**MULTI-THREADED WEB PROXY SERVER**

CSE-5344: Computer Networks

First name : Sai Kavya

Last name : Dukkipati

ID # 1000980778

**INTRODUCTION**

Web proxy server:

A common proxy application is a caching Web proxy. This provides a nearby cache of Web pages and files available on remote Web servers, allowing local network clients to access them more quickly or reliably.

When it receives a request for a Web resource (specified by a URL), a caching proxy looks for the resulting URL in its local cache. If found it returns the document immediately. Otherwise it fetches it from the remote server, returns it to the requester and saves a copy in the cache.

Multi-Threading :

If the server is able to handle multiple simultaneous service requests in parallel then the server is said to be multi-threaded. In the main thread, the server listens to a fixed port. When it receives a TCP connection request, it sets up a TCP connection through another port and services the request in a separate thread.

HTTP methods - GET and POST :

The Hypertext Transfer Protocol (HTTP) is designed to enable communication between clients and servers. HTTP works as a request-response protocol between a client and a server.

Two commonly used methods for request-response between a client and a server are - GET and POST.

GET: Requests data from a specified source.

POST: Submits data to be processed to specified source.

**IMPLEMENTATION**

* The proxy server relays HTTP requests that it receives through a well-known port number 8080.
* A browser (client) submits an HTTP request to the proxy server .
* Each time the Proxy server receives a HTTP connection, it calls the multi-threaded class ProxyServer and passes the HTTP connection to it for handling the actual request.
* The HTTP request thread processes the request by connecting to the destination Web server, retrieving the content corresponding to the request and forwarding to the Web browser.
* After the class ProxyServer completes HTTP request, it closes the connection of both the destination Web server and the WEB browser.

**DEVELOPMENT ENVIRONMENT**

Eclipse

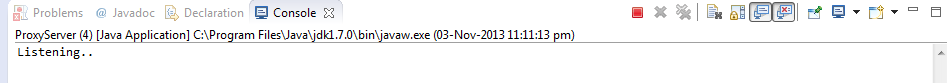
**INSTRUCTIONS ON HOW TO RUN THE CODE**

**Step 1:** Copy the ProxyServer project folder to the Eclipse workspace

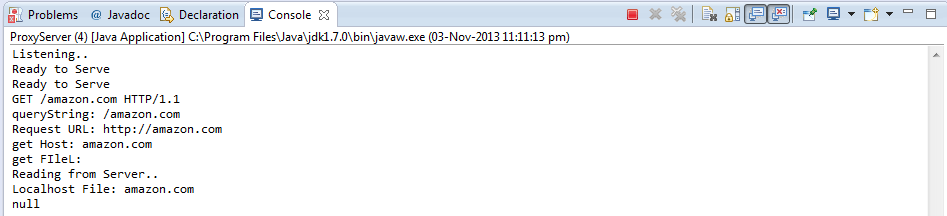
**Step 2:** Import this folder in Eclipse using the Import option. Then open the workspace.

**Step 3:** There is a file with .java extension – ProxyServer.java.

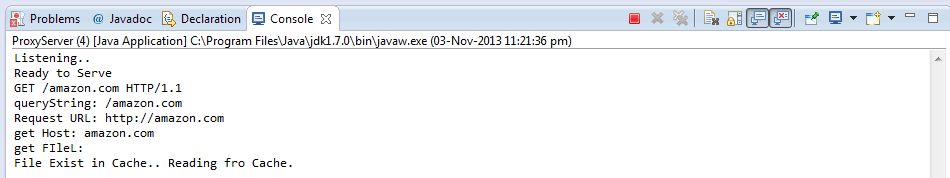
Run the ProxyServer.java,It displays on the console that proxy server is listening to the requests from the browser.

