

PYTHON PROGRAMING(INT-213)

Name : Anubhav soni

Reg no : 12008665

Program : CSE. B.Tech



* Acknowledgement:-

- * I would like to thanks my mentor - Prof.Sagar Pandey for his advice and input on this project.
- * Special thanks for Different teachers and professor who give information about different Python libraries and

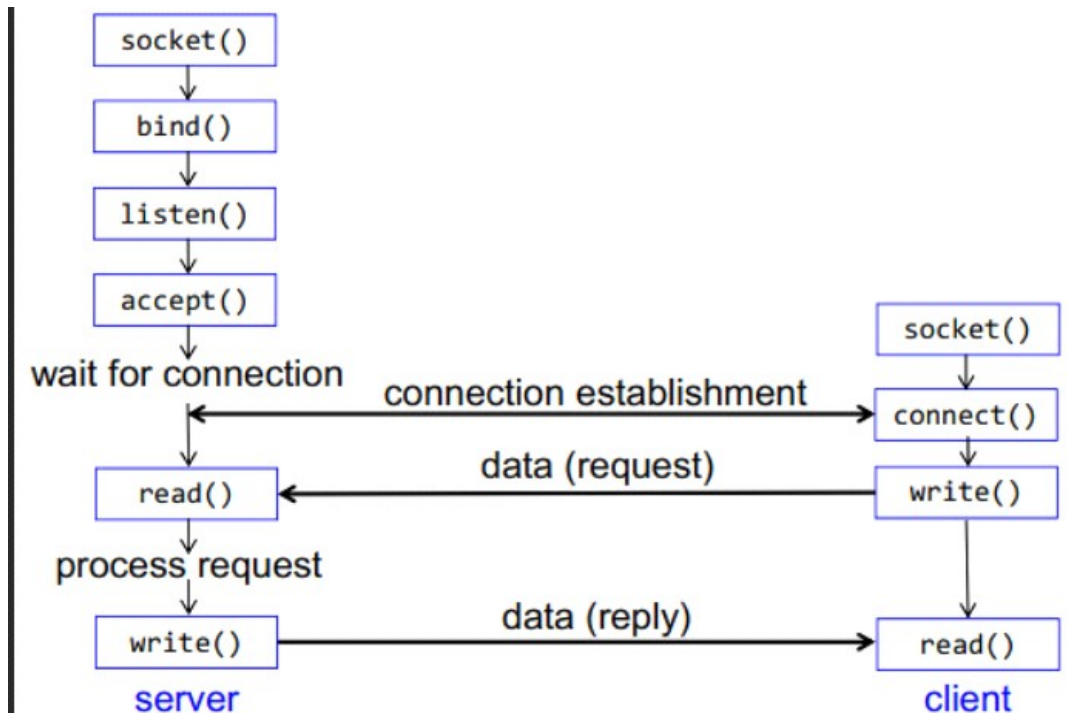
elements , i spend 6Hr nearby in learning about these different elements

- *Python_web is a working prototype of socket chatroom server .

- *It's a server program that use Socket library of Python in this project.

- * In this image you can see how server are essential.

- * It is is a prototype and also a part of a huge program so it can not display the real work but it represent the basic structure of transfer of data between multiple users.



*This is solo project.

*This project is done by only and only me.

this is my code.

```
import socket

import select


HEADER_LENGTH = 10

IP = "127.0.0.1"

PORT = 1234


server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)


server_socket.bind((IP, PORT))

server_socket.listen()


socket_list = [server_socket]


client = {}


def receive_message(client_socket):

    try:

        message_header = client_socket.recv(HEADER_LENGTH)

        if not len(message_header):
```

```
    return False
```

```
    messege_length = int(messege_header.decode('utf-8').strip())
```

```
    return {"header": messege_header, "data": client_socket.recv(messege_length)}
```

```
except:
```

```
    return False
```

```
while True:
```

```
    read_sockets, _, exception_sockets = select.select(socket_list, [], socket_list)
```

```
    for notified_socket in read_sockets:
```

```
        if notified_socket == server_socket:
```

```
            client_socket, client_address = server_socket.accept()
```

```
            user = receive_message(client_socket)
```

```
            if user is False:
```

```
                continue
```

```
            sockets_list.append(client_socket)
```

```
            clients[client_socket] = user
```

```
            print(f"Accepted new connection from {client_address[0]}:{client_address[1]}  
Username:{user['data'].decode('utf-8')}")
```

