

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач випускової кафедри

Аліна САВЧЕНКО.

«_____» _____ 2024р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”

Тема: “Інформаційний вебпортал фільмів ”

Виконавець: студент групи УС-413Б Камінський Костянтин Володимирович

Керівник: старший викладач КІТ Шевченко Олександр Петрович

Нормоконтролер: _____ Олександр ШЕВЧЕНКО

Київ – 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

ЗАТВЕРДЖУЮ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
«_____» _____ 2024р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи (проєкту)

Камінського Костянтина Володимировича
(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Інформаційний вебпортал фільмів» затверджена наказом ректора від “05”квітня 2024 р. за № 517/ст.
- 2. Термін виконання роботи:** 15.04.2024 – 04.06.2024
- 3. Вихідні дані до роботи:** аналіз технологічних підходів створення веб-додатків, теоретичні відомості веб-розробки, інструментальні засоби розробки веб-сервісів.
- 4. Зміст пояснювальної записки:** вступ, аналіз інструментів створення веб-додатків, опис та вибір інструментів та технологій для реалізації веб-сервісу, проектування веб-додатку, розробка веб-додатку.
- 5. Перелік обов'язкового графічного матеріалу:** слайди презентації в MS PowerPoint.

6. Календарний план-графік

№ п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу, створення плану дипломної роботи та побудова плану-графіку виконання робіт.	15.04.2024 – 17.04.2024	
2.	Розроблення та затвердження календарного плану виконання дипломної роботи.	18.04.2024 – 24.04.2024	
3.	Проведення консультацій з науковим керівником.	25.04.2024 – 26.04.2024	
4.	Огляд та аналіз наукової літератури по темі дипломної роботи та написання Розділу 1.	27.04.2024 – 03.05.2024	
5.	Написання Розділу 2 дипломної роботи.	05.05.2024 – 15.05.2024	
6.	Написання Розділу 3 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	16.05.2024 – 25.05.2024	
7.	Оформлення та друк пояснювальної записки.	26.05.2024 – 29.05.2024	
8.	Створення презентації, доповіді та підготовка до захисту дипломної роботи.	30.05.2024 – 02.06.2024	
9.	Підготовка матеріалів дипломної роботи для передачі секретарю ДЕК (папка, конверт, диск із файлом диплому, рецензія, відгук).	03.06.2024 – 04.06.2024	

7. Дата видачі завдання: «15» квітня 2024 р.

Керівник дипломної роботи _____ Олександр ШЕВЧЕНКО
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Костянтин КАМІНСЬКИЙ
(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Інформаційний вебпортал фільмів»: містить 64 сторінки, 31 рисунок, 12 літературних джерел.

Об'єкт дослідження: веб-сервіс кінопорталів.

Мета роботи: Розробка веб-сервісу кінопорталу.

Методи дослідження: Методи порівняльного аналізу існуючих веб-сервісів та технологічних засобів, обробка літературних джерел

Отримані результати та їх новизна: Створений сучасний інформаційний вебпортал фільмів, який дозволить користувачам переважно у Україні взаємодіяти з приємним функціоналом та кіноконтентом веб-сервісу. Платформа містить соціальні функції для користувачів, і буде масштабованою та оптимізованою під сучасні потреби користувачів.

Результати роботи рекомендується використовувати під час реалізації схожих комерційних продуктів.

ВЕБСЕРВІС, КІНОПОРТАЛ, MERN, БАЗИ ДАНИХ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	7
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1. Визначення та способи використання	11
1.2. Аналіз ринку кінопорталів	12
1.3. Функціонал веб-сервісу	17
1.4. Технологічні компоненти	19
1.4.1. Аналіз бек-енд технологій.....	19
1.4.2. Аналіз фронт-енд технологій	21
1.4.3. Аналіз систем баз даних	23
1.4.4. Аналіз систем створення НТТР запитів.....	24
1.5. Висновки до Розділу 1	26
РОЗДІЛ 2. АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ ВЕБ-СЕРВІСУ	28
2.1. Стек технологій MERN.....	28
2.1.1. Структура стеку	29
2.1.2. MongoDB	31
2.1.3. Express	33
2.1.4. React.....	34
2.1.5. Node.JS.....	36
2.1.6. Забезпечення захисту стеку.....	39
2.2. Програмні залежності для веб-сервісу.....	40
2.3. OMDV API.....	43
2.4. Середовище розробки.....	44
2.5. Висновки до розділу 2	46
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РОЗРОБКА КІНОПОРТАЛУ.....	47
3.1. Система збереження ключей	47
3.2. Реалізація структури БД	48
3.3. Реалізація серверного рівня веб-сервісу	50
3.4. Реалізація веб-рівня	52
3.5. Висновки до розділу 3	61

ВИСНОВКИ	63
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- IMDB** Internet Movie Database – найвідоміша база даних фільмів, телешоу та акторів.
- API** Application Programming Interface – інтерфейс програмування додатків, який дозволяє різним програмним системам взаємодіяти між собою.
- OMDB API** Open Movie Database API – безкоштовний API, що надає доступ до даних IMDB.
- REST API** Representational State Transfer API – архітектурний стиль API, що використовує HTTP-методи для отримання, створення, оновлення та видалення даних.
- JS** JavaScript – мова програмування, що використовується для створення динамічного контенту на веб-сторінках.
- NPM** Node Package Manager – менеджер пакетів для JavaScript, що використовується для встановлення та керування бібліотеками.
- MVC** Model-View-Controller – архітектурний шаблон для веб-застосунків, що розділяє логіку програми на три компоненти: модель, вид та контролер.
- ORM** Object-Relational Mapping – техніка, що використовується для відображення об'єктів у мові програмування на структури даних в базі даних.
- DOM** Document Object Model – деревоподібна структура, що представляє HTML-документ.
- HTML** HyperText Markup Language – мова розмітки для створення веб-сторінок.
- CSS** Cascading Style Sheets – мова стилів для форматування веб-сторінок.
- XML** Extensible Markup Language – мова розмітки для структурування та обміну даними.
- JSX** JavaScript XML – розширення JavaScript, що дозволяє використовувати XML-синтаксис для створення React-компонентів.
- СКБД** Система керування базами даних – програмне забезпечення для створення, керування та доступу до баз даних.

SQL	Structured Query Language – мова для роботи з реляційними базами даних.
NoSQL	Не реляційні бази даних – тип баз даних, що не використовує реляційну модель даних.
JSON	JavaScript Object Notation – формат даних, що використовується для обміну даними між веб-застосунками.
BSON	Binary JSON – двійковий формат JSON, що використовується в MongoDB.
HTTP	Hypertext Transfer Protocol – протокол для передачі даних у веб-мережі.
PHP	Hypertext Preprocessor – мова програмування для веб-розробки.
Promise	Об'єкт JavaScript, що представляє асинхронну операцію.
URL	Uniform Resource Locator – унікальний ідентифікатор ресурсу в Інтернеті.
RDBMS	Relational Database Management System – реляційна система керування базами даних.
UI	User interface – частина програми, з якою взаємодіє користувач.
ES6	EcmaScript 2015 - нова версія JavaScript, що включає багато нових функцій.
JWT	JSON Web Token - формат для безпечного обміну даними між веб-застосунками.
CORS	Cross-Origin Resource Sharing - механізм, що дозволяє веб-сторінкам з одного домену отримувати доступ до ресурсів з іншого домену.
ENV	Environment variables - файл з навколишнім середовищем, що містить змінні середовища для веб-застосунка.
БД	База даних.

ВСТУП

У сучасному світі кінематограф відіграє значну роль у житті людей, надаючи їм розваги, інформацію та джерело натхнення. З розвитком технологій та зростанням доступності інтернету, з'явилась потреба у зручних та інформативних платформах, де люди могли б знаходити інформацію про фільми, читати рецензії та ділитися власними враженнями.

Наразі світовий ринок захопили такі англomовні платформи як IMDb, Rotten Tomatoes або Metacritic. В Україні активно розвиваються онлайн-спільноти любителів кіно, де люди обговорюють фільми, пишуть рецензії та діляться своїми враженнями. Кінопортал, який буде об'єднувати такі спільноти, може стати цінним майданчиком для спілкування та обміну думками.

