**DAY-1 TASK**

1. **Write a blog on Difference between HTTP1.1 vs HTTP2?**

* HTTP is abbreviated as Hypertext Transfer Protocol which is the foundation of data communication on the World Wide Web.
* HTTP is like the language that allows your computer to talk to other computers on the internet. It's the foundation for how information travels from servers to your screen when you're browsing the web.
* Imagine the internet is like a vast postal system where you're constantly sending and receiving letters (data) to access websites and information.
* HTTP is like the language these letters are written in.
* It's a set of rules that computers follow to communicate with each other over the internet.
* HTTP/1.1 and HTTP/2 are the two different version of the HTTP protocol.
* Here are some differences between HTTP/1.1 and HTTP/2.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP/2** |
| * HTTP/1.1 uses multiple connections for parallel downloads of resources. | * HTTP/2 multiplexes multiple requests and responses within a single connection. |
| * HTTP/1.1 sends header information in plain text for each request and response, leading to redundancy and increased overhead. | * HTTP/2 compresses headers, reducing overhead and improving efficiency. |
| * HTTP/1.1 lacks this capability of server push. | * HTTP/2 introduces server push, where the server can proactively send resources to the client's cache without waiting for explicit requests. |
| * HTTP/1.1 relies on a textual format. | * HTTP/2 uses a binary framing layer, making it more efficient to parse and process. |
| * This feature is absent in HTTP/1.1. | * HTTP/2 allows for stream prioritization, enabling the client to specify the importance of different resources. |
| * This feature is not present in HTTP/1.1. | * HTTP/2 supports flow control mechanisms to avoid overwhelming the receiver with data. |
| * HTTP/1.1 connections handle one request and response at a time. | * HTTP/2 multiplexes multiple streams within a single connection, improving efficiency. |
| * HTTP/1.1, offering better mechanisms for handling protocol errors. | * HTTP/2 provides more robust error handling compared to its previous version HTTP/1.1 |
| * HTTP/1.1 does not have this encouragement. | * HTTP/2 encourages the use of HTTPS due to certain browser implementations that require it for HTTP/2 support. |
| * HTTP/1.1 requires an upgrade to a different protocol (like WebSocket) for full-duplex communication. | * HTTP/2 natively supports full-duplex communication without requiring protocol upgrades. |

These are some of the differences that contribute to HTTP/2's improved performance, reduced latency, and better utilization of resources compared to its previous version HTTP/1.1.

1. **Write a blog about objects and its internal representation in JavaScript?**

* Objects in JavaScript can be described as a container that can hold various pieces of information or tools.
* These containers are versatile and can store almost anything you want, from basic pieces of data like numbers and text to more complex things like lists, functions, or even other containers.
* For example, in a toolbox we can keep different things like screwdrivers, hammers, nails, and more. Each tool has its purpose, and we can organize them the way we want.
* Object in JavaScript stores data in the form of Key & values, like how each tool in the toolbox have a label (like "screwdriver" or "hammer"), objects in JavaScript have labels called "keys." These keys are like the names of the things inside the toolbox. Each key has a corresponding "value," which is the actual piece of information or tool it represents.
* We can store even an object inside another object.
* We can give even an array as a value.
* We can write a function inside an object
* Objects can hold different types of values, including numbers, strings, arrays, other objects, and functions. This flexibility allows you to store and organize diverse information.
* It is easy to access the information inside objects using their keys, just like pick a tool from toolbox by reading its label.
* In the below given example code, we can see how an object is store inside another object, how we can write a function inside an object, and how to easily access the data inside it.

**Example Code**

var person = {

name: 'Aasha',

age: 28,

hobbies: ['Gardening', 'Watching TV'],

address: {

city: 'Chennai',

street: 'vanagaram'

},

greet: function() {

return 'Hello everyone! Have a great day';

}

};

console.log(person.name);

console.log(person.hobbies[0]);

console.log(person.address.city);

console.log(person.greet());

Output:

Aasha

Gardening

Chennai

Hello everyone! Have a great day