

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Таначев Егор

Группа К33412

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Привязать проект из Лабораторной работы №1 к внешнему API средствами fetch/axios/xhr.

Ход работы

Для лабораторной работы я использовал файл data.json, который лежит на моем рабочем репозитории в GitHub по ссылке <https://egota1n.github.io/DressImpress/data.json>. В этом файле содержится информация о пользователях и товара (id, наименование, цена, категория, ссылка на фотографию), как показано на Рисунке 1.

```
1  {
2    "users": [
3      {
4        "id": 1,
5        "email": "admin@admin.ru",
6        "password": "admin123"
7      }
8    ],
9    "products": [
10   {
11     "id": 1,
12     "image": "assets/img/cards/card1.jpg",
13     "title": "Худи оливкового цвета",
14     "price": "7.000 ₽",
15     "category": "Худи"
16   },
17   {
18     "id": 2,
19     "image": "assets/img/cards/card2.jpg",
20     "title": "Пальто свободного стиля",
21     "price": "15.000 ₽",
22     "category": "Пальто"
23   },
24   {
25     "id": 3,
26     "image": "assets/img/cards/card3.jpg",
27     "title": "Футболка зеленого цвета",
28     "price": "2.500 ₽",
29     "category": "Футболки"
30   },
31 }
```

Рисунок 1 – Файл data.json

Я реализовал несколько скриптов, которые выводят карточки на страницу с помощью fetch. Скрипт генерирует html-код и заполняет содержимое тегов из файла data.json Скриншоты со страницей и кодом для каталога представлено на Рисунках 2-3. Для вывода карточек похожих товаров на Рисунках 4-5.

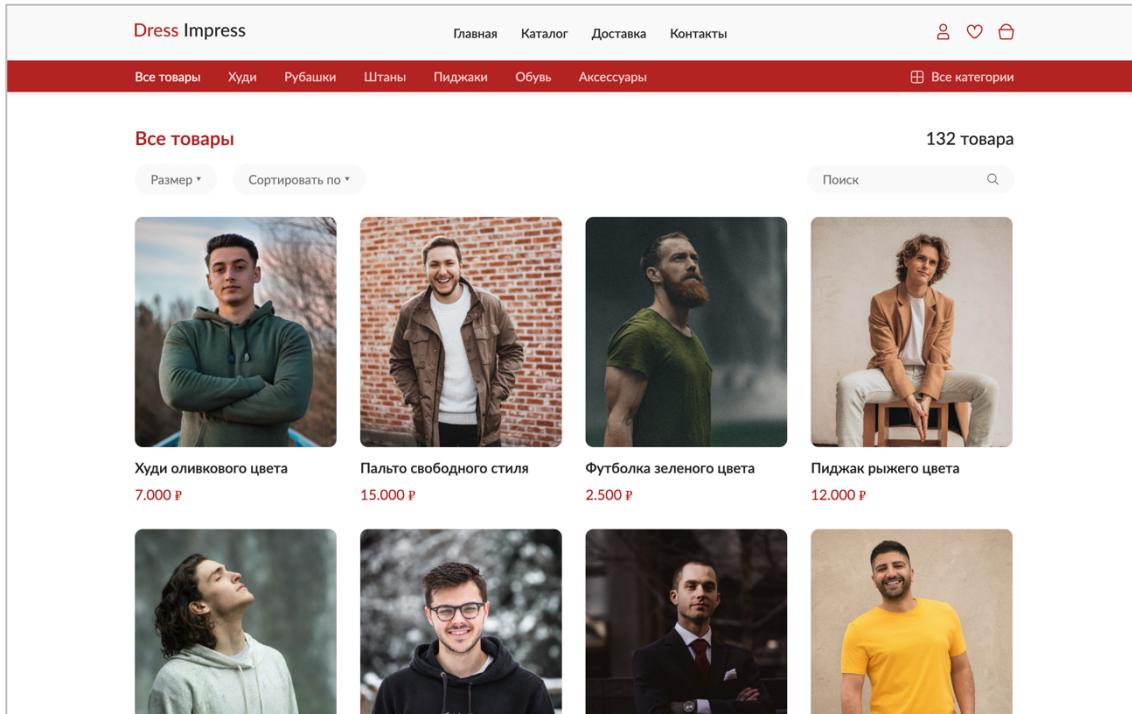


Рисунок 2 – Страница с каталогом

```

60 document.addEventListener('DOMContentLoaded', function () {
61     const catalogElement = document.getElementById('catalog');
62
63     // ФУНКЦИЯ ДЛЯ ЗАГРУЗКИ ДАННЫХ ИЗ data.json
64     function loadData() {
65         fetch('https://egota1n.github.io/DressImpress/data.json')
66             .then((response) => response.json())
67             .then((data) => {
68                 // ГЕНЕРИРУЕМ КАРТОЧКИ ТОВАРОВ ИЗ ДАННЫХ
69                 data.products.forEach((item) => {
70                     const card = document.createElement('div');
71                     card.classList.add('section__catalog-card');
72
73                     const img = document.createElement('img');
74                     img.classList.add('section__catalog-card-img');
75                     img.src = item.image;
76
77                     const title = document.createElement('h3');
78                     title.classList.add('section__catalog-card-title');
79                     title.textContent = item.title;

```

```

80
81
82
83
84
85
86
87
88
89
90
91
92

```

```

        const price = document.createElement('p');
        price.classList.add('section__catalog-card-price');
        price.textContent = item.price;

        card.appendChild(img);
        card.appendChild(title);
        card.appendChild(price);

        catalogElement.appendChild(card);
    );
}

.catch(error) => {