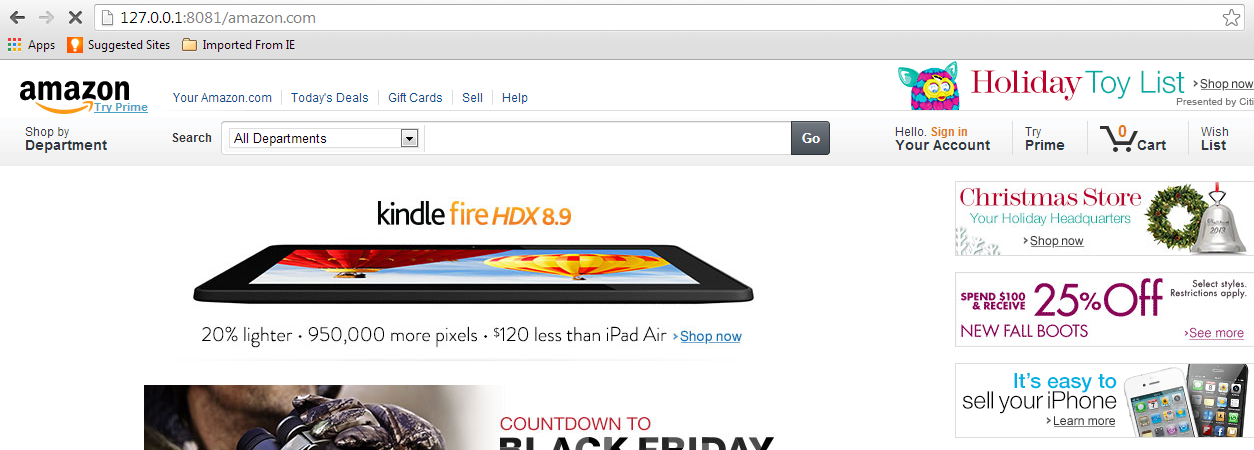


Open the browser and give the URL 127.0.0.1:8080/www.amazon.com. The get method gives access to the website by checking the cache. If the URL is in the cache it will retrieve it from the cache or else the request is sent to the original web server and the web page is read from the server.



If the URL is already in the cache it will read the web page from cache and serves the request of the browser.





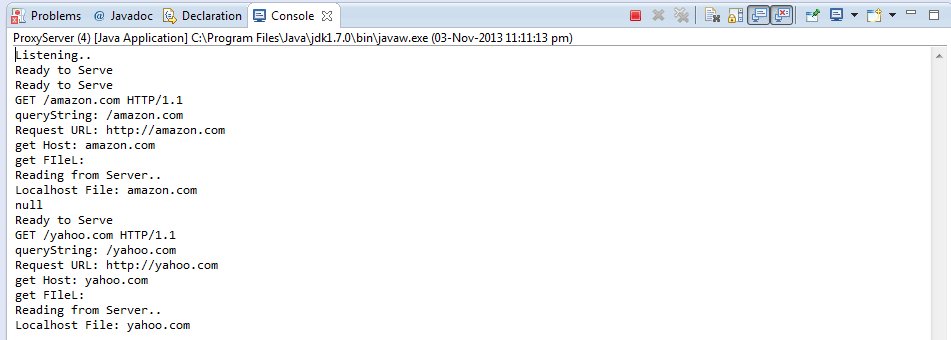
**Step 4:** The post method retrieves the website by giving the URL of the uploading website with the local port number.

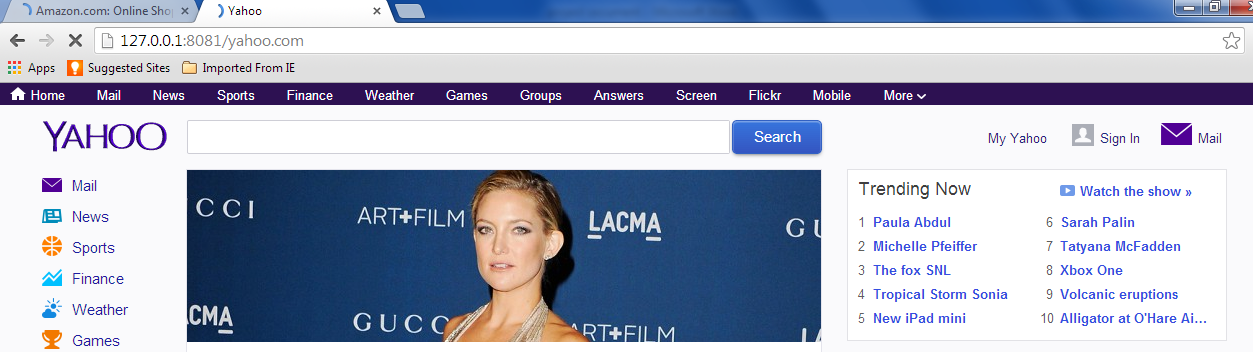
In the web browser, after retrieving the page if we click the upload file button on the webpage it will give a message.

After sometime that it has been uploaded. This shows that the post method is working properly.

**Step 5:** Open another tab in the web browser and give address in the web browser such as 127.0.0.1:8080/www.yahoo.com

The page is read from the server. This shows multithreading is also working





**Step 6:** Stop the server after the request has been served.

**CONCLUSION**

A web proxy server is implemented which receives HTTP requests from a browser and forwards them to the origin server and then sends corresponding HTTP responses received from the origin server to the client. It can also handle errors, caches webpages and multiple requests at the same time.