САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронд-энд разработка

Отчет

Лабораторная работа 2

Выполнили:

Егоров Мичил
Группа К33401

Кривошапкина Айталина
Группа К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2021 г.

Задача

Привязать сайт к внешнему API средствами fetch/axios/xhr.

Проект: Сервис доставки/заказа товаров для жителей общежитий

В Delmitary клиенты смогут договориться с жителями общежитий доставить товар/еду из ближайших магазинов/ресторанов за символическую плату прямо до двери комнаты

Есть несколько сущностей: курьер, заказчик, администратор сайта. Заказчики имеют доступ к списку магазинов и их товаров, могут добавлять их в корзину. После набора корзины клиент оставляет запрос.

Курьерам предоставляется доступ к списку заказов с именами, телефонами, корзинами и их описаниями, где они смогут решить какой заказ брать или не брать.

Наполнением и актуализацией информации занимаются администраторы сервиса.

У всех ролей есть возможность редактировать личную информацию. Также у курьеров есть график работы (смены) доставки товаров. Администратор может уточнять и редактировать информацию курьеров/клиентов.

Ход работы:

Серверная часть приложения была реализована с помощью Python/Django с использованием RESTFullAPI и выложена в хостинг по адресу 172.28.121.168:8000.

Напишем класс, который будет отображать страницу товаров (Листинг 1). Сначала получим данные магазина с помощью метода **getShopInfo** и отобразим на странице. Далее нужно получить товары и добавить пагинацию **getNextGoods**. Далее добавим метод, который будет инициализировать страницу при первичном запуске. Привяжем к кнопке onClick метод, который просто будет вызывать метод **getNextGoods**. Результат работы продемонстрирован на рисунке 1.

Листинг 1. Пример кода из файла goods.js

```
var BASE_URL = 'http://172.28.121.163:8000/api/';


class GoodParser {
    constructor(shop_id) {
        this.shop_id = shop_id
        this.setInitialUrl();
    }

    // @filters - строка вида param1=val1&param2=val2
    setInitialUrl(filters = '') {
        let fitler_str = filters ? `&${filters}` : ''
        this.next_url = `${BASE_URL}good/?shop=${this.shop_id}${fitler_str}`;
    }

    async getShopInfo() {
        return await fetch(`${BASE_URL}shop/${this.shop_id}/`)
            .then(function(response) {
                return response.json();
            })
    }

    async getNextGoods() {
        let response = await fetch(this.next_url)
            .then(function(response) {
                return response.json();
            });

        if (response['next']) {
            this.next_url = response['next'];
        }

        return response;
    }

    async setShopInfo() {
        let response = await this.getShopInfo()
        document.getElementById('shop-address').innerHTML = response['address'];
        document.getElementById('shop-mean-price').innerHTML = response['mean_price'] + '₽';
        document.getElementById('shop-walk-duration').innerHTML = response['walk_time'] + '
минут';
        document.getElementById('shop-bike-duration').innerHTML = response['bike_time'] + '
минут';
    }

    async setNextGoods() {
        let responseJson = await this.getNextGoods();
        let response = responseJson['results']
        let goodListDiv = document.getElementById("goods-list");

        for (let i = 0; i < response.length; i++) {
            let good = response[i];
            let container = document.createElement('div');

            container.className = 'card good-card col-mb-12';
            container.innerHTML = `
                <div class="row g-0">
```

```
                    <div class="col-md-4">
                        <img src="${good['icon']}" class="img-fluid rounded-start"
alt="${good['name']}" height="120px">
                    </div>
                    <div class="col-md-8">
                        <div class="card-body">
                            <h5 class="card-title">${good['name']}</h5>
                            <p class="card-text">
                                ${good['description']}
                            </p>
                            <p class="card-text"><small
class="text-muted">${good['price']}₽</small></p>
                        </div>
                    </div>
                </div>
            `;

            goodListDiv.appendChild(container);
        }
    }

    async setPage() {
        this.setShopInfo();
        this.setNextGoods();
    }
}
```

