File Sharing App - 5

What is our GOAL for this MODULE?

The goal of this module is to learn desktop File transfer Protocol Application server part also create a folder on the server and then learn how to use FTP to download the files and save them to the computer's download folder

What did we ACHIEVE in the class TODAY?

- How to make shared folder on server
- How to download files from the server

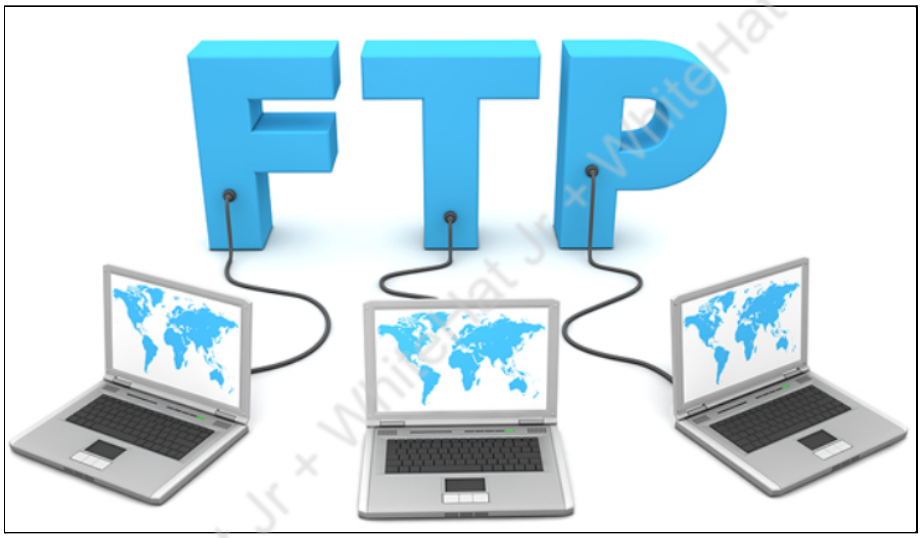
Which CONCEPTS/CODING BLOCKS did we cover today?

- We learned how to make shared folder on FTP server
- We learned how to download files from the server.

The KEY CONCEPT

1. What is FTP?

   The File Transfer Protocol is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network



   File transfer protocol (FTP) is a set of rules that computers follow for the transferring of files from one system to another over the internet. It may be used by a business to transfer files from one computer system to another, or websites may use FTP to upload or download files from a website's server.

How did we DO the activities?

1. Define variables sending_file, downloading_file, and filetodownload and set their values to None

```
sending_file = None
downloading_file = None
filetodownload  = None
```

2. Create functions for **grantAccess()** at the server side, which will be called when downloading the file

```
def grantAccess(client_name):
    global clients

    other_client_name = clients[client_name]["connected_with"]
    other_client_socket = clients[other_client_name]["client"]
    msg = "access granted"
    other_client_socket.send(msg.encode())
```

3. Create functions for **declineAccess()** at the server side, which will be called when downloading the file

```
def declineAccess(client_name):
    global clients

    other_client_name = clients[client_name]["connected_with"]
    other_client_socket = clients[other_client_name]["client"]
    msg = f"Oops!!! {client_name} decline your request..."
    other_client_socket.send(msg.encode())
```

4. Create function **handleSendFile()**

```
def handleSendFile(client_name, file_name, file_size):
    global clients

    clients[client_name]["file_name"] = file_name
    clients[client_name]["file_size"] = file_size
    other_client_name = clients[client_name]["connected_with"]
    other_client_socket = clients[other_client_name]["client"]
    msg = f"\n{client_name} want to send {file_name} file with size {file_size} bytes. Do you want to download ? Y/N "
    other_client_socket.send(msg.encode())
    time.sleep(1)
    msgdown=f"Download:{file_name}"
    other_client_socket.send(msgdown.encode())
```

