**Computer Communications and Networks**

**Project-1**

**HTTP Client and Server**

**Arun Kunnumpuram Thomas -801027386**

**Sanjay Tiwari – 800968058**

**Implementation:**

**We have implemented HTTP Client and Server using Java. Socket Programming is implemented for communication between Client and Server. GET and PUT methods are implemented in this project.**

If client generates a GET request then the file is fetched from server and displayed on the client side. For fetching the file, various steps are involved -

1. **Establish a TCP connection with the server.**
2. **Send a valid HTTP/1.1 GET request to the server having the intended file name.**
3. **Upon receiving the request, server will look for the file.**
4. **If the file is present, it will generate a 200 OK response along with file contents.**
5. **If the file is not present, it will generate a 404 Not Found message.**

**If it is a PUT request, the file is sent to the server and saved locally on server. Steps involved in PUT request are –**

1. **Establish a TCP connection with the server.**
2. **Send a valid HTTP/1.1 PUT request to the server.**
3. **File will be sent to the Server.**
4. **If the file is saved successfully, server will respond back to client with 200 OK File Created.**

**HTTP Client:**

1. **We created a file called MyClient.java to implement HTTP Client.**
2. **In the command line, we are passing parameters as-**

**MyClient HostName Port Method FileName**

1. **Client program will check whether the parameters are correct or not. If incorrect, it will throw an error.**
2. **Client will create a socket to connect to the Server.**
3. **If Method is GET, then requested string is passed to the Server using PrintStream.**
4. **Response of GET is handled by using DataInputStream.**
5. **If Method is PUT, then file is read using FileInputStream and passed using PrintStream.**
6. **Response of PUT is handled by DataInputStream.**

**HTTP Server:**

1. **We created a file called MyServer.java to implement HTTP Server.**
2. **In the command line, we are passing only one parameter as –**

**MyServer 8080**

1. **Server program will check whether number of parameter is one or not. If incorrect, it will throw an error.**
2. **After passing correct input parameter on command line, server will start listening for connection.**
3. **Server creates a separate thread for handling each client.**
4. **On getting GET request from Client, Server will look for the file.**
5. **If file is not present, then it will generate a 404 Not Found response to the client.**
6. **If the file is present, then it will read file content using FileInputStream class and send the file to the client along with 200 OK message.**
7. **On getting PUT request, Server will read the file using DataInputStream class and write the file using FileOutputStream class.**
8. **Server will create 200 OK File Created response using PrintStream class if the file was saved successfully.**
9. **We used termination signal Ctrl+C to shut down the server and closing all the sockets.**

**Execution Instructions:**

**Server will check for the files in ‘Files’ directory (attached) when server receives a GET request and Server will put the files in ‘Files’ directory when server receives a PUT request.**

1. **Compile both server and client using command –**

**javac MyServer.java**

**javac MyClient.java**

1. **Start the Server program using command –**

**java MyServer 8080**

1. **Sample GET and PUT requests**

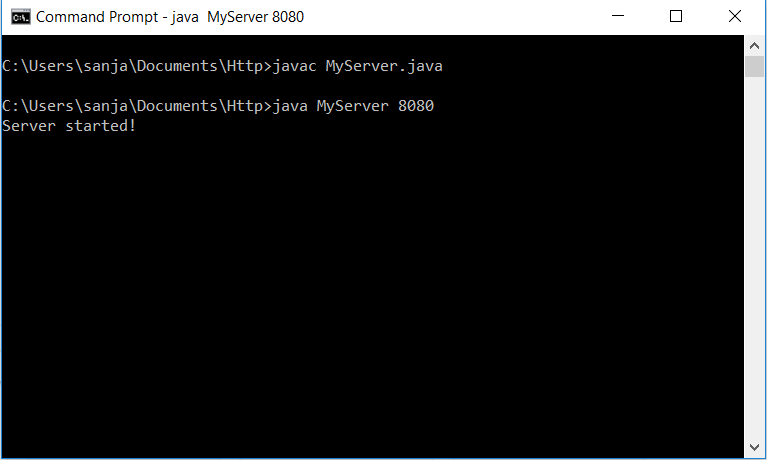
**>java MyClient localhost 8080 GET index.txt**

**>** **java -cp . MyClient localhost 8080 PUT demo.txt**

1. **Server is shut down using Ctrl+C command.**
2. **Server saves all files in “Files” directory.**

