**Task-1**

**1. Difference between http 1.1 and http2**

Ans:

HTTP/2 can send multiple requests for data in parallel over a single TCP connection. This allows users to download web files asynchronously from one server. This removes the order and blocking constraints of the HTTP/1.x protocol.

HTTP/2 compresses a large number of header frames. It uses HPACK specification as the simple and secure approach to header compression. This removes duplication and overhead of data transmitted.

HTTP/1.1 used to process the text commands to complete request and response cycles, HTTP/2 uses binary commands to execute the same tasks. Hence, improved optimization techniques can now be implemented

HTTP2 Server push capability allows the server to send additional cacheable information to the client that isn’t requested but is anticipated in the future requests.

**2. Http Version hitory**

Ans:

HTTP (Hyper Text Transfer Protocol is an underlying protocol of the world wide web. It is developed by tim-berners-lee and his team.

HTTP/0.9:

Requests consist of a single line, and start with only possible method GET followed by the path to the resource.

The response was very simple; It only consisted of the file itself .

There were no http headers(only html file could be transferred). There were no status or error codes.

HTTP/1.0:

Versioning information is sent within each request(http1.0 is appended to the GET line).

Status code line is also sent at the beginning of the response, allowing browser itself to understand the success or failure of the request

Notion of the HTTP headers has been introduced, both for requests and responses, allowing the metadata to be transferred and making the protocol extremely flexible and extensible

with the help of the new httpheaders, the ability to transmit the other documents then the plain html files has been added.

It had a lot of interoperability problems.

HTTP/1.1:

HTTP 1.1 clarified ambiguities and introduced numerous improvements.

A Connection can be reused, saving the time to reopen it numerous times.

Pipelining has been added, allowing to send the second request, before the answer for the first one is fully transmitted, lowering the latency of the communication

chunked responses are also supported

additional cache mechanisms have been introduced.

HTTP/2:

In the first half of 2010, google demonstrated the alternative way of exchanging data between client and server, by implementing the experimental protocol SPDY. This amassed the interest from the developers working on both browsers and servers. Defining an increase in responsiveness, and solving the problem of the duplication of data transmitted. SPDY served as foundation of the http2 protocol

Post HTTP/2 evolution:

HTTP didn’t stop evolving upon the release of HTTP/2. Like with HTTP/1.x previously, HTTPs extensibility is still being used to add new features:

Support of Alt-Svc allows the dissociation of the identification and the location of the given resource, allowing for the smarter CDN caching mechanism.

The introduction of the client-Hints allows the browser or client, to proactively communicate information about its requirements, or hardware constraints to the server.

The introduction of the security related prefixes in the cookie header, now guarantee a secure cookie has not been altered.

This evolution of HTTP proves its extensibility and simplicity, liberating creation of many applications and compelling the adoption of the protocol.

**3. List 5 difference between Browser JS(console) vs Nodejs**

Ans:

Both browser and node.js use javascript as their programming language. Building apps that run in browser is completely different thing then building a node.js application:

Node.js brings a huge advantage because front end and back end can be done using single programming language javascript.

In browsers we interact with DOM or web platform APIs like cookies. These do not exist in Node.js, we don't have document, window etc that are provided by the browser.

In browsers we don’t have all the nice APIs that Node.js provides through its modules, like the file system access functionality.

Usually in Node.js we control the environment, we know which version of node.js we run the application on. But, in a browser environment, we don't get the luxury to choose what browser our visitors will use.

Since the javascript moves so fast, browsers can be a bit slow to upgrade, sometimes on the web we are stuck with using older javascript. We may need Bable to transform our code to be ES-5 compatible before shipping it to the browser, but in node.js we won’t need that.

Node.js uses the common js module system, while in browsers we are starting to see the ES modules standard being implemented. Ex: require in Node.js and import in browser.

**4. what happens when you type a URL in the address bar in the browser?**

Ans:

When we type url in the browser:

Brower looks for the IP address for the domain name via Domain Name Server.

Once the IP address matched, browser sends HTTP request to the server then server sends back the HTTP response.

The browser begins rendering html on the page while also requesting any additional resources such as CSS, JavaScript, images etc.

Each subsequent request completes the request/response cycle and is rendered in turn by the browser.

Then once page is loaded if necessary some sites will make further asynchronous requests