

**Network mini project- Socket programming (Client-Server)**

**Dr. Iyad Tumar**

**Partners:**

**Mustafa Bi’rat 1160813**

**Nidal Dhabrah 1161274**

**Report Date: 5/3/2019**

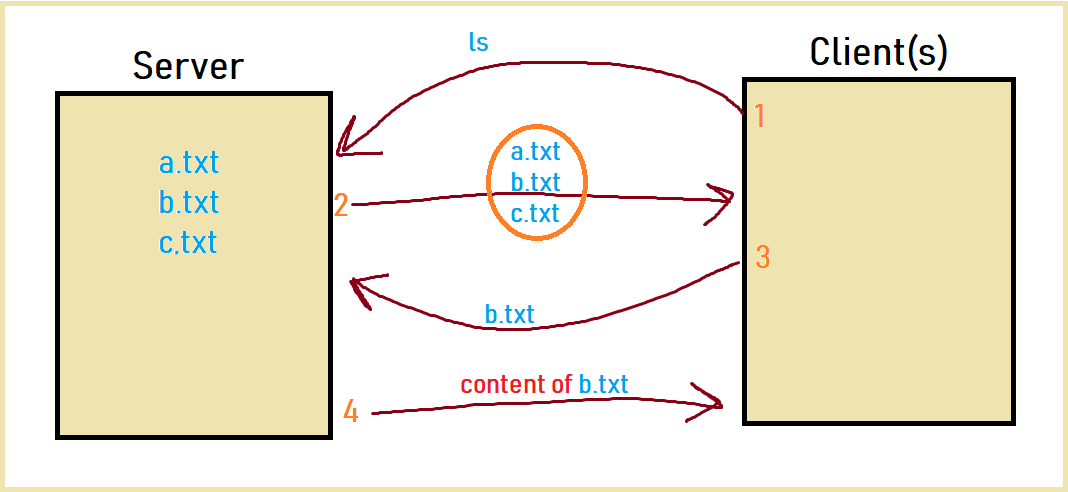
**Programming Language used:**

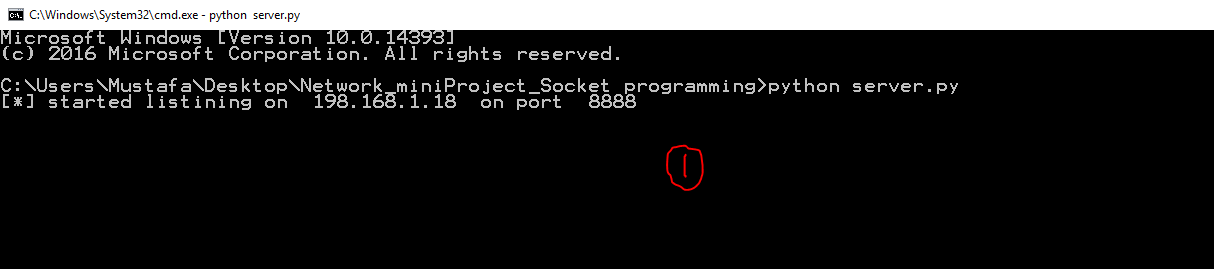
**Python**

**Network mini project (client-server)**

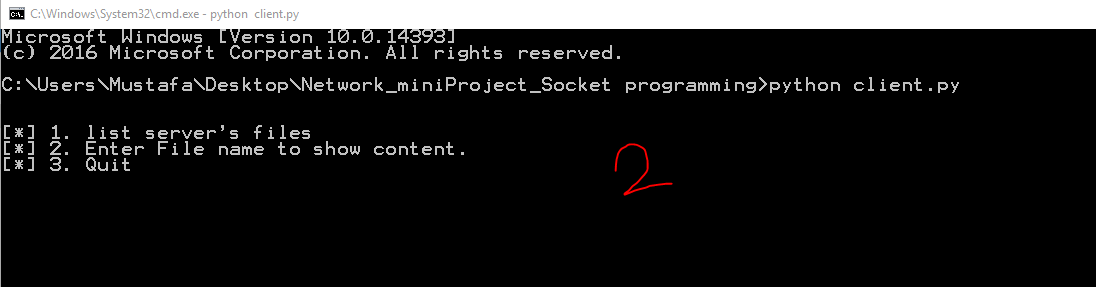
**Main idea:** The main idea of the project is to create an implementation of client-server model (Socket-programming)… that there will be .txt files on the server and the client will have menu to choose from which includes