```

Рисунок 3 – Скрипт для генерации карточек

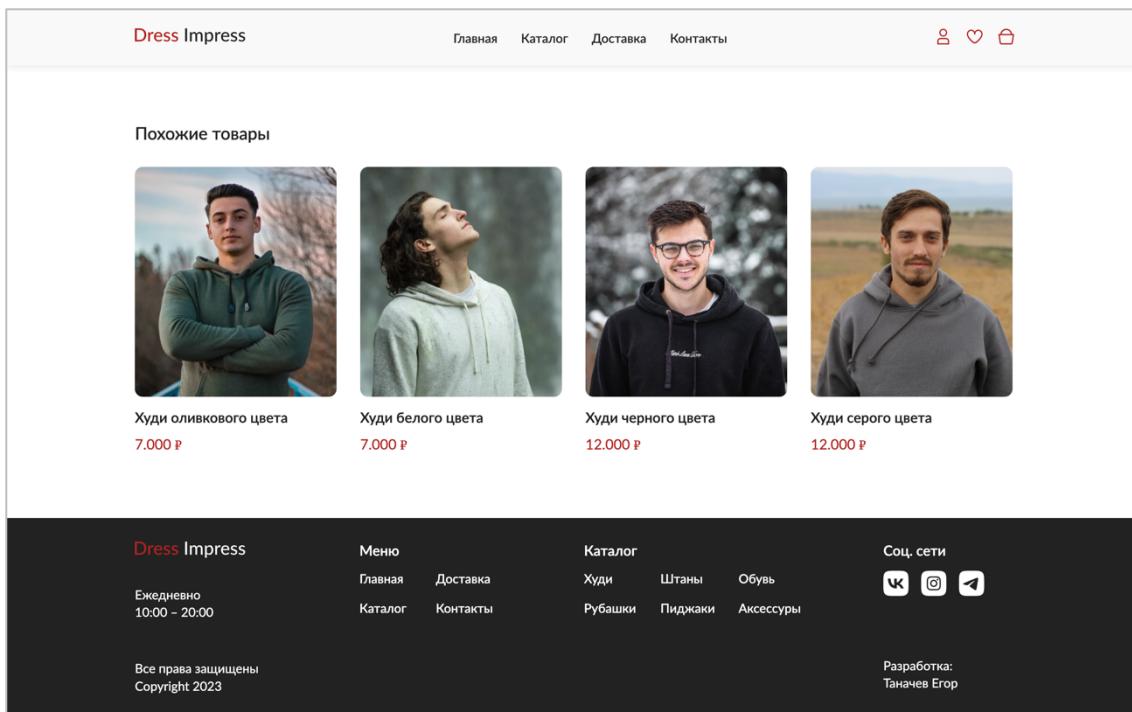


Рисунок 4 – Похожие товары

```

1 // ВЫВОД ПОХОЖИХ ТОВАРОВ В КАТАЛОГ
2
3 document.addEventListener('DOMContentLoaded', function () {
4     const catalogElement = document.getElementById('catalog');
5
6     // Функция для загрузки данных из data.json
7     function loadData() {
8         fetch('https://egota1n.github.io/DressImpress/data.json')
9             .then((response) => response.json())
10            .then((data) => {
11                let hoodieCount = 0; // Переменная для отслеживания количества худи
12                const maxHoodies = 4; // Максимальное количество
13
14                // Генерируем карточки товаров из данных
15                data.products.forEach((item) => {
16                    if (item.category === 'Худи' && hoodieCount < maxHoodies) {
17                        const card = document.createElement('div');
18                        card.classList.add('section__catalog-card');
19
20                        const img = document.createElement('img');
21                        img.classList.add('section__catalog-card-img');
22                        img.src = item.image;
23
24                        const title = document.createElement('h3');
25                        title.classList.add('section__catalog-card-title');
26                        title.textContent = item.title;
27
28                        const price = document.createElement('p');
29                        price.classList.add('section__catalog-card-price');
30                        price.textContent = item.price;
31
32                        card.appendChild(img);
33                        card.appendChild(title);

```

Рисунок 5 – Скрипт для добавления похожих карточек

Также я реализовал скрипт, который сравнивает содержимое input с данными пользователя из data.json и осуществляет переход в личный кабинет, как показано на Рисунках 6-9.

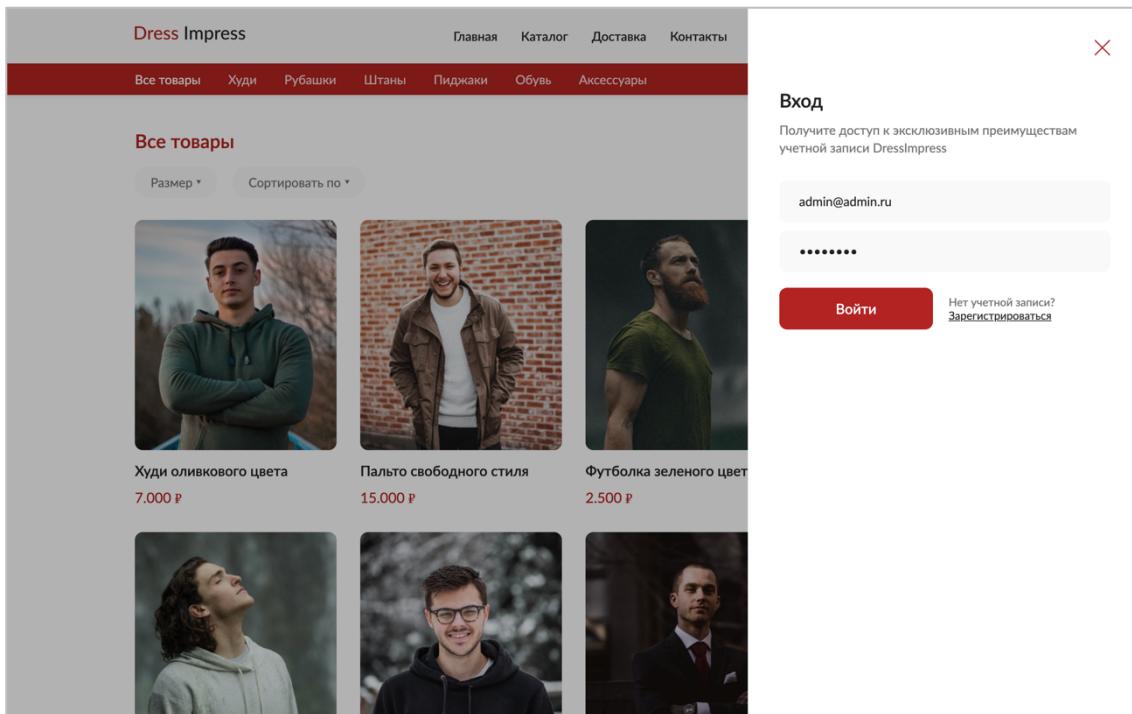


Рисунок 6 – Модальное окно с авторизацией

The screenshot shows the 'Личный кабинет' (Personal Cabinet) page. At the top, it displays the user's name 'Егор Таначев', phone number '+7 (991) 812 07 77', and email 'e.tanachev@inbox.ru'. Below this, there are sections for 'Адрес доставки:' (Delivery address), 'Способ оплаты:' (Payment method), and a map showing the delivery area. The map highlights several landmarks in St. Petersburg, including the Hermitage Museum, the Winter Palace, and the Nevsky Prospekt. At the bottom, there is a footer with links for 'Dress Impress', 'Меню' (Menu), 'Каталог' (Catalog), 'Соц. сети' (Social networks), and various delivery and contact options.

