VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER NETWORK (CO3094)

Assignment 1

Develop a network application

Advisor:	Phạm Trần Vũ Nguyễn Mạnh Thìn
Group:	3
Members:	Đinh Lê Dũng – 2152483 (CC04) Nguyễn Hồ Tiến Đạt – 2152503 (CC03) Phan Hữu Thành – 2112305 (CC03) Nguyễn Hoàng Quốc Tuấn – 2153078 (CC03)

Ho Chi Minh, December 2023

Contents

1 Introduction

1.1 Case study

A P2P file sharing application is a computer network where computers connect directly to each other to transmit data. This is contrary to the traditional model, where data is transmitted through a central server. With the allocation of resources to avoid server overload, the P2P file sharing model was invented to avoid too much dependence on the server and server resources, leading to congestion due to too much access at one time.

- File sharing between individuals: P2P file sharing applications can be used to share files between individuals, such as friends, family, or colleagues. This can be done over a local network or over the Internet.
- File sharing in businesses: P2P file sharing applications can be used to share files within a business. This can be used to share documents, source code, or other types of files between employees.
- File sharing in non-profit organizations: P2P file sharing applications can be used to share files within non-profit organizations. This can be used to share documents, images, or videos between members of the organization.

1.2 Requirement

The objective of the assignment is to build a simple file-sharing application using the TCP/IP protocol stack. The system consists of a server and clients connected to the server. The clients will notify the server about the files they want to publish and can request to retrieve files from other clients.

For the server:

- The server holds information about the clients participating in the system.
- The server manages the list of files that the clients share.
- When receiving a file download request from a client, the server must be able to identify which client the file belongs to and respond accordingly.

For the client:

- The client can send a file download request to the server and receive a response from the server about the client that contains the file to be downloaded.
- The client can interact with other clients in a Peer-to-Peer mechanism to be able to send and receive files between clients.
- The client can upload files from their machine to their repository and notify the server when completed.
- The client can also delete files from their repository and notify the server when completed.

1.3 Function of the application

Function of server:

- Check the list of clients

- Live check clients (ping hostname)
- Discover the list of local files of clients (discover hostname)

Function of clients:

- Upload file (publish lname fname)
- Download file from other clients (fetch fname)
- Remove file from repository (delete fname)

2 Protocols

2.1 Socket Programming

Sockets and the socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer, or one that's physically connected to an external network, with its own connections to other networks.
Python's socket module provides an interface to the sockets API.The primary socket API functions and methods in this module are:

- socket()
- .bind()
- .listen()
- .accept()
- .connect()
- .connect_ex()
- .send()
- .recv()
- .close()

Python provides a convenient and consistent API that maps directly to system calls.

2.2 TCP Socket

In this assignment, we will stick to TCP Socket.

Transmission Control Protocol (TCP): is connection-oriented, and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established. TCP provides reliable data transfer, congestion control.

3 Demonstration

3.1 Server view

The server interface goes with all function Ping hostname, Discover hostname. Besides, there are List of hostname, and the hostname's repository.

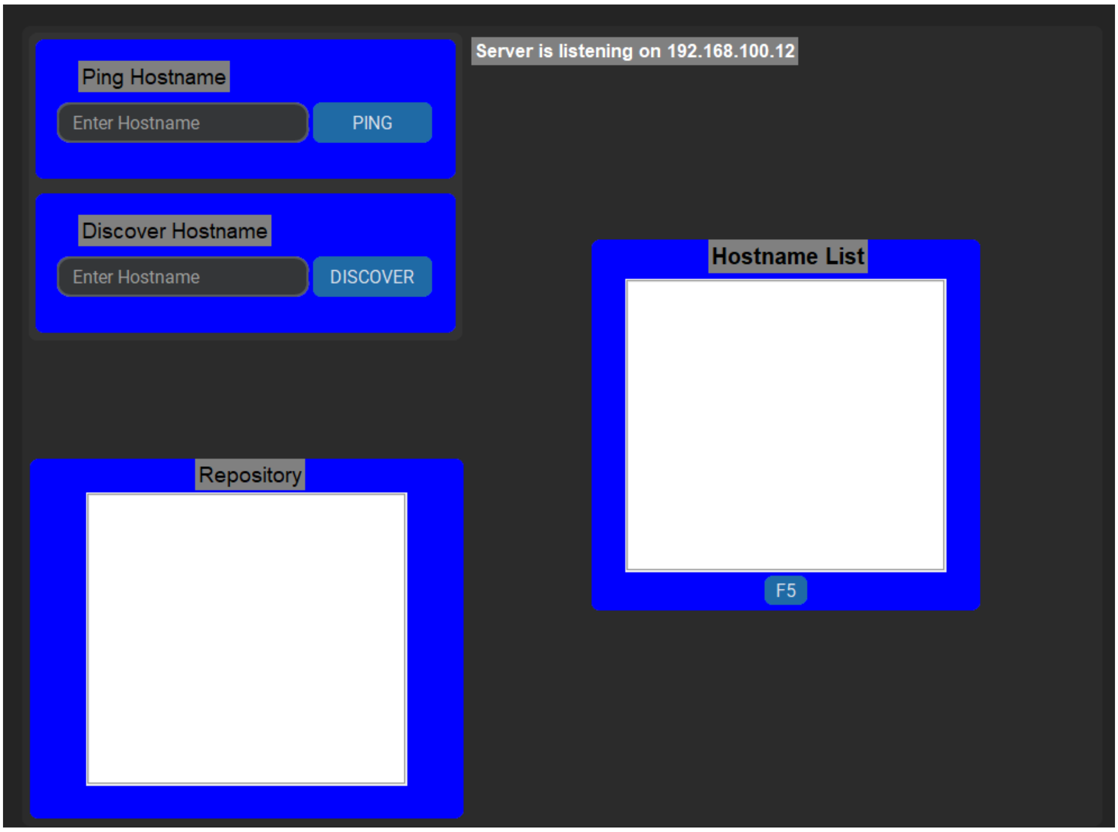
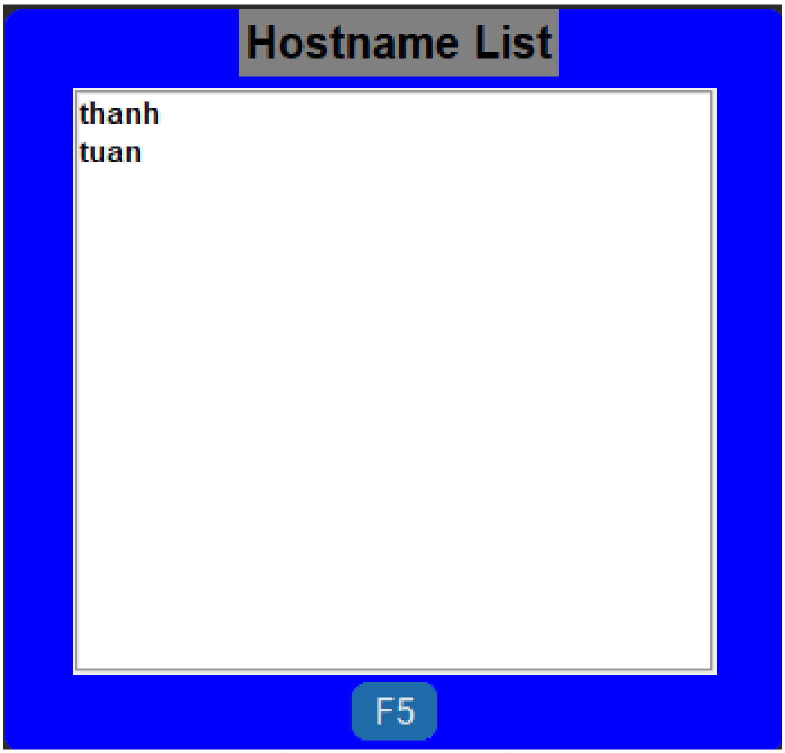