1. Ls ”list txt files on the server”
2. Get file’s content “Get the content of given file if it’s valid one”
3. Exit

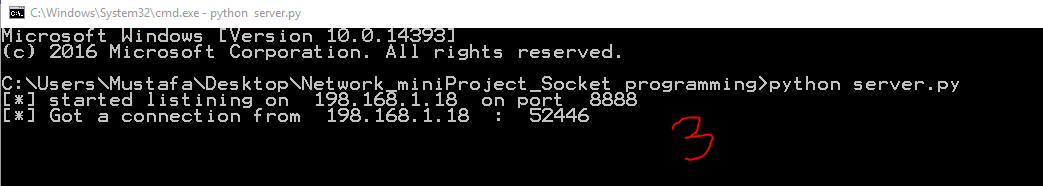




1. **Run server.py script (server script), by making a TCP socket and listen to localhost IP on port 8888(we chose it randomly) this script starts the server and keep listening for a client connection.**

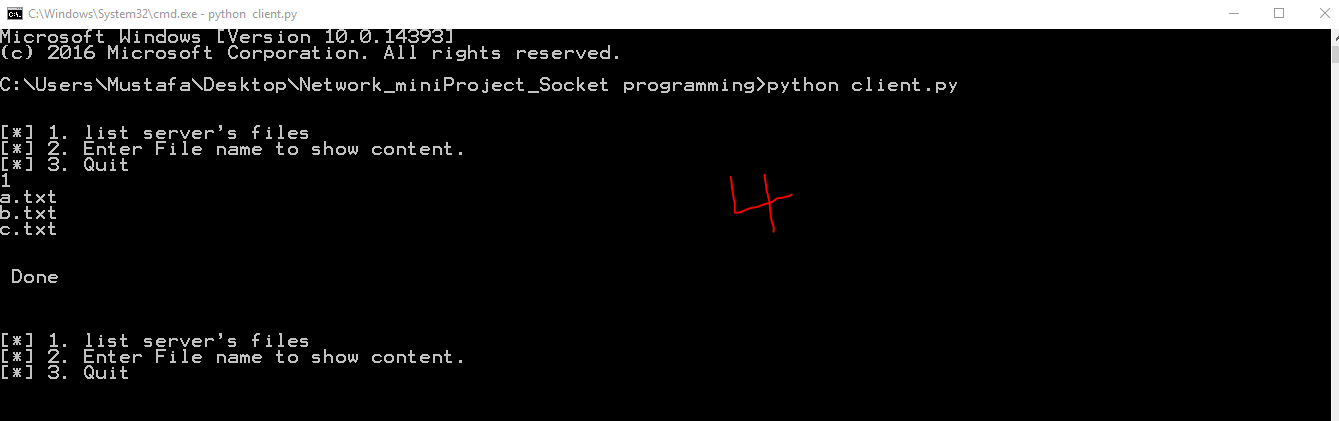


1. **Run client.py script (client script), client here send connection request to the server, 3-way handshake is handled here.**

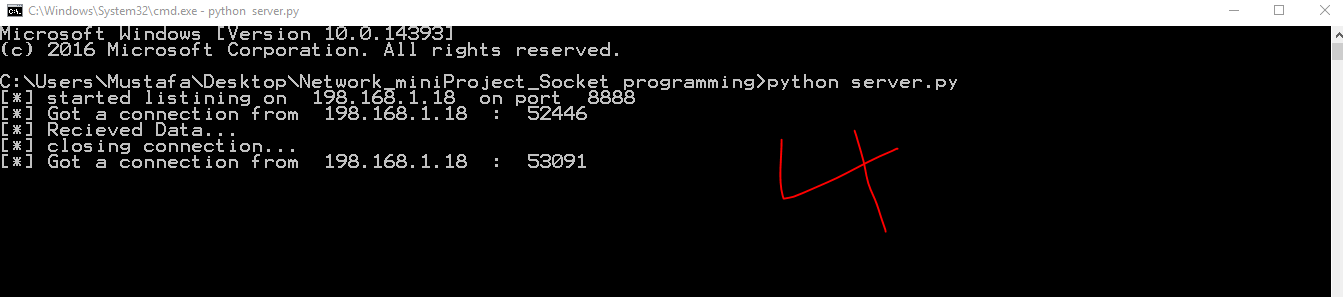


1. **The server receives the request as part of the 3-way handshake and then connects to the client through printing out the message “Got connection from local IP on random available port number”.**

