Лабораторна робота №11

Тема: Обмін даними між клієнтом та сервером на основі технології AJAX JavaScript

Мета: Навчитися здійснювати запити для зчитування даних з веб-сервера.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Ключовим елементом AJAX є об'єкт XMLHttpRequest.

- Створення об'єкта XMLHttpRequest
- Визначення функції зворотного виклику
- Відкрийте об'єкт XMLHttpRequest
- Надіслати запит на сервер

Основні характеристики:

- Працює у всіх браузерах.
- Підтримує як GET, так і POST запити.
- Має подієву модель (onreadystatechange, onload, onerror).
- Потрібно вручну перевіряти стан запиту (readyState) і код відповіді (status).

Приклад

```
let xhr = new XMLHttpRequest();
xhr.open("GET", "data.json", true);
xhr.onload = function () {
  if (xhr.status === 200) {
    console.log(JSON.parse(xhr.responseText));
  }
};
xhr.send();
```

Fetch API

fetch — сучасна альтернатива XMLHttpRequest, що базується на Promises.

Основні харктеристики:

- Повертає Promise, що робить код чистішим і зручнішим.
- Легше читати та писати.
- Не потрібно вручну відстежувати readyState.
- Не "ловить" HTTP-помилки як помилки JavaScript — потрібно перевіряти response.ok**.**

Приклад

```
fetch("data.json")
  .then(response => {
    if (!response.ok) {
      throw new Error("HTTP error " + response.status);
    }
    return response.json();
  })
  .then(data => console.log(data))
  .catch(error => console.error("Fetch error:", error));
```

ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

Завдання на XMLHttpRequest

Прості завдання (базовий рівень)

1. Отримати випадковий жарт з API

Зробити AJAX-запит до публічного API (наприклад, https://official-joke-api.appspot.com/random_joke) і вивести жарт на сторінку після натискання кнопки.

2. Завантажити текстовий файл без перезавантаження

Використовуючи XMLHttpRequest, завантажити локальний .txt файл і відобразити його в .

3. Форма з AJAX-обробкою без перезавантаження сторінки

Створити просту HTML-форму (наприклад, введення імені), яка відправляє дані на сервер (можна імітувати відповідь JSON-файлом) і виводить відповідь на сторінці.

4. Вивести список користувачів із JSON-файлу

Після натискання кнопки завантажити локальний JSON-файл з користувачами та відобразити список імен.

5. AJAX-запит з обробкою помилки

Надіслати AJAX-запит на неіснуючу URL-адресу та відобразити повідомлення про помилку, якщо запит не вдасться.

Завдання середньої складності

6. Автозаповнення (autocomplete) за допомогою AJAX

При введенні тексту в поле відправляти AJAX-запит до сервера (або JSON-файлу) і виводити підказки (наприклад, список міст, імен тощо).

7. Чат-інтерфейс з AJAX-оновленням

Створити інтерфейс чату, який кожні 5 секунд оновлює список повідомлень з серверного JSON-файлу (імітація живого чату).

8. Динамічне сортування таблиці даних через AJAX

Завантажити таблицю даних через AJAX (наприклад, список продуктів) та реалізувати сортування по стовпцях (наприклад, по ціні, алфавіту).

9. AJAX-запит із параметрами (GET-запит з query params)

Надіслати AJAX-запит на публічне API (наприклад, пошук зображень, книг, фільмів тощо) із параметром з текстового поля, і вивести результати.

10. Каскадні списки (країна → місто) з AJAX

При виборі країни надсилати AJAX-запит на отримання відповідних міст і динамічно заповнювати другий <select>.

Завдання на Fetch

1. Отримати дані про випадкового користувача з API

Зробити fetch-запит до https://randomuser.me/api/ і вивести ім’я, фото та email користувача.

2. Завантажити список постів із JSONPlaceholder

Отримати дані з https://jsonplaceholder.typicode.com/posts і вивести заголовки у вигляді списку.

3. Додавання нового посту методом POST

Реалізувати форму, яка надсилає POST-запит за допомогою fetch до https://jsonplaceholder.typicode.com/posts і виводить результат.

4. Кнопка “Завантажити ще” з Fetch-пагінацією

Завантажити перші 5 елементів з API (наприклад, ?_limit=5&_page=1), потім при натисканні “Завантажити ще” завантажити наступні 5.

5. Fetch із обробкою статусів (успіх / помилка)

Надіслати fetch-запит до будь-якого ресурсу і при успіху відобразити повідомлення “Успішно!”, а при помилці — код статусу та повідомлення “Щось пішло не так”.