Цей дипломний проект присвячується розробці інтуїтивно зрозумілого веб-сервісу для перегляду інформації про будь-який кінофільм з акцентом на функції соціальної взаємодії користувачів.

Цільовою групою користувачів є завзяті кіномани або ті, хто хоче відстежувати інформацію про те, які кінострічки вони бачили і які не бачили. Вони зможуть відмітити, які фільми вони переглянули, як вони їх оцінили, що думають про них, а також побачити, що дивилися інші користувачі.

Веб-сервіс буде розроблений на мові JavaScript на платформі Node.js з використанням фреймворка React.js, який наразі є досить популярним у світі, і використовується в розробці такими гігантами як Netflix, Instagram, Facebook та багато іншими.

У роботі будуть розглянуті технічні аспекти роботи з JavaScript бібліотеками, методи впровадження API інтерфейсу у веб-сервіс, який буде використовуватися як основне джерело даних. Також до веб-сервісу буде інтегрована база даних для збереження інформації про користувачів та їх дані.

Метою кваліфікаційної роботи є проектування та реалізація функціонального та зручного для користувача кінопорталу з використанням сучасних технологій JavaScript,

що надає користувачам вичерпну інформацію про фільми та функціонал для перегляду та надання інформації про них.

Важливість цієї роботи обумовлена зростаючим попитом на централізовані платформи, що пропонують вичерпну інформацію про фільми та функціонал для їх перегляду, особливо в таких регіонах, як Україна, де не вистачає спеціалізованих баз даних фільмів.

Результати цього проекту будуть характеризуватися успішною реалізацією функціонального кінопорталу, що відповідає заданим вимогам. Це включає в себе зручний інтерфейс для реєстрації користувачів, пошуку фільмів та написання рецензій, а також безперешкодну інтеграцію із зовнішніми API для отримання даних про фільми та надійну систему баз даних для зберігання інформації про користувачів та фільми.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Визначення та способи використання

Кінопортал - це централізована онлайн-платформа, яка надає вичерпну інформацію, ресурси та функції, пов'язані з кіно. Він слугує центром, де користувачі мають доступ до широкого спектру контенту, пов'язаного з кіно, включно з деталями фільмів, інформацією про акторів та знімальну групу, рецензіями та рейтингами, трейлерами та промо-матеріалами, новинами та оновленнями, а також супутніми товарами. Кінопортали часто включають такі функції, як пошук і перегляд кіноматеріалів, користувацький контент, елементи соціальних мереж і персоналізовані рекомендації, щоб покращити користувацький досвід і полегшити взаємодію з кіноспільнотою. Ці платформи також можуть пропонувати додаткові послуги, такі як бронювання квитків, потокове мовлення та форуми спільнот, щоб ще більше залучити та задовольнити різноманітні потреби та інтереси кінолюбителів.

Крім того, кінопортали відіграють важливу роль у кіноіндустрії, надаючи цінні дані та інформацію режисерам, студіям, дистриб'юторам та експонентам. Дані про касові збори, демографічні показники аудиторії та відгуки користувачів допомагають формувати маркетингові стратегії, рішення щодо дистрибуції та створення контенту, визначаючи траєкторію розвитку індустрії в цілому.

Загалом, кінопортали стали незамінними інструментами як для пересічних кіноглядачів, так і для професіоналів індустрії, слугуючи віртуальними воротами до величезного та різноманітного світу кіно.

Кафедра КІТ (47)				НАУ 24 33 92 000 ПЗ			
<i>Виконав</i>	<i>Камінський К.В.</i>			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	<i>Літера</i>	<i>Аркуш</i>	<i>Аркуші</i>
<i>Керівник</i>	<i>Шевченко О.П.</i>				У	11	17
<i>Консулт.</i>					УС-413Б 122		
<i>Норм. контр.</i>	<i>Шевченко О.П.</i>						

1.2. Аналіз ринку кінопорталів

Internet Movie Database

IMDb - це найбільша у світі база даних фільмів, яку можна вільно редагувати. Була заснована в жовтні 1990 року молодим програмістом та кіноманом Коліном Нідлі. Цей сайт належить Amazon з 1998 року, тому на цьому сайті є багато посилань на інші та продукти цієї компанії. Функції бази даних поділяються на дві основні версії:

- Звичайні користувачі можуть використовувати цей сайт для перегляду інформації та розважальних цілей: шукати інформацію про фільми, серіали, відео-контент та шоу від авторів з усього світу, дізнаватися новини про кіноіндустрію, оцінювати переглянуті фільми, писати коментарі та огляди (але лише англійською мовою та в помірних кількостях), відзначати цікаві фільми які хочеться подивитися пізніше;
- Для досвідчених любителів кіноіндустрії є платна Pro-версія з дещо іншими функціями: портфоліо, професійні контакти, перегляд статистики, можливість створювати резюме фільмів у більш розширеній формі (наприклад, додавати фотографії та відео).

Принцип наповнення сайту в основному нічим не відрізняється від Вікіпедії: будь-який зареєстрований користувач може додати інформацію, після чого вона схвалюється або відхиляється модераторами. Редакційний матеріал на сайті створюються офіційною командою IMDb.

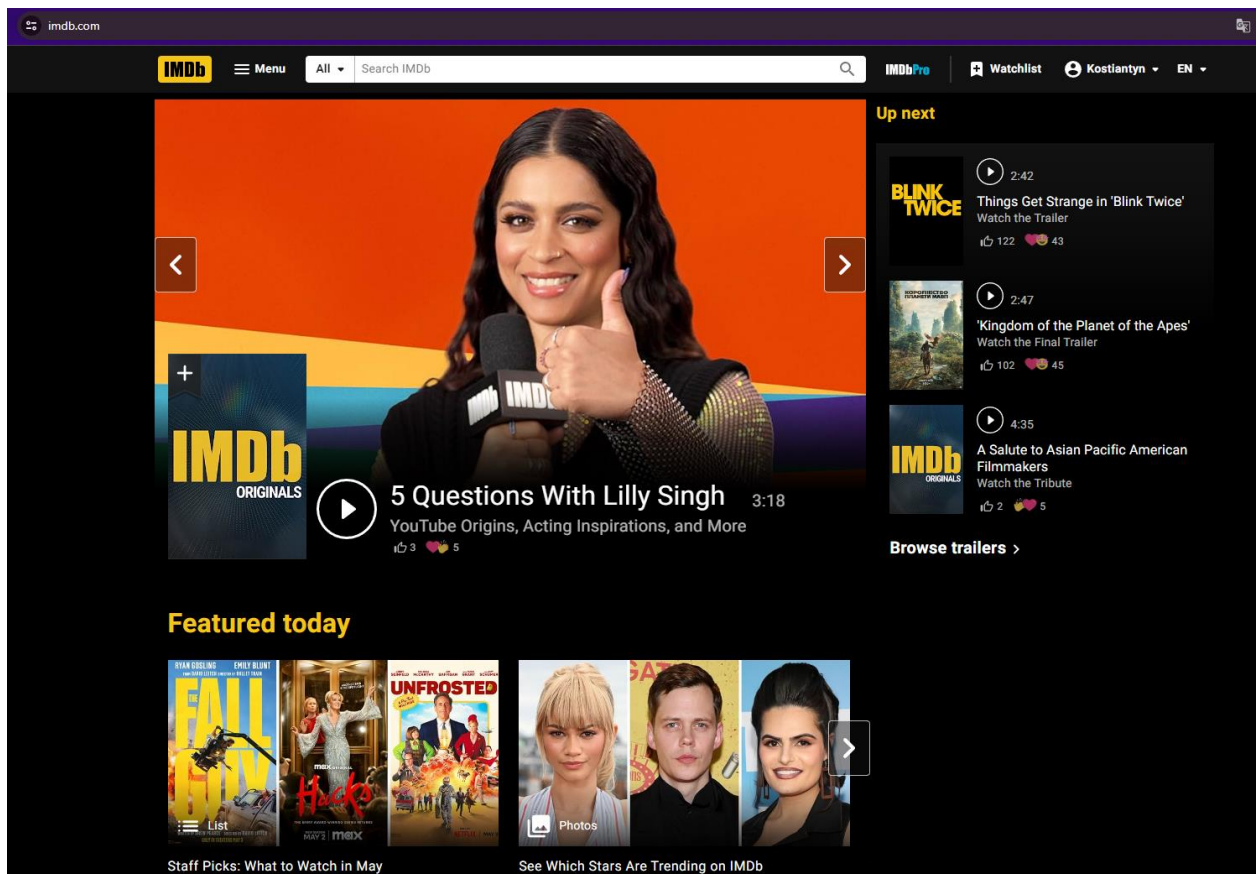


Рис. 1.1. Вигляд веб-сервісу IMDb

Донедавна в Україні цей сайт був популярний лише серед користувачів, які добре володіли англійською мовою, оскільки ресурс був англomовним. Його головним конкурентом на пострадянському просторі являється «КіноПоиск», російський клон IMDb, звідки він також запозичив багато інформації про фільми. Однак епоха популярності «КіноПоиска» в Україні закінчилася в 2017 році, коли його внесли в санкційний список і фактично заблокували його у вільному доступі, а сам ресурс фактично перетворився на російський онлайн-кінотеатр.

