OCTOBER 2021

“<u>LAB -1 REPORT”</u>

Submitted for the course

Of

DISTRIBUTED SYSTEMS

Under the guidance of

Dr. CHANCE R EARY

Submitted by

SHIVANI MANOJKUMAR PANCHIWALA – 1001982478

IMPLEMENTATION DETAILS

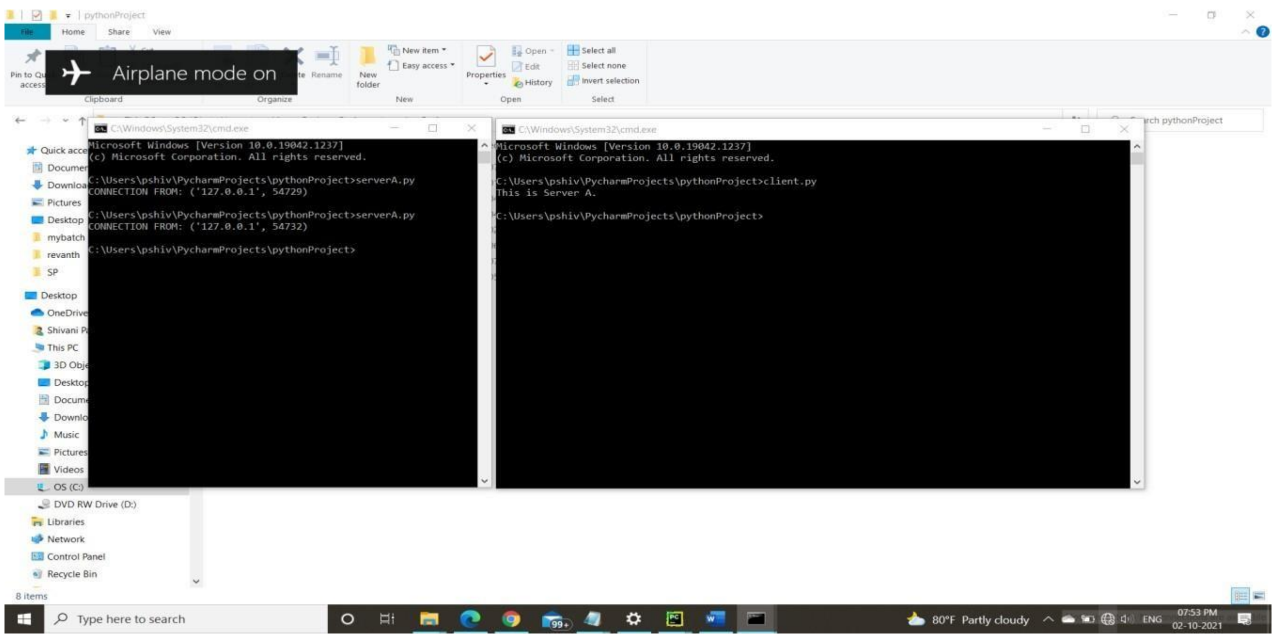
I implement this project into python language and used PyCharm for the Programming.

In this project, first I was trying to simple basic establish the connection between client to server. For establish the connection I import the socket and OS module. And then write the server code and client code for as per below from the given reference.
https://stackoverflow.com/questions/47539028/transfer-contents-of-a-folder-over-network-by-python

```
# SERVER  CODE
sock = socket()            # Build Socket Object
sock.bind(('', 5000))              # bind the socket with server and port number
sock.listen(2)                   #allow maximum 2 connection to the socket
client, addr = sock.accept()      #wait till a client accept and establish the connection
print("CONNECTION FROM:", str(addr))   # display client address
```

```
# Client Code
# Make a directory for the received files.
os.makedirs('client',exist_ok=True)

sock = socket()             # Build Socket Object
sock.connect(('localhost',5000))      #bind host address and port together and connect to the server A
with sock,sock.makefile('rb') as clientfile:
    while True:
        raw = clientfile.readline()     # read the file
        if not raw: break # no more files, server closed connection.
        print(raw.decode())  # print and decode the serverA file
        break     # no more files, server closed connection
```

Then I got the output like this.



```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54729)

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54732)

C:\Users\pshiv\PycharmProjects\pythonProject>
```

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>
```

Then I write the same client & server code for server B and establish the connection. After, Establish the connection between Server B to Server A to Client, In Server A, I give the path of directory of my folder and list out the files from directory.

```
dir_name = 'C:\\Users\\pshiv\\PycharmProjects\\pythonProject\\MP'    # path of
the folder
arr = os.listdir(dir_name)        # list out the files from directory
```

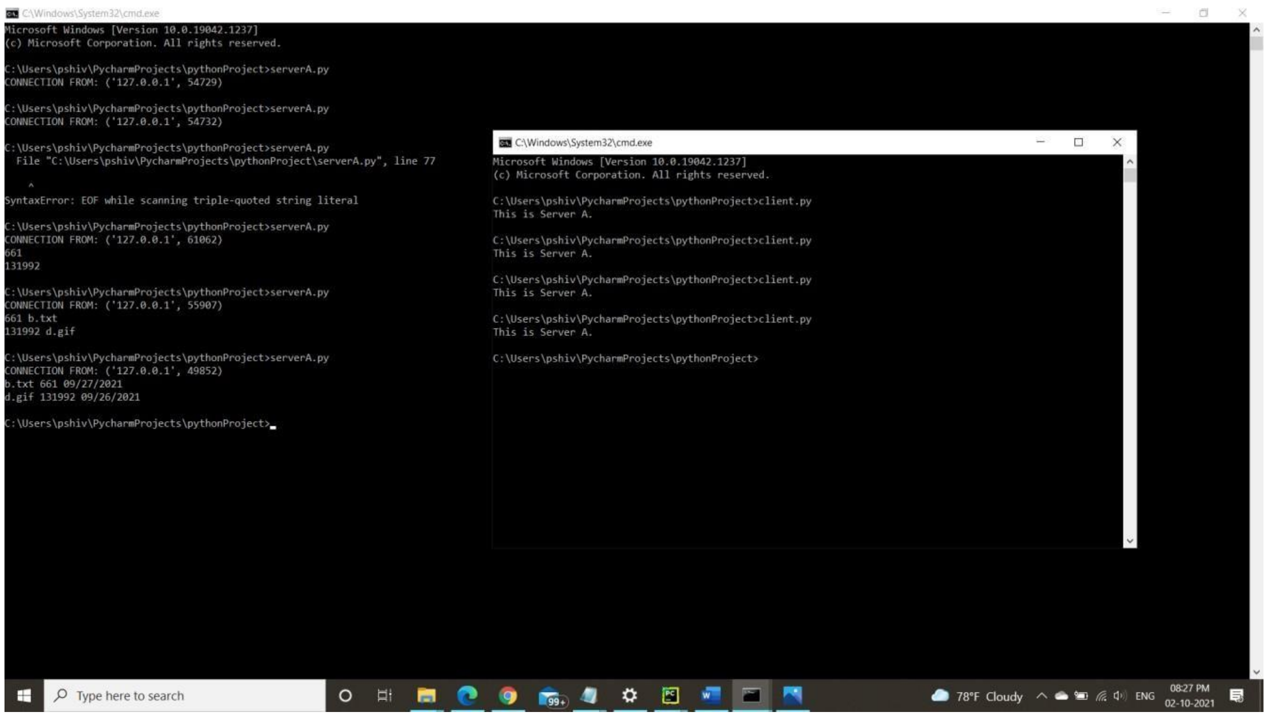
After that I create the for loop and join the file path with directory and display the last modification of date of file from the given reference.
https://thispointer.com/python-get-list-of-files-in-directory-sorted-by-date-and-time/

```
file_path = os.path.join(dir_name, file_name)# join the file path with directory
timestamp_str = time.strftime(  '%m/%d/%Y',      # last modification date of file
                              time.gmtime(os.path.getmtime(file_path)))
```

After that I display the size into byte using os.stat().st_size ae per given reference.
https://stackoverflow.com/questions/40783029/os-stat-st-size-gives-me-incorrect-size-in-python and https://www.journaldev.com/32067/how-to-get-file-size-in-python

```
files_with_size = (os.stat(file_path).st_size)      # Get file Size in bytes
```

In Output, I got Filename, Size, and Date as per below.



