بسم الله الرحمن الرحیم

**Nextjs**

**What is Nextjs ?**

**it is a framework that build on top of the react that gives you flexibility of building scalable apps by allowing you to render content on the server.**

در ری اکت همانطور که می دانیم تمامی محتوای برنامه سمت client ، رندر می شود اما درNext میتوانیم سمت سرور رندر کنیم.

**مزایای Next js :**

1. حالت‌های مختلف رندر شدن :

* Static site generation : سمت client
* Server side rendering (SSR) : سمت سرور
* Incremental site Regeneration : ترکیبی از کلاینت و سرور.

2 – Performance :

* Code Spliting : برنامه ی ما را به chunk های کوچکتری تقسیم می کند پس ما فقط chunk های مورد نظرمان را Load می کنیم.
* Minifying Files: در هنگام لود شدن حجم فایل ها را کم می کند که در ری اکت به سادگی قابل انجام شدن نیست.
* Image Optimizing: نکست یک کامپوننتی دارد به نام Next Image که این برای رندر کردن تصاویر است که به صورت ریسپانسیو عمل می کند.
* Pre-Fetching assets: در هنگام اسکرول کردن به پایین اطلاعات شروع به fetch کردن می شود و به صورت یکجا fetch نمی شود.

3 – File Based Routing : Routing در نکست بر اساس فایل ها خواهد بود برخلاف ری اکت که باید از ابزار React-router-dom استفاده می کردیم.

4 – ابزار های بسیار خوب برای SEO که باعث بالا امدن رنک صفحات در گوگل است.

5 – Serverless functoins :‌فایل های داخل دایرکتوری api ، بدون نیاز به هیچ کانفیگی در بک اند اجرا می شود. فرض کنیم که یه فایل hello.js در داخل این دایرکتوری موجود است زمانی که سرور شروع به کار می کند این فایل را هم اجرا می کند و زمانی که اجرایش تمام شود دوباره سرور shut down می کند. پس می توانیم بدون اینکه دانشی از بک اند داشته باشیم کدی را بنویسیم که با اجرای سرور اجرا شود. به این معنا هم است که نیازی نیست که سرور همیشه روشن باشد هر وقت خودش بخواهد اجرا شود سرور را روشن می کند و اجرا می شود.

با دستور npx-create-next-app یک پروژه ی next ساخته می شود و با دستور npm run dev این پروپه اجرا می شود.

در داخل فولدر pages اگر هر فایلی را قرار بدهیم به صورت خودکار برایش یک route درست می‌شود.

فایل ابتدایی که با بالا امدن سرور ان رندر می شود فایل index.js است.

اگر کامپوننتی داشته باشیم که بخواهیم در تمامی صفخات رندر شود باید ان را در فایل \_app.js قراردهیم . مثل فوتر و هدر

Global.css : کانفیگ های کلی که روی کل فایل های css و پروژه اعمال می شود و میدانیم که فایل های css باید به صورت ماژولار باشد.

نکست یک قابلیتی دارد به نام fast refresh که با کوچکترین تغییر در فایل های کد سریع سایت اپدیت می شود.

یک نکته ی css : میتوانیم در css هم یک فیچر را به صورت var مقدار دهی کنیم به این صورت:

Color : var(--textColor). We should use – before name.

مقدار دهی این متغیر ها را هم می توانیم در جایی بگذاریم که در همه جا اعمال شود که دقیقا فایل app.js است. در این فایل فایل css ای را import کردیم به نام global.css پس در این فایل global.css این متغیر ها را مقدار دهی می کنیم.

نکست یک تگ Head دارد که در ان می توانیم به صورت مستقیم title صفحه را عوض کنیم و سختی های javascript و react را ندارد و دیگر نیازی نیست که به فایل index.html برویم و انجا را تغییر دهیم. پس این تگ <Head></Head> دقیقا معادل همان تگ های head در index.html است که کار ما را راحت تر کرده است.

در نکست Routing به سه دسته تقسیم می شود. 1- index routes 2 – nested Routes 3 – dynamic Routes .

