

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

SPRING-ФРЕЙМВОРК ТА ЙОГО ВИКОРИСТАННЯ У WEB-ПРОЕКТАХ

Виконав студент 4-го курсу
Федоренко Максим Сергійович



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Кузенко Володимир Федорович



(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії та технології
програмування
« 01 » червня _____ 2022 р.
протокол № 10
Завідувач кафедри
М. С. Нікітченко



(підпис)

Київ – 2022

РЕФЕРАТ

Обсяг роботи 51 сторінка, 33 ілюстрації, 23 таблиці, 22 джерела посилання та 1 додаток.

ВЕБ-ЗАСТОСУНОК, SPRING, TELEGRAM BOT, БАЗА ДАНИХ, JPA, POSTGRESQL, КІНОМЕРЕЖА.

Об'єктом дослідження є сфера кіно та кінотеатрів.

Мета кваліфікаційної роботи: дослідити Spring-фреймворк та розробити на ньому власний веб-додаток міні-соціальної кіномережі і системи бронювання та купівлі білетів у кінотеатрах.

Методи розроблення: розробка програмного продукту на основі поєднання системи бронювання та купівлі квитків в різних кінотеатрах з міні-соціальною мережею для кіноманів. Інструменти розроблення: інтегроване середовище розробки IntelliJ IDEA, мова програмування Java, Spring-фреймворк, СУБД PostgreSQL.

Результати кваліфікаційної роботи: було проведено огляд та аналіз існуючих рішень на ринку, досліджено основні етапи розробки веб-сайту, вимоги та програмні засоби для його створення. Також було спроектовано та розроблено веб-застосунок «A.Movie».

Веб-сайт «A.Movie» можна водночас використовувати як і сервіс для бронювання та купівлі квитків в різних кінотеатрах та містах, так і для пошуку фільмів, серіалів, персон, ведення свого облікового запису з переглянутими фільмами та збереженими до них оцінками.

ЗМІСТ

| | |
|--|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ | 5 |
| ВСТУП..... | 6 |
| РОЗДІЛ 1. ОГЛЯД ТА ПОРІВНЯННЯ НАЯВНИХ НА РИНКУ РІШЕНЬ..... | 8 |
| 1.1 IMDb | 8 |
| 1.2 Multiplex | 9 |
| 1.3 Деякі інші рішення та їх порівняння..... | 9 |
| РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ..... | 11 |
| 2.1 Spring-фреймворк | 11 |
| 2.2 PostgreSQL | 13 |
| 2.3 Firebase API..... | 14 |
| 2.4 Google and Facebook APIs..... | 15 |
| 2.5 Stripe | 15 |
| 2.6 Telegram..... | 16 |
| 2.7 FreeMarker | 17 |
| РОЗДІЛ 3. ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ | 19 |
| 3.1 Призначення застосунку..... | 19 |
| 3.2 Цілі створення застосунку..... | 19 |
| 3.3 Архітектурні особливості системи..... | 20 |
| РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ | 23 |
| 4.1 Реалізація бази даних | 23 |
| 4.1.1 Логічна структура бази даних..... | 23 |
| 4.1.2 Опис таблиць бази даних програми | 24 |

| | |
|--|----|
| 4.2 Розробка застосунку..... | 30 |
| 4.3 Сценарії використання веб-додатку..... | 33 |
| 4.3.1 Сценарій адміністратора..... | 33 |
| 4.3.2 Сценарій звичайного користувача..... | 38 |
| ВИСНОВКИ..... | 48 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... | 49 |
| ДОДАТКИ..... | 51 |
| Додаток А. Діаграма класів..... | 51 |

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – Application Programming Interface;

СУБД – Система Управління Базами Даних;

SQL – Structured Query Language;

HTTP – HyperText Transfer Protocol;

ORM – Object-Relational Mapping;

CRUD – Create, Read, Update, Delete

БД – База Даних;

CSS – Cascading Style Sheets;

HTML – HyperText Markup Language;

JS – JavaScript;

IDE – Інтегроване Середовище Розробки;

JPA - Java Persistence API;

ВСТУП

Оцінка сучасного стану об'єкта розробки. Інтернет як інструмент комунікації, який не має територіальних кордонів, дозволяє обмінюватись різними видами інформації. Кількість людей, які використовують Інтернет як найважливіший засіб для отримання потрібних відомостей про послуги, значно зросла останнім часом. Щоб надавати свої послуги в мережі Інтернет, компанії необхідно розробити веб-сайт. Він буде як візитівкою для отримання контактних даних та уявлення про сферу послуг компанії, так і функціональним засобом для безпосереднього отримання клієнтами цих послуг. Наявність власного веб-застосунку позитивно впливає на імідж компанії, підвищує довіру клієнтів.

В наш час існує багато сайтів для того, щоб онлайн можна було бронювати та купувати квитки на сеанси в кінотеатрах замість того, щоб як раніше потрібно було дзвонити, витрачаючи на це додатковий час, з можливістю втратити своє місце, якщо ти не прийшов вчасно. І з кожним роком відкривається дедалі більше нових кінотеатрів, тому розробка веб-сайтів для них є затребуваною роботою.

Актуальність роботи та підстави для її виконання. Під час карантину та пандемії люди більше почали проводити своє дозвілля за переглядом фільмів та серіалів. І вони б хотіли мати якийсь засіб для збереження своїх переглянутих кінокартин та їх оцінок для того, щоб можна було з друзями ділитися своїми враженнями від перегляду. Тому розробка кіномережі для таких людей у сьогоденнішніх реаліях актуальна.

А якщо поєднати цю ідею з сервісом для бронювання, купівлі квитків, то можна отримати цікавий продукт, тому що на ринку мені не відомі рішення, які б містили обидва ці пункти.

Мета й завдання роботи. Метою кваліфікаційної роботи є розробка веб-застосунку «A.Movie», який буде поєднанням ідеї кіномережі з сервісом для бронювання та купівлі квитків в різних кінотеатрах. Для досягнення цієї мети було поставлено такі завдання:

- дослідити існуючі застосунки кіномереж та сервісів, які надають послуги онлайн купівлі квитків у кінотеатрах
- розглянути, дослідити та застосувати на практиці навички у роботі зі Spring-фреймворком
- дослідити застосування різних технологій для проектування та реалізації веб-сайту
- розробити вимоги до функціоналу продукту
- побудувати модель бази даних, яка відповідатиме всім вимогам, що висуваються до застосунку
- розробити інтерфейс та дизайн веб-застосунку
- розробити власне сам веб-додаток

Об'єкт, методи й засоби розроблення. Об'єктом дослідження веб-додатку «A.Movie» є процес бронювання, купівлі квитків в кінотеатрі та ведення власного облікового запису для збереження переглянутих фільмів чи телесеріалів.

Предметом роботи є власний розроблений застосунок.

В якості інструменту створення програмного засобу було обрано IntelliJ IDEA IDE – інтегроване середовище розробки мовою програмування Java від компанії JetBrains [1]. В якості СУБД було обрано PostgreSQL.

Сфера застосування. Програмний продукт «A.Movie» можна використовувати у сфері кіно та кінотеатрів: для оцінювання переглянутих фільмів чи серіалів, пошуку відповідних фільмів, обміну з друзями своїми враженнями від перегляду тих чи інших картин та бронювання, купівля квитків в різних кінотеатрах різних міст.

РОЗДІЛ 1. ОГЛЯД ТА ПОРІВНЯННЯ НАЯВНИХ НА РИНКУ РІШЕНЬ

На сьогодні існує велика кількість веб-додатків, які пропонують послуги бронювання, купівлі квитків та додатків з великою базою даних фільмів та телесеріалів у вигляді соціальної мережі для кіноманів. Давайте розглянемо найпопулярніші з них, які надають ту чи іншу послугу.

1.1 IMDb

Internet Movie Database (IMDb [2]) - веб-сайт з вільно редагованою та найбільшою у світі базою даних про кінематограф. Станом на січень 2021 року, в базі зібрано інформацію про більш ніж 6,5 млн кінофільмів, телесеріалів та окремих їх серій, а також про 10,4 млн персон, пов'язаних з кіно та телебаченням, - акторів, режисерів, сценаристів та інших.

З 1998 базою даних IMDb володіє компанія Amazon. Майже вся інформація IMDb є у вільному доступі, сайт IMDb.com функціонує на основі вільного програмного забезпечення (Apache, Perl та ін.) [3].

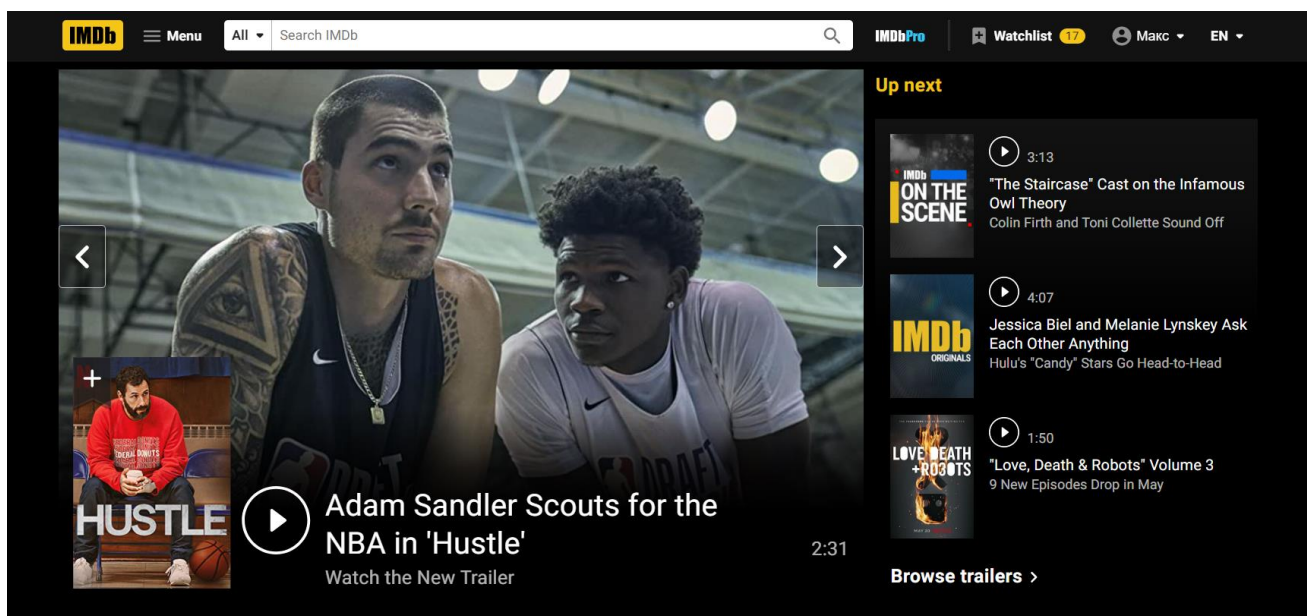


Рисунок 1 - Головна сторінка IMDb.com

Однією з головних особливостей IMDb є те, що редагувати інформацію в базі даних може будь-який користувач Інтернету, який пройшов реєстрацію та модерацію на сайті. Також серед переваг можна виділити простий у використанні і гарний інтерфейс, список кіноновин, який оновлюється щотижня, та детальна інформація про всі кіноелементи.

1.2 Multiplex

Multiplex [4] - це найбільша мережа кінотеатрів в Україні заснована у 2004 році. Окрім цього мережа має власний сайт, де можна в залежності від вибраного міста та кінотеатру бронювати та купувати квитки на сеанси кінофільмів. Сайт має мінімалістичний і простий в користуванні інтерфейс.

Бетмен

Вік: 12+
Рік: 2022
Оригінальна назва: The Batman
Режисер: Метт Ріва
Період прокату: 12.05.2022 - 18.05.2022
Мова: Українська мова
Жанр: Екшн, Кримінал, Детектив, Драма
Тривалість: 2:55
Виробництво: США
Студія: Warner Bros. Pictures, DC Comics

У головних ролях: Роберт Паттінсон Зої Кравіц Пол Дано Джеффі Райт Джон Туртурро Енді Серкіс Пітер Сарсгаард Колін Фаррелл Баррі Кеоган Алекс Фернс

Якщо в тебе залишились попередні активні квитки на фільм "Бетмен", звернись до нас на форму зворотнього зв'язку та отримай промокод на відкриті сеанси фільму "Бетмен" або на будь-який інший сеанс із репертуару.

Минуло майже 2 роки, відколи Бетмен з'явився на вулицях Готема. Самотній борець за правосуддя став утіленням розплати для мешканців міста. Міста, яке кишить корумпованими чиновниками і високопосадовцями. Там Брюс Вейн може довіряти лише двом людям: Альфреду Пенніворту і лейтенанту Джеймсу Гордону.

Розклад сеансів

Сб, 14 травня

Lavina IMAX Лазер

| | | | | |
|-----------|-----------|---------|-------|-------|
| 10:00 | 10:30 | 12:00 | 13:00 | 13:30 |
| ATMOSL... | LUX | | | |
| 14:00 | 14:30 | 15:00 | 15:30 | 15:50 |
| ATMOSL... | IMAXL2D | | LUX | |
| 16:30 | 17:00 | 17:30 | 18:00 | |
| | ATMOSL... | IMAXL2D | | |

ЦУМ

Немає сеансів у цей день

Respublika Park IMAX

| | | | | |
|-------|---------|-----------|-------|-------|
| 10:10 | 10:30 | 11:00 | 13:00 | 13:30 |
| LUX | IMAXL2D | CHILL OUT | | |
| 14:00 | 14:30 | 16:00 | 16:30 | 17:00 |
| | | | | |

Atmos ? IMAX Лазер 2D ?
LUX ? Chill out ?

Рисунок 2 - Сайт Multiplex

1.3 Деякі інші рішення та їх порівняння

Також існує багато інших конкурентів, які надають обидві ці послуги. Наприклад «Кінопошук» надає послуги онлайн кінотеатру за окрему підписку та можливість підписуватися на інших користувачів, але має менш інтуїтивний інтерфейс, застарілий дизайн на деяких сторінках та програє в кількості даних в порівнянні з IMDb. Letterboxd більше орієнтований на соціальну кіномережу, ніж

на велику базу даних фільмів. Наприклад, використовуючи цей додаток, легше взаємодіяти з іншими кіноманами та зручніше відстежувати фільми та телесеріали. Але він програє у кількості функціоналу. Серед сервісів, які надають послуги бронювання та купівлі квитків на кіносеанси, можна виділити «Планета кіно», «Оскар» та Cinema city. Всі вони відрізняються один від одного дизайном та невеликими відмінностями в функціоналі додатку.

