Technical writing assignment: "How the web functions"
Amandine Velamala

When we hit https://www.techtonicgroup.com/ what happens? Don't focus too much on architecture (Monolithic, SOA, Microservices, etc.). Try to focus more on how the web functions.

When we type the url of the webpage (www.techtonicgroup.com) and press enter, the browser first consults a domain name server to translate the requested page's domain name into an IP address. The browser then sends an HTTP request to the server associated with that IP address. That server knows how to service the request. The server looks for the requested documents and sends them back to the browser that made the request. This data (code files: HTML code, CSS code, Java Script code as well as the images and videos) is sent back to the client browser through an HTTP response.

From start to finish how that data reach you to be rendered in the browser?

Once the sever receives the request, it sends back to the browser all the files needed to satisfy the request using the TCP/IP protocol. Once the data (text files containing HTML, CSS, JavaScript etc) from the server reaches the browser, the browser's rendering engine parses through the HTML code and create a DOM tree (Document Object Model). It then parses through the CSS code (style) and creates a CSSOM tree. It then creates a rendering tree, a visual representation of the web page. Finally it displays that visual representation of the page content in the browser window.

What code is rendered in the browser?

HTML and CSS are rendered in the browser. The browser applies the CSS styling to the HTML elements and display the visual content in the browser.

What is the server-side code's main function?

The server-side code's main function is to handle requests from the browser. This code runs on the server, not in the browser. It responds to the request and choose which content needs to be returned. In order to do that, the server-side code needs to process what is asked in the request. The server-side code allows to create a more tailored and customized experience. Some sites are dynamic, the content that is sent back is different from user to user. Depending on the information provided by the user, user preferences, browsing history, etc..., the content will be different from user to user, the server-side code handles these factors and responds accordingly.

What is the client-side code's main function?

The client-side code handles how the data is displayed in the browser. It's written in HTML, CSS and JavaScript and is run in the browser. Client side programming is supposed to handle how things are displayed in a specific browser, it can also handle issues of browser compatibility.
Client-side scripts are embedded in the webpage and are processed in the browser.
JavaScript is a language that can be used as a client-side code. It runs on your browser and can provide interactive components such as form validation, user input validation, etc...

How many instances of the client-side assets (HTML, CSS, JS, Images, etc.) are created?

I think one instance of the client-side assets is created. When the HTML, CSS, JS code runs, it creates one instance of this code.

How many instances of the server-side code are available at any given time?

The server-side code is not available directly from the browser. The sever-side code is available only on the server. The browser does not need to know how the server-side code functions and does not need to access the code. The server-side code runs on the server when an HTTP request is processed.

What is runtime?

1- Runtime is the time during which a program is running. A runtime error is an error that happened during the execution of a program.

2- Runtime system specifies how a program will execute. Every browser has a JS runtime environment, that allows the JS scripts to execute.

How many instances of the the databases connected to the server application are created?

I think it depends on the server side application logic. Depending on the need, the application could have multiple instances of databases.