**Task 1**

* **Difference between HTTP1.1 vs HTTP2:**

These are the high-level differences between HTTP1 and HTTP2:

* HTTP2 is binary, instead of textual
* HTTP2 is fully multiplexed, instead of ordered and blocking
* HTTP2 can, therefore, use one connection for parallelism
* HTP2 uses header compression to reduce overhead
* HTTP2 allows servers to “push” responses proactively into client caches
* **HTTP version history:**

HTTP (HyperText Transfer Protocol) is the underlying protocol of the World Wide Web. Developed by Tim Berners-Lee and his team between 1989-1991.

**HTTP0.9:** In 1989, while he was working at CERN, Tim Berners-Lee wrote a proposal to build a hypertext system over the Internet.Initially calling it the Mesh, it was later renamed to World Wide Web during its implementation in 1990. Built over the existing TCP and IP protocols, it consisted of 4 building blocks: 1. HTML 2. HTTP 3. Client 4. Server. This initial version was then named as HTTP0.9 to distinguish between other versions.

This version only had one GET request followed by path which in response returns only a single file.

**HTTP1:** HTTP/0.9 was very limited and both browsers and servers quickly extended it to be more versatile:

* Versioning information is now sent within each request (HTTP/1.0 is appended to the GET line)
* A status code line is also sent at the beginning of the response, allowing the browser itself to understand the success or failure of the request and to adapt its behavior in consequence
* The notion of HTTP headers has been introduced, both for the requests and the responses, allowing metadata to be transmitted and making the protocol extremely flexible and extensible.
* With the help of the new HTTP headers, the ability to transmit other documents than plain HTML files has been added.

In November 1996, an informational document describing the common practices has been published. This is the definition of HTTP/1.0 and it is notable that, in the narrow sense of the term, it isn't an official standard.

**HTTP1.1:** The standardized version of HTTP i.e. HTTP1.1 was published in early 1997 just after months of publishing HTTP1.0.

HTTP/1.1 clarified ambiguities and introduced numerous improvements:

* A connection can be reused, saving the time to reopen it numerous times to display the resources embedded into the single original document retrieved.
* Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted, lowering the latency of the communication.
* Chunked responses are now also supported.
* Additional cache control mechanisms have been introduced.
* Content negotiation, including language, encoding, or type, has been introduced, and allows a client and a server to agree on the most adequate content to exchange.
* Thanks to the [Host](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Host) header, the ability to host different domains at the same IP address now allows server colocation.

**HTTP2:** Officially standardized, in May 2015, HTTP/2 has had much success. By July 2016, 8.7% of all Web sites[1] were already using it, representing more than 68% of all requests[2]. High-traffic Web sites showed the most rapid adoption, saving considerably on data transfer overheads and subsequent budgets.

The HTTP/2 protocol has several prime differences from the HTTP/1.1 version:

* It is a binary protocol rather than text. It can no longer be read and created manually. Despite this hurdle, improved optimization techniques can now be implemented.
* It is a multiplexed protocol. Parallel requests can be handled over the same connection, removing the order and blocking constraints of the HTTP/1.x protocol.
* It compresses headers. As these are often similar among a set of requests, this removes duplication and overhead of data transmitted.
* It allows a server to populate data in a client cache, in advance of it being required, through a mechanism called the server push.
* **List 5 differences between Browser JS vs Node Js.**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Browser JS** | **NodeJS** |
| 1 | JavaScript is a programming language that is used for writing scripts on the website. | NodeJs is JavaScript runtime environment. |
| 2 | It can only run in browser. | It can run outside the browser |
| 3 | It is mostly a client side language and used for the front end development. | It is used in the server side development i.e. backend development. |
| 4 | It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | It is written in C, C++ and JavaScript |
| 5 | JavaScript can add HTML elements and has ability to play with the DOM. | NodeJS does not have capability to add HTML elements. |

* **What happens when you type a URL in the address bar in the browser?**

1. You enter a URL into a web browser
2. The browser looks up the IP address for the domain name via DNS
3. The browser sends a HTTP request to the server
4. The server sends back a HTTP response
5. The browser begins rendering the HTML
6. The browser sends requests for additional objects embedded in HTML (images, css, JavaScript) and repeats steps 3-5.
7. Once the page is loaded, the browser sends further async requests as needed.