else:

 messege = receive_message(notified_socket)

 if messege is False:

 print(f"closed connection from {clients[notified_socket]['data'].decode('utf-8')}")

 sockets_list.remove(notified_socket)

 del client[notified_socket]

 continue

 user = clients[notified_socket]

 print(f"Recived message
from{user['data'].decode('utd-8')}:{messege['data'].decode('utd-8')}")

 for client_socket in client:

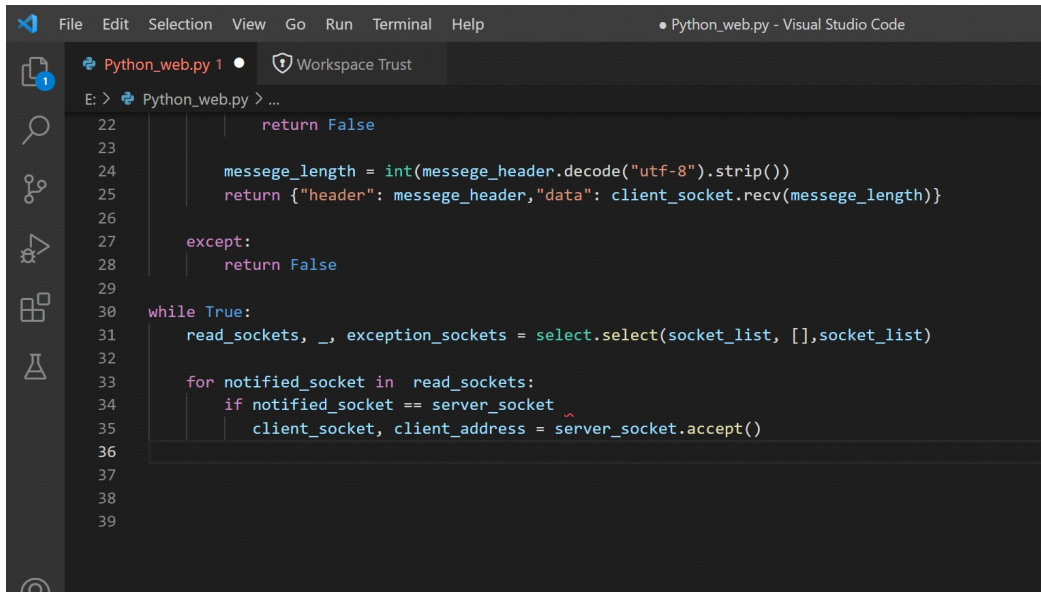
 if client_socket is notified_socket:

 client_socket.send(user['header'] + user['data'] + messege['header'] +
messsge['data'])

for notified_socket in exception_sockets:

