

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

**Отчет**

**Лабораторная работа 2**

**Выполнил:**

**Старовойтова Елизавета**

**Группа:**

**К33402**

**Проверил:**

**Добряков Д. И.**

**Санкт-Петербург**

**2023 г.**

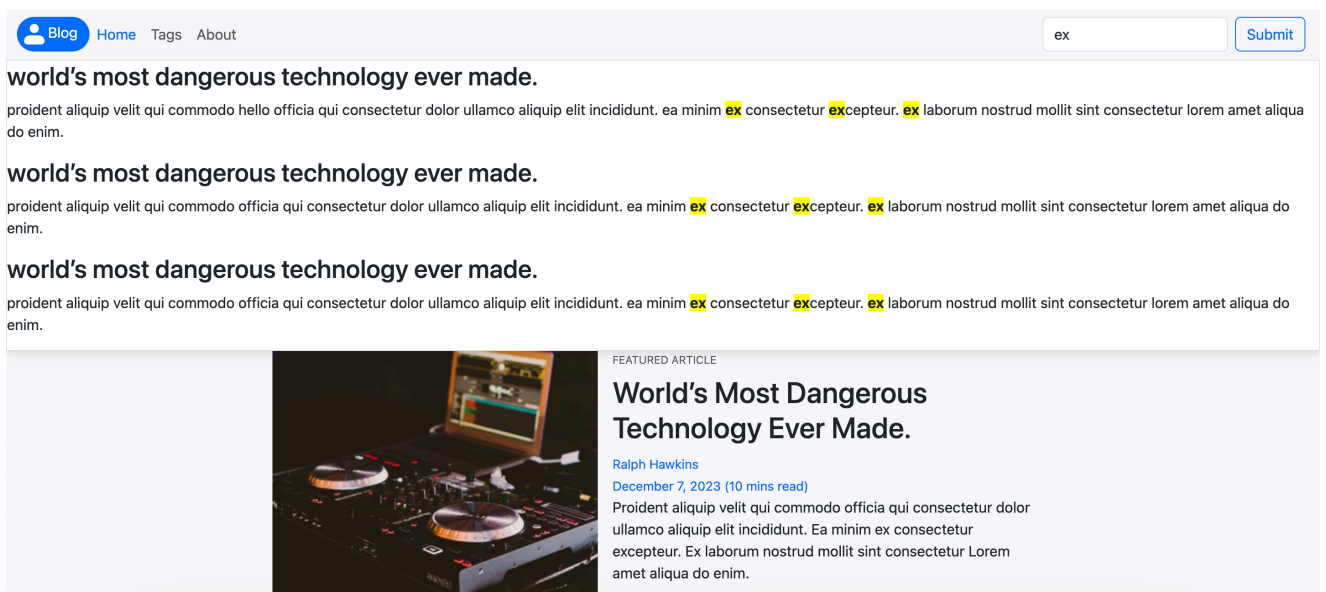
## Задача

Взаимодействие с внешним API. Нужно реализовать имитацию API и запланированный функционал.

