مهندسی اینترنت **تمرین شماره‌ی 5**

مدرس: **دکتر خامس‌پناه**

طراح: پرنا اسدی، امیر مهدی محمدیان، امیرحسین علیزاد

مهلت تحویل: ۸ اردیبهشت

Baloot

مقدمه

اهداف پروژه پنجم درس در دو بخش کلی قابل بیان است:

بخش اول) هدف اولیه آشنایی با مکانیزم CORS[1]، استانداردسازی API و تبدیل خروجی آن به فرمت JSON و همینطور آشنایی و استفاده از ابزاری مثل Postman می‌باشد. دقت کنید که از این فاز به بعد دیگر نیازی به صفحات JSP و Server Side Rendering ندارید.

بخش دوم) هدف بعدی استفاده از چارچوب[2] React برای پیاده سازی سمت کاربر پروژه با استفاده از معماری RIA[3] و اتصال آن به سرور است.

همچنین در این فاز تغییراتی در سرویس خارجی‌ای که در فازهای قبلی با آن کار کردید داده شده است و داده‌ی عکس برای کالا اضافه شده است. برای بررسی این تغییرات می‌توانید داده‌ها را از آدرس زیر دریافت کنید:

http://5.253.25.110:5000/api/v2/commodities

[1] Cross-Origin Resource Sharing
[2] Framework
[3] Rich Internet Applications

بخش اول (مقدمات RIA)

آشنایی با CORS

هنگامی که از یک آدرس بخواهیم به آدرس یا پورت دیگری درخواست HTTP ارسال کنیم، نیاز است که سایت مقصد به ما اجازه‌ی دسترسی به منابعش را داده باشد. این امر با مقداردهی چند پارامتر در Header پاسخ ارسال شده از سوی سایت مقصد مشخص می‌شود و به این مکانیزم CORS گفته می‌شود.

به عنوان اولین فعالیت این تمرین لازم است تا ابتدا با CORS آشنا شوید و سپس Header های مورد نیاز را در کدهای خود قرار دهید. برای قرار دادن هدرهای مربوطه در هر درخواست از Filter استفاده کنید. Filter یک شئ است که قبل از درخواست به API های مورد نظر و همچنین بعد از پاسخ ارسالی آنها قرار می‌گیرد و امکانات بسیاری از قبیل امکان ایجاد تغییرات در درخواست‌ها، پاسخ‌ها و... را در اختیار قرار می‌دهد.

استانداردسازی APIها

در ادامه تمرین لازم است تا API های طراحی شده در مراحل قبل را استاندارد کنید. توصیه می‌کنیم از این کتاب و منابع گفته شده در کلاس استفاده کنید. برای اینکار لازم است تا علاوه بر استانداردسازی URLهای API خود، متد HTTP درخواست‌های خود را نیز استاندارد کنید.

تبدیل خروجی API ها به فرمت JSON

سومین فعالیت شما در این تمرین یکسان کردن فرمت خروجی API های توسعه داده شده است. برای اینکار لازم است تا خروجی آن‌ها را به فرمت JSON تبدیل کنید. API ها باید خروجی خود را به فرمت JSON و به همراه Code Status مناسب برگردانند. برای اینکار کافیست از سرویس‌های

اسپرینگ در بخش Backend برنامه‌ی خود استفاده کنید. این سرویس‌ها به صورت خودکار خروجی را به فرمت JSON تبدیل می‌کنند.

آشنایی و استفاده از Postman

Postman ابزاری برای ارسال درخواست با متدهای HTTP و فرمت دلخواه به یک سرور است. از این ابزار می‌توان برای آزمایش و مشاهده خروجی سرویس‌هایتان استفاده کنید. لازم است تا این ابزار را در هنگام تحویل تمرین روی لپ تاپ خود نصب داشته باشید. در هنگام تحویل، یکی از API های شما به صورت تصادفی با Postman آزموده خواهد شد. توجه کنید در بخش React تمامی سرویس‌های گفته شده باید در Frontend دسترس پذیر باشند و از این ابزار تنها برای آزمایش صحت خروجی سرویس‌هایتان استفاده می‌شود.

بخش دوم (React)

بخش دوم این تمرین شامل پیاده‌سازی تمامی صفحات سامانه در چارچوب React است. در این راستا باید به موارد زیر نیز توجه داشته باشید.

- صفحه جدید Provider به صفحات اضافه شده که طبق UI باید محصولات آن Provider در داخل صفحه نمایش داده شود. با کلیک کردن روی اسم Provider در صفحه محصول، کاربر به صفحه Provider مربوطه Redirect می‌شود.
- در این فاز نیاز است که هنگام زدن روی دکمه Pay Now در صفحه کاربر، Modal ای باز شود که قیمت کل خرید را در آن نشان دهد و امکان استفاده از کد تخفیف در آن وجود داشته باشد و در نهایت خرید با زدن دکه Buy تکمیل شود.
- در صفحه کاربر با زدن روی دکمه Add more credit می‌بایست Modal ای باز شود برای Confirmation و در صورت تایید کاربر سپس به Credit او اضافه شود.
- در صفحه کاربر دکمه جدیدی زیر اطلاعات او اضافه شده که برای Logout کردن از حساب کاربری استفاده می‌شود.
- با کلیک کردن روی محصولات چه به صورت Card و چه به صورت آیتم های داخل History یا سبد خرید، باید کاربر به صفحه محصول مربوطه Redirect شود.
- در صفحه اصلی(Home)، صفحه‌بندی به زیر محصولات اضافه شده است.

