

مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران



React / JS

مکتب ۱۱۸





پروژه پایانی

موضوع پروژه

سایت فروشگاهی محصولات، شامل صفحات بازدید کنندگان (مشتریان) و صفحات مدیریت سایت مطابق با فایل ماکآپ ارسالی پیاده سازی شود و طی چند فاز ریلیز شود

مشخصات فنی پروژه

- یک SPA با Next js , app router برای تمام صفحات سایت و ادمین حاوی روت ها و لیاوت های لازم
- روش استایل دهی و انتخاب Library کاملاً آزاد. توصیه میشود از tailwind و لایبرری های تیلویندبیس استفاده کنید.
- استفاده از لایبرری axios جهت ارتباط با backend
- استفاده از لایبرری redux یا لایبرری استیت منیجر جهت مدیریت state
- رپوی github خود را بصورت private با فرمت maktab118-project-name-family بسازید و به اساتید و مربیان جهت بازدید دسترسی بدهید. در نظر داشته باشید که پروژه انفرادی است.
- برای هر فاز، بر مبنای اصول git flow ریلیز صورت بگیرد، باگهای اعلامی توسط مربیان یا اساتید در کارتابل نیز بصورت hotfix اعمال، تست و ریلیز شود.
- برای اجرای بک اندی که در اختیارتون قرار میگیره نیاز دارید که نرم افزارهای MongoDB و Nodejs خودتون را به آخرین ورژن ارتقا بدید.
- برای سبد خرید محصولات باید خودتون یک API توسط Next بنویسید.

(اطلاعات بیشتر در بخش فازبندی و تحویل)



موجودیت های سیستم و فیلدهای آنها

مشخصات کالاها: عکس، اسم، گروه، سرگروه، توضیحات بصورت html، قیمت، موجودی، تاریخ درج در سایت

مشخصات گروه کالاها: آیکون، عنوان، عنوان سرگروه

مشخصات سفارشها: فیلدهای مشخصات کاربر (نام، آدرس، تلفن)، مبلغ کل سبد، زمان درخواست تحویل، زمان ثبت

سفارش، زمان پایان تحویل سفارش

صفحات پروژه و قابلیت های آنها

صفحات مدیریت

- لاگین پنل مدیریت : با نام کاربری و رمز کاربر ادمین، دکمه ی ورود
- صفحه ی سفارش ها:

- صفحه ی لندینگ ورود به پنل این گزینه از منوی ادمین میباشد.
- لیست سفارش های ثبت شده با pagination و filter سفارش های تحویل شده یا نشده
- بررسی هر سفارش در مودال و نمایش اطلاعات کامل شامل جدول سبد خرید
- بابت سفارش های تحویل نشده، دکمه ی ثبت اتمام تحویل

• صفحه مدیریت کالا:

- لیست کالاها به همراه عکس، اسم و گروه بندی
- قابلیت حذف هر کالا
- قابلیت افزودن یا ویرایش در مودال فرم
- قابلیت انتخاب یا تغییر عکس کالا در مودال فرم
- قابلیت درج html در فیلد توضیحات

• مدیریت موجودی و قیمت ها:

- لیست کامل کالاها به همراه موجودی و قیمت هر کالا با pagination
- با کلیک روی هر عدد موجودی یا هر قیمت کالا آن فیلد به textbox تبدیل شده و عدد قبلی در آن قابل تغییر است (کد محصول تغییر یافته باید در آرایه ای نگهداری شود)
- پس از اعمال تغییرات به روش فوق در لیست، بعد از زدن ذخیره، با استفاده از آرایه های ردیف های تغییر یافته، promise call های متعددی بصورت سری جهت ویرایش هر محصول اتفاق می افتد و با پاسخ موفق آن ردیف از حالت textbox خارج میشود.



- با زدن دکمه ی `esc` روی هر `textbox` به منزله ی انصراف از تغییر `textbox` مخفی و مقدار قبلی نمایش داده میشود (کد محصول از آرایه ی تغییرات حذف شود)
- در صورت عدم کلیک روی عددها و خالی بودن آرایه ی کد های محصولات تغییر یافته دکمه ی ذخیره غیر فعال باشد.