Розглянувши ці застосунки, можна побачити, що вони надають один з двох варіантів послуг: або кімережі, або ж сервіси купівлі та бронювання квитків. Тому природним постає питання інтеграції подібних сервісів та, до того ж, їх збагачення, зокрема в напрямках, пов'язаних із сучасними тенденціями спілкування на основі месенджерів, наприклад, Telegram ботів.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ

2.1 Spring-фреймворк

Spring - один з найбільш широко використовуваних фреймворків для розробки інтерпрайз-додатків, що забезпечує продуману модель програмування та конфігурації. Метою створення даного фреймворку сприяло бажання спростити розробку додатків на популярному на той час Java EE (Java Platform, Enterprise Edition) [5] стеку технологій, який на той момент був дуже складний і не завжди зручний у використанні.

На відміну від інших фреймворків, Spring зосереджується на кількох областях програми та надає широкий спектр функцій. Однією з головних особливостей фреймворка Spring є ін'єкція залежностей. Це допомагає зробити речі простішими, дозволяючи нам розробляти слабо пов'язані програми.

Хоча фреймворк Spring зосереджується на наданні гнучкості, Spring Boot [6] має на меті скоротити довжину коду та надати найпростіший спосіб розробки веб-програми. Завдяки конфігурації анотацій і кодам за замовчуванням, Spring Boot скорочує час розробки програми. Це допомагає створити окрему програму з меншою або майже нульовою конфігурацією. Тому в якості серверної частини було прийнято рішення використати саме Spring Boot.

Архітектура Spring Boot. Spring Boot дотримується багаторівневої архітектури, в якій кожен рівень взаємодіє з іншими. Документація Spring Boot містить таке визначення: *«Spring Boot дозволяє легко створювати окрему, виробничу програму на основі Spring, яку можна «просто запусити»*. Основною метою Spring Boot є видалення XML і параметрів конфігурації на основі анотацій із програми.

Spring Boot складається з наступних чотирьох рівнів [7]:

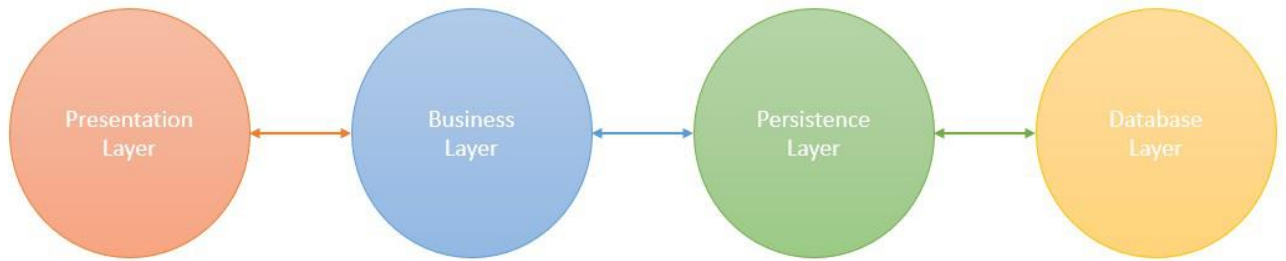


Рисунок 3 - Рівні Spring Boot

1. **Presentation Layer.** Рівень презентації є верхнім рівнем архітектури Spring Boot. Він складається з фронтенд частини програми. Він обробляє HTTP-запити та відповідає за перетворення параметра поля JSON в об'єкти Java і навпаки. Після того, як він виконує аутентифікацію запити, він передає його на наступний рівень, тобто бізнес-рівень.
2. **Business Layer.** Цей рівень містить бізнес-логіку, яка керує основними функціями програми. Подібно до прийняття рішень, обчислень, оцінок та обробки даних, що передаються між двома іншими рівнями.
3. **Persistence Layer.** Рівень збереження містить всю логіку зберігання бази даних. Він відповідає за перетворення бізнес-об'єктів у рядок бази даних і навпаки.
4. **Database Layer.** Рівень бази даних містить усі бази даних, такі як MySQL, PostgreSQL тощо. Цей рівень може містити декілька баз даних. Він відповідає за виконання операцій CRUD.

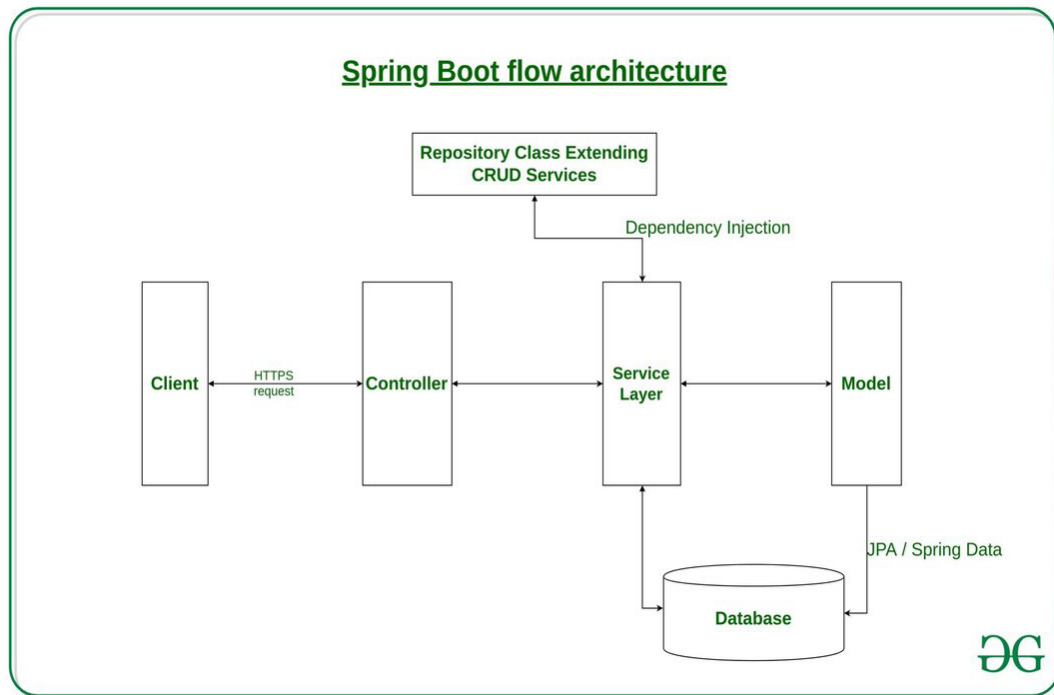


Рисунок 4 - Архітектура Spring Boot [7]

- Клієнт робить HTTP-запит (GET, PUT, POST тощо)
- HTTP-запит пересилається до контролера. Контролер відображає запит. Він обробляє дескриптори та викликає логіку сервера.
- Бізнес-логіка виконується на рівні сервісу. Spring Boot виконує всю логіку над даними бази даних, які зіставляються з класом моделі Spring Boot через Java Persistence Library(JPA).
- Контролер повертає представлення (JSP, Thymeleaf, тощо) для відтворення у браузері.

2.2 PostgreSQL

В ролі СУБД було вирішено взяти PostgreSQL.

PostgreSQL[8] – це відкрита, кросплатформова об’єктно-реляційна система управління базами даних, яка офіційно написана мовою програмування С. Вона підтримує мови запитів SQL і NoSQL, пропонує дані для безпечного зберігання та може відновлюватися щоразу, коли запит обробляється. Вона має MVCC

(контроль одночасного виконання кількох версій), що дозволяє кільком користувачам працювати в системі одночасно.

Переваги PostgreSQL перед MySQL [8]:

- PostgreSQL - це дуже безпечна, незалежна і багатофункціональна база даних з відкритим вихідним кодом.
- PostgreSQL підтримує багато розширених типів даних, таких як багатовимірні масиви, визначені користувачем типи тощо.
- PostgreSQL дуже добре дотримується стандартів SQL і підтримує «розширені» речі SQL, такі як віконні функції або загальні табличні вирази.
- Він забезпечує всю оптимізацію продуктивності, яка підтримується в Oracle і SQL Server.

2.3 Firebase API

В якості збереження зображень кінокартин, кінотеатрів, персон і користувачів було використано Firebase API, а саме Firebase Storage.

Firebase Storage [9] забезпечує надійне завантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firebase підтримується Google Cloud Storage.

Основна перевага і причина, з якої вибір впав на Firebase Storage - це те, що Firebase надає 10 гігабайтів сховища, яке можна використовувати безкоштовно без будь-яких обмежень на запити до цього сховища, на відміну від інших конкурентів. Також доволі зручно зберігати зображення з нашого проекту, так як все що потрібно – це зберігати унікальний ідентифікатор картини в базі даних, по якому можна отримати посилання на зображення і відобразити його на сторінках сайту. Тобто не потрібно кожен раз посилати запит на отримання зображень, що дає змогу швидко відобразити їх на HTML сторінках.

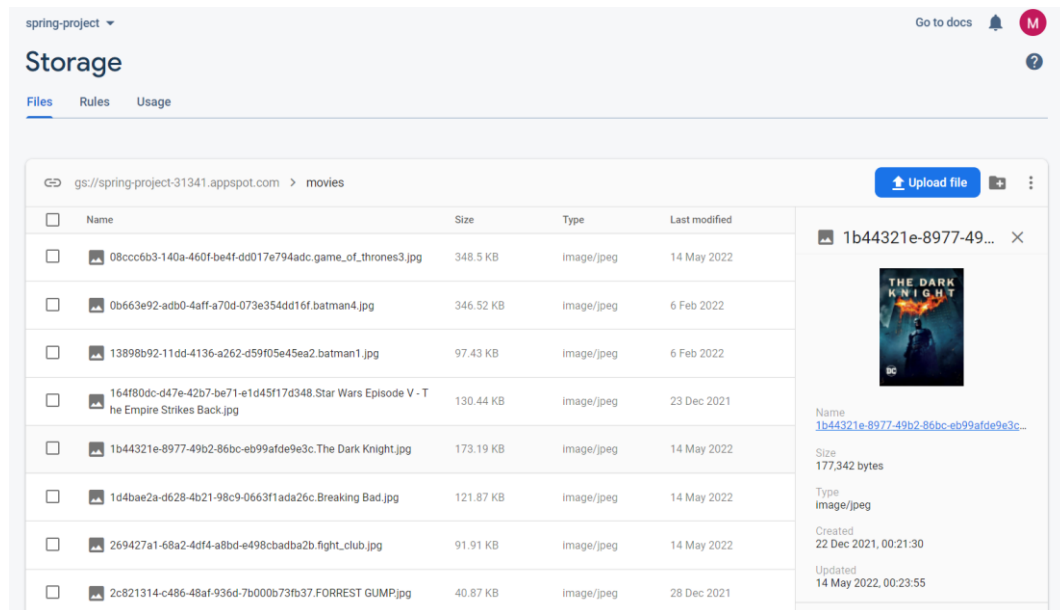


Рисунок 5 - Папка в Firebase Storage, в якій зберігаються постери фільмів

2.4 Google and Facebook APIs

У веб-додатку також реалізовано вхід за допомогою соціальних мереж, таких як Google та Facebook.

Для авторизації за допомогою Google акаунту було використано Google OAuth2 [10]. OAuth 2.0 дозволяє користувачам обмінюватися певними даними з програмою, зберігаючи при цьому свої імена користувачів, паролі та іншу інформацію в таємниці. Цей потік OAuth 2.0 призначений спеціально для авторизації користувачів. Для авторизації за допомогою облікового запису Facebook було обране відповідне API [11].

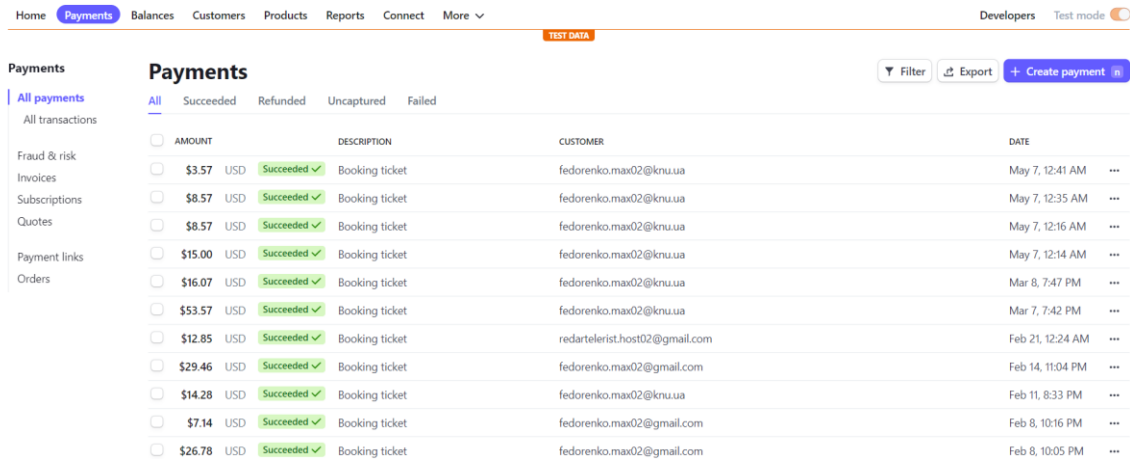
Також для відображення кінотеатрів на мапі було використано Maps JavaScript API. Maps JavaScript API [12] дозволяє налаштовувати карти з власним вмістом і зображеннями для відображення на веб-сторінках і мобільних пристроях.

2.5 Stripe

Stripe, Inc. [13] - ірландсько-американська компанія, що займається фінансовими послугами та програмним забезпеченням як послуга (SaaS). Компанія насамперед пропонує програмне забезпечення для обробки платежів та

API для веб-сайтів електронної комерції та мобільних додатків. Використовуючи Stripe, веб-розробники можуть інтегрувати платіжний процесинг у свій сайт без необхідності реєструвати рахунок мерчанта [14].

У власному проєкті за допомогою Stripe API можна обробляти платежі купівлі квитків (в тестовому режимі звичайно).