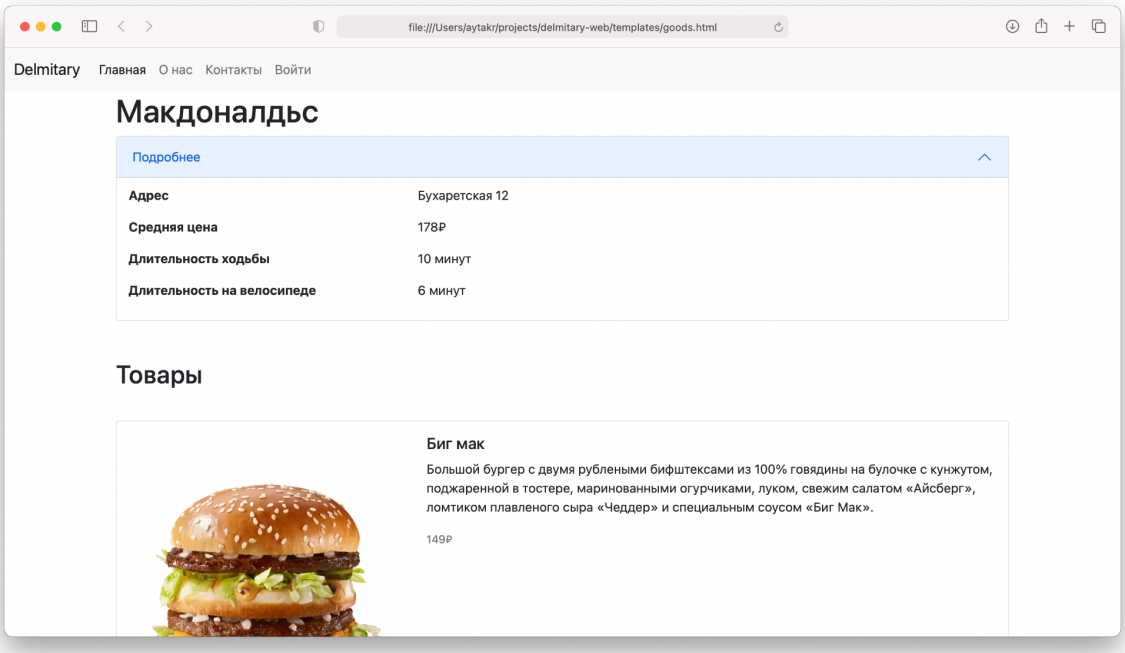

***Рисунок 1**. Страница товаров index.html.*

После аутентификации клиенту отправляется токен и он сохраняется в **sessionStorage** (Рис. 2-3).

