

РОЗРОБКА ВЕБ-ДОДАТКУ СТРИМІНГОВОГО СЕРВІСУ ДЛЯ ФІЛЬМІВ

Вінницький національний технічний університет

Анотація

Розроблено стримінговий веб-сайт для пошуку та перегляду фільмів. Обґрунтовано доцільність використання при розробці бібліотеки React та мови програмування JavaScript. Розроблений сервіс надає можливість користувачу переглядати фільми та знаходити інформацію про акторів, також є можливість зареєструвати особистий кабінет, в якому можна знайти фільми, що сподобалися вам раніше

Ключові слова: фільми, стримінговий сервіс, аутентифікація, React, JavaScript

Abstract

A streaming website for searching and watching movies has been developed. The expediency of using the React library and the JavaScript programming language in the development is substantiated. The developed service allows the user to watch movies and find information about the actors, there is also the opportunity to register a personal account where you can find movies that you liked before

Keywords: movies, streaming service, authentication, React, JavaScript

Вступ

В двадцять першому столітті значно розвинулась сфера технологій, до такої міри, що люди звикли покладатися на неї майже у всіх сферах своєї діяльності, починаючи від робочої діяльності і закінчуючи дозволями.

Перегляд фільмів зазвичай дає колосальну можливість самим обирати свій настрій. Залежно від настрою ти можеш бути супергероєм чи поліцейським, лікарем чи водієм. Найбільш дружнім для користувача є веб-додаток, завдяки якому можна знайти фільм саме на ваш смак і настрої без додаткового скачування, а також збиткової реклами. Отже, завдання розробника полягає у створення функціонального та повноцінного продукту, який включає в себе: гарний та зрозумілий інтерфейс, можливість знайти будь-який фільм, подивитися його опис та акторів, які у ньому знімаються.

Метою роботи є розробка функціонально-повного веб-додатку, який дасть змогу знайти та переглянути фільми.

Аналіз дослідження

Веб-додаток - це програма-клієнт-сервер, що зберігається на віддаленому сервері, яка використовує веб-браузери та веб-технології для виконання певних функцій через Інтернет через інтерфейс браузера.

"Клієнт" використовується в середовищі клієнт-сервер для позначення програми, яку людина використовує для запуску програми. Середовище клієнт-сервер - це середовище, в якому кілька комп'ютерів обмінюються інформацією, наприклад введенням інформації в базу даних. "Клієнт" - це програма, яка використовується для введення інформації, а "сервер" - програма, що використовується для зберігання інформації.

Із зростанням кількості користувачів, що означає збільшення кількості клієнтів, більш очевидною стає потреба у використанні трирівневої архітектури. Дана архітектура може бути розширена до багаторівневої, тобто трирівнева архітектура є окремим випадком багаторівневої архітектури. Головна особливість такої архітектури полягає в тому, що функції обробки, зберігання та представлення даних розподілені на декілька серверів, що впливає на ефективність роботи. Схема трирівневої архітектури наведена на рис. 1.

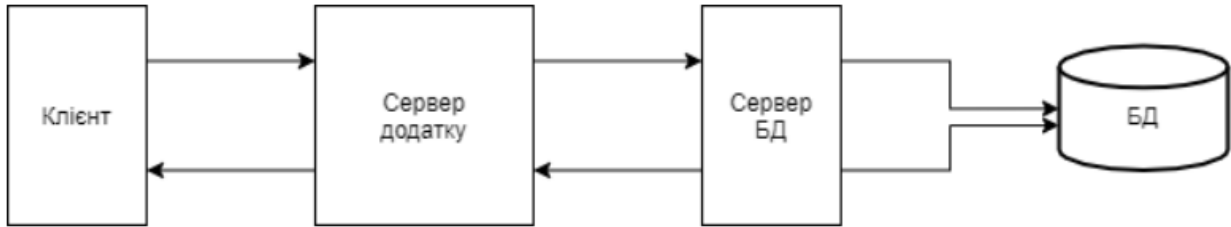


Рис. 1 – Трирівнева архітектура «клієнт-сервер»

Складається з наступних рівнів:

- 1) клієнт – інтерфейс, що надається користувачу;
- 2) сервер додатків –рівень, на якому зосереджена більша частина логіки;
- 3) сервер баз даних – зберігання даних та винесення їх на окремий рівень.