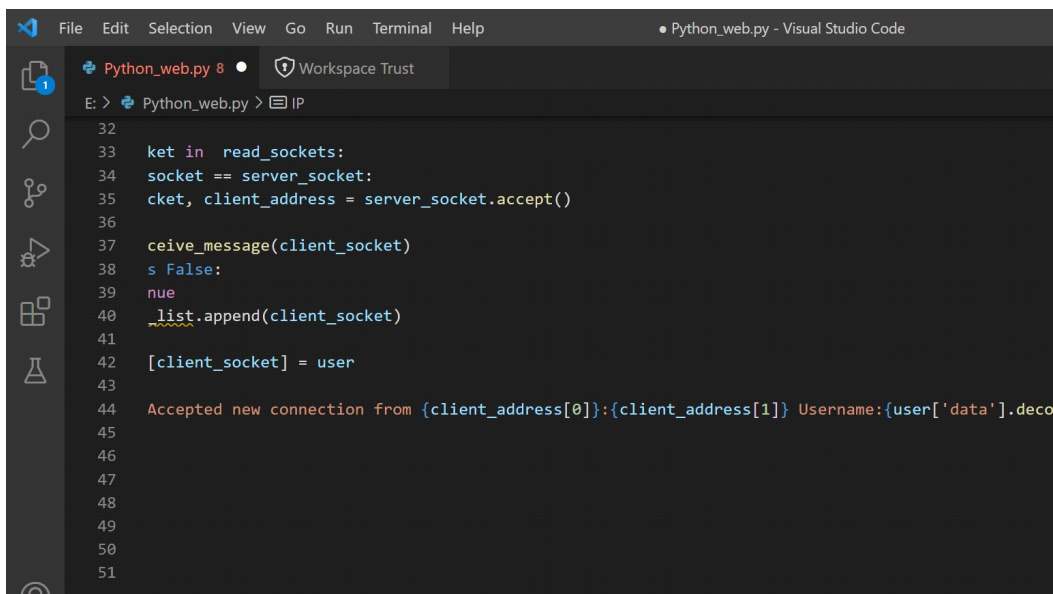
 socket_list.remove(notifid_socket)

 del client[]



```
Python_web.py 1 • Workspace Trust
E: > Python_web.py > ...

22         return False
23
24     message_length = int(message_header.decode("utf-8").strip())
25     return {"header": message_header, "data": client_socket.recv(message_length)}
26
27     except:
28         return False
29
30 while True:
31     read_sockets, _, exception_sockets = select.select(socket_list, [], socket_list)
32
33     for notified_socket in read_sockets:
34         if notified_socket == server_socket:
35             client_socket, client_address = server_socket.accept()
36
37
38
39
```



```
Python_web.py 8 • Workspace Trust
E: > Python_web.py > IP

32
33     ket in read_sockets:
34     socket == server_socket:
35     cket, client_address = server_socket.accept()
36
37     ceive_message(client_socket)
38     s False:
39     nue
40     _list.append(client_socket)
41
42     [client_socket] = user
43
44     Accepted new connection from {client_address[0]}:{client_address[1]} Username:{user['data'].deco
45
46
47
48
49
50
51
```

```
Python_web.py 4 • Workspace Trust
E: > Python_web.py > receive_message
2 import select
3
4 HEADER_LENGTH = 10
5 IP = "127.0.0.1"
6 PORT = 1234
7
8 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
10
11 server_socket.bind((IP, port))
12 server_socket.listen()
13
14 socket_list = [server_socket]
15
16 client = {}
17
18 def receive_message(client_socket):
19     try:
20         message_header = client_socket.recv(HEADER_LENGTH)
21         if not len(message_header):
```

```
Python_web.py 9+ • Workspace Trust
E: > Python_web.py > ...
44
45     clients[client_socket] = user
46
47     print(f"Accepted new connection from {client_address[0]}:{client_address[1]} Username:{user['data'].decode('utf-8')}")
48
49     else:
50         message = receive_message(notified_socket)
51
52         if message is False:
53             print(f"closed connection from {clients[notified_socket]['data'].decode('utf-8')}")
54             sockets_list.remove(notified_socket)
55             del client[notified_socket]
56             continue
57
58
59
60     user = clients[notified_socket]
61     print(f"Received message from {user['data'].decode('utf-8')}: {message['data'].decode('utf-8')}")
62
63     for client_socket in client:
64         if client_socket is notified_socket:
65             client_socket.send(user['header'] + user['data'] + message['header'] + message['data'])
66
67
68     for notified_socket in exception_sockets:
```

This code represent making of a server from line 61.

```
File "server.py", line 61
    print(f"Received message from {user['data'].decode('utf-8')}: {message['data'].decode('utf-8')}")
    ^
```


This is the only output given by it but dont underestimate this this line is enough to represent or it is major factor of making a server in online chatting platform by Python.

This project is a essential part of a big project that can't work without this essential part of it .

This is a Server making by using Socket This project explain basics for how to create a structure of Socket base Serve.