دوره فرانت‌اند

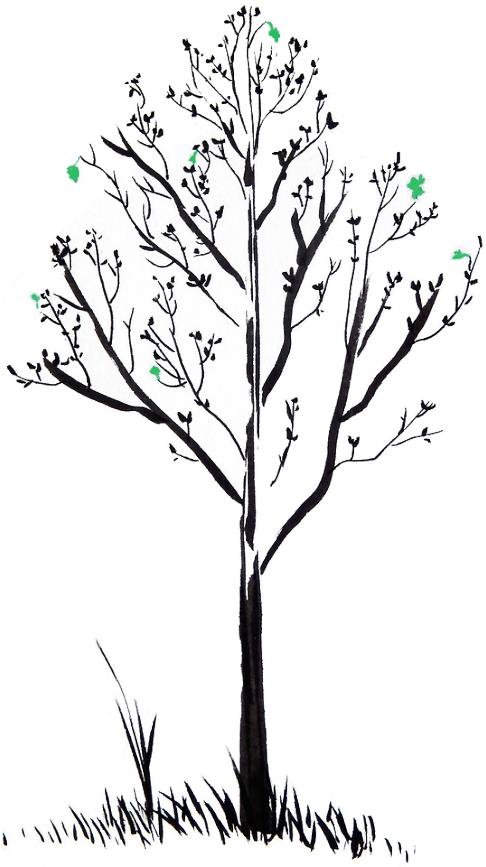
ACM Summer of Code 2024

مدرس: میثاق محقق

منتور: سنا ساری نوایی

تمرین ششم

<API>



فاز دوم

در این تمرین به ادامه پیاده‌سازی سایت رزرو میز رستوران می‌پردازید.
برای این تمرین ابتدا کد CA5 خود را کپی کنید و سپس تغییرات را روی آن اعمال کنید.

در این فاز در کل یک صفحه لاگین اضافه شده و داده‌هایی که قبلا به صورت استتیک در آرایه آبجکت‌های Restaurant و Review بودند، از API بک‌اند این سایت دریافت می‌شوند.

API این سایت در URL روبه‌رو هاست شده است: http://127.0.0.1:8080/api
جهت بررسی اتصال خود به سرور، دستور curl زیر را در Terminal/CMD وارد کنید:

```
curl http://127.0.0.1:8080/api/status/health

Response:
{
    "timestamp": "2024-08-31 18:14:20",
    "status": 200,
    "success": true,
    "message": "server is up",
    "data": {
        "condition": "healthy"
    }
}
```

در ادامه به ازای هر endpoint یک نمونه response آورده شده که منجر به تعداد صفحه‌های زیاد شده است.
خیلی به آنها توجه نکنید و صرفا ببینید که چطوری آبجکت‌های Restaruant و Review را می‌توانید از آنها استخراج کنید. همچنین بهتر است که تیترها را به ترتیب انجام دهید.

دریافت لیست رستوران‌ها

در هنگام ورود به صفحه Home، با استفاده از fetch در useEffect و ذخیره نتیجه آن در useState-ها طبق صحبت‌های کلاس، لیست تمامی رستوران‌ها را ذخیره کنید.
به ازای هر رستوران، یک کارت خواهیم داشت که به صفحه رستوران مدنظر Link دارد.
در صورت بروز مشکل در ریکوئست، یا با استفاده از لایبرری react-toastify یک toast.error نشان دهید و یا با داشتن یک state برای ارور، رخ دادن ارور را در صفحه نشان دهید.
همانطور که در کلاس نشان داده شد، به ازای هر ریکوئست می‌توان isLoading، error و داده نهایی را نگه داشت. نیازی به نمایش حالت load شدن صفحه نیست و صرفا در صورت وجود ارور می‌توانید از این راه استفاده کنید.