## Ход работы

### 1 страница (home page)

В меню есть строка поиска, элемент для перехода в профиль (если пользователь зарегистрирован) и ссылки на вкладки tags, about. На главной странице можно посмотреть последние статьи либо узнать о тех, которые скоро будут публиковаться. Добавлена реализация поиска.



```
function searchArticles(event) {
  event.preventDefault();

  const searchInput = document.getElementById('searchInput').value.toLowerCase();
  const articles = document.querySelectorAll('.editors-picks-card');

  const searchResults = [];

  articles.forEach(article => {
    const title = article.querySelector('.second-section-article-h2').textContent.toLowerCase();
    const text = article.querySelector('.second-section-article-textp').textContent.toLowerCase();

    if (title.includes(searchInput) || text.includes(searchInput)) {
      searchResults.push({
        title: title,
        text: text,
      });
    }
  });

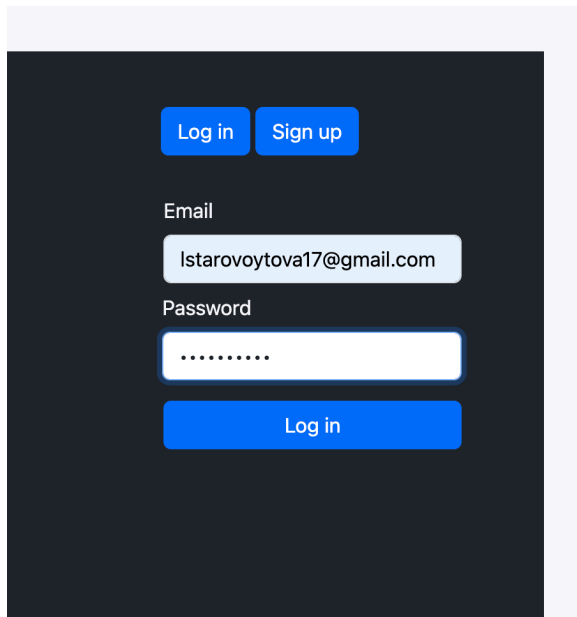
  displaySearchResults(searchResults, searchInput);
}

function displaySearchResults(results, searchInput) {
  const searchResultsContainer = document.getElementById('searchResults');
  searchResultsContainer.innerHTML = '';

  if (results.length === 0) {
    searchResultsContainer.innerHTML = 'No results found.';
    return;
  }

  results.forEach(result => {
    const resultItem = document.createElement('div');
    const highlightedText = highlightSearchTerm(result.text, searchInput);
    resultItem.innerHTML = `<h3>${result.title}</h3><p>${highlightedText}</p>`;
  });
}
```

Также реализована авторизация/регистрация.

A dark-themed login and registration form. At the top, there are two blue buttons: "Log in" and "Sign up". Below them, the "Email" field contains the text "Istarovoytova17@gmail.com". The "Password" field is masked with eight dots. At the bottom, there is a blue "Log in" button.

Log in Sign up

Email

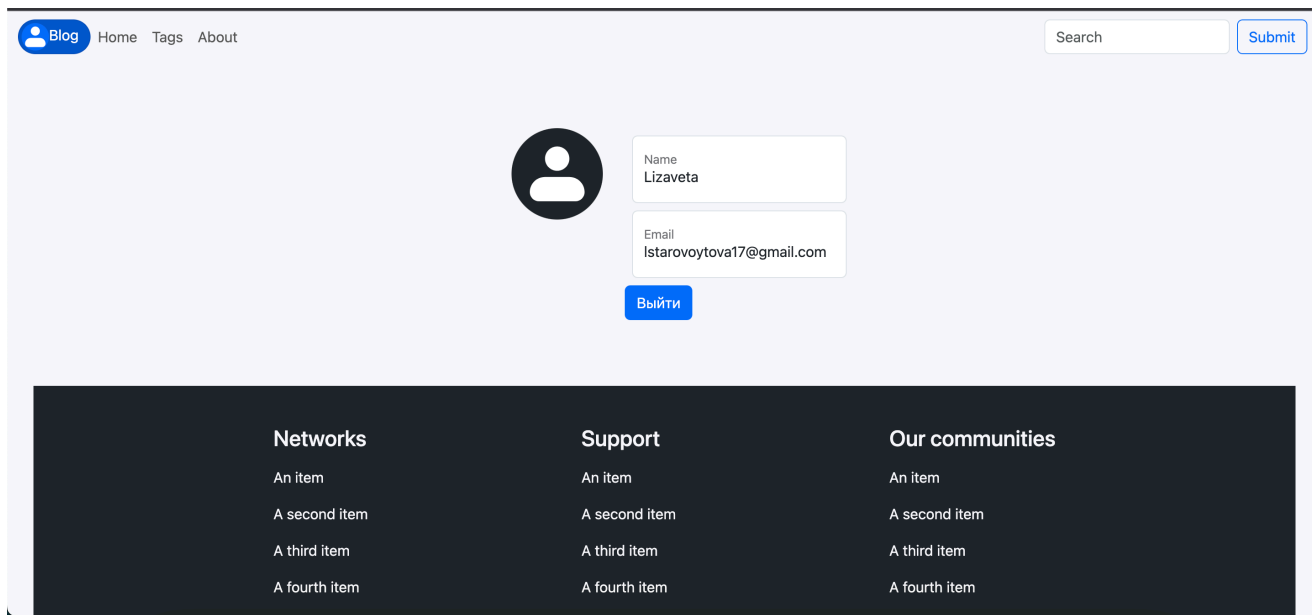
Istarovoytova17@gmail.com

Password

.....

