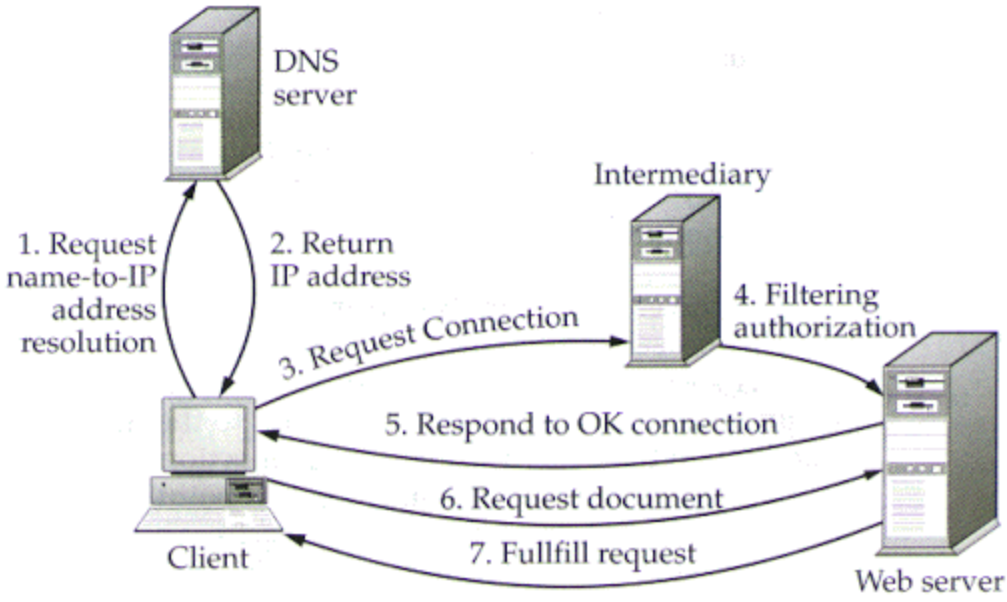
Tech Concepts

Created	@July 11, 2021 12:23 AM
Created By	Ⓛ Luke Landau
Last Edited By	Ⓛ Luke Landau
Last Edited Time	@July 11, 2021 2:09 PM
Note	
Stakeholders	
Status	
Type	

Intro to the web

- Hyper Text Transfer Protocol is the agreed way for computers to communicate with each other over the internet
- it is achieved by client side requests creating server side responses
- before the client-server process can begin, a DNS lookup is required to convert a domain name (www.google.com) into an IP address (8.8.8.8. 8.8.4.4.)



- HTTP responses contains HTML, CSS, JS, and assets (images, video, fonts)
  - in reality loading a page is broken into multiple requests, usually starting with the smallest files (HTML, CSS, JS etc.) and loading larger files once complete
  - physical distance, network activity, environmental conditions and even political stability can all have an effect on the time required to complete a request-response cycle