Крім того, наразі IMDb доступний у 9 мовних варіантах із додаванням перекладу в браузері. На IMDb також є українська мова, але лише для назв фільмів (звісно, якщо їх додають контрибутори). Також великою перемогою є те, що після 10 років безперервної дискусії на форумі сайту про необхідність додавання літер «і, є, г» до стандартної кирилиці це нарешті було зроблено.

Як і будь-який сайт, який покладається на довіру користувачів, IMDb має проблеми з помилками або зловживаннями. Крім того, додавати інформацію на сайт

відносно легко, але редагувати або видаляти її дуже важко навіть для досвідчених учасників. Ми всі знаємо про «війни редакторів» у Вікіпедії – те ж саме відбувається на IMDb. Наприклад, список нагород неіснуючої «Асоціації кінокритиків» залишався на сайті 2 роки. Крім того, один із користувачів сайту змінив опис російського пропагандистського фільму «Крим» на «фільм про двох людей, які кричать один на одного». Інший приклад: директор української кіностудії наклеїла «продюсерський бейдж» на всі фільми цієї продакшн-компанії, хоча в титрах була позначена як адміністративний працівник. Загалом, «матеріалу» там достатньо, тому інформація на цій сторінці потребує перевірки – втім, як і будь-який інший інтернет-ресурс.

Letterboxd

Альтернативою IMDb є молодший, але не менш популярний сайт для кіноманів Letterboxd, створений у жовтні 2011 року новозеландськими дизайнерами Метью Бьюкененом і Карлом фон Рендоу. Відрізняється цей кінопортал від IMDb відкриттям профілів і можливістю ділитися відгуками, щоденниками переглядів або віртуальними списками. На IMDb усі ці функції також доступні в тій чи іншій формі, але вони там не є поширеними, оскільки їх важко реалізувати.

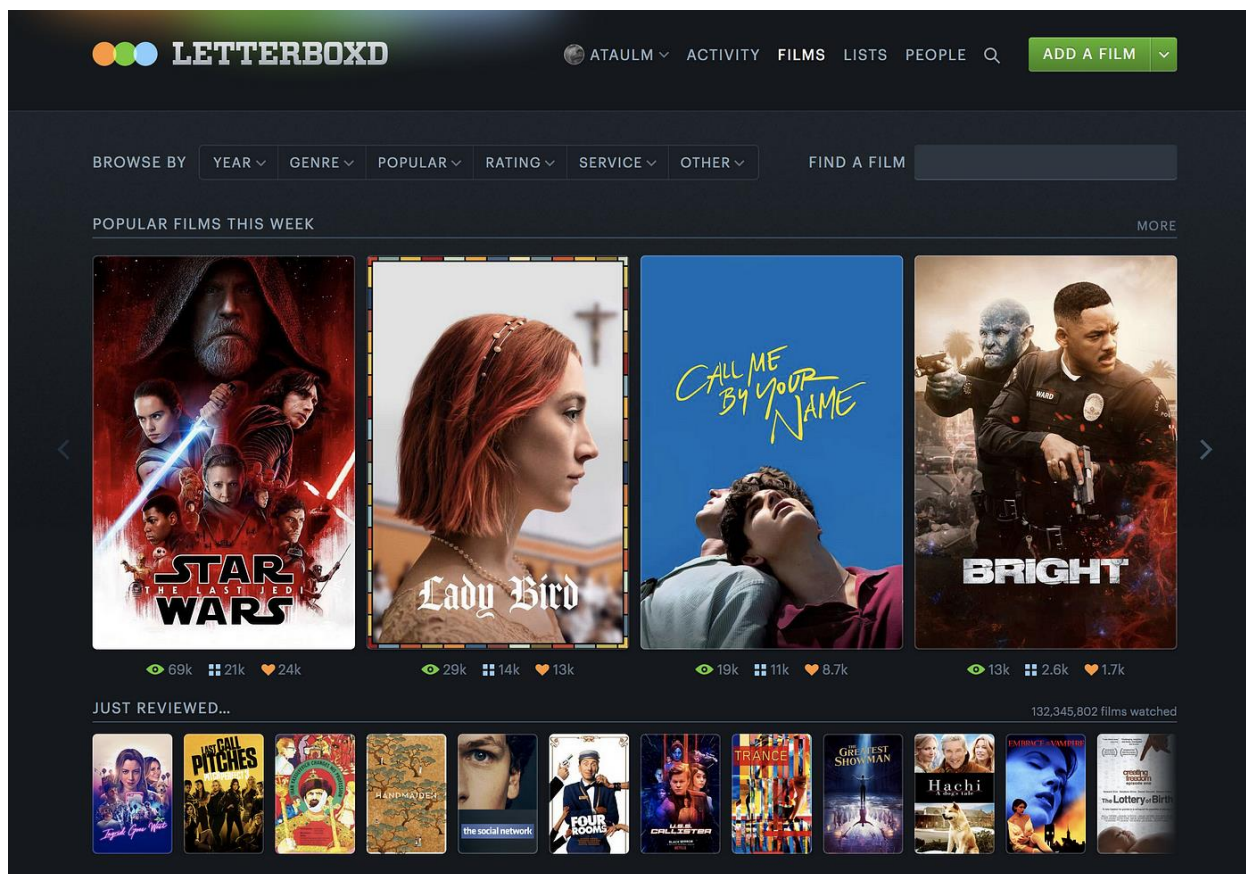


Рис. 1.2. Вигляд веб-сервісу Letterbod

Популярності Letterboxd сприяла реклама та грамотне тестування. Перше закриті бета-тестування було протестовано на 17 000 кіногіків. Через півроку була запущена публічна бета-версія, яка працювала за запрошеннями – і саме цей принцип допоміг створити платформу для нішевої аудиторії. І тільки в лютому 2013 року громадськість отримала доступ до цього сайту.

Особливість Letterboxd в тому, що в його базі немає сезонних серіалів (типу «Гри престолів» чи «Друзів») – такою є основна позиція авторів сайту. Але є міні-серіали та окремі серії, які є самостійними аудіовізуальними творами (наприклад, кожна серія серіалу Netflix «Любов, смерть і роботи» оформлена на окремій сторінці).

Як і IMDb, Letterboxd є безкоштовним сервісом, тому він «живе» за рахунок реклами, а також за рахунок преміальних і професійних підписок: облікові записи Pro, Patron і професійні облікові записи HQ пропонуються верифікованим користувачам. Преміум-підписка надає доступ до додаткових функцій, видаляє рекламу, а також є важливим фактором довговічності сайту. Вартість «оновлення» порівняно низька: Pro

коштує 16 євро, а «патрон» — 41 євро на рік зі знижкою 50% у «чорну п'ятницю». Для порівняння, обліковий запис IMDb Pro коштує 150 доларів на рік і доступний лише через обліковий запис Amazon.

Letterboxd дозволяє імпортувати рейтинги користувачів з IMDb, а також посилення на IMDb для вузькоспеціалізованої інформації (якою камерою знято фільм, який саундтрек використовувався). Проте на Letterboxd ви можете миттєво писати та публікувати відгуки (без попередньої модерації) будь-якою мовою світу. При необхідності його легко відредагувати або видалити. Тут також набагато простіше додати новий фільм до бази даних.

