

Project Report

CSE – 5344 Computer Networks



Project: **Building a MultiThreaded Web Server**

Name: **Shreyas Jagadeep Shete**

eMail ID: `sxs8861@mavs.uta.edu`

UTA ID: **1001888859**

October 28, 2022

ABSTRACT

This project presents a Q-learning algorithm for learning to play a simple game of pong. The game is played by two players; opponent and agent, each with a paddle, and a ball is bounced between them. The goal of the game is to hit the ball with the paddle so that it goes past the opponent's paddle and hits the wall. The game is played for a set number of iterations, and the player with the most points at the end of the game wins. The Q-learning algorithm is used to learn the optimal strategy for playing the game. The algorithm is based on the idea of reinforcement learning, where the agent is rewarded for taking actions that lead to successful outcomes. The Q-values of the state-action pairs are updated using the Bellman equation. Temporal difference is used to keep track of the score and to move the paddle.

$$\text{Temporal difference} = \text{TD}(S_t, a_t) = r_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t)$$

$$\text{Bellman equation} = Q^{\text{new}}(S_t, a_t) = Q^{\text{old}}(S_t, a_t) + \alpha \text{TD}(S_t, a_t)$$

The agent chooses the action that maximizes the Q-value. The state of the game is represented by the position of the ball and the paddle. The action space is the set of possible paddle movements. The reward is 1 if the ball is hit, and -1 if the ball is missed. The goal is to learn a policy that maximizes the expected reward. The algorithm is tested on a simple game environment, and the results show that it is able to learn the optimal strategy for playing the game.

OBJECTIVE

Build a multi-threaded Web server that is capable of processing multiple simultaneous service requests in parallel. Here the Web Server delivers the homepage to Web Browser (nothing but the client).

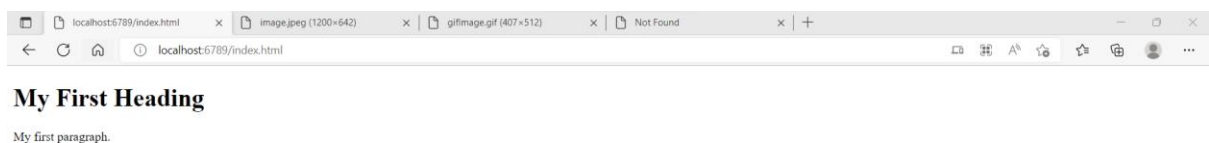
RESULTS

We first run the web server using WebServer.java calling the HttpRequest.java for each request (new threads are created). Here the client is nothing but a web browser. Open any web browser and try to access the server by using the URL: <http://localhost:6789> (Note we need to mention port number : 6789 as we have hosted it on that port number else it will take default port number 80).

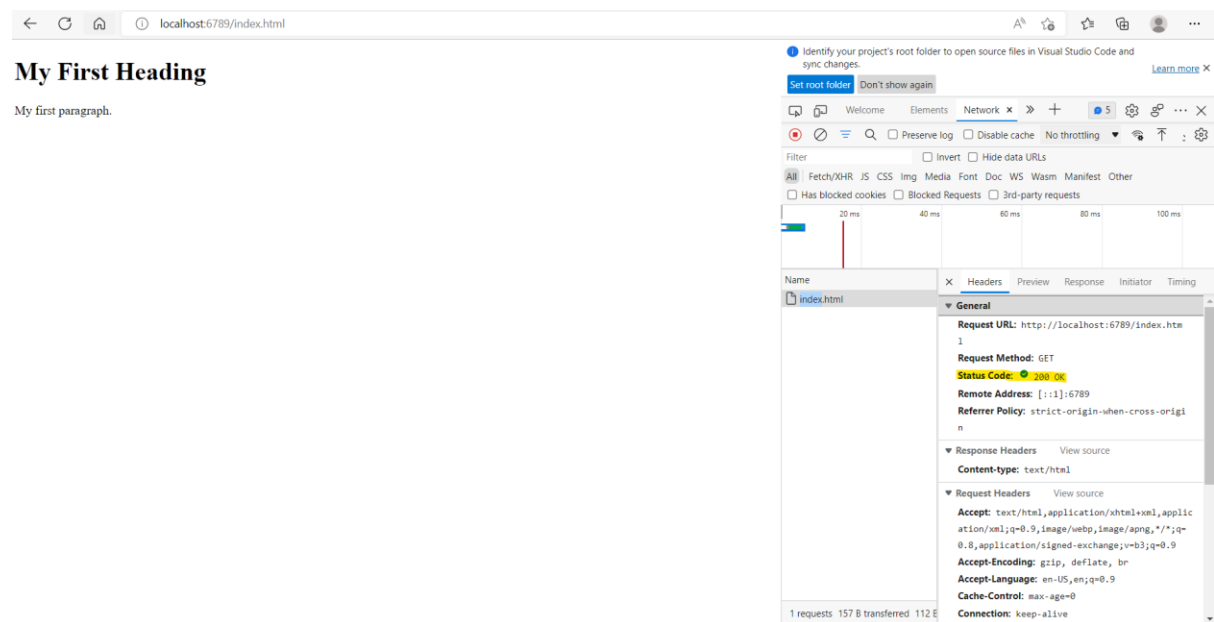
Output at Client (Web Browser):

