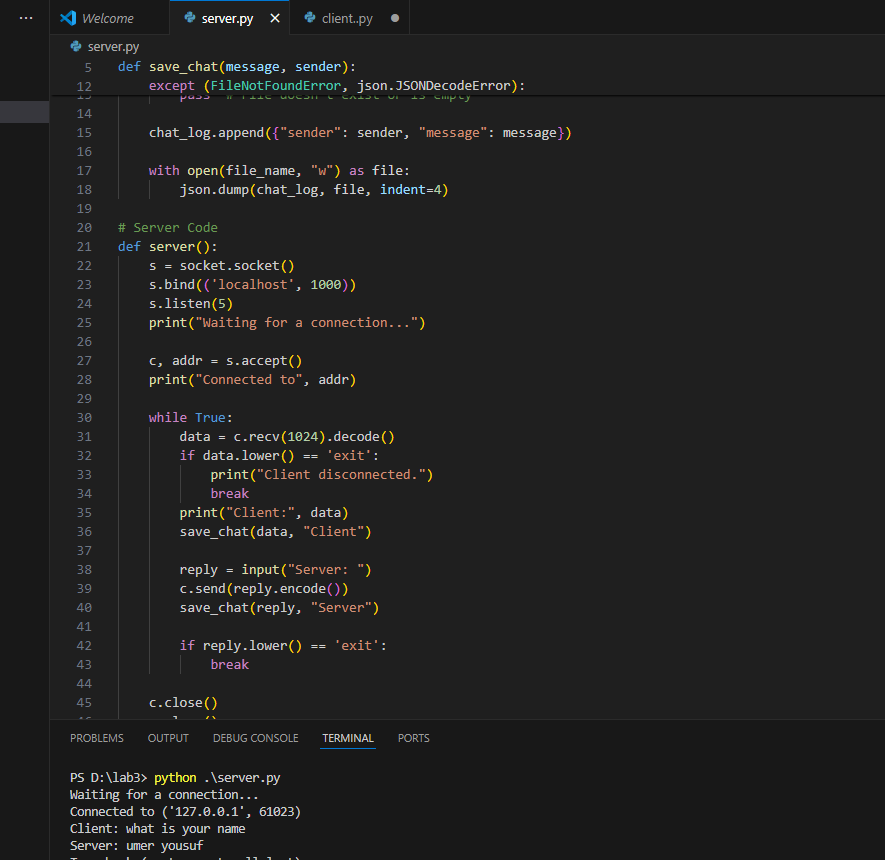
Name umer yousuf

22k-4515



Server code

import socket

import json

# Function to store chat messages in a JSON file

def save\_chat(message, sender):

chat\_log = []

file\_name = "chat\_history.json"

try:

with open(file\_name, "r") as file:

chat\_log = json.load(file)

except (FileNotFoundError, json.JSONDecodeError):

pass # File doesn't exist or is empty

chat\_log.append({"sender": sender, "message": message})

with open(file\_name, "w") as file:

json.dump(chat\_log, file, indent=4)

# Server Code

def server():

s = socket.socket()

s.bind(('localhost', 1000))

s.listen(5)

print("Waiting for a connection...")

c, addr = s.accept()

print("Connected to", addr)

while True:

data = c.recv(1024).decode()

if data.lower() == 'exit':

print("Client disconnected.")

break

print("Client:", data)

save\_chat(data, "Client")

reply = input("Server: ")

c.send(reply.encode())

save\_chat(reply, "Server")

if reply.lower() == 'exit':

break

c.close()

s.close()

if \_\_name\_\_ == "\_\_main\_\_":

server()

Client code

import socket

import json

# Function to store chat messages in a JSON file

def save\_chat(message, sender):

chat\_log = []

file\_name = "chat\_history.json"

try:

with open(file\_name, "r") as file:

chat\_log = json.load(file)

except (FileNotFoundError, json.JSONDecodeError):

pass # File doesn't exist or is empty

chat\_log.append({"sender": sender, "message": message})

with open(file\_name, "w") as file:

json.dump(chat\_log, file, indent=4)

# Client Code

def client():

s = socket.socket()

s.connect(('localhost', 1000))

print("Connected to the server.")

while True:

message = input("Client: ")

s.send(message.encode())

save\_chat(message, "Client")

if message.lower() == 'exit':

break

response = s.recv(1024).decode()

if response.lower() == 'exit':

print("Server disconnected.")