Figure 1: *Server view*

3.1.1 Hostname List

When clients connect to server, the server can view all the hostname

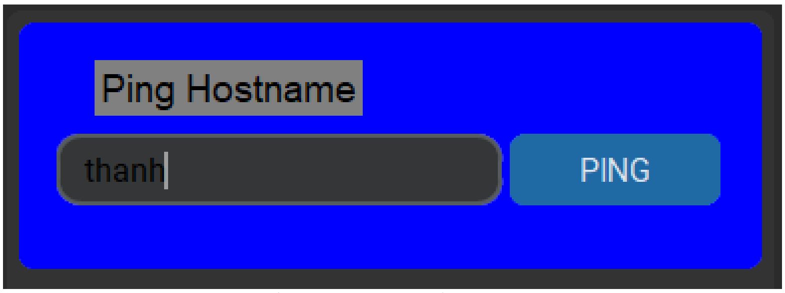


thanh
tuan

Figure 2: *Hostname list*

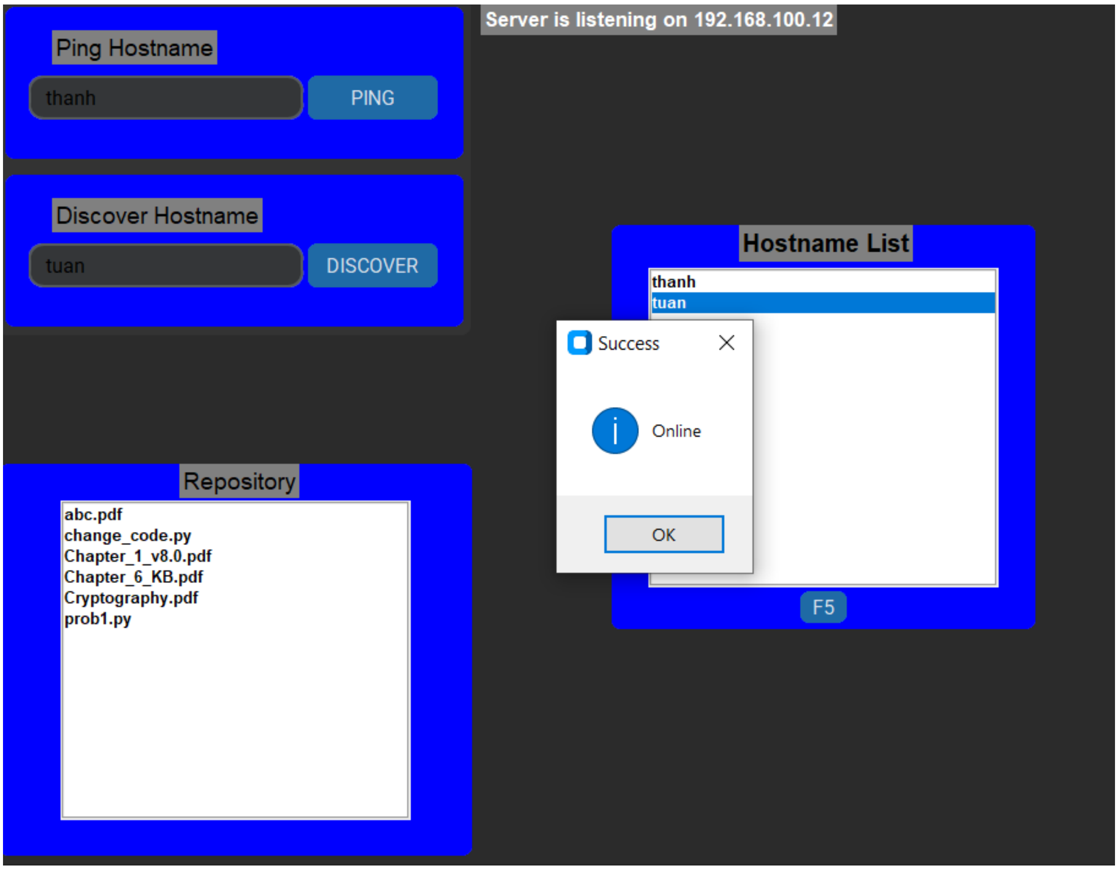
3.1.2 Ping hostname

When pinging to a host, the status of that host will be displayed.