NOTE: “index.html”, “gifImage.gif”, “image.jpeg” files should be present in the local folder where the WebServer.java and HttpRequest.java file is present at the time of the run.

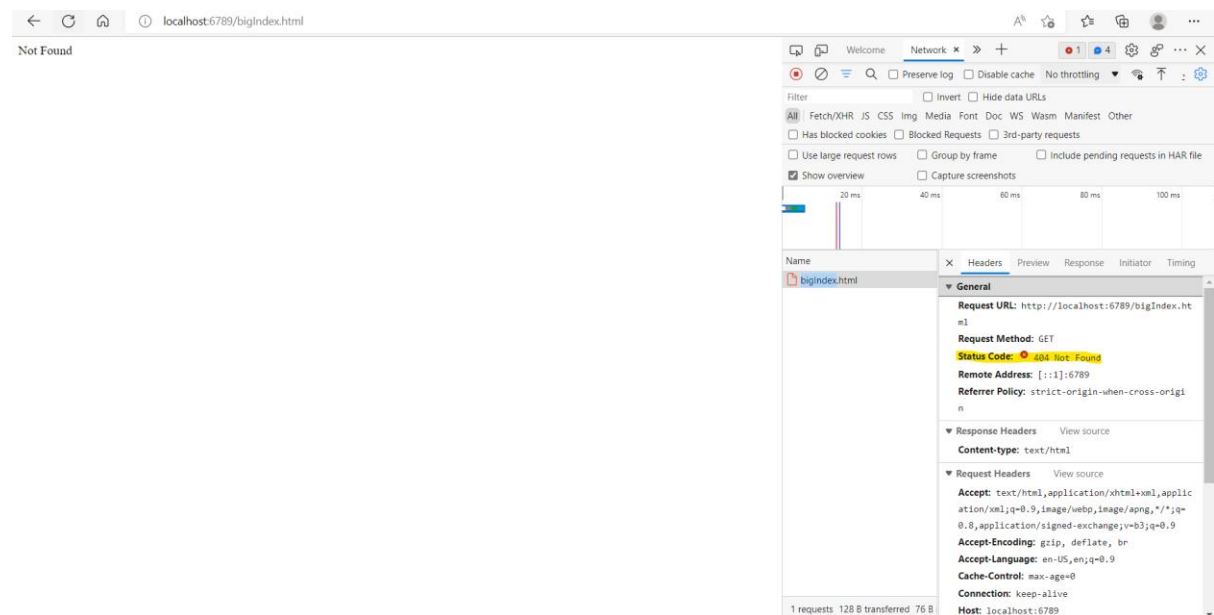
Case1: Accessing index.html page which is placed on web browser, using URL: <http://localhost:6789/index.html> .We get the following output,