Популярність кінопорталів серед українських кінолюбителів

Точної статистики на рахунок популярності зарубіжних майданчиків серед українських кінолюбителів та професіоналів немає, проте можна точно сказати, що певний вплив у сфері кіноіндустрії вони привносять. Наприклад, база даних IMDb за ідентифікатором «Україна» наразі містить 4235 найменувань, а аналогічний індекс на Letterboxd показує цифру 2506. Ще рік тому ці показники були значно меншими.

На IMDb частина українських кінематографістів вже має Pro-акаунт(така необхідність зумовлена ринковою кон'юнктурою та співпрацею з іноземними продюсерами, які перш за все звертають увагу на профіль IMDb). Зацікавленість користувачів поки що залишає бажати кращого, адже навіть успішні фільми у вітчизняному прокаті мають до 50 оцінок рецензентів, а значить, дуже мало українців активно користуються даними веб-сервісу. Українські кінофестивалі та дистриб'ютори також не бажають використовувати потенціал сайту: сторінки, які наповнені інформацією про дати виходу та нагороди українською мовою, це все ще є здебільшого роботою волонтерів-ентузіастів.

Схожа ситуація на Letterboxd, який не є дуже популярним серед українських кінопрофесіоналів. Нещодавно українська незалежна VOD-платформа Takflix відкрила власний головний HQ-обліковий запис на сайті, ставши першим українським користувачем із подібним статусом. Проте український сегмент сайту вже має свою невелику фан-базу: на його просторі періодично можна дивитися не лише українське кіно, а й українську мову. Російська військова агресія також сприяла

популяризації України: 2 березня Letterboxd опублікував редакційну статтю «Культура як зброя», присвятивши її українському кіно та його просуванню на цій платформі.

1.3. Функціонал веб-сервісу

Аналіз функцій кінопорталу передбачає оцінку того, як кожна з них впливає на загальний користувацький досвід і корисність платформи. Кожна функція відіграє важливу роль у покращенні користувацького досвіду та формуванні почуття спільноти серед кінолюбителів:

- Реєстрація користувача. Автентифікація користувачів є наріжним каменем безпеки та персоналізації порталу. Вимагаючи від користувачів реєстрації та входу в систему, портал гарантує, що тільки авторизовані особи можуть отримати доступ до його функцій. Це не тільки захищає дані користувачів, але й дозволяє надавати персоналізовані рекомендації та соціальну взаємодію на основі індивідуальних уподобань та історії переглядів.
- Пошук. Функція пошуку - це двері до величезної бази даних фільмів та інформації порталу. Дозволяючи користувачам шукати фільми за різними критеріями, такими як сюжет, рік і назва, портал полегшує навігацію і знаходження контенту. Ця функція підвищує доступність для користувачів, економить час і зусилля на пошук потрібних фільмів, а також сприяє задоволенню та утриманню користувачів.
- Позначення переглянутого контенту. Можливість позначати кіноматеріал як "переглянутий" або "хочу переглянути" надає користувачам цінний функціонал для організації та управління їхнім переглядом. Ведучи персоналізований список перегляду, користувачі можуть відстежувати свій вибір розваг і легко знаходити новий контент відповідно до своїх інтересів. Ця функція підвищує залученість користувачів і заохочує їх до

повторних відвідувань порталу.

- Список переглянутого контенту. Надання користувачам можливості переглядати переглянутий контент та рейтинги забезпечує прозорість та підзвітність їхніх звичок перегляду. Користувачі можуть розмірковувати про свій минулий досвід перегляду, переглядати улюблені фільми і ділитися своїми думками зі спільнотою. Ця функція збагачує користувацький досвід, сприяючи самоаналізу і полегшуючи дискусії навколо спільних інтересів.
- Взаємодія з кіноконтентом. Надання користувачам можливості оцінювати фільми, залишати відгуки та записувати дати перегляду дає їм змогу ділитися цінною інформацією та думками на порталі. Користувацький контент розширює багатство і глибину бази даних фільмів на платформі, надаючи різноманітні точки зору і рекомендації іншим користувачам. Крім того, запис дат перегляду дозволяє користувачам відстежувати свої звички перегляду фільмів у часі та відзначати важливі віхи своєї кінематографічної подорожі.
- Відстеження акаунтів друзів. Соціальний зв'язок є ключовим аспектом зусиль порталу з розбудови спільноти. Надання користувачам можливості стежити за акаунтами своїх друзів полегшує соціальну взаємодію, розвиває почуття товариства серед користувачів, а також сприяє обміну рекомендаціями та дискусіям. Маючи доступ до списків перегляду та рецензій своїх друзів, користувачі можуть відкривати для себе нові фільми, брати участь у змістовних розмовах і зміцнювати свої зв'язки всередині спільноти.
- Вподобання рецензій. Включення функції вподобання рецензій друзів додає порталу ще один рівень соціального визнання. Користувачі можуть висловити вдячність за змістовні відгуки, продемонструвати підтримку думок своїх друзів і сприяти створенню позитивного зворотного зв'язку в спільноті. Ця функція заохочує залучення користувачів, стимулює створення високоякісного контенту та посилює відчуття приналежності

до спільноти кіноманів, що підтримує їх.

Надаючи користувачам можливість персоналізувати свої глядацькі вподобання, ділитися думками та спілкуватися з однодумцями, портал стає не просто платформою для доступу до кіноінформації, а й яскравим центром для виховання пристрасті до кіномистецтва. І саме таким чином і буде обраний функціонал для створення кінопорталу.

1.4. Технологічні компоненти

Кінопортал складається з кількох технологічних компонентів, кожен з яких слугує певній меті у забезпеченні захопливого та зручного для користувача досвіду для аудиторії. Ці компоненти включають фронтенд-фреймворки для побудови користувацького інтерфейсу, бекенд-технології для обробки даних і бізнес-логіки, системи баз даних для зберігання та пошуку інформації, а також API для інтеграції зовнішніх сервісів і джерел даних.

1.4.1. Аналіз бек-енд технологій

Бек-енд фреймворк - це заздалегідь створений набір інструментів, бібліотек, структур коду та інших компонентів, які полегшують розробку серверних систем програми.

Він забезпечує структуроване середовище для бекенд-розробників для побудови інфраструктури та функціональності програмного забезпечення, заощаджуючи їхній час та зусилля, вирішуючи спільні завдання та проблеми.

Видаляючи низькорівневі рутини, бекенд-фреймворки дозволяють розробникам зосередитися на створенні конкретних функцій і можливостей.

Node.js

Node.js - це серверне середовище виконання JavaScript з відкритим вихідним кодом, яке дозволяє розробникам виконувати JavaScript-код на сервері. На ньому побудовані такі відомі бекенд-фреймворки, як Express, Meteor, Koa, Sails, Nest та Napi.

Як провідна технологія для бекенда, Node.js здобула величезну популярність завдяки своїй швидкості, масштабованості та неблокуючій архітектурі вводу/виводу.

Node.js та його фреймворки добре підходять для додатків, що працюють у реальному часі та потребують миттєвого оновлення даних, таких як чат-додатки, онлайн-ігрові платформи, платформи електронної комерції, сервіси потокового мультимедіа та інструменти для спільної роботи. Крім того, ці технології часто використовують для створення API та мікросервісів завдяки їхній легкій природі та швидкості виконання.

Переваги:

- Висока продуктивність: Node.js відомий своєю швидкістю та ефективністю. Його неблокуюча модель вводу/виводу дозволяє обробляти велику кількість одночасних з'єднань з низькою затримкою;
- Велика екосистема: Node.js має багату екосистему бібліотек і пакетів, доступних через npm, що значно прискорює розробку;
- Активна спільнота: середовище виконання може похвалитися процвітаючою спільнотою розробників, що означає постійне вдосконалення, підтримку та велику кількість ресурсів;
- Масштабованість: Node.js розроблений для масштабованих проєктів і може ефективно справлятися з високими навантаженнями, що робить його придатним для додатків з різним рівнем використання.

Недоліки:

- Однопоточковість: архітектура Node.js, керована подіями, може бути не найкращим рішенням для завдань, що вимагають багато процесорних ресурсів і не можуть бути легко розпаралелені;
- Проблеми зворотних викликів: у складних додатках асинхронним кодом, заснованим на зворотних викликах, може стати важко керувати, що призводить до проблем зворотних викликів.

