**Task – 01**

1. Difference between HTTP1.1 vs HTTP2

* HTTP/1.1 Developed by Timothy Berners-Lee in 1989 as a communication standard for the World Wide Web.
* HTTP/2 began as the SPDY protocol, developed primarily at Google with the intention of reducing web page load latency by using techniques such as compression, multiplexing, and prioritization.
* HTTP/1.1 was the third version of HTTP and the standard protocol for over 15 years. It introduced persistent connections for improved performance and laid the foundation for standard requests, such as GET, HEAD, PUT, and POST.
* HTTP/2 was released in 2015 as a major revision to the HTTP/1.1 protocol. It was derived from the SPDY protocol as a way to improve the online experience by speeding up page loads and reducing round-trip time (RTT), especially on resource-heavy web pages.
* HTTP/1.1, which keeps all requests and responses in plain text format.
* HTTP/2 uses the binary framing layer to encapsulate all messages in binary format, while still maintaining HTTP semantics, such as verbs, methods, and headers.
* HTTP/1.1’s limitations began to show as websites became more resource-intensive. Specifically its use of one outstanding request per TCP connection created significant overhead, slowing down page load times.
* While HTTP/2 was initially modeled after SPDY, it was soon modified to include unique features, including a fixed header compression algorithm in contrast to SPDY’s dynamic stream-based compression which increased the speed of page load.
* In HTTP/1.1 multiple data packets cannot pass each other when traveling to the same destination, there are situations in which a request at the head of the queue that cannot retrieve its required resource will block all the requests behind it. This is known as head-of-line (HOL) blocking, and is a significant problem with optimizing connection efficiency in HTTP/1.1.
* In HTTP/2 the communication channel consists of a bunch of binary-encoded frames, each tagged to a particular stream. The identifying tags allow the connection to interleave these frames during transfer and reassemble them at the other end. The interleaved requests and responses can run in parallel without blocking the messages behind them, a process called multiplexing. Multiplexing resolves the head-of-line blocking issue in HTTP/1.1 by ensuring that no message has to wait for another to finish. This also means that servers and clients can send concurrent requests and responses, allowing for greater control and more efficient connection management.

2. HTTP Version History

* HTTP/0.9 - 1991
* HTTP/1.0 - 1996
* HTTP/1.1 - 1997
* HTTPS - 1999
* HTTP/2.0 - 2015

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

|  |  |
| --- | --- |
| **Javascript** | **NodeJs** |
| * Javascript is a programming language that is used for writing scripts on the website. It can only be run in the browsers. | * NodeJS is a Javascript runtime environment. NodeJS code can be run outside the browser. |
| * Javascript is basically used on the client-side. Javascript is capable enough to add HTML and play with the DOM. | * NodeJS is mostly used on the server-side. Nodejs does not have capability to add HTML tags. |
| * Some of the javascript frameworks are RamdaJS, TypedJS, etc. | * Some of the Nodejs modules are Lodash, express etc. These modules are to be imported from npm. |
| * Javascript can run in any browser engine as like JS core in safari and Spidermonkey in Firefox and it is used in frontend development. | * Nodejs can only run in V8 engine of google chrome and It is used in server-side development. |
| * Javascript is the upgraded version of ECMA script that uses Chrome’s V8 engine written in C++. | * Nodejs is written in C, C++ and Javascript. |