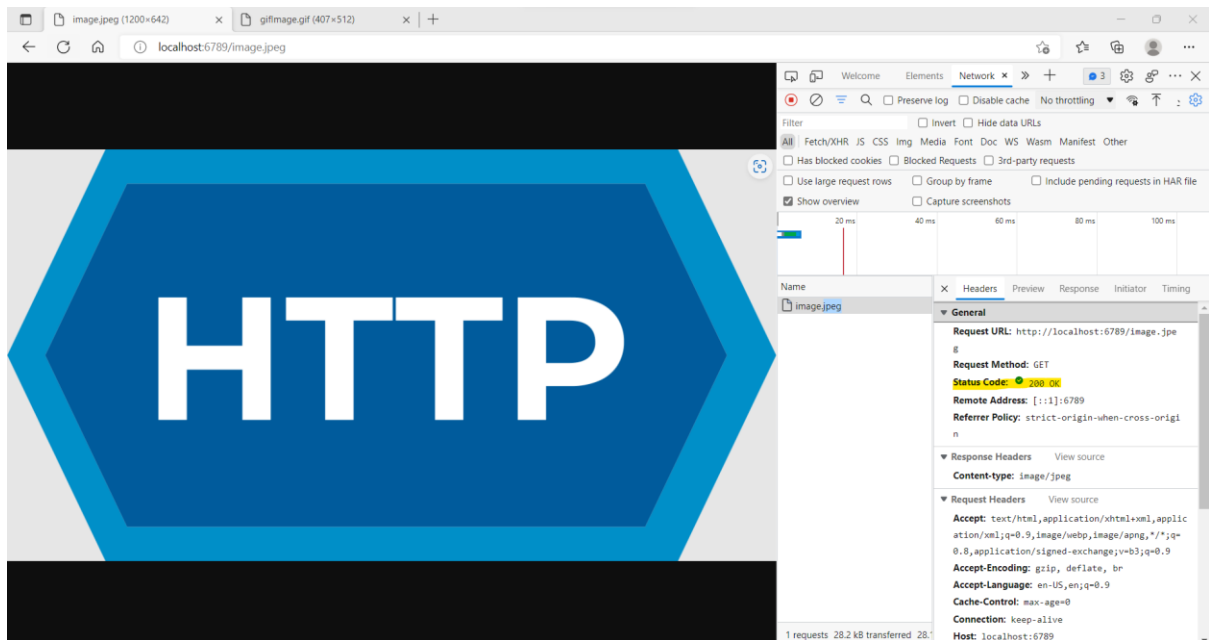
Here the StatusCode is 200 (HTTP 1.0 GET)



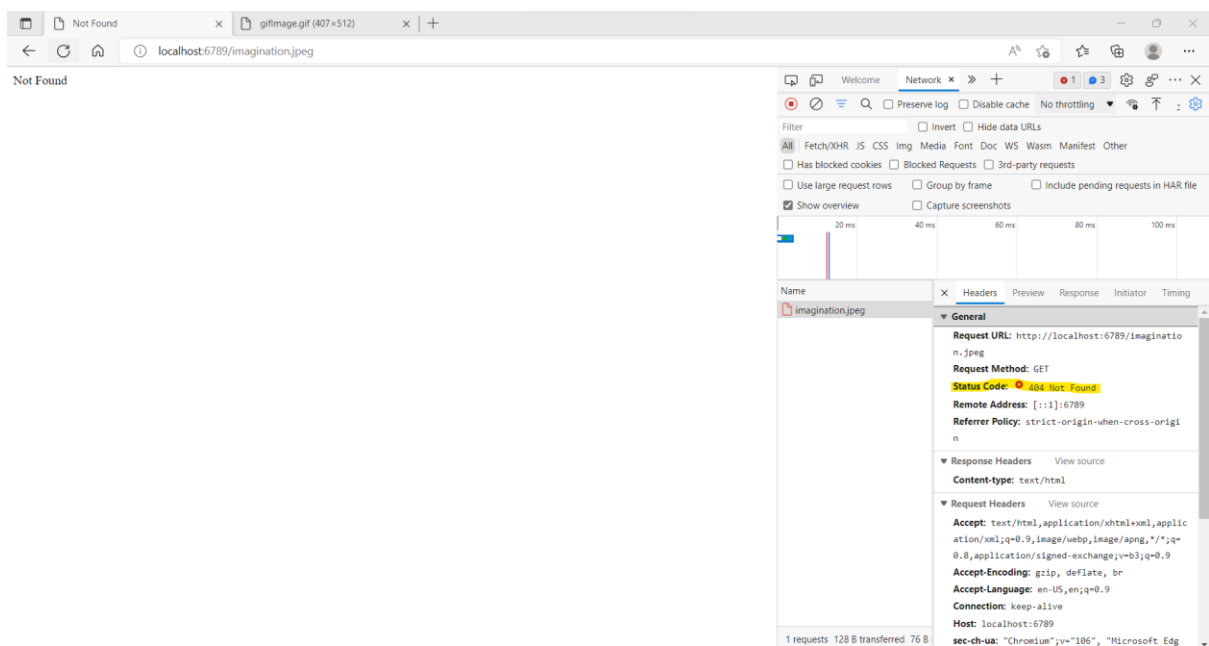
Case2: If the .html file is not present in the server we get “Not Found” with Status code : “404 Not Found”



