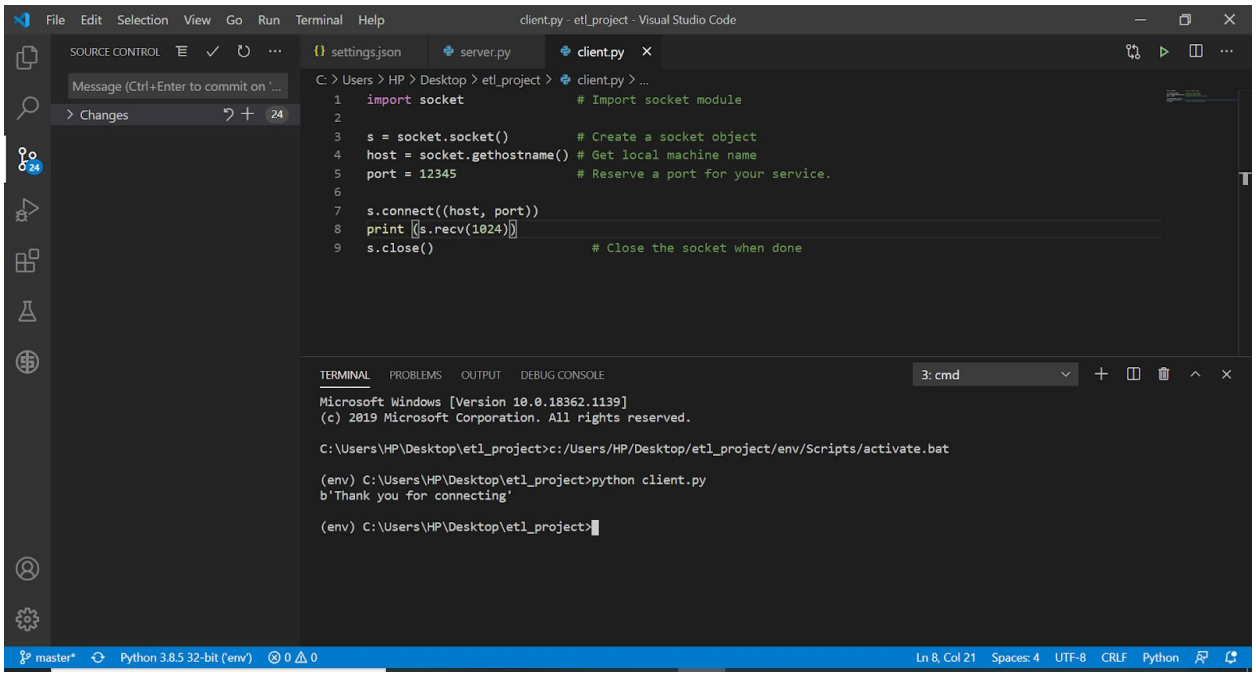
```
import socket                # Import socket module

s = socket.socket()          # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345                 # Reserve a port for your service.

s.connect((host, port))
print (s.recv(1024))
s.close()                      # Close the socket when done
```

OUTPUT:





CONCLUSION: 1.From this experiment,I learnt about the concepts of socket programming in brief.

2.I also successfully implemented socket programming to establish a connection between client and server.

REFERENCES: [1] https://www.tutorialspoint.com/python/python_networking.htm
[2] https://realpython.com/python-sockets/
[3] https://www.geeksforgeeks.org/socket-programming-cc/
[4] https://www.youtube.com/watch?v=T0rYSFPAR0A