

## نکات ضروری:

- 1- نصب کردن node در کامپیوتر
- 2- اجرا دستور npm install در root فولدر پروژه در command line
- 3- اجرا دستور npm start برای اجرا شدن پروژه

## شرح پروژه:

این پروژه یک frontend Web-base application است که responsive نیز است.

هدف از پروژه دسته بندی تراکنش ها شامل هزینه خرید ظرفیت - هزینه شارژ حساب - هزینه سفر - هزینه متفرقه در یک پیک یا تاکسی اینترنتی است که قابلیت دسته بندی و مرتب کردن تراکنش ها را دارد و قابلیت جست و جو نیز به کاربر می دهد.

## شرح فنی:

ما در این پروژه از react برای درست کردن اپلیکیشن استفاده کرده ایم و می دانیم react یک کتابخانه قدرتمند برای زبان JavaScript است که از آن برای ساخت web application ها استفاده می شود.

همچنین در این پروژه از type script به جای java script استفاده می کنیم و می دانیم type script همان java script است که به آن type اضافه شده است و زبان java script را قانونمندتر کرده است.

ما در این پروژه یک فایل data.json داریم که در پوشه پابلیک ما است که data ما در این فایل قرار دارد، در اینجا جوری رفتار می کنیم که این فایل انگار از سرور می آید ، حال ما برای نگه داشتن این data در پروژه از redux استفاده میکنیم که ما پنج action داریم که در پوشه store قرار دارد و با استفاده از آنها وضعیت دیتا ها را مشخص می کنیم و برای هر وضعیت یک منطقی را در reducer های خود پیاده سازی کرده ایم که بر اساس آن وضعیت برنامه تغییر می کند و دیتا نشان داده شده به کاربر نیز تغییر می کند ، در اینجا به پنج action می پردازیم:

- 1- FETCH\_DATA: این اکشن برای این است که دیتا را از فایل data.json بخواند.
  - 2- FETCH\_DATA\_SUCCESS: این وقتی است که store ما بتواند دیتا را به طور موفقیت آمیز از data.json بخواند.
  - 3- FETCH\_DATA\_FAILURE: این وقتی است که store ما نتواند دیتا را به طور موفقیت آمیز از data.json بخواند و ارور دهد.
  - 4- ALL\_DATA: این اکشن برای نمایش تمام داده هاست.
  - 5- FETCH\_DATA\_FILTER: برای نمایش داده ها به صورت فیلتر شده است.
- ما در پروژه پنج component داریم که اپلیکشین ما را تشکیل می دهد.

- 1- LoadingSpinner: این component به ما وقتی که صفحه در حال load کردن دیتاست به ما یک loading چرخشی نشان میدهد.
- 2- SelectBox: این component بخش select box، component ماست که از طریق آن کاربر می تواند دیتا را فیلتر کند.
- 3- TransactioContainer: این component قالب UI و اپلیکیشن ماست که دیتا را در آن می خوانیم.

4- Transaction: در این component ما از store دیتا مورد نیاز کاربر را میگیریم و اگر تراکنشی نداشت به کاربر اطلاع می دهیم.

5- TransactionCard: این component جزییات هر تراکنش را به شکل کارت به کاربر نشان می دهد.

ما برای استایل دهی در این پروژه از sass استفاده کرده ایم بدینگونه که ما یک متغیر ها و ماژول ها برای استایل دهی در پوشه style مشخص کرده ایم و با اجرای دستور sass در این پوشه خود یک فایل css برای ما می سازد که ری اکت برای استایل دهی استفاده می کند.

احال در پوشه utils یک تعدادی تابع تریف کرده ایم که هرکدام را شرح خواهیم داد:

1- ما در ابتدا یک سری متغیر ثابت داریم که از آنها برای تابع هایمان استفاده می کنیم که در فایل constants در این پوشه قرار دارند.

2- whatIsDateFieldName : با استفاده از این تابع مشخص می کنیم که فیلد تاریخ در داده ای که در data.json داریم چیست چون در بعضی موارد در فایل data.json فیلد های تاریخ با هم متفاوت است.

3- mergeTransactionsToOneArray: با استفاده از این تابع همه ی آرایه تراکنش هارا به یک آرایه تبدیل میکنیم تا داده هایمان پخش نباشد.

4- replaceEnglishToPresianNumber: با استفاده از این تابع اعداد انگلیسی را در جهت راحتی کاربر به فارسی تبدیل می کنیم.

5- sortTransActionInTime: این تابع تراکنش ها را از جدید ترین به قدیمی ترین sort می کند

6- convertMiladiToShamsi: با استفاده از این تابع تاریخ میلادی که از سرور دریافت می شود را به شمسی تبدیل می کنیم.

7- getShamsiDetailSate: این تابع جزییات تاریخ شمسی را به ما نمایش میدهد.