**Task 1:**

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

Hypertext Transfer Protocol, or HTTP- for communication on the World Wide Web- its invention in 1989- Developed by Timothy Berners-Lee.

HTTP is a top-level application protocol that exchanges information between a client computer (sends a text-based request to a server by calling a method like GET or POST.) and a local or remote web server using the Transmission Control Protocol, or TCP.

🡪 In response, the server sends a resource like an HTML page back to the client. in addition to any images, stylesheets, or other resources called for in the HTML.

Ex: GET /index.html HTTP/1.1

Host: [www.example.com](http://www.example.com)

**Difference between HTTP1.1 vs HTTP2**

|  |  |  |
| --- | --- | --- |
| **S NO** | **HTTP1.1** | **HTTP2** |
|  | It Is released in **1997**. | It Is released in **2015**, by the Internet Engineering Task Force (IETF). |
|  | It keeps all requests and responses in plain **text format**. | It is **binary** instead of text format. Binary framing layer (encodes requests into binary). |
|  | The client had to break and remake the **TCP connection with every new request**. | Using a single TCP connection to deliver **multiple requests** and responses **in parallel**. |
|  | Not try new approaches to data delivery | try new approaches to data delivery, and the page starts to render straight away. |
|  | **head-of-line** (HOL) blocking- a request at the head of the queue that cannot retrieve its required resource will block all the requests behind it. | It is fully **multiplexed**, instead of ordered and blocking (binary framing layer, greatly increasing the flexibility of data transfer. |
|  | Solving the problem by introducing persistent connections and pipelining | It offered several methods to decrease latency. |
|  | Multiple TCP connections to lessen the effect of HOL blocking. | It can use one connection for parallelism. Multiple streams of data in a single connection and Each stream consists of multiple messages in the familiar request/response format. Finally, each of these messages split into smaller units called frames. |
|  |  | It can **prioritize** the responses. It can set a numeric prioritization in a batch of requests. such as getting a webpage CSS before its JS files |
|  | This version added **six extra methods** (using in already existing resources.): PUT, PATCH, DELETE, CONNECT, TRACE, and OPTIONS, |  |
|  | It is a sequential protocol. So, we can send a single request at a time. | It can do multiple requests at the same time using a single connection. |
|  | Must explicitly require the compression of requests and responses. | It is, executes a GZip compression automatically |
|  | This version added many performance enhancements, including keepalive connections, caching mechanisms, request pipelining, transfer encodings, and byte-range requests. | A functionality that allows closing a connection between a server and a client for some reason, thus immediately opening a new one |
|  | It is allowed multiple requests, but the number of requests was limited to around 6 or 8, depending on the browser. | It has **server push** functionality. With that, the server tries to predict the resources that will be requested soon. So, the server proactively pushes these resources to the client cache. |
|  | For to do multiple parallel requests, must use a technique such as Domain Sharding. | It Removes unnecessary HTTP/1.1 Domain Sharding, Image Sprites, etc. |
|  |  | More secure, as security concerns associated with textual attacks, such as splitting, are no longer relevant. |
|  |  | Less error-prone than HTTP/1.1. |
|  |  | Used to header compression to reduce **overhead** |

1. Write a blog about objects and its internal representation in Javascript.

**JavaScript Object:**

**Syntax:**

var objectName = {

key1: value1,

key2: value2

}

Key: Value pair is an Object member, these are separated by commas and enclosed in curly braces

Ex:

const student = {

name: 'John',

age: 21

};

console.log(typeof person); // object

or can define as in a single line,

const student = { name: 'John', age: 21 };

Accessing Object Properties in dot notation/ bracket Notation:

Ex:

const student = {

name: 'John',

age: 21,

};

// accessing in dot notation console.log(student.name); // John

console.log(student ["name"]); // John

// accessing property

JavaScript Nested Objects

An object can also contain another object.

Ex:

const student = {

name: 'John',

age: 21,

marks: {

Tamil: 80,

English: 65

}

}

console.log(student.marks);

// {Tamil: 80, English: 65}

console.log(student.marks.English); // 65

JavaScript Object Function

In JavaScript, an object can also contain a function.

Ex:

const student = {

name: 'John',

age: 21,

award: function(){

console.log('Distinction')

} }

//call function

student. award (); // Distinction

-------------------------------------------