The screenshot shows the Stripe Payments interface. The main content is a table of payments, all of which are 'Booking ticket' transactions that have 'Succeeded'. The table includes columns for AMOUNT, DESCRIPTION, CUSTOMER, and DATE. The amounts range from \$3.57 to \$53.57 USD. The dates range from February 8 to May 7. The customer email addresses are either 'fedorenko.max02@knu.ua' or 'redartelister.host02@gmail.com'.

| AMOUNT | DESCRIPTION | CUSTOMER | DATE |
|-------------|----------------|--------------------------------|------------------|
| \$3.57 USD | Booking ticket | fedorenko.max02@knu.ua | May 7, 12:41 AM |
| \$8.57 USD | Booking ticket | fedorenko.max02@knu.ua | May 7, 12:35 AM |
| \$8.57 USD | Booking ticket | fedorenko.max02@knu.ua | May 7, 12:16 AM |
| \$15.00 USD | Booking ticket | fedorenko.max02@knu.ua | May 7, 12:14 AM |
| \$16.07 USD | Booking ticket | fedorenko.max02@knu.ua | Mar 8, 7:47 PM |
| \$53.57 USD | Booking ticket | fedorenko.max02@knu.ua | Mar 7, 7:42 PM |
| \$12.85 USD | Booking ticket | redartelister.host02@gmail.com | Feb 21, 12:24 AM |
| \$29.46 USD | Booking ticket | fedorenko.max02@gmail.com | Feb 14, 11:04 PM |
| \$14.28 USD | Booking ticket | fedorenko.max02@knu.ua | Feb 11, 8:33 PM |
| \$7.14 USD | Booking ticket | fedorenko.max02@gmail.com | Feb 8, 10:16 PM |
| \$26.78 USD | Booking ticket | fedorenko.max02@gmail.com | Feb 8, 10:05 PM |

Рисунок 6 - Список транзакцій купівлі білетів в Stripe API

2.6 Telegram

Telegram [15] - кросплатформова система миттєвого обміну повідомленнями (месенджер) з функціями VoIP, що дозволяє обмінюватися текстовими, голосовими та відеоповідомленнями, стікерами та фотографіями, файлами багатьох форматів. За допомогою ботів функціонал програми практично необмежений.

Варто зазначити, що було вирішено додатково до веб-додатку реалізувати телеграм бота в якості «родзинки» проєкту, оскільки телеграм на сьогодні один з найважливіших засобів комунікації. Для цього використовується Telegram Bot API - це інтерфейс на основі HTTP, створений для розробників, які прагнуть створювати ботів для Telegram. Telegram боти поділяються на два типи:

- long polling - ваш додаток сам опитуватиме сервер telegram на предмет наявності повідомлень з певною періодичністю, що доволі повільно.

- `webhook` - сервер telegram миттєво перенаправлятиме повідомлення на вказаний вами сервер.

Мною був обраний другий тип реалізації телеграм боту. Функціонал розробленого боту дозволяє пі'єднатися до облікового запису веб-застосунку та переглядати власні куплені квитки за весь час. Також за допомогою бота можна знайти сеанси фільмів, які йдуть в прокаті в цей час. Окрім цього при купівлі нового білета користувачу прийде відповідне повідомлення від бота в телеграм. Також реалізована система сповіщень від бота, яка повідомляє за дві години про твій сьогоднішній візит до кінотеатру [16].

2.7 FreeMarker

В якості фронтенд частини використовується Freemarker.

Apache FreeMarker [17] - це механізм шаблонів: бібліотека Java для створення текстового виводу (веб-сторінки HTML, електронні листи, файли конфігурації, вихідний код тощо) на основі шаблонів і змінених даних. Зазвичай для підготовки даних використовується мова програмування загального призначення (наприклад, Java). Потім Apache FreeMarker відображає підготовлені дані за допомогою шаблонів. У шаблоні ви зосереджуєтесь на тому, як представити дані, а поза шаблоном ви зосереджуєтесь на тому, які дані представити.

Кілька основних моментів та переваг FreeMarker:

- потужна мова шаблонів: умовні блоки, ітерації, присвоєння, рядкові й арифметичні операції та форматування, макроси та функції
- багатоцільовий і легкий: нульові залежності, будь-який вихідний формат, можна завантажувати шаблони з будь-якого місця (з можливістю підключення), багато варіантів конфігурації
- з урахуванням інтернаціоналізації/локалізації: чутливе до місцевої мови форматування числа та дати/часу, локалізовані варіації шаблонів.
- можливості обробки XML

- універсальна модель даних: об'єкти Java відкриваються шаблону як дерево змінних

Також для оформлення веб-сторінок, їх стилю використовується CSS. Для прискорення графічного інтерфейсу, надання йому динамічності, використовуються скрипти JavaScript.

РОЗДІЛ 3. ПРИЗНАЧЕННЯ І ЦІЛІ РОЗРОБКИ ЗАСТОСУНКУ

3.1 Призначення застосунку

Призначенням веб-системи «A.Movie» є реалізація сервісу бронювання та купівлі квитків, надання інформації про велику кількість фільмів, серіалів та персон, можливість оцінювання фільмів для збереження в своєму обліковому записі для того, щоб ділитися своїми враженнями та оцінками з друзями.

Для вирішення поставленої задачі було проведено:

- аналіз методів і моделей, що застосовуються для розв'язання задач створення комплексних веб-систем
- аналіз наявних програмних засобів кіномереж і сервісів купівлі квитків на кіносеанси
- проектування та програмну реалізацію системи «A.Movie»
- апробація розробленого веб-застосунку

3.2 Цілі створення застосунку

Веб-застосунок «A.Movie» розроблено з метою забезпечення:

- зручного пошуку бажаних до перегляду фільмів та серіалів
- слідкування за новинками та популярними кінокартинами
- перегляду детальної інформації кінофільмів та телесеріалів
- оцінювання та збереження переглянутих кінокартин в своєму обліковому записі для подальшого обміну враженнями та оцінками зі своїми друзями-підписниками
- слідкування за кіноактивністю своїх товаришів
- розвитку кінокультури шляхом обговорення кінофільмів у коментарях та залучення нових користувачів до сервісу
- бронювання та купівлі квитків на кіносеанси для гарного проведення часу та відпочинку від буденних справ

3.3 Архітектурні особливості системи

З огляду на поставлені задачі в системі «А.Movie» були виділені наступні функціональні підсистеми:

- підсистема власника, призначена для оперування адміністраторами
- підсистема адміністратора, призначена для контролю за веб-сайтом: додавання нових елементів, редагування та видалення існуючих, контролю за користувачами з можливістю блокування та видалення їх із системи, відповіді на надіслані запитання та інші.
- підсистема користувача, призначена для ведення облікового запису, можливості оцінювання кінокартин, додавання їх в списки улюблених та бажаних до перегляду та купівлі квитків в кінотеатрах.

Вимоги до функцій, які виконуються програмою

Перелік основних функцій адміністратора наведено в табл. 1.

Таблиця 1. Перелік функцій та задач адміністратора що підлягають автоматизації

| Функція | Задача |
|---------------------------------------|---|
| Управління користувачами | Перегляд користувачів, їх квитків, коментарів |
| | Редагування користувачів без ролей |
| | Видалення та блокування користувачів із системи |
| Управління кіно елементами | Створення, редагування, видалення кінокомпаній |
| | Створення, редагування, видалення кінокартин |
| | Створення, редагування, видалення персон |
| Управління кінотеатрами | Створення, редагування та видалення кінотеатрів |
| | Створення, редагування та видалення кінозалів |
| | Додавання сеансів в кінотеатри та їх видалення |
| | Перегляд всіх квитків куплених за весь час з можливістю видалення їх із системи |
| Управління комунікативними елементами | Перегляд та видалення коментарів до кінокартин |
| | Перегляд, видалення та можливість відповіді на надіслані запитання |

Перелік основних функцій користувача наведено в табл. 2.

Таблиця 2. Перелік функцій та задач користувача що підлягають автоматизації

| Функція | Задача |
|-----------------------------|---|
| Реєстрація | Введення короткої інформації про себе(логін та пароль) |
| | Введення повної інформації про себе та підтвердження акаунту по пошті |
| | Редагування профілю та зміна паролю |
| | Відновлення паролю по пошті |
| | Вихід із системи |
| Робота з кіномережею | Перегляд, пошук, фільтрація кінокартин |
| | Перегляд, пошук персон |
| | Додавання коментарів до кінокартин |
| | Оцінювання кінофільмів та серіалів |
| | Додання до списку улюблених та бажаних до перегляду |
| | Підписування на користувачів та перегляд їх списків збережених кінокартин |
| | Надання запитань до адміністрації сайту |
| Робота з онлайн кінотеатром | Перегляд та вибір кінотеатрів |
| | Перегляд та вибір сеансів |
| | Вибір місць та їх резервування |
| | Оформлення замовлення для придбання квитків |
| Робота з телеграм ботом | Приєднання телеграму до облікового запису веб застосунка |
| | Перегляд куплених квитків |
| | Пошук сеансів |
| | Нагадування про візит до кінотеатру |

Технічні вимоги до застосунку

В результаті виконання практичної частини випускної роботи було розроблено проект, що забезпечує реалізацію наступних вимог:

- застосунок розроблений використовуючи технологію Spring Boot
- джерелом даних є СУБД PostgreSQL

- додаток має адміністративну панель, яка чітко відділена від основного сайту
- у застосунку наявна реєстрація з підтвердженням створеного облікового запису по пошті
- у застосунку реалізована система авторизації і автентифікації
- на сторінках наявна пагінація, фільтрація та рядок пошуку
- на всіх сторінках з формами присутня валідація як на стороні клієнта, так і на стороні сервера
- застосунок коректно реагує на помилки і виключення різного роду
- веб-додаток має простий і інтуїтивний у використанні інтерфейс
- сайт коректно відображається в інтернет-браузерах Google Chrome, Microsoft Edge, Mozilla Firefox, Opera.
- Сайт коректно відображається на мобільних пристроях

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

4.1 Реалізація бази даних

4.1.1 Логічна структура бази даних

База даних розробленого веб-застосунку складається з 21 основної таблиці, структура яких зображена на діаграмі класів (див. додаток А), та 9 таблиць, які відображають зв'язок «багато до багатьох». Опис кожної з таблиць наведено у табл. 3.

Таблиця 3. Короткий опис таблиць бази даних

| Номер | Таблиця | Опис |
|-------|----------------------|--|
| 1 | user | Таблиця для збереження інформації про користувачів. |
| 2 | confirmation_token | Таблиця для збереження інформації про токен для підтвердження облікового запису по пошті |
| 3 | password_reset_token | Таблиця для збереження інформації про токен для відновлення паролю |
| 4 | company | Таблиця для збереження інформації про кінокомпанії |
| 5 | movie | Таблиця для збереження інформації про кінокартини |
| 6 | image_movie | Таблиця для збереження інформації про зображення кінокартин |
| 7 | star | Таблиця для збереження інформацію про персон (акторів чи режисерів) |
| 8 | cinema | Таблиця для збереження інформації про кінотеатр |
| 9 | image_cinema | Таблиця для збереження інформації про зображення кінотеатрів |
| 10 | hall | Таблиця для збереження інформації про кінозали |
| 11 | row | Таблиця для збереження інформації про ряди кінозалу |
| 12 | seat | Таблиця для збереження інформації про місця кінозалу |
| 13 | seance | Таблиця для збереження інформації про сеанси фільмів в кінотеатрах |

| | | |
|----|--------------------|--|
| 14 | seat_reserved | Таблиця для збереження інформації про зарезервовані місця |
| 15 | reservation | Таблиця для збереження інформації про бронювання |
| 16 | ticket | Таблиця для збереження інформації про продані квитки |
| 17 | comment | Таблиця для збереження інформації про коментарі користувачів до кінокартин |
| 18 | contact_us_message | Таблиця для збереження інформації про надіслані запитання від користувачів |
| 19 | rating | Таблиця для збереження інформації про оцінки користувачів до кінокартин |
| 20 | favourite | Таблиця для збереження інформації про улюблені кінокартини користувачів |
| 21 | watch_list | Таблиця для збереження інформації про список бажаного до перегляду користувача |
| 22 | user_role | Таблиця для збереження інформації про ролі користувачів |
| 23 | user_subscriptions | Таблиця для збереження інформації про підписки на інших користувачів |
| 24 | movie_cast | Таблиця для збереження інформації про акторів фільмів |
| 25 | movie_genres | Таблиця для збереження інформації жанрів фільмів |
| 26 | movie_companies | Таблиця для збереження інформації кінокомпаній фільмів |
| 27 | movie_countries | Таблиця для збереження інформації країн виробників фільмів |
| 28 | movie_directors | Таблиця для збереження інформації про режисерів фільмів |
| 29 | comment_likes | Таблиця для збереження інформації користувачів, які лайкнули коментар |
| 30 | comment_dislikes | Таблиця для збереження інформації користувачів, які дизлайкнули коментар |

4.1.2 Опис таблиць бази даних програми

Таблиці складають основу бази даних – саме в них зберігаються всі дані. Передусім, було спроектовано структуру кожної таблиці. У таблицях 4-22 наведено детальну інформацію про структуру основних таблиць БД (ім'я, тип та опис поля).