API مدنظر برای دریافت لیست همه رستوران‌ها به شکل زیر تعریف شده است:

```
curl http://127.0.0.1:8080/api/restaurants?page=1&sort=rating

Response:
{
    "timestamp": "2024-08-31 18:33:45",
    "status": 200,
    "success": true,
    "message": "restaurants listed",
    "data": {
        "page": 1,
        "hasNext": false,
        "totalPages": 1,
        "size": 1,
        "pageList": [
            {
                "id": 0,
                "name": "M Restaurant",
                "type": "Kebab",
                "startTime": "09:00",
                "endTime": "18:00",
                "description": "Lorem ipsum.",
                "address": {
                    "country": "Iran",
                    "city": "Karaj",
                    "street": "Bahar"
                },
                "starCount": 4,
                "averageRating": {
                    "food": 4.0,
                    "service": 4.0,
                    "ambiance": 3.0,
                    "overall": 4.0
                },
                "maxSeatsNumber": 6,
                "managerUsername": "Misagh",
                "image": "/restaurant-placeholder.jpg",
                "totalReviews": 1
            }
        ]
    }
}
```

همانطور که می‌بینید، می‌توانید صرفا به بخش pageList توجه کنید که لیستی از آبجکت‌های Restaurant است. یعنی پس از خواندن body ریسپانس، مقدار body.data.pageList را در یک state ذخیره می‌کنیم که لیستی از رستوران‌ها است. با چرخه زدن روی آن می‌توانیم کارت‌ها را بسازیم.

بیشتر بدانید: اگر به URL دقت کنید، در انتهای آن یکی از Query Parameter-های تعریف شده page=1 است. API مدنظر بدون داشتن پارامتر page به ما پاسخ نمی‌دهد (400 Bad Request می‌دهد).

این به این خاطر است که در بک‌اند اگر هر ریکوئست تمامی رستوران‌ها را بخواهد از دیتابیس بخواند، بار بسیار سنگینی به آن وارد می‌شود. برای همین دسترسی به داده‌های بزرگ در page-ها صورت می‌گیرد. اینجا نیازی به درنظر گرفتن آن نیست و صرفا محتوای pageList صفحه اول را در کارت‌ها می‌گذاریم.

دریافت اطلاعات یک رستوران

جهت دریافت اطلاعات مربوط به یک رستوران با استفاده از ID آن، از endpoint زیر استفاده کنید:

```
curl http://127.0.0.1:8080/api/restaurants/0

Response:
{
    "timestamp": "2024-08-31 18:36:40",
    "status": 200,
    "success": true,
    "message": "restaurant found",
    "data": {
        "id": 0,
        "name": "M Restaurant",
        "type": "Kebab",
        "startTime": "09:00",
        "endTime": "18:00",
        "description": "Lorem ipsum.",
        "address": {
            "country": "Iran",
            "city": "Karaj",
            "street": "Bahar"
        },
        "starCount": 4,
        "averageRating": {
            "food": 4.0,
            "service": 4.0,
            "ambiance": 3.0,
            "overall": 4.0
        },
        "maxSeatsNumber": 6,
        "managerUsername": "Misagh",
        "image": "/restaurant-placeholder.jpg",
        "totalReviews": 1
    }
}
```

پس از کلیک بر روی هر Card، به صفحه /restaurant/:id وارد می‌شویم که مقدار id: همان مقدار فیلد id آبجکت رستوران Card است. توجه کنید که URL صفحات سایت ری‌اکت ما هیچ ربطی به لینک‌های API ندارد. در صفحه رستوران، با یک fetch اطلاعات آن را از روی id می‌گیریم و در یک state ذخیره و نمایش می‌دهیم. اینجا مقدار body.data همان آبجکت رستوران است که در صفحه Home یک لیست از آن داشتیم.

دریافت نظرات یک رستوران

برای دریافت لیست Review-ها از endpoint زیر استفاده کنید:

```
curl http://127.0.0.1:8080/api/reviews/0?page=1

Response:
{
    "timestamp": "2024-08-31 19:26:36",
    "status": 200,
    "success": true,
    "message": "reviews for restaurant (0): M Restaurant",
    "data": {
        "page": 1,
        "hasNext": false,
        "totalPages": 1,
        "size": 1,
        "pageList": [
            {
                "id": 1,
                "user": {
                    "id": 1,
                    "username": "TheMM",
                    "email": "themm@gmail.com"
                },
                "rating": {
                    "food": 4.0,
                    "service": 4.0,
                    "ambiance": 3.0,
                    "overall": 4.0
                },
                "comment": "Lorem truly the worst food ipsum.",
                "datetime": "2024-08-31 12:27:02",
                "starCount": 4
            }
        ]
    }
}
```

اینجا هم مشابه Home، یک pageList داریم که لیستی از آبجکت‌های Review است.

صفحه لاگین

یک صفحه لاگین ساده طراحی کنید که دارای یک فرم با فیلدهای username و password می‌باشد.

این صفحه را روی مسیر /login در React Router تنظیم کنید.

