#BINDIYA SUDARSUN 22MIC0040
#server.py:

import socket import threading from rpc_runtime import send_msg, recv_msg, validate_types

#RPC FUNCTION

def calculate_grade_average(profile): grades = profile[“grades”] return sum(grades) / len(grades)

#CLIENT HANDLER

def handle_client(conn, addr): print(f”[+] Connected: {addr}“)

```
try:
    while True:
        request = recv_msg(conn)
        if not request:
            break

        method = request.get("method")
        params = request.get("params")

        try:
            if method == "calculate_grade_average":

                profile = params[0]

                #TYPE CHECKING
                validate_types(profile)

                result = calculate_grade_average(profile)

                send_msg(conn, {"result": result})

            else:
                send_msg(conn, {"error": "Unknown method"})

        except TypeError as e:
            send_msg(conn, {"error": str(e)})

finally:
    conn.close()
    print(f"[-] Disconnected: {addr}")
```

#START SERVER def start_server(): host = “127.0.0.1” port = 5000

```
server_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_sock.bind((host, port))
server_sock.listen()

print(f"RPC Server running on {host}:{port}")

while True:
    conn, addr = server_sock.accept()
    threading.Thread(
        target=handle_client,
        args=(conn, addr),
        daemon=True
    ).start()
```

if **name** == “**main**”: start_server()