Log in

Заполняется форма, при нажатии на кнопку, если вход выполнен, то переход на страницу Профиля, где высвечиваются данные о пользователе.

A user profile page with a light purple background. The top navigation bar includes a "Blog" button with a user icon, and links for "Home", "Tags", and "About". A search bar with the text "Search" and a blue "Submit" button is on the right. The main content area features a circular profile picture placeholder, a "Name" field with the value "Lizaveta", an "Email" field with the value "Istarovoytova17@gmail.com", and a blue "Выйти" (Logout) button. The footer is a dark blue section with three columns: "Networks", "Support", and "Our communities". Each column contains a list of four items, each labeled "An item", "A second item", "A third item", and "A fourth item".

Blog Home Tags About

Search Submit

Name  
Lizaveta

Email  
Istarovoytova17@gmail.com

Выйти

**Networks**

- An item
- A second item
- A third item
- A fourth item

**Support**

- An item
- A second item
- A third item
- A fourth item

**Our communities**

- An item
- A second item
- A third item
- A fourth item

Для обновления данных на странице с Профилем.

```
index.html # index.css JS index.js # tags.css JS tags.js JS profile.js JS server.js {} db.json
K33402 > Старовойтова Елизавета > lab2 > js > JS profile.js > updateProfileView
1 function updateProfileView() {
2   // Извлекаем параметры из URL
3   const params = new URLSearchParams(window.location.search);
4   const name = params.get('name');
5   const email = params.get('email');
6
7   // Получаем элементы DOM, которые нужно обновить
8   const nameInput = document.getElementById('nameInput');
9   const emailInput = document.getElementById('emailInput');
10
11   // Обновляем значения только если они определены (не null)
12   if (name !== null && email !== null) {
13     nameInput.value = name;
14     emailInput.value = email;
15
16     // Сохраняем данные в localStorage
17     localStorage.setItem('profileName', name);
18     localStorage.setItem('profileEmail', email);
19   }
20 }
21
22 document.addEventListener('DOMContentLoaded', updateProfileView);
23
24 // При загрузке страницы извлекаем данные из localStorage
25 document.addEventListener('DOMContentLoaded', function () {
26   const storedName = localStorage.getItem('profileName');
27   const storedEmail = localStorage.getItem('profileEmail');
28
29   // Получаем элементы DOM
30   const nameInput = document.getElementById('nameInput');
31   const emailInput = document.getElementById('emailInput');
32
33   // Обновляем значения только если они определены в localStorage (не null)
34   if (storedName !== null && storedEmail !== null) {
35     nameInput.value = storedName;
36     emailInput.value = storedEmail;
37   }
38 });
39
40 function logout() {
41   // Очищаем данные в localStorage
42   localStorage.removeItem('profileName');
43   localStorage.removeItem('profileEmail');
44
45   // Перенаправляем на главную страницу
46 }
```

```
402 > Старовойтова Елизавета > lab2 > js > JS index.js > ...
async function login(event) {
  event.preventDefault();

  const email = document.getElementById('inputEmail1').value;
  const password = document.getElementById('inputPassword1').value;

  try {
    const response = await fetch('http://localhost:4000/login', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({ email, password }),
    });

    if (response.ok) {
      const data = await response.json();
      console.log('Login successful!', data);

      localStorage.setItem('token', data.token);
      localStorage.setItem('name', data.name);
      localStorage.setItem('email', data.email);

      // После успешного входа перенаправляем на страницу профиля
      window.location.href = `profile.html?name=${data.name}&email=${data.email}`;
    } else {
      const errorData = await response.json();
      console.error('Login failed:', errorData);
      // Обработка ошибок входа
    }
  } catch (error) {
    console.error('Login failed:', error);
  }
}
```

