ساختار پروژه react :

Node\_modules : تمام وابستگی های پروژه مثل پکیج های نصب شده در این دایرکتوری قرار دارند.

Public : تمام asset های پروژه مانند تصاویر, ویدیو ها و فایل های موزیک که در صفحات نیاز است در این دایرکتوری قرار دارند.

Src : شامل فایل های js css هست که ساختار اصلی صفحات پروژه در آن قرار دارد و بیشتر با این قسمت سرو کار داریم.

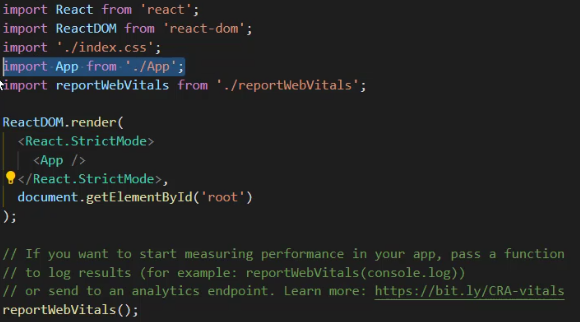
Package.json: اطلاعات کلی مربوط به پروژه در این فایل json قرار دارد.

Package-lock.json: در فایل قبلی وابستگی های پروژه مشخص شده است و هر کدام از آن ها هم میتواند وابستگی های خودش را داشته باشد در این فایل

تمام وابستگی ها با جزییات کامل نوشته میشود.

Component: یک تابع است که یک ساختار jsx را return میکند.برای هر بخش از پروژه یک کامپوننت میسازیم و هر جایی که خواستیم ان را ایمپورت میکنیم و استفاده میکنیم.برای ایمپرت کردن نیازی به نوشتن .js در آخر نام فایل کامپوننت نیست. در هر کامپوننت میتوانیم به صورت تو در تو از کاپوننت های دیگر استفاده کنیم.

ReactDOM : از این پکیج برای رندر کردن کامپوننت استفاده میکنیم به صورتی که ابتدا همین پکیج را ایمپورت میکنیم و سپس از متود .render() استفاده میکنیم.

کامپوننت اپ را به المنتی با ایدی روت اضافه میکند.

نکته : زمانی که بخواهیم چیزی را از پکیج های نصب شده در پروژه ایمپورت کنیم نیازی به نوشتن ./ در ابتدا نیست.

روش استفاده از کامپوننت: مثل تگ های html به صورت باز و بسته و یا به صورت تگ تکی.

React.StrictMode: حالت strict در جاوااسکریپت که کمی سختگیرانه تر کد ها را اجرا میکند را فعال میکند.

Export default: برای اینکه ماژول یک کامپوننت را به صورت پبش فرض اکسپورت کند باید در انتهای کد با این دستور اکسپورت کنیم همچنین میتوانیم در هنگام تعریف تابع قبل از کلمه function بنویسیم

استفاده از css: یک فایل css میسازیم و استایل دهی ها را انجام میدهیم و سپس در داخل فایل کامپوننت ایمپورت میکنیم تا استایل ها در کامپوننت اعمال بشوند.برای ایمپورت نیازی به دستور from نیست و با import به تنهایی انجام میشود.

نکته: هر کامپوننت که شامل فایل js css … میشود را درون یک دایرکتوری مجزا مینویسیم تا کاملا جدا از هم باشد و ساختار پروژه بهتر باشد.

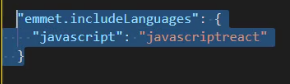
className : در ری اکت به جای class با این کلمه به المنت ها کلاس میدهیم.

نکته : در هر کامپوننت در نهایت به عنوان خروجی باید یک تگ return بشود.

Jsx: (javascript xml) یک ساختار برای ذخیره اطلاعات است که به ما اجازه میدهد که تگ هایی با نام دلخواه بسازیم این ویژگی به ما این امکان را میدهد که در جاوااسکریپت مستقیم تگ هایی با نام دلخواه بنویسیم.برای نوشتن عبارت از {} استفاده میکنیم و اگر داخل آن یک عبارت مثلا 2+3 بنویسیم مقدار5 را نشان میدهد.حتی میتوانیم کد های js هم بنویسیم.

Virtual DOM: در ری اکت یک دام مجازی وجود دارد که وقتی ما تغییری در یک المنت اعمال میکنیم فقط همان بخش که تغییر کرده را در دام واقعی real DOM اعمال میکند در حالت واقعی اگر یک تغییر در المنت والد انجام بشود المنت های فرزند هم دوباره رندر میشود که در پروژه های بزرگ باعث کاهش کارایی میشود.

React snippet: این اکستنشن برای ایجاد شورت کات هایی برای کاهش حجم کد زنی است مثلا با نوشتن imr میتوانیم react را ایمپورت کنیم.

Emmet: این افزونه در فایل های html کار میکند و برای اینکه در فایل js هم بتوانیم استفاده کنیم باید در فایل settings.json کانفیگ اضافه کنیم

Class component: یک کلاس میسازیم و از React.Component ارث بری میکنیم.یک متود به نام render در آن میسازیم و کد های jsx که میخواهیم را return میکنیم.

Class vs func: بهتر است که تا وقتی مجبور نیستیم از تابع برای ساخت کامپوننت استفاده کنیم زیرا هم حجم کدی که مینویسیم کاهش میابد و همچنین وقتی کد را به فرمتی که مرورگر بفهمد تبدیل میکنیم باز هم حجم کد تبدیل شده از تابع کمتر است.

Props: وقتی یک کامپوننت میسازیم که قرار است دیتایی داینامیک رو رندر کند به تابع کامپوننت یک ورودی به نام props میدهیم و هر جا که خواستیم کاموننت را به صورت تگ استفاده کنیم به صورت tag attr موارد را مینویسیم که به هرکدام از آن ها prop گفته میشود.تمام موارد به صورت یک object به نام props در تابع در دسترس خواهد بود.

استفاده شرطی از props: داخل کامپوننت میتوانیم یک شرط بنویسیم که فقط برای ابجکت هایی با شرط خاص کد jsx برگردانده بشود. در این حالت باید برای مواردی که شرط برقرار نباشد باید مقدار null را برگردانیم.دراین حالت برای کاهش حجم کد از عملگر های && و || میتوانیم استفاده کنیم.

Props destructuring: میتوانیم مقادیر داخل props رو داخل چند متغیر بریزیم و بدون نوشتن مثلا props.name مستقیم name را بنویسیم.در ورودی کامپوننت میتوانیم دسترکاتینگ را به صورت {name, age} انجام بدیم.

Spread props: زمانی که داریم کامپوننت را به صورت تگ اجرا میکنیم میتوانیم بدون نوشتن تک تک attr ها از spread syntax استفاده کنیم.

Prop rename: اگر در کامپوننت متغیری همنام با مقداری که از spread دریافت کردیم داشته باشیم میتوانیم نام آن را در تابع تغییر بدیم. {name: UserName, age}