thanh

Figure 3: *Ping hostname*



Server is listening on 192.168.100.12

thanh

tuan

abc.pdf
change_code.py
Chapter_1_v8.0.pdf
Chapter_6_KB.pdf
Cryptography.pdf
prob1.py

thanh
tuan

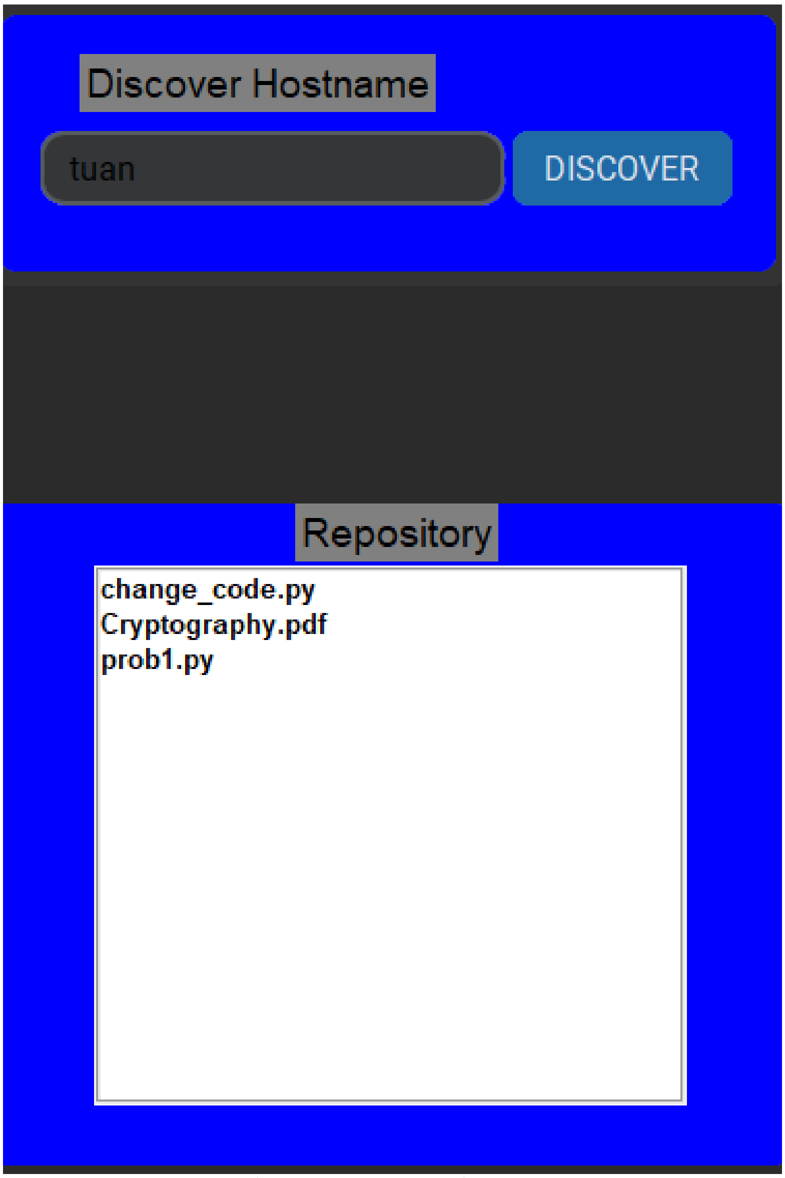
Success

Online

Figure 4: *Ping result*

3.1.3 Discover hostname

When server discovers a hostname, it's publish file will be listed in Repository box.



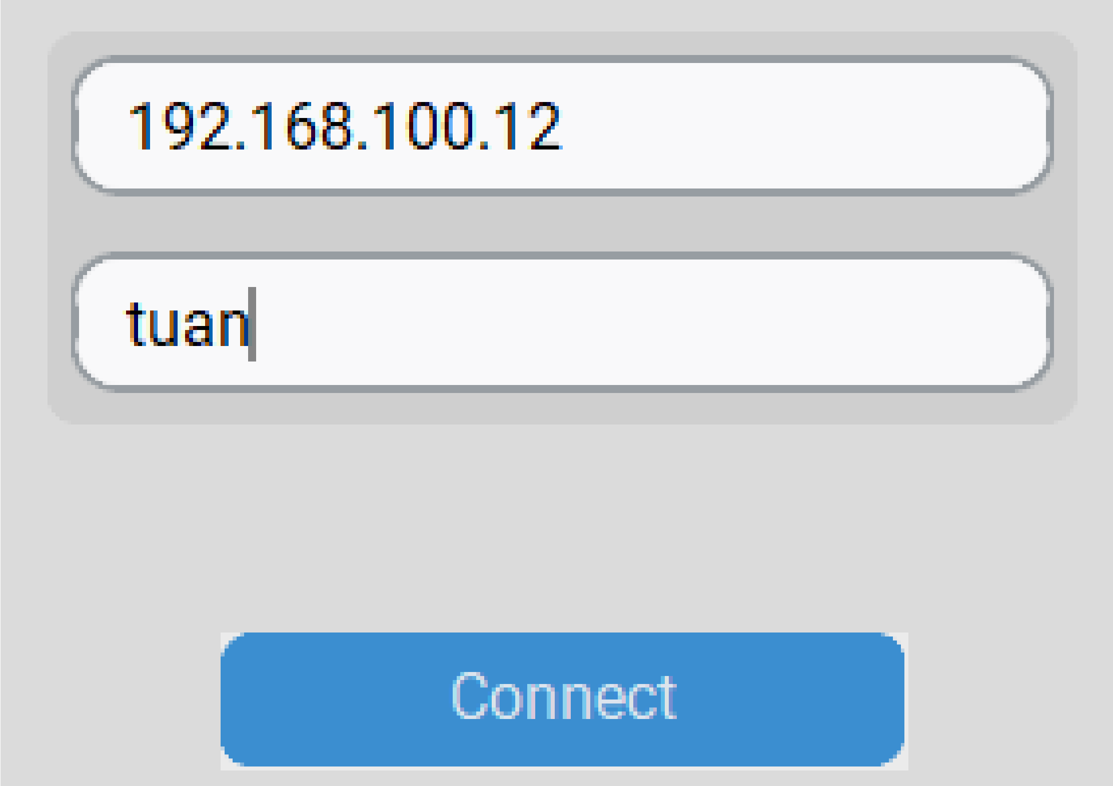
change_code.py
Cryptography.pdf
prob1.py

Figure 5: *Discover hostname*

3.2 Client view

3.2.1 Client Connection

The clients will connect to server IP address and name their hostname.



