Next.js Grouping and Rendering Techniques

# Grouping in Next.js

Grouping in Next.js allows you to create separate `layout.tsx` files for each group, which enhances maintainability and structure. This way, each group has its own layout, making it easy to manage different sections of your application.

However, it's important to note:

- You cannot create folders with the same name within a group (e.g., in the `dashboards` group).
- Group names will not appear in the URL as Next.js automatically ignores them.

# SEO and Web Basics

Search Engine Optimization (SEO) is essential for making your website visible on search engines like Google. Here's a breakdown of how websites and SEO work:

- Domain: A website's domain is its unique address, like `.com`.
- Server: Your website is stored on a server. When a request is made, the server delivers the necessary HTML, CSS, and JavaScript files to the browser.
- Browser: The browser (also known as the client) is where the website is displayed. It combines the files and renders the website for the user.
- Bots: Search engine bots crawl websites and rank them, deciding whether they appear at the top or lower in search results.

# Rendering Techniques in Next.js

Next.js offers different rendering methods to improve the user experience and performance of websites.

## Client-Side Rendering (CSR)

Client-Side Rendering involves generating HTML on the browser using JavaScript. Once the HTML is ready, the browser displays the website. However, this can slow down the site and negatively impact SEO.

## Server-Side Rendering (SSR)

In Server-Side Rendering, Next.js prepares the HTML on the server and sends it to the browser, improving performance. First, the HTML loads, and then JavaScript is added, which enhances the speed of the initial page load.

# Static vs Dynamic SSR

## Static SSR

Static SSR involves pre-building HTML on the server, which provides better performance. This method is best for websites that don't change frequently, like portfolios and blogs.

## Dynamic SSR

Dynamic SSR creates HTML at the time of the request for each user. While useful for websites with frequently updated content, like e-commerce, this approach can slow down performance due to the generation of unique pages for each request.