```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54729)

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54732)

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
  File "C:\Users\pshiv\PycharmProjects\pythonProject\serverA.py", line 77

    ^
SyntaxError: EOF while scanning triple-quoted string literal

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 61062)
661
131992

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 55907)
661 b.txt
131992 d.gif

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 49852)
b.txt 661 09/27/2021
d.gif 131992 09/26/2021

C:\Users\pshiv\PycharmProjects\pythonProject>
```

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>
```

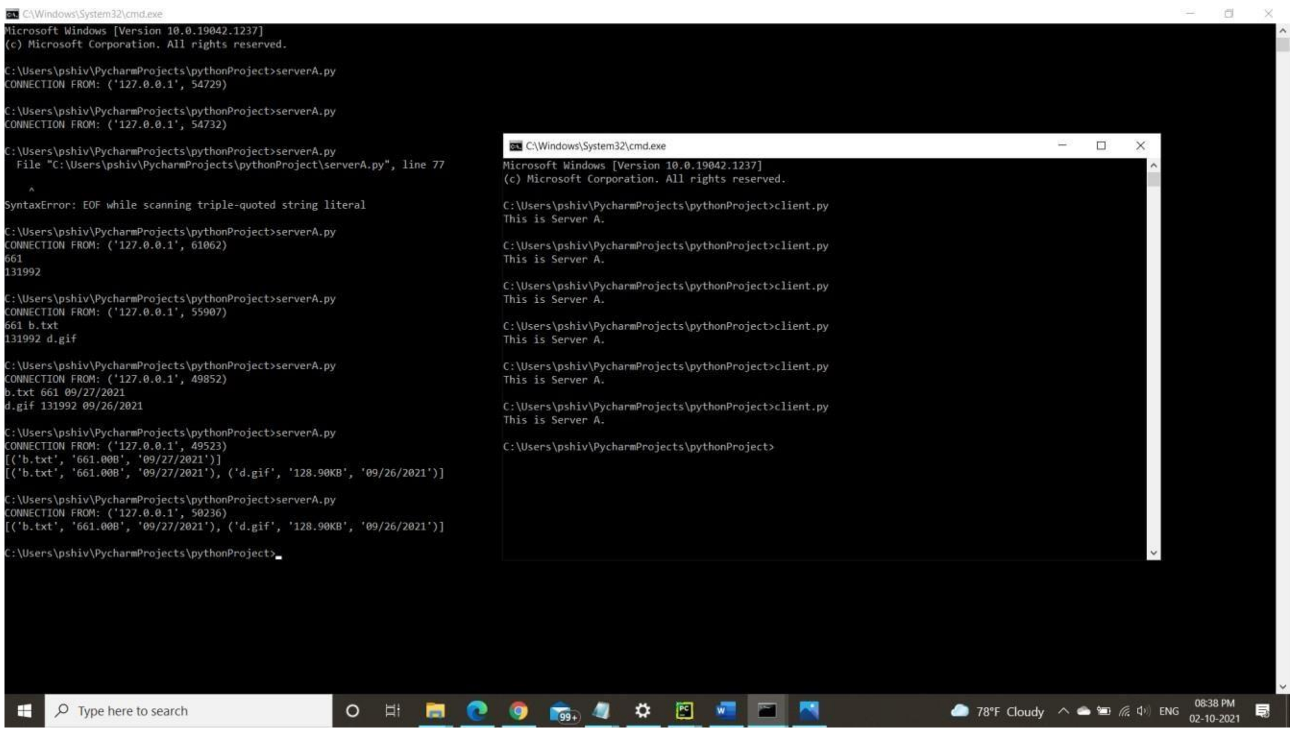
After got the size, I converted that size into Human readable size and convert into ‘KB’, ‘MB’, ‘GB’ etc. using given reference.

https://stackoverflow.com/questions/1094841/get-human-readable-version-of-file-size

```
def human_readable_size(size, decimal_places=2):    # Get human readable version of file size
    for unit in ['B', 'KB', 'MB', 'GB', 'TB']:
        if size < 1024.0:
            break
        size /= 1024.0
    return f"{size:.{decimal_places}f}{unit}"
human_size = (human_readable_size(files_with_size))
```

After applying above code, I append the data and got the output like this,

```
data1 = (file_name, human_size, timestamp_str)     # Get filename, human readable size, date into data
d.append(data1)             # append all three file into d
```



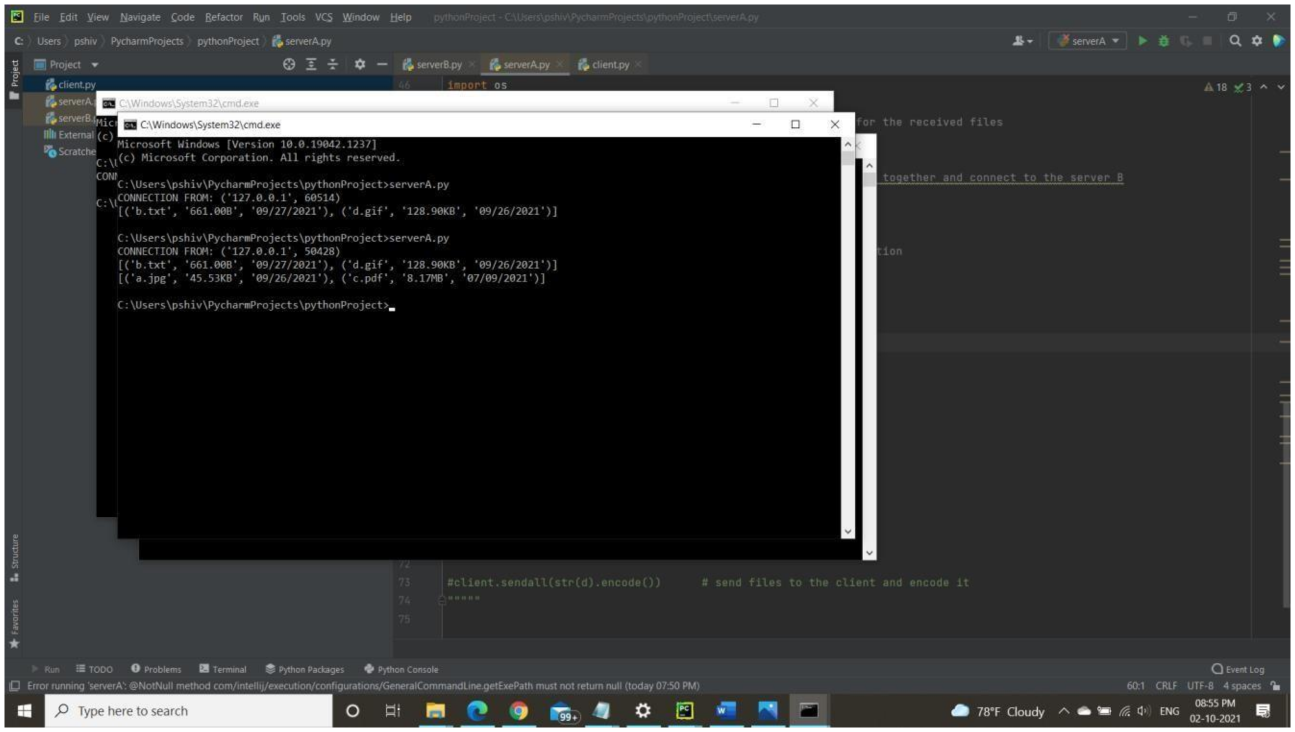
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54729)