Django

Найпопулярніший фреймворк для Python з відкритим вихідним кодом Django сприяє швидкій розробці та чистому, прагматичному дизайну. Фреймворк

дотримується архітектурного шаблону MVC, розділяючи логіку, представлення та дані додатку на окремі компоненти.

Вбудований інтерфейс адміністратора Django робить його ідеальним для розробки систем управління контентом. Крім того, його надійне ORM добре підходить для додатків, які значною мірою покладаються на взаємодію з базами даних.

Переваги:

- Швидка розробка: філософія Django "з батарейками в комплекті" прискорює розробку завдяки вбудованим компонентам і функціям;
- Інтерфейс адміністратора: фреймворк пропонує панель адміністратора з коробки, що спрощує управління контентом та адміністративні завдання.

Недоліки:

- Монолітний підхід: уперта структура Django може бути не такою гнучкою для проєктів з дуже індивідуальними вимогами.

1.4.2. Аналіз фронт-енд технологій

Фронт-ендом – називаються графічний інтерфейс користувача або додатку у веб-розробці. Іншими словами, це та частина програми, яку в бачать користувачі і з якою вони можуть взаємодіяти. Ця частина повинна бути зручною для користувача, щоб ефективно передавати знання з боку сервера.

Фронтенд-фреймворки - це програмний продукт/інструмент/платформа, який слугує основою для розвитку інтерфейсних компонентів веб-рішень.

React.js

React, також відомий як ReactJS, - це добре відома інтерфейсна JavaScript-бібліотека для розробки користувацьких інтерфейсів та пов'язаних з ними компонентів. Вона використовує архітектуру MVC, з іншою таблицею представлення та доступності даних. Однією з її відмінних рис є віртуальний DOM і те, як він управляє доступом до документів і маніпуляціями з ними. DOM добре взаємодіє з HTML та XML документами, змушуючи їх поводитися подібно до деревовидної

структури, а кожен HTML елемент функціонує як об'єкт.

React, який працює на базі Facebook, був визнаний фантастичною бібліотекою в сімействі фронтенд-інструментів. Для створення компонентів використовується стиль кодування JSX з поєднанням HTML-цитат і синтаксису тегів. Вона розбиває великі компоненти на менші, якими можна керувати окремо та індивідуально. Завдяки цій функції загальна продуктивність розробників неодмінно зросте.

Плюси:

- Повторне використання компонентів полегшує співпрацю та повторне використання в інших частинах програми;
- Завдяки використанню віртуального DOM ви можете досягти стабільної та безперебійної роботи;
- Це найкраще рішення для написання компонентів у хуках React, оскільки дозволяє писати компоненти без класів і полегшує вивчення React.

Мінуси:

- Через численні та постійні оновлення цього фронтенд-фреймворку важко створити належну документацію, що впливає на криву навчання для початківців;
- Починаючи роботу з фреймворком, розробникам важко зрозуміти складнощі JSX;
- Він надає рішення лише для фронтенду.

Angular

Angular – це один з найкращий фреймворків, який домінує у списку фреймворків для веб-інтерфейсів. Це фреймворк для створення ефективних і просунутих односторінкових додатків. Це платформа розробки на основі Typescript, розроблена компанією Google. Angular - це компонентний фреймворк для розробки масштабованих веб-додатків, а також набір інструментів для розробників для створення, збірки, тестування і модифікації коду, і колекція добре інтегрованих бібліотек.

Angular дозволяє масштабувати односторінкові додатки до додатків корпоративного рівня залежно від ваших потреб. Наразі його використовують

YouTube та Google Translate. Він також має багато можливостей, що робить його одним з найпопулярніших фреймворків.

Плюси:

- Вбудований функціонал для оновлення налаштувань, зроблених у моделі, до представлення і навпаки;
- Зменшує кількість коду, оскільки більшість важливих функцій, таких як двостороннє зв'язування даних, надаються за замовчуванням;
- Визначаючи їх як зовнішні елементи, компоненти відокремлюються від своїх залежностей;
- Компоненти можна використовувати багаторазово і ними легко керувати за допомогою ін'єкції залежностей;
- Велика спільнота навчання та підтримки.

Мінуси:

- Оскільки Angular є повноцінним динамічним рішенням, існує безліч способів виконання роботи, а отже, шляхи навчання є непередбачуваними;
- Динамічні додатки іноді не працюють належним чином через свою складну структуру та розмір.

1.4.3. Аналіз систем баз даних

Системи баз даних відіграють вирішальну роль у зберіганні та пошуку даних на кінопорталах, включаючи інформацію про фільми, користувачів, рецензії та рейтинги.

MySQL

MySQL - це популярна, безкоштовна та відкрита реляційна СКБД, розроблена компанією Oracle. Як і інші реляційні системи, MySQL зберігає дані за допомогою таблиць і рядків, забезпечує цілісність посилань і використовує мову структурованих запитів (SQL) для доступу до даних. Коли користувачам потрібно отримати дані з бази даних MySQL, вони повинні створити SQL-запит, який об'єднує кілька таблиць

разом, щоб створити потрібне їм представлення даних.

Схеми бази даних і моделі даних повинні бути визначені заздалегідь, а дані повинні відповідати цій схемі, щоб зберігатися в базі даних. Такий жорсткий підхід до зберігання даних забезпечує певний рівень безпеки, але в обмін на гнучкість. Якщо в базі даних потрібно зберігати новий тип або формат даних, необхідно здійснити переміщення схеми, що може стати складним і дорогим процесом зі збільшенням розміру бази даних.

MongoDB

MongoDB також вільна у використанні та має відкритий вихідний код, проте її принципи побудови відрізняються від традиційних реляційних систем. MongoDB, яку часто називають нереляційною (або NoSQL) системою, використовує значно інший підхід до зберігання даних, представляючи інформацію у вигляді серії JSON-подібних документів (фактично зберігаються у вигляді бінарного JSON, або BSON), на відміну від формату таблиць і рядків реляційних систем.

Документи MongoDB складаються з серії пар ключ/значення різних типів, включаючи масиви і вкладені документи; однак, основна відмінність полягає в тому, що структура пар ключ/значення в даній колекції може змінюватися від документа до документа. Такий гнучкіший підхід можливий завдяки тому, що документи самоописуються.

1.4.4. Аналіз систем створення HTTP запитів

У сучасній веб-розробці створення HTTP-запитів є фундаментальним аспектом побудови надійних додатків. Дві найпоширеніші бібліотеки JavaScript для обробки HTTP-запитів - це Fetch і Axios. Ці бібліотеки спрощують процес надсилання запитів до бази даних та обробки відповідей.

Axios

Axios - популярна бібліотека сторонніх розробників, яка спрощує процес створення HTTP-запитів на JavaScript. Вона надає простий у використанні API з вбудованими функціями для роботи з перехоплювачами запитів і відповідей,

перетворення даних та обробки помилок.

Переваги:

- Широка підтримка браузерів: Axios підтримує широкий спектр браузерів, включаючи старіші версії та Internet Explorer. Це досягається завдяки використанню об'єкта `XMLHttpRequest` та наданню узгодженого API для різних браузерних середовищ;
- Автоматичне скасування запиту: Axios має вбудовану підтримку скасування запитів. Вона дозволяє створювати токени скасування і пов'язувати їх з конкретними запитами, що дозволяє ефективно скасовувати і очищати відкладені запити;
- Обробка помилок та перехоплювачі: Axios забезпечує надійний механізм обробки помилок. Він автоматично відхиляє Promise для будь-яких кодів статусу помилки (наприклад, 404 або 500) і дозволяє використовувати перехоплювачі для глобальної обробки запитів і відповідей.

Недоліки:

- Додаткова залежність: Axios потребує зовнішньої залежності і має бути встановлений окремо. Хоча це незначний недолік, він додає додатковий крок до процесу налаштування проекту;
- Більш складна конфігурація: Axios надає багатий набір опцій конфігурації, який може бути непосильним для простих випадків використання. Додаткова складність може збільшити час навчання для початківців.

Fetch

Fetch - це власний JavaScript API для створення асинхронних HTTP-запитів. Він надає простий та інтуїтивно зрозумілий інтерфейс для обробки мережеских запитів та роботи з відповідями.

