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

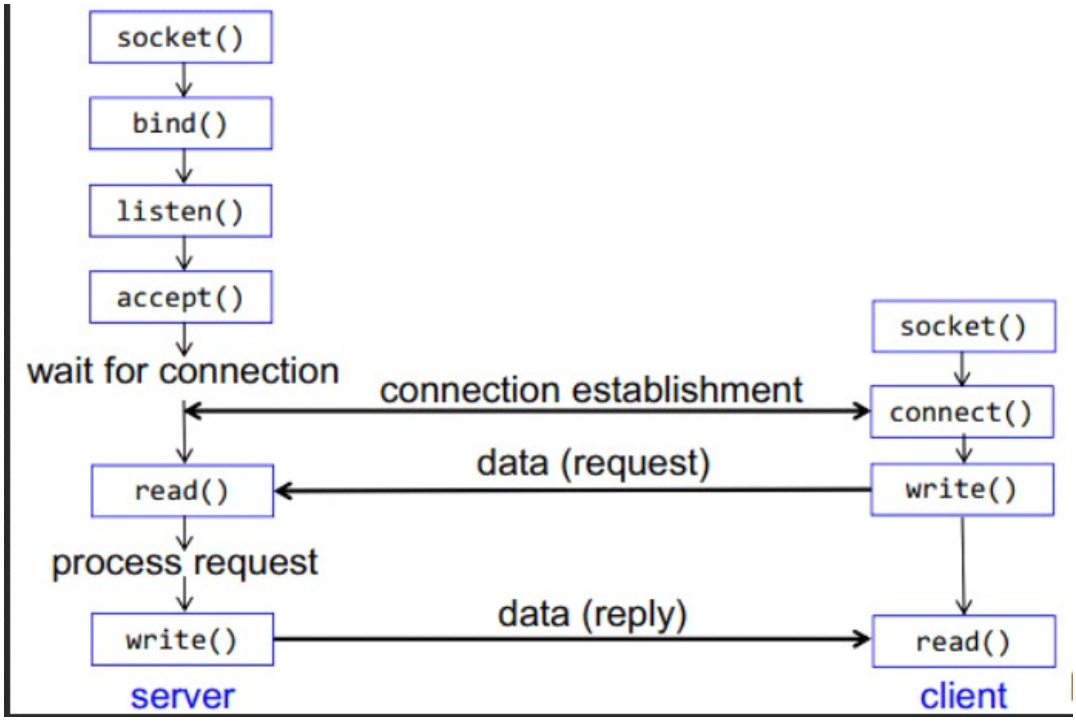
elements , i spend 6Hr nearby in learning about these different elements

*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 huje 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.

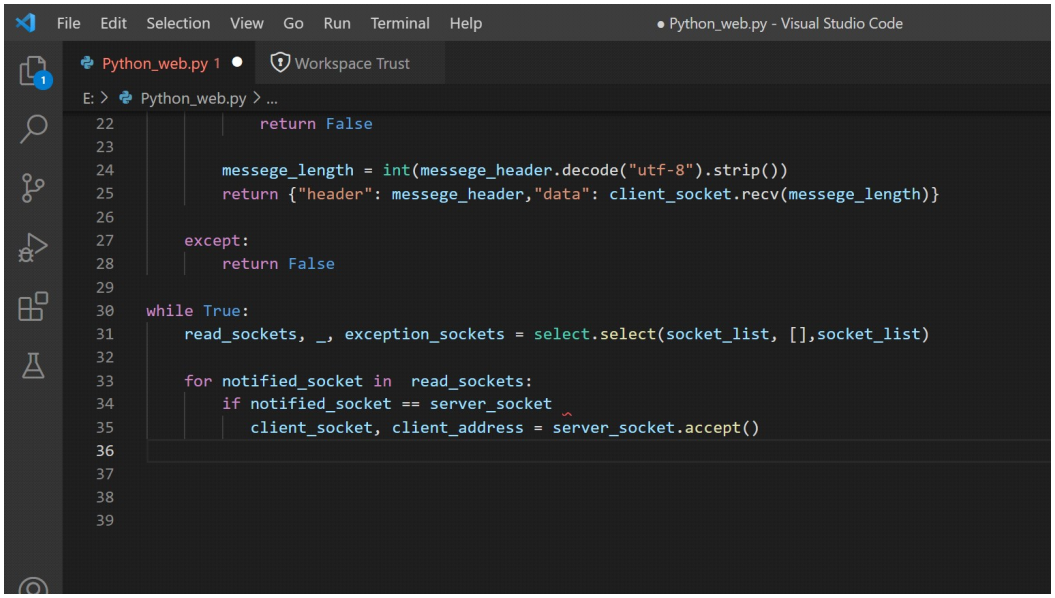
*This project is done by only and only me.