Рисунок 7 – Личный кабинет

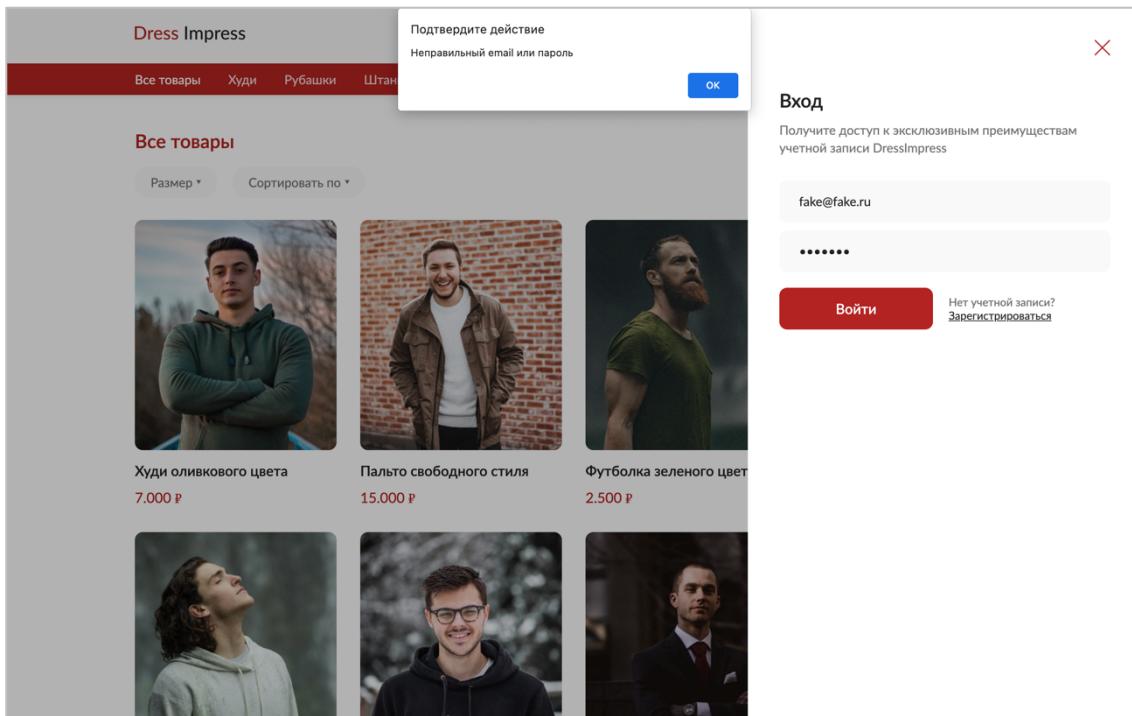


Рисунок 8 – Ошибка авторизации

```

58 // АВТОРИЗАЦИЯ
59
60 // Получаем форму и её элементы
61 const form = document.getElementById('accountForm');
62 const emailInput = form.querySelector('input[type="email"]');
63 const passwordInput = form.querySelector('input[type="password"]');
64
65 // Обработка события отправки формы
66 form.addEventListener('submit', (e) => {
67 e.preventDefault(); // Предотвращаем отправку формы по умолчанию
68
69 // Получаем данные из JSON
70 fetch('https://egotaInn.github.io/DressImpress/data.json')
71   .then((response) => response.json())
72   .then((data) => {
73     const users = data.users;
74
75     // Проверяем данные из формы с данными из JSON
76     const user = users.find(user => user.email === emailInput.value && user.password === passwordInput.value);
77
78     if (user) {
79       // Если данные совпадают, переходим на страницу product.js
80       window.location.href = 'account.html';
81     } else {
82       alert('Неправильный email или пароль');
83     }
84   })
85   .catch((error) => {
86     console.error('Ошибка при загрузке данных из JSON', error);
87   });
88 });

```

Рисунок 9 – Скрипт с авторизацией

Вывод

В результате выполнения лабораторной работы было успешно решено основное задание - привязка проекта из Лабораторной работы №1 к внешнему API с использованием средств fetch/axios/xhr. В данной работе был использован файл data.json, размещенный в моем репозитории на GitHub