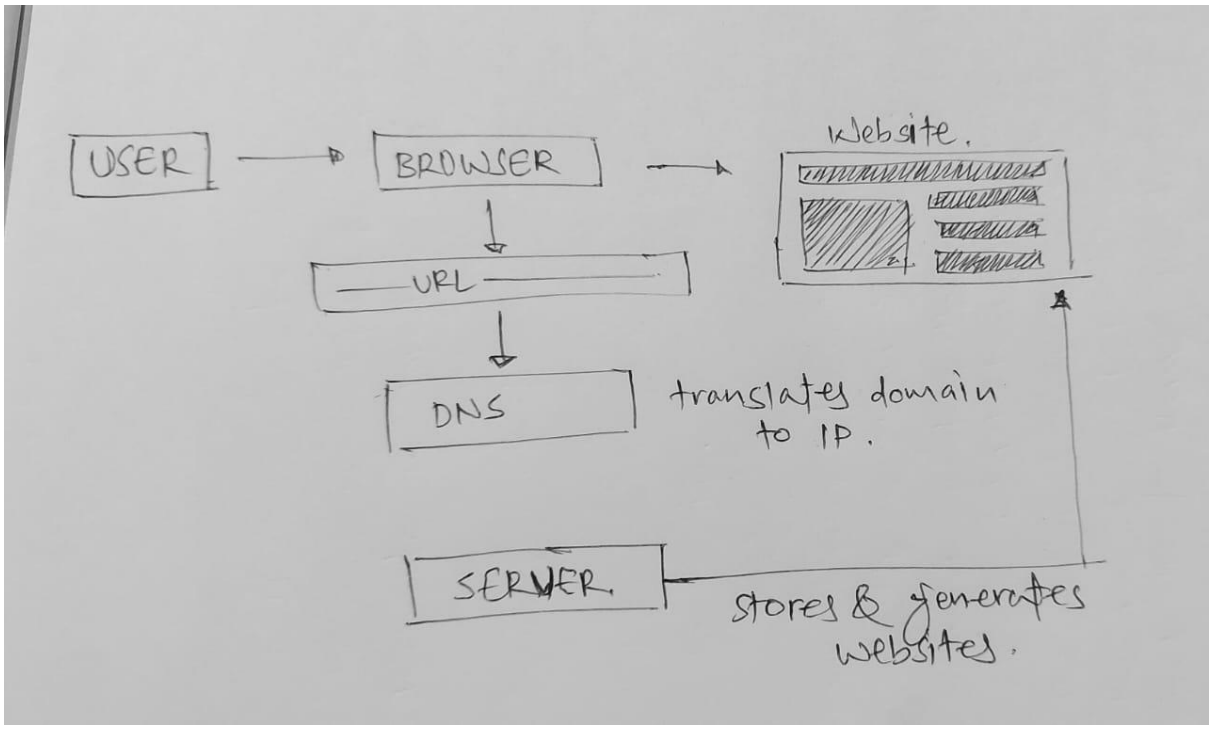
WEEK 1 : Assignment

Exercise1.1:1.

When a user enters an URL in the browser, how does the browser fetch the desired result ?

Ans: Browser is nothing but an interpreter between user and the server. Websites are stored on server. when the user enters an URL(uniform resourse locator) in the browser; the browser contacts DNS(domain name system) server, which is like a dictionary in which each domain is mapped to an IP. DNS server translates domain to IP. This IP is again given to browser. Then the browser makes the request to the server which has this IP/ domain. Then the sourse code of that IP i.e. data stored on that IP is send back to the browser. That perticular data is then displayed on screen.



What is the main functionality of the browser?

Browser's main function is to fetch the web pages from the server along with the necessary files like, images, flashes, videos etc, interpret them and then display it on the screen.

High Level Components of a browser ?

The user interface : This includes the address bar, back/forward button, bookmarking menu, etc. Every part of the browser display except the window where you see the requested page.

The browser engine : marshals actions between the UI and the rendering engine.

