Task-1

1.Difference between http 1.1 vs http2

Multiplexing: HTTP/1.1 sends and receives requests and responses sequentially, which means only one request can be handled at a time over a single TCP connection. In contrast, HTTP/2 supports multiplexing, allowing multiple requests and responses to be sent and received in parallel over a single connection. This can significantly improve performance, especially for web pages with many resources.

1. Binary Protocol: HTTP/2 is a binary protocol, whereas HTTP/1.1 is a text-based protocol. The binary format of HTTP/2 allows for more efficient parsing and reduced overhead compared to the text-based format of HTTP/1.1.
2. Header Compression: HTTP/2 introduces header compression, which reduces overhead by compressing header data before transmission. This can lead to faster load times and reduced bandwidth usage compared to HTTP/1.1, especially for requests with large headers.
3. Server Push: HTTP/2 supports server push, a feature that allows servers to proactively send resources to the client before they are requested. This can be used to efficiently send resources that the server knows the client will need, potentially reducing the number of round trips required to load a web page.
4. Stream Prioritization: HTTP/2 introduces stream prioritization, allowing clients to assign priority levels to individual streams. This enables more efficient resource allocation and can improve the overall performance of web applications.

2.Object and its internal representation in javascript

In JavaScript, objects are fundamental data types and are used to store collections of key-value pairs. They are versatile and can represent various entities, from simple data structures to complex entities like functions and prototypes. Understanding how objects are internally represented in JavaScript can provide insights into their behavior and usage.

### **Internal Representation of Objects:**

1. Properties and Methods:
   * Objects in JavaScript consist of properties and methods.
   * Properties are essentially key-value pairs where the key is a string (or a Symbol) and the value can be any JavaScript data type, including other objects.
   * Methods are functions that are associated with an object and can be invoked using dot notation.
2. Prototype Chain:
   * Each object in JavaScript has a prototype, which is another object from which it inherits properties and methods.
   * If a property or method is not found on the object itself, JavaScript looks up the prototype chain until it finds it or reaches the end of the chain.
   * This mechanism allows for inheritance and the creation of object hierarchies.
3. Hidden Classes and Inline Caching:
   * JavaScript engines like V8 use hidden classes to optimize object property access.
   * When an object is created, the JavaScript engine assigns it a hidden class based on its structure.
   * If an object's structure changes (e.g., by adding or removing properties), the hidden class may change, potentially impacting performance.
   * Inline caching is a technique used by JavaScript engines to speed up property access by caching property lookups based on an object's hidden class.
4. Property Descriptors:
   * Each property in JavaScript has associated property descriptors, which define its behavior.
   * Property descriptors include attributes such as value, writable, enumerable, and configurable.
   * These descriptors determine whether a property can be modified, enumerated, or deleted.
5. Creating Objects:
   * Objects in JavaScript can be created using object literals, constructor functions, or the class syntax introduced in ECMAScript 2015 (ES6).
   * Object literals provide a simple way to create objects with predefined properties and methods.
   * Constructor functions and class syntax allow for the creation of objects with custom behaviors and inheritance.

3.codeketa practice

// Creating an object using object literal

let person = {

name: "John",

age: 30,

greet() {

console.log(`Hello, my name is ${this.name} and I'm ${this.age} years old.`);

}

};

// Accessing properties and invoking methods

console.log(person.name); // Output: John

person.greet(); // Output: Hello, my name is John and I'm 30 years old.

4.IP address, port, HTTP methods, MAC address.

**1. IP Address (Internet Protocol Address):**

* An IP address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.
* IP addresses serve two main functions: host or network interface identification and location addressing.
* There are two versions of IP addresses currently in use: IPv4 (Internet Protocol version 4) and IPv6 (Internet Protocol version 6). IPv4 addresses are represented as four numbers separated by dots (e.g., 192.168.1.1), while IPv6 addresses are represented in hexadecimal format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
* IP addresses allow devices to communicate with each other over a network by routing data packets to their destination.

### **2. Port:**

* Ports are virtual endpoints used for communication in computer networks.
* In the context of networking, a port is a communication endpoint that is associated with a specific IP address.
* Ports are identified by numbers ranging from 0 to 65535.
* Ports are used to differentiate between different types of network traffic (e.g., web traffic, email traffic, etc.) on the same IP address.
* Well-known ports (0-1023) are reserved for specific protocols (e.g., port 80 for HTTP, port 443 for HTTPS).

### **3. HTTP Methods (Hypertext Transfer Protocol Methods):**

* HTTP methods are verbs used by clients to request actions to be performed on a resource identified by a URL (Uniform Resource Locator).
* Some common HTTP methods include:
  + GET: Retrieves data from a specified resource.
  + POST: Submits data to be processed to a specified resource.
  + PUT: Updates a specified resource with new data.
  + DELETE: Deletes the specified resource.
  + PATCH: Partially updates a resource with new data.
  + HEAD: Retrieves metadata about a resource without transferring the entire content.
  + OPTIONS: Returns the supported HTTP methods for a specified resource.
* These methods define the operations that can be performed on resources hosted on web servers.

### **4. MAC Address (Media Access Control Address):**

* A MAC address is a unique identifier assigned to a network interface controller (NIC) for communications on a network.
* Unlike IP addresses, which can change based on network configuration, MAC addresses are hardcoded into the network interface hardware by the manufacturer.
* MAC addresses are used at the data link layer of the OSI model to uniquely identify devices within a network segment.
* MAC addresses are usually represented as a series of hexadecimal numbers separated by colons or hyphens (e.g., 00:1A:2B:3C:4D:5E).
* MAC addresses facilitate the routing of data packets within a local network (LAN) and are used for technologies like Ethernet and Wi-Fi.