مدرس: دکتر احسان خامس‌پناه سنا ساری نوایی، میثاق محقق

MioBook

مقدمه

در این فاز از پروژه، از فریم‌ورک React برای پیاده‌سازی بخش فرانت‌اند پروژه و اتصال آن به سرور بک‌اند خود استفاده می‌کنید. در این فاز تغییراتی در برخی از صفحات به وجود آمده است و همچنین صفحات جدیدی اضافه شده‌اند که شما باید به درستی پیاده‌سازی کنید.

طراحی صفحات

این تمرین شامل پیاده‌سازی تمامی صفحات در فریم‌ورک React است. در ادامه به توضیح جزئیات هر صفحه پرداخته شده است. شما باید صفحات داده شده را مانند طراحی‌هایی که از محیط کاربری پروژه انجام شده و از این لینک قابل دسترسی اند، پیاده‌سازی نمایید.

در این فاز طراحی Responsive صفحات نیز اضافه شده‌اند که شما باید با استفاده از فریم‌ورک Bootstrap (و در شرایط خاص، media query-ها)، آنها را هندل کنید. پیاده‌سازی شما باید روی همه عرض صفحه‌ها از 576px (برک‌پوینت sm در بوت‌استرپ) تا بالای 1400px که دسکتاپ است به درستی نمایش داده شود و از container-ها استفاده کرده باشید. همچنین نمایش صحیح صفحات در ابعاد 360px (که از عرض‌های معمول برای گوشی‌ها است) نمره امتیازی دارد.

همچنین در این فاز، امکان استفاده از فریم‌ورک کامل Bootstrap را دارید (یعنی از اجزایی از آن که به JS نیاز داشتند نیز می‌توانید استفاده کنید).

توجه کنید که برای برخی نیازمندی‌های فرانت‌اند، ممکن است ملزم به اضافه کردن API-های جدید به بک‌اند و یا تغییر آنها باشید. این کار مانعی ندارد و به احتمال زیاد نیاز می‌شود. همچنین ممکن است که نیازمند اضافه کردن فیلدهای جدید به کلاس‌ها بشوید. مثلا باید لینک عکس کتاب و نویسنده را اضافه کنید.

صفحه Sign up

- برای ثبت نام به نام کاربری، رمز عبور، ایمیل، کشور، شهر و نقش کاربر نیاز دارید.
- تا زمانی که تمامی فیلدها پر نشده‌اند، نباید دکمه ثبت نام فعال شود.
- در حالت عادی رمز عبور را به‌صورت کاراکترهای پنهان (●●●) نشان دهید اما با کلیک کاربر روی آیکون چشم کنار ورودی، متن آن باید به صورت خوانا نمایش داده شود.
- نام کاربری و ایمیل هر کاربر یکتا است. در صورت وارد کردن نامی تکراری، border فیلد را به رنگ قرمز تغییر دهید و پیام خطا را با رنگ قرمز در زیر آن نمایش دهید.

- سایر خطاها (مانند معتبر نبودن ایمیل و رعایت نشدن حداقل کاراکترهای رمز عبور) نیز باید بررسی شوند.
- با کلیک بر روی لینک Sign in وارد صفحه Sign in می‌شویم.

صفحه Sign in

- برای لاگین به نام کاربری و رمز عبور نیاز دارید.
- نمایش پسورد در اینجا مانند صفحه ثبت نام است و تا قبل از پر شدن تمامی فیلدها، دکمه لاگین غیرفعال است.
- در صورت وجود نداشتن کاربر و یا اشتباه بودن رمز عبور او، خطا نمایش داده می‌شود.