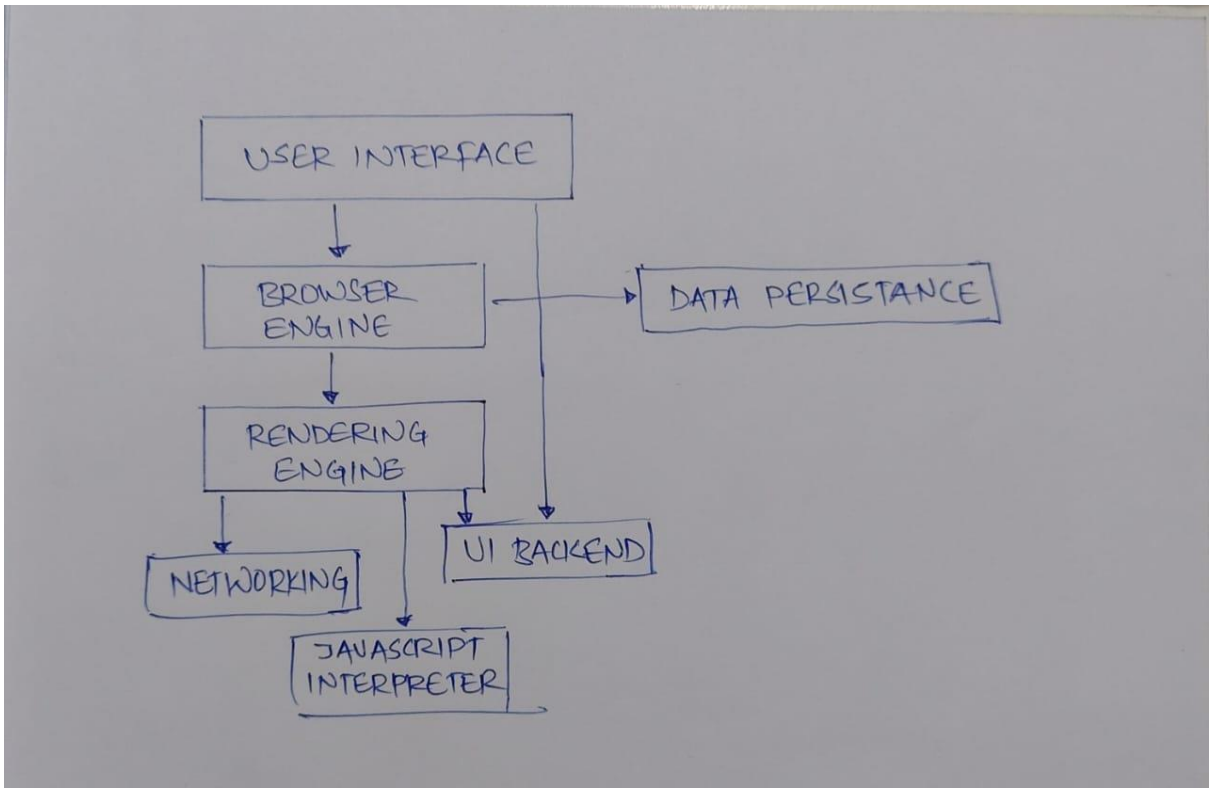
The rendering engine : responsible for displaying requested content. For example if the requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen.

Networking: For network calls such as HTTP requests, using different implementations for different platform behind a platform-independent interface.

UI backend: Used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. Underneath it uses operating system user interface methods.

JavaScript interpreter : Used to parse and execute JavaScript code.