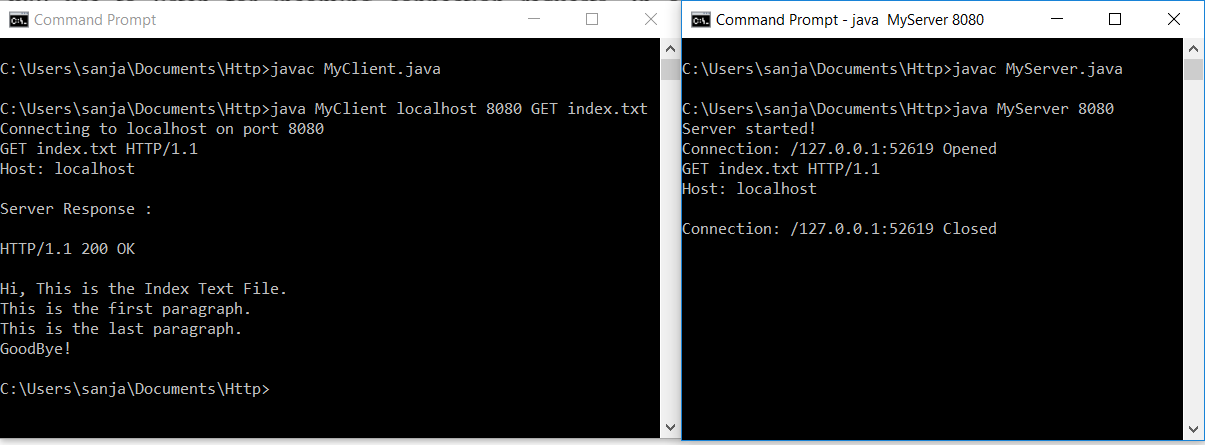
**Execution Snapshots:**

1. Starting the Server program.



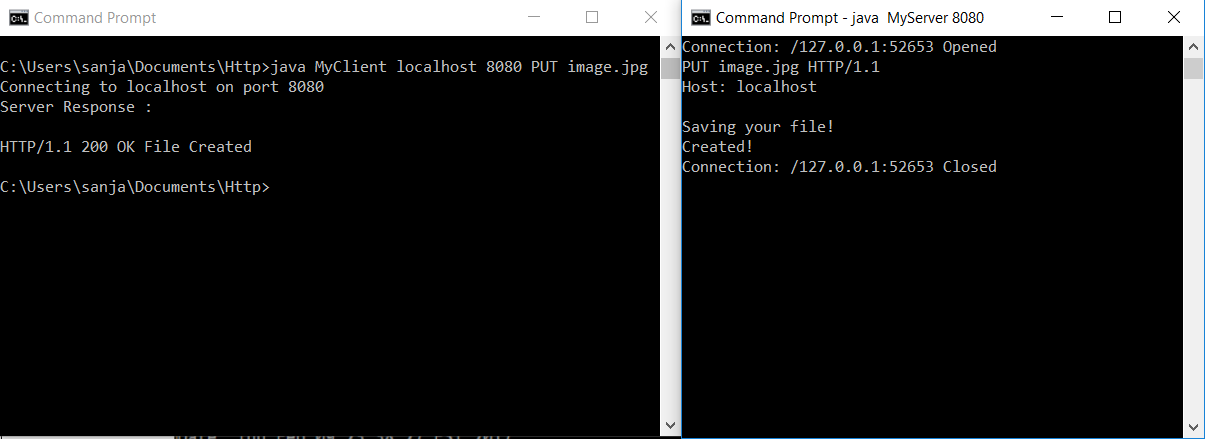
2. Creating a GET request from Client program to retrieve a text file index.txt.

java MyClient localhost 8080 GET index.txt

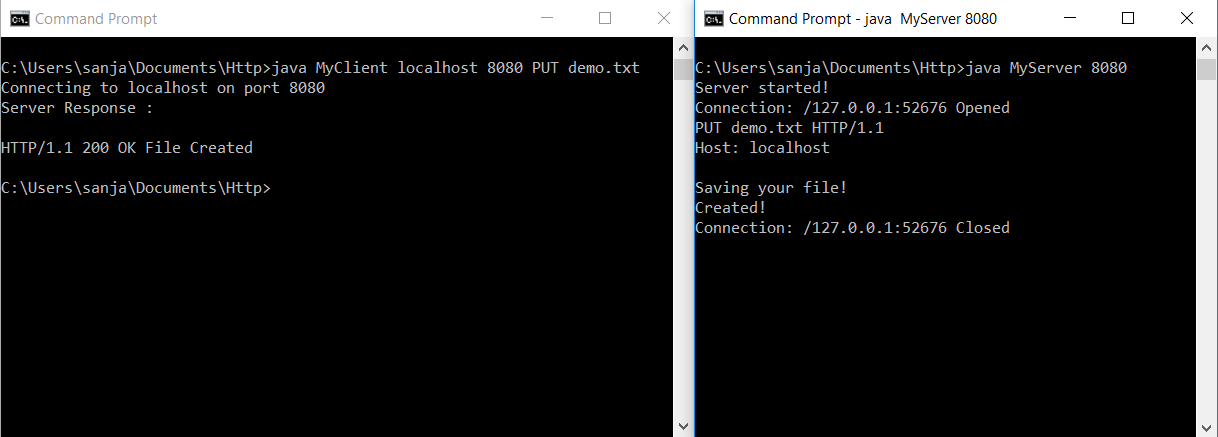


3. Creating a PUT request from client program to save image.jpg file on Server.

java MyClient localhost 8080 PUT image.jpg



4. Creating another PUT request for saving text file demo.txt on Server.



5. Server is shut down using **Ctrl+C** command.