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 54732)

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
  File "C:\Users\pshiv\PycharmProjects\pythonProject\serverA.py", line 77

    ^
SyntaxError: EOF while scanning triple-quoted string literal

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 61062)
661
131992

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 55907)
661 b.txt
131992 d.gif

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 49852)
b.txt 661 09/27/2021
d.gif 131992 09/26/2021

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 49523)
[('b.txt', '661.00B', '09/27/2021')]
[('b.txt', '661.00B', '09/27/2021'), ('d.gif', '128.90KB', '09/26/2021')]

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 50236)
[('b.txt', '661.00B', '09/27/2021'), ('d.gif', '128.90KB', '09/26/2021')]

C:\Users\pshiv\PycharmProjects\pythonProject>_
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>client.py
This is Server A.

C:\Users\pshiv\PycharmProjects\pythonProject>
```

After get the list of files with files metadata then I used same server A code for server B using different directory of folder and also write same client code for server B into the server A file.
After getting server B file I send that server B file to the client and encode it.

```
client.sendall(str(sending).encode())          # send files to the client and encode it
```

Now, Server A has both server files.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 60514)
[('b.txt', '661.00B', '09/27/2021'), ('d.gif', '128.90KB', '09/26/2021')]

C:\Users\pshiv\PycharmProjects\pythonProject>serverA.py
CONNECTION FROM: ('127.0.0.1', 50428)
[('b.txt', '661.00B', '09/27/2021'), ('d.gif', '128.90KB', '09/26/2021')]
[('a.jpg', '45.53KB', '09/26/2021'), ('c.pdf', '8.17MB', '07/09/2021')]

C:\Users\pshiv\PycharmProjects\pythonProject>_
```

```
#client.sendall(str(d).encode())      # send files to the client and encode it
"""""
```

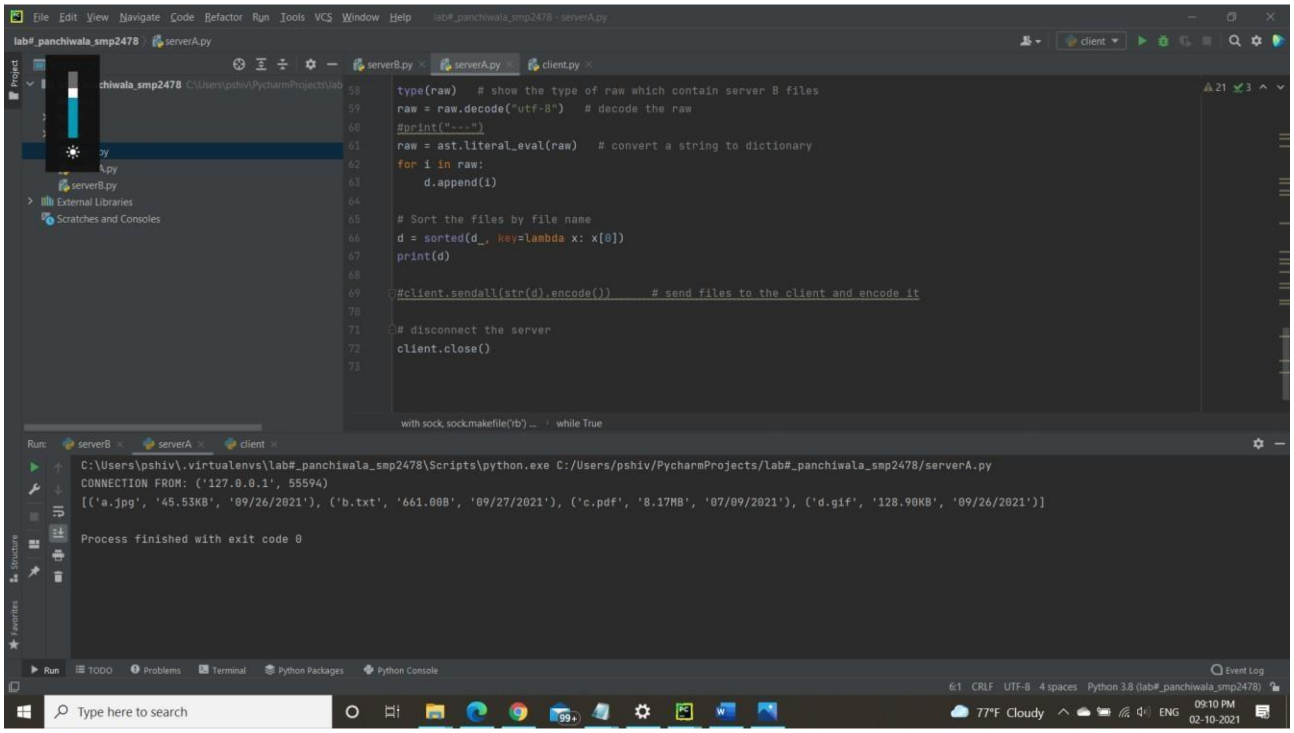
```
Error running 'serverA': @NotNull method com/intellij/execution/configurations/GeneralCommandLine.getExePath must not return null (today 07:50 PM)
```

After get the both server files on server A, I append both server files and sort the files by file name.

```
# Merge Server A and Server B file


type(raw)    # show the type of raw which contain server B files
raw = raw.decode("utf-8")    # decode the raw
#print("---")
raw = ast.literal_eval(raw)    # convert a string to dictionary
for i in raw:
    d.append(i)

# Sort the files by file name
d = sorted(d , key=lambda x: x[0])
```

For convert a string to dictionary format I used ast.literal_eval() using given reference.
https://www.kite.com/python/docs/ast.literal_eval