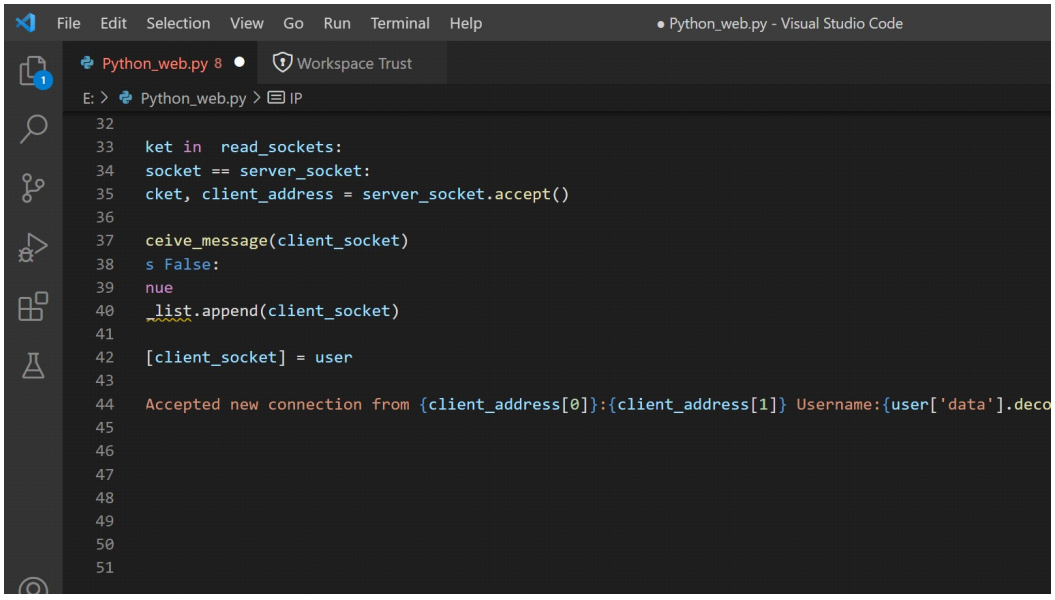
this is my code.

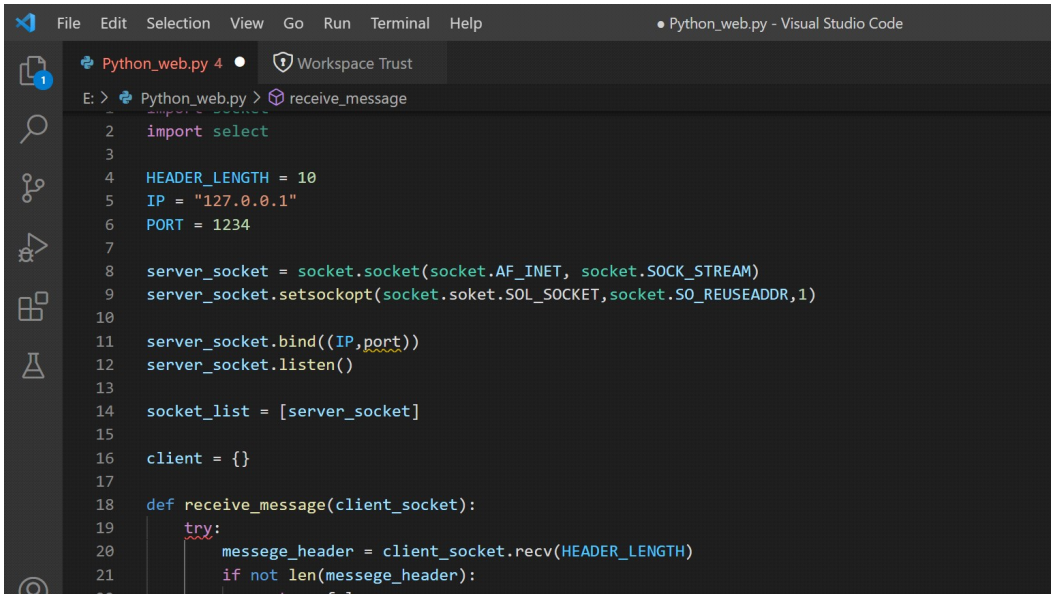
```
import socket

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.SQL_SOCKET,socket.SO_REUSEADDR,1)


server_socket.bind((IP,PORT))

server_socket.listen()


socket_list = [server_socket]


client = {}



def receive_message(client_socket):

    try:

        messege_header = client_socket.recv(HEADER_LENGTH)

        if not len(messege_header):
```

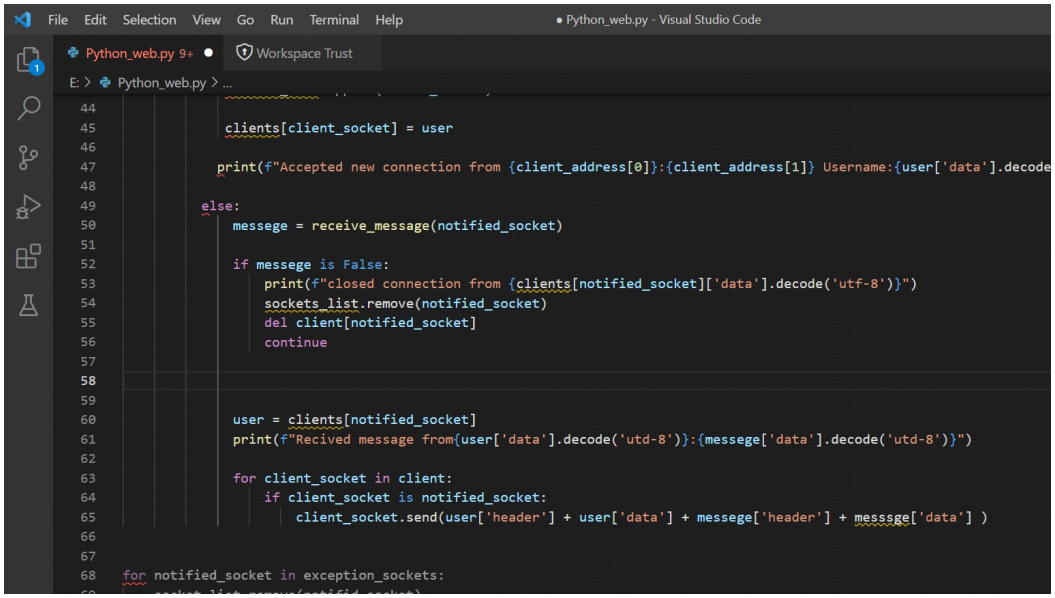
```
                return False

            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[]
```

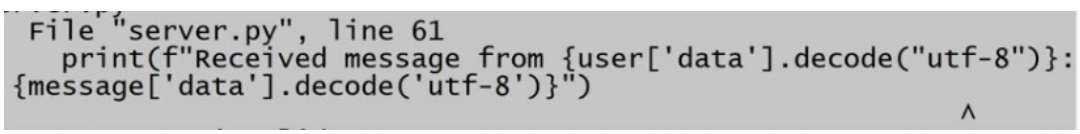

File Edit Selection View Go Run Terminal Help
• Python_web.py - Visual Studio Code
Python_web.py 1
Workspace Trust
E: > Python_web.py > ...
return False
messege_length = int(messege_header.decode("utf-8").strip())
return {"header": messege_header,"data": client_socket.recv(messege_length)}
except:
return 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()


File Edit Selection View Go Run Terminal Help
• Python_web.py - Visual Studio Code
Python_web.py 8
Workspace Trust
E: > Python_web.py > IP
ket in read_sockets:
socket == server_socket:
cket, client_address = server_socket.accept()
ceive_message(client_socket)
s False:
nue
_list.append(client_socket)
[client_socket] = user
Accepted new connection from {client_address[0]}:{client_address[1]} Username:{user['data'].deco





This code represent making of a server from line 61.



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.