5. Modify function **handleMessages()** to add message to be displayed to grant or decline access

```
def handleMessges(client, message, client_name):
    if(message == 'show list'):
        handleShowList(client)
    elif(message[:7] == 'connect'):
        handleClientConnection(message, client, client_name)
    elif(message[:10] == 'disconnect'):
        disconnectWithClient(message, client, client_name)
    elif(message[:4] == "send"):
        file_name = message.split(" ")[1]
        file_size  = int(message.split(" ")[2])
        handleSendFile(client_name, file_name, file_size)
        print(client_name+" "+file_name+" "+file_size)
    elif(message == "y" or message == "yes"):
        grantAccess(client_name)
    elif(message == "n" or message == "no"):
        declineAccess(client name)
    else:
        connected = clients[client_name]["connected_with"]
        if(connected):
            sendTextMessage(client_name, message)
        else:
            handleErrorMessage(client)
```

6. Modify our **receiveMessage()** function after **grantAccess()** and **declineAccess()** function.

```
def receiveMessage():
    global SERVER
    global name
    global textarea
    global BUFFER_SIZE
    global downloading_file
    global filetodownload

    while True:
        chunk = SERVER.recv(BUFFER_SIZE)
        try:
            if("tiul" in chunk.decode() and "1.0," not in chunk.decode()):
                letter_list = chunk.decode().split(",")
                listbox.insert(letter_list[0],letter_list[0]+":"+letter_list[1]+": "+letter_list[3]+" "+letter_list[5])
                print(letter_list[0],letter_list[0]+":"+letter_list[1]+": "+letter_list[3]+" "+letter_list[5])

            elif(chunk.decode() == "access granted"):
                labelchat.configure(text="")
                textarea.insert(END,"\n"+chunk.decode('ascii'))
                textarea.see("end")
            elif(chunk.decode() == "Oops!!! client decline your request..."):
                labelchat.configure(text="")
                textarea.insert(END,"\n"+chunk.decode('ascii'))
                textarea.see("end")
            elif("download ?" in chunk.decode()):
                downloading_file = chunk.decode('ascii').split(" ")[4].strip()
                BUFFER_SIZE = int(chunk.decode('ascii').split(" ")[8])
                textarea.insert(END,"\n"+chunk.decode('ascii'))
                textarea.see("end")
                print(chunk.decode('ascii'))

            elif("Download:" in chunk.decode()):
                getfilename = chunk.decode().split(":")
                filetodownload = getfilename[1]

            else:
                textarea.insert(END,"\n"+chunk.decode('ascii'))
                textarea.see("end")
```

7. Make changes in **browseFiles()** function

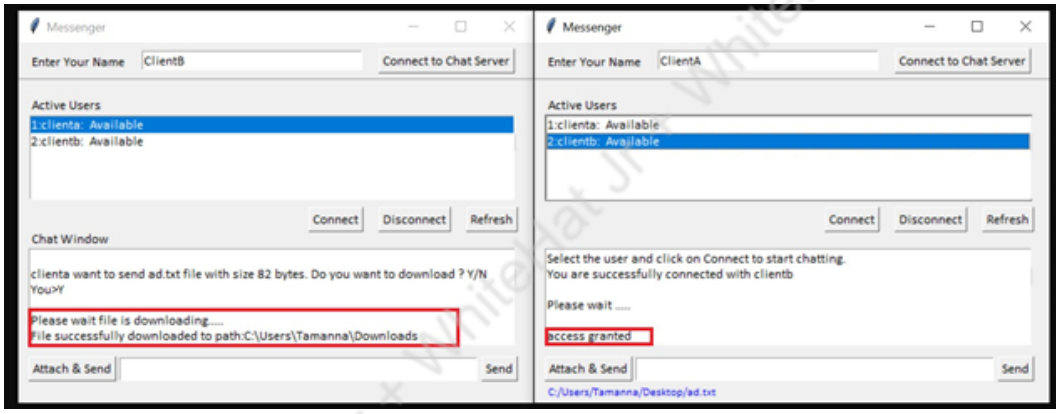
```
def browseFiles():
    global sending_file
    global textarea
    global filePathLabel

    try:
        filename = filedialog.askopenfilename()
        filePathLabel.configure(text=filename)
        HOSTNAME = "127.0.0.1"
        USERNAME = "lftpd"
        PASSWORD = "lftpd"

        ftp_server = ftplib.FTP(HOSTNAME, USERNAME, PASSWORD)
        ftp_server.encoding = "utf-8"
        ftp_server.cwd('shared_files')
        fname=ntpath.basename(filename)
        with open(filename, 'rb') as file:
            ftp_server.storbinary(f"STOR {fname}", file)

        ftp_server.dir()
        ftp_server.quit()

        message=("send "+fname)
        if(message[:4] == "send"):
            print("Please wait .....\n")
            textarea.insert(END,"\n"+"\nPlease wait .....\n")
            textarea.see("end")
            sending_file = message[5:]
            file_size = getFileSize("shared_files/"+sending_file)
            final_message = message + " " + str(file_size)
            SERVER.send(final_message.encode())
            textarea.insert(END,"file successfully sent..")

    except FileNotFoundError:
        print("Cancle Button Pressed")
```

