**Державний університет телекомунікацій**

**Кафедра технологій цифрового розвитку**

**З В І Т**

**З лабораторного проекту**

**WEB-програмування JavaScript**

**Виконав: студент групи ТЦР-23**

**Старостенко Анатолій Володимирович**

**Перевірив: Дзюба Вадим Віталійович**

**Київ-2025**

**Мета роботи:** Закріпити знання про асинхронне програмування в JavaScript (Fetch API, async/await, обробка помилок). Освоїти роботу з даними з API та їх обробку для відображення у веб-інтерфейсі. Відпрацювати роботу з DOM: створення, маніпуляція, динамічне оновлення розмітки. Розробити додатковий функціонал (фільтрація/сортування) для інтерактивності застосунку. Використати сучасні можливості ES6+: стрілочні функції, шаблонні рядки, деструктуризацію.

**Хід роботи**

(весь код продемонстровано в додатку А до звіту (ст.5)

**1. Планування:** Для початку було спроєктовано логіку вебзастосунку — односторінкова аплікація, яка динамічно завантажує дані про фільми з API [TV Maze](https://api.tvmaze.com/shows) та відображає їх у вигляді карток.  
Передбачено функціонал пошуку за назвою, сортування за назвою або рейтингом, а також фільтрацію за жанрами.

Основні компоненти:

* Поле пошуку
* Випадаючі списки сортування та жанру
* Список фільмів
* Повідомлення про помилки