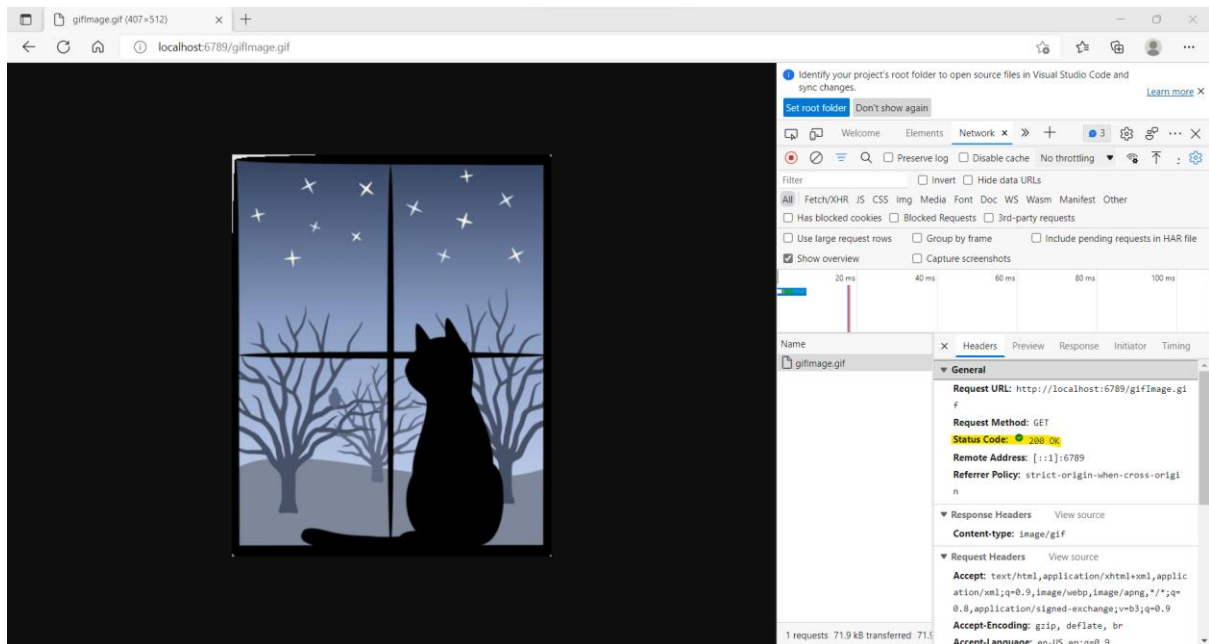
Case3: Accessing image.jpeg page which is placed on web browser, using URL: `http://localhost:6789/image.jpeg` (NOTE: Here the Status Code: “200 OK”). We get the following output,



