

به نام خدا

تمرین دوم

۱- state management چیست ؟

State management یکی از ویژگی های مهم و اجتناب ناپذیر هر برنامه پویا است . React یک API ساده و انعطاف پذیر برای پشتیبانی از state management در یک component ارائه می دهد. در واقع React یک data store داینامیک برای هر component فراهم می کند. اگر کاربر با تعامل با برنامه ، state را تغییر دهد ، ممکن است رابط کاربری بعد از آن کاملاً متفاوت بنظر برسد زیرا component با فراخوانی متد render() به همراه state جدید خود را دوباره رندر می کند.

برنامه های React، state خود را بصورت داخلی مدیریت می کنند و برای برنامه هایی با component کم به خوبی کار می کنند. اما زمانی که برنامه بزرگتر می شود پیچیدگی مدیریت state های به اشتراک گذاشته شده در بین component ها دشوار می شود و پیدا کردن اینکه وقتی مشکلی پیش می آید چه اتفاقی می افتد واقعا سخت است . به همین دلیل است که به state management نیاز داریم و برای این کار از Redux, Hooks, Recoil می توان کمک گرفت .

۲- functional components چیست ؟

Functional components از رایج ترین component هایی هستند که هنگام کار با در React با آن ها مواجه می شویم . این توابع ممکن است داده ها را بعنوان پارامتر دریافت کنند یا نکنند . در functional components مقداری که return می کنیم کد JSX است تا به درخت DOM رندر شود . نکته کلیدی که این نوع component را از class component متمایز می کند ، فقدان state و lifecycle ها است. به همین دلیل functional component بعنوان stateless components ها هم شناخته می شوند.

۱) خواندن functional component تا حد زیادی آسان تر است زیرا شما از قبل همه چیز هایی که نمی توانند انجام دهند را می دانید مثل ورودی های مخفی یا state پنهان .

۲) تست functional component آسان تر است . زیرا لازم نیست نگران state پنهان یا side effects باشید . برای هر ورودی (props) functional component دقیقا یک خروجی دارد

۳) functional component می تواند عملکرد بهتری داشته باشد . متاسفانه اینطور نیست که بدلیل نبود state و lifecycle ها از هزینه ها اضافی اجتناب شود اما حتی بدون این بهینه سازی منجر به کد کمتر باندل شدن سریع تر می شود .

۴) debug کردن functional component آسان تر است . زیرا فقط به props ای که می گیرد تا خروجی را تولید کند بستگی دارد و نیازی به ثبت مداوم state برای درک آن چه در جریان است وجود ندارد.

۵) functional component بیشتر قابل استفاده مجدد هستند.

۳- Hook چیست ؟

Hook ها توابعی هستند که شما را به React state و ویژگی های function component ها متصل می کنند. Hook ها در class ها کار نمی کنند و به شما اجازه می دهند تا از ویژگی های React بدون نوشتن class استفاده کنید.

Hook ها backwards-compatible هستند ، به این معنا که هیچ تغییر (break change) را نگه نمی دارند. React چند built-in Hook مثل useState, useEffect, useReducer دارد . هم چنین می توانید Hook های سفارشی بسازید .

منابع :

https://www.tutorialspoint.com/reactjs/reactjs_state.htm

<https://www.loginradius.com/blog/async/react-state-management>

<https://programmingwithmosh.com/react/react-functional-components>