Таблиця 4. Опис таблиці user

| Атрибут | Тип | Опис |
|----------------|------------|---|
| id | bigint | Ідентифікатор користувача |
| username | varchar | Унікальний юзернейм |
| first_name | varchar | Ім'я користувача |
| last_name | varchar | Прізвище користувача |
| email | varchar | Унікальна електронна пошта користувача |
| password | varchar | Пароль користувача |
| birthday | date | Дата народження |
| locked | boolean | Заблокований чи ні |
| active | boolean | Пройшов підтвердження по пошті чи ні |
| created_at | timestamp | Дата реєстрації |
| updated_at | timestamp | Дата останнього оновлення облікового запису |
| image_name | varchar | Унікальна назва зображення профілю |
| image_url | varchar | Посилання на зображення |
| gender | varchar | Стать |
| telegram_id | int | Ідентифікатор телеграм акаунта |

Таблиця 5. Опис таблиці confirmation_token

| Атрибут | Тип | Опис |
|----------------|------------|---------------------------|
| id | bigint | Ідентифікатор токена |
| token | varchar | Токен |
| created_at | timestamp | Дата створення |
| user_id | bigint | Ідентифікатор користувача |

Таблиця 6. Опис таблиці password_reset_token

| Атрибут | Тип | Опис |
|----------------|------------|--------------------------------|
| id | bigint | Ідентифікатор токена |
| token | varchar | Токен |
| created_at | timestamp | Дата створення |
| expired_at | timestamp | Дата закінчення терміну токена |
| user_id | bigint | Ідентифікатор користувача |

Таблиця 7. Опис таблиці company

| Атрибут | Тип | Опис |
|---------|---------|----------------------------|
| id | bigint | Ідентифікатор кінокомпанії |
| name | varchar | Назва кінокомпанії |

Таблиця 8. Опис таблиці movie

| Атрибут | Тип | Опис |
|-------------|---------|---|
| id | bigint | Ідентифікатор кінокартини |
| title | varchar | Назва кінокартини |
| description | varchar | Короткий опис |
| year | int | Рік випуску |
| budget | int | Бюджет (в доларах) |
| box_office | int | Касові збори (в доларах) |
| release | date | Дата виходу |
| time | int | Час тривалості (в хвиликах) |
| poster_name | varchar | Унікальна назва зображення постера |
| poster_url | varchar | Посилання на зображення постера |
| trailer | varchar | Посилання на трейлер |
| status | boolean | Статус видимості |
| mpaa | varchar | Рейтинг кінокартини щодо цільової аудиторії |
| category | varchar | Категорія (фільм чи серіал) |

Таблиця 9. Опис таблиці movie_cast

| Атрибут | Тип | Опис |
|----------|--------|----------------------|
| movie_id | bigint | Ідентифікатор фільму |
| star_id | bigint | Ідентифікатор актора |

Таблиця 10. Опис таблиці image_movie

| Атрибут | Тип | Опис |
|------------|---------|----------------------------|
| id | bigint | Ідентифікатор зображення |
| image_name | varchar | Унікальна назва зображення |
| image_url | varchar | Посилання на зображення |
| movie_id | bigint | Ідентифікатор фільму |

Таблиця 11. Опис таблиці star

| Атрибут | Тип | Опис |
|-------------|---------|------------------------------------|
| id | bigint | Ідентифікатор персони |
| first_name | varchar | Ім'я персони |
| last_name | varchar | Прізвище персони |
| gender | varchar | Стать |
| birthday | date | Дата дня народження |
| birth_place | varchar | Опис місця народження |
| death | date | Дата смерті (якщо є) |
| image_name | varchar | Унікальна назва зображення персони |
| image_url | varchar | Посилання на зображення |
| status | boolean | Статус видимості |

Таблиця 12. Опис таблиці comment

| Атрибут | Тип | Опис |
|------------|-----------|---------------------------------------|
| id | bigint | Ідентифікатор коментаря |
| text | varchar | Текст коментаря |
| created_at | timestamp | Дата створення |
| movie_id | bigint | Ідентифікатор фільму |
| user_id | bigint | Ідентифікатор користувача |
| parent_id | bigint | Ідентифікатор батьківського коментаря |

Таблиця 13. Опис таблиці contact_us_message

| Атрибут | Тип | Опис |
|------------|-----------|------------------------------|
| id | bigint | Ідентифікатор повідомлення |
| text | varchar | Текст повідомлення |
| created_at | timestamp | Дата створення |
| name | bigint | Ім'я відправника |
| email | bigint | Електронна пошта відправника |

Таблиця 14. Опис таблиці rating

| Атрибут | Тип | Опис |
|---------|--------|----------------------|
| id | bigint | Ідентифікатор оцінки |
| value | int | Оцінка (від 1 до 10) |

| | | |
|------------|-----------|---------------------------|
| movie_id | bigint | Ідентифікатор кінокартини |
| user_id | bigint | Ідентифікатор користувача |
| created_at | timestamp | Дата додання оцінки |
| updated_at | timestamp | Дата оновлення оцінки |

Таблиця 15. Опис таблиці cinema

| Атрибут | Тип | Опис |
|------------|---------|--|
| id | bigint | Ідентифікатор кінотеатру |
| name | varchar | Назва кінотеатру |
| city | varchar | Місто |
| address | varchar | Адреса кінотеатру |
| phone | varchar | Мобільний телефон кінотеатру |
| latitude | double | Координата широти, де знаходиться кінотеатр |
| longitude | double | Координата довготи, де знаходиться кінотеатр |
| rating | double | Рейтинг кінотеатру |
| image_name | varchar | Унікальна назва зображення |
| image_url | varchar | Посилання на зображення кінотеатру |
| active | boolean | Статус видимості |

Таблиця 16. Опис таблиці hall

| Атрибут | Тип | Опис |
|-----------|---------|--------------------------|
| id | bigint | Ідентифікатор залу |
| name | varchar | Назва залу |
| num_seats | int | Кількість місць в залі |
| cinema_id | bigint | Ідентифікатор кінотеатру |
| active | boolean | Статус видимості |

Таблиця 17. Опис таблиці row

| Атрибут | Тип | Опис |
|-----------|---------|------------------------|
| id | bigint | Ідентифікатор ряду |
| name | varchar | Назва ряду |
| num_seats | int | Кількість місць в ряді |
| hall_id | bigint | Ідентифікатор залу |
| type | varchar | Тип ряду |

Таблиця 18. Опис таблиці seat

| Атрибут | Тип | Опис |
|---------|---------|---------------------|
| id | bigint | Ідентифікатор місця |
| number | int | Номер місця |
| row_id | bigint | Ідентифікатор ряду |
| type | varchar | Тип місця |

Таблиця 19. Опис таблиці seance

| Атрибут | Тип | Опис |
|------------|---------|-----------------------|
| id | bigint | Ідентифікатор сеансу |
| hall_id | bigint | Ідентифікатор залу |
| movie_id | bigint | Ідентифікатор фільму |
| date | date | Дата сеансу |
| start_time | time | Початок показу сеансу |
| end_time | time | Кінець показу сеансу |
| price | int | Ціна за квиток |
| available | boolean | Статус доступності |

Таблиця 20. Опис таблиці reservation

| Атрибут | Тип | Опис |
|-----------|-----------|---------------------------------|
| id | bigint | Ідентифікатор бронювання |
| seance_id | bigint | Ідентифікатор сеансу |
| user_id | bigint | Ідентифікатор користувача |
| created | timestamp | Дата створення бронювання |
| expired | timestamp | Дата закінчення бронювання |
| session | bigint | Ідентифікатор сесії користувача |
| active | boolean | Статус активності |

Таблиця 21. Опис таблиці seat_reserved

| Атрибут | Тип | Опис |
|----------------|--------|-------------------------------------|
| id | bigint | Ідентифікатор |
| seat_id | bigint | Ідентифікатор зарезервованого місця |
| reservation_id | bigint | Ідентифікатор бронювання |
| price | int | Ціна зарезервованого місця |

Таблиця 22. Опис таблиці ticket

| Атрибут | Тип | Опис |
|----------------|-----------|--|
| number | varchar | Унікальний номер квитка |
| created_at | timestamp | Дата купівлі |
| date | timestamp | Дата сеансу |
| hall_id | bigint | Ідентифікатор залу |
| movie_id | bigint | Ідентифікатор фільму |
| price | int | Ціна за квиток |
| seats | varchar | Місця, які було куплено |
| email | varchar | Електронна пошта покупця |
| first_name | varchar | Ім'я покупця |
| last_name | varchar | Прізвище покупця |
| phone | varchar | Мобільний телефон покупця |
| user_id | bigint | Ідентифікатор користувача (можливо порожнє значення) |
| reservation_id | bigint | Ідентифікатор бронювання (можливо порожнє значення) |

4.2 Розробка застосунку

Для підключення Java додатку до бази даних та роботою з нею використовується бібліотека Spring Data JPA [18]. Java Persistence API надає специфікацію для збереження, читання та керування даними з вашого об'єкта Java до реляційних таблиць у базі даних. Spring Data є частиною Spring Framework. Метою абстракції репозиторію Spring Data є суттєве зменшення кількості шаблонного коду, необхідного для реалізації рівнів доступу до даних.

Spring Data JPA не є постачальником JPA. Це бібліотека/фреймворк, який додає додатковий шар абстракції у верхній частині нашого постачальника JPA. В якості реалізації JPA використовується Hibernate.

Hibernate [19] - це інструмент ORM на основі Java, який забезпечує структуру для відображення об'єктів домену програми з таблицями реляційної бази даних і навпаки. Hibernate надає еталонну реалізацію Java Persistence API, що робить його чудовим вибором як інструмент ORM з перевагами слабкого з'єднання.

Для доступу до даних використовуються репозиторії Spring Data JPA. Запити JPA створюються автоматично з імен ваших методів. Для генерації запитів з власною логікою використовується JPQL [20]. JPQL - це потужна мова запитів, яка дозволяє визначати запити до бази даних на основі моделі вашої сутності. Її структура та синтаксис дуже схожі на SQL. JPQL використовує об'єктну модель сутності замість таблиць бази даних для визначення запиту. Це робить його дуже зручним для розробників Java, але БД все ще використовує мову запитів SQL. Тому як раз Hibernate перетворює запити JPQL в SQL.

В якості реалізації бізнес логіки використовуються сервіси. Всього в додатку реалізовано 24 класи сервісів.

В якості об'єктів, які передаються у відображення та приймаються контролерами використовуються DTO [21]. Шаблон проектування об'єкта передачі даних (DTO) – це часто використовуваний шаблон проектування. В основному він використовується для передачі даних із кількома атрибутами за один кадр від клієнта до сервера, щоб уникнути кількох викликів на віддалений сервер. Ще одна перевага використання DTO, полягає в тому, що вони можуть допомогти приховати деталі реалізації об'єктів домену (об'єктів JPA). Відкриття об'єктів через кінцеві точки може стати проблемою безпеки, якщо ми не ретельно обробляємо, які властивості можна змінити за допомогою яких операцій.

Контролери відповідають за обробку запитів користувачів та передачу даних модулю представлення для обробки до сервісів. В наступній таблиці описано всі контролери і за що вони відповідають.

Таблиця 23. Опис контролерів

| Номер | Назва контролеру | Опис |
|-------|-------------------------|--|
| 1 | RegistrationController | Відповідає за реєстрацію нового користувача та підтвердження облікового запису по пошті |
| 2 | PasswordResetController | Відповідає за відновлення паролю по пошті |
| 3 | ProfileController | Відповідає за перегляд та редагування інформації про себе, зміну паролю, перегляд власних квитків та функцію підписки та відписки на/від користувачів. |

| | | |
|----|-----------------------|---|
| 4 | UserController | Відповідає за відображення профілю користувача: коротку інформацію про нього, його оцінки, улюблені фільми, список бажаного до перегляду, коментарі, а також список його підписок та підписників. |
| 5 | AdminMainController | Відповідає за відображення головної сторінки для адміністраторів, основної статистики по сайту |
| 6 | CompanyController | Відповідає за CRUD кінокомпаній |
| 7 | AdminMovieController | Відповідає за CRUD кінокартин |
| 8 | AdminStarController | Відповідає за CRUD персон |
| 9 | AdminCinemaController | Відповідає за CRUD кінотеатрів |
| 10 | AdminHallController | Відповідає за CRUD кінозалів |
| 11 | AdminSeanceController | Відповідає за CRUD сеансів |
| 12 | AdminUserController | Відповідає за перегляд, редагування, видалення та блокування користувачів з роллю «User». |
| 13 | CommentController | Відповідає за перегляд та видалення коментарів |
| 14 | AdminTicketController | Відповідає за перегляд та видалення квитків |
| 15 | ContactUsController | Відповідає за перегляд повідомлень, надісланих від користувачів, їх видалення та можливість відповідати на повідомлення адміністратором |
| 16 | CinemaController | Відповідає за відображення списку всіх кінотеатрів та сторінки конкретного кінотеатру з інформацією про нього |
| 17 | SeanceController | Відповідає за відображення сеансу, доступних місць та можливість зарезервувати місця |
| 18 | ReservationController | Відповідає за перегляд, скасування бронювання та купівлю квитків |
| 19 | MainController | Відповідає за відображення головної сторінки, пошуку та «contact us» сторінки |
| 20 | MovieController | Відповідає за відображення списку кінокартин з можливістю фільтрації та відображення детальної сторінки фільму з переглядом інформації про нього та можливістю залишати коментарі. |
| 21 | RatingController | Відповідає за виставлення оцінок фільмам, їх видалення, додання фільмів до улюблених та до бажаних для авторизованих користувачів |

| | | |
|----|----------------|--|
| 22 | StarController | Відповідає за відображення списку персон та сторінки конкретної персони з її фільмографією за весь час |
|----|----------------|--|

4.3 Сценарій використання веб-додатку

4.3.1 Сценарій адміністратора

При успішній авторизації в ролі адміністратора, користувач може перейти до адміністративної панелі. Головна сторінка відображає загальну статистику по всьому сайту.

