**TASK 01**

1. **Difference between HTTP 1.1 vs 2?**

**Ans.**

**HTTP 1.1:**

* HTTP1.1 can only load requests one at a time.
* HTTP1.1 can only process one request per one TCP connection.
* HTTP1.1 made browsers run parallel requests to multiple TCPs for the same Web asset.
* HTTP1.1 clogs up “the wire” with multiple duplicate data requests, and can hurt performance if too many requests are made.

**HTTP 2:**

* HTTP2 is binary, instead of textual.
* HTTP2 is fully multiplexed, instead of ordered and blocking.
* HTTP2 can, therefore, use one connection for parallelism.
* HTP2 uses header compression to reduce overhead.
* HTTP2 allows servers to “push” responses proactively into client caches.

1. **How does the the browser actually render a Website?**

**Ans.**

**High Level Flow:**

Paint

Layout

Render Tree

Parse HTML

Parse CSS

**Parsing HTML:**

* HTML is forgiving by nature.
* Parsing isn’t straight forward.
* Can be halted.
* Will do speculative parsing.
* It’s reentrant.

**Parsing CSS:**

* Standard parsing.
* It has the structure of CSS Object Model.

**Render/Frame Tree:**

* Combines the Two object models, style resolution.
* This is the actual representation of what will show on screen.
* Not a 1 to 1 mapping of your HTML.

**DOM node to render Object:**

* Visual output.
* Geometric Information.
* Can layout and paint.
* Hold style and computed metrics.

1. **List 5 differences between Browser JS(Console) vs Node JS?**

**Ans.**

* In browser “window” is a predefined global object which has functions and attributes, where as Nodejs doesn’t have it.
* In browser “location” is another predefined object, where as Nodejs doesn’t have it.
* In browser “require” is not predefined object, where as Nodejs has it.
* In browser module is not required, where as in Nodejs you have to keep your code inside the module.
* In browser “document” is a predefined object, where as Nodejs doesn’t have it.

1. **What happens when you type a url in a browser?**

**Ans.**

* You enter a URL into a web browser
* The browser looks up the IP address for the domain name via DNS
* The browser sends a HTTP *request* to the server
* The server sends back a HTTP *response*
* The browser begins rendering the HTML
* The browser sends requests for additional objects embedded in HTML (images, css, JavaScript) and repeats steps 3-5.
* Once the page is loaded, the browser sends further async requests as needed.