192.168.100.12

tuan

Figure 6: *Client connection*

3.2.2 Client Interface

The client GUI goes with all function Fetch file, Delete file, Publish file

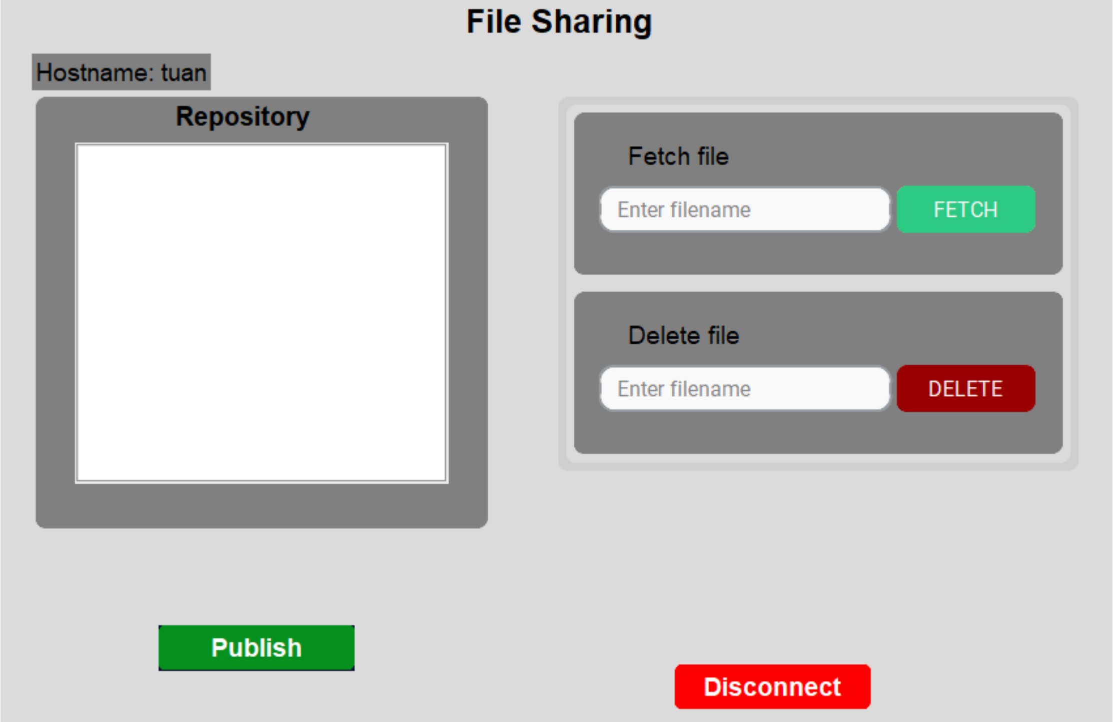
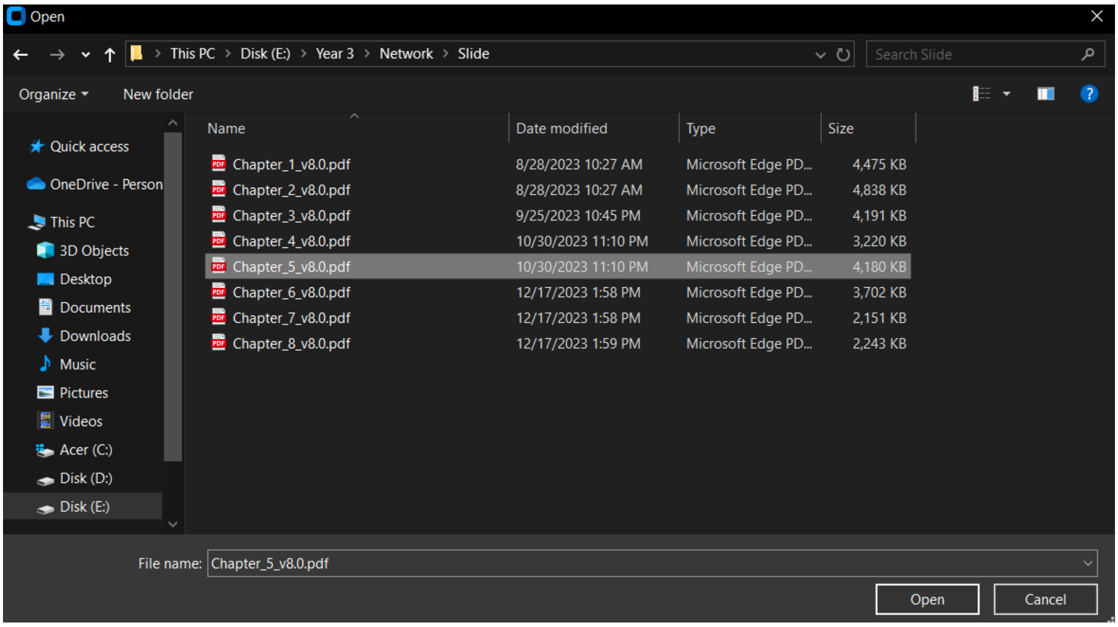


Figure 7: *Client Interface*

3.2.3 Publish File

Initially, the repository of file published is empty as the above figure since the client has not published any files. Then, when clients want to publish file, they will browse a file in their computer to the application.