Переваги:

- Підтримка браузером: Fetch - це вбудований браузерний API, доступний у більшості сучасних браузерів. Це усуває необхідність у зовнішніх залежностях, що робить його легким і простим у використанні без будь-

яких додаткових установок;

- API на основі Promises: Fetch використовує Promises, що дозволяє елегантно обробляти асинхронні операції. Promises забезпечують більш чистий синтаксис для ланцюжкових операцій, обробки помилок і виконання складних перетворень даних;
- Гнучкість: Fetch забезпечує тонкий контроль над об'єктами запиту та відповіді. Він дозволяє конфігурувати заголовки, методи запиту, параметри URL та тіло запиту. Крім того, він підтримує різні формати даних, такі як JSON, FormData та URL-кодовані дані.

Недоліки:

- Відсутність підтримки в старих версіях браузерів: Fetch не підтримується в старих браузерах, таких як Internet Explorer. Однак, це обмеження можна пом'якшити, використовуючи полізаповнювачі або транскриптори, такі як Babel, які дозволяють використовувати Fetch у старіших середовищах;
- Відсутність вбудованого скасування запиту: Fetch не має вбудованого механізму для скасування запитів. Це обмеження може вимагати додаткових обхідних шляхів для ефективною реалізації скасування запитів;
- Обмежена обробка помилок: Fetch відхиляє обіцянку лише у випадку мережевої помилки. Він не відхиляє коди статусу помилки HTTP, такі як 404 або 500. Обробка таких помилок вимагає ручної перевірки статусу відповіді і викидання помилки вручну.

1.5. Висновки до Розділу 1

Даний розділ був присвячений аналізу предметного середовища, де від початку було визначено значення та мету тематики дипломного проекту, а також проведено аналіз ринку кінопорталів, з визначенням їх переваг та недоліків для українського кіноринку.

Після чого було визначено та проаналізовано функціонал майбутнього веб-сервісу.

Для реалізації розробки веб-сервісу було проаналізовано різні технології створення. Аналізу піддалися такі бек-енд технології як Node.js та Django, а також фронт-енд фреймворки такі як React та Angular.

Для інтеграції інформації між сервером і браузером, було проаналізовано декілька різних за будовою систем баз даних, а також систем створення HTTP-запитів.

Отримані в ході аналізу знання і висновки дозволять визначити оптимальний підхід до проектування та реалізації майбутнього веб-сервісу.

РОЗДІЛ 2

АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ ВЕБ-СЕРВІСУ

2.1. Стек технологій MERN

Стек MERN – це набір потужних і надійних технологій для створення онлайн-додатків з бекендом, фронтендом і компонентами бази даних. JavaScript – це мова сценаріїв, яку можна використовувати для швидкого та легкого створення онлайн-додатків на основі стеку. MERN Stack – це просте у використанні JavaScript-рішення для створення динамічних додатків і веб-сторінок.



Рис. 1.3. Стек технологій MERN

Стек MERN складається з чотирьох основних технологій:

- MongoDB (база даних) - база даних NoSQL (неструктурована мова запитів), яка здебільшого використовується для збереження документів;
- Літера E означає Express - мова програмування, яка часто використовується для створення вузлів;

Кафедра КІТ (47)				НАУ 24 33 92 000 ПЗ			
<i>Виконав</i>	<i>Камінський К.В.</i>			АНАЛІЗ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ В РОЗРОБЦІ ВЕБ-СЕРВІСУ	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Шевченко О.П.</i>				У	28	19
<i>Консульт.</i>					УС-413Б 122		
<i>Норм. контр.</i>	<i>Шевченко О.П.</i>						

- R - це React, який здебільшого використовується для створення клієнтських JavaScript-фреймворків;
- Літера N позначає Node.js - широко використовувану мову програмування для створення веб-серверів на JavaScript.

Кожна з цих чотирьох технологій є критично важливою для створення фреймворку з точки зору розробника. У створенні онлайн-додатків ці чотири технології відіграють значну роль.

2.1.1. Структура стеку

MERN має 3-рівневу архітектуру, яка складається з трьох рівнів:

Веб-рівень

React.js відповідає за верхній рівень стеку MERN. Це одна з найпоширеніших бібліотек JavaScript для створення інтерфейсу веб-додатків. Вона добре відома завдяки створенню надійних клієнтських додатків. React дозволяє створювати складні робочі середовища за допомогою лише одного компонента. Він також пов'язує дані на внутрішньому сервері з такими складними взаємодіями. React використовується для створення як мобільних, так і онлайн-додатків (React Native). React дозволяє повторне використання коду та легко підтримує його, що має ряд переваг та економить значну кількість роботи. Це дозволяє користувачам створювати великі веб-додатки з можливістю змінювати дані на сторінці без необхідності перезавантаження сторінки.

Серверний рівень

Він є рівнем нижче верхнього рівня і в основному керується двома компонентами стека MERN: Express.js та Node.js. Оскільки Express.js підтримує окремий серверний фреймворк, який працює всередині сервера Node.js, ці два компоненти керують додатком одночасно. Одним з найпопулярніших підходів до вдосконалення JavaScript Frameworks є використання Express.js. Це дозволяє розробникам швидко і легко знаходити надійні API і веб-сервери. В об'єкти Node.js

HTTP також можна додати корисні функції. З іншого боку, Node.js відіграє важливу роль сам по собі. Це сервер з відкритим вихідним кодом, який працює разом з кодом JavaScript, не вимагаючи використання браузера. Node.js широко використовує JavaScript, дозволяючи користувачам комп'ютерів легко створювати будь-які мережеві сервіси, веб- або мобільні додатки

Рівень бази даних

Основна функція бази даних - зберігати всі дані, пов'язані з вашим додатком, такі як контент, статистика, інформація, профілі користувачів та коментарі. Веб-сайт як категорія бекенду - це один з найважливіших рівнів стеку MERN, який значною мірою управляється MongoDB; основна роль бази даних полягає у зберіганні всіх даних, пов'язаних з вашим додатком, наприклад, контенту, статистики, інформації, профілів користувачів, коментарів і так далі. Вона зберігає більшість даних з міркувань безпеки. Дані зберігаються на веб-сайті. База даних створює дві або більше копій файлів даних, щоб у разі збою системи користувач міг отримати ідентичну інформацію або дані, які він запитував. MongoDB - це добре відомий веб-сайт NoSQL (NoSQL або мова неструктурованих запитів), який є текстовою базою даних. Фраза "NoSQL" зазвичай відноситься до окремого веб-сайту, який не потребує визначеної схеми або пов'язаних таблиць взаємодії для зберігання необхідних йому даних. MongoDB зберігає дані в унікальній структурі, яка не містить схожих таблиць, рядків або стовпців.

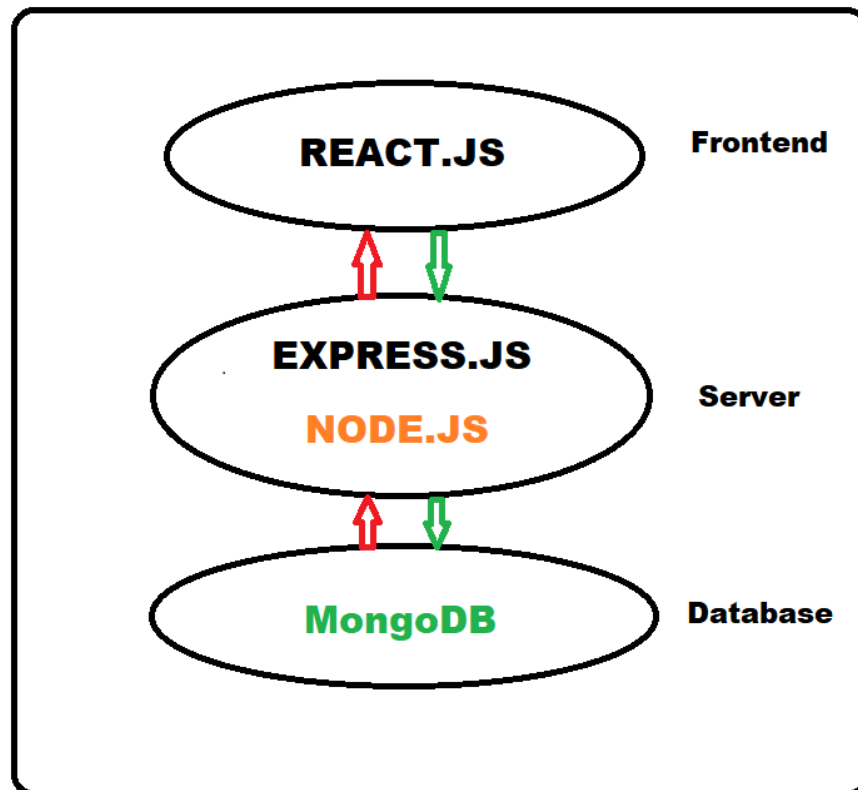


Рис. 1.4. Структура MERN стеку

2.1.2. MongoDB

MongoDB - це добре відомий веб-сайт NoSQL (NoSQL або Non Structured Query Language), який є текстовою базою даних.

