

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнила:

Жигалова Анастасия

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задание: Привязать то, что мы делали в ЛР1 к внешнему API средствами fetch/axios/xhr.

Магазин одежды: Разработка интернет-магазина одежды, включающего функции добавления товаров в корзину, оформления заказа и фильтрации по категориям.

Ход работы:

Я использовала базу данных – pocketbase.

```
C:\Windows\system32>cd C:\Users\79045\Downloads\pocketbase_0.19.0_windows_amd64

C:\Users\79045\Downloads\pocketbase_0.19.0_windows_amd64>pocketbase.exe serve
2023/10/23 11:43:07 Server started at http://127.0.0.1:8090
└─ REST API: http://127.0.0.1:8090/api/
└─ Admin UI: http://127.0.0.1:8090/_/
```

Создала бд с каталогом.

Edit collection ...

Name * catalog Type: Base

Fields API Rules

System fields: id, created, updated.

T	title	⚙
🔗	imageUrl	⚙
#	price	⚙
T	cardName	⚙

+ New field

Collections / catalog ⚙️ 🔄 </> API Preview + New record

🔍 Search term or filter like created > "2022-01-01"...

<input type="checkbox"/>	id	T title	@ imageUrl	...
<input type="checkbox"/>	sbewsnjq31to6z	Футболка	https://ae01.alicdn.com/kf/H93a9b9274d514efcafe587f49f0f8634C.jpg_300x300Q70.jpg__webp	→
<input type="checkbox"/>	3biqurprkfmr3q	Спортивный костюм	http://www.2buylist.ru/images/product/s/3519353b6c.jpg	→
<input type="checkbox"/>	tm2hergbq94i	Свитер	https://ae01.alicdn.com/kf/H33c374b9a7b84855851746b6388ab455/-_jpg_300x300.jpg	→
<input type="checkbox"/>	v4fe1wwabs40z	Пальто	https://vyatkakirov.ru/wp-content/uploads/2022/10/htb1e96dxjnuk1rksmfng6auzfvad-300x300.webp	→
<input type="checkbox"/>	56lqhghcvyzz24	Куртка	https://i.pinimg.com/736x/d2/18/9c/d2189c5d1163a01208a72274c32b03ba.jpg	→
<input type="checkbox"/>	ex87c64a2y38sci	Кепка	https://vftys.ru/img/content/main/men_caps_300x300.jpg	→
<input type="checkbox"/>	th1h7jsozoiuwt	Брюки	https://gaitaobao3.alicdn.com/tfscom/i2/2508948728/TB2erupzblmpuFjSZFIXXbdQXXa_112508948728.jpg_300x300q20.jpg	→

Создала бд для добавленных в корзину товаров из каталога.

Edit collection ...

Name *
cart Type: Base

Fields API Rules

System fields: `id`, `created`, `updated`.

T title ⚙️

@ imageUrl ⚙️

price ⚙️

+ New field

Collections / cart ⚙️ 🔄 </> API Preview + New record

🔍 Search term or filter like created > "2022-01-01"...

<input type="checkbox"/>	id	T title	@ imageUrl	# price	📅 created	...
<input type="checkbox"/>	v2o1bnbhp2ajj	Кепка	https://i.ebayimg.com/thumbs/images/g/i0pAAOSwFPBfaaoC/s:1300.jpg	1000	2023-10-23 19:16:33 UTC	→
<input type="checkbox"/>	3kvc2yqr581ecq	Юбка	https://www.dhresource.com/300x300/f2/albu/p14/M01/83/49/rBajF9KChGAHFByAMCuzy1kp1E856.jpg	2300	2023-10-23 15:52:11 UTC	→