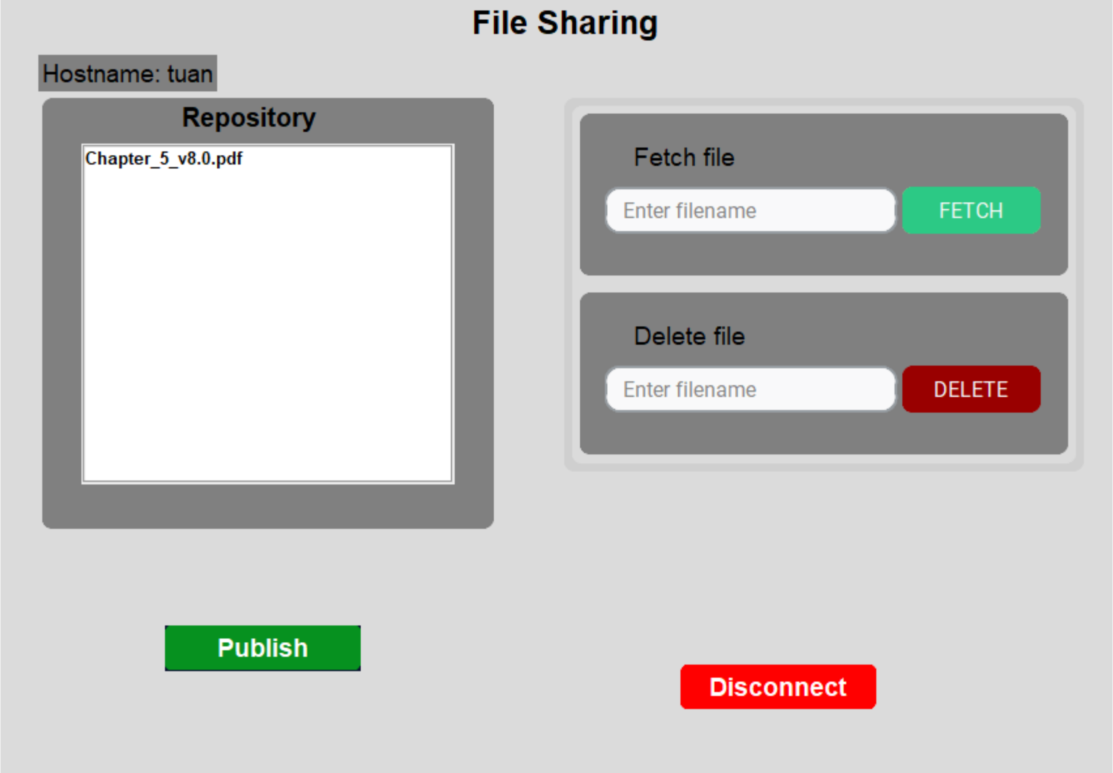


Name	Date modified	Type	Size
Chapter_1_v8.0.pdf	8/28/2023 10:27 AM	Microsoft Edge PD...	4,475 KB
Chapter_2_v8.0.pdf	8/28/2023 10:27 AM	Microsoft Edge PD...	4,838 KB
Chapter_3_v8.0.pdf	9/25/2023 10:45 PM	Microsoft Edge PD...	4,191 KB
Chapter_4_v8.0.pdf	10/30/2023 11:10 PM	Microsoft Edge PD...	3,220 KB
Chapter_5_v8.0.pdf	10/30/2023 11:10 PM	Microsoft Edge PD...	4,180 KB
Chapter_6_v8.0.pdf	12/17/2023 1:58 PM	Microsoft Edge PD...	3,702 KB
Chapter_7_v8.0.pdf	12/17/2023 1:58 PM	Microsoft Edge PD...	2,151 KB
Chapter_8_v8.0.pdf	12/17/2023 1:59 PM	Microsoft Edge PD...	2,243 KB

File name: Chapter_5_v8.0.pdf

Figure 8: *Publish file*

And the result is:

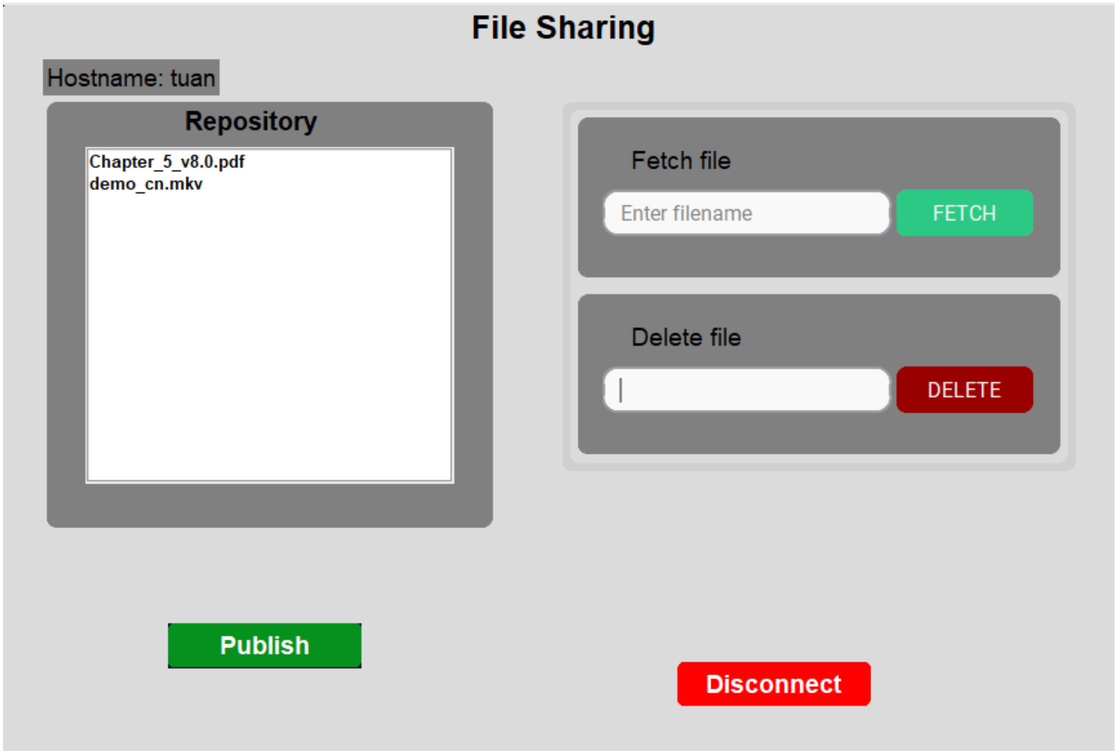


Hostname: tuan

Chapter_5_v8.0.pdf

Figure 9: *Publish file result*

A video is also required for the demonstration. The file type is mkv, and is about 7MB.



