**Data Communications -1**

**Project-1**

**Name: Yakaiah.B**

**UML ID: UMS 01244568**

**Email:yakaiah\_bommishetti@student.uml.edu**

The project contains implementation of HttpClient and HttpServer.which supports

1. GET action
2. PUT action

The working of HttpClient is described below

***HttpClient* Working:**

It is a very simple HTTP client, which takes the GET/PUT requests and takes in server name or IP address and send the request to the server and prints the response.

1. GET :

The format of the command line is

myclient host port\_number GET filename

1. PUT:

The format of the command line is

myclient host port\_number PUT filename

Client algorithm:

1. Get the command-line parameters: host, port, GET/PUT and file name.
2. Connect to the server via a connection-oriented socket.
3. Submit a valid GET/PUT request.
4. Read the server's response and display it.

**HttpServer Working:**

It is a very simple HTTP (web) server. It’s a multi threaded server. I implemented it with two classes, and it supports only the GET and PUT commands for HTML, text, GIF, and JPEG files.

Server command line is:

* Myserver PortNumber

Server Algorithm:

1. Get the command-line parameters: port. If the port is not specified, set it to 80.
2. Open a server socket on the specified port.
3. Begin loop.
   1. Wait for client connection, getting a reference to the client socket when the connection is received.
   2. Create and start a new request thread, passing it the client socket.
4. End loop.
5. Close the server socket and exit.

**ServerThread Working:**

This class services client requests.

ServerThread Algorithm:

1. Get references to the socket input and output streams.
2. Print the request information.
3. If the command was "GET", send the response. If the file is found, it it is included as the content of the response. If it is not found, send "404 Not Found".
4. If the command was “PUT”, it will save the file. If the received file from client is created for the first time, the server sends back a "Created 201" response to the client. If an existing file is modified, the server sends back a "OK 200" response to the client.