Фраза "NoSQL" зазвичай відноситься до окремого веб-сайту, який не потребує визначеної схеми або пов'язаних таблиць взаємодії для зберігання необхідних йому даних. MongoDB зберігає дані в унікальній структурі, яка не містить схожих таблиць, рядків або стовпців.

Це означає, що MongoDB не сумісна з табличною структурою сайту. З іншого боку, це абсолютно новий підхід до збору та зберігання даних. Формат зберігання даних відомий як BSON. Його двійкова структура включає коди для довжини та типу інформації, що дозволяє швидко її ідентифікувати. При зберіганні документів у

кластерах MongoDB використовує BSON, що дозволяє створювати максимально пристосований та адаптований формат документів.

Завдяки покращеним процедурам обслуговування та ідентифікації, вона значно швидша за RDBMS.



Рис. 1.5. База даних MongoDB

Складні методи об'єднання недоступні в MongoDB, тому вона не може обробляти складні транзакції.

Однією з ключових переваг MongoDB є те, що вона використовує JavaScript для шифрування даних.

База даних немає схеми для даних, які зберігаються в окремому документі. У MongoDB немає поняття зв'язку або структури таблиць, як у RDBMS, де таблиці мають певний зв'язок між собою. Вона також має налаштовану архітектуру документів, що дозволяє будь-якому інженеру писати документи набагато швидше.

Однією з найпопулярніших форм баз даних NoSQL є MongoDB. Коли ми розуміємо, що веб-сайт, який працює на одному сервері, наближається до своєї межі, він масштабується і потребує виняткову продуктивність. MongoDB - це база даних NoSQL, яка розширюється за рахунок додавання нових серверів і підвищує продуктивність завдяки гнучкій архітектурі документів.

2.1.3. Express

Express - це серверний JavaScript-компонент та один з найефективніших інструментів для створення JavaScript Frameworks. Він дає розробнику основу для створення та управління надійними серверами.

Express - це мова програмування, яку можна використовувати для швидкої та легкої розробки онлайн і мобільних додатків. Як результат, Express часто використовується для створення серверної логіки для мобільних та онлайн додатків.

Express дозволяє розробникам швидко і легко знаходити надійні API (інтерфейси прикладного програмування) і веб-сервери.

Express спрощує налаштування надійних веб-серверів для вашого додатку за допомогою маршрутизатора та проміжного програмного забезпечення.

Також він додає корисні функції до об'єктів Node.js HTTP (HyperText Transfer Protocol). Це важливий компонент стеків MERN і MEAN, який використовується для створення швидших, безпечніших і ефективніших веб-додатків.



Рис. 1.6. Серверний фреймворк ExpressJS

Серед інших важливих аспектів Express можна виділити наступні:

- Express спрощує та прискорює створення онлайн та мобільних додатків на Node.js;
- Express можна легко встановити та налаштувати на комп'ютері без особливих зусиль;

- Express спрощує підключення до баз даних, таких як MongoDB;
- Express дозволяє вам вказувати шляхи вашого додатку за допомогою HTTP-методів та URL-адрес. Основна мета маршруту - описати код, який повинен виконуватися у відповідь на кожен запит, який отримує сервер. Найпоширенішим способом маршрутизації є використання послідовності шаблонів URL-адрес і методу HTTP, пов'язаного з додатком;
- Якщо ви припустилися помилки і хочете швидко її виправити, ви можете скористатися налагоджувальним проміжним програмним забезпеченням. Проміжне програмне забезпечення використовується локально протягом життєвого циклу запиту або відповіді у вигляді коду. Воно широко використовується для покращення продуктивності або поведінки веб-сервера;
- REST API також можуть бути побудовані за допомогою Express;
- Можливість використовувати величезний набір сторонніх бібліотек Express для підвищення продуктивності, безпеки та швидкості роботи;
- Він добре працює і його можна знайти майже в усіх програмних середовищах, і використовувати в декількох програмах одночасно.

2.1.4. React

React - одна з найпоширеніших бібліотек JavaScript з відкритим кодом для веб-розробки.

Перш ніж використовувати React, необхідно виконати декілька умов, наприклад, встановити найновіші пакети Node на комп'ютер, і бути знайомим з HTML, CSS та JavaScript.

React використовується для створення користувацьких посилань, особливо в односторінковому веб-інтерфейсі. Тут немає фреймворку JavaScript, бо це, по суті, бібліотека JavaScript, розроблена Facebook для вирішення труднощів, які раніше можна було вирішити за допомогою інших бібліотек при розробці онлайн і мобільних додатків.

У мобільних та онлайн-додатках React також використовується для захоплення шару перегляду, що дозволяє розробляти багаторазові UI компоненти.

React був створений SE Jordan Walke, який на той час працював у Facebook.

Вперше React був опублікований на сервері новин Facebook, і він дозволяв розробникам створювати великі веб-додатки з можливістю змінювати дані на сторінці без необхідності перезавантаження сторінки.

Інші бібліотеки JavaScript або фрейми також використовуються разом з React.

Існує деякі й інші фреймворки з відкритим вихідним кодом, які роблять складні онлайн та мобільні додатки простішими, наприклад, Angular js в MVC, але React - це заміна стеку MEAN на Angular. Багато розробників зараз віддають перевагу стеку MERN, оскільки він порівняно швидший і має більше переваг, ніж його попередники.



Рис. 1.7. Бібліотека React

React також має такі переваги як:

- Простота у вивченні та використанні. Однією з найпривабливіших особливостей react є те, наскільки просто новачкам вивчити та використовувати цей просунутий фреймворк для створення веб- та мобільних додатків;
- Якщо порівнювати його з Angular, то будь-хто, хто має фундаментальне розуміння систем, може швидко розібратися з React. Оскільки Angular відома як "мова прямого домену", вважається, що її складно зрозуміти.

Щоб почати працювати з React JS, нам знадобиться коротке розуміння CSS та HTML;

- React - це один з найпростіших фреймворків JavaScript для розробки онлайн і веб-додатків з відкритим вихідним кодом. Він використовує поетапну стратегію, простий і зрозумілий JavaScript та чітко визначений життєвий цикл, що робить роботу з ним простішою і зрозумілішою;
- Кращий для простого читання та розробки мобільних і веб-додатків промислового рівня. Вона використовує базовий синтаксис, відомий як JSX, що дозволяє читачам або розробникам змішувати HTML з JavaScript для створення більш ефективних онлайн і мобільних додатків. Втім, вам не обов'язково використовувати JSX; ви можете використовувати звичайний JavaScript. Однак, завдяки своїй простоті та граматиці, JSX є значно кращим вибором;
- React надзвичайно простий у тестуванні.

2.1.5. Node.JS

Node.js - це сервер з відкритим вихідним кодом, який дозволяє використовувати код JavaScript без використання браузера.

Багато фонових додатків, таких як мережеві та мобільні додатки, часто створюються та оновлюються з його допомогою. Uber, PayPal, Netflix та інші великі корпорації використовують його переважно у виробництві.

Він може використовуватися як безкоштовний ASCII-архів текстових файлів будь-ким, а також він працюватиме на різних операційних системах, включаючи Windows, Mac OS X, Linux, Unix та інші.