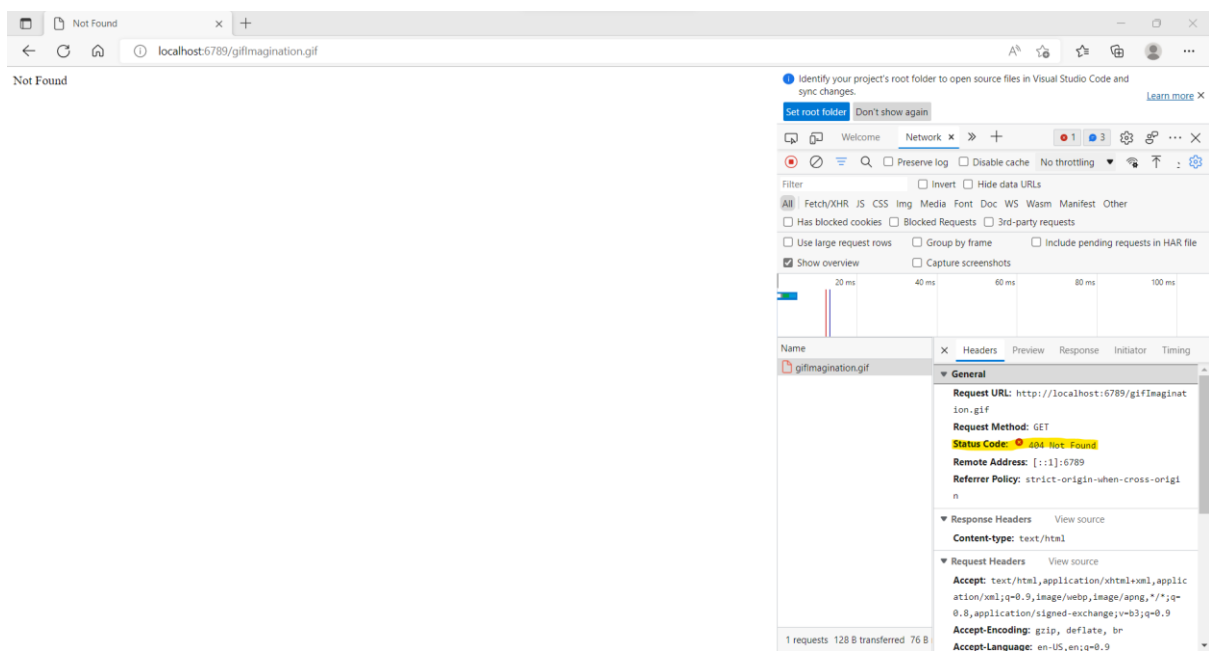
Case4: If the .jpeg file is not present in the server we get “Not Found” with Status code : “404 Not Found”



Case5: Accessing gifImage.gif page which is placed on web browser, using URL: <http://localhost:6789/gifImage.gif> (NOTE: Here the Status Code: “200 OK”). We get the following output,



Case6: If the .gif file is not present in the server we get “Not Found” with Status code : “404 Not Found”



Output from WebServer console for several runs is as shown below:

Client Connection accepted

GET /index.html HTTP/1.1

Host: localhost:6789
Connection: keep-alive

sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Socket is closed now
Client Connection accepted

GET /favicon.ico HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
sec-ch-ua-platform: "Windows"
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: image
Referer: http://localhost:6789/index.html
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Socket is closed now
Client Connection accepted

GET /bigIndex.html HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Client Connection accepted
Socket is closed now

GET /image.jpeg HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
, */*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Client Connection accepted
Socket is closed now

GET /imagination.jpeg HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
, */*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Client Connection accepted
Socket is closed now

GET /gifImage.gif HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Client Connection accepted

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Socket is closed now

GET /gifImagination.gif HTTP/1.1

Host: localhost:6789
Connection: keep-alive
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106",
"Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36 Edg/106.0.1370.52
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng
,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Client Connection accepted
Socket is closed now

References

1. <https://www.youtube.com/watch?v=FqufxoA4m70>
2. <https://www.geeksforgeeks.org/socket-programming-in-java/>
3. <https://www.geeksforgeeks.org/multithreaded-servers-in-java/>
4. <https://www.youtube.com/watch?v=BWjGQllkgT4&t=200s>
5. <https://www.youtube.com/watch?v=h2zi2IVNhtk>
6. <https://github.com/samruddhikapileshwar/Building-a-Simple-Web-Client-and-a-Multithreaded-Web-Server/blob/master/WebServer.java>
7. <https://www.youtube.com/watch?v=8J7jS4G6AkE>
8. <https://www.youtube.com/watch?v=ZlzoerHHQo&t=3s>
9. <https://docs.oracle.com/javase/7/docs/api/java/net/ServerSocket.html>
10. <https://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>
11. <https://www.youtube.com/watch?v=FqufxoA4m70>