Then I got the output like this

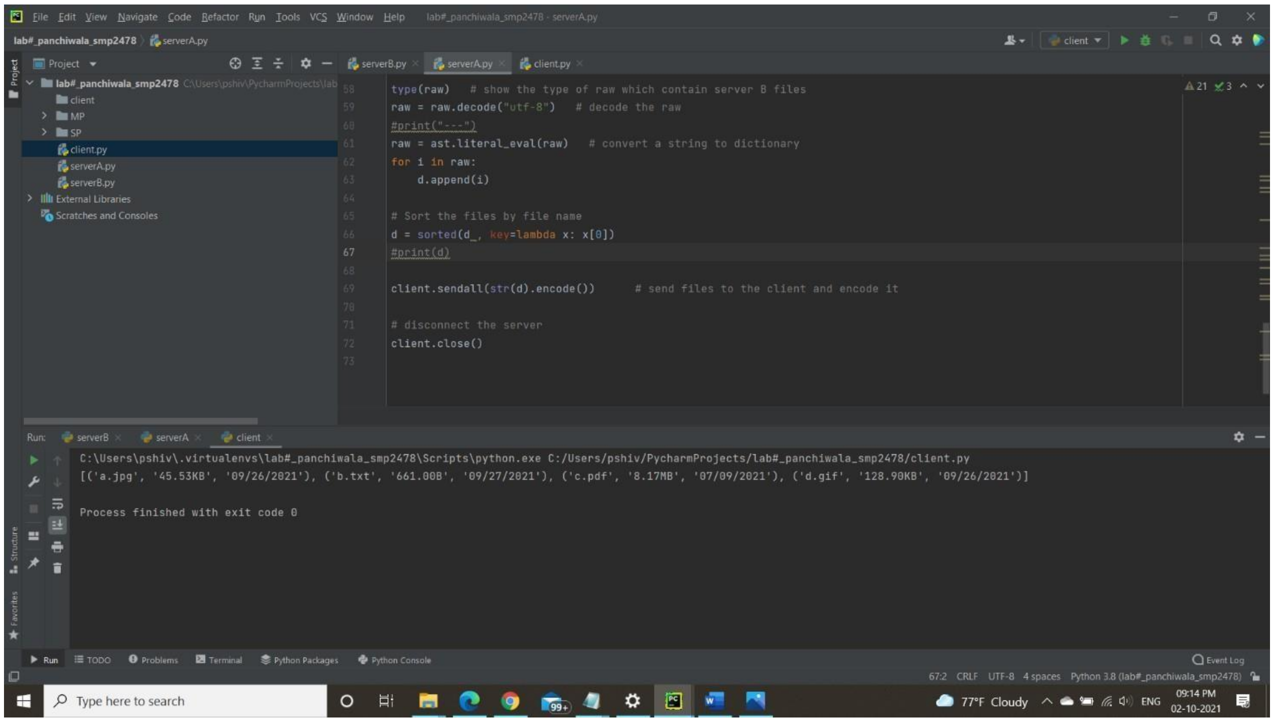


```
type(raw)   # show the type of raw which contain server B files
raw = raw.decode("utf-8")   # decode the raw
#print("---")
raw = ast.literal_eval(raw)   # convert a string to dictionary
for i in raw:
    d.append(i)

# Sort the files by file name
d = sorted(d_, key=lambda x: x[0])
print(d)

#client.sendall(str(d).encode())      # send files to the client and encode it

# disconnect the server
client.close()
```

```
C:\Users\pshiv\.virtualenvs\lab#_panchiwala_smp2478\Scripts\python.exe C:/Users/pshiv/PycharmProjects/lab#_panchiwala_smp2478/serverA.py
CONNECTION FROM: ('127.0.0.1', 55594)
[('a.jpg', '45.53KB', '09/26/2021'), ('b.txt', '661.00B', '09/27/2021'), ('c.pdf', '8.17MB', '07/09/2021'), ('d.gif', '128.90KB', '09/26/2021')]

Process finished with exit code 0
```

After that I send all the data from Server A to Client and encode it.

```
client.sendall(str(d).encode())      # send files to the client and encode it
```



```
type(raw)   # show the type of raw which contain server B files
raw = raw.decode("utf-8")   # decode the raw
#print("---")
raw = ast.literal_eval(raw)   # convert a string to dictionary
for i in raw:
    d.append(i)

# Sort the files by file name
d = sorted(d_, key=lambda x: x[0])
#print(d)

client.sendall(str(d).encode())      # send files to the client and encode it

# disconnect the server
client.close()
```

```
C:\Users\pshiv\.virtualenvs\lab#_panchiwala_smp2478\Scripts\python.exe C:/Users/pshiv/PycharmProjects/lab#_panchiwala_smp2478/client.py
[('a.jpg', '45.53KB', '09/26/2021'), ('b.txt', '661.00B', '09/27/2021'), ('c.pdf', '8.17MB', '07/09/2021'), ('d.gif', '128.90KB', '09/26/2021')]

Process finished with exit code 0
```

After sending files from Server A to the client, I got the all the files on Client.

In this Lab2, I modified the code based on lab 2. I need some functions which is required simultaneously. So, I created helper.py file and add that function in that file and given the reference in the code.

After that I used multithreading and create the function for sync the files from both server and list function for listing the files. Using the given reference.

https://analyticsindiamag.com/how-to-run-python-code-concurrently-using-multithreading/

https://docs.python.org/2/library/socketserver.html#module-SocketServer

For Sync the files, I give the directory path and used lambda and set function for syncing the files from Server B to A and A to B. Then I create a for loop to find the common files from both server. Reference from https://www.journaldev.com/37089/how-to-compare-two-lists-in-python

Then I checked that which one is latest file from both server's file.

I also write the function for that both server is alive and continuously work and syncing the files until if users manually closed it.

Reference from https://stackoverflow.com/questions/8627986/how-to-keep-a-socket-open-until-client-closes-it

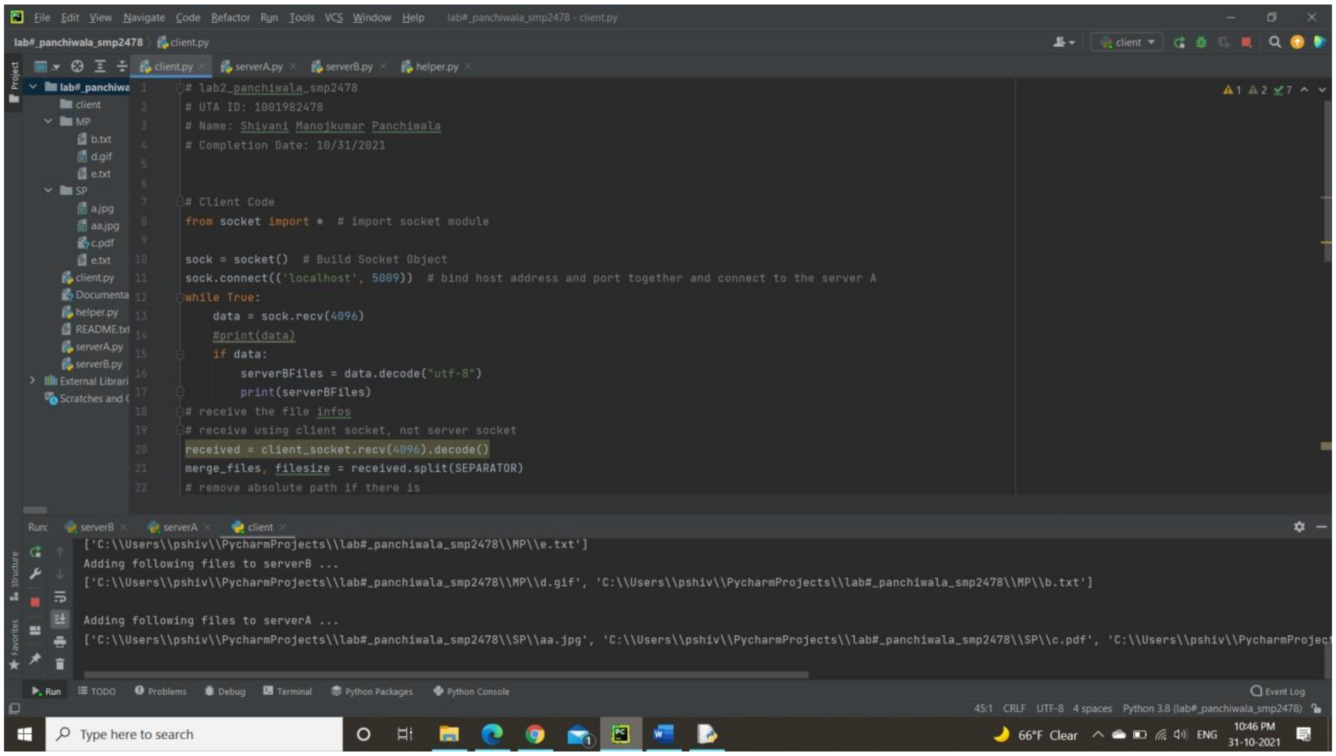
For keep the latest file to create function and check date and which one is latest file.

https://stackoverflow.com/questions/41635547/convert-python-datetime-to-timestamp-in-milliseconds

I also write the code for the read the data to client side and and write the data from server side to transfer the files. Using Reference https://www.thepythoncode.com/article/send-receive-files-using-sockets-python

After that Run the Server B then Server A and then Client. Both Server runs continuously and syncing the files and also give the notifications to any updates until the user kill the the both server manually.