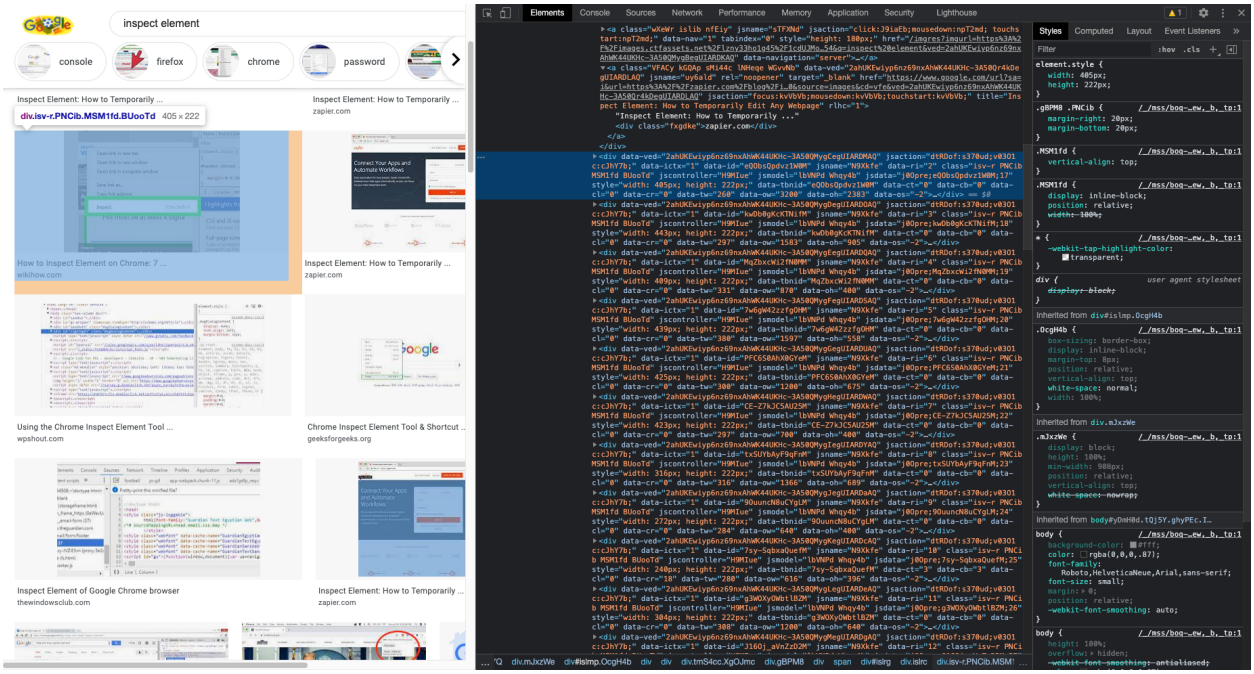
User Interface & Experience

- The User Interface (UI) is the sections of code with which a user interacts with
  - this can be a webpage
  - a desktop program like Notion
  - a command line (matrix) terminal
  - or the controls of a microwave or jumpjet
- The User eXperience (UX) is the entire experience of the user when interacting with a bit of software and can often be ruined by a singular buggy UI. It is influenced by;

- the efficiency of the system (how well it works, how slow it might be)
- the feelings that user experiences while using the system
- the customer services
- everything and everything about the user's relationship with your system

Inspect Elements / CSS

- right click on a page and select **Inspect** to see the code of a webpage
- allows changing of HTML and CSS in browser, but modifications will be lost if the page is refreshed



Separation of Concerns (SoC):

- Separation of Concerns is applied to avoid tight coupling (muddying of different languages / information / functionality to the extent a change of one breaks the others)

- it avoids code duplication, saving time, energy and money for all parties involved
- best practice is to link external files rather than write mixed files of different types of code
- broadly achieved through encapsulation

Debugging and Cross-browser Issues

- browser compatibility is a nightmare - there are so many different devices and OS's
- modern day standards are to use **evergreen** browsers - automatically updated
  - Chrome, Firefox, Edge and Safari are examples of evergreen browsers
- many dev teams only support evergreen browsers as they dominate the market
- CanIUse is a website for determining whether or not a browser will support a particular feature
  - if ever using a super custom or recently released feature, check that all browsers your dev team supports also support this feature
- BrowserStack is software that offers virtual machines to simulate the output of different devices, operating systems and browsers
- extremely expensive, most companies have a single account - can get free 1 from Git

Rendering Engines

- A rendering engine is software that draws text and images on the screen.
- the engine draws structured text from a document (often HTML), and formats it properly based on the given style declarations (often given in CSS).
- Examples of layout engines: Blink, Gecko, EdgeHTML, WebKit.

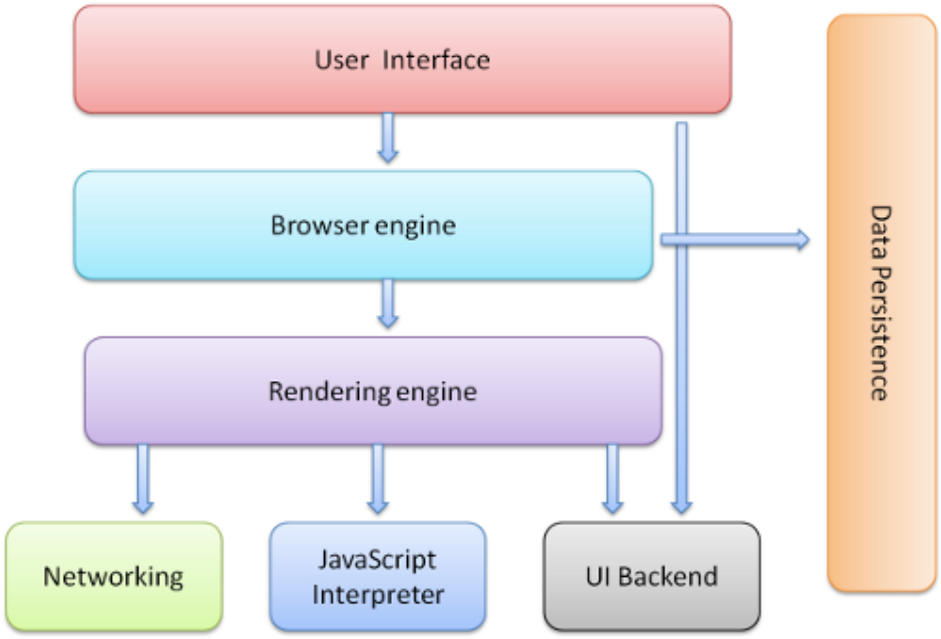


Figure : Browser components

- different browsers have different rendering engines, so they may display code differently
  - Safari - webkit
  - Chrome - Blink (based on webkit but modified)
  - Opera & Edge - Chromium (open source Chrome project)
  - Firefox - Gecko
- vendor prefixes in CSS allow a dev to target styling toward different rendering engines
  - `-webkit-` `-moz-` `-o-` `-ms-`
- writing this code yourself can be very long, so programs like Autoprefixer exist