صفحه Home

- پس از ثبت‌نام یا ورود، به این صفحه منتقل می‌شوید.
- اگر کاربر لاگین نباشد، نمی‌تواند هیچ صفحه‌ای به جز صفحه ورود و ثبت‌نام را مشاهده کند. به عبارت دیگر، URL-های مربوط به صفحات دیگر باید به صورت protected route تنظیم شوند و در صورت دسترسی غیرمجاز، کاربر به صفحه ورود منتقل شود.
- با کلیک بر روی دایره پروفایل کاربر در هدر (که در آن حرف اول نام کاربر است)، منوی پروفایل باز می‌شود.
- هر کدام از آیتم‌های منوی پروفایل، کاربر را به صفحه مد نظر آن هدایت می‌کنند.
- با کلیک بر روی آیکون سایت در هدر، باید صفحه Home نشان داده شود.
- هدر به صورت sticky می‌باشد و با اسکرول صفحه در جای خود می‌ماند و با صفحه پایین می‌آید.
- در قسمت جستجو، کلمه مورد نظر را وارد می‌کنید و سپس به صفحه Search Result منتقل می‌شوید.
- در بخش New Releases، کتاب‌های تازه اضافه شده و در بخش Top Rated، کتاب‌هایی را که بیشترین امتیاز را دارند نشان داده می‌شوند.
- در طراحی ریسپانسیو، در صورت کوچک شدن هدر، نوار سرچ از آن حذف شده و به داخل صفحه منتقل می‌شود. همچنین کارت‌های کتاب، باید بنا بر عرض صفحه کنار هم قرار گیرند و مثلا اگر جا بود، 2 تا در هر ردیف باشند.

صفحه Search Result

- اگر تعداد نتایج بیشتر از 10 عدد بود، باید صفحه بندی برای آنها انجام شود.
- در اینجا نیز مانند صفحه Home، در طراحی ریسپانسیو کتاب‌ها به هر تعداد که جا می‌شوند در یک ردیف قرار می‌گیرند.

- با کلیک روی دکمه Add to Cart، یک Modal نمایش دهید که از کاربر بپرسد آیا می‌خواهد کتاب را قرض بگیرد یا بخرد. در نهایت، کتاب را به سبد خرید او اضافه کنید. این Modal در صفحه Book بیشتر توضیح داده شده است.
- با کلیک بر روی هر کارت، به صفحه کتاب مورد نظر منتقل می‌شوید.
- با زدن دکمه Filter، یک sidebar باز شده که در آن انواع فیلترهای کتاب‌ها آورده شده و کاربر می‌تواند اعمال کند.
- برای فیلتر ژانر، در dropdown آن لیست همه ژانرهای موجود را نشان دهید.

صفحه Book

- در بخش اول این صفحه اطلاعات کتاب نظیر عکس کتاب، اسم آن، قیمت، خلاصه و میانگین امتیاز کتاب را نشان می‌دهید.
- در بخش دوم نظرات مربوط به این کتاب نمایش داده می‌شود. در صورتی که تعداد Review-ها از 4 عدد بیشتر باشد، باید صفحه بندی برای آنها انجام شود.
- با کلیک روی دکمه Add Review، مودال مربوطه باز می‌شود. تا زمانی که فیلد امتیاز و نظر تکمیل نشده باشند، دکمه Submit Review فعال نمی‌شود.
- پس از ثبت Review، کاربر می‌تواند در همان لحظه بازخورد خود را ببیند و تعداد نظرات و میانگین امتیاز کتاب نیز بدون Refresh شدن صفحه بروزرسانی می‌شود.
- اگر کاربر کتاب مورد نظر را قرض نگرفته یا نخریده باشد، نباید امکان ثبت نظر برای آن را داشته باشد. این مورد را در بخش نمایش ارورها به روشی صحیح به کاربر نشان دهید.
- در Modal مربوط به اضافه کردن کتاب به سبد خرید، قیمت کتاب را مشاهده می‌کنیم و در صورت زدن تیک قرض کتاب، Radio button-های مربوط به تعداد روز قرض ظاهر می‌شوند که با زدن آنها، قیمت نهایی بروزرسانی می‌شود.
- پس از زدن دکمه Add برای سبد خرید یا اضافه کردن بازخورد، موفقیت آمیز بودن کار را با Toast نشان دهید.
- بر روی عکس کتاب، یک Badge وجود دارد که می‌تواند سه مقدار Available، Owned و Borrowed را بگیرد. همچنین در صورتی که کتاب در دارایی‌های کاربر باشد، امکان اضافه کردن آن به سبد خرید وجود نخواهد داشت.