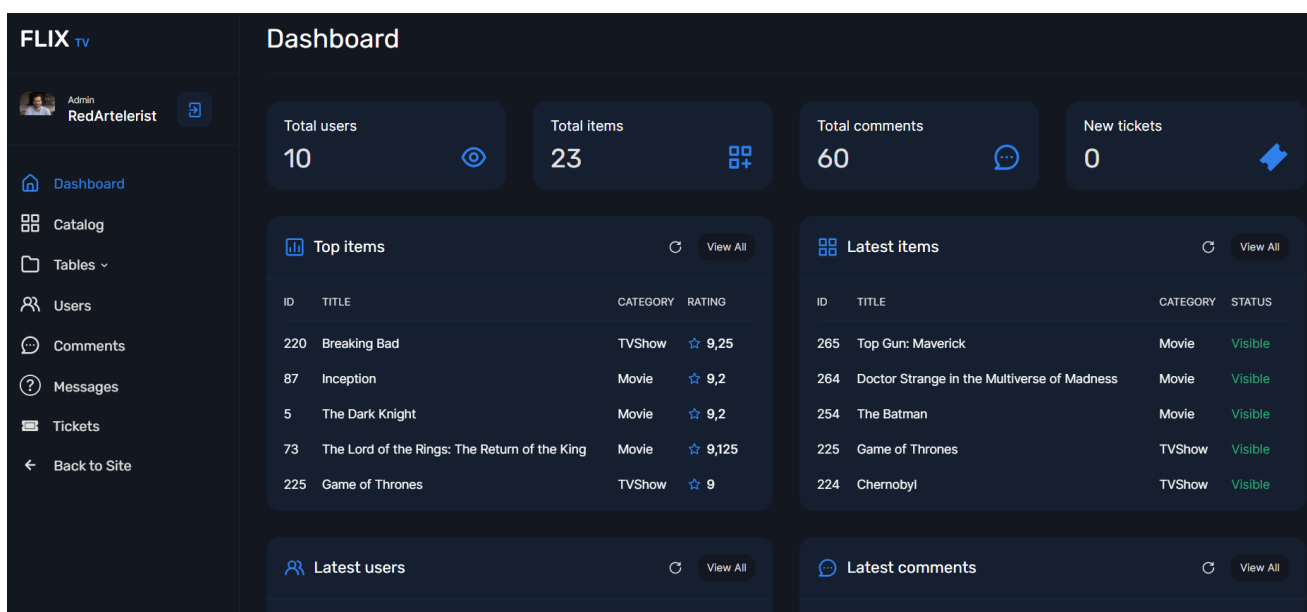
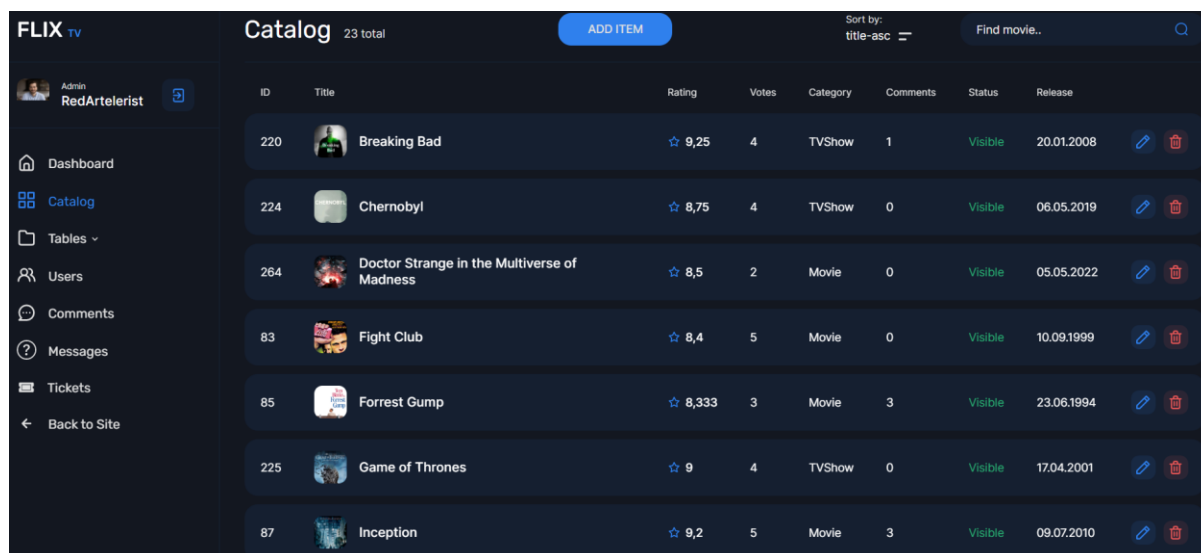


Рисунок 7 - Головна сторінка адміністративної панелі

Далі адміністратор може перейти до каталогу для перегляду всіх фільмів та серіалів. На сторінці відображається коротка інформація про фільми, також є можливість сортувати по деяким полям, наявний пошук по назві та пагінація. Ми можемо перейти до конкретного елемента для перегляду повної інформації з можливістю редагування або видалити конкретний елемент чи додати новий.



FLIX TV Catalog 23 total

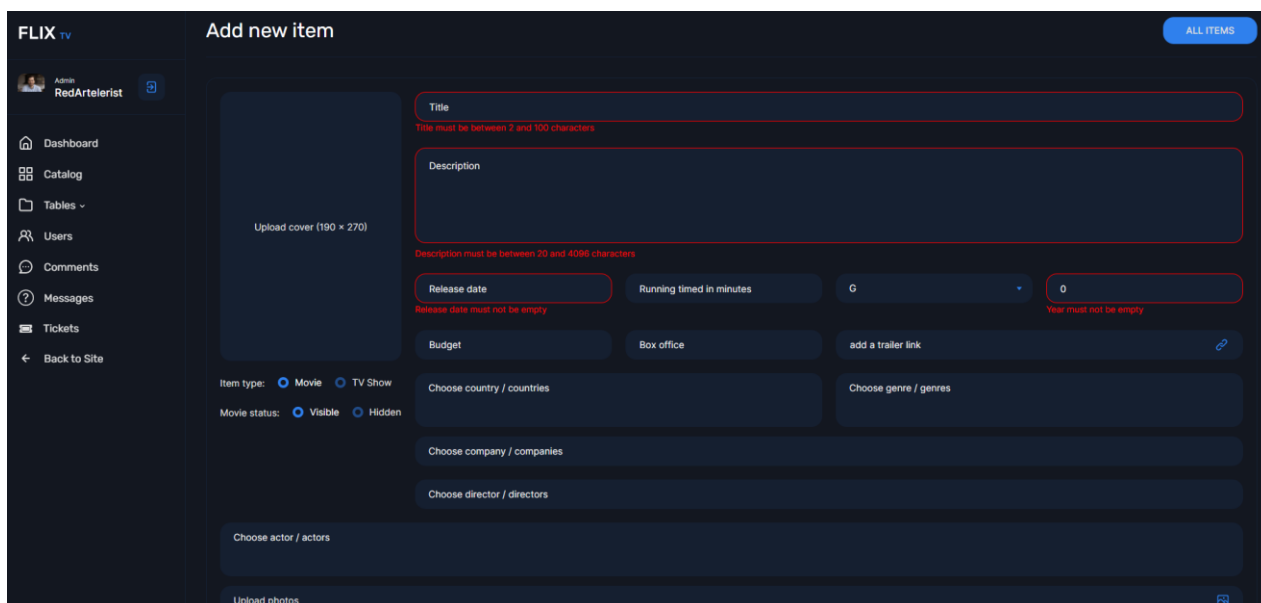
Sort by: title-asc

Find movie..

| ID | Title | Rating | Votes | Category | Comments | Status | Release |
|-----|---|---------|-------|----------|----------|---------|------------|
| 220 | Breaking Bad | ☆ 9,25 | 4 | TVShow | 1 | Visible | 20.01.2008 |
| 224 | Chernobyl | ☆ 8,75 | 4 | TVShow | 0 | Visible | 06.05.2019 |
| 264 | Doctor Strange in the Multiverse of Madness | ☆ 8,5 | 2 | Movie | 0 | Visible | 05.05.2022 |
| 83 | Fight Club | ☆ 8,4 | 5 | Movie | 0 | Visible | 10.09.1999 |
| 85 | Forrest Gump | ☆ 8,333 | 3 | Movie | 3 | Visible | 23.06.1994 |
| 225 | Game of Thrones | ☆ 9 | 4 | TVShow | 0 | Visible | 17.04.2001 |
| 87 | Inception | ☆ 9,2 | 5 | Movie | 3 | Visible | 09.07.2010 |

Рисунок 8 - Каталог фільмів та серіалів

При виборі додати новий елемент треба заповнити форму. На всіх формах в адміністративній панелі наявна валідація.



FLIX TV Add new item

ALL ITEMS

Admin RedArterlist

Upload cover (190 × 270)

Title
Title must be between 7 and 100 characters

Description
Description must be between 20 and 4096 characters

Release date
Release date must not be empty

Running time in minutes
Year must not be empty

Budget

Box office

add a trailer link

Item type: Movie TV Show

Movie status: Visible Hidden

Choose country / countries

Choose genre / genres

Choose company / companies

Choose director / directors

Choose actor / actors

Upload photos

Рисунок 9 - Додавання нового фільму чи серіалу з відповідною валідацією

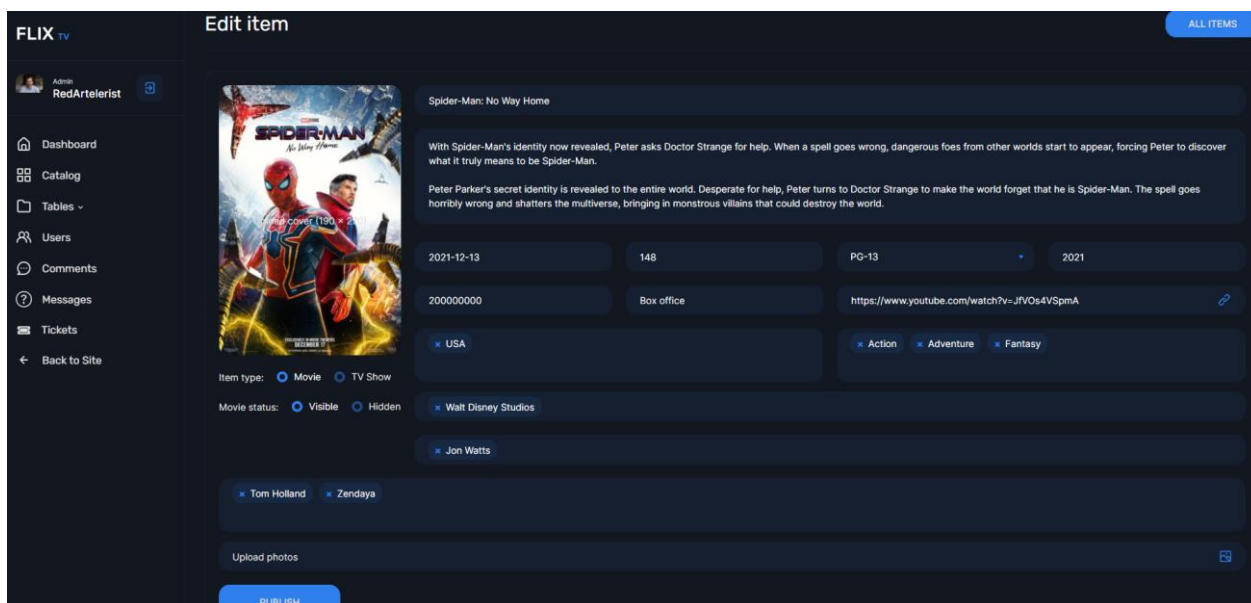


Рисунок 10 - Редагування фільму

Також окремо на сторінці з редагуванням ми можемо видалити будь-яке зображення з галереї.

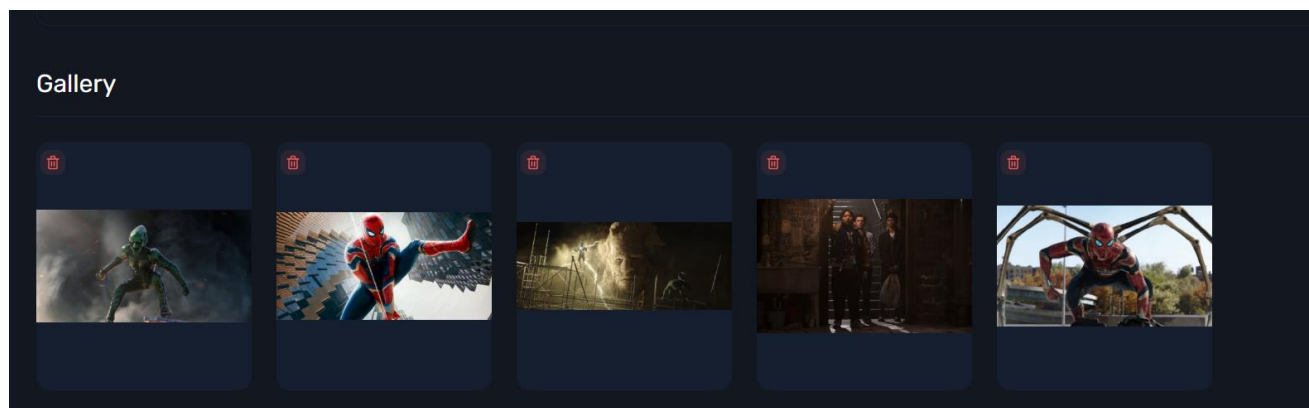


Рисунок 11 - Галерея фільму

Далі можна перейти до списку персон та кінокомпаній з можливістю створення, редагування та видалення аналогічно до кінокартин. Теж саме можна робити з кінотеатрами з одною відмінністю: при створенні та редагуванні можна скористатися картою для збереження координат місцезнаходження кінотеатру.

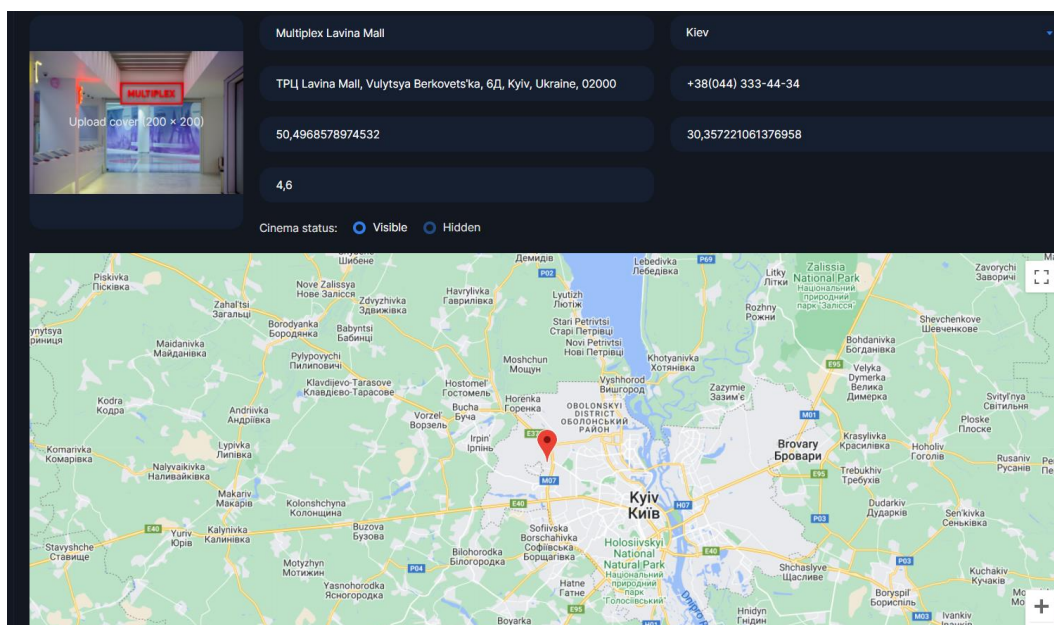


Рисунок 12 - Форма редагування кінотеатру

Також на сторінці редагування кінотеатру наявний список всіх кінозалів з можливістю редагування та видалення їх чи створення нових. Можна переглянути список всіх сеансів в даному кінотеатрі, додати новий сеанс чи редагувати або видалити існуючі. Після закінчення сеансів вони автоматично видаляються із системи.