**Client:**

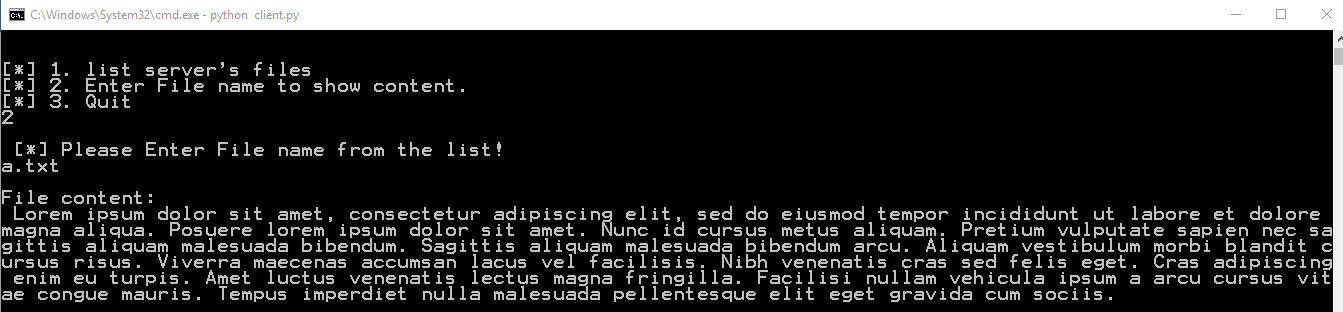


**Server**

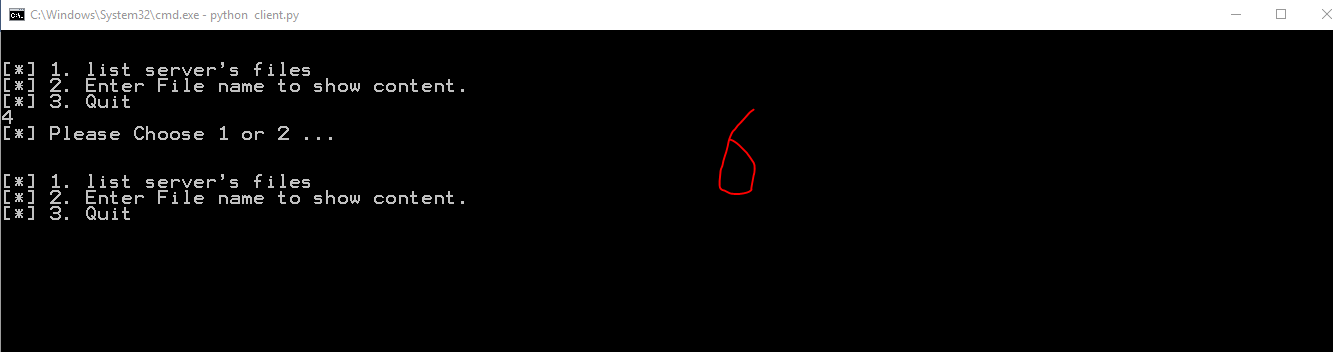


1. **The client then sends request for example (sends 1 or “ls”) to the server in order to get a list of txt files located on the server. The request arrives to the server, server then prints message saying “Received Connection” from the client and then sends back a list of txt files located on the server. The client receives the response and prints it out to the council. Client sends ack to the server that the data has been received well and the client is willing to close the connection, then the server receives the ack with closing the connection.**

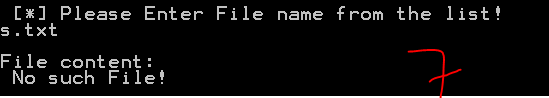
**Let us take another scenario:**



1. **The client asks for the content of a certain file on the server… as above sends request and receives response and then sends an ack with closing….**

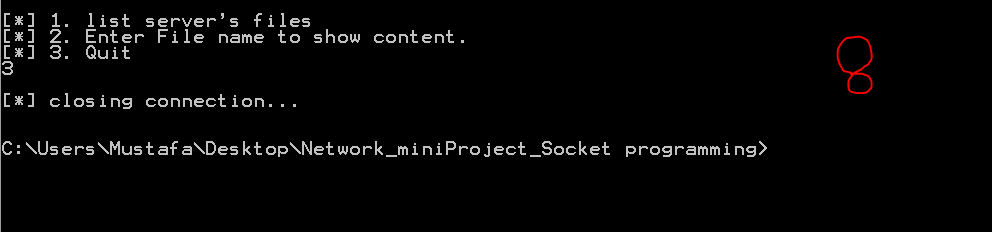


1. **If the client enters wrong choice or wrong syntax**



1. **If the file that the client asked for is not valid file….**

**Client:**

**Server:**



1. **If the client is willing to end the script…..**