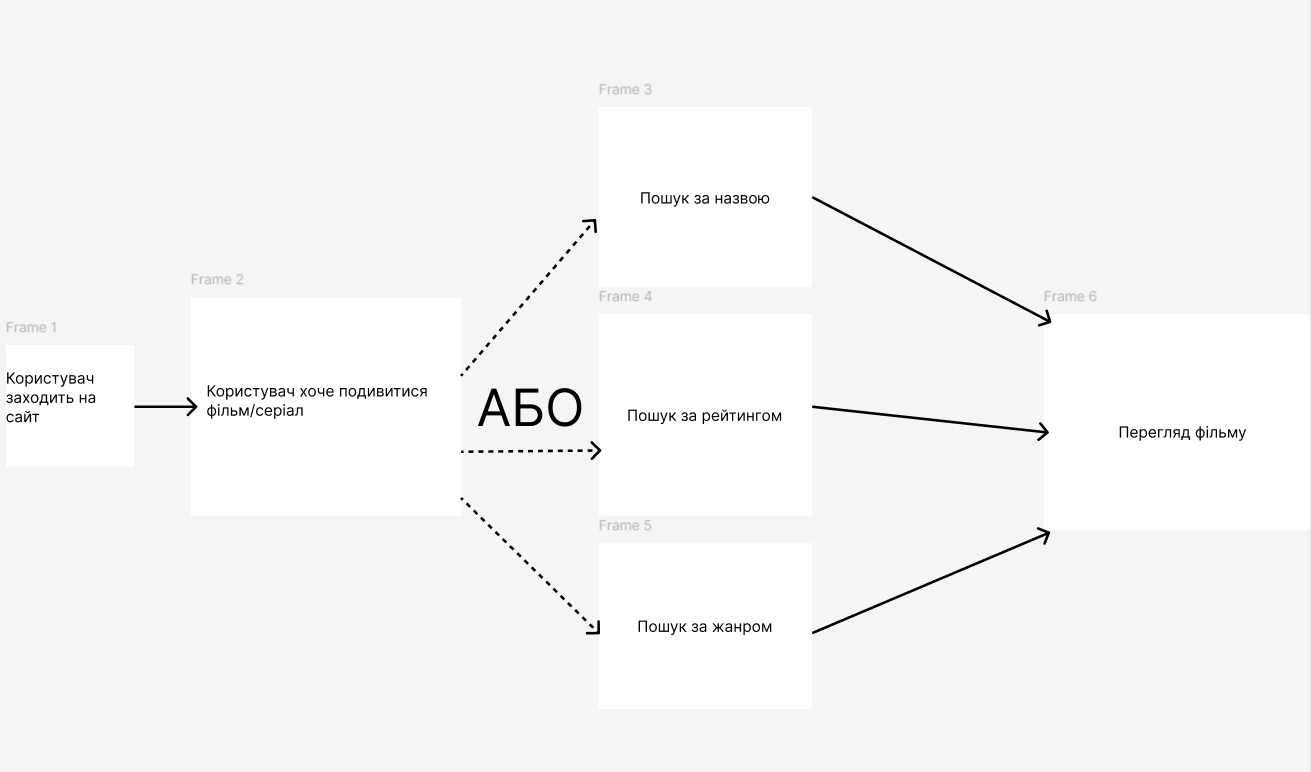


Рисунок1.1-Діаграма планування сайту

**2. Розробка HTML шаблону:** Створено HTML-файл, який містить: Базову структуру сторінки.Елементи керування (input для пошуку, select для сортування і жанру, кнопка "Знайти").Контейнер для карток фільмів.Підключено зовнішній CSS-файл зі стилями.Підключено скрипт app.js, у якому реалізовано всю логіку

**3. Отримання даних з API та обробка помилок:** Використовуючи async/await і Fetch API, реалізовано асинхронне завантаження даних із відкритого API TV Maze.

Запит опрацьовується з обробкою можливих помилок (try/catch). У випадку помилки відображається повідомлення для користувача.