8. Create a function **sendMessage()** to send a text message to the other client.

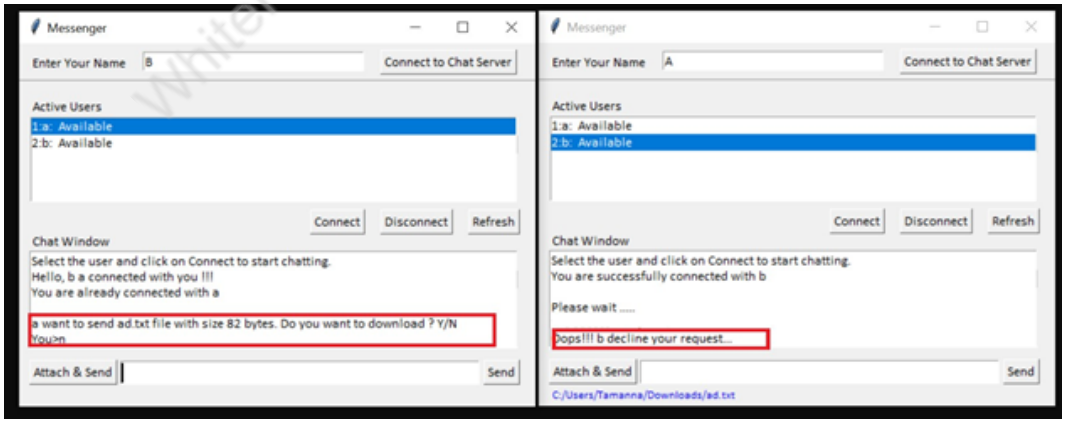
```
def sendMessage():
    global SERVER
    global textarea
    global text_message

    msgtosend= text_message.get()

    SERVER.send(msgtosend.encode('ascii'))
    textarea.insert(END,"\n"+"You>"+msgtosend)
    textarea.see("end")
    text_message.delete(0, 'end')
    if(msgtosend == "y" or msgtosend == "Y"):
        #print("\nPlease wait file is downloading.....")
        textarea.insert(END,"\n"+"\nPlease wait file is downloading.....")
        textarea.see("end")
        HOSTNAME = "127.0.0.1"
        USERNAME = "lftpd"
        PASSWORD = "lftpd"
        home = str(Path.home())
        download_path=home+"/Downloads"
        ftp_server = ftplib.FTP(HOSTNAME, USERNAME, PASSWORD)
        ftp_server.encoding = "utf-8"
        ftp_server.cwd('shared_files')
        fname=filetodownload
        local_filename = os.path.join(download_path, fname)
        file = open(local_filename, 'wb')
        ftp_server.retrbinary('RETR '+ fname, file.write)
        ftp_server.dir()
        file.close()
        ftp_server.quit()
        print("File successfully downloaded to path:"+download_path)
        textarea.insert(END,"\n"+"File successfully downloaded to path:"+download_path)
        textarea.see("end")
```

9. server.py in terminal/cmd looks like -

client.py in the terminal/cmd looks like -Click on Attach & send button and access the file from the computer system. If Client press **"Y" or "y"**



After pressing N or n



We have completed our first File sharing Desktop Application.

What's NEXT?

In the next class we will _____

EXTEND YOUR KNOWLEDGE

You can learn more about messaging from https://en.wikipedia.org/wiki/Windows_Messenger_service.