**Task 1**

1.] Difference between HTTP1.1 vs HTTP2

HTTP 1.1:

* HTTP/1.1 practically allows only one request per TCP connection.
* The other problem with HTTP/1.1 is the duplication of data across requests (cookies and other headers). Too many requests means too much redundant data, which would impact performance.
* Much slower to load a page, compared to HTTP 2.

HTTP 2:

1. HTTP/2 can send multiple requests for data in parallel over a single TCP connection. Most modern browsers limit TCP connections to one server.
2. Website load faster without any optimization
3. HTTP/2 compress a large number of redundant header frames. HPACK compresses the individual value of each header before it is transferred to the server, which then looks up the encoded information in a list of previously transferred header values to reconstruct the full header information.
4. The latest HTTP version has evolved significantly in terms of capabilities and attributes such as transforming from a text protocol to a binary protocol.
5. This capability allows the server to send additional cacheable information to the client that isn’t requested but is anticipated in future requests.

2.] http version history :

1. HTTP/0.9 – The one-line protocol:

The initial version of HTTP had no version number; it has been later called 0.9 to differentiate it from the later versions. HTTP/0.9 is extremely simple, requests consist of a single line and start with the only possible method 'GET' followed by the path to the resource. Unlike subsequent evolutions, there were no HTTP headers, meaning that only HTML files could be transmitted, but no other type of documents.

1. HTTP/1.0 – Building extensibility :
2. Versioning information is now sent within each request.
3. 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 behaviour in consequence.
4. 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.
5. With the help of the new HTTP headers, the ability to transmit other documents than plain HTML files has been added.
6. HTTP/1.1 – The standardized protocol:

The first standardized version of HTTP, HTTP/1.1 was published in early 1997, only a few months after HTTP/1.0.

* 1. A connection can be reused, saving the time to reopen it numerous times to display the resources embedded into the single original document retrieved.
  2. 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.
  3. Chunked responses are now also supported.
  4. Additional cache control mechanisms have been introduced.
  5. 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.
  6. Thanks to the Host header, the ability to host different domains at the same IP address now allows server colocation.

1. HTTP/2 – A protocol for greater performance:

Officially standardized, in May 2015.

* 1. 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.
  2. 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.
  3. It compresses headers. As these are often similar among a set of requests, this removes duplication and overhead of data transmitted.
  4. It allows a server to populate data in a client cache, in advance of it being required, through a mechanism called the server push.

3.] List 5 difference between Browser JS (console) vs Nodejs :

|  |  |
| --- | --- |
| JS (Console) | Nodejs |
| 1. Javascript is a programming language that is used for writing scripts on the website. | 1. NodeJS is a Javascript runtime environment. |
| 2. Javascript can only be run in the browsers. | 2. NodeJS code can be run outside the browser. |
| 3. Used on the client-side | 3. Used on the server-side. |
| 4. Javascript is used in frontend development | 4. Nodejs is used in server-side development |
| 5. Javascript is capable enough to add HTML | 5. Nodejs does not have capability to add HTML tags. |

4.] What happens when you type a URL in the address bar in the browser?

1. The URL is first entered in the address bar. When you type any URL you basically want to reach the server where the website is hosted.
2. The browser looks for the IP address of the domain name in the DNS (Domain Name Server).

The DNS checks at the following places for the IP address.

1. Check Browser Cache: The browser maintains a cache of the DNS records for some fixed amount of time. It is the first place to run a DNS query.
2. Check OS Cache: If the browser doesn't contain the cache then it requests to the underlying Operating System as the OS also maintains a cache of the DNS records.
3. Router Cache: If your computer doesn't have the cache, then it searches the routers as routers also have the cache of the DNS records.
4. ISP (Internet Service Provider) Cache: If the IP address is not found at the above three places then it is searched at the cache that ISP maintains of the DNS records. If not found here also, then ISP’s DNS recursive search is done. In "DNS recursive search", a DNS server initiates a DNS query that communicates with several other DNS servers to find the IP address.
5. The Browser initiates a TCP connection with the server.

When the browser receives the IP address, it will build a connection between the browser and the server using the internet protocol. The most common protocol used is TCP protocol. The connection is established using a three-way handshake. It is a three-step process.

1. Step 1 (SYN): As the client wants to establish a connection so it sends an SYN (Synchronize Sequence Number) to the server which informs the server that the client wants to start a communication.
2. Step 2 (SYN + ACK): If the server is ready to accept connections and has open ports then it acknowledges the packet sent by the server with the SYN-ACK packet.
3. Step 3 (ACK): In the last step, the client acknowledges the response of the server by sending an ACK packet. Hence, a reliable connection is established and data transmission can start now.

#### The browser sends an HTTP request to the server.

#### The server handles the incoming request and sends an HTTP response.

The server handles the HTTP request and sends a response. The first line is called the status line. A Status-Line consists of the protocol version (e.g HTTP/1.1) followed by numeric status code (e.g 200) and its associated textual phrase (e.g OK). The status code is important as it contains the status of the response.

1. 1xx: Informational: It means the request was received and the process is continuing.
2. 2xx: Success: It means the action was successful.
3. 3xx: Redirection: It means further action must be taken in order to complete the request. It may redirect the client to some other URL.
4. 4xx: Client Error: It means some sort of error in the client’s part.
5. 5xx: Server Error: It means there is some error on the server-side.

#### The browser displays the HTML content.