- فیلد های ثبت نام باید شامل birthDate، email، password، username و address باشد.
- با کلیک کردن روی username و یا Cart داخل navigation bar بالای صفحه می‌بایست صفحه کاربر نمایش داده شود.
- با کلیک کردن روی آیکن baloot باید صفحه اصلی نشان داده شود.
- جست و جو در محصولات با استفاده از name، category و provider name می‌بایست قابل انجام باشد.
- کاربر تا زمانی که Login نکرده، نباید بتواند صفحه‌ای جز صفحه ورود و ثبت نام را ببیند و در صورت logout باید به صفحه login ریدایرکت شود. (همچنین اگر کاربری لاگین نباشد url های مربوط به صفحات به جز صفحه login و sign up باید به صورت protected routes باشند و باید به صفحه login ریدایرکت شوند.)
- در صورت عدم موفقیت در انجام یک درخواست یا بروز مشکل، نمایش پیام مناسب حائز اهمیت است. برای مثال می‌توانید از کتابخانه React Toastify استفاده کنید. (همچنین می‌توانید پیام‌های موفقیت آمیز مثل "درخواست شما با موفقیت انجام شد" و امثال آن را هم نمایش دهید.)

علاوه بر موارد بالا، سایر امکانات نرم افزار طبق طراحی در فیگما می‌بایست قابل اجرا بوده و تعامل سمت کاربر و سمت سرور در آن به درستی برقرار باشد. برای مثال:

- احراز هویت اولیه و خروج از حساب کاربری مطابق با فازهای قبلی
- امکان ثبت نام کاربر جدید
- امکان اضافه کردن Credit (صفحه یوزر)
- مشاهده لیست محصولات، sort و filter (جست و جو) آن‌ها (صفحه اصلی)
- مشاهده صفحه یک محصول
- افزودن یک محصول به سبد
- امتیازدهی به محصول
- مشاهده سبد خرید و امکان کم و زیاد کردن محصولات در آن (صفحه یوزر)
- خرید نهایی و استفاده از کد تخفیف
- امکان کامنت گذاری زیر محصولات (صفحه محصول)
- امکان رای‌دهی به کامنت‌های کاربران (صفحه محصول)
- مشاهده محصولات پیشنهادی (صفحه محصول)

نکات تکمیلی و راهنمایی

- در صورتی که برای پیاده‌سازی قسمت خاصی از سایت از کدهای آماده موجود در اینترنت استفاده می‌کنید، نحوه‌ی کارکرد آن‌ها را نیز یاد بگیرید و هنگام تحویل با قابلیت‌های مورد استفاده خودتان نیز آشنایی اولیه‌ای داشته باشید.
- توجه داشته باشید که کارت‌های محصولات به صفحه خود محصول لینک دارند، همچنین در صفحه محصول، لینک به توزیع کننده محصول وجود دارد.
- برای Styling می‌توانید از راه‌های مختلفی مثل styled components، sass، css modules و ... استفاده کنید. (بهتر است از css خالی استفاده نکنید)
- برای پیاده‌سازی navigation بین صفحات می‌توانید از کتابخانه React Router Dom استفاده کنید.
- می‌توانید از کتابخانه MUI V5 برای کامپوننت‌هایی که طراحی می‌کنید استفاده کنید.
- توجه داشته باشید که تمیزی کد و استفاده چندباره از کامپوننت‌ها اهمیت زیادی دارد. بنابراین با استفاده از قابلیت‌هایی که React در اختیارتان قرار می‌دهد، سعی در داشتن حداقل کد تکراری داشته باشید.
- مدیریت حالت[4] در این پروژه بسیار مهم است. بنابراین سعی کنید به این موضوع اهمیت زیادی دهید و در ساختار پروژه از آن بهره ببرید.
- برای برقراری ارتباط با سرور و ارسال یا دریافت اطلاعات می‌توانید از Fetch Api و یا ابزار Axios استفاده کنید.
- با مفاهیم Life Cycle و انواع آن در React و همینطور hook ها آشنا باشید، زیرا در حین پیاده سازی به آن احتیاج پیدا خواهید کرد.
- به پوشه‌بندی و نحوه مدیریت فایل‌ها دقت داشته باشید.

مخزن پروژه

یک پروژه‌ی جدید در گیت لب برای کدهای سمت کاربر خود ایجاد کنید و اکانت درس را نیز با دسترسی مناسب به مخزن خود اضافه کنید. نحوه‌ی کامیت کردن در گیت و پیام همراه با آن، امر بسیار مهمی در پروژه‌ها است. تمامی تغییرات خود در هر بخش را به گیت مربوطه اضافه کنید و در نهایت، در مخزن خود بارگذاری کنید. توجه کنید که پروژه شما پس از clone شدن باید به راحتی قابل اجرا باشد.

[4] State Management

نکات پایانی

- کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه سمت کاربر و سمت سرور را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.

- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.

- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آنها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

- ایمیل طراحان پروژه:

parna80as@gmail.com
therealamirmahdi79@gmail.com
aalizad79@gmail.com

موفق و پیروز باشید.