Ganesh Kulkarni

GUVI Zen Class Dec – 20 Batch

**Task - 1 Assignment**

**Question 1: Difference between HTTP 1.1 and HTTP 2:**

* HTTP to supports request multiplexing in which multiple requests can be forwarded parallels on a single TCP connection. This is one of the differences between HTTP 2 and HTTP one, as it does not support request multiplexing.
* Due to parallel request manipulation of HTTP 2, it makes website loading faster due to a round trip time RTT reduction.
* HTTP 1 is the text protocol where II at processes text command to complete request-response cycle.
* On the other hand, HTTP, two is based on a binary protocol where binary commands in the form of zeros and ones are useful for the task's r execution.
* There are fewer chances of errors to happen during the transmitting of packets in HTTP 2.
* Apart from parallel multiplexing, HTTP 2 also supports different features such as prioritization compression flow control and effective handling of TLS, which are not present in HTTP 1.
* In HTTP 2.0, the server can be privatized the post resources; this is considered one of the vital performance differences between HTTP 2 and HTTP 1.

**Question 2: History of HTTP Versions:**

* HTTP stands for hypertext transfer protocol, which Tim Berners Lee invented from 1989 to 1991. HTTP is primarily based on the request-response model in a client-server architecture.
* The internet engineering task force and World Wide Web Consortium W3C develop HTTP standard. Following are the different HTTP version which is evolved during the time:

1. **HTTP 0.9:**

* It is the initial version of the HTTP protocol, which is also known as one line protocol.
* A simple client server-based model, which works on request-response type Telnet friendly protocol.
* HTTP 0.9 only supports the GET method.

1. **HTTP 1.0:**

* The HTTP 1.0 is browser friendly protocol that is also known for its building extensibility.
* HTTP 1.0 supports header field consists of reach metadata such as HTTP version number status code and content type for both request and response communication.
* The HTTP 1.0 protocol supports the GET HEAD and POST method.

1. **HTTP 1.1**

* This is the most common and popular version of HTTP, which is also currently in use.
* It supports performance optimization using different features such as:
* Persistent and pipelined connections, chunked transfers, compression/decompression, content negotiations, virtual hosting.
* The HTTP 1.1 comes with different methods for request and response communication such as GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS.

1. **HTTPS**

* It is one of the secured forms of the HTTP protocol, which comes with SSL / TLS for a secured encryption mechanism.

**Question 3: Difference between browser JS and node JS**

Following is the difference between browser JS and node JS:

* 1. The browser JS is the JavaScript programming language that runs in any e browser with a JavaScript engine. On the other hand, node, JS is the interpreter / Runtime environment for the JavaScript programming language.
  2. Browser JS is primarily used for different client-side activities such as user authentication, event-driven operations, etc., where node JS is used to access and perform operations on the operating specific devices for no blocking operations JavaScript.
  3. There are different browser-specific JavaScript engines available such as spider monkey (Firefox), JavaScript Core (Safari), V8 (Google Chrome), which makes it very easy to write a JavaScript code and run it in any Browser. In the case of node JS, it only supports the V8 engine provided by Google.
  4. Browser JS does not support standalone application where it can only run in browsers, which support JavaScript engine on the other hand node JS supports standalone application environment where it can work as an interpreter for running JavaScript code in the operating system.
  5. When we write a JavaScript code and run it in the browser, we are mainly interacting with the Dom elements; on the other hand, node.js does not interact with the Dom elements. For example, we do not have documents; the browser in the node JS environment supports Windows objects.

**Question 4: What happens when we type URL in the browser?**

Following is the sequence of the action happens when we type URL in the browser:

* At the very first, the browser checks for cache whether the DNS record of the URL is available for finding out the corresponding IP address of that particular URL.
* If that requested URL is not available in the cache, then ISP’s DNS server will initiate a DNS query to find whether the server's IP address hosts the particular URL.
* After that, the browser starts a TCP connection with that specific server.
* After initiating the TCP connection, the browser sends an HTTP request to the webserver; the server will handle the request and ship the response with the requested data/web page.
* The browser renders the webpage/response received from the server on the browser's display window.