На середньому рівні відбувається обмін між користувачами та базами даних. Наявність середнього рівня звільняє сервер баз даних від більшої частини роботи, оскільки виконує запити від клієнта.

Як і дворівнева архітектура, трирівнева також має свої переваги і недоліки. Серед переваг наступні:

- 1) сервер баз даних звільняється від більшості операцій, які забирає на себе сервер додатків;
- 2) зменшено розмір клієнтських додатків, оскільки вони мають менший код;
- 3) сбалансоване навантаження на мережу;
- 4) забезпечення єдиної поведінки усіх клієнтів, тобто при зміні загального кода сервера автоматично змінюється поведінка клієнтів.

На відміну від звичайних веб-сайтів, додаток на React JS — це можливість створити інтерактивний продукт, в якому інтерфейс буде максимально швидко відповідати на будь-які дії відвідувача платформи.

Також він дозволяє реалізувати більш складні, інтерактивні елементи інтерфейсу, якщо порівнювати з реалізацією на сайті.

На стороні клієнта можна виконувати:

- більш інтерактивну анімацію;
- складні розрахунки;
- створювати offline-first додатки, що дозволяють функціонувати при втраті інтернет-з'єднання;
- максимально чуйні форми і таблиці, які не потребують перезавантаження сторінки.

Односторінковий додаток React (SPA) не є чимось особливим. Всі веб-ресурси, зроблені за допомогою фреймворків, односторінкові за своєю структурою. SPA дозволяють відразу повністю завантажити всю бізнес-логіку продукту, і працювати подібно настільним додаткам, без постійних запитів сторінок окремо. Подібні сайти запитують тільки той контент, що вимагає поновлення.

Що стосується такого розділу, як мобільні додатки на React, висока продуктивність і гнучкість бібліотеки знайшла застосування в реалізації фреймворка для мобільних додатків — React Native, в якому реакт виконує роль бібліотеки відображення інтерфейсу.

Даний фреймворк наділений наступними властивостями:

- 1) однонаправленість передачі даних – передача властивостей від основних компонентів до дочірніх, при чому компоненти отримують властивості у незмінному вигляді, що не дає можливості їх змінити;
 - 2) віртуальний DOM –об'єктна модель документа, з допомогою якої здійснюється доступ до HTML.
- Використання цієї особливості допомагає оптимально оновити структуру розроблюваного інтерфейсу. При роботі зі сторінкою оновлюється тільки необхідні компоненти;

3) розширення JSX – розширення синтаксису мови JavaScript, він використовується для пояснення React зовнішнього вигляду інтерфейсу користувача (UI);

4) react Hooks допомагає використовувати можливості React. При цьому відпадає необхідність у написанні класів;

5) методи життєвого циклу – методи, які допомагають запустити код на будь-якій стадії життєвого циклу компонента. Наприклад, метод `render` необхідний для відображення даних у інтерфейсі.

Основним недоліком фреймворку є те, що даний фреймворк охоплює тільки рівень представлення вебсистеми, тобто призначений тільки для створення клієнтської частини додатку, інтерфейсу користувача.

Висновки

Розроблено веб-додаток для перегляду та пошуку фільмів без додаткової реклами та скачувань. Досліджено архітектура «клієнт - сервер», яка дала змогу обґрунтувати використання фреймворку React, який забезпечує однонаправленість передачі даних, віртуальний DOM, розширення JSX, react Hooks та методи життєвого циклу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Клієнт-серверна архітектура та ролі серверів. 2017. URL: <https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>(дата звернення 07.05.2022).
2. Розробка додатків на React JS. 2022.URL: <https://artjoker.ua/uslugi/razrobotka-prilozheniy-na-react-js/> (дата звернення 07.05.2022).
3. Додатки для перегляду фільмів, серіалів і мультфільмів. 2022. URL: <https://dtv.plus/uk/video> (дата звернення 07.05.2022).

Федорова Вероніка Володимирівна – студентка групи 2КН-18Б, факультету інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця, email : stud.fedorova.veronika@vntu.edu.ua.

Озеранський Володимир Сергійович - кандидат технічних наук, старший викладач, Вінницький національний технічний університет, м. Вінниця, e-mail: ozeransky@ukr.net.

Fedorova Veronika - Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: stud.fedorova.veronika@vntu.edu.ua.

Ozeransky Volodumir - Ph.D., senior lecturer, Vinnytsia National Technical University, Vinnytsia, email: ozeransky@ukr.net.