Chapter_5_v8.0.pdf
demo_cn.mkv

Figure 10: *Publish video*

3.2.4 Fetch File

A client will fetch a file from other client. In this demonstration, client with hostname *thanh* will try to fetch file from *tuan*

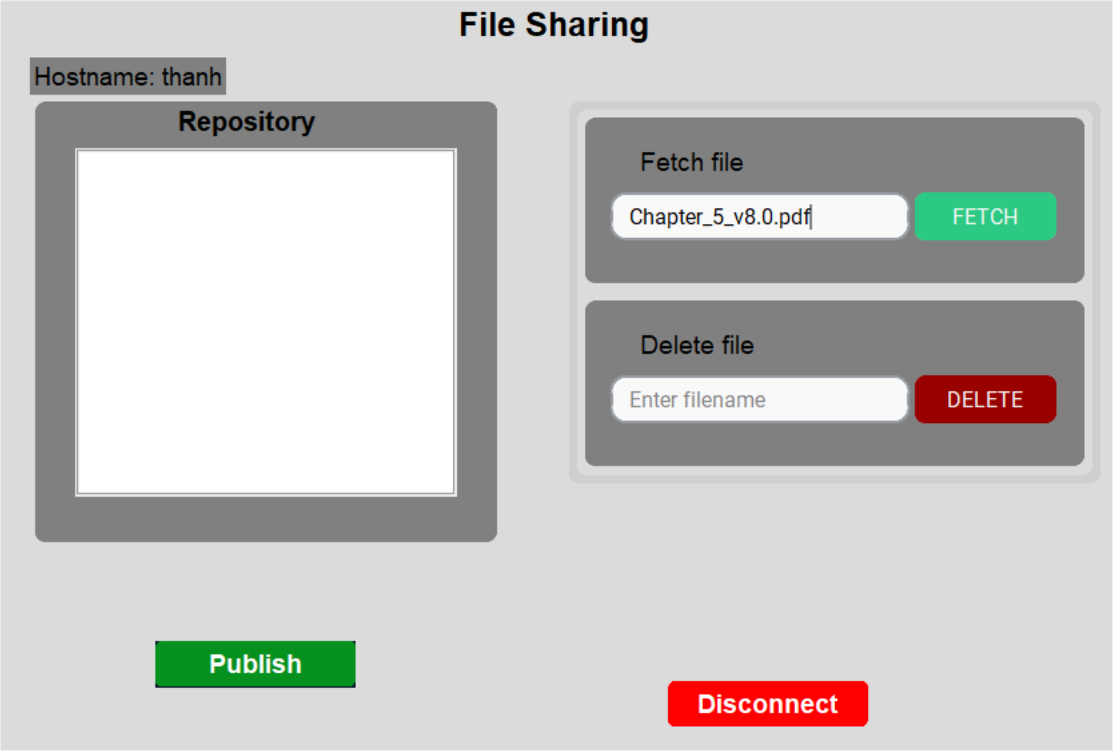
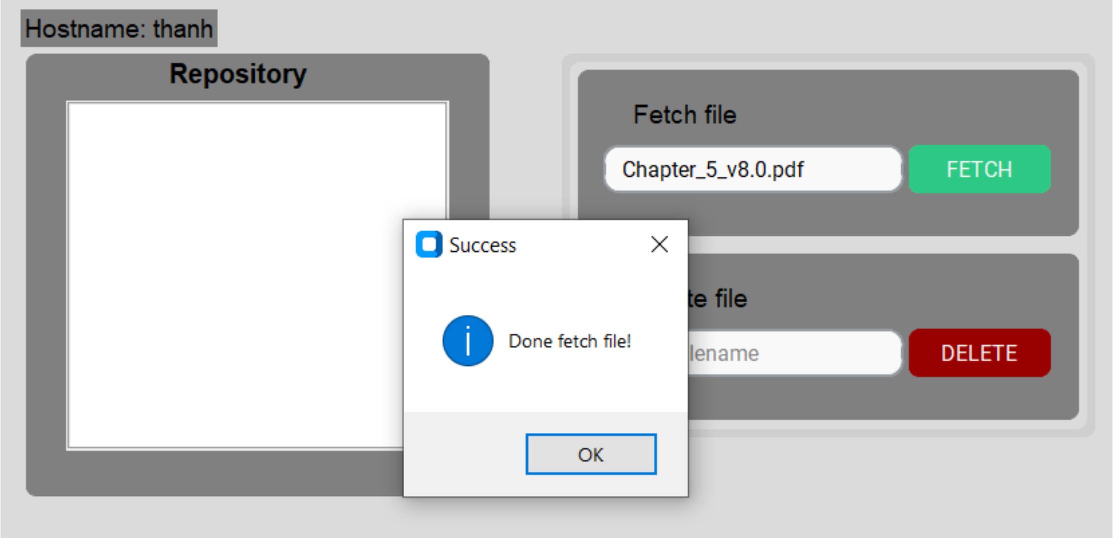


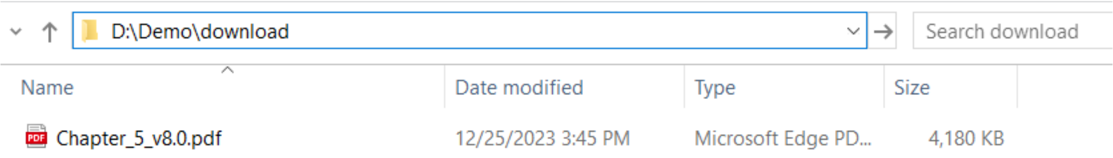
Figure 11: *Fetch file*



Done fetch file!

Figure 12: *Fetch file done*

And the file fetched will be located in download folder which is not represented in publish repository.



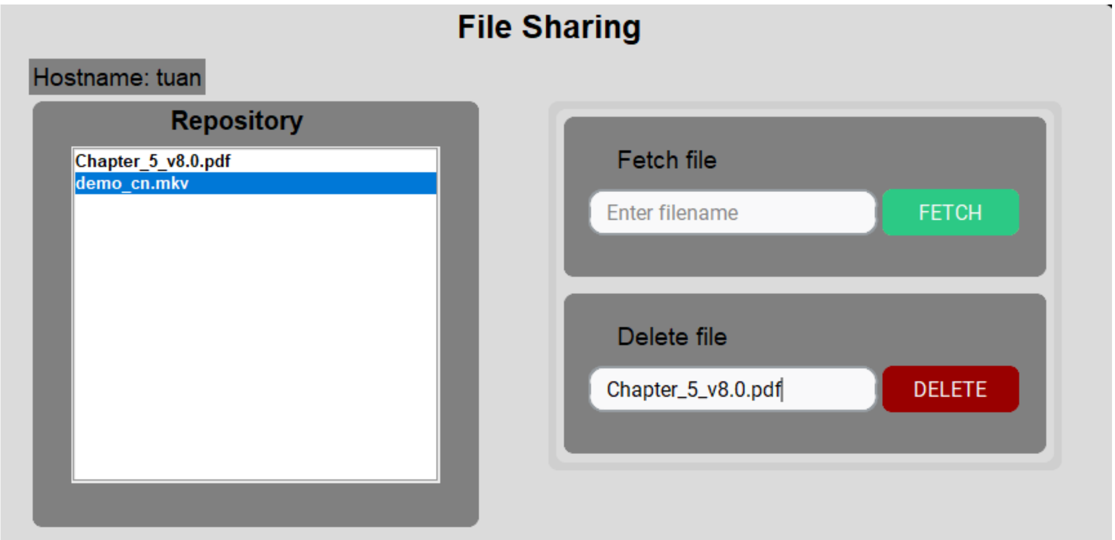
D:\Demo\download

Name	Date modified	Type	Size
Chapter_5_v8.0.pdf	12/25/2023 3:45 PM	Microsoft Edge PD...	4,180 KB