صفحه Author

- با کلیک بر روی نام نویسنده در صفحه کتاب، به این صفحه منتقل می‌شوید.
- در این صفحه مشخصات نویسنده به همراه لیست همه کتاب‌های او را نمایش دهید.

صفحه User

- اطلاعات کاربر نظیر نام کاربری و ایمیل را در صفحه نمایش دهید.
- با کلیک بر روی دکمه Logout، کاربر را به صفحه ورود منتقل کنید.
- طراحی ریسپانسیو را باید برای این صفحات که طرح موبایل ندارند نیز انجام دهید.
- فیلد مربوط به افزودن موجودی باید فقط اعداد را بپذیرد.
- در این صفحه لیست کتاب‌های دارایی کاربر نیز نشان داده می‌شود و با کلیک بر دکمه Read وارد صفحه محتوای کتاب می‌شویم.
- در صورت خالی بودن دارایی کاربر، یک طراحی مناسب برای خالی بودن لیست نشان دهید.
- با کلیک کردن بر روی عکس یا نام کتاب در لیست، به صفحه کتاب هدایت می‌شویم و با کلیک بر روی نام نویسنده، به صفحه نویسنده آن می‌رویم.
- ستون Status در صورت قرضی بودن کتاب، تاریخ اتمام قرض را نشان می‌دهد.
- دکمه پروفایل و My Books در منوی کاربر در هدر، هر دو به همین صفحه می‌آیند.

صفحه Buy Cart

- کاربر امکان حذف کتاب از سبد خرید را دارد.
- برای کتاب‌های قرض شده، قیمت اصلی را به همراه قیمت پس از قرض آن نمایش دهید.
- اگر مجموع هزینه‌های سبد خرید از دارایی کاربر بیشتر باشد، با نمایش یک پیام خطای مناسب، او را از مشکل مطلع کرده و از انجام خرید جلوگیری کنید.
- در صورت خالی بودن سبد خرید، با طراحی مناسب آن را نشان دهید.
- در اینجا نیز با کلیک بر روی اجزای لیست، به صفحه آن می‌رویم.

صفحه Purchase History

- در این صفحه به ازای هر بار تکمیل خرید کاربر، یک بخش جدید داریم که با کلیک بر روی آن، جدول کتاب‌های خریده شده در آن تاریخ را نشان می‌دهد.
- در کنار تاریخ خرید، جمع مبلغ پرداخت شده برای آن سفارش را نیز نشان دهید.

صفحه Admin

- در این صفحه، دو دکمه افزودن کتاب و افزودن نویسنده وجود دارد که با کلیک روی هرکدام، مودال مربوطه باز می‌شود.
- برای افزودن نویسنده به اطلاعاتی نظیر نام، نام مستعار، تاریخ تولد و فوت و ملیت نیاز دارید.
- نوع ورودی فیلدهای تاریخ تولد و تاریخ فوت را روی date تنظیم کنید.

- نام نویسنده و نام کتاب نباید تکراری باشد. در صورت تکراری بودن، خطای مناسب به کاربر نشان دهید.
- اگر کتاب یا نویسنده‌ای نداشتید، جدول‌ها را نمایش ندهید و یک طراحی مناسب برای نمایش خالی بودن صفحه در نظر بگیرید.
- در جدول کتاب‌ها، تعداد خرید آنها توسط کاربران را نیز نمایش دهید.

صفحه Book Content

- در این صفحه، تیتر، نام نویسنده و محتوای کتاب را نشان دهید.
- توجه داشته باشید که اگر کاربری با تغییر دستی URL وارد این صفحه می‌شود، نباید بتواند کتاب‌هایی که در دارایی او نیست را ببیند.