**4**. **Введення, обробка і валідація даних:** Отримані дані форматуються у вигляді карток за допомогою шаблонних рядків. Для кожного фільму відображається:

-Постер

-Назва

-Жанри

-рейтинг

У розмітці використовуються сучасні можливості JavaScript, такі як деструктуризація, стрілочні функції, шаблонні рядки.

**5. Реалізація фільтрації та сортування:**

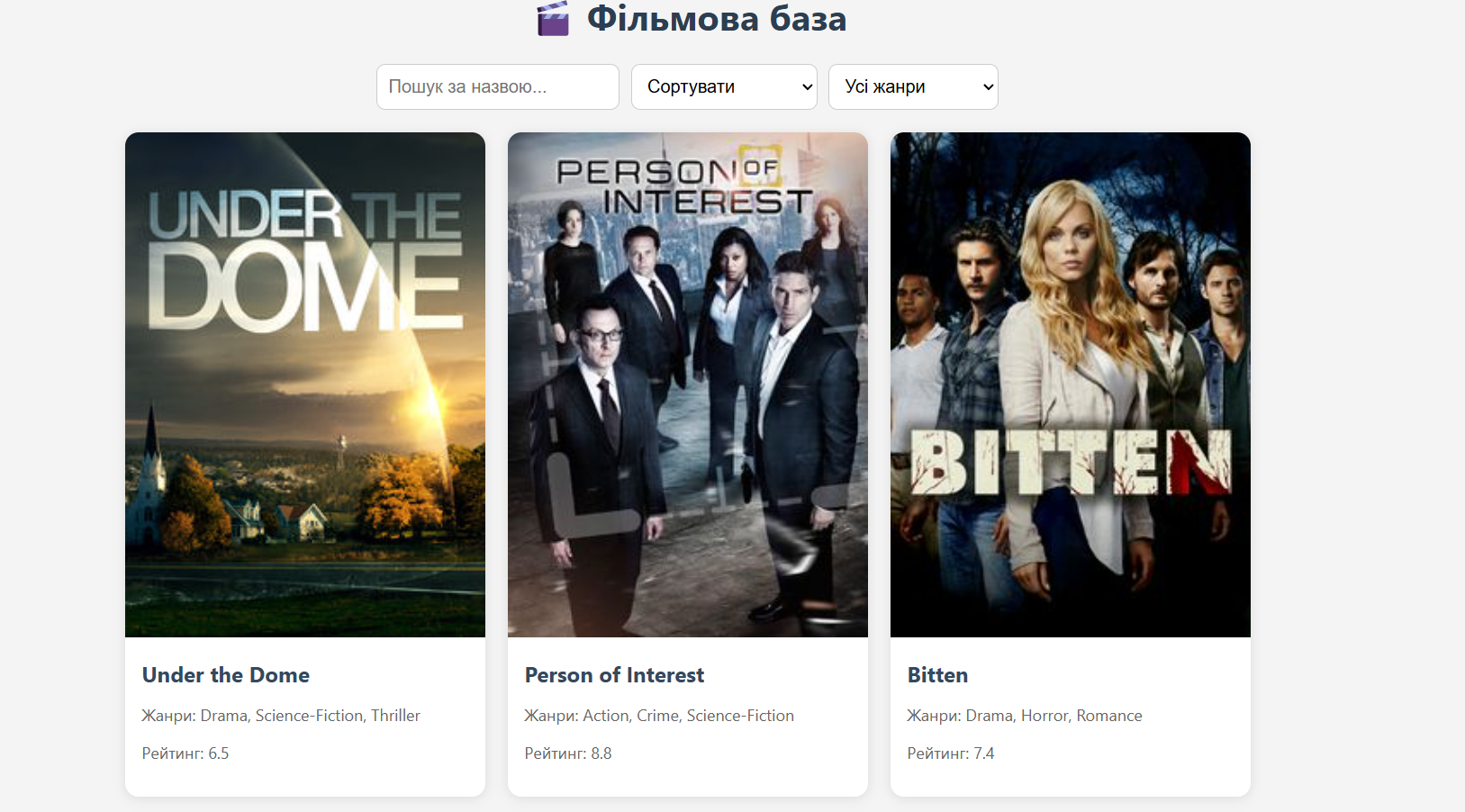
* Пошук за назвою у реальному часі або після натискання кнопки "Знайти"
* Сортування за назвою (алфавіт) або за рейтингом (від більшого до меншого)
* Фільтрація за жанром, при цьому жанри автоматично зчитуються з API та формують список у select