Figure 13: *Fetch file result*

3.2.5 Delete file

Delete function will delete the file that already published by client. Client *tuan* wants to delete file:



File Sharing

Hostname: tuan

Repository

Chapter_5_v8.0.pdf
demo_cn.mkv

Fetch file

Delete file

Chapter_5_v8.0.pdf

Figure 14: *Delete file*

And the result is:

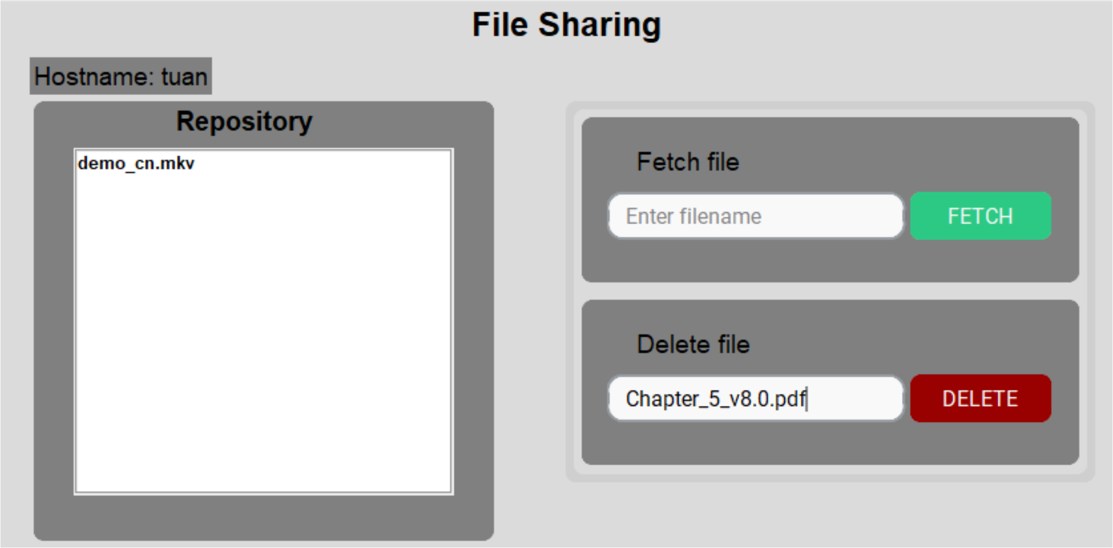


Figure 15: *Delete file result*

3.2.6 Multi-clients fetch a file

Finally, we will let many clients fetch a video file from a client. The video is published by hostname *tuan* with the name *demo_ cn.mkv*. Two hostname *thanh* and *dat* will try to fetch that video.

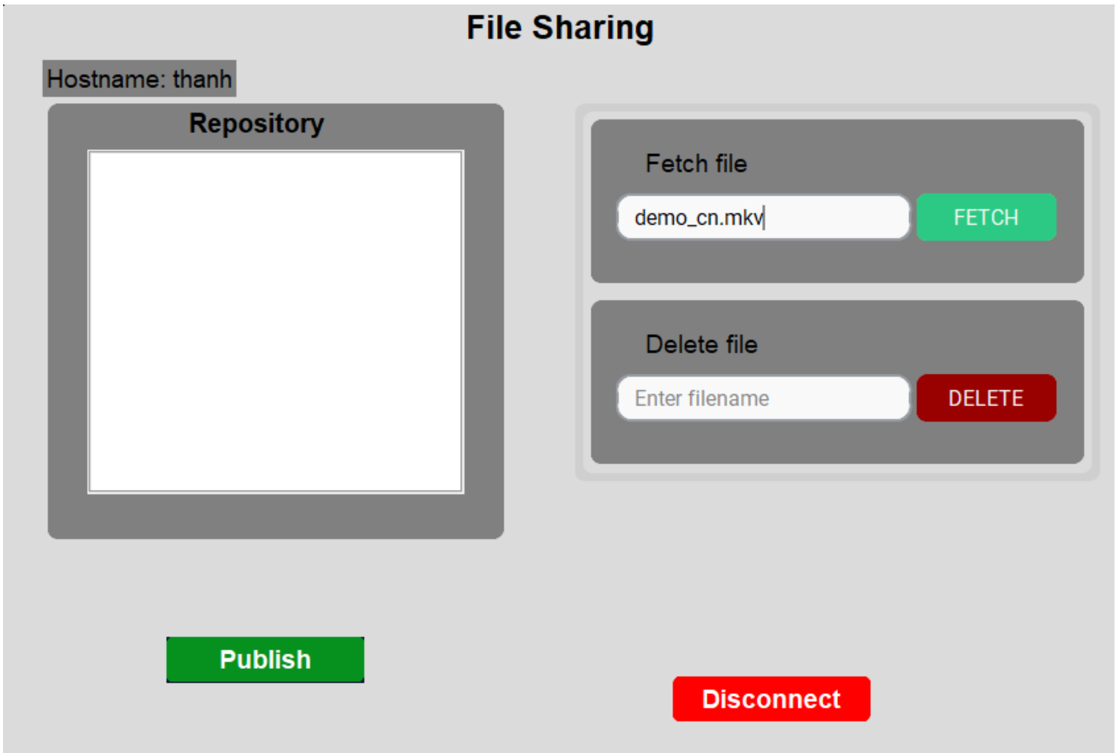
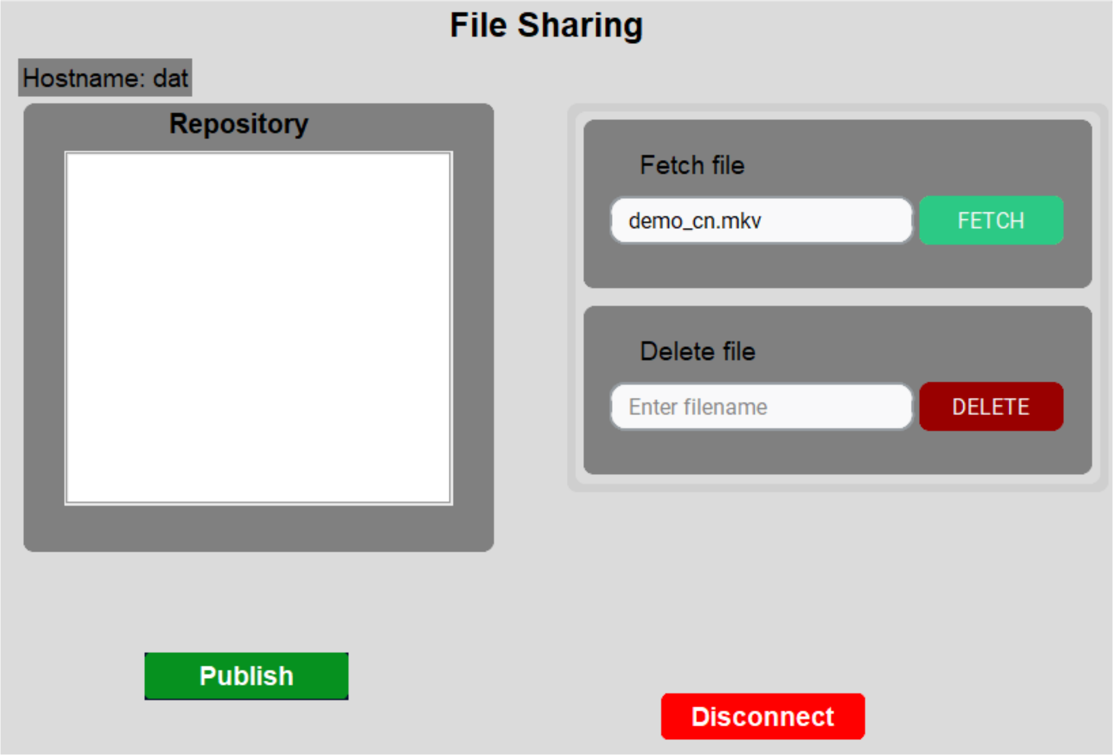