صفحه Error

- در صورت رخ دادن خطای غیر منتظره، کاربر به صفحه خطا هدایت شده و متن خطا در آنجا نمایش داده می‌شود.
- در صورت دسترسی کاربر به یک صفحه و URL که وجود ندارد، کاربر به صفحه 404 منتقل می‌شود.
- طراحی این دو صفحه با خود شما می‌باشد.

فریم‌ورک React

آشنایی

در واقع، ری‌اکت یک کتابخانه برای JavaScript محسوب می‌شود. این کتابخانه با گذاشتن یک تگ <script> در HTML به سایت ما اضافه می‌شود و ما می‌توانیم از امکانات آن استفاده کنیم. می‌توان از جمله امکانات آن به موارد زیر اشاره کرد:

- ساختار Component-ها
- Virtual DOM برای سریع شدن re-rendering
- Hook-ها برای مدیریت Life Cycle و State

شما در تمرین دوم با HTML و CSS آشنا شدید. برای این تمرین خوب است در ابتدا مروری بر مفاهیم JavaScript داشته باشید و سپس وارد ری‌اکت بشوید. JS زبان بزرگی می‌باشد (شامل خود زبان و ترکیب آن با مرورگر و DOM) و عمیق شدن در آن وقت زیادی می‌خواهد. در اینجا هدف آشنایی کلی با آن جهت یادگیری ری‌اکت می‌باشد. برای این کار می‌توانید مقدمات اینجا را مطالعه کنید و یا به این لینک برای توضیحات کامل‌تر مراجعه کنید. همچنین همانند HTML و CSS، MDN منبع خوبی برای درک توابع جزئی‌تر است.

برای آشنایی با نحوه کارکرد ری‌اکت، داکیومنتیشن خود آن بسیار مفید می‌باشد (لینک). بخش Quick Start و Tutorial آن را حتما مطالعه کنید تا با Component-ها و State و Hook-ها آشنا شوید. بقیه بخش‌های آن نیز مفید بوده و خوب است به آنها نگاه اندازید. مثلا چالش‌های مدیریت State را ببینید و با Context آشنا شوید.

ساخت پروژه

برای ساخت یک پروژه ری‌اکت راه‌های متعددی وجود دارد. در واقع، ابزارهای ساخت پروژه ری‌اکت آن را از یک کتابخانه فراتر می‌برند و شبیه یک فریم‌ورک می‌کنند.

در دنیای وب، برای ارائه یک صفحه به کاربر استراتژی‌های متعددی داریم:

- Template Rendering
- Client-Side Rendering (CSR)
- Server-Side Rendering (SSR)
- Static Site Generation (SSG)

یک نمونه از نوع اول، JSP است که در درس دیده‌اید. در آن روش صفحه به طور کامل در بک‌اند پردازش شده و HTML نهایی به مرورگر ارسال می‌شود.

هدف ما در این تمرین CSR در یک Single Page Application یا همان SPA می‌باشد. این یعنی برنامه ری‌اکت ما در نهایت به یک فایل HTML و JS کامپایل می‌شود که توسط یک static file server قابل دریافت است و سپس در سمت مرورگر کاربر پردازش می‌شود.

در استراتژی SSR، بخش زیادی از پردازش‌ها روی سرور انجام شده و سپس فایل‌های پردازش شده به کاربر ارسال می‌شود تا لود مرورگر و ریکوئست‌های مورد نیاز کاهش یابند. در استراتژی SSG، کل فرانت‌اند در مرحله کامپایل کردن تولید می‌شود و سایت Static می‌باشد. این راه برای Blog-ها مناسب می‌باشد.
یکی از ابزارهایی که مدت زیادی برای ساخت پروژه ری‌اکت استفاده می‌شد، Create React App بود که اخیرا منسوخ شده است. این ابزار یک پروژه ری‌اکت مخصوص CSR می‌ساخت که از Webpack استفاده می‌کرد.
هم اکنون داک رسمی ری‌اکت، ابزار Next.js را به عنوان یک full-stack framework پیشنهاد می‌کند که علاوه بر CSR، قابلیت‌های SSR و غیره را نیز دارد. ولی از آنجا که می‌خواهیم تمرکز تمرین را بر روی خود ری‌اکت بگذاریم، از ابزار **Vite** استفاده می‌کنیم که تا حدودی جایگزین Create React App است و در داک ری‌اکت نیز به آن اشاره شده است. برای استفاده از آن این لینک را دنبال کنید.
همانطور که در لینک هم مشخص است، شما به ابزار Node.js نیاز دارید و npm یا node package manager به همراه آن نصب خواهد شد.