جهت لاگین شدن در سامانه از endpoint زیر استفاده کنید و یک ریکوئست POST بزنید.

در صورتی که response داده شده 200 OK نباشد، یعنی لاگین موفق نبوده و باید خطا به کاربر نشان داده شود.

```
curl -X POST -H "Content-Type: application/json" -d
"{\"username\":\"test\",\"password\":\"test\"}"
http://127.0.0.1:8080/api/login

Request Body:
{
    "username": "test",
    "password": "test"
}

Response:
{
    "timestamp": "2024-08-31 23:13:43",
    "status": 200,
    "success": true,
    "message": "login successful",
    "token": "eyJ0--longstring--kdZj",
    "data": {
        "id": 0,
        "username": "Misagh",
        "email": "misaghmhgg@gmail.com",
        "address": {
            "country": "Iran",
            "city": "Karaj"
        },
        "role": "client"
    }
}
```

پس در ابتدا مانند بقیه فرم‌هایی که در ری‌اکت دیدیم، داده‌های فرم (username و password) را در یک state ذخیره و آپدیت می‌کنیم. در onSubmit فرم، یک fetch می‌گذاریم که ریکوئست POST می‌فرستد:

```
const response = await fetch("/api/login", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(formData)
}
```

پس از لاگین شدن، اطلاعات کاربر (body.data) را ذخیره کنید و در هدر و صفحه Customer آنها را نمایش دهید. همچنین فیلد token را هم حتما سیو کنید.

برای نگه‌داری این اطلاعات، می‌توانید از Context استفاده کنید (کل کد را در یک AuthContext.Provider قرار دهید) و یا اطلاعات را در local storage ذخیره کنید (یا از هر دو استفاده کنید).

در صورت زدن دکمه logout در صفحه Customer، اطلاعات کاربر را پاک کنید و به صفحه لاگین Navigate کنید.

به ازای هر نفر یک کاربر با نام user_n ساخته شده که n عدد ردیف شما در شیت‌های CA-n Scores.pdf است. رمز آن نیز pass_n می‌باشد. لطفا وارد اکانت دیگران نشوید.

گذاشتن نظر بر روی یک رستوران

جهت گذاشتن نظر بر روی یک رستوران، از API زیر استفاده کنید.

توجه کنید که باید حتما در ۲۴ ساعت اخیر لاگین کرده باشید و مقدار token دریافت شده پس از لاگین را در فیلد هدر ریکوئست Authorization قرار دهید.

```
const response = await fetch(`/api/reviews/${restaurantId}`, {
    method: "POST",
    headers: {
        "Content-Type": "application/json",
        "Authorization": `Bearer ${token}`
    },
    body: JSON.stringify(formData)
}
```

این ریکوئست در صورت خطا پاسخی به جز 200 OK می‌دهد.

در صورتی که نظر مجدد بگذارید، نظر ثبت شده قبلی شما توسط بک‌اند آپدیت می‌شود.

در اینجا محتوای formData باید به صورت زیر باشد:

```
const initialFormData = {
    comment: '',
    rating: {
        food: 0,
        ambiance: 0,
        service: 0,
        overall: 0
    }
}
```

و پس از پر کردن فرم Modal اضافه کردن نظر، باید مقدار هر فیلد rating بین ۱ تا ۵ باشد.

در نهایت بر روی یک رستوران یک نظر حاوی نام و نام‌خانوادگی خود بگذارید.

امتیازی: محدود کردن سایت در صورت لاگین نبودن

در صورتی که کاربر لاگین نیست، وارد کرد مسیرهای معتبر سایت (یعنی /، /customer، /restaurant/:id) منجر به redirect شدن به /login می‌شود.

برای راحتی پیاده‌سازی این بخش، می‌توانید از یک component استفاده کنید که دور element-های اصلی در لیست React Router را می‌گیرد و قبل از اینکه محتوای خود را return children کند، چک می‌کند که اگر کاربر لاگین نیست (طبق محتوای context یا local storage)، به لاگین useNavigate شود.

همچنین در صورت لاگین بودن کاربر، عکس این رخ داده و با وارد کردن صفحه /login کاربر باید به صفحه / هدایت شود.

نحوه تحویل

در Repository خود یک پوشه به نام CA6 ایجاد نمایید و داخل آن کد مربوط به پروژه را قرار دهید.

می‌توانید سوال‌های خودتان را در گروه تلگرامی درس، در بخش پرسش و پاسخ بپرسید.

موفق باشید.