

Висновок. Рендеринг в хмарі дає змогу використовувати велику кількість обчислювальних ресурсів, можна рендерити кілька кадрів одночасно, що може значно прискорити рендеринг об'ємних сцен. Використання вебплатформи дозволяє сплачувати тільки за обчислювальний час і ресурси, які використовуються, що може бути економічно вигідно, особливо для невеликих студій та фрілансерів. Істотною перевагою даної системи є те, що вона розроблялась на основі відкритих стандартів та з використанням розповсюдженого програмного забезпечення з відкритим кодом.

Список використаних літературних джерел

- [1] Programming 3D Animations and Visualizations for the Web with HTML5 and WebGL. Jos Dirksen. 2018.
- [2] Three.js Cookbook. Jos Dirksen. 2015.
- [3] Programming TypeScript. Making Your JavaScript Applications Scale. Boris Cherny. 2019.
- [4] React [Електронний ресурс] (<https://ru.legacy.reactjs.org/>);
- [5] Three.js [Електронний ресурс] (<https://threejs.org/>);
- [6] TypeScript [Електронний ресурс] (<https://www.typescriptlang.org/>).

М.І. Притика, В.І. Яркун*

студент КН-6, НЛТУ України, Львів, Україна, prytyka@nltu.lviv.ua

*старший викладач кафедри ІПЗ, НЛТУ України, Львів, Україна.

yarkun@nltu.edu.ua, <https://orcid.org/0000-0002-4876-1111>

Розроблення веб-орієнтованої рекомендаційної системи “Медіа контент”

Анотація. Створено веб-орієнтовану рекомендаційну систему “Медіа контент”, що надає можливість організовувати списки фільмів, серіалів, а також перегляду статистики, рекомендацій тощо. У системі є функціонал, що дозволяє виставити рейтинг чи рекомендацію для певного виду медіа контенту. Рекомендаційна система створена сучасними засобами побудови веб-орієнтованих застосунків із збереженням даних в БД. Для запуску СУБД MySQL використовується ПЗ, що автоматизує розгортання і управління додатками в середовищах з підтримкою контейнеризації Docker. Для реалізації збору

даних про відгуки, використовується публічний та безкоштовний в навчальних цілях сервіс themoviedb.org.

Ключові слова – Java, веб додаток, рекомендації, Angular, BootStrap, TypeScript, MySQL, Docker, GitHub.

Враховуючи те, що багато людей любить дивитись цікавий контент, починаючи з фільмів та закінчуючи мультфільмами і телешоу, було прийняте рішення розробити рекомендаційну систему, яка дозволить вирішити багато питань та спростить людям вибір контенту в медіа сфері. Користувач сайту зможе створити свій особистий список переглянутого контенту, додати туди будь-який фільм, серіал, мультфільм тощо, вказати кількість переглянутих епізодів та виставити свій особистий рейтинг і рекомендацію для перегляду. На основі вподобань та світового рейтингу, користувачу будуть надані рекомендації та графіки популярності контенту.

Для розробки системи використано найбільш доступні та популярні технології. Java, Spring [1], Maven для backend-частини, Angular [2], TypeScript та Bootstrap для frontend-частини (Рис.1). Також використані 3rd party API для того щоб отримувати актуальну інформацію, проводити її опрацювання, розрахунки тощо. Таким чином користувач завжди буде отримувати свіжі дані та статистику про ту чи іншу одиницю контенту. Для зберігання інформації була обрана СУБД MySQL, яка в свою чергу запускається через підготовлений набір команд docker-compose та працює в контейнері Docker. Це дозволяє швидко “підняти” систему на будь-якому сервері та полегшує роботу з розгортанням середовища.

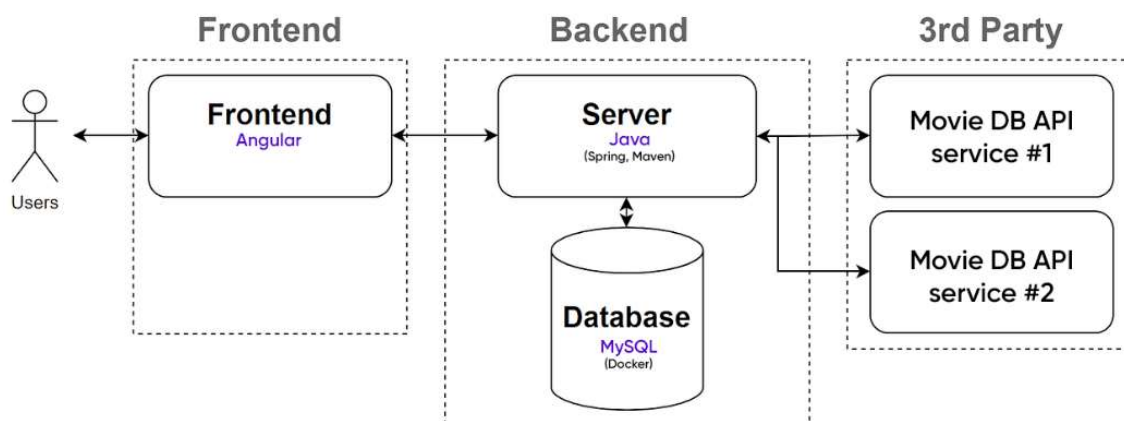


Рис.1. Комунікація між залежностями системи.