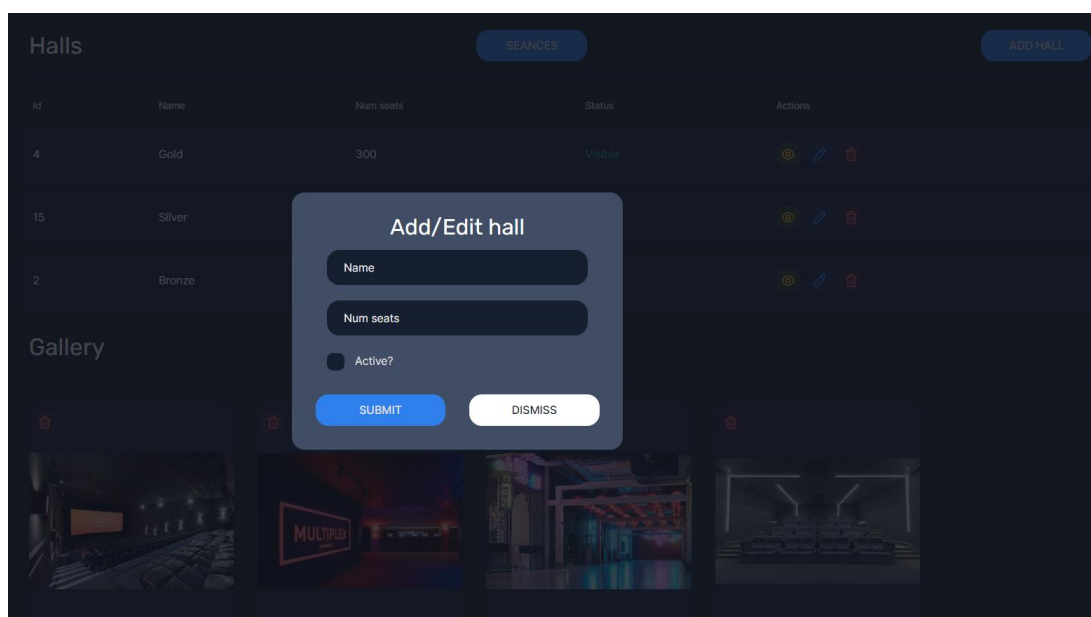
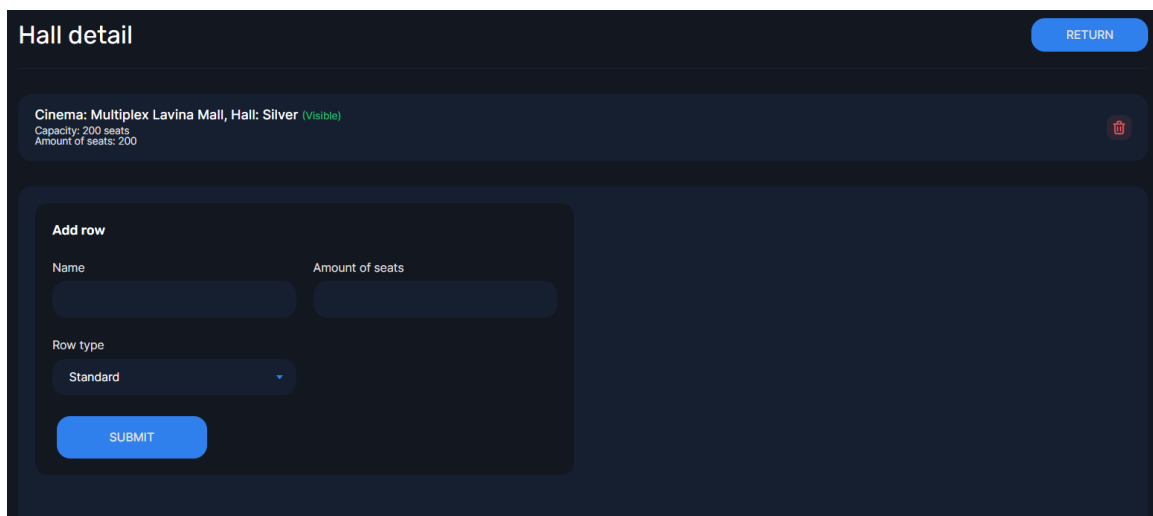


Рисунок 13 - Список кінозалів та форма додання нового залу

Ми можемо перейти до конкретного кінозалу та переглянути його план відповідно до типу місць. Також можна додати новий ряд, якщо є можливість, або видалити та редагувати вже існуючі.



The screenshot shows a dark-themed interface for a cinema hall. At the top, it says "Hall detail" and "RETURN". Below that, it displays "Cinema: Multiplex Lavina Mall, Hall: Silver (Visible)" with a trash icon. The main content is a form titled "Add row" with two input fields for "Name" and "Amount of seats", a "Row type" dropdown menu set to "Standard", and a "SUBMIT" button.

Рисунок 14 - Форма додавання ряду

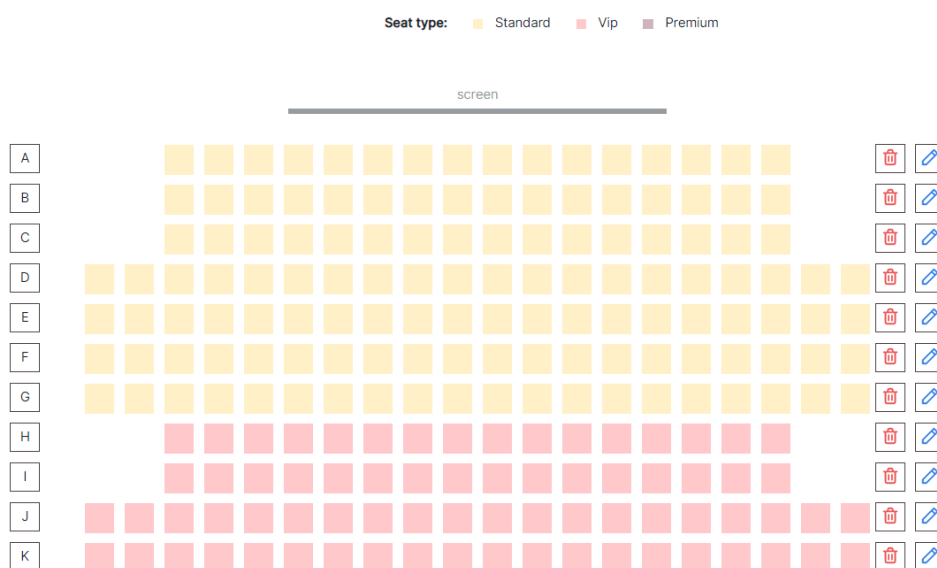


Рисунок 15 - План кінозалу

Далі можна переглянути список користувачів та редагувати їх інформацію, видалити чи заблокувати, якщо вони мають виключно роль «User». Також в системі є роль власника і відповідно єдиний користувач до неї, який може

оперувати адміністраторами. Таким чином він може надавати користувачам або відбирати у них роль адміністраторів.

| Id | Basic info | Username | Comments | Status | Active | Created date | Updated date | Actions |
|----|---|----------------------|----------|----------|--------|------------------|------------------|---------|
| 25 | Maksym Fedorenko redartelerist.host02@gmail.com USER | redartelerist.host02 | 0 | Approved | Yes | 21.02.2022 21:04 | 16.05.2022 22:29 | 🔒 ✎ 🗑️ |
| 23 | Max Fedor maks.fedor6110@gmail.com USER | maks.fedor | 0 | Approved | Yes | 21.02.2022 00:45 | 16.05.2022 22:29 | 🔒 ✎ 🗑️ |
| 17 | Willam Defo nepig45450@ztymm.com USER | nepig | 0 | Approved | Yes | 24.12.2021 00:28 | 04.03.2022 21:17 | 🔒 ✎ 🗑️ |
| 16 | Araina Grande kokes15682@wolffpat.com USER | kokes | 0 | Approved | Yes | 19.12.2021 19:05 | 20.12.2021 17:55 | 🔒 ✎ 🗑️ |
| 15 | Daria Kudanova elitki8806@ztymm.com USER | user1 | 0 | Approved | Yes | 18.12.2021 01:23 | 18.12.2021 21:13 | 🔒 ✎ 🗑️ |
| 14 | Robin Williams repet18062@videour.com USER | repet | 0 | Approved | Yes | 15.12.2021 18:04 | 18.12.2021 23:19 | 🔒 ✎ 🗑️ |
| 6 | Denis Postilyak feyox12925@mediafate.com ADMIN, USER | Denis | 0 | Approved | Yes | 15.12.2021 00:54 | 15.12.2021 00:55 | 👁️ |
| 5 | Max Fedos fed.max020401@gmail.com USER | Maksym | 0 | Approved | Yes | 15.12.2021 00:52 | 19.02.2022 22:24 | 🔒 ✎ 🗑️ |

Рисунок 16 - Список користувачів.

Аналогічно до всіх попередніх функцій можна переглядати списки коментарів, запитань та квитків, видаляти їх. Для запитань є можливість надіслати відповідь на пошту після чого запитання автоматично видалиться із системи.

4.3.2 Сценарій звичайного користувача

При переході на головну сторінку ми можемо побачити фільми, які зараз йдуть у прокаті, або можемо скористатися розширеним пошуком.

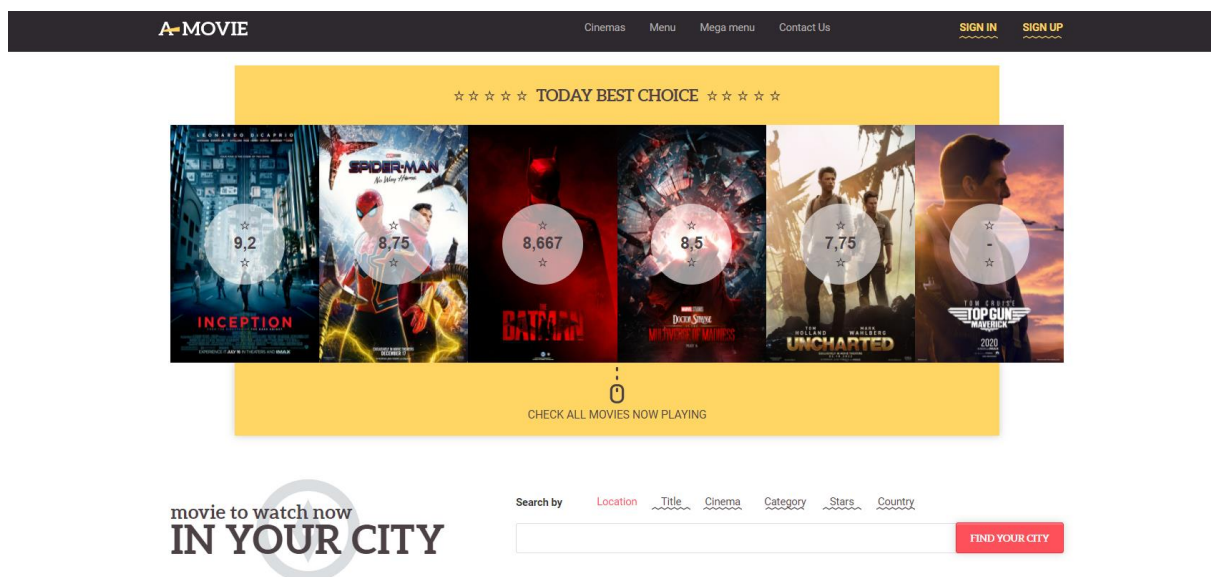
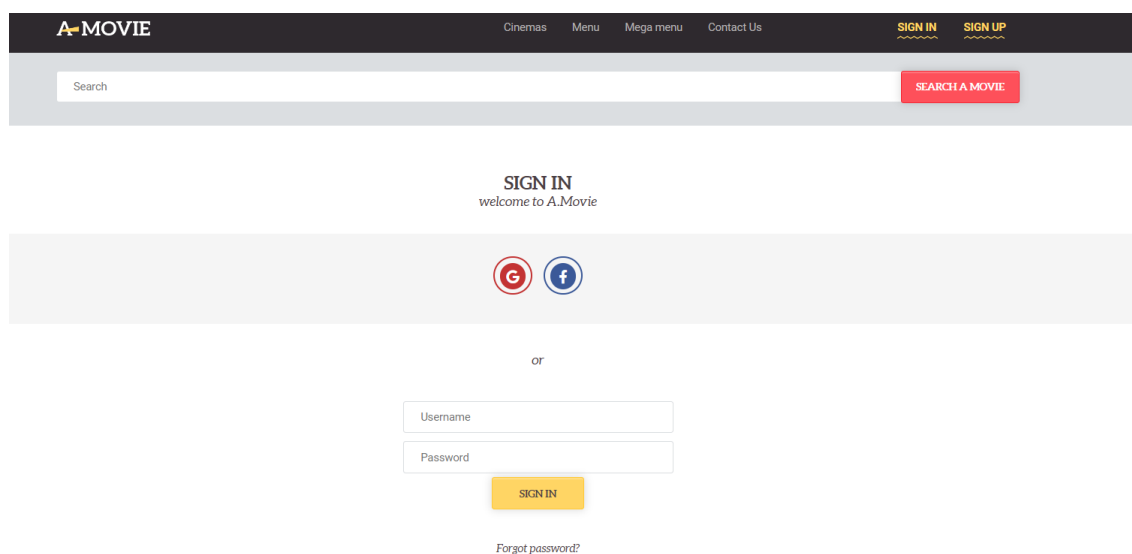


Рисунок 17 - Головна сторінка сайту «A.Movie»

Далі у нас є можливість продовжити як анонімний користувач або авторизуватися чи зареєструвати новий обліковий запис. При реєстрації треба заповнити декілька полів, при чому поля пошти та юзернейму повинні бути унікальними в системі. Після чого вам надійде повідомлення на пошту для підтвердження свого акаунта. І тільки після того, як перейти за ним, ваш обліковий запис стане активованим і буде можливість авторизуватися.

