Task 1

1.Difference between http1.1 vs http2?

|  |  |  |
| --- | --- | --- |
| Key features | **http1.1** | **http2** |
| Protocol Type | As opposed to HTTP/1.1, which keeps all requests and responses in plain text format, | HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers |
| Header Comparision | Headers are sent on every request leading to a lot of duplicate data being sent uncompressed across the wire. | Header compression is included by default in HTTP/2 using HPACK. |
| Authentication mechanism | It is relatively secure since it uses digest authentication, NTLM authentication. | Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| Security | SSL is not required but recommended. Digest authentication used in HTTP1.1 is an improvement over HTTP1.0. HTTPS uses SSL/TLS for secure encrypted communication. | Though security is still not mandatory, it is mostly encrypted (though it is not enforced) since almost all clients require traffic to be encrypted. It also has some minimum standards, such as minimum key size for encryption. TLS 1.2 etc. |
| caching | Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |

2.Http version history:

**HTTP/0.9 (1991)** -One Line protocol used to transfer plain HTML files

**HTTPS(1994)** – Netscape created httpps to be used with SSL for its browser

**(1996)-** Concept of headers,version information,status codes were introduced.

**HTTP/1.1 (1997)** -Introduced persistent connection,pipelining,cache control and many other features

**HTTP/2 (2015)** -Based on Googles SPDY allows multiplesing amd server push

**HTTP/3 (2019)** – Based On Googlles QUIC that uses UDP instead of TCP

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

|  |  |
| --- | --- |
| Browser JS | Node.Js |
| Javascript is a programming language that is used for writing scripts on the website. | NodeJS is a Javascript runtime environment. |
| Javascript can only be run in the browsers | NodeJS code can be run outside the browser. |
| It is basically used on the client-side. | It is mostly used on the server-side. |
| Javascript is used in frontend development. | Nodejs is used in server-side development. |
| 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 the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | Nodejs is written in C, C++ and Javascript.s |

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

1. Browser checks cache for DNS entry to find the corresponding [IP address](https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/) of website.  
   It looks for following cache. If not found in one, then continues checking to the next until found.
   * Browser Cache
   * Operating Systems Cache
   * Router Cache
   * ISP Cache
2. If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
   The requests are sent using small data packets that contain information content of request and IP address it is destined for.
3. Browser initiates a [TCP (Transfer Control Protocol)](https://www.geeksforgeeks.org/tcp-and-udp-in-transport-layer/) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
4. Browser sends an [HTTP](https://www.geeksforgeeks.org/http-non-persistent-persistent-connection/) request to the web server. GET or POST request.
5. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, [XML](https://www.geeksforgeeks.org/xml-basics/) and HTML.
6. Server sends out an HTTP response along with the status of response.
7. Browser displays [HTML](https://www.geeksforgeeks.org/html-tutorials/) content
8. Finally, Done.