TASK – DAY - 1

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

HTTP - The Hypertext Transfer Protocol, or HTTP, is an application protocol that has been the de facto standard for communication on the World Wide Web since its invention in 1989.

HTTP 1.1 -

Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web, HTTP is a top-level application protocol that exchanges information between a client computer and a web server. In this process, a client sends a text-based request to a server by calling a method like GET or POST. In response, the server sends a resource like an HTML page back to the client.

This request uses the GET method, which asks for data from the main server.In response to this request,the web server returns an HTML page to the requesting client.The requests and responses will go back and forth between the server and client until the web browser has received all the resources necessary to render the contents of the HTML page on your screen.

CLIENT —>*request* →**SERVER**

SERVER→*response*→**CLIENT**

HTTP 2 -

HTTP/2 began as the SPDY(Speedy) protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization.This protocol served as a template for HTTP/2 when the Hypertext Transfer Protocol which later resulted in the publication of HTTP/2 in May 2015.

From the beginning, many browsers supported this standardization effort, including Chrome, Opera, Internet Explorer, and Safari.From a technical point of view,***HTTP/2 is the binary framing layer.***

HOW IT DIFFERS ??

HTTP/1.1 – *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***.An application level API would still create messages in the conventional HTTP formats, but the underlying layer would then convert these messages into binary.

This ensures that web applications created before HTTP/2 can continue functioning as normal when interacting with the new protocol.The conversion of messages into binary allows HTTP/2 to try new approaches to data delivery not available in HTTP/1.1, a contrast that is at the root of the practical differences between the two protocols.In HTTP/2, the binary framing layer encodes requests/responses and cuts them up into **smaller packets of information**

Greatly increasing the flexibility of data transfer.As opposed to **HTTP/1.1**, which must make **use of multiple TCP connections** to lessen the effect of HOL blocking, **HTTP/2** establishes a **single connection** object between the two machines. Within this connection there are multiple streams of data. Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into **smaller units called frames**.

CONCLUSION -

As you can see,HTTP/2 differs from HTTP/1.1,with some features providing greater levels of control that can be used to better optimize web application performance. You can consider how such factors as multiplexing, stream prioritization, flow control and compression in **HTTP/2** will affect the changing landscape of web development.

TASK – DAY - 1

A blog about objects and its internal representation in Javascript

Objects, in JavaScript, is it's most important **data-type**(non-primitive) and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript's primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each (depending on their types).

An object is a **reference data type**. Variables that are assigned a reference value are given a reference or a pointer to that value. That reference or pointer points to the location in memory where the object is stored.**The variables don't actually store the value** rather it stores the references of the object.

Loosely speaking, objects in JavaScript, **in the form of "key: value" pairs**. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

The **properties** of an object define the **characteristics of the object**. You access the properties of an object with a simple dot-notation.For Eg. If your **object is a student**, it will have **properties like name, age, address, id**, etc and methods like **student.name,student.age** etc.

Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive.

```
var myBike = new Object();
myBike.color=”red”;
myBike.maker = “RE”;
myBike.model = “Classic 350 - Stealth Black”;
```

One of easiest way to create a javascript object is object literal, simply define the property and values inside curly braces as shown below:

```
let bike = {color:”red”,maker:”RE”,model:“Classic 350 - Stealth
Black”};
```