Figure 16: *Client thanh fetch video file*



demo_cn.mkv

Figure 17: *Client dat fetch video file*

And the result:

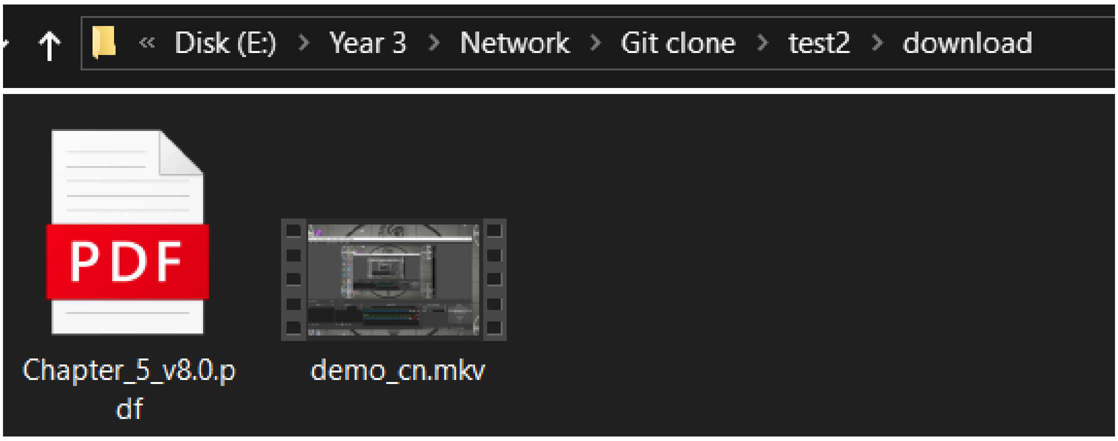


Figure 18: *thanh fetch result*

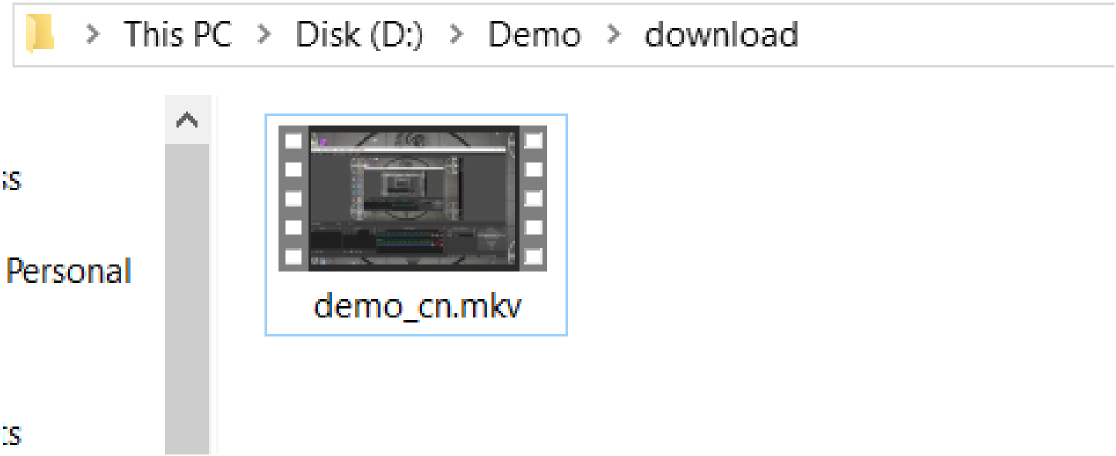


Figure 19: *dat fecth result*

These videos work with clear sound and quality.

4 References

- P2P (Peer To Peer) File sharing, https://www.geeksforgeeks.org/p2p-peer-to-peer-file-sharing/
- Socket programming in Python (Guide), https://realpython.com/python-sockets/
- Transmission Control Protocol, https://en.wikipedia.org/wiki/Transmission_Control_Protocol