صفحات مشتریان

- **صفحه اصلی:** لیست گروه های منتخب ادمین به همراه حداقل ۶ محصول از هر گروه
- **لیست محصولات (صفحه ی گروه محصولات):**
 - به ازای هر گروه منحصر بفرد لیست کامل محصولات زیر مجموعه
 - دارای pagination (به صورت `a href` و قابل باز شدن در تب جدید)
 - طبق سورتینگ پیشفرض جدیدترین محصولات بالاتر هستند.
 - ستون سمت راست لیست کامل گروهها و زیر گروهها به همراه آیکون که با کلیک به صفحه لیست آنها میرویم. (سرگروه صفحه ندارد)
- **صفحه محصول:**
 - مشخصات کامل کالا، بخش توضیحات باید `html` بدرستی نمایش داده شود
 - در صورت عدم موجودی: پیام «اتمام موجودی»
 - در صورت موجودی: امکان افزودن کالا با تعداد مد نظر به سبد خرید (`validation` با در نظر گرفتن موجودی)
- **صفحه سبد خرید:**
 - لیست کالاهای موجود در سبد، شامل عکس، اسم، قیمت، تعداد، قیمت تعداد
 - قابلیت تغییر تعداد و حذف (`validation` با در نظر گرفتن موجودی)
 - نمایش جمع کامل
 - دکمه نهایی کردن خرید
- **صفحه نهایی کردن خرید:**
 - فرم اطلاعات تماس کاربر و تاریخ تحویل (`validation`)
 - دکمه پرداخت که با صدا زدن یک `api` در پاسخ لینک صفحه ی پرداخت که باید ریدایرکت کنیم.
 - **صفحه ی ماک درگاه:** یک صفحه `fake` جهت ذخیره وضعیت سفارش و ریدایرکت به صفحه نتیجه
 - **صفحه نتیجه ی پرداخت:** بر اساس `query param` کد سفارش، نتیجه را از سرور نمایش میدهیم.

نکته: صفحات ماک درگاه و نتیجه (برگشت از درگاه) بصورت ریدایرکتی و رفرش شدنی باید کار کنند.



✓ پیشنهاد جهت تکمیل اختیاری و تمرینی

- برای هر ماژول از پروژه لایه ی `service` یا `model` در نظر بگیرید که مسئول ذخیره و بازیابی اطلاعات با فرمت مناسب است. مثلاً سبد خرید در لایه ی `model`
- بجای `ajax` از `Axios` استفاده کنید.
- برای تمام لودینگ های `ajax` ای گرافیک مناسب `loading` در بخش مربوطه در `ui` درج کنید
- ماژول اصلی `ajax` داشته باشید که از `axios` استفاده میکند و بجای استفاده مستقیم از `axios` در `model` ها از ماژول خود استفاده کنید. همچنین خطاهای `4xx` و `5xx` را بطور یکجا در ماژول اصلی `ajax` خود هندل کنید.
- همانند لودینگ برای حالات خطای هر `ajax call` یک `ui` مناسب خطا در بخش مربوطه نمایش دهید.
- افزودن فیلترینگ و سورتینگ های بیشتر و کاربردی به صفحات سایت و ادمین.
- برای سمت کاربر قابلیت جستجوی محصولات در هدر سایت قرار دهید و صفحه ی نتایج جستجو داشته باشید.
- برای انتخاب تاریخ از یک لایبرری آماده ی `date picker` شمسعی استفاده کنید.
- برای فیلد توضیحات `html` ای محصولات از یک لایبرری مثل `WYSIWYG` ادیتور استفاده کنید.
- برای بخش ادمین یک صفحه ی اصلی داشبورد اضافه کنید و نمودار بار چارت تعداد سفارش روزانه و بار چارت جمع مبلغ فروش روزانه و نمودار پای چارت تعداد کالای هر گروه اصلی و گیج درصد کالاهای بدون موجودی نمایش دهید.
- پروژه را بعد از تکمیل نهایی به عنوان نمونه کار خود با تغییر نام رپو بصورت `Public` منتشر کنید.
- جهت ارائه ی راحت این نمونه کار خود به کمک سرویس های رایگان اینترنتی نظیر `heroku`، `netlify` و یا نهایتاً یک هاست اشتراکی ارزان قیمت (سایت خود را بارگزاری کنید). شاید نیاز به اعمال تغییرات در بک‌اند و لایه‌های `model` ماژول هایتان داشته باشید.
- در صورت بارگزاری سایت در جهت بهینه کردن ساینز باندل نهایی تلاش کنید.
- به پروژه ی خود `SSR` اضافه کنید. سعی کنید از `Next Auth` استفاده کنید.