So, It's continuously syncing the files. And show the updates.



```
# lab2_panchiwala_smp2478
# UTA ID: 1001982478
# Name: Shivani Manojkumar Panchiwala
# Completion Date: 10/31/2021


# Client Code
from socket import *  # import socket module

sock = socket()  # Build Socket Object
sock.connect(('localhost', 5009))  # bind host address and port together and connect to the server A
while True:
    data = sock.recv(4096)
    #print(data)
    if data:
        serverBFiles = data.decode("utf-8")
        print(serverBFiles)
# receive the file infos
# receive using client socket, not server socket
received = client_socket.recv(4096).decode()
merge_files, filesize = received.split(SEPARATOR)
# remove absolute path if there is
```

```
['C:\\Users\\pshiv\\PycharmProjects\\lab#_panchiwala_smp2478\\MP\\e.txt']
Adding following files to serverB ...
['C:\\Users\\pshiv\\PycharmProjects\\lab#_panchiwala_smp2478\\MP\\d.gif', 'C:\\Users\\pshiv\\PycharmProjects\\lab#_panchiwala_smp2478\\MP\\b.txt']

Adding following files to serverA ...
['C:\\Users\\pshiv\\PycharmProjects\\lab#_panchiwala_smp2478\\SP\\aa.jpg', 'C:\\Users\\pshiv\\PycharmProjects\\lab#_panchiwala_smp2478\\SP\\c.pdf', 'C:\\Users\\pshiv\\PycharmProjec
```

# Lab 3 Update

I used Direct Sync for syncing the files in python using below link https://stackoverflow.com/questions/52718889/can-dirsync-for-python-sync-files-and-folders-in-two-directions

After that I used pandas data frame for lock and unlock the files. I create a Functionality like Queue, Lock and Unlock.

```
serverA_que=Queue() # Create a Queue for server A
serverB_que=Queue() # Create a Queue for server B
client_que=Queue() # Create a Queue for Client
lock_que=Queue()  # Create Lock Queue
unlock_que=Queue() # Create Unlock Queue
```

Then I receive the data from server to server A and load that data using the pickle. https://sites.pitt.edu/~naraehan/python2/pickling.html

https://datascienceparichay.com/article/read-pickle-file-as-pandas-dataframe/

After that I create a function for check the modification like Adding , Removing, modifying the data.

After that I create update and append and Check_queue function.

https://stackoverflow.com/questions/34595041/fcntl-file-lock-example-not-working

After that I Append the data

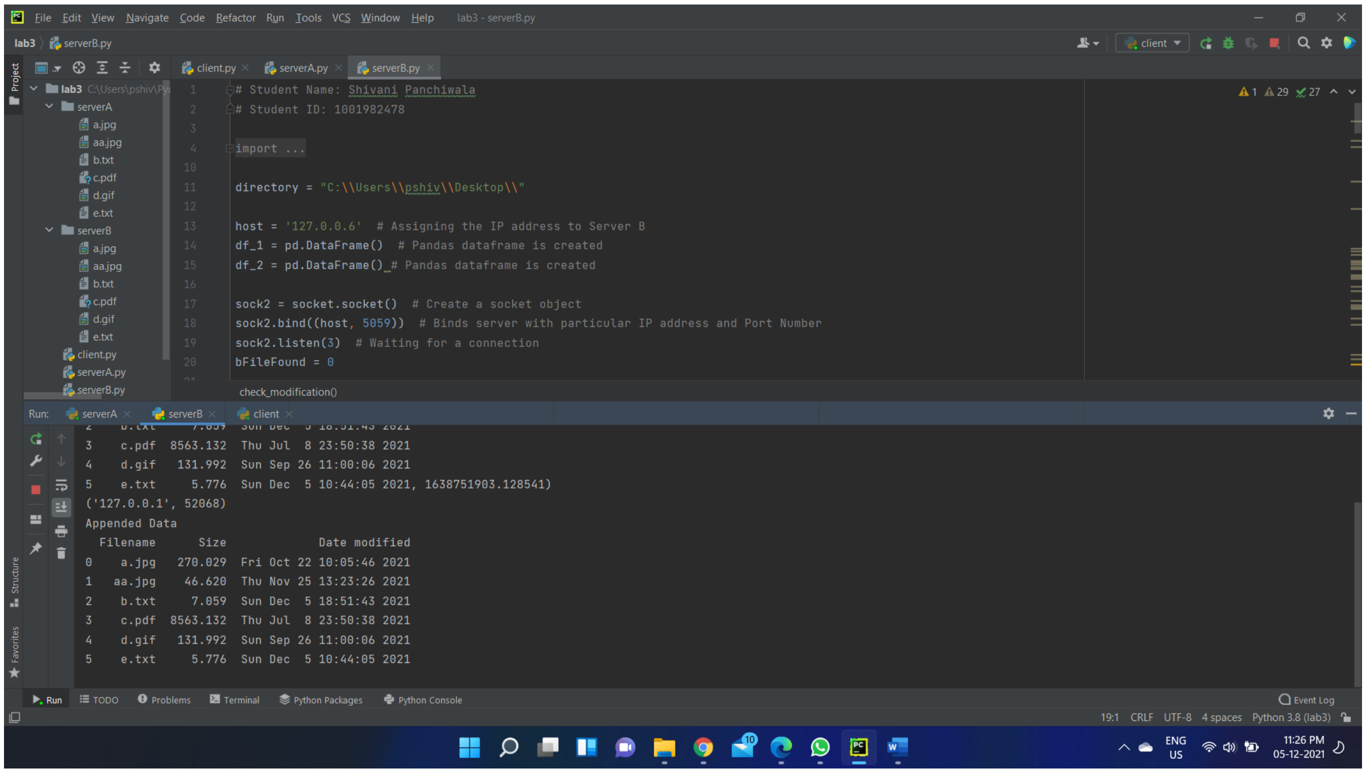
```
def append(server1data, server2data): # Function to append data from both the servers
    t = server1data.merge(server2data, 'outer', on='Filename')
    t = t.replace(np.NaN, 0)
    final_data=pd.DataFrame(columns=['Filename','Size','Date modified'])
    for values in t[['Date modified_x', 'Filename' , 'Size(KB)_x', 'Date modified_y', 'Size(KB)_y']].values:
        row_dict = {"Filename": values[1]}

        if values[0] == 0:
            row_dict['Date modified'] = values[3]
            row_dict['Size'] = values[4]

        elif values[3] == 0:
            row_dict['Date modified'] = values[0]
            row_dict['Size'] = values[2]

        elif datetime.datetime.strptime(values[0], '%c') > datetime.datetime.strptime(values[3], '%c'):
            row_dict['Date modified'] = values[0]
            row_dict['Size'] = values[2]

        else:
            row_dict['Date modified'] = values[3]
            row_dict['Size'] = values[4]

        final_data = final_data.append(row_dict, ignore_index=True)

    return final_data
```