Рисунок 18 - Форма реєстрації

Також наявна можливість увійти за допомогою соціальної мережі Google чи Facebook. Вона під'єднається до існуючого акаунту з такою ж електронною поштою або якщо такого не існує, то система автоматично створить його з тимчасовим паролем, який прийде в якості повідомлення на пошту. Також, якщо користувач забув пароль від свого облікового запису, він може ввести пошту, на яку прийде повідомлення з формою відновлення пароля, яка дійсна півгодини, і змінити свій пароль після чого форма стане недійсною.



The screenshot displays the A.MOVIE website's sign-in interface. At the top, there is a dark navigation bar with the site logo 'A.MOVIE' on the left and links for 'Cinemas', 'Menu', 'Mega menu', and 'Contact Us' on the right. Further right are 'SIGN IN' and 'SIGN UP' links. Below the navigation bar is a search bar with the placeholder text 'Search' and a red 'SEARCH A MOVIE' button. The main content area features the heading 'SIGN IN' and the sub-heading 'welcome to A.Movie'. Below this are social media login options for Google+ and Facebook. A separator 'or' is placed below the social icons. The sign-in form consists of two input fields: 'Username' and 'Password', followed by a yellow 'SIGN IN' button. A link for 'Forgot password?' is located below the button.

Рисунок 19 - Форма авторизації

Далі ми можемо перейти до навігатору для перегляду списку всіх наявних на сайті фільмів та серіалів. На даній сторінці можна відсортувати елементи по рейтингу, даті, популярності та ін. Також наявна розширена форма для фільтрації елементів по категорії, кінокомпанії, жанру, країні та рокам.

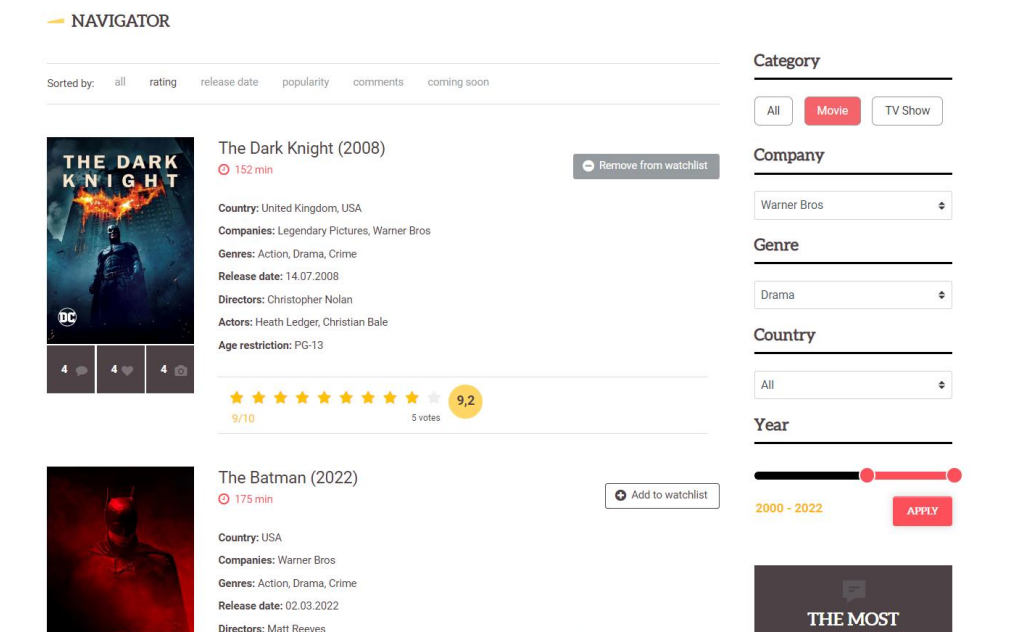


Рисунок 20 - Сторінка навігації по фільмам та серіалам

На сторінці кінокартини можна переглянути детальну інформацію про неї. Також, якщо користувач авторизований, є можливість оцінити фільм, додати його до списку улюблених чи списку бажаного до перегляду. За допомогою Аjax [22] всі ці запити працюють динамічно без перезавантаження сторінки. Також є можливість залишити коментар, відповісти на інший та лайкнути або дизлайкнути коментар.

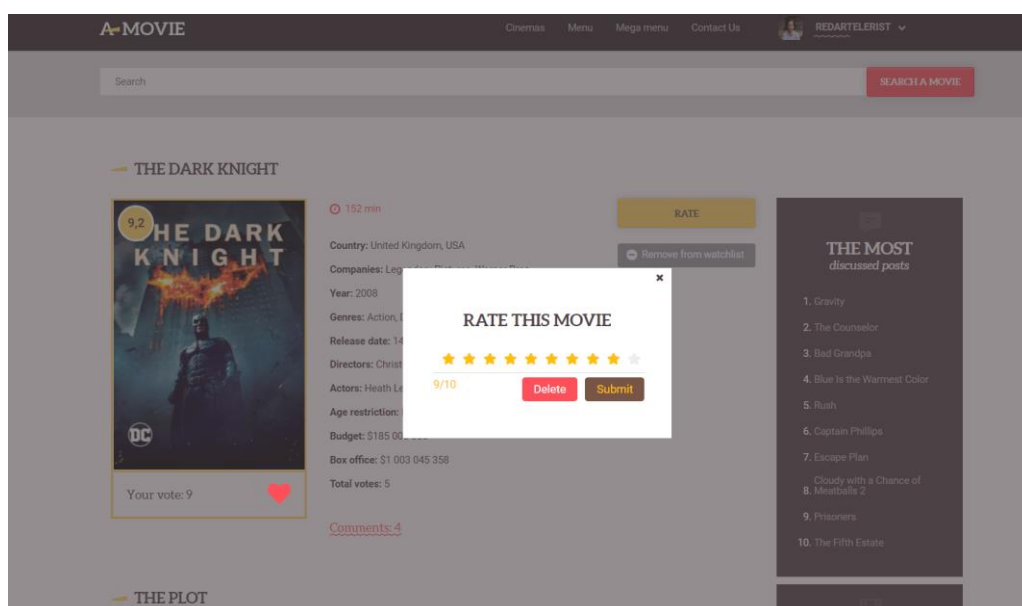



Рисунок 21 - Детальна сторінка фільму та форма виставлення оцінки


PHOTOS & VIDEOS 4 photos videos





COMMENTS (4)

Add your comment here

1000 characters left ADD COMMENT

 **Wiilam Defo**
28.12.2021 | 21:06 Reply

About as near perfect of a film as you can get for a comic book or action film. The script was brilliant, Heath Ledger's performance was well worth the Oscar he received. Even the intriguing, yet simplistic first scene sets the tone for the entire film. It says something about a film when every other film in its sub-genre tries to emulate it in some shape or form. This really kicked off the style that Marvel is going with now in that the films are first and foremost genre films but with superheroes added to them (this being a crime thriller w superheroes). Definitely one that should be watched by everyone. This film commonly is called the best comic book movie ever and it is hard to argue with people that make that claim.

 1  0



 **Maksym Fedorenko**
28.12.2021 | 21:03 Reply

Рисунок 22 - Список коментарів фільму

Можна переглянути профіль користувача, його оцінки, коментарі та ін. Також, можна підписатися на нього, якщо ви авторизовані.



RedHost SUBSCRIBE

Full Name / Maksym Fedorenko
E-mail / fedorenko.max02@gmail.com
Birthday / 10.12.1998, 23 y.o
Registered / 15.12.2021

| | | | | | |
|---------------------|------------------------|------------------------|-----------------------|---------------------------|-------------------------|
| 19 Scores | 13 Watchlist | 8 Favourites | 23 Comments | 4 Subscriptions | 1 Subscribers |
|---------------------|------------------------|------------------------|-----------------------|---------------------------|-------------------------|

Рисунок 23 - Сторінка користувача

[REDHOST](#) > SCORES

Sorted by: [date desc](#) [date asc](#) [rating](#) [score](#)

| Item | Data and time | Score | My score | Rate | Watch |
|---|-------------------|-------|----------|------|-------|
| 1. Uncharted (2022) 7,75 (4 votes) 120 min | 18.02.2022, 00:30 | 9 | 7 | ☆ | 🔖 |
| 2. Spider-Man: No Way Home (2021) 8,75 (8 votes) 148 min | 08.02.2022, 22:17 | 9 | 9 | ☆ | 🔖 |
| 3. Inception (2010) 9,2 (5 votes) 148 min | 05.02.2022, 23:06 | 10 | 9 | ☆ | 🔖 |
| 4. The Matrix (1999) 8,5 (4 votes) 136 min | 25.01.2022, 13:13 | 8 | 8 | ☆ | 🔖 |
| 5. Breaking Bad (2008) 9,25 (4 votes) 49 min | 09.01.2022, 21:20 | 9 | 10 | ☆ | 🔖 |
| 6. Fight Club (1999) 8,4 (5 votes) 138 min | 06.01.2022, 22:00 | 10 | 8 | ☆ | 🔖 |
| 7. The Lord of the Rings: The Two Towers (2002) 9 (3 votes) 170 min | 06.01.2022, 22:00 | 8 | 10 | ☆ | 🔖 |
| 8. Game of Thrones (2011) 9 (4 votes) 57 min | 04.01.2022, 00:11 | 10 | 9 | ☆ | 🔖 |
| 9. Chernobyl (2019) 8,75 (4 votes) 307 min | 04.01.2022, 00:11 | 9 | 8 | ☆ | 🔖 |
| 10. The Shawshank Redemption (1994) | 30.12.2021, 00:20 | 8 | 10 | ☆ | 🔖 |

Category

All

Genre

All

Year

1972 - 2022

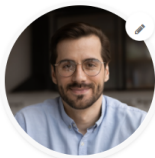
THE MOST discussed posts

- Gravity
- The Counselor
- Bad Grandpa
- Blue Is the Warmest Color
- Rush

Рисунок 24 - Список оцінок користувача

Зі своїм акаунтом можна робити теж саме, додатково є можливість перегляду куплених білетів за весь час. У своєму профілі можна редагувати інформацію про себе, додати зображення та змінити пароль при необхідності.

[USER PAGE](#) > [EDIT PROFILE](#)



Upload a different photo...Maximum size 2mb

Personal info

Username:

Email:

First Name:

Last Name:

Birth day:

Gender:

Change password

New password:

Confirm password:

Рисунок 25 - Сторінка власного профілю

Для того щоб замовити квитки, користувач може перейти до списку кінотеатрів, відфільтрувати їх по місту та вибрати потрібний. На головній сторінці відображається інформація про кінотеатр, його місцезнаходження та список сеансів. Можна вибрати дату із можливих, яка вас цікавить, та вибрати сеанс на бажаний фільм.

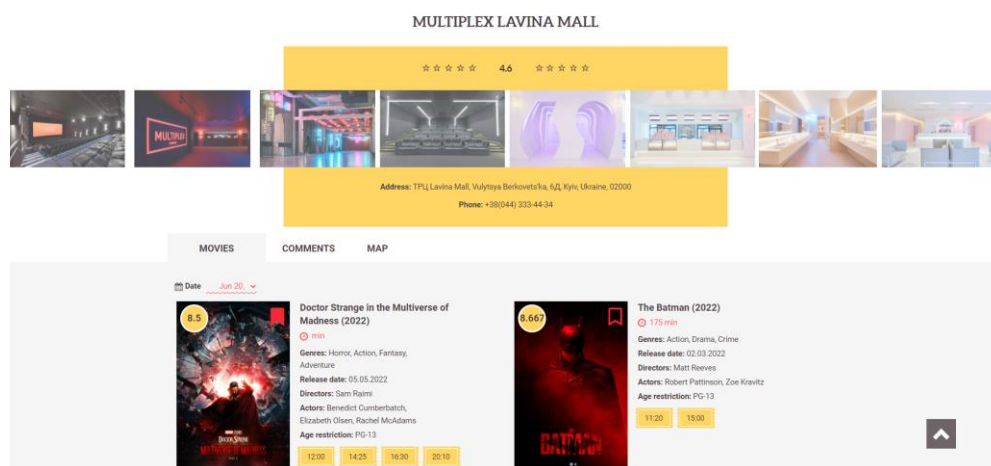


Рисунок 26 - Сторінка кінотеатру

Далі можна вибрати вільні місця і перейти до наступного кроку.

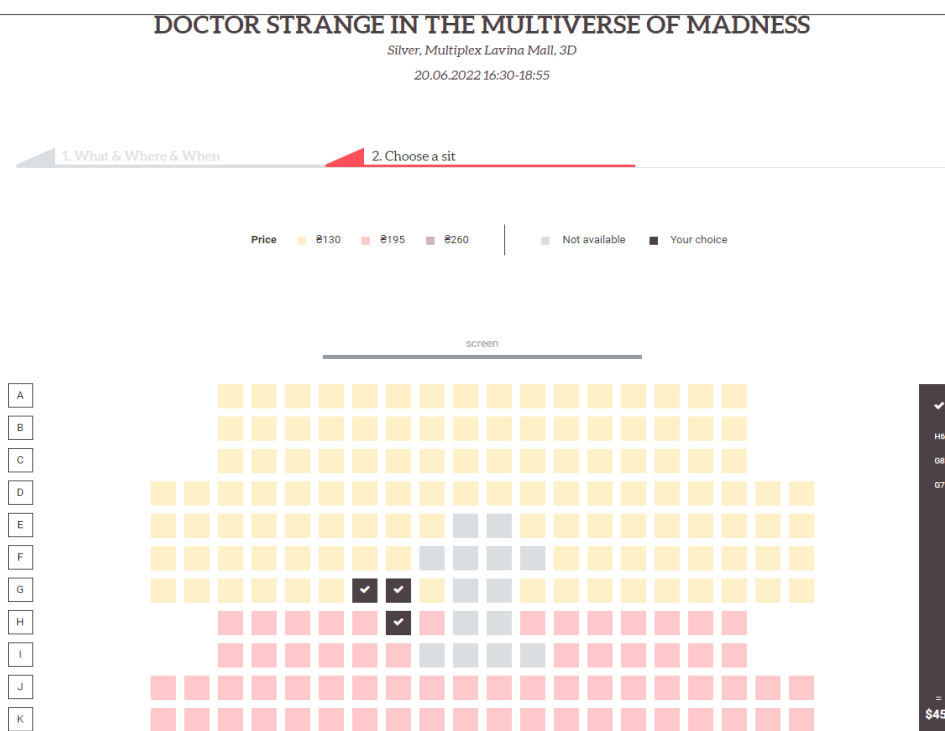


Рисунок 27 - Сторінка сеансу

При чому, коли ви переходите далі, то ваші місця стають недоступними для інших протягом семи хвилин. За цей час вам треба заповнити інформацію про замовлення та оплатити карткою за допомогою сервісу Stripe.

