**TASK 1**

1. **Difference between HTTP 1.1 vs HTTP 2**

|  |  |
| --- | --- |
| **HTTP 1.1** | **HTTP 2.0** |
| HTTP/1.1 loads resources one after the other. | HTTP/2 is able to use a single TCP connection to send multiple streams of data at once |
| A server only serves content to a client device if the client asks for it | A server is allowed to push content to a client before the client asks for it |
| Less faster and less efficient than HTTP 2.0. | Much faster and efficient than HTTP 1.1.  It prioritizes content during the loading process fastly. |
| keeps all requests and responses in plain text format | uses the binary framing layer to encapsulate all messages in binary format |

1. **HTTP version history :**

* **HTTP/0.9** **-** The One-line Protocol (**1991**) **--** Initial version of HTTP **--** a simple client-server, request-response, telenet-friendly protocol
* **HTTP/1.0** **-** Building extensibility (**1996**) **--** Browser-friendly protocol **--** Provided header fields including rich metadata about both request and response (HTTP version number, status code, content type)

# HTTP/1.1 - The standardized protocol (2015) -- This is the HTTP version currently in common use -- Introduced critical performance optimizations and feature enhancements -- persistent and pipelined connections, chunked transfers, compression/decompression, content negotiations, virtual hosting, faster response and great bandwidth savings by adding cache support.

* **HTTP/2 (1995) -** high-level compatibility **--** Decrease latency to improve page load speed **--** Support common existing use cases of HTTP.
* **HTTP/3 -- (2020)**

1. **List 5 differences between Browser JS vs Node Js.**

|  |  |
| --- | --- |
| **Browser JS** | **Node Js** |
| It is used for writing scripts on the website | Node Js is a Javascript runtime environment |
| Some of the javascript frameworks are RamdaJS, TypedJS, etc. | Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm. |
| It is basically used on the client side | It is mostly used on the server side |
| It is capable enough to add HTML and play with the DOM | Node Js does not have the capability to add HTML tags |

1. **what happens when you type a URL in the address bar in the browser?**
2. You type URL into the address bar of your browser.
3. The browser checks the cache for a DNS record to find the corresponding IP address of the URL.
4. If the requested URL is not in the cache, ISP’s DNS server initiates a DNS query to find the IP address of the server that hosts the particular URL.
5. The browser initiates a TCP connection with the server.
6. The browser sends an HTTP request to the webserver.
7. The server handles the request and sends back a response.
8. The server sends out an HTTP response.
9. The browser begins rendering the HTML (response). The browser displays the HTML content after rendering.