Server.js для обработки маршрута

```

9 // Обработка маршрута /login
10 server.post('/login', (req, res) => {
11   const { email, password } = req.body;
12   const user = server.db.get('users').find({ email }).value();
13
14   if (user && user.password === password) {
15     const token = jwt.sign({ userId: user.id }, '14ce11d9-a347-41a2-8eef-8a3fa2c652be', { expiresIn: '1h' });
16     res.json({ token, name: user.name, email: user.email });
17   } else {
18     res.status(401).json({ error: 'Unauthorized' });
19   }
20 });
21
22 server.use(auth);
23
24 server.use(router);
25
26 const PORT = 4000;
27 server.listen(PORT, () => {
28   console.log(`JSON Server with Auth is running on port ${PORT}`);
29 });
30
31

```

#### JSON-файл для хранения сведений

33402 > Старовойтова Елизавета > lab2 > {} db.json > [ ] users > {} 5

```

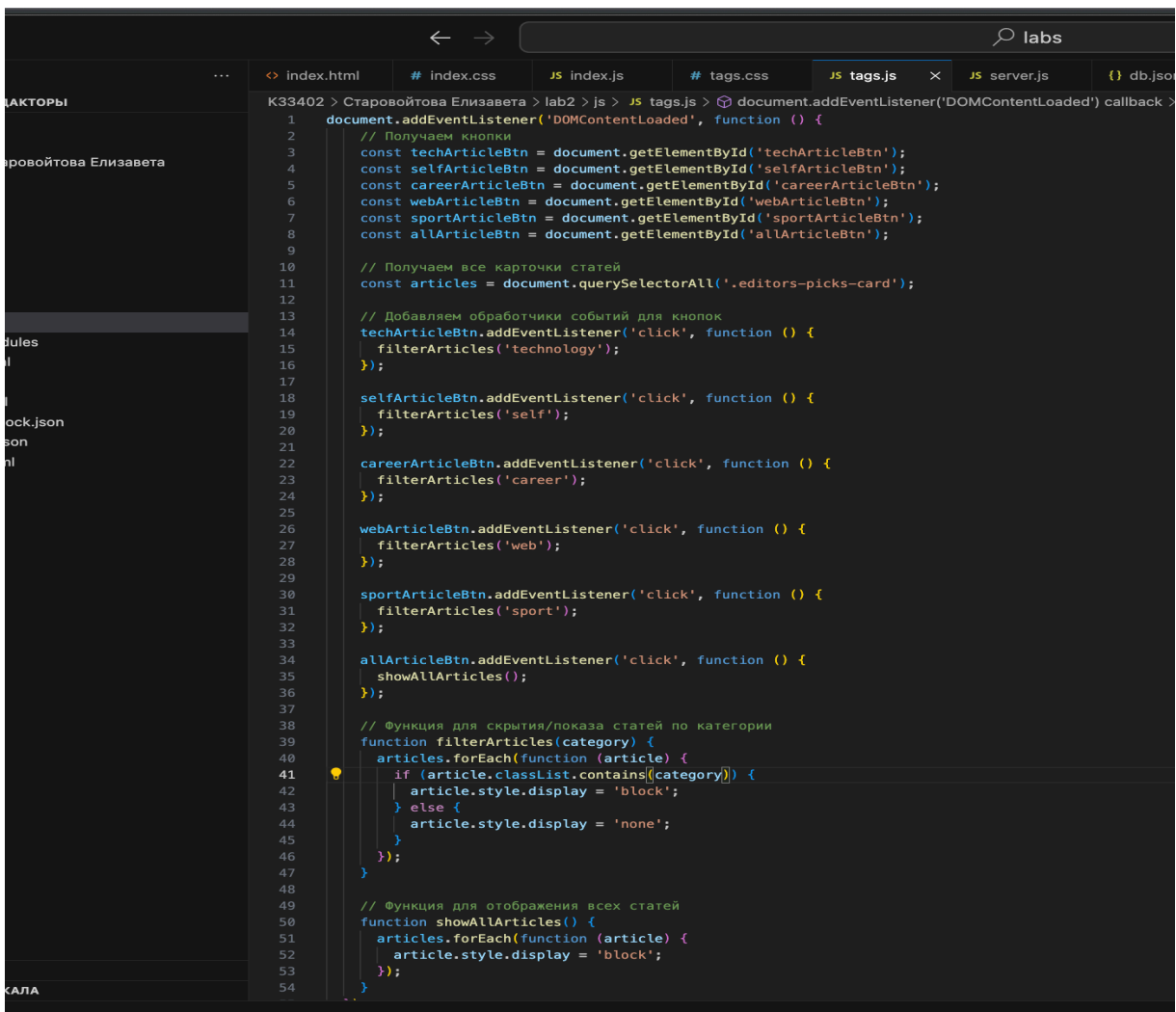
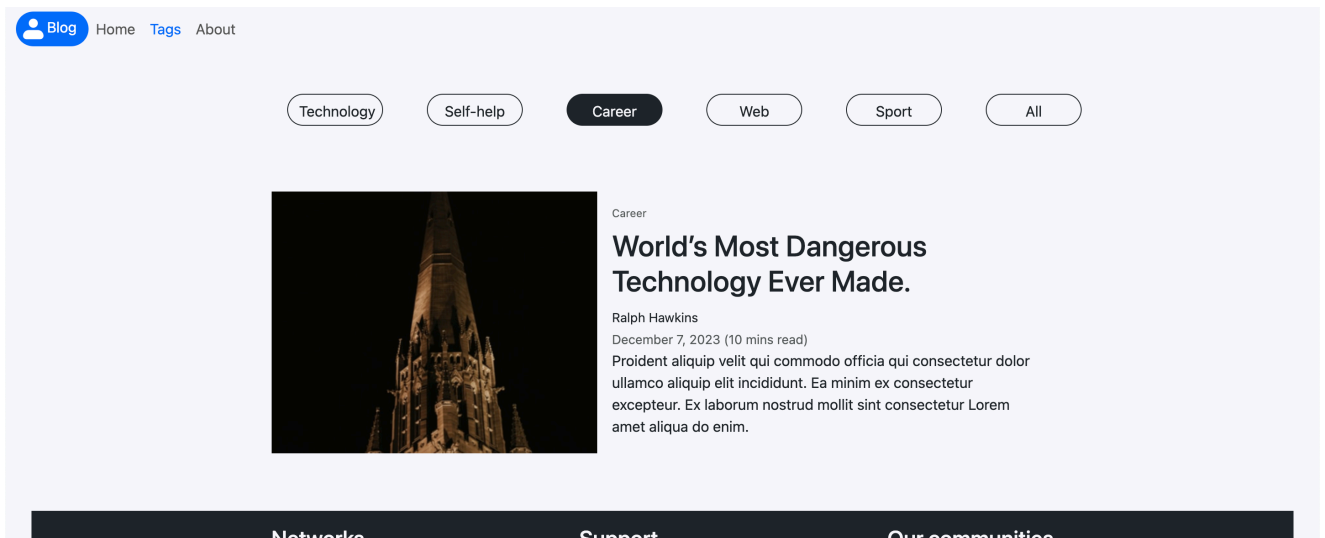
1  {
2    "users": [
3      {
4        "id": 1,
5        "name": "John",
6        "email": "mail@example.com",
7        "password": "1234"
8      },
9      {
10       "id": 1701039268701,
11       "name": "Lizaveta",
12       "email": "lstarovoytova17@gmail.com",
13       "password": "likast1233"
14     },
15     {
16       "id": 1701376625112,
17       "name": "Katerina",
18       "email": "starovoytova12@gmail.com",
19       "password": "betastar12"
20     },
21     {
22       "id": 1701377178759,
23       "name": "Tatsiana",
24       "email": "tata@mail.com",
25       "password": "sycheva123"
26     },
27     {
28       "id": 1701379502243,
29       "name": "Nikita",
30       "email": "korotki@mail.ru",
31       "password": "geniy12"
32     },
33     {
34       "id": 1701385530700,
35       "name": "frida",
36       "email": "shaha@gmail.com",

```

## 2 страница (tags page)

На данной странице можно выбрать тег, по которому хочется посмотреть статьи, все остальное без изменений. При нажатии на определенную кнопку высвечиваются

статьи только по выбранной теме.



## Вывод

В данной лабораторной работе я добавила функциональности приложению и изучила работу с

API и json-серверо