Используя fetch, получила данные с бд на страницу catalog (метод GET), с помощью POST добавляла элементы одежды из каталога в бд cart.

```
document.addEventListener('DOMContentLoaded', function(){
  fetch('http://localhost:8090/api/collections/catalog/records')
    .then(response => response.json())
    .then(data => {

      data.items.forEach(item => {
        let column = document.createElement("div")
        column.className = "col-md-3"

        let card = document.createElement("div")
        card.className = item.cardName

        column.appendChild(card)

        let itemImage = document.createElement("img")
        itemImage.src = item.imageUrl

        card.appendChild(itemImage)

        let cardBody = document.createElement("div")
        cardBody.className = "card-body"

        let cardTitle = document.createElement("h5")
        cardTitle.className = "card-title"
        cardTitle.textContent = item.title
        cardBody.appendChild(cardTitle)

        let cardText = document.createElement("p")
        cardText.className = "card-text"
        cardText.textContent = "Цена: " + item.price + " руб."
        cardBody.appendChild(cardText)

        let addToCartLink = document.createElement("a")
        addToCartLink.href = "cart.html"
        addToCartLink.className = "btn btn-primary"
        addToCartLink.textContent = "Добавить в корзину"
        addToCartLink.onclick = function() {
          fetch('http://localhost:8090/api/collections/cart/records', {
            method: 'POST',
            headers: {
              'Content-Type': 'application/json'
            }
          })
        }
      })
    })
})
```

```

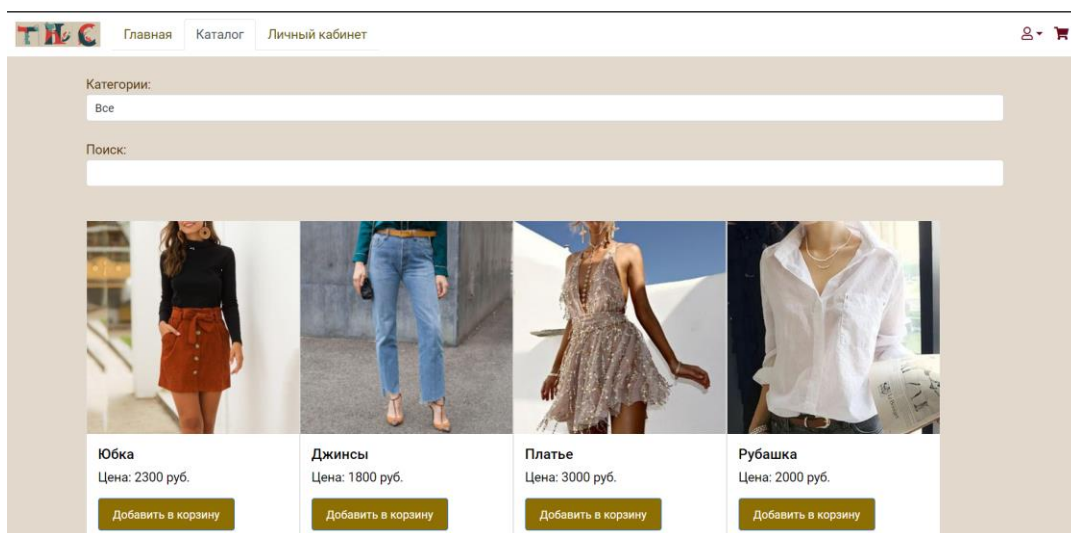
    },
    body: JSON.stringify(item)
  })
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.log(error));
}
cardBody.appendChild(addToCartLink)

card.appendChild(cardBody)

document.querySelector(".items-cards").appendChild(column)

  })
})
});

```



В корзине товары можно удалять (метод DELETE)

```

document.addEventListener('DOMContentLoaded', function(){
  fetch('http://localhost:8090/api/collections/cart/records')
    .then(response => response.json())
    .then(data => {

      data.items.forEach(item => {
        let column = document.createElement("div")
        column.className = "col-md-3"

        let card = document.createElement("div")
        card.className = "card"