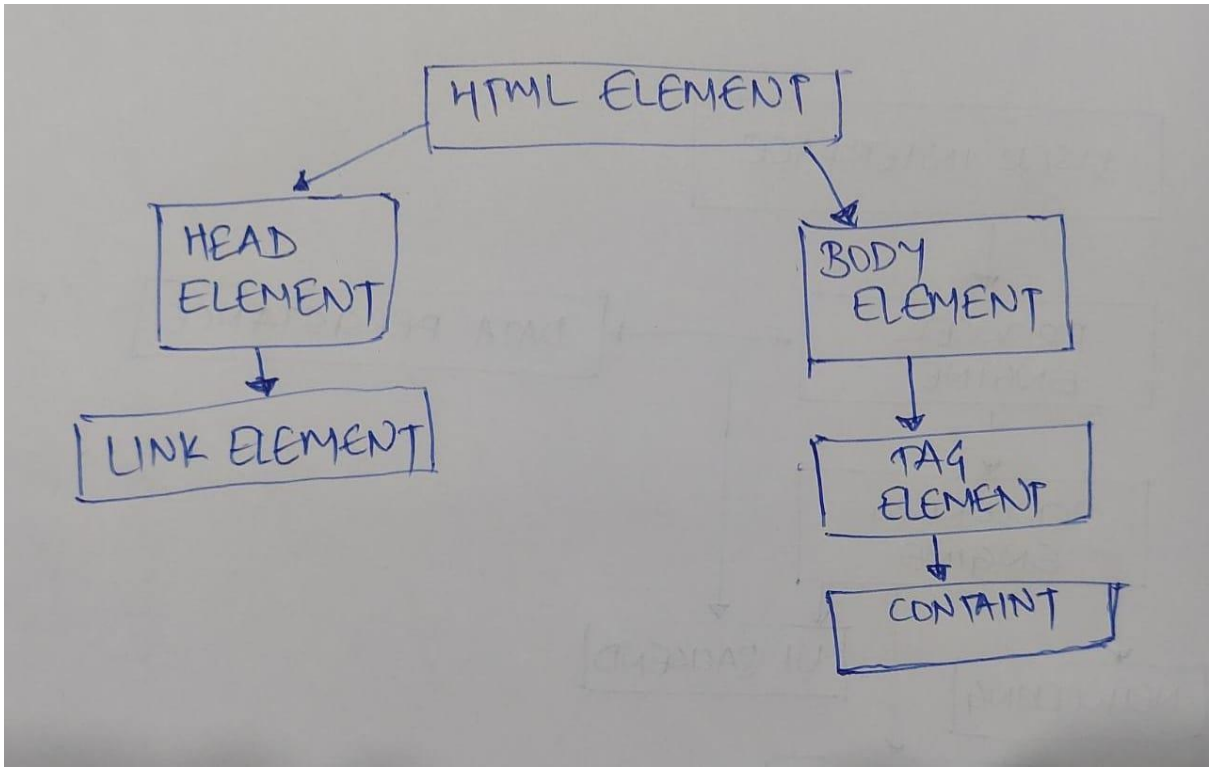
Data storage : This is a persistence layer. The browser may need to save all sorts of data locally, such as cookies. Browsers also support storage mechanisms such as local Storage, Indexed DB, Web SQL and File System.



Parsers (HTML, CSS, etc) ?

HTML, CSS content at the beginning which goes through a process called tokenization, tokenization is a common process in almost every programming language where code is split into several tokens which are easier to understand while parsing. This is where the HTML's parser understands which is the start and which is the end of the tag, which tag it is and what is inside the tag.

html tag starts at the top and then the head tag starts before the html ends so we can figure out that the head is inside html and create a tree out of it. Thus we then get something called a parse tree which eventually becomes a DOM tree.



Script Processors

The **script** processor executes Javascript code to process an event.

Tree construction

HTML describes the contain and CSS applies the styling part. So, in the process of tree construction, The CSSOM and DOM are combined into a rendering tree, which is used to compute the layout of Visible element and serve as an input to the paint process that renders the pixels on the screen.

Order of script processing

Script tags are executed in the order they appear. Means script which appear later on the page can depend on things scripts which appear earlier.

Layout and Painting

Layout is the decider to compute the position and dimentions of each node on the screen. We have to specify the screen dimentions in the meta tag or else the browser takes 980px as default dimentions of the layout screen.

Now layout provides the information about which nodes are visible and the position, geometry of nodes along with the computed styles. This is info is passed to the final stage. Which converts each node in the render tree to actual pixels on the screen. This process is known as Painting.