Assignment - 05

1) **Write a program to Perform File Transfer in Client & Server Using TCP/IP.**

Server side code:

```
import socket
import tqdm
import os

BUFFER_SIZE = 1024
s = socket.socket()
s.bind(('127.0.0.1', 5001))
s.listen(1)
print("Listening as 127.0.0.1:5001")
client_socket, address = s.accept()
print(f"Connected to {address[0]}:{str(address[1])}")
filename, filesize =
client_socket.recv(BUFFER_SIZE).decode().split('||')
filename = os.path.basename(filename)
filesize = int(filesize)
progress = tqdm.tqdm(range(filesize), f"Receiving
{filename}", unit="B", unit_scale=True,
unit_divisor=1024)
with open(filename, "wb") as f:
    while True:
        bytes_read = client_socket.recv(BUFFER_SIZE)
        if not bytes_read:
            break
        f.write(bytes_read)
        progress.update(len(bytes_read))
client_socket.close()
s.close()
```

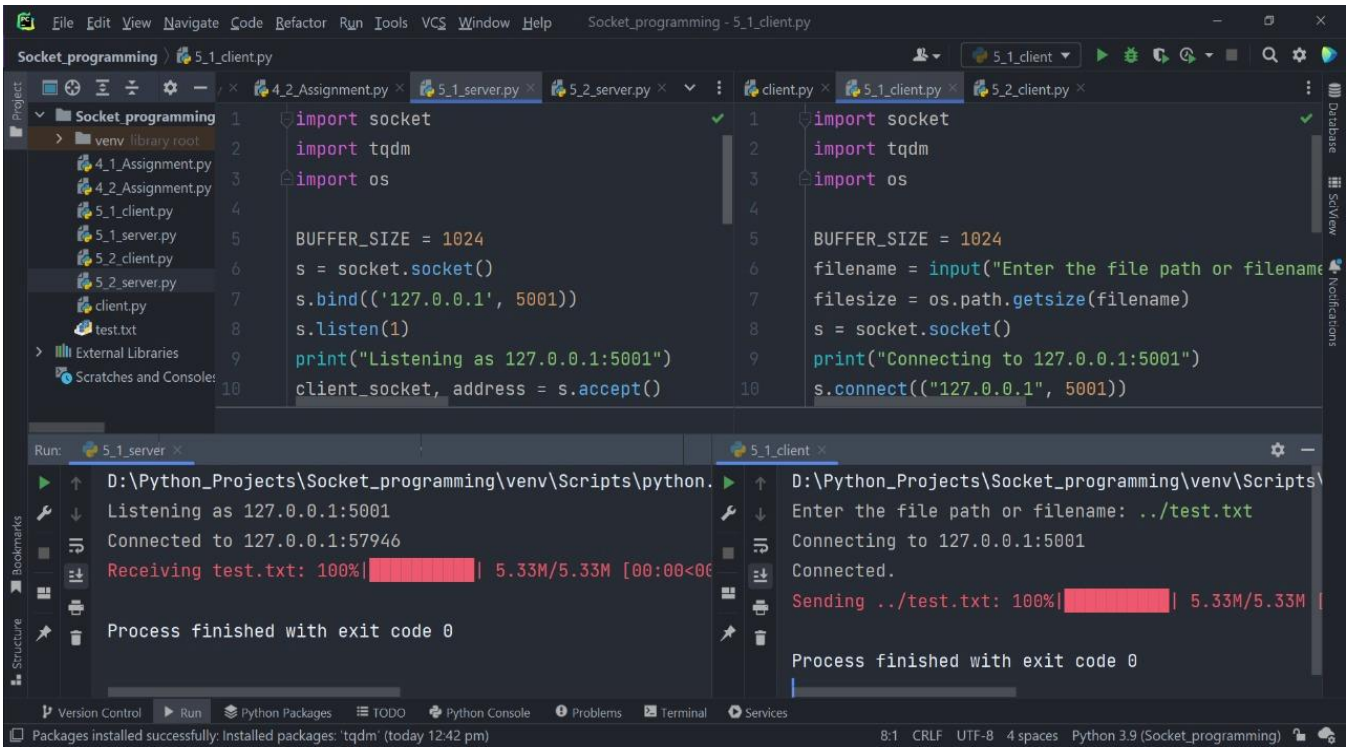
Client side code:

```
import socket
import tqdm
import os

BUFFER_SIZE = 1024
filename = input("Enter the file path or filename: ")
filesize = os.path.getsize(filename)
s = socket.socket()
print("Connecting to 127.0.0.1:5001")
s.connect(("127.0.0.1", 5001))
print("Connected.")
s.send(f"{filename}||{filesize}".encode())
```

```
progress = tqdm.tqdm(range(filesize), f"Sending
{filename}", unit="B", unit_scale=True,
unit_divisor=1024)
with open(filename, "rb") as f:
    while True:
        bytes_read = f.read(BUFFER_SIZE)
        if not bytes_read:
            break
        s.sendall(bytes_read)
        progress.update(len(bytes_read))
s.close()
```

Output:



2) Write a program to implement Remote Command Execution (RCE)

Server side code:

```
import socket
import os

BUFFER_SIZE = 4096
s = socket.socket()
s.bind(('127.0.0.1', 5001))
s.listen(1)
print("Listening as 127.0.0.1:5001")
client_socket, address = s.accept()
print(f"{address[0]}:{str(address[1])} is Connected to
terminal")
while True:
    print("\nClient@Server>>", end=" ")
```

```
    command = client_socket.recv(BUFFER_SIZE).decode()
    print(command)
    if command == 'exit':
        break
    os.system(command)
print("Closing remote connection with client")
client_socket.close()
s.close()
```

Client side code:

```
import socket

BUFFER_SIZE = 4096
s = socket.socket()
print("Connecting to 127.0.0.1:5001")
s.connect(("127.0.0.1", 5001))
print("Connected to Server Terminal")
while True:
    command = input("\nServer>> ")
    s.sendall(command.encode())
    if command == 'exit':
        break
print("Closing remote connection with Server")
s.close()
```

Output: