How to Build Stuff:

I was going to use another word for stuff, but I decided to keep it PG for this assignment.
Specifically, I want to talk about the steps to building a scalable and fast website.

Building a scalable and fast website is essential for startups to F500s looking to reach a larger audience and improve customer satisfaction. A website that can handle a high volume of traffic and load quickly will provide a better user experience and increase the likelihood of conversions.

Here are some steps to follow in order to build a scalable and fast website:

Choose the right hosting provider: Your website's hosting provider plays a crucial role in its scalability and speed. A reliable hosting provider will have servers that are optimized for speed and can handle a large amount of traffic. It's important to do your research and choose a hosting provider that has a good reputation and offers a variety of hosting options to suit your business needs.

Optimize your website's code: The code of your website can impact its speed and scalability. To optimize your website's code, it's important to use clean, organized, and well-structured code that is free of errors and unnecessary elements. This will help your website load faster and be more scalable. Specifically, minify and concatenate your CSS and JavaScript files to reduce their size and the number of HTTP requests made to the server. One can also use a CSS preprocessor, such as Sass or Less, to write modular and maintainable stylesheets.
When deploying the website, use a code-splitting tool, such as Webpack or Rollup, to split your code into smaller chunks that can be loaded on demand. Furthermore, developers should implement lazy loading, which delays the loading of below-the-fold content until the user scrolls down the page.

Use a content delivery network (CDN): A CDN is a network of servers that are distributed across different geographic locations. By using a CDN, your website's content will be delivered to users from the nearest server, which will improve its speed and scalability. This is especially useful for businesses with a global audience, as it will provide a faster and more consistent user experience. Speaking about the server side, using/implementing a load balancer to distribute incoming traffic across multiple servers allows the website to handle more concurrent users. Developers can consider using a cloud-based solution, such as Amazon Web Services (AWS) or Google Cloud Platform (GCP), which allows you to easily and quickly scale your website's resources as needed.

Use caching: Caching is the process of storing frequently accessed data in temporary storage so that it can be accessed quickly. By using caching, your website will be able to handle a high volume of traffic without slowing down or crashing. There are various caching options available, such as browser caching, server-side caching, and application caching, and it's important to choose the right one for your website's needs.

Protect it: In other words, actually, pay attention when GitHub issues Security Advisories inside your codebase. Here's an example of one that I have been ignoring: "Prototype pollution in webpack loader-utils" (I have no idea what that means). Furthermore, keep your website's software and plugins up to date, as these updates often include security fixes.