Рис. 1.8. Середовище виконання Node.JS

Певними перевагами є:

- Легка масштабованість: завдяки використанню єдиної архітектури послідовних подій, Node.js є неймовірно масштабований. Завдяки процесу обробки подій, сервер зазвичай відповідає не блокуючи. Це також дозволяє серверу масштабуватися набагато більше, ніж звичайні сервери, які створюють лише певну кількість потоків для обробки додатків. Node.js базується на односерійному додатку, який може надавати послуги багатьом додаткам;
- Прискорення: оскільки цикл обробки подій Node.js обробляє всі пов'язані з ним процеси, він функціонує як швидкий пакет, і всі завдання, мережева комунікація, читання і запис на веб-сторінці, програмування файлів - це ті завдання, які виконуються швидко. Він базується на двигуні V8 від Google;
- Оскільки Node використовує JavaScript для запуску коду для веб-проектів та інших онлайн і мобільних додатків, її легко читати і виправляти. Ви можете просто конструювати і виконувати програмне забезпечення на Node.js і досліджувати його скільки завгодно, якщо ви успішні в

просунутому програмуванні і добре володієте JavaScript; він покладається на ваші сили;

- Веб-додатки в режимі реального часу: JavaScript необхідний для веб-додатків, що працюють в режимі реального часу. Ви також можете використовувати PHP для створення мобільного телефону або веб-додатку, хоча це займе стільки ж часу, як і Node.js. Однак, завдяки швидкій синхронізації, Node.js є найкращим рішенням для створення ігрових додатків та чат-систем;
- Перевага кешування: js дозволяє створювати тимчасове сховище для одного модуля. Вам не завжди потрібно повторно використовувати ті самі рядки коду, оскільки вони вже збережені в архіві Node.js;
- Розподіл даних: У Node.js запити протоколу передачі гіпертексту (HTTP) та блоки відповідей розглядаються як 2 окремі події. Вони є потоками інформації, тому після того, як ви відредагували файл під час завантаження, він сповільниться і створиться, як тільки інформація буде надана в стилі передачі. Ви також можете передавати музичні та відеофайли з шаленою швидкістю;
- Найбільшою критикою Node.js був його JavaScript, який часто мав багато спагеті-скриптів для обробки даних. Coffee Script і Typescript були створені для подолання цих проблем, але вони стали перешкодою для інших, хто більше переймався стандартами коду. З регулярним випуском і прийняттям ES6 класи були включені у фреймворк, і код став ідентичним до C#, Java та SWIFT;
- Сервери на основі Node.js не повинні чекати, поки пов'язаний API поверне інформацію. Коли сервер працює, він переходить до наступного API, а система сповіщень Node.js Events допомагає серверу отримувати зворотній зв'язок про попередні обрані API;
- Підтримка бізнесу: Node.js використовується для розробки додатків кількома відомими компаніями, такими як PayPal, Wal-Mart, Microsoft та

Google. Оскільки Node.js базується на JavaScript, більшість компаній об'єднують свої фронт- і бек-офіси на Node.js.

2.1.6. Забезпечення захисту стеку

У цифрових інноваціях стек MERN виділяється як потужна структура для створення динамічних веб-додатків. Однак із великими можливостями виникає відповідальність за забезпечення надійних заходів безпеки. Оскільки додатки стека MERN продовжують швидко зростати, вони неминуче привертають увагу зловмисників. Таким чином, пріоритет безпеки є важливим для захисту конфіденційних даних і підтримки довіри користувачів.

Щоб запобігати несанкціонованим діям з боку зловмисників, потрібно дотримуватися декільком заходам безпеки:

- **Перевірка та обробка введених даних:** перевірка введених даних гарантує, що дані, введені користувачами, відповідають певним критеріям, таким як формат, довжина та тип, тоді як дезінфекція видаляє потенційно шкідливі символи чи код, щоб запобігти атакам або іншим уразливостям системи безпеки;
- **Автентифікація та авторизація:** автентифікація перевіряє особу користувачів, які намагаються отримати доступ до програми, а авторизація визначає дії, які їм дозволено виконувати на основі їхніх ролей і дозволів. Захищені методи автентифікації, такі як JWT або OAuth, забезпечують надійну структуру для автентифікації користувачів, а також хешування паролів за допомогою надійних криптографічних алгоритмів, таких як bcrypt, забезпечує надійне зберігання паролів у базі даних;
- **Безпечна конфігурація:** безпечна конфігурація передбачає налаштування середовища програми та інфраструктури таким чином, щоб мінімізувати ризики безпеки та вразливості. Правильна конфігурація CORS обмежує перехресні запити, щоб запобігти неавторизованому доступу до конфіденційних ресурсів;

- Обробка та запис помилок: ефективні механізми обробки помилок і журналювання забезпечують видимість поведінки програми, полегшують налагодження та моніторинг, а також допомагають виявити потенційні загрози безпеці;
- Керування сеансами: захищене керування сеансами передбачає безпечну підтримку сеансів користувачів, щоб запобігти несанкціонованому доступу, фіксації сеансів або атакам на захоплення сеансів;
- Оновлення залежностей: регулярне оновлення залежностей є важливим для пом'якшення вразливостей безпеки та забезпечення загальної стабільності програми. Застарілі залежності можуть містити відомі недоліки безпеки, якими можуть скористатися зловмисники.

2.2. Програмні залежності для веб-сервісу

Програмна залежність - це багаторазово використовувана бібліотека коду або пакет, який дозволяє швидко створювати програмне забезпечення, оскільки дозволяє розробникам спиратися на попередні напрацювання. Середовище виконання JavaScript Node.js використовує Node Package Management, або npm, як стандартний менеджер пакетів. Він складається з клієнта командного рядка npm та реєстру npm, онлайн-репозиторію для безкоштовних і платних приватних пакетів. Клієнт використовується для доступу до реєстру, а веб-сайт npm використовується для перегляду та пошуку пакетів, які наразі доступні.

Існують програмні залежності з боку сервера такі як:

- **mongoose**. Драйвер MongoDB Node.js дозволяє додаткам Node.js підключатися до MongoDB і працювати з даними за допомогою асинхронного API, який використовує Promises або звичайні виклики для взаємодії;
- **mongoose** – це в'язок між MongoDB та середовищем виконання JavaScript Node.js, який забезпечується за допомогою об'єктно-орієнтованого пакету програмування JavaScript під назвою Mongoose;

- **nodemon** - це програма, яка полегшує створення додатків на основі Node.js, перезапускаючи нод-додаток щоразу, коли виявлено зміну файлу в каталозі. Це замітник обгортки вузла;
- **jsonwebtoken**. Стислий, URL-безпечний метод кодування запитів, якими потрібно обмінятися між двома сторонами, - це JSON Web Token (JWT). У JSON Web Token (JWT) вимоги кодуються як JSON-об'єкт. Він використовується як відкритий текст структури JSON Web Encryption (JWE) або як корисне навантаження структури JSON Web Signature (JWS), що дозволяє зашифрувати, підписати цифровим підписом або захистити цілісність за допомогою коду аутентифікації повідомлення (MAC);
- **express**. Express - це простий та адаптивний фреймворк для веб-додатків Node.js, який пропонує широкий вибір функціональних можливостей як для веб-додатків, так і для мобільних додатків. Він полегшує швидке створення потужного API, надаючи доступ до широкого спектру методів HTTP-утиліт та проміжного програмного забезпечення;
- **dotenv**. Dotenv - це модуль з нульовою залежністю, який завантажує process.env змінними оточення з файлів .env. Техніка Twelve-Factor App використовується для зберігання налаштувань у середовищі окремо від коду. Тут ми зберігаємо неопубліковані змінні оточення, такі як MONGODB_URL, OMDb_API, SECRET_KEY, доступ до яких здійснюється з файлу .env в нашому коді;
- **cors**. Сервер може вказати будь-яке походження (домен, схему або порт), відмінне від його власного, з якого браузер повинен мати можливість завантажувати ресурси за допомогою протоколу Cross-Origin Resource Sharing (CORS), який базується на заголовку HTTP. В результаті ми можемо запустити сервер і клієнт на різних доменах, розділивши їх;
- **bcrypt** - функція хешування паролів Нільса Провоса та Девіда Мазьєреса, bcrypt, яка була вперше представлена в 1999 році на конференції USENIX, і була побудована на основі шифру Blowfish. Bcrypt є адаптивною

функцією, яка може сповільнюватися з часом при збільшенні кількості ітерацій, що робить її більш стійкою до атак грубого перебору навіть при зростаючих обчислювальних потужностях.

Натомість, є програмні залежності і з клієнтської сторони:

- **axios**. Axios - це клієнтська бