Task 1

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

* **Binary Protocols**:Binary Protocols consume less bandwidth,efficiently parsed and are less-error prone compared to the textual protocols used by HTTP1.1 .
* **Multiplexing**: HTTP2 can initiate multiple requests paralelly over a single TCP connection. These capabilities solve the head-of-line blocking problem in HTTP1.1,in which a packet at the front of the line stops other packets from being transmitted.
* **Header Compression**:HTTP2 uses Header compression to overcome the overhead caused by TCP’s slow-start mechanism.
* **Server push**:HTTP2 servers push likely-to-be-used resources to browser’s cache before requesting.This cuts down additional request cycles.
* **Increased Security**:Web browsers support HTTP2 only via encrypted connections,securing users and applications.

### **HTTP version History.**

* HTTP/0.9

It was the first and simple version. Single line requests and responses, requests being the only method,GET, followed by the path to the resource and responses consisted of files.Only HTML files could be transmitted.In case of a problem, a specific HTML was sent back for human consumption.

* HTTP/1.0

In HTTP/1.0, the notion of headers has been introduced.Additionally, versioning information 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 status of the request and to adapt its behaviour in consequence. Headers facilitate transmitting of metadata and making the protocol extremely flexible and extensible.The ability to transmit other documents than plain HTML has also been added.

* HTTP/1.1

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 header, the ability to host different domains at the same IP address now allows server colocation.
* HTTP/2
* 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 and Node Js.**

|  |  |
| --- | --- |
| **Browser Js** | **Node Js** |
| Javascript is a programming language. | Node Js is a Javascript runtime environment. |
| It is used on client-side. | It is mostly used on server-side. |
| Javascript is capable enough to add HTML and to deal with DOM. | Node Js is not capable of including HTML tags. |
| Javascript can run in any browser engine. | Nodejs can only run in the V8 engine of google chrome. |
| It is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | Node js is written in C, C++ and Javascript. |

### 

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

* Browser checks cache for DNS entry to find the corresponding IP address of the website.

It looks for the following cache,if not found then continues checking in the next one.

Browser Cache

Operating Systems cache

Router Cache

ISP cache

* If not found in cache,ISP’s DNS server initiates a DNS query to find IP address of the server that hosts the website.
* Browser initiates a TCP connection with the server using synchronization and acknowledge messages.
* Browser sends a HTTP request to the server.
* Server on the host handles the request and sends back a response.It assembles the response in some format like JSON,XML or HTML .
* Server send out response along with status of response
* Browser displays the content.