

# Web Server using Sockets

## Computer Networks Project Report



Batch: B8

### **Submitted By-**

Rohit Kumar (12103628)

Aman Saxena (12103619)

Pulkit Bhardwaj (12103621)

Abhishek Rai (12103617)

# **ABSTRACT**

We designed a simple web server, which has basic functions to parse HTTP request from a browser and make a response back to the browser with the information requested.

To communicate between a web browser (client) and our server, we established a connection with socket programming, in addition, we used the select function to concurrently serve multiple connections from different clients. After a connection was set up, the server could receive request message from a client and dumped the message to the console, at the same time, request message was parsed such that the server could extract the file's URL requested by the client. We also extracted IP address and port number of the client for further use.. After that, the server could check whether the requested file existed in the server system or not and built up different HTTP response message. A HTTP response message is consist of a header and the data body; we constructed the header first and appended the data to it, then sent the whole response to the client. For the header part, we referenced the example request message from the textbook, wrote functions for each header line and constructed them together into an integrated header. For the data part, we read the data form requested file and appended it to the header, the server supported various types of data like html, jpg, mp4, pdf, etc. If the file could not be found in the server, the response message would be 404 not found.

# Results and Analysis

## **A request for test.html from browser**

To test the server performance I have used three different browsers (FireFox, Safari, and Chrome) to send request to the server. It turned out that there were some difference between browsers when dealing with video files.

In the screenshot below, we can observe the response message from the browser ( request for test.html). In the test.html, some text (“ni hao”), a picture (Lucy.jpg), a gif (test.gif), and a video (test.mp4) were included. The browser firstly sent a HTTP GET for test.html to the server.

```
^C
cs118@ubuntu:~$ gcc server9.c -o server9
cs118@ubuntu:~$ ./server9 6666
Here is the message:
GET /test.html HTTP/1.1
Host: 127.0.0.1:6666
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
```

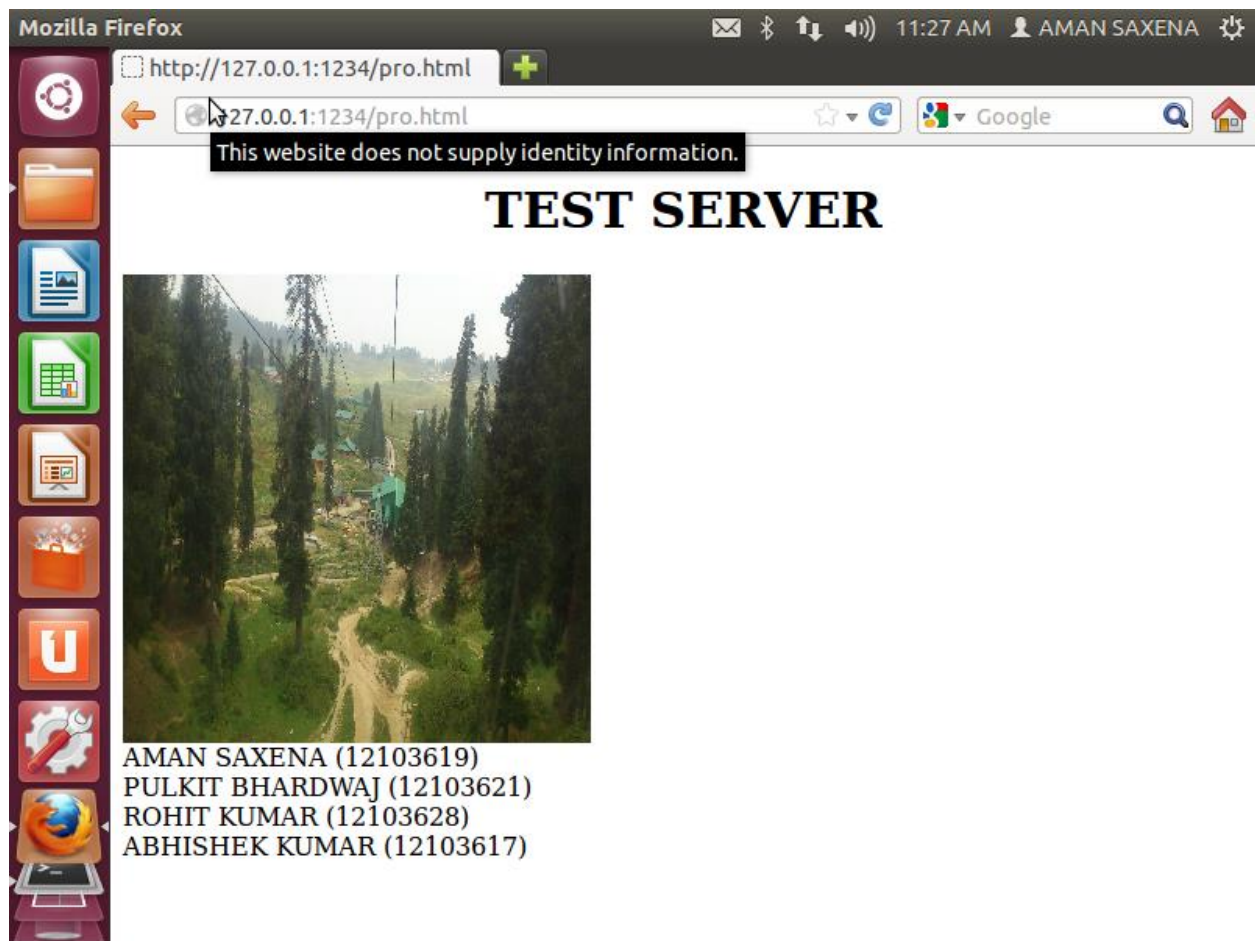
Right after the first request above, the browser sent three requests for the picture, gif and video respectively. The three requests are shown as following screenshots.