Всі ці дії динамічно оновлюють вміст сторінки без перезавантаження.

**6. Відображення та зручність користувача:** Дані відображаються у зручному UI у вигляді сітки. Додано адаптивну верстку.  
Для стилізації використано кастомний CSS: градієнтний фон, приємні тіні, закруглені кути, ефекти наведення.

Користувач отримує візуально приємний інтерфейс з можливістю фільтрації, пошуку та швидкого перегляду необхідної інформації про фільми.

**Результат:**



Русунок.1.1-Головна сторінка Фільмової бази

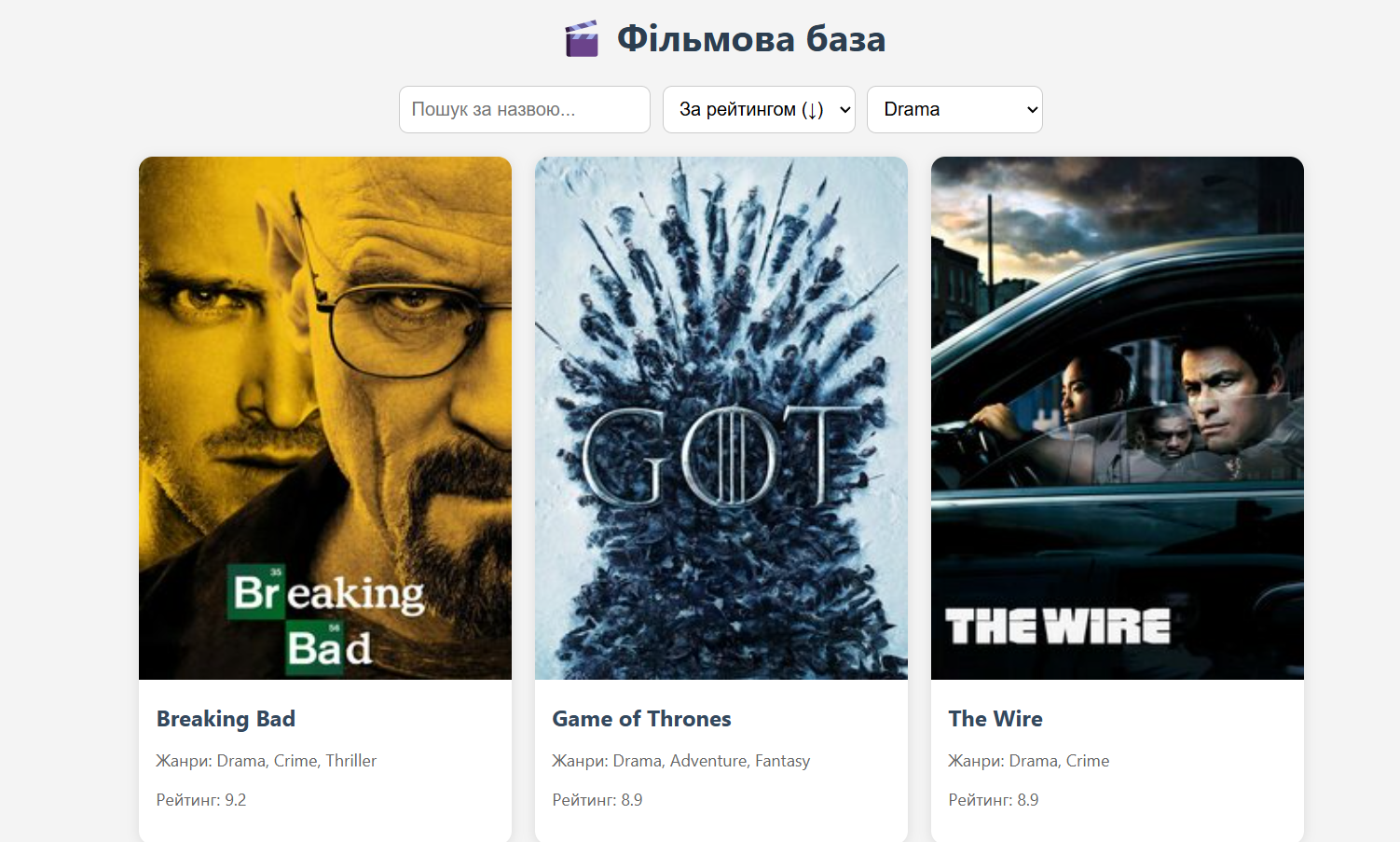


Рисунок1.2.-Приклад пошуку за “Рейтенгом” та “Жанром”

