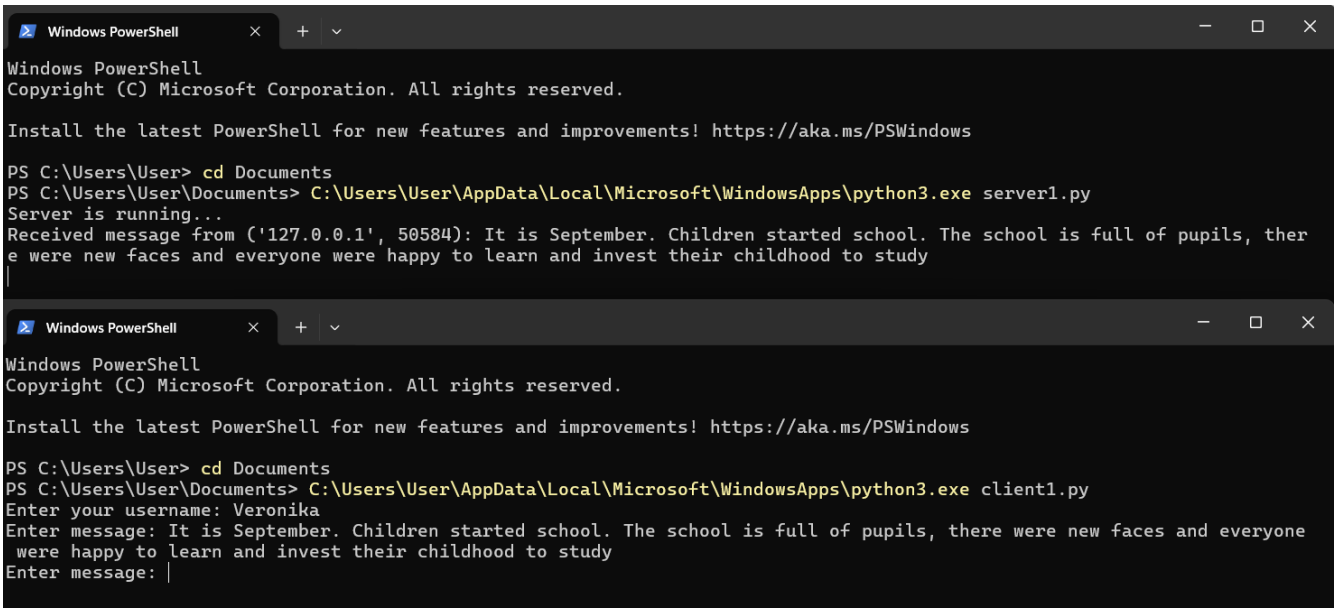
Computer Network: Socket UDP Client-Server Application

Simple chat application using UDP (User Datagram Protocol) for communication between a client and a server. UDP is a connectionless protocol, making it suitable for applications where a small amount of data needs to be transmitted quickly.
Components:

1. Server:
    - Create a UDP socket and bind it to a specific port.
    - Listen for incoming messages from clients.
    - Upon receiving a message, process it and send a response back to the client.
    - Handle multiple clients concurrently using threading or asynchronous programming.
2. Client:
    - Create a UDP socket.
    - Allow the user to input messages to send to the server.
    - Send messages to the server.
    - Receive responses from the server and display them to the user.



Implemented this UDP connection in Python:

Client code:

```
import socket

def main():
    server_ip = '127.0.0.1'
    server_port = 12000

    client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
    username = input('Enter your username: ')

    while True:
        message = input('Enter message: ')
        client_socket.sendto(message.encode(), (server_ip, server_port))

if __name__ == "__main__":
    main()
```

Server code:

```
import socket

def main():
    server_ip = '127.0.0.1'
    server_port = 12000

    server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    server_socket.bind((server_ip, server_port))

    print('Server is running...')

    while True:
        message, client_address = server_socket.recvfrom(2048)
        message = message.decode()
        print(f'Received message from {client_address}: {message}')

if __name__ == "__main__":
    main()
```

Project is done by Abdullakim Zamirbek uulu
Subject: Computer Networks