مهندسی اینترنت

تمرین شماره‌ی ۴

مدرس: **دکتر خامس‌پناه**

طراحان: امیر فراهانی، پرنا اسدی

مهلت تحویل: جمعه ۷ اردیبهشت ساعت ۲۳:۵۹

MizDooni

مقدمه

اهداف پروژه چهارم درس در سه بخش کلی قابل بیان است:

بخش اول: اتصال به سرویس‌های خارجی و دریافت داده از آن‌ها

بخش دوم: هدف اولیه آشنایی با مکانیزم CORS، 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 سوم: هدف بعدی استفاده از چارچوب React برای پیاده‌سازی سمت کاربر پروژه با استفاده از معماری RIA و اتصال آن به سرور است. لینک فیگمای فاز جدید پروژه را از طریق این لینک می‌توانید مشاهده کنید.

بخش اول: اتصال به سرویس‌های خارجی

برای دریافت بعضی از اطلاعات، در این فاز شما به یک سرویس خارجی متصل خواهید شد. این سرویس خارجی در آدرس http://91.107.137.117:55/ قابل دسترسی می‌باشد. شما با رفتن به این آدرس می توانید لیست endpointها و فرمت خروجی هر یک را مشاهده کنید. همچنین این امکان وجود دارد که endpointها را در همان‌جا تست و بررسی کنید.

لطفا توجه داشته باشید که یک فیلد با نام "image" به موجودیت رستوران اضافه شده است که در آن یک لینک به عکسی از رستوران قرار دارد. **شما نیز باید این فیلد را به موجودیت خود اضافه کنید.**

بخش دوم: 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 اجازه 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 Headerهای 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 APIهای 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 APIهای 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 Status Code 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 می توان برای آزمایش و مشاهده خروجی سرویس هایتان 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 سوم: React

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

- در این فاز نیاز است که هنگام زدن روی دکمه Add Review در صفحه رستوران، Modal ای باز شود که امتیاز دهی به رستوران در قسمت های مختلف را برای کاربر فراهم کند. و همینطور کاربر باید توانایی اضافه کردن متنی به کامنت خود در این Modal داشته باشد.
- در صورتی کاربر وارد شده مشتری باشد، میتواند در صفحه مشتری رزرو های قبلی خود را با حالت‌های مختلف آن ببیند. اگر هنوز زمان رزرو نرسیده است، مشتری میتواند آن را کنسل کند. اگر مشتری به رزرو خود رفته باشد می‌تواند یک Review برای آن رستوران در همین صفحه اضافه کند. (Modal در همین صفحه باز می‌شود) و اگر رزرو کنسل شده باشد، کاری نمی‌تواند انجام دهد.
- در هنگام کنسل کردن یک رزرو یک Modal باید باز شود که از کاربر تاییدیه بگیرد و در صورت تایید کاربر این اجازه را به کاربر بدهد تا رزرو خود را کنسل کند.
- اگر کاربر وارد شده یک مدیر باشد در صفحه مدیر می‌تواند یک رستوران جدید درست کند و یا رستوران‌های قبلی را مدیریت کند.
- اگر مدیر بر روی دکمه‌ی Add کلیک کند یک Modal برای او باز می‌شود که اطلاعات رستوران را می‌گیرد. توجه کنید که در صورت اشتباه بودن ورودی ها و یا تکراری بودن نام رستوران خطای مناسب را نشان دهید.
- در صفحه‌ی مدیریت رستوران با زدن بر روی دکمه‌ی Add Table یک Modal باز می‌شود و تعداد صندلی میز را گرفته و یک میز در رستوران ساخته می‌شود. توجه فرمایید که آیدی‌های میزها را خودتان باید بر اساس تعداد میزهای هر رستوران تولید کنید. لطفا توجه داشته باشید که بدون Refresh کردن صفحه میزهای داخل رستوران باید بروزرسانی شوند.
- با کلیک کردن بر روی هر میز اطلاعات رزرو آن در سمت چپ نمایش داده می‌شود. همچنین مدیر این امکان را دارد که در قسمت بالا سمت چپ با کلیک کردن بر روی علامت تقویم روز مورد نظر را نیز تغییر دهد و اطلاعات رزروی آن روز باید برای هر میز نمایش داده بشود.
- جست و جو در رستوران‌ها با استفاده از نام، مکان و دسته‌بندی رستوران باید قابل انجام باشد. لیست مکان‌ها باید ترکیبی از نام کشور و شهر باشد که کاربر می‌تواند از بین این لیست برای جستجو انتخاب کند.
- در زیر هر صفحه میانگینی از امتیازات هست که با اضافه شدن یک Review به طور خودکار بروزرسانی می‌شود.
- در صفحه ی هر رستوران امکان رزرو میز در ساعات مشخص وجود دارد. فرآیند رزرو به این شکل است که روز و تعداد افراد مشخص می‌شود و ساعاتی که میز خالی وجود دارند برای کاربر فعال می‌شوند. در نظر داشته باشید که میزهای تعداد بالاتر از تعداد افراد نیز مناسب برای رزرو هستند ولی اولویت برای رزرو برای میزهای با

تعداد کمتر و آیدی کوچکتر است.به طور مثال برای دو نفر اگر در ساعت ۱۴ میز دو نفره وجود نداشته باشد، می‌توان میز ۳ نفره به آنها اختصاص داد.