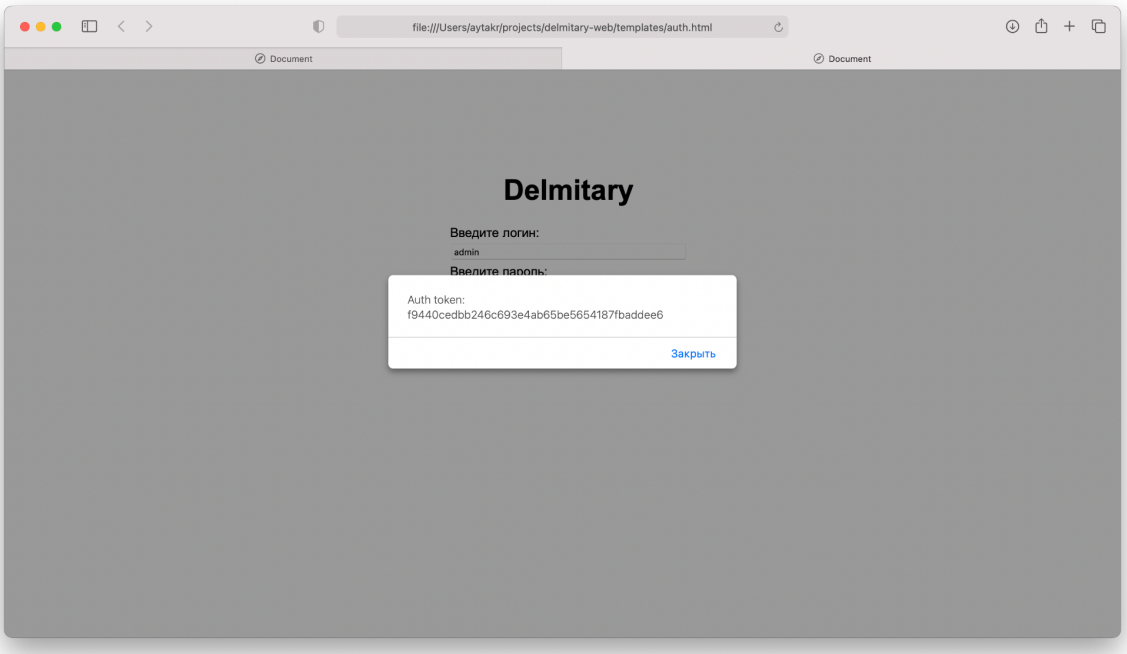

Рисунок 2. *Пример успешной аутентификации auth.html.*

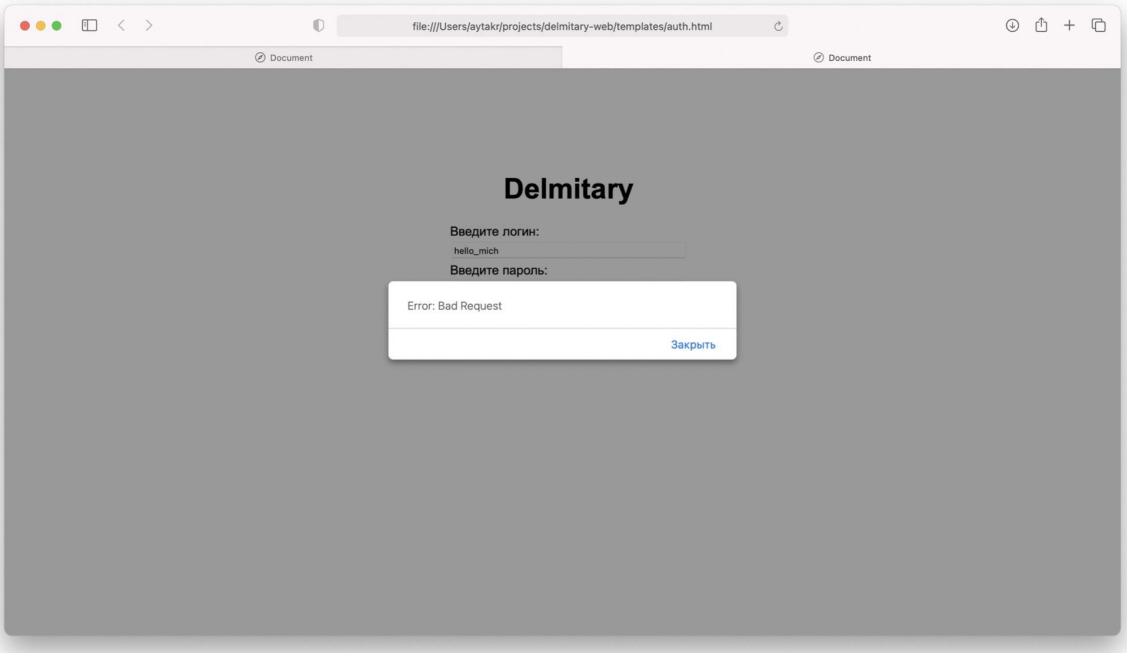

Рисунок 3. *Пример неуспешной аутентификации auth.html.*

Реализация отображения страницы магазинов аналогична странице товаров (Рис. 4).

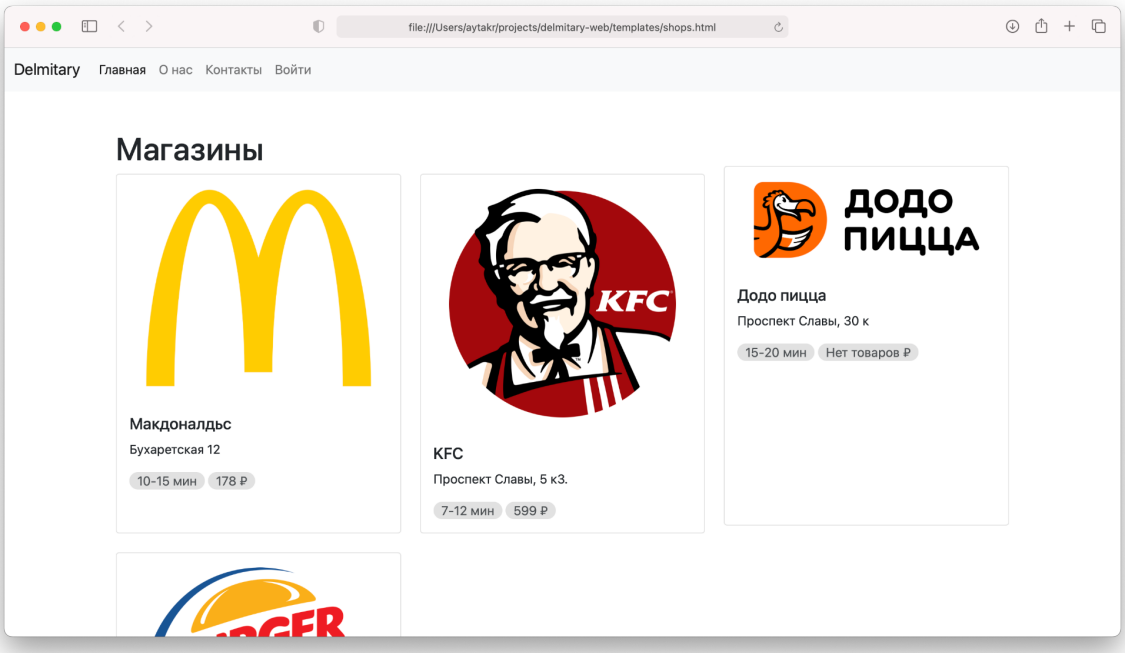

Рисунок 3. *Страница магазинов.*

Вывод:

В ходе данной работы была реализована клиентская сторона, которая динамически подгружает контент из серверной стороны с помощью await/fetch. Была реализована аутентификация и отображение страниц магазинов и товаров.