3.- Implementing a simple HTTP Server

Task 3.1.- The basic HTTP server

Q1: What method is specified in the request line?

The GET method

Q2: What is the HTTP format requested by the client?

HTTP /1.1

Q3: What information is specified in the header lines?

Host ,connection, Cashe-control, Upgrade-insecure-Requests, User-agent, Accept, Sec-fetch-site, Sec-fetch-mode, sec-fetch-dest, Accept-Encoding, Accept-language.

Q4: What is displayed in the web browser?

Hello world

Task 3.2.- Improving the HTTP server

Q1: Add additional lines in order to iterate through “data\_received” and print its content in the python console (hint: use a FOR loop).

Q2: From “data\_received”, extract the requested url that is contained in the request line. Hint: You should find first where the request line is in “data\_received”, most likely is the first element of the array and then, you should use split again using a different separator. Store the requested url in a variable named “requested\_resource”

Q1: How many requests do the client sends?

One time

Q2: Does the browser display the requested html file?

Yes it dose

Q1:Is the browser displaying the requested webpage?

Yes when using <http://127.0.0.1/C:/Users/mohammed/Desktop/Gasbot_sepia.jpg>

Yes when using <http://127.0.0.1/C:/Users/mohammed/Desktop/text_image.html>

Q2: How many requests does the client make in order to properly display the requested webpage?

Two requests

Task 3.3.- HTTP Sever ++

Q: Notice that so far you have accessed the HTTP server locally. Try to access your HTTP server using a different computer to request ”text\_image.html”. Was the webpage successfully displayed? If not, what changes are needed in order to access the server remotely?

Nothing because its local. Make a live server instead of local.