Для запуску СУБД MySQL використовується ПЗ, що автоматизує розгортання і управління додатками в середовищах з підтримкою контейнеризації під назвою Docker. Для цього був створений спеціальний docker-compose файл (Рис.2.), який при введенні команди docker-compose up, автоматично завантажує та запускає СУБД MySQL.

```
version: '3.1'

services:
  db:
    image: mysql:latest
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: '12345678'
      MYSQL_DATABASE: 'kinolist'
    ports:
      - "3306:3306"
```

Рис.2. – Вміст docker-compose.yaml файла

Інтерфейс реалізований за допомогою Bootstrap (Рис.3.), використовуючи Angular, комунікує з сервером Java, який в свою чергу збирає необхідну інформацію і обробляє її так, як потрібно користувачу, зберігає її в базу даних та повертає на frontend.



Рис.3. Інтерфейс системи (зліва). Графіки розподілу оцінок контенту та істо-

рія середнього рейтингу (справа).

Система “Медіаконтент” оснащена рейтинговою складовою. Тут графік розподілу оцінок - це кількість оцінок які поставили користувачі на кожну одиницю рейтингу. На другому графіку вказана історія зміни середнього рейтингу від дати виходу до сьогоднішнього дня, завдяки цьому можна зрозуміти в який час свого життєвого циклу контент мав найвищий рейтинг. Інформація для цього також береться з API, обробляється та формує графіки для кожної одиниці контенту. Для обробки використовується як backend так і frontend.

Цінність контенту полягає в його якості, а для того щоб вберегти користувача від неякісного контенту створено систему, яка буде враховувати його побажання, світові рейтинги критиків, інших користувачів та дозволить вести список контенту з особистою оцінкою та рекомендацією для інших. Таким чином користувач отримає те, що цікаво йому і водночас допоможе іншим шляхом висловлювання власної оцінки та рекомендації.

Для реалізації збору даних про відгуки, використовується публічний та безкоштовний в навчальних цілях сервіс themoviedb.org [3]. Цей сервіс робить комбінацію даних фільму по його назві використовуючи всі популярні сервіси по зберіганню інформації про контент, наприклад [imdb](https://www.imdb.com/). Для розробників, після генерації приватного безкоштовного ключа доступу, надається доступ до API, завдяки якому можна без проблем знайти потрібний фільм, його відгуки та огляди від користувачів.

Висновок. Створена веб-орієнтована рекомендаційна система під назвою "Медіа контент" надає користувачам можливість організовувати список фільмів та серіалів, а також отримувати рекомендації та статистику з їх перегляду. Система включає в себе функціонал, який дозволяє користувачам виставляти рейтинги та рекомендації для різних видів медіа контенту. Використані сучасні ІТ-засоби для розробки веб-орієнтованої системи. Дані користувачів та медіа контент зберігаються в базі даних (СУБД MySQL), а для її розгортання та управління використовується програмне забезпечення, яке автоматизує процеси в середовищах з підтримкою контейнеризації Docker. Для збору даних про відгуки

та інформацію про медіа контент був використаний публічний та безкоштовний сервіс themoviedb.org. Система "Медіа контент" може мати потенціал застосування у різних галузях, включаючи розваги та медіа індустрії, де рекомендації та статистика є важливими складовими.

Список використаних літературних джерел

- [1] Spring Boot 2: найкращі практики для професіоналів. Гутьєррес Феліпе. Print2print. 2020.
- [2] Angular і TypeScript. Сайтобудівництво для професіоналів. Яків Файн. 2017.
- [3] База даних фільмів [Електронний ресурс] : [Вебсайт]. – Режим доступу: <https://www.themoviedb.org/> (дата звернення 13.09.2023). – Назва з екрана.

Михайло Ченчак, Ірина Борецька*

студент 6-го курсу, НЛТУ України, Львів, Україна.

mykhailo.chenchak@gmail.com

к.т.н., завідувач кафедри КН НЛТУ України, Львів, Україна,

<https://orcid.org/0000-0002-6767-104X>. iryna.boretska@gmail.com.

Розроблення архітектури вебзастосунок для комунікації між онлайн-видавцями та рекламодавцями

Анотація - В процесі розробки архітектури вебзастосунок для взаємодії між онлайн-видавцями і рекламодавцями створено архітектуру потужного інструменту, який спростить комунікацію, підвищить продуктивність роботи та покращить задоволення клієнтів. Це сприятиме зміцненню партнерських відносин між видавцями та рекламодавцями, що своєю чергою позитивно вплине на обидві сторони та допоможе досягти більших успіхів у віртуальному середовищі. Результати дослідження та розробки можуть бути використані в інформаційних технологіях для побудови даної системи.

Ключові слова – вебзастосунок, архітектура, онлайн-видавці, рекламодавці, комунікація, SPA, React, без серверна архітектура, хмарна платформа, Azure, NoSQL.