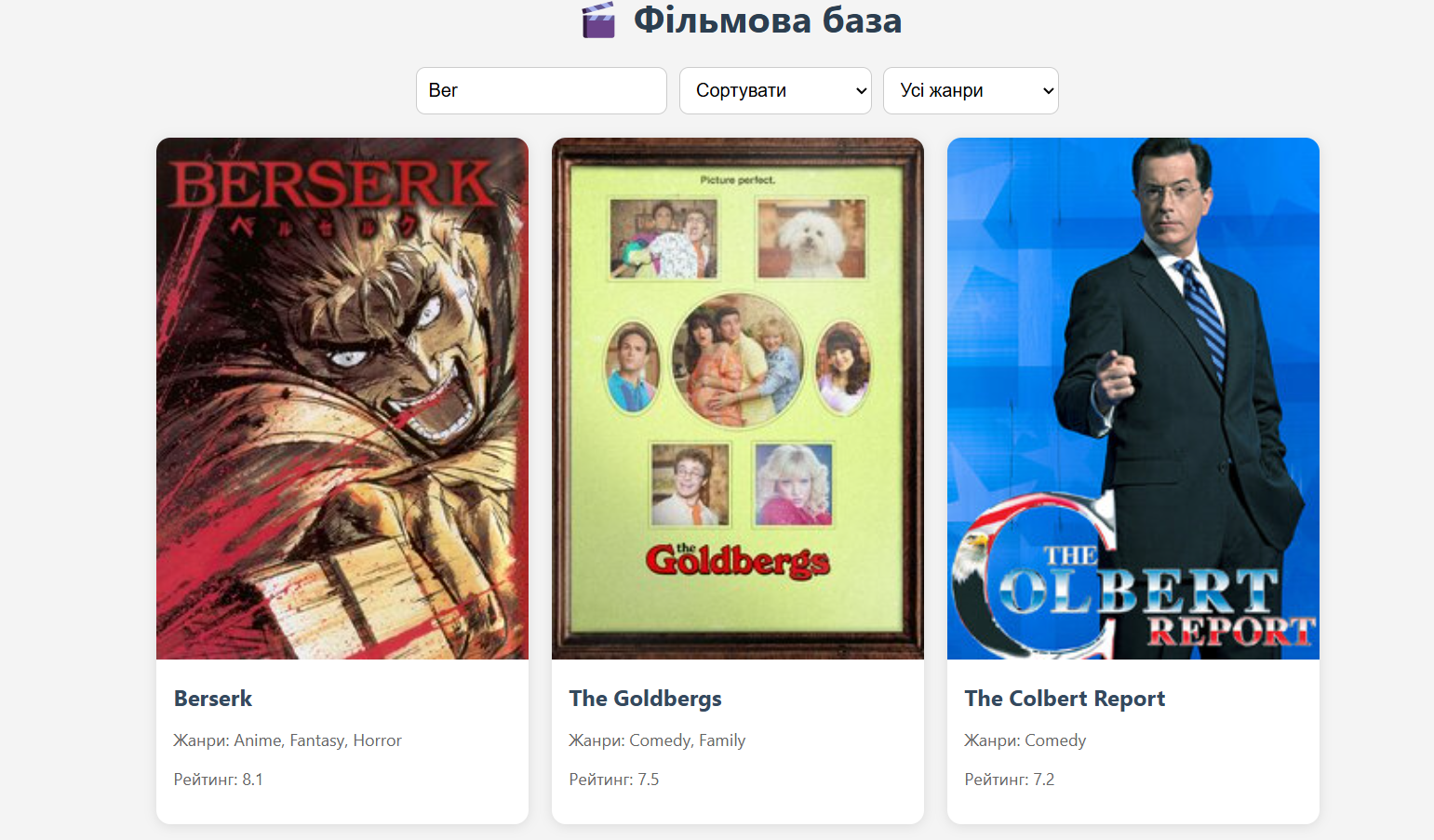


Рисунок.1.3-Приклад пошуку за нозвою фільма

**Посилання на проект:** [**Посилання**](https://shokoladkafav.github.io/Shoko1.io/SecondYEARS/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0_2/index.html)

**Висновок:** У результаті виконання лабораторної роботи створено повноцінну базу фільмів,в яких реалізовано всі вимоги: отримання даних з API, обробка помилок, відображення даних на сторінці, реалізація функціоналу фільтрації та сортування. В роботі ефективно застосовано асинхронне програмування на JavsScript, робота з данними із API, роботу з DOM та використання сучасних можливостей ES6+.

**Додаток А**

**index.html:**

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Фільмова база</title>

<link rel="stylesheet" href="style.css" />

</head>

<body>

<div class="app">

<h1>🎬 Фільмова база</h1>

<div class="controls">

<input type="text" id="searchInput" placeholder="Пошук за назвою..." />

<select id="sortSelect">

<option value="">Сортувати</option>

<option value="name">За назвою (А-Я)</option>

<option value="rating">За рейтингом (↓)</option>

</select>

<select id="genreSelect">

<option value="">Усі жанри</option>

</select>

</div>

<div id="moviesContainer" class="movies-container">

<!-- Тут будуть картки фільмів -->

</div>

<div id="errorMessage" class="error-message"></div>

</div>

<script src="app.js" type="module"></script>

</body>

</html>

**style.css:**

body {

margin: 0;

font-family: "Segoe UI", sans-serif;

background-color: #f4f4f4;

color: #333;

}

.app {

max-width: 1000px;

margin: auto;

padding: 20px;

