Ursula Moreno-VanderLaan
Techtonic Academy Pre-Work
September Session

● 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.

I'll be addressing through this question from the perspective of someone using a standard browser.

When a user attempts to access Techtonic's website through an url, the first thing that happens is the ISP does a DNS lookup. Servers are just another machine hooked up to the internet, and in order to find the appropriate server, we'll need to know the appropriate IP address, just as the server will need to know ours to return information. Once the server has been identified by its IP, our ISP forwards our request to view the site. Then, the server returns an appropriate response. This response will include the index file for the site, and the content type so that our browser can render the site appropriately.

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

What this looks like from the perspective of the user varies based on what documents the server returns. In Techtonic's case, this includes an index.html file, several CSS stylesheets, some custom font packs, and various Javascript libraries and files which enable the various animations, styling options, and click triggered functions. So, when we initially receive index.html from the server, our browser begins to read the html document and render the code. In general, the document will be read from top to bottom, pausing in order to send further requests for assets along the way. Some of these assets, namely CSS and Javascript libraries will be requested in the <head> portion of the index file. This is the case of Techtonic's website, which loads several files including the Bootstrap library, jQuery, Google Analytics and content management tools among others. Each of these assets requires a new request sent to the server, and the server returns each file with its content type just as it did with the index.html document. Some of those files also have assets of their own which require further server requests.

Further requests can be made based on user actions by using Javascript. Javascript can make AJAX requests to the server based on things like user clicks or form entries, and then render appropriate responses within the page without having to refresh the entire site. In general, when the user wants more information from the site, they are sending a GET request. When they are providing new data to the website, they are sending a POST request. When they are editing existing data they are sending a PUT or PATCH request. When they are destroying data, they are sending a DELETE request. A well designed website will be choosy about what data it allows a user to access, and how users are allowed to manipulate it.

● What code is rendered in the browser?

In the case of Techtonic's website, HTML code is rendered in the browser. The various elements are checked against CSS documents, and then scripts are parsed at the end of the html document to add functionality to specific elements as needed. The CSS and Javascript affect how the elements are styled and react to our input, but ultimately it is the HTML that is rendered according to a set of web-standard hierarchies.

Ursula Moreno-VanderLaan
Techtonic Academy Pre-Work
September Session

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

Server side code mostly makes sure that users are receiving the appropriate individualized content. For example, it checks to see that I am logged into an account before it shows me the data associated with that account. If I am a valid user, it will return a successful response and the resource I requested. If I am not, it will return a failed response and no additional data. This can be something as simple as allowing me access to an otherwise static site that requires a login, and as complicated as a dynamic website like Facebook or Gmail which serves the user data that is particular to their identity. Server-side code is generally concerned with the data a user receives.

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

Client side code focuses on optimizing the UI/UX of rendered sites. CSS is a prime example of a client side coding language which is meant to simplify and beautify a user's experience on a given site. Client-side code is generally concerned with how a user interacts with the data received from a server.

For the following underlined questions I'm making a few assumptions about what you're asking. I asked some clarifying questions and would be happy to update my answers if I gain more information about what some of these questions mean.

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

Each asset should only be loaded once for the page. Some of the scripts may require that additional assets be requested once a user interacts with them, but by design that shouldn't impact the initial rendering of the page.

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

If you mean versioning, there should only be one version of the site available on the server at a given time. Sometimes sites with a lot of user generated content may need to migrate and then two or more different versions may be available on the server, but only one will be returned to a client as the service updates across all users. A recent example is Android's recent rollout of web connected texting.

If you mean how many users can access the site at any given time without the servers being overburdened, that entirely depends on the infrastructure of the site, and I can't give even an approximate answer without knowing more about how Techtonic hosts their site.

● **What is runtime?**

In the case of a website, it is the time it takes for a page to be fully rendered in a browser. Techtonic's webpage took 8.67 seconds to finish loading all of its assets.

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

My browser made 198 requests to the site's server. If you're asking how many times Techtonic connected to an external database to receive an asset then the site requested then a large portion of the JS requests were made to a Google or Twitter API, most of the fonts were loaded from Google, and your images are largely self hosted on the US West 2 region's Amazon AWS S3 server.