ROADMAP DAY-1 TASK M.SWATHI

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

HTTP1.1	HTTP2
1. HTTP/1.1 has been the backbone of the World Wide Web for over a decade, facilitating the exchange of information between clients and servers.	1. HTTP/2, introduced in 2015, addresses the shortcomings of its predecessor by introducing several key features aimed at optimizing the communication between clients and servers.
2. HTTP/1.1 Processes requests and responses sequentially, leading to potential delays (head-of-line blocking).	2. Allows multiple streams of data to be sent and received simultaneously over a single connection, reducing latency and improving page load times.
3. Headers are sent as plain text, leading to unnecessary overhead.	3. Utilizes header compression to reduce the size of headers, minimizing the amount of data transmitted and improving overall efficiency.
4. In HTTP/1.1 it relies on a plain text format	4. Adopts a binary protocol, simplifying parsing and enhancing processing efficiency for both browsers and servers.
5. Lacks the ability for the server to proactively push resources to the client.	5. Employs server push, allowing servers to anticipate and push resources to the client without waiting for explicit requests, further optimizing page loads.
6. Security is optional, leading to potential vulnerabilities.	6. Requires the use of Transport Layer Security (TLS), enhancing security by encrypting data during transmission.
7. Resource loading depends on the order of requests and responses.	7. Resources can be loaded in parallel, reducing latency and improving overall loading times.
8. Text-based protocol with a more human-readable format.	8. Utilizes a binary frame structure for more efficient processing by machines.
9. Widely supported by all major browsers	9. Most modern browsers have adopted support for HTTP/2, ensuring compatibility and enabling users to benefit from its performance improvements
10. It compresses data by itself..	10. It uses HPACK for data compression.

2) Write a blog about objects and its internal representation in Javascript?

OBJECT:

- In JavaScript, an object is a standalone entity, with properties and type. It is a collection of key-value pairs, where each key is a string (or Symbol) and each value can be any data type, including other objects.
- Objects in JavaScript are used to store and manipulate data, and they are often used to represent real-world entities.

INTERNAL REPRESENTATION OF OBJECTS:

- Internally, objects in JavaScript are represented as a collection of properties, with each property consisting of a key and a reference to its corresponding value.
- When an object is created, memory is allocated to store the object's properties and their values.
- The internal representation typically involves a hash table, a specialized data structure that maps keys to values. This hash table enables the engine to quickly retrieve and manipulate object properties.

THE ROLE OF MEMORY MANAGEMENT:

- Objects, being dynamic data structures, consume memory during their existence.
- Memory management, the process of allocating and deallocating memory as objects are created and destroyed, plays a crucial role in JavaScript's efficiency.
- JavaScript engines employ sophisticated memory management algorithms to ensure that objects are efficiently allocated and released from memory, preventing memory leaks and ensuring that memory usage remains optimized.

THE PROTOTYPE CHAIN:

- JavaScript objects inherit properties and methods from their prototypes, forming a hierarchical inheritance structure.
- This mechanism allows objects to share common characteristics and functionalities without explicitly defining them within each instance.
- When accessing a property or method that is not directly defined within an object, JavaScript seamlessly traverses the prototype chain, searching for the first occurrence of the desired property or method.