IndexRoot : دقیقا همین Routing ای که با قرار دادن فایل ها در پوشه pages اعمال می شود. چون با اجرا شدن سرور در ابتدا فایل index.js بالا می اید به این روش index routing می گویند که نام default route هم دارد.

Nested Route : این هم شبیه بالایی است. تفاوتش را متوجه نشدم !!!!!

Dunamic route : Route هایی که با براکت معرفی می شوند که مثلا داخل براکت product\_id قرار می دهند

هر فایلی که در دایرکتوری pages می گذاریم برای تبدیل شدن به Route باید دو شرط را ارضا کند.

1 – Page Needs to be a React Component

2 – Component needs to be exported by default.

برای اینکه Routing ما به صورت index انجام شود حتما باید به اسم فایل توجه داشته باشیم چرا که نام فایل دقیقا به عنوان pathname در location ما حضور خواهد داشت. مثلا اگر اسم فایل باشد MohammadReza ان وقت با اجرا کردن پروژه به این صورت صفحه MohammadReza را می بینیم.

Localhost:3000/MohammadReza

یعنی دقیقا نام فایل را به عنوان pathName داریم.

Dynamic Routing : به این صورت:

Localhost:3000/stores/[id]

نکته مهمممم:

اگر در داخل pages یک پوشه دیگری قرار دادیم برای اینکه فایل js داخل این پوشه اجرا شود باید default Route باشد یعنی اینکه مثلا اگر اسم فایل store.js است باید ان را به index.js تبدیل کنیم چون که index.js ، default Route خواهد بود.

Dynamic Routing : اگر داخل pages یک دایرکتوری جدید با هر اسمی مثلا coffee-stores بسازیم و داخل این دایرکتوری از داینامیک روتینگ استفاده کنیم (منظور از داینامیک روتینگ این است که در داخل ان دایرکتوری یک فایل js بسازیم که اسم آن به صورت [name] باشد) روند path سایت به این صورت خواهد شد.

Localhost:3000/coffee-stores/Golestan

یعنی اسم دایرکتوری به عنوان ثابت کار می اید و name می تواند هر اسمی باشد.

حالا اگر بخواهم name موجود در url را در دست داشته باشم و استفاده کنم باید از hook استفاده کنم. Next js یک hook دارد به نام useRouter که این از next/router ، import می شود.

Import {useRouter} from “next/router”

Const Router = useRouter();

Console.log(Router);

Localhost:3000/coffee-store/bestCoffee

یک ابجکتی را برمی گرداند که در فیلد query یک ابجکت دیگری داریم که به صورت زیر است

query:{id : “bestCoffee”}

که کلمه ی bestCoffee دقیقا url ما خواهد بود و id دقیقا نام فایل جاوا اسکریپت ما خواهد بود و اگر نام فایل را تغییر دهیم این هم قطعا تغییر می کند (چون اسم فایل [id].js بود.)

در HTML برای رفتن به صفحه ای از تگ a استفاده می کردیم که این کار باعث می شد که زمانی که روی آن دکمه کلیک کنیم کلا صفحه رفرش می شد اما در اینجا همانند ری اکت می توانیم از Link استفاده کنیم با این تفاوت که اولا از next/link باید import شود ثانیا دیگر to:{pathname:”path”}} ندارد بلکه href دارد.

از تگ لینک تنها برای جابه جا شدن در داخل خود Application استفاده می کنیم یعنی مثلا وسط کار خواستیم بریم به سایت مثلا ورزش 3 لزومی نداره که از تگ link استفاده کنیم و از a استفاده کنیم بهتره.

Next/link ، property های بینظیر دیگه ای هم داره که خیلییی بدرد می خوره مثلا prefetch و scroll و ...

برای استفاده از عکس حتما باید از کامپوونت Image داخل خود next استفاده کرد.

Import Image from “next/image”

که این تگ Image ، سورس داره که این سورس به صورت پیش فرض از public است یعنی اگر فایل های عکس داخل public باشد دیگه در داخل src نیازی نیست بزنیم src = “public/…..” یعنی فرض بر اینه که توی public هست