1. What happens when you type a URL in the address bar in the browser?

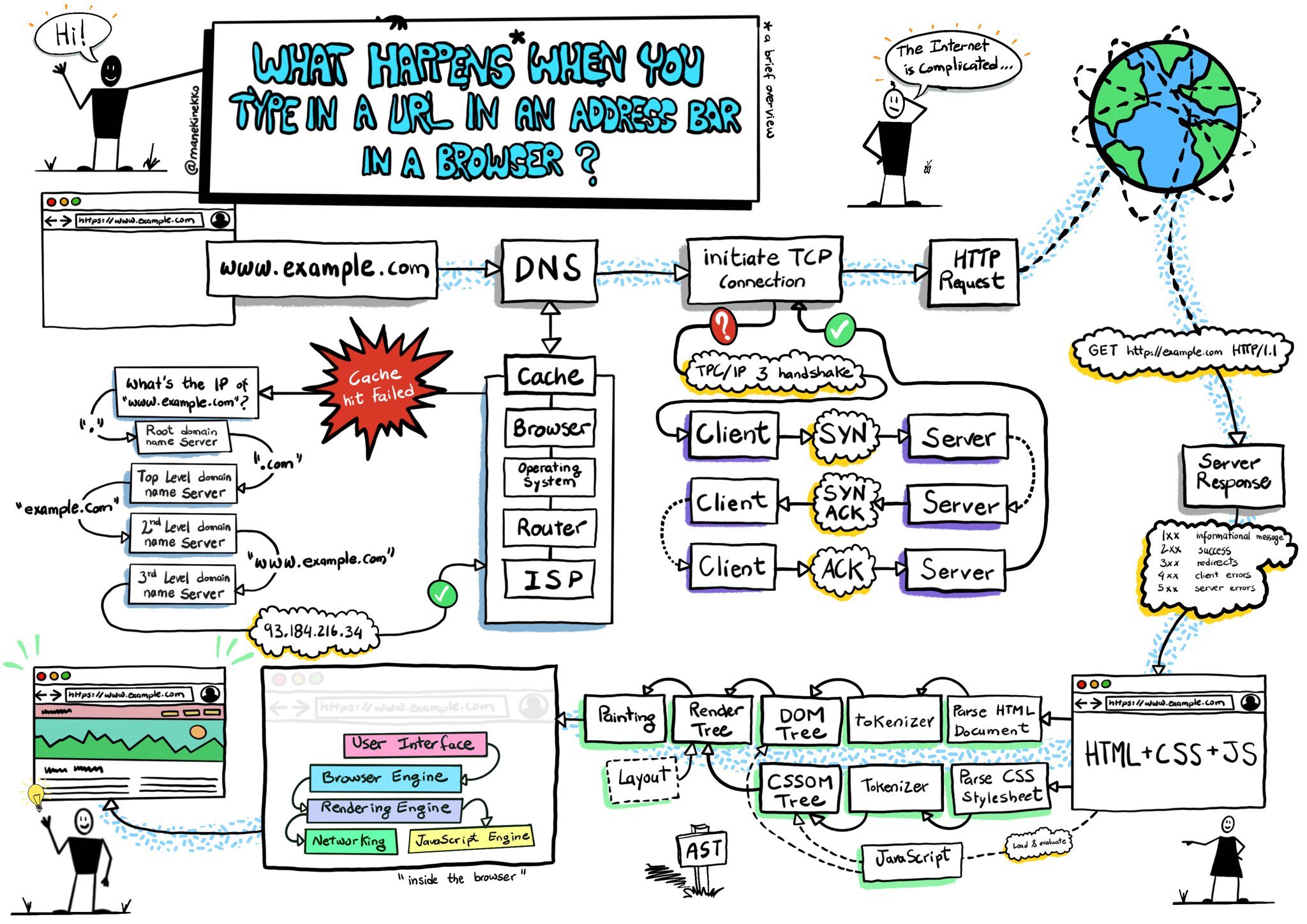


Fig 1.0 what happens when URL is entered in a browser.

* Browser checks cache for DNS entry to find the corresponding [IP address](https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/) of website.  
  It looks for following cache. If not found in one, then continues checking to the next until found.
  + Browser Cache
  + Operating Systems Cache
  + Router Cache
  + ISP Cache
* If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
  The requests are sent using small data packets that contain information content of request and IP address it is destined for.
* Browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize (SYN) and acknowledge(ACK) messages.
* Browser sends an [HTTP](https://www.geeksforgeeks.org/http-non-persistent-persistent-connection/) request to the web server. GET or POST request.
* Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, [XML](https://www.geeksforgeeks.org/xml-basics/) and HTML.
* Server sends out an HTTP response along with the status of response.
* Browser displays all the content.