Then I check all the queue step by step

```
def check_queue(q1,q2,q3,lockq,unlockq): # A function to check queues for the
changes made in the directory
    servera_data = pd.DataFrame()
    serverb_data = pd.DataFrame()

    while True:
        if not q1.empty() and not q2.empty():
            servera_data = q1.get()
            serverb_data = q2.get()
            combined_data=append(servera_data, serverb_data)
            print("Appended Data")
            print(combined_data)
            send_data(combined_data)
            q3.put(combined_data)
            break

    while True:
        if not lockq.empty():
            print("====== lockq accessed ============================")
            if "locked/unlocked" not in combined_data.columns:
                combined_data["locked/unlocked"]=["" for i in
range(combined_data.shape[0])]

            locked=lockq.get()
            print(locked)
            combined_data['locked/unlocked'] = ["locked" if filenme== locked else
value for filenme, value in combined_data[['Filename', 'locked/unlocked']].values]
            q3.put(combined_data)
            print(combined_data)

        if not unlockq.empty():
            print("====== unlockq accessed ==================================")
            unlocked=unlockq.get()
            print(unlocked)
            combined_data['locked/unlocked'] = ["unlocked" if filenme== unlocked
else value for filenme, value in combined_data[['Filename',
'locked/unlocked']].values]
            combined_data_ = append(servera_data, serverb_data)
            combined_data = update(combined_data, combined_data_)
            q3.put(combined_data)
            print(combined_data)

        if not q2.empty():
            print("Modification in ServerB")
            serverb_data = q2.get()
            combined_data_ = append(servera_data, serverb_data)
            print("Appended Data")
            # print(combined_data_)
            send_data(combined_data_)
            combined_data = update(combined_data, combined_data_)
            print(combined_data)
            q3.put(combined_data)
```

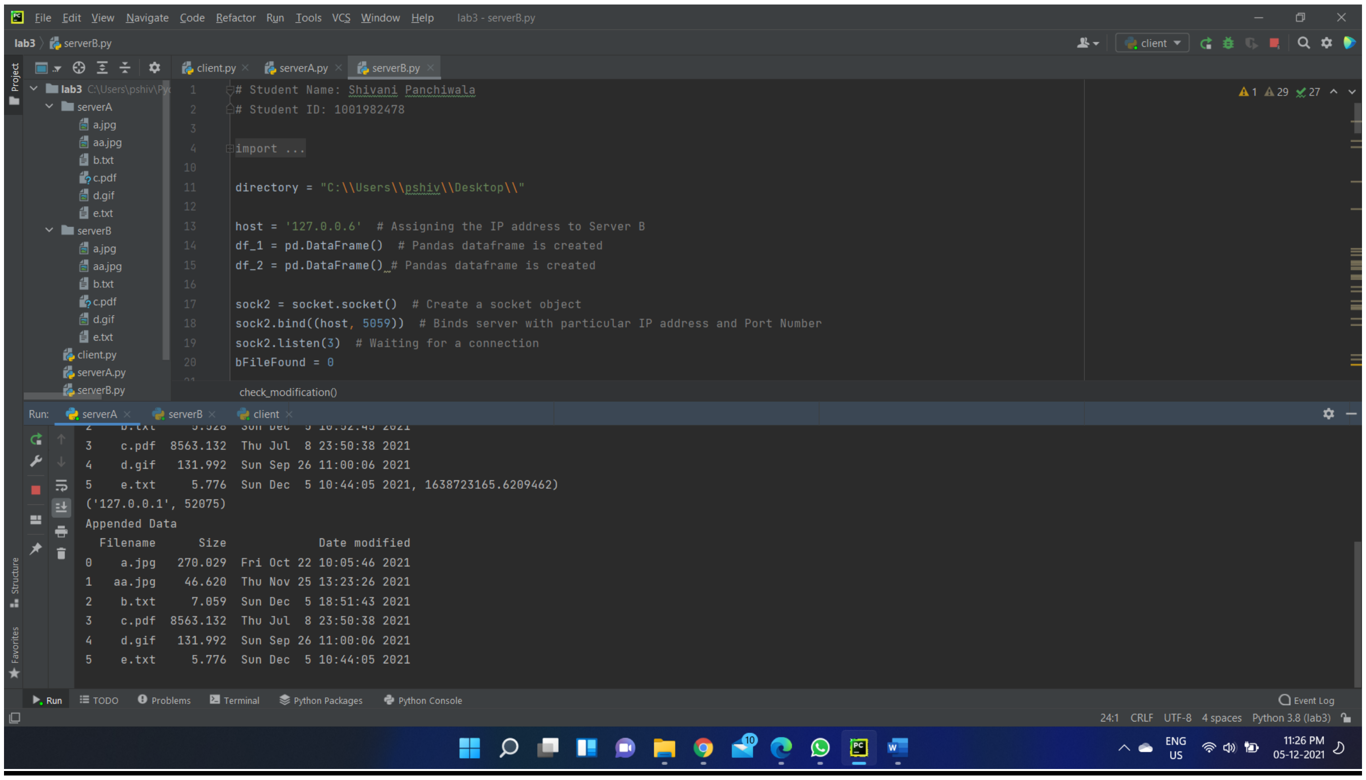
```python
        if not q1.empty():
            print("Modification in ServerA")
            servera_data = q1.get()
            combined_data_ = append(servera_data, serverb_data)
            print("Appended Data")
            #print(combined_data_)
            send_data(combined_data_)
            combined_data = update(combined_data, combined_data_)
            print(combined_data)
            q3.put(combined_data)
```

After print the list of file in server B and server A



```python
# Student Name: Shivani Panchiwala
# Student ID: 1001982478

import ...

directory = "C:\\Users\\pshiv\\Desktop\\"

host = '127.0.0.6'  # Assigning the IP address to Server B
df_1 = pd.DataFrame()  # Pandas dataframe is created
df_2 = pd.DataFrame() # Pandas dataframe is created

sock2 = socket.socket()  # Create a socket object
sock2.bind((host, 5059))  # Binds server with particular IP address and Port Number
sock2.listen(3)  # Waiting for a connection
bFileFound = 0
```