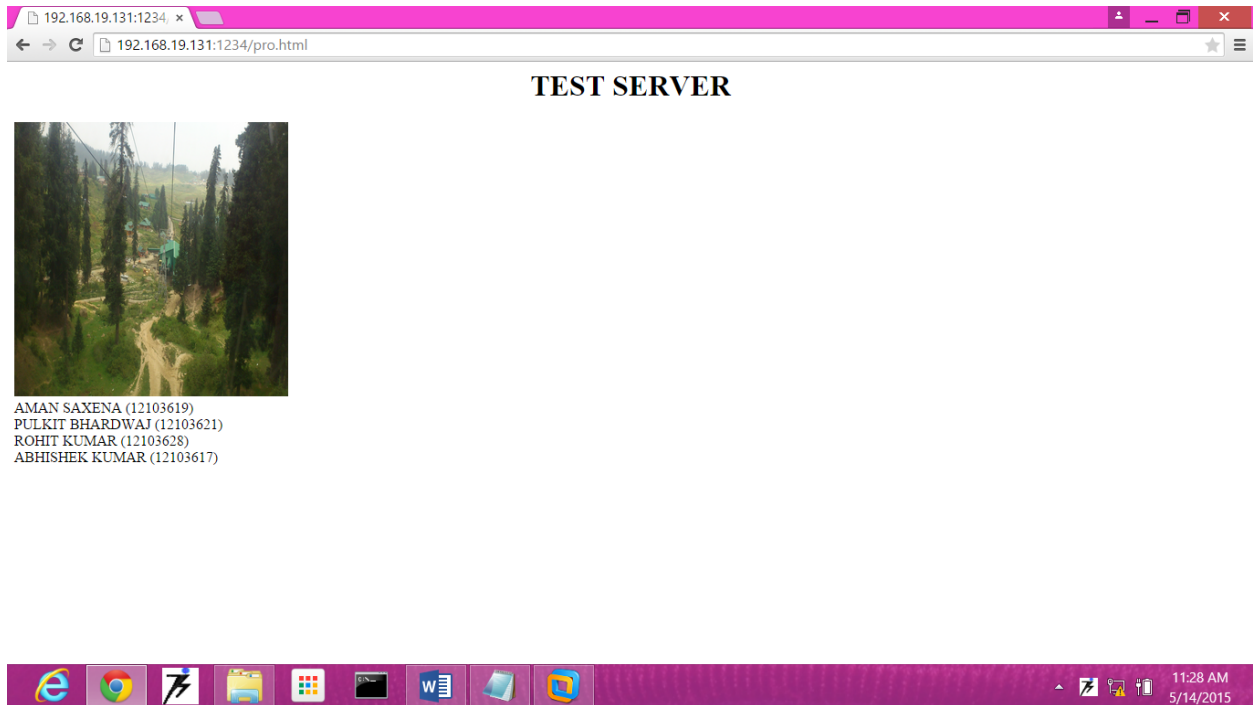
```
Here is the message:
GET /Lucy.jpg HTTP/1.1
Host: 127.0.0.1:6666
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://127.0.0.1:6666/test.html
```

```
Here is the message:
GET /test.gif HTTP/1.1
Host: 127.0.0.1:6666
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:7.0.1) Gecko/20100101 Firefox/7.0.1
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Referer: http://127.0.0.1:6666/test.html
```

As for the response to the browser, all three browsers could successfully show the text, picture and gif.

The following screenshot shows the result on Chrome and Fire Fox





## A request for invalid filename

Here is the result of wrong filename in the request. Server will check the filename, if there has no such file at local, an HTTP 404 response will be send back.