روتر

از آنجا که ما با ری‌اکت یک Single Page Application می‌سازیم، به طور پیش‌فرض قابلیت داشتن route-های مختلف مانند /user/settings و navigation بین صفحات را نداریم. برای این کار فریم‌ورک Next.js روتر خود را عرضه کرده است و بقیه ابزارها هم از یک کتابخانه به نام React Router استفاده می‌کنند.
این کتابخانه در نسخه جدید خود سه mode فریم‌ورک، دیتا و عادی را دارد که ما برای راحتی استفاده در Vite از مورد آخر استفاده می‌کنیم. طریقه استفاده از آن در این لینک آمده است.
توجه کنید که در SPA-ها، هیچ‌گاه صفحه نباید reload بشود. تغییر بین صفحات نیز از طریق React Router انجام شده و هیچ وقت ریلود شدن صفحه را مانند زدن دکمه ریلود مرورگر نداریم. در این راستا باید همیشه به جای تگ <a> از <Link> استفاده کنیم و برای تغییر صفحه در JS از هوک useNavigate استفاده کنیم.
کارهای دیگری مانند گرفتن Query Parameter-ها و یا گرفتن قسمتی مانند ID در خود URL Path را نیز می‌توانیم با همین کتابخانه انجام دهیم و در لینک داده شده مثال آن آورده شده است.

فراخوانی API

برای برقراری ارتباط با سرور و ارسال یا دریافت اطلاعات می‌توانید از Fetch API و یا Axios استفاده کنید.
Fetch API در خود استاندارد JS وجود دارد و قابلیت ارسال ریکوئست‌های HTTP مانند GET و POST و... را در مرورگر فراهم می‌کند. می‌توانید در این لینک با آن آشنا شوید. از آنجا که سایت ما نباید در هنگام ریکوئست فریز شود، این نوع کارها به صورت Asynchronous انجام می‌شوند. به این منظور از promise-ها (ساختار then.) و یا همانند لینک، از سینتکس async / await استفاده می‌شود.
کتابخانه Axios نیز به همین منظور و با سینتکسی مشابه با Fetch API وجود دارد که در کل انجام ریکوئست را به نسبت آن راحت‌تر می‌کند ولی باید به پروژه اضافه شود.

در ری‌اکت معمولا ریکوئست‌ای را باید در هنگام تغییر صفحات برای لود کردن محتوای آن انجام دهیم. برای این کار معمولا از هوک useEffect استفاده می‌کنند. در بخش‌های اول این لینک نحوه این کار با Fetch و Axios توضیح داده شده است.

شما در محیط development، بک‌اند اسپرینگ خود را روی localhost بالا می‌آورید و می‌خواهید که از طریق فرانت‌اند به آن ریکوئست بزنید. از آنجا که برای انجام ریکوئست باید URL مقصد را تعیین کنیم، در اینجا لینک‌ها همانند http://localhost:8080/api/example می‌شوند. آوردن این لینک در وسط کد ری‌اکت خوب نیست و ما صرفا باید api/example/ را به عنوان لینک مقصد ریکوئست استفاده کنیم. این کار باعث انجام ریکوئست روی خود سرور ری‌اکت می‌شود که پاسخی برای آن ندارد. در محیط production، با استفاده از ابزاری مانند Nginx، چنین لینک‌هایی را به بک‌اند forward می‌کنیم. برای شبیه‌سازی این کار در محیط development، از قابلیت proxy در Vite استفاده می‌کنیم و همه لینک‌هایی که با api/ شروع می‌شوند را به IP بک‌اند forward می‌کنیم. تنظیمات این قابلیت در فایل vite.config.js انجام می‌شود.