}

h1 {

text-align: center;

color: #2c3e50;

}

.controls {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 10px;

margin-bottom: 20px;

}

.controls input,

.controls select {

padding: 10px;

font-size: 1rem;

border: 1px solid #ccc;

border-radius: 8px;

}

.movies-container {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 20px;

}

.movie-card {

background-color: #fff;

border-radius: 12px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

overflow: hidden;

transition: transform 0.2s ease;

}

.movie-card:hover {

transform: scale(1.02);

}

.movie-card img {

width: 100%;

height: auto;

}

.movie-content {

padding: 15px;

}

.movie-title {

font-size: 1.2rem;

margin: 0 0 10px;

color: #34495e;

}

.movie-info {

font-size: 0.9rem;

color: #666;

}

.error-message {

margin-top: 20px;

color: red;

text-align: center;

font-weight: bold;

}

**script.js:**

'use strict';

const API\_URL = 'https://api.tvmaze.com/shows';

const moviesContainer = document.getElementById('moviesContainer');

const searchInput = document.getElementById('searchInput');

const sortSelect = document.getElementById('sortSelect');

const genreSelect = document.getElementById('genreSelect');

const errorMessage = document.getElementById('errorMessage');

let allMovies = [];

// Отримання даних з API

async function fetchMovies() {

try {

const response = await fetch(API\_URL);

if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);

const data = await response.json();

allMovies = data;

populateGenres(data);

renderMovies(data);

} catch (error) {

errorMessage.textContent = 'Помилка при завантаженні фільмів. Спробуйте пізніше.';

console.error(error);

}

}

// Заповнення випадаючого списку жанрів

function populateGenres(movies) {

const genres = new Set();

movies.forEach(movie => {

movie.genres.forEach(genre => genres.add(genre));

});

const sortedGenres = Array.from(genres).sort();

sortedGenres.forEach(genre => {

const option = document.createElement('option');

option.value = genre;

option.textContent = genre;

genreSelect.appendChild(option);

});

}

// Відображення фільмів

function renderMovies(movies) {

moviesContainer.innerHTML = '';

movies.forEach(({ name, image, rating, genres }) => {

const card = document.createElement('div');

card.className = 'movie-card';

card.innerHTML = `

<img src="${image?.medium || 'https://via.placeholder.com/210x295?text=No+Image'}" alt="${name}">

<div class="movie-content">

<h2 class="movie-title">${name}</h2>

<p class="movie-info">Жанри: ${genres.join(', ')}</p>

<p class="movie-info">Рейтинг: ${rating.average ?? 'N/A'}</p>

</div>

`;

moviesContainer.appendChild(card);

});

}

// Пошук за назвою

function filterMoviesByName(query) {

return allMovies.filter(({ name }) =>

name.toLowerCase().includes(query.toLowerCase())

);

}

// Фільтрація за жанром

function filterMoviesByGenre(genre) {

if (!genre) return allMovies;

return allMovies.filter(movie => movie.genres.includes(genre));

}

// Сортування

function sortMovies(movies, criteria) {

const sorted = [...movies];

if (criteria === 'name') {

sorted.sort((a, b) => a.name.localeCompare(b.name));

} else if (criteria === 'rating') {

sorted.sort((a, b) => (b.rating.average ?? 0) - (a.rating.average ?? 0));

}

return sorted;

}

// комбінована фільтрація + сортування

function updateResults() {

const nameQuery = searchInput.value;

const genreQuery = genreSelect.value;

const sortQuery = sortSelect.value;

let filtered = allMovies;

if (nameQuery) {

filtered = filterMoviesByName(nameQuery);

}

if (genreQuery) {

filtered = filtered.filter(movie => movie.genres.includes(genreQuery));

}

const finalResult = sortQuery ? sortMovies(filtered, sortQuery) : filtered;

renderMovies(finalResult);

}

// Події

searchInput.addEventListener('input', updateResults);

genreSelect.addEventListener('change', updateResults);

sortSelect.addEventListener('change', updateResults);

// Запуск

fetchMovies();