**Frontend vs Backend**

What are the differences between frontend and backend developers

Frontend and backend are terms commonly used in our industry.

**Frontend developers** typically concentrate on how a website *looks*. Don’t let this fool you – frontend is very complicated and very important. Frontend developers can create static sites without any backend code (by a *static* I just mean a site that does not require a database or any complicated logic. As an example, think about a law firm. Its likely their site will be very static and won’t require database calls or user memberships. Here, a frontend developer can handle the entire site).

**Backend developers** concentrates on how a website *works*. Backend developers deal with a lot more business logic, and have to decide where the data is stored (like user information), how to secure the data (encryption), how to scale a site for more visitors, whether to add new functionality to a site (like posting comments) … the list goes on.

You need to understand site rendering

To better understand what frontend and backend means, you need to understand the concept of **site rendering.**

Site rendering just means generating HTML output.

Now to blow your mind … ready? Site rendering can happen on both server-side **and** client-side level.

Let me explain both quickly.

**Server-side rendering**

Server-side rendering (or backend web development) used to be the only way to create websites and web applications. You visit a page, send a request for content, the server processes this request and creates a response that is sent back to your browser.

This means that all the processes required to create an HTML page are handled on a remote server hosting the web application. This includes querying databases for information and processing any logic that your web application requires.

What is the downside to this? Well, while the remote server is busy at work, your web browser is doing nothing. It's being lazy. It has to wait for the web server to finish processing the request and sending a response. When the response is finally received, the web browser will only then interpret it and display the content on the screen.

This is server side rendering.

**Client-side rendering**

Recently, a new form of site rendering has been implemented and this is called client-side rendering or frontend development. Here, the rendering of the content happens on your computer instead of the remote web server. Your browser does this using JavaScript.

What does this mean for the server?

It means that the server is only needed to serve the raw web application, and from then on the browser will be in charge of rendering the application in its final form (which as you know is HTML). This means that most of the logic involved in creating the web page, especially the one in charge of dealing with how things are presented to a user, are handled on the client-side.

Clientside rendering became popular with the rise of JavaScript libraries such as **Angular**, **React**and **Vue**.

What languages does a frontend developer and backend developer use

**Frontend developers**primarily use three languages: HTML, CSS and JavaScript. But as we have just discussed above, the role of the frontend developer has evolved, and so professional frontend devs now have to learn more intermediate to advanced JavaScript skills using command-line tools and a framework such as React and Angular.

**Backend developers** need to be proficient in server programming languages. The most popular backend programming languages are PHP, Ruby, Python, Node.js, and Java. A backend developer should be a master at one of these languages, but should also know enough to dabble into a 2nd or 3rd language.

Backend developers will also need to proficient in working with databases (because a large part of your work will be dealing with handling data). Some popular databases are MySQL, Oracle, mongoDB and SQL Server.

JavaScript lives in different environments

JavaScript has historically been a programming language that ran only inside a web browser. But as we've discussed, this has drastically changed now. As I mentioned in the previous video, you can now use JavaScript to develop frontend code, make database queries as well as writing server side code.

What does this mean?

It means that instead of a developer having to learn JavaScript for the browser, PHP for the server and MySQL for database queries (as an example), you can concentrate on using JavaScript all around. For example, you can now use JavaScript to build the frontend, make server calls (with Node.js), and database queries (with MongoDB).

And this is why JavaScript is so powerful.

Now don’t get me wrong.

Just because you are writing JavaScript in 3 environments doesn’t mean its easy. Each environment still has their own unique functions, keywords and nuances. But at least the underlying syntax of your programming language (in this case JavaScript) stays the same.

Hope this has provided some clarity.

Keep going and I can't wait to see you in the next section.