```

```
column.appendChild(card)

let itemImage = document.createElement("img")
itemImage.src = item.imageUrl

card.appendChild(itemImage)

let cardBody = document.createElement("div")
cardBody.className = "card-body"

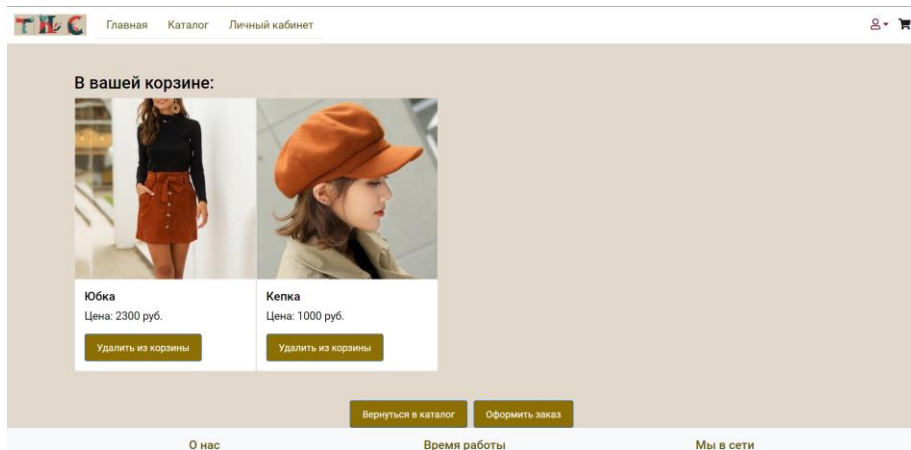
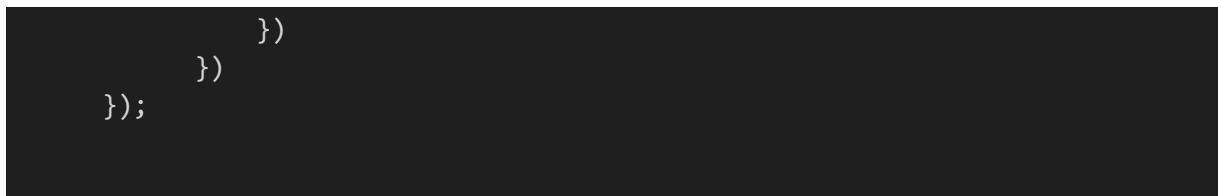
let cardTitle = document.createElement("h5")
cardTitle.className = "card-title"
cardTitle.textContent = item.title
cardBody.appendChild(cardTitle)

let cardText = document.createElement("p")
cardText.className = "card-text"
cardText.textContent = "Цена: " + item.price + " руб."
cardBody.appendChild(cardText)

let addToCartLink = document.createElement("a")
addToCartLink.href = "cart.html"
addToCartLink.className = "btn btn-primary"
addToCartLink.textContent = "Удалить из корзины"
addToCartLink.onclick = function() {
    fetch(`http://localhost:8090/api/collections/cart/records/
${item.id}`, {
        method: 'DELETE',
    })
    .then(response => response.json())
    .then(data => console.log(data))
    .catch(error => console.log(error));
}
cardBody.appendChild(addToCartLink)

card.appendChild(cardBody)

document.querySelector(".items-cards").appendChild(column)
```



Создала бд с пользователями.

Edit collection

Name *

users

Type: Base

Fields

API Rules

System fields: `id`, `created`, `updated`.

✉ email

Nonempty

⚙

T password



Nonempty

⚙

+ New field

Unique constraints and indexes (0)

+ New index

Collections / users   API Preview + New record

Search term or filter like created > "2022-01-01"...