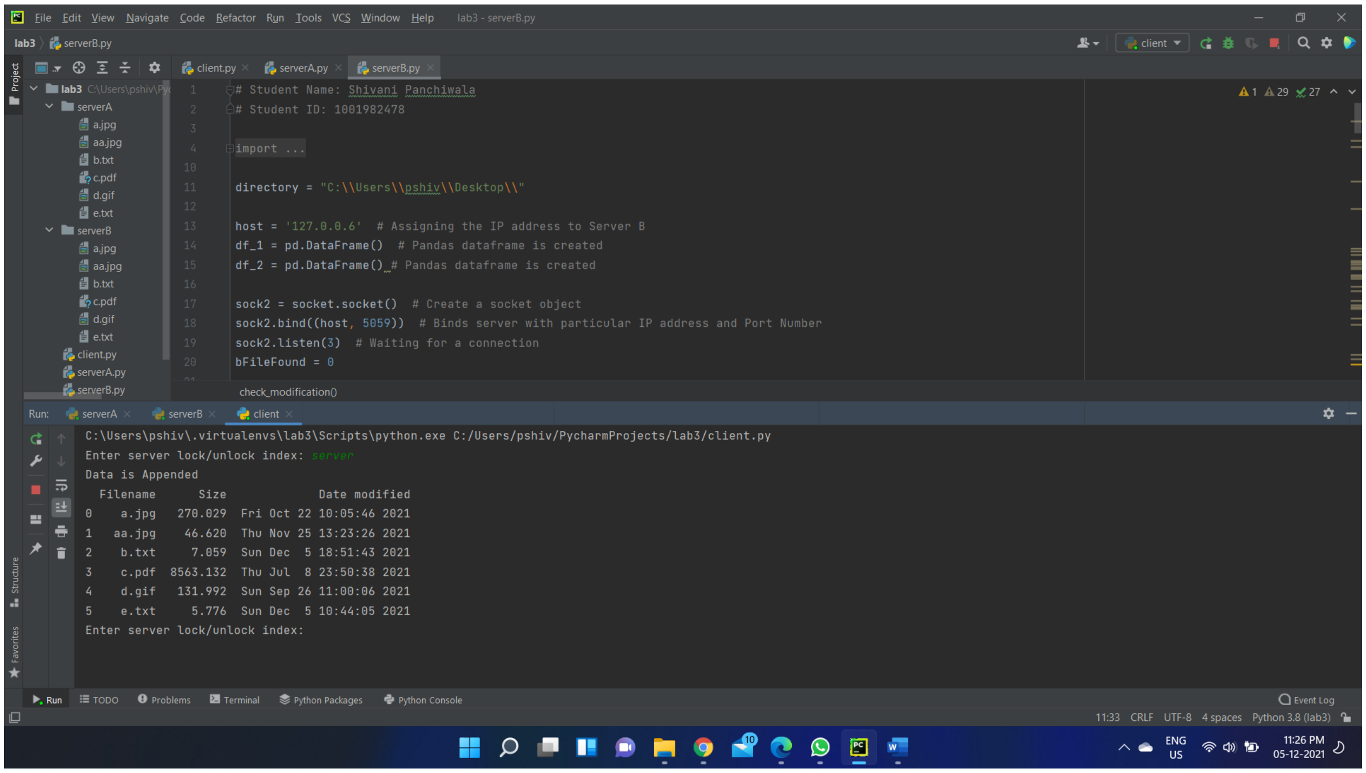
```
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021, 1638751903.128541)
('127.0.0.1', 52068)
Appended Data
  Filename      Size              Date modified
0    a.jpg   270.029  Fri Oct 22 10:05:46 2021
1   aa.jpg    46.620  Thu Nov 25 13:23:26 2021
2    b.txt     7.059  Sun Dec  5 18:51:43 2021
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
```



```
# Student Name: Shivani Panchiwala
# Student ID: 1001982478

import ...

directory = "C:\\Users\\pshiv\\Desktop\\"

host = '127.0.0.6'  # Assigning the IP address to Server B
df_1 = pd.DataFrame()  # Pandas dataframe is created
df_2 = pd.DataFrame() # Pandas dataframe is created

sock2 = socket.socket()  # Create a socket object
sock2.bind((host, 5059))  # Binds server with particular IP address and Port Number
sock2.listen(3)  # Waiting for a connection
bFileFound = 0
```

```
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021, 1638723165.6209462)
('127.0.0.1', 52075)
Appended Data
  Filename      Size              Date modified
0    a.jpg   270.029  Fri Oct 22 10:05:46 2021
1   aa.jpg    46.620  Thu Nov 25 13:23:26 2021
2    b.txt     7.059  Sun Dec  5 18:51:43 2021
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
```

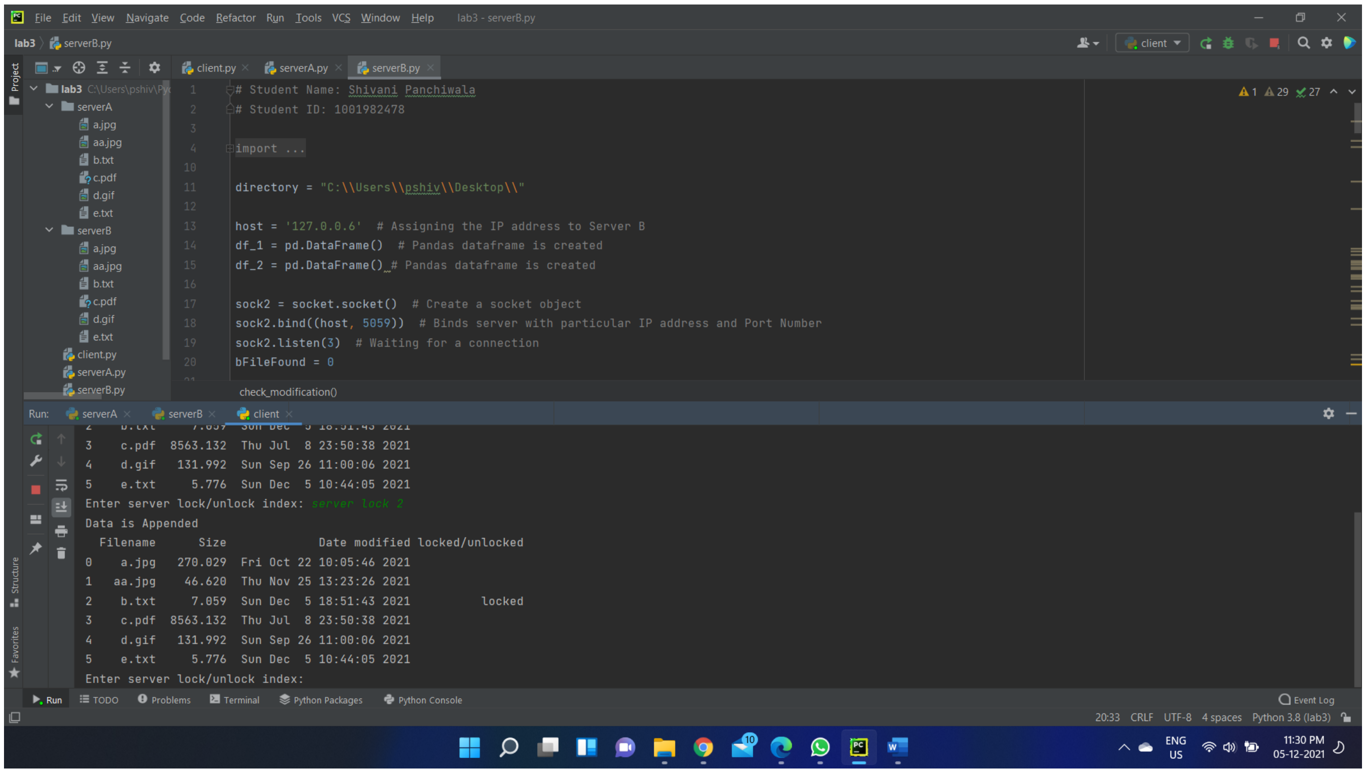
After that on Client type only “server” to get the list of files



```
# Student Name: Shivani Panchiwala
# Student ID: 1001982478

import ...

directory = "C:\\Users\\pshiv\\Desktop\\"

host = '127.0.0.6'  # Assigning the IP address to Server B
df_1 = pd.DataFrame()  # Pandas dataframe is created
df_2 = pd.DataFrame() # Pandas dataframe is created

sock2 = socket.socket()  # Create a socket object
sock2.bind((host, 5059))  # Binds server with particular IP address and Port Number
sock2.listen(3)  # Waiting for a connection
bFileFound = 0
```

