Name: Elsy Fernandes
SID:1004602254

Webserver.java:
This file consists of the web proxy server Code. It has the following class.
Webserver
ProcessClient

Webserver Class:
This class constructor creates a ServerSocket which is listening on port 5555.
Whenever there is a Client Request to this port, the server accepts the request and creates an instance of the thread **ProcessClient**. Client request is basically from a web browser(Chrome and Firefox were used for testing) .

ProcessClient Class:
This is a thread class. A new instance of this thread is created by the web server class whenever a connection is requested by the client (browser). This thread processes the get request fired by the client and responds back to the client with the available response.

If the response from the provided url is 404 File not found, then appropriate response is sent back to the client from this thread. If file is found in the server the content of the file is displayed on the server with response code of 200. The output in the console shows the HTTP request as well as HTTP Response along with the socket id.

If the file is requested from the server for the first time response will be fetched from the server. If the requested url was already requested before then the response is fetched from the cache and returned to the requesting client.

Development Environment:
Eclipse IDE was used for this project

Instructions:
1) Create a java project in Eclipse.
2) Create a package with name WebProxyServer inside the src folder of the project.
3) Import the file Webserver.java in to the package created above (Webserver.java already has the package name at the beginning)
4) Right click on **Webserver.java**, click on **Run as** and select **Java Application**. **"Server Created"** message is displayed on Server console.
5) After you receive the message type this URL in chrome http://127.0.0.1:5555/ gaia.cs.umass.edu/wireshark-labs/alice.txt (Clear the cache before the URL is hit on the address field of the browser)

6) Alternate is to configure the proxy as below:
   Proxy Address: 127.0.0.1
   Port:5555
   Then enter the address as http://gaia.cs.umass.edu/wireshark-labs/alice.txt in the browser. This request will be routed through the proxy address into the web proxy server.

7) Browser will show the HTTP Response by showing the content of Alice.txt
8) Console will display the http request, and http response with the response code and thread Id . And content of the requested object is displayed on the server console.(along with the request from Favicon- See **Note1**)
9) When you hit the URL http://127.0.0.1:5555/gaia.cs.umass.edu/wireshark-labs/alice.txt to the address field of the browser(same browser tab) for the second time (without clearing the cache in the same browser tab)the response you get will be from the cache. (Note:If you use different browser tab then you will have to clear the cache See **Note2**).
10) Since the content of Alice.txt is long you will not be able to see the messages printed in the server console. (Because message is printed on the top).
11) To test if the response is from the cache or from the server type the URL http://127.0.0.1:5555/gaia.cs.umass.edu/wireshark-labs/alice.aaa into the browser (by clearing the cache).Browser will display the message saying file not found.
12) Server Console will display the response code 404 and 'File not found' message along with the response has come from server for the first time.
13) When you hit the URL http://127.0.0.1:5555/gaia.cs.umass.edu/wireshark-labs/alice.aaa for the second into the browser (without clearing the cache) Browser will display the message saying file not found.
14) Server Console will display the response code 404 and 'File not found' message along with the response has come from cache for the second time.

Note1: When each page is loaded, the browsers(Chrome and Firefox) request for http://favicon.ico. Since separate thread is created for each request received by the web proxy server, whenever a new thread is created for each get request, if the url of the get request has favicon, I am closing the connection as our task is only to process the requested file.

Note2: Before opening the given url, make sure that the caching is disabled on the browser. To clear the cache go to setting of the browser and clear the browsing data. If not, the request is responded from the browser cache and will not reach the Web proxy server.

References: The server code provided with the project was used to create the Webproxyserver

----------xxxxxxxxxxx----------