TASK 1

1. Evolution of HTTP

Invented by Tim Berners-Lee at CERN in the years 1989-1991, HTTP is the language of communication on World Wide Web.

HTTP works as a request-response protocol between the client-server computing model. HTTP standards are developed by IETF(Internet Engineering Task Force) and W3C(World Wide Web), culminating in the publication of a series of Request for Comments(RFCs). HTTP has four versions till date. HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2.

HTTP/0.9 ---- The One-line Protocol -----1991

- Initial version of HTTP, a simple client-server, request response, telenet-freindly protocol.
- Request nature: single-line(method + path for requested document)
- Methods supported: GET only.
- Response Type: hypertext only.
- Connectio Nature: terminated immediately after the response.
- No HTTP headers(cannot transfer other content type files). No status/error codes, No URLs, No versioning
- Web servers like Apache and Nginx still support HTTP/0.9.

HTTP/1.0 ----- Building Extensibility ------ 1996

- Browser friendly protocol.
- Provided header fields including rich metadata about both request and response(HTTP version number, status code, content type)
- Response: not limited to hypertext(Content-type header provided ability to transfer files other than plain HTML files- e.g scripts, media)
- Methods supported: GET, HEAD, POST.
- Connection Nature: terminated immediately after the response.

Establishing a new connection for each request - major issue in both HTTP/0.9, and HTTP/1.0

HTTP/1.1 ----The standardized protocol----1997

- This is the currently used most common version of HTTP.
- Introduced critical performance optimizations and feature enhancements --
    - Persistent and pipelined connections
    - Chunked transfers.
    - Compression/ decompression
    - Content negotiations
    - Virtual hostings(a server with single IP address hosting multiple domains)
    - Faster response and great bandwidth savings by adding cache support.
- Methods supported: GET HEAD POST PUT DELETE TRACE OPTIONS
- Connection nature: long-lived.

HTTP/2.0 and beyond -----2015

- Evolution of HTTP -- MDN Web Docs
- HTTP keep-alive header
- Protocol upgrade mechanism - MDN web docs
- Brief History of HTTP - High performance browser networking
- Improves page loads and reduces round trip time especially on resource-heavy pages.
- Included more parallelism in request-respond cycles.

2. Difference between HTTP/1.1 and HTTP/2.0

HTTP/2 improved on HTTP/1.1 in a number of ways that allowed for speedier content delivery and improved user experience.

- Binary Protocols: consume less badwidth, are more efficiently parsed and are less error-prone than textual protocol.
- Multiplexing: HTTP/2 is multiplexed, i.e it can initiate multiple requests in parallel over a single TCP connection. Resolves head-of-line problem in HTTP/1.1 in which a packet at the front line blocks all other packets from being transmitted.
- Header Compression: to reduce the overhead caused by TCPs slow-start mechanism
- Server Push -- HTTP/2 servers push likely-to-be-used resources into a browser cache, even before they're requested.
- Increased security -- Web browser only support HTTP/2 via encrypted connections, increasing user and application security.

3. Difference between BrowserJS and NodeJS

Both the browser and Node.js use JavaScript as their programming language. However, building apps that run in the browser is a completely different thing than building a Node.JS application.

- JS is a programming language used for writing scripts on the website, NodeJs is a javascript runtime environment.
- Javascript can only be run in the browsers while NodeJS code can run outside the browser.
- JS is mainly used in client-side.NodeJs mostly used on the server side.
- JS can add HTML, NodeJs does not have capability to add HTML tags.
- JS can run in any browser engine while NodeJs can only run in V8 engine of google chrome.

4. What happens when we enter URL

URL(Universal Resource Locator), is a reference to a resource on the internet.

- Browser checks cache for DNS(Domain Name System) entry to find the corresponding IP address of the website. It looks in following cache, if not found in one then continues checking the next until found.
    - Browser Cache.
    - Operating Systems Cache
    - Router Cache
    - ISP Cache

- If not found in cache ISP’s 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 connection with the server(if DNS query resolved) using synchronize(SYN) and AcknowledgeI(ACK) messages.
- Browser sends an HTTP 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 and HTML.
- Browser displays HTML content