Mpa:(multiple page app): با تغییر در path یکبار صفحه به طور کامل ریلود میشود.از نظر پرفورمنس ضعیف هستند.

Spa(single page app): کل صفخه یکبار لود میشود و با تغییر path فقط محتوای صفحه بدون ریلود شدن عوض میشود.

نکته: به هر مسیر یک route گفته میشود و برای مدیریت این ها از react router استفاده میکنیم.

React router: نصب npm install react-router-dom

از Routes برای تعریف روت ها و برای هر روت از Route استفاده میکنیم که به صورت prop برای آن مقادیر path برای تعیین کردن مسیر مثلا /users و element برای مشخص کردن چیزی که قرار است نمایش داده شود که معمولا یک کامپوننت است.کل آن کامپوننت که درآن Routes را تعریف کردیم باید داخل BrowserRouter قرار بدهیم.

Link: لینک هایی که با تگ a میسازیم باعث ریلود شدن صفحه میشوند ولی برای spa بودن سایت نباید این اتفاق بیوفتد پس از Link که از reat-routert-dom ایمپورت میشود استفاده میکنیم.به عنوان prop با to مسیر را مشخص میکنیم.با children المنت یا متنی که قرار است داخل لینک نمایش بدهیم مشخصی میکنیم.با state میتوانیم یک مقدار را به صفحه بعدی که قرار است باز شود منتقل کنیم.

NavLink: تمام موارد بالا را دارد علاوه بر آن یک prop به نام end دارد که مشخص میکند الان داخل مسیر همان لینک هستیم برای استایل دهی لینک فعال استفاده میشود.در className به یک مقدار به نام isActive دسترسی داریم که به صورت شرط میتوانیم استایل مد نظرمان را مشخص کنیم.

className={isActive ? active : deactive }

useParams: در تعریف مسیر میتوانیم با : یک پارامتر تعریف کنیم مثلا => /posts/:id برای استفاده از این پارامتر در کامپوننت مقصد از هوک useParams استفاده میکنیم که یک آبجکت از تمام پارامتر ها به ما میدهد.

Navigate: از این کامپوننت تگ برای redirect کردن کاربر استفاده میکنیم که بدون ریلود شدن این کار را میکند و به to مسیر را مشخص میکنیم.

404: یک Route با مقدار \* اضافه میکنیم و اگر کاربر وارد هر مسیر دیگری که در روت های پروژه نباشد وارد آن کامپوننت میشود که میتوانیم بسازیم.

subRoutes: اگر مسیرمان به این باشد مثلا /users/profile به قسمت پروفایل یک subRoute از users میگویند.به این صورت که ابتدا کامپونتت users رندر میشود و پس از آن کامپوننت مربوط به profile رندر میشود.برای این کار ابتدا در Routes اصلی یک مسیر به صورت /users/\* مینویسیم که برای هر ساب روتی کامپوننت user اجرا بشود سپس داخل خود کامپوننت users یک Routes جدا مینویسیم و Route ها را مشخص میکنیم. برای نوشتن path در این حالت نباشد در ابتدا / بذاریم چون از root پروژه حساب میشود.

Outlet: در روش قبلی نیاز هست در جاهای مختلف پروژه Routes تعریف کنیم که از خوانایی پروژه می کاهد.به جای آن میتوانیم در Routes اصلی پروژه از تگ Route به صورت باز و بسته استفاده کنیم و subRoutes را داخل آن بنویسیم. در این حالت نباید از \* در Route اصلی استفاده کنیم و ساب روت ها را باید بدون / در ابتدا بنویسیم.در جایی که میخواهیم sub رندر بشه باید از Outlet استفاده کنیم.

useRoutes: باید از react-router-dom ایمپورت بشود و به عنوان ورودی یک لیست از آبجکت ها میگیرد که دو مقدار path ,element مانند قبل باید برایش ارسال کنیم subRoute ها را هم با کلید children به صورت یک آرایه مینویسیم.در jsx هم هر جا که میخواستیم رندر شدن روتر ها انجام شود از متغیری که ساختیم استفاده میکنیم با نام دلخواه router.برای خوانایی بیشتر تمام route ها را داخل فایلی به نام routes.js در src مینویسیم.