نکات تکمیلی

- در صورت عدم موفقیت در انجام یک درخواست یا بروز مشکل، نمایش پیام مناسب حائز اهمیت است. در این راستا در برخی جاها همانطور که در طراحی داده شده مشخص شده است، خطا به صورت متن قرمز روی صفحه نشان داده می‌شود. سایر خطاها و همچنین پیام‌های موفقیت‌آمیز مانند «درخواست شما با موفقیت انجام شد» را با استفاده از Toast-ها نمایش دهید. برای این کار از کتابخانه React Toastify استفاده کنید (لینک سایت سازنده).
- سعی کنید پیاده‌سازی شما تا حد ممکن با طراحی فیگما هماهنگ باشد. نیازی به دقت کامل در اندازه‌ها نیست، اما باید شکل کلی پروژه تا حد زیادی مشابه طراحی ارائه‌شده باشد.
- در صورتی که برای پیاده‌سازی قسمت خاصی از سایت از کدهای آماده موجود در اینترنت استفاده می‌کنید، نحوه کارکرد آنها را نیز یاد بگیرید و هنگام تحویل با آنها آشنایی داشته باشید.
- 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.
- با مفاهیم Life Cycle و انواع آن در React و همینطور Hook-ها آشنا باشید، زیرا در حین پیاده‌سازی به آنها احتیاج پیدا خواهید کرد.
- مدیریت حالت (State) در این پروژه بسیار مهم است. بنابراین سعی کنید به این موضوع اهمیت زیادی دهید و در ساختار پروژه از آن بهره ببرید.
- توجه داشته باشید که کامپوننت‌های Class-based راه قدیمی ساخت کامپوننت در ری‌اکت اند و شما باید از کامپوننت‌های Function-based به همراه Hook-ها استفاده کنید.
- از TypeScript برای کدها استفاده نکنید و در هنگام ساخت پروژه Vite، از JavaScript استفاده کنید.
- به پوشه‌بندی و نحوه مدیریت فایل‌ها دقت داشته باشید.

Best Practice-ها

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 خوانا و معنادار باشند و قابلیت استفاده چند باره را داشته باشند.
همچنین سعی کنید پکیج‌های زیادی نصب نکنید، زیرا نصب پکیج‌های جدید سرباری بر DX (Developer Experience) خواهند داشت.
اگر فانکشنالیتی‌های کوچکی مانند Pagination یا محاسبه DateTime داشتید و پیاده‌سازی آنها پیچیده نبود، خودتان آن بخش را پیاده‌سازی کنید. در غیر این صورت، مانعی برای استفاده از پکیج‌های موجود وجود ندارد.
همچنین پوشه‌بندی حتما رعایت شود و کامپوننت‌های یک صفحه به راحتی قابل دسترسی باشند.

نکات پایانی

- این تمرین در گروه‌های حداکثر دو نفره انجام می‌شود. برای تحویل آن کافی است که یکی از اعضای گروه، لینک مخزن گیت‌هاب و Hash مربوط به آخرین کامیت پروژه را در سایت درس آپلود کند. پروژه شما بر روی این کامیت مورد ارزیابی قرار می‌گیرد.
- حتما کاربر IE-S04 را به پروژه خود اضافه کنید.
- ساختار مناسب و تمیزی کد برنامه، بخشی از نمره همه پروژه‌های شما خواهد بود. بنابراین در طراحی ساختار برنامه و همچنین خوانایی کد دقت زیادی به خرج دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده شباهت 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 گروه 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 تمرین ارتباط برقرار کنید.