break

print("Server:", response)

save\_chat(response, "Server")

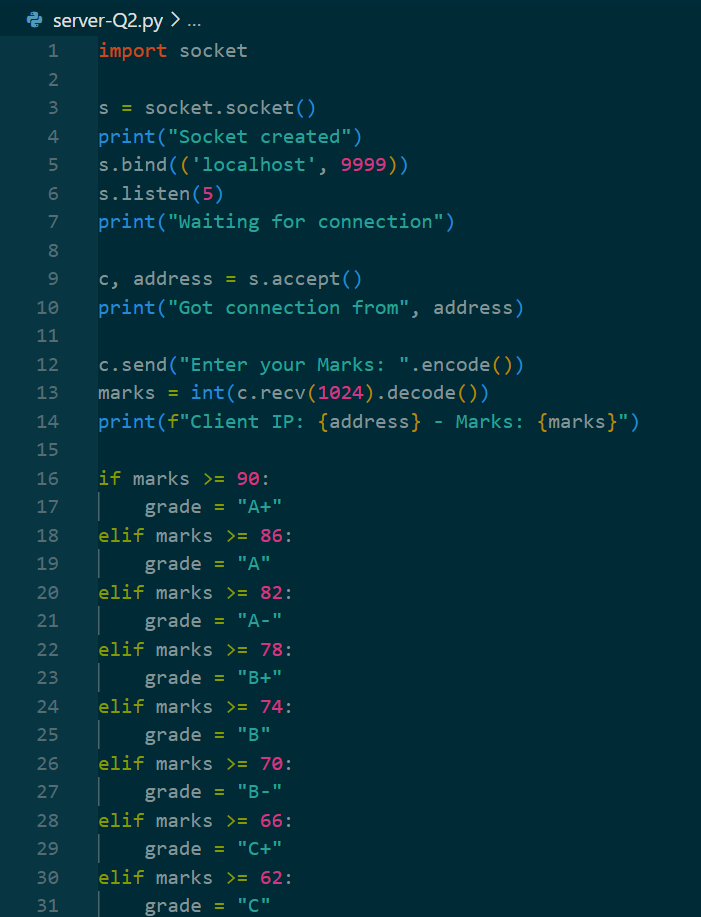
s.close()

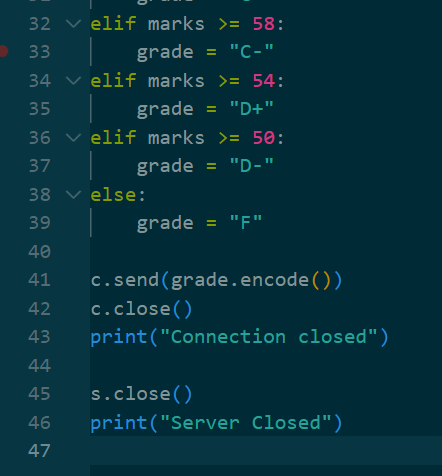
if \_\_name\_\_ == "\_\_main\_\_":

client()

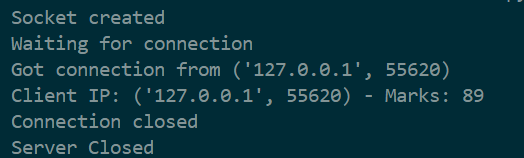
**Q2**

**SERVER**

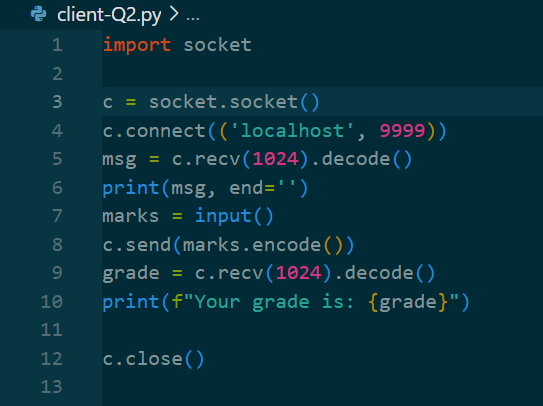


****

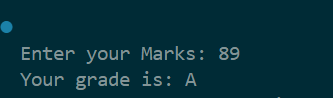
**SERVER OUTPUT**

****

**CLIENT CODE**

****

**CLIENT OUTPUT**

****