- در صفحه کاربر دکمه جدیدی زیر اطلاعات او اضافه شده که برای Logout کردن از حساب کاربری استفاده می‌شود.
- در صفحه‌ی خانه در یک ردیف باید ۶ تا از بهترین رستوران‌ها را نشان دهید که بر اساس میانگین تمام امتیازات آن‌ها در تمام قسمت‌ها رده بندی می‌شوند. و همچنین ۶ عدد از مناسب ترین رستوران‌ها برای کاربر نشان داده می‌شوند که رستوران‌ها با بهترین امتیاز هستند که در مکان خود کاربر هستند.
- هر وقت که برروی یک رستوران چه در نتیجه‌ی جستجو و چه در صفحه‌ی خانه و بقیه‌ی قسمت‌هایی که نامی از رستوران برده میشود کلیک شود، باید به صفحه‌ی آن رستوران redirect شود.
- در بخش Review ها اگر تعداد Review ها بیشتر از ۵ عدد بود باید صفحه‌بندی برای آنها انجام شود.
- در بخش نتیجه‌ی Search اگر تعداد نتایج بیش از ۱۲ عدد بود باید صفحه‌بندی برای آنها انجام شود.
- صفحه‌های ثبت‌نام و ورود باید پیاده‌سازی شوند و بشود با آنها کاربر جدید ساخت و همچنین Login کرد. توجه کنید که کنترل داده‌ی ورودی رو برای اطلاعات داخل فیلدها انجام دهید.(خالی بودن فیلد و یا فرمت اشتباه)
- با کلیک کردن روی آیکن Mizdooni باید صفحه اصلی نشان داده شود.
- کاربر تا زمانی که Login نکرده، نباید بتواند صفحه ای 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 signu up باید به صورت routes protected 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 Toastify React 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 به تمامی نکات جزئی که داخل فیگما گذاشته شده‌اند توجه کنید که بخشی از نمره‌ی شما را شامل می‌شوند. به طور مثال اگر در صفحه‌ی کاربر رزروی وجود نداشته باشد باید متن "No Reservations yet" نمایش داده شود.

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.
- برای Styling میتوانید از راههای مختلفی مثل modules css، sass، components styled و ... استفاده کنید.(بهتر است از css خالی استفاده نکنید.)
- برای پیاده‌سازی navigation بین صفحات میتوانید از کتابخانه Dom Router React استفاده کنید.
- میتوانید از کتابخانه 5V MUI 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 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 Api Fetch و یا ابزار Axios استفاده کنید.
- با مفاهیم Cycle Life و انواع آن در React و همینطور hook ها آشنا باشید، زیرا در حین پیاده سازی به آن احتیاج پیدا خواهید کرد.
- به پوشه بندی و نحوه مدیریت فایلها دقت داشته باشید.

Software Engineering Best Practices

با توجه به اصول گفته شده در فاز قبل، آن‌ها را در این فاز هم رعایت کنید و کامپوننت‌ها را تا حد ممکن کوچک تعریف کنید که قابلیت reusable شدن را داشته باشند.

همچنین سعی کنید پکیج های زیادی نصب نکنید، زیرا نصب پکیج های جدید سرباری بر DX خواهند داشت.

اگر فانکشنالیتی کوچک داشتید مثل pagination یا حساب کردن datetime، اگر پیاده‌سازی خیلی پیچیده‌ای نداشت خودتان آن بخش را پیاده‌سازی کنید، در غیر این صورت مانعی برای نصب پکیج وجود ندارد.

پوشه بندی حتما رعایت شود و کامپوننت‌های یک صفحه به راحتی قابل دسترسی باشند (برای این اسکیل پروژه تحقیق کنید که چه نوعی می‌تواند پوشه بندی بهتری باشد).

Git Commit

همان‌طور که در پروژه اول توضیح داده شد، کامیت‌ها اهمیت زیادی در توسعه پروژه‌های نرم‌افزاری دارند. در این پروژه نیز باید مواردی که در پروژه اول گفته شدند، رعایت شوند. رعایت این قسمت، بخشی از نمره شما را در این پروژه تعیین می‌کند.

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.

موفق و پیروز باشید.