<input type="checkbox"/>	id	email	password	created	updated	...
<input type="checkbox"/>	ttup5mufnvs080	12@gmail.com	1111	2023-10-23 11:00:05 UTC	2023-10-23 11:00:05 UTC	→
<input type="checkbox"/>	xze1qelyp64ujz	12@gmail.com	111111	2023-10-23 11:27:46 UTC	2023-10-23 11:27:46 UTC	→
<input type="checkbox"/>	wl0pkb9wiaw9c	77777@gmail.com	111111	2023-10-23 11:44:58 UTC	2023-10-23 11:44:58 UTC	→
<input type="checkbox"/>	pcoowgb7m0v2ju	77777@gmail.com	1111111	2023-10-23 11:45:26 UTC	2023-10-23 11:45:26 UTC	→
<input type="checkbox"/>	cknfzyuatq4pp2	77777@gmail.com	111111	2023-10-23 11:45:43 UTC	2023-10-23 11:45:43 UTC	→
<input type="checkbox"/>	9wipfthvugrnpqe	aaaaa@gmail.com	12345	2023-10-23 10:55:19 UTC	2023-10-23 10:55:19 UTC	→
<input type="checkbox"/>	zjbe5yrfyfrxt	mmm@gmail.com	123	2023-10-23 10:57:55 UTC	2023-10-23 10:57:55 UTC	→
<input type="checkbox"/>	2eyhshb2dmsp6m	mmm@gmail.com	11111111111	2023-10-23 11:41:20 UTC	2023-10-23 11:41:20 UTC	→

Записывала новых пользователей, которые регистрировались (метод POST), проверяла есть ли уже пользователь в базе (метод GET).

```
form = document.getElementById('registrationForm');
form.addEventListener('submit', function(e) {
    e.preventDefault();

    const email = document.getElementById('exampleInputEmail1').value;
    const password = document.getElementById('exampleInputPassword1').value;

    fetch('http://localhost:8090/api/collections/users/records', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({
            email: email,
            password: password
        })
    })

    fetch('http://localhost:8090/api/collections/users/records')
    .then(response => response.json())
    .then(data => {
        const users = data.items;
        const existingUser = users.find(user => user.email === email);
        if (existingUser) {
            alert("Вы уже зарегистрированы");
        } else {
            document.getElementById('welcomeMessage').style.display = 'block';
            document.getElementById('userEmail').innerText = email;
            document.getElementById('registrationForm').style.display =
'none';
        }
    })
}
```



```

        document.getElementById('registrationPrompt').style.display =
'none';
    }
    })
});

```

Проверяла наличие пользователей в бд, которые хотят войти в личный кабинет (метод GET).

```

document.querySelector('form').addEventListener('submit', function(event) {
    event.preventDefault();

    const email = document.getElementById('exampleInputEmail1').value;
    const password = document.getElementById('exampleInputPassword1').value;

    fetch('http://localhost:8090/api/collections/users/records')
        .then(response => response.json())
        .then(data => {
            const users = data.items;
            const user = users.find(user => user.email === email && user.password
=== password);

            if (user) {
                document.getElementById('welcomeMessage').style.display = 'block';
                document.getElementById('userEmail').innerText = email;
                document.getElementById('entaranaceForm').style.display = 'none';
                document.getElementById('entaranacePrompt').style.display = 'none';
            } else if (users.some(user => user.email === email)) {
                alert('Неверный пароль');
            } else {
                alert('Сначала пройдите регистрацию');
            }
        })
        .catch(error => {
            console.error('Ошибка:', error);
        });
});

```

Пользователь проходит регистрацию и записывается в бд.

Пройти регистрацию

Почта

Пароль

Зарегистрироваться

Добро пожаловать!
lid@gmail.com

Если почта уже есть в бд, пользователь увидит модальное окно.

Личный кабинет

Подтвердите действие
Вы уже зарегистрированы

OK

Почта

Пароль

Зарегистрироваться

Если введен неправильный пароль, пользователь увидит модальное окно с этой информацией.

Личный кабинет

Подтвердите действие
Неверный пароль

OK

Почта

Пароль

Войти

Добро пожаловать!

ssss@gmail.com

Вывод:

В ходе лабораторной работы научилась взаимодействию с внешним API, используя базу данных rocketbase.