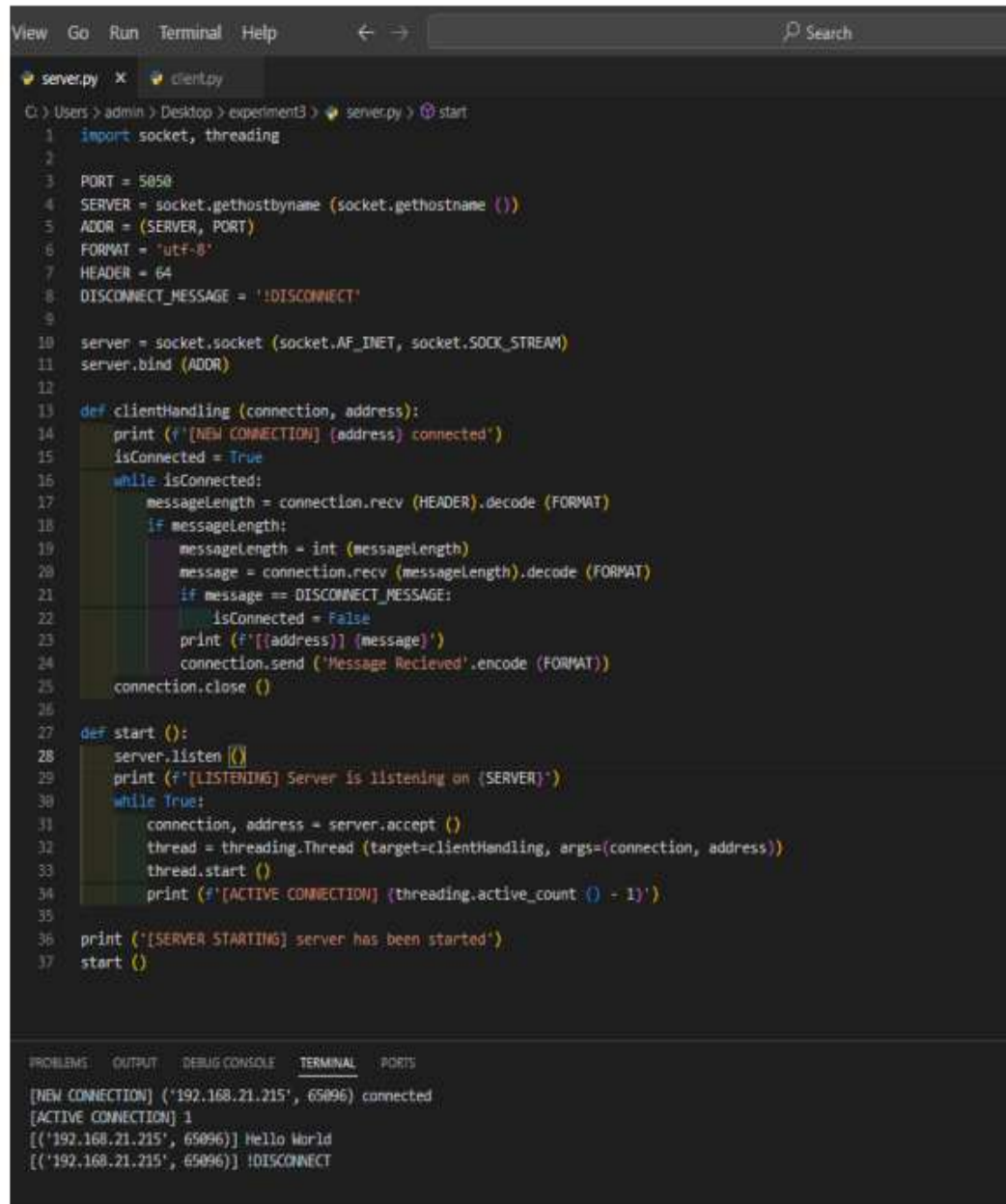


Heet Jain - 22



The image shows a code editor window with a dark theme. The editor has tabs for 'server.py' and 'client.py', with 'server.py' being the active tab. The code is a Python script for a multi-threaded socket server. It imports 'socket' and 'threading'. It defines a 'PORT' of 5050, a 'SERVER' address using 'socket.gethostname()', and an 'ADDR' tuple. It sets 'FORMAT' to 'utf-8' and a 'HEADER' of 64 bytes. A 'DISCONNECT_MESSAGE' is defined as '!DISCONNECT'. The 'clientHandling' function takes a 'connection' and 'address', prints a connection message, sets 'isConnected' to True, and enters a loop where it receives messages. If the message is the disconnect message, it sets 'isConnected' to False, prints the message, and sends a 'Message Recieved' response. The 'start' function listens on the server, accepts connections, and starts a new thread for each connection, printing the active connection count. The terminal at the bottom shows the execution output.

```
View Go Run Terminal Help
server.py x client.py
C:\Users> admin > Desktop > experiment3 > server.py > start
1 import socket, threading
2
3 PORT = 5050
4 SERVER = socket.gethostname (socket.gethostname ())
5 ADDR = (SERVER, PORT)
6 FORMAT = 'utf-8'
7 HEADER = 64
8 DISCONNECT_MESSAGE = '!DISCONNECT'
9
10 server = socket.socket (socket.AF_INET, socket.SOCK_STREAM)
11 server.bind (ADDR)
12
13 def clientHandling (connection, address):
14     print (f'[NEW CONNECTION] {address} connected')
15     isConnected = True
16     while isConnected:
17         messageLength = connection.recv (HEADER).decode (FORMAT)
18         if messageLength:
19             messageLength = int (messageLength)
20             message = connection.recv (messageLength).decode (FORMAT)
21             if message == DISCONNECT_MESSAGE:
22                 isConnected = False
23             print (f'[{address}] {message}')
24             connection.send ('Message Recieved'.encode (FORMAT))
25     connection.close ()
26
27 def start ():
28     server.listen (1)
29     print (f'[LISTENING] Server is listening on {SERVER}')
30     while True:
31         connection, address = server.accept ()
32         thread = threading.Thread (target=clientHandling, args=(connection, address))
33         thread.start ()
34         print (f'[ACTIVE CONNECTION] {threading.active_count () - 1}')
35
36 print (f'[SERVER STARTING] server has been started')
37 start ()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[NEW CONNECTION] ('192.168.21.215', 65096) connected
[ACTIVE CONNECTION] 1
[('192.168.21.215', 65096)] Hello World
[('192.168.21.215', 65096)] !DISCONNECT
```

```
View Go Run Terminal Help  Search

server.py client.py x

C:\Users\admin\Desktop> experiment3> client.py ...
1 import socket
2
3 PORT = 5050
4 SERVER = '192.168.21.215'
5 ADDR = (SERVER, PORT)
6 FORMAT = 'utf-8'
7 HEADER = 64
8 DISCONNECT_MESSAGE = '!DISCONNECT'
9
10 client = socket.socket (socket.AF_INET, socket.SOCK_STREAM)
11 client.connect (ADDR)
12
13 def send (messages):
14     message = messages.encode (FORMAT)
15     messageLength = len (message)
16     sendLength = str (messageLength).encode (FORMAT)
17     sendLength += b' ' * (HEADER - len (sendLength))
18     client.send (sendLength)
19     client.send (message)
20     print (client.recv (2048).decode (FORMAT))
21
22 if __name__ == '__main__':
23     ## Pass the Hello World message as an input
24     send ("Hello World")
25
26     send (DISCONNECT_MESSAGE)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\admin> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/admin/Desktop/experiment3/client.py
PS C:\Users\admin> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe c:/Users/admin/Desktop/experiment3/client.py
Message Recieved
Message Recieved
PS C:\Users\admin>
```

Activ
Go to S