1. What & Where & When 2. Choose a sit 3. Check out

6:40

ORDER INFORMATION

Movie: **Doctor Strange in the Multiverse of Madness**
 Date: **20.06.2022, 16:30**
 Cinema: **Multiplex Lavina Mall** Hall: **Silver**

PRICE

Tickets: **3**

| | | | |
|-----------|----------|-----------------|-------------|
| Standard: | 2 | One item price: | €130 |
| Vip: | 1 | One item price: | €195 |

Total: **€455**

CONTACT INFORMATION

fedorenko.max02@knu.ua Phone number

Maksym Fedorenko

PURCHASE

Рисунок 28 - Сторінка купівлі квитків

1. What & Where & When 2. Choose a sit 3. Check out

5:37

ORDER INFORMATION

Movie: **Doctor Strange in the Multiverse of M**
 Date: **20.06.2022, 16:30**
 Cinema: **Multiplex Lavina Mall** Hall: **Silv**

PRICE

Tickets: **3**

| | | | |
|-----------|----------|-----------------|-------------|
| Standard: | 2 | One item price: | €130 |
| Vip: | 1 | One item price: | €195 |

Total: **€455**

A.Movie
 Booking tickets
 fedorenko.max02@knu.ua

4242 4242 4242 4242 VISA
 01 / 23 123

✓

Рисунок 29 - Оплата картою

Після успішної оплати вам на пошту, а також в телеграм бот прийде квиток.

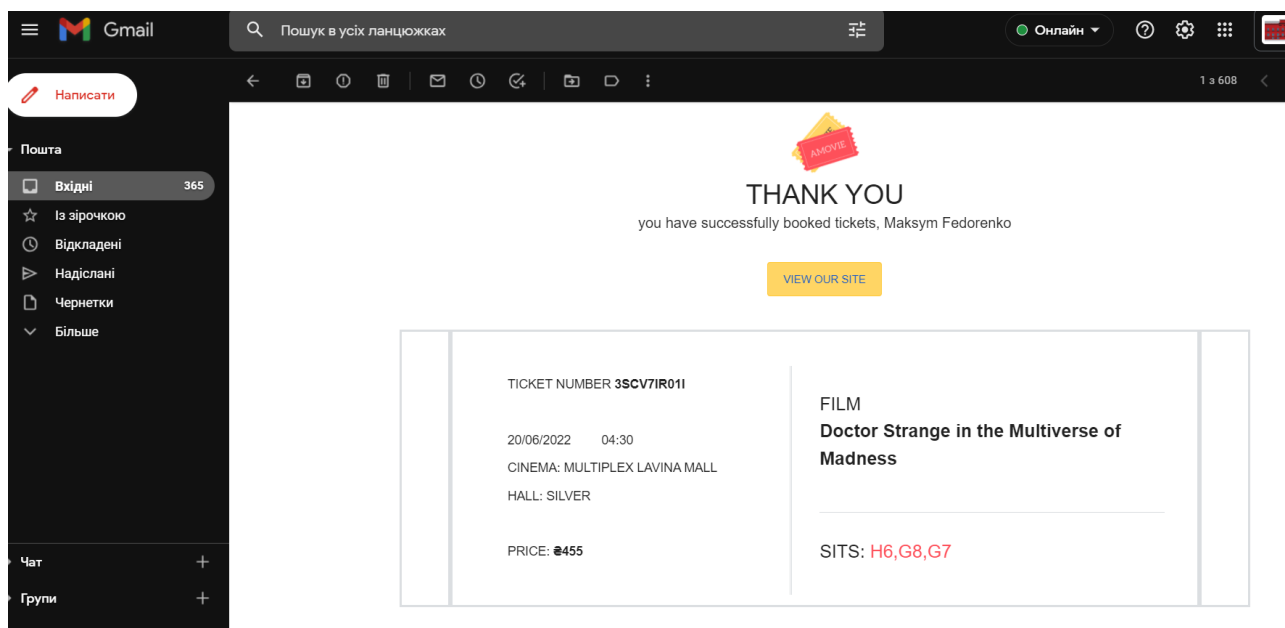


Рисунок 30 - Квиток на пошті

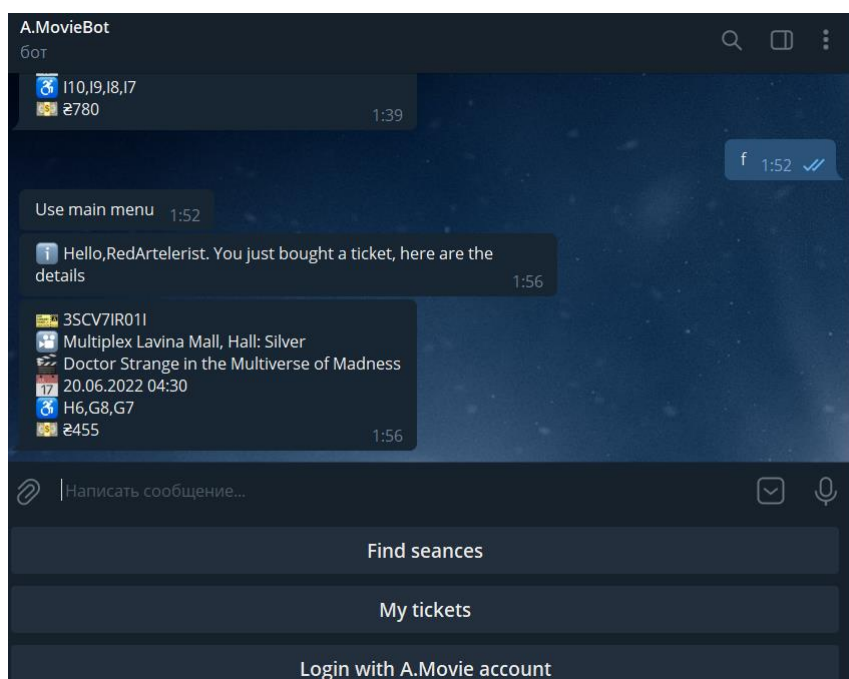


Рисунок 31 - Квиток в телеграм боті

Також користувач може скористатися телеграм ботом від веб-додатка. В ньому можна по юзернейму та паролю приєднатися до облікового запису «A.Movie».

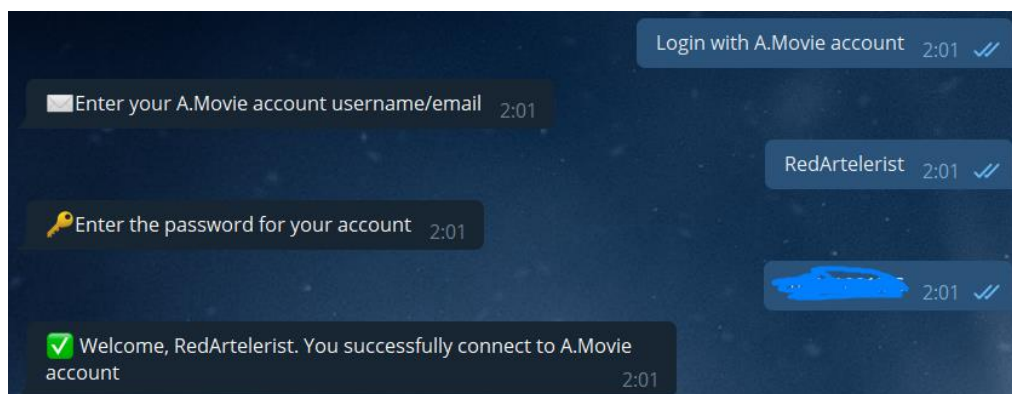


Рисунок 32 - Приєднання до «A.Movie» акаунту в телеграм боті

Функціонал бота дозволяє переглядати свої білети куплені за весь час, знаходити сеанси по кінотеатру, фільму та даті, а також бот має систему сповіщень, яка буде вам надсилати відповідне повідомлення за дві години до вашого сеансу.

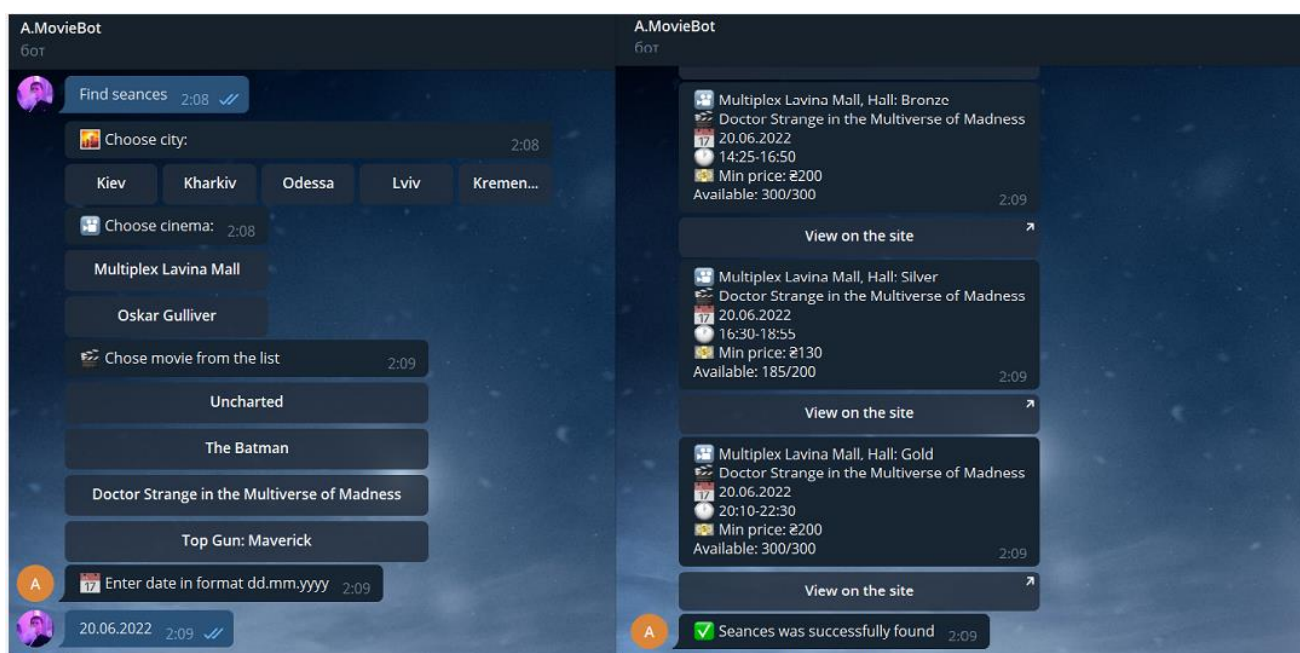


Рисунок 33 - Пошук сеансів в телеграм боті

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було виконано такі поставлені задачі:

- досліджено готові рішення соціальних кіномереж та сервісів бронювання, купівлі квитків, виявлено їх недоліки та переваги в результаті чого сформовано власні пріоритети для формування функціоналу сайту
- розглянуто та досліджено Spring-фреймворк, його особливості, бібліотеки та архітектуру
- поглиблено навички у розробці веб-застосунків
- впроваджено в проєкт різні API (Stripe, Firebase, Telegram Bot, Google, тощо). Деякі з них використовувалися вперше (Stripe, Telegram Bot), а з іншими були закріплені та поглиблені знання, отримані раніше
- розроблено технічне завдання, інтерфейс та дизайн веб-застосунку «A.Movie»
- розроблено власний веб-сайт, який є поєднанням ідеї соціальної мережі для кіноманів з послугами бронювання і купівлі квитків в онлайн кінотеатрах, а також додатково до сайту розроблено телеграм бота, який допомагає у роботі і доповнює веб-застосунок.

В майбутньому дана система може бути розширена шляхом подальшого збагачення функціональних можливостей: наприклад, стрічки кіноновин або навіть онлайн кінотеатру для перегляду кінофільмів та серіалів за окрему підписку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. JetBrains [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com>.
2. IMDb [Електронний ресурс] – Режим доступу до ресурсу: <https://www.imdb.com/>
3. Інформація про вебсайт IMDb [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/IMDb>
4. Multiplex [Електронний ресурс] – Режим доступу до ресурсу: <https://multiplex.ua/>
5. Java EE [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Java_EE
6. Understanding the Basics of Spring vs. Spring Boot [Електронний ресурс] – Режим доступу до ресурсу: <https://dzone.com/articles/understanding-the-basics-of-spring-vs-spring-boot>
7. Spring Boot – Architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/spring-boot-architecture/>
8. PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/postgresql-vs-mysql>
9. Firebase [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/docs/storage>
10. Google OAuth API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/identity/protocols/oauth2>
11. Facebook API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.facebook.com/docs/facebook-login/web/>
12. Maps JavaScript API [Електронний ресурс] – Режим доступу до ресурсу: <https://developers.google.com/maps/documentation/javascript/overview>

13. Stripe [Електронний ресурс] – Режим доступу до ресурсу:
[https://en.wikipedia.org/wiki/Stripe_\(company\)](https://en.wikipedia.org/wiki/Stripe_(company))
14. Stripe documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://stripe.com/docs/api>
15. Телеграм [Електронний ресурс] – Режим доступу до ресурсу:
<https://ru.wikipedia.org/wiki/Telegram>
16. Приклад реалізації телеграм боту на основі webhook [Електронний ресурс] – Режим доступу до ресурсу: <https://javarush.ru/groups/posts/3493-telegram-bot--napominalka-cherez-webhook-na-java-ili-skazhi-net-google-kalendarju>
17. FreeMarker [Електронний ресурс] – Режим доступу до ресурсу:
<https://freemarker.apache.org/>
18. Spring Data JPA [Електронний ресурс] – Режим доступу до ресурсу:
<https://thorben-janssen.com/what-is-spring-data-jpa-and-why-should-you-use-it/>
19. Hibernate [Електронний ресурс] – Режим доступу до ресурсу:
<https://dzone.com/articles/what-is-the-difference-between-hibernate-and-sprin-1>
20. JPQL [Електронний ресурс] – Режим доступу до ресурсу: <https://thorben-janssen.com/jpql/>
21. DTO [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.javaguides.net/2021/02/spring-boot-dto-example-entity-to-dto.html>
22. AJAX [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.wikipedia.org/wiki/AJAX>

ДОДАТКИ

Додаток А. Діаграма класів

