# **DAY 1 TASK**

Q.NO.1 :- Write a blog on Difference between HTTP1.1 vs HTTP2?

|  |  |
| --- | --- |
| **HTTP1.1** | **HTTP 2** |
| It works on the textual format. | It works on the binary protocol. |
| There is head of line blocking that blocks all the requests behind it until it doesn’t get its all resources. | It allows multiplexing so one TCP connection is required for multiple requests. |
| It uses requests resource Inlining for use getting multiple pages | It uses PUSH frame by server that collects all multiple pages |
| It compresses data by itself. | It uses HPACK for data compression. |
| It was released in 1997 | It was release in 2015 |
| 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. |
| 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. |
| No Stream Prioritization | Improved Stream Prioritization |
| No Header Compression | Header compression using improved algorithms that improve performance as well as security |

Q.No.2 :- Write a blog about objects and its internal representation in Javascript

* Objects in JavaScript are collections of key-value pairs, where the keys can be strings or numbers, and the values can be any data type, including functions and other objects.
* Objects are used to store and manipulate complex data, such as user information, web pages, or game states,etc.
* JavaScript uses different internal representations for objects, depending on their structure and usage. One of the most common representations is a map, which is a data structure that associates keys with values. A map can be implemented using a hash table, a tree, or a linked list. Each object has a pointer to a map that describes its properties and their locations in memory.
* Another representation is an array, which is a data structure that stores values in a contiguous block of memory. Arrays are used for objects that have numeric keys, such as arrays, typed arrays, or arguments objects. Arrays can be faster than maps for accessing elements by index, but they may waste memory if the keys are sparse.
* A third representation is a hidden class, which is a data structure that encodes the shape and behaviour of an object. Hidden classes are used by JavaScript engines to optimize object access and method calls. Hidden classes are created dynamically when objects are created or modified, and they can be shared by objects that have the same structure.
* These are some of the internal representations of objects in JavaScript, but they are not the only ones.
* Different JavaScript engines may use different representations, or switch between them, depending on the performance and memory requirements of the program.
* The internal representations of objects are usually hidden from the developer, and they do not affect the semantics of the language.