DAY 1- Activities

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

HTTP1.1	HTTP2
→ HTTP stands for hypertext transfer protocol & it is used in client-server communication. HTTP/1.1 which was created in 1997	→ By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP. The new HTTP/2 which was created in 2015.
→ 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.
→ HTTP/1.1 did not anticipate.	→ HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.

Day –1

→ Prioritization affects a webpage's load time. For example, certain resources, like large JavaScript files, may block the rest of the page from loading if they have to load first.	→ HTTP/2 offers a feature called weighted prioritization. This allows developers to decide which page resources will load first, every time. In HTTP/2, when a client makes a request for a webpage, the server sends several streams of data to the client at once, instead of sending one thing after another.
→ **Server push:** Typically, a server only serves content to a client device if the client asks for it. However, this approach is not always practical for modern webpages, which often involve several dozen separate resources that the client must request.	→ **Server push:** HTTP/2 solves this problem by allowing a server to "push" content to a client before the client asks for it. The server also sends a message letting the client know what pushed content to expect
→ Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 and HTTP/2 compress HTTP messages to make them smaller.	→ HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involved in loading even a single webpage, those bytes add up quickly, resulting in faster loading.

2. Write a blog about objects and its internal representation in JavaScript.

→ A **JavaScript object** is a collection of **properties** and **methods** that define the characteristics and behaviors of an entity. Properties are name-value pairs that can store any type of data, such as strings, numbers, Booleans, functions, or other objects. Methods are properties that contain functions as values, and they can perform actions on the object or its data.

→ There are two ways to create a JavaScript object: using an **object literal** or using an **object constructor**

Day –1

- Object Literal

An object literal is a syntax that allows you to define and create an object with a set of properties and methods enclosed in curly braces `{}`. Each property or method is separated by a comma, and each name-value pair is separated by a colon.

- Object Constructor

An object constructor is a function that defines the template or blueprint for creating objects of the same type. The function name usually starts with a capital letter to distinguish it from regular functions. The function can take parameters to initialize the properties of the object, and it can also define methods using the `this` keyword.

→ JavaScript also provides some built-in object constructors, such as `Object`, `Array`, `String`, `Number`, `Date`, etc., that you can use to create objects of different types[2]. For example:

```
// Create an object using the Object constructor
const obj = new Object();
obj.name = "Alice";
obj.age = 20;
// Create an array using the Array constructor
const arr = new Array(1, 2, 3);
// Create a string using the String constructor
const str = new String("Hello");
// Create a number using the Number constructor
const num = new Number(42);
// Create a date using the Date constructor
const date = new Date();
```

The syntax to declare an object is:

```
const object_name = {
  key1: value1,
  key2: value2
}
```