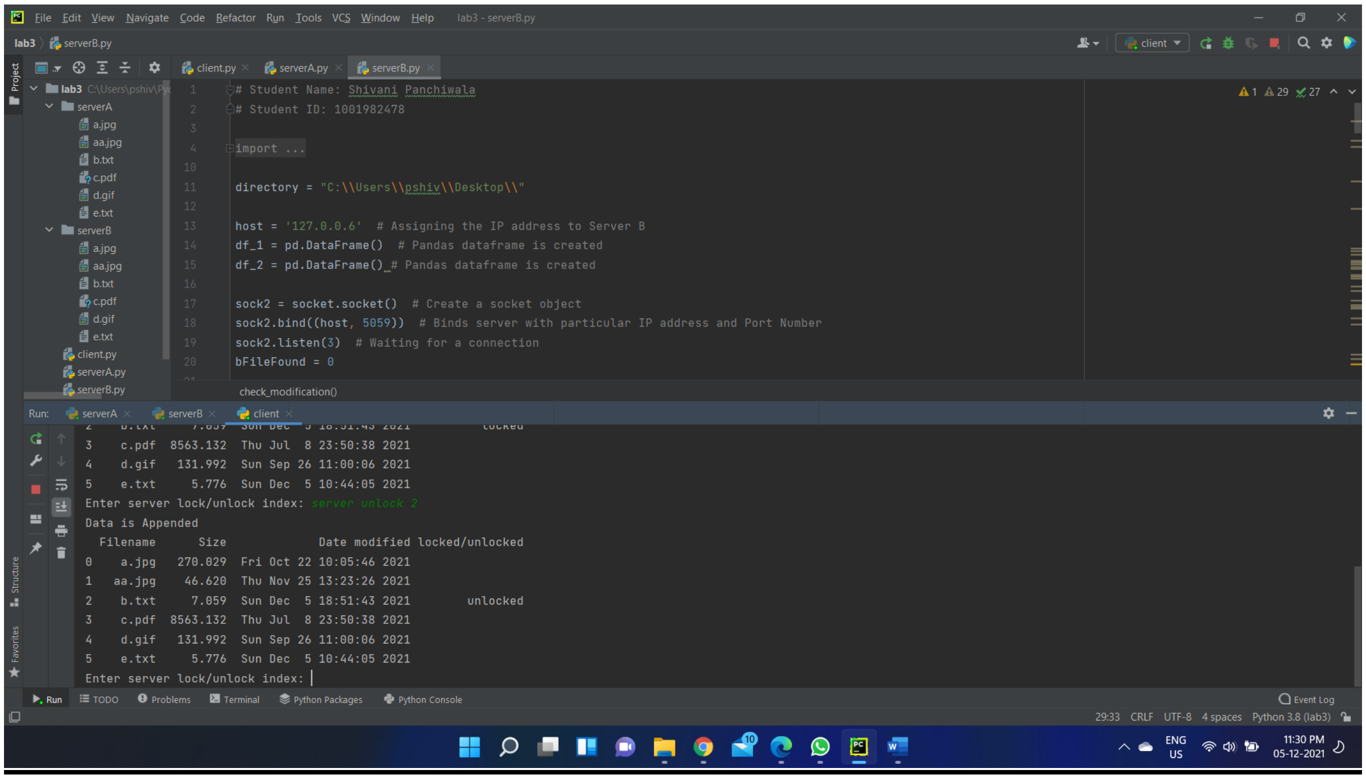
```
C:\Users\pshiv\.virtualenvs\lab3\Scripts\python.exe C:/Users/pshiv/PycharmProjects/lab3/client.py
Enter server lock/unlock index: server
Data is Appended
  Filename      Size              Date modified
0    a.jpg   270.029  Fri Oct 22 10:05:46 2021
1   aa.jpg    46.620  Thu Nov 25 13:23:26 2021
2    b.txt     7.059  Sun Dec  5 18:51:43 2021
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
Enter server lock/unlock index:
```

After that enter “server lock index” to lock the files. Ang get this result.



```
# Student Name: Shivani Panchiwala
# Student ID: 1001982478

import ...

directory = "C:\\Users\\pshiv\\Desktop\\"

host = '127.0.0.6'  # Assigning the IP address to Server B
df_1 = pd.DataFrame()  # Pandas dataframe is created
df_2 = pd.DataFrame() # Pandas dataframe is created

sock2 = socket.socket()  # Create a socket object
sock2.bind((host, 5059))  # Binds server with particular IP address and Port Number
sock2.listen(3)  # Waiting for a connection
bFileFound = 0
```

```
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
Enter server lock/unlock index: server lock 2
Data is Appended
  Filename      Size           Date modified locked/unlocked
0    a.jpg   270.029  Fri Oct 22 10:05:46 2021
1   aa.jpg    46.620  Thu Nov 25 13:23:26 2021
2    b.txt     7.059  Sun Dec  5 18:51:43 2021          locked
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
Enter server lock/unlock index:
```



```
# Student Name: Shivani Panchiwala
# Student ID: 1001982478

import ...

directory = "C:\\Users\\pshiv\\Desktop\\"

host = '127.0.0.6'  # Assigning the IP address to Server B
df_1 = pd.DataFrame()  # Pandas dataframe is created
df_2 = pd.DataFrame() # Pandas dataframe is created

sock2 = socket.socket()  # Create a socket object
sock2.bind((host, 5059))  # Binds server with particular IP address and Port Number
sock2.listen(3)  # Waiting for a connection
bFileFound = 0
```

```
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
Enter server lock/unlock index: server unlock 2
Data is Appended
  Filename      Size           Date modified locked/unlocked
0    a.jpg   270.029  Fri Oct 22 10:05:46 2021
1   aa.jpg    46.620  Thu Nov 25 13:23:26 2021
2    b.txt     7.059  Sun Dec  5 18:51:43 2021        unlocked
3    c.pdf  8563.132  Thu Jul  8 23:50:38 2021
4    d.gif   131.992  Sun Sep 26 11:00:06 2021
5    e.txt     5.776  Sun Dec  5 10:44:05 2021
Enter server lock/unlock index:
```

Then File is unlock like this.

HOW TO IMPLEMENT THE PROJECT

- Open the Command Prompt and write the **serverB.py**
- After that open another Command Prompt and write the **serverA.py**
- And open again another Command Prompt and write **client.py**
- After that Write server on client then get the list of files and then write server lock 2 where 2 is index and for unlock write server lock 2

REFERENCES

1) https://stackoverflow.com/questions/47539028/transfer-contents-of-a-folder-over-network-by-python
2) https://thispointer.com/python-get-list-of-files-in-directory-sorted-by-date-and-time/
3) https://stackoverflow.com/questions/40783029/os-stat-st-size-gives-me-incorrect-size-in-python
4) https://www.journaldev.com/32067/how-to-get-file-size-in-python
5) https://stackoverflow.com/questions/1094841/get-human-readable-version-of-file-size
6) https://www.kite.com/python/docs/ast.literal_eval
7) https://analyticsindiamag.com/how-to-run-python-code-concurrently-using-multithreading/
8) https://docs.python.org/2/library/socketserver.html#module-SocketServer
9) https://www.journaldev.com/37089/how-to-compare-two-lists-in-python
10) https://stackoverflow.com/questions/8627986/how-to-keep-a-socket-open-until-client-closes-it
11) https://stackoverflow.com/questions/41635547/convert-python-datetime-to-timestamp-in-milliseconds
12) https://www.thepythoncode.com/article/send-receive-files-using-sockets-python
13) **https://stackoverflow.com/questions/52718889/can-dirsync-for-python-sync-files-and-folders-in-two-directions**
14) https://sites.pitt.edu/~naraehan/python2/pickling.html
15) https://datascienceparichay.com/article/read-pickle-file-as-pandas-dataframe/
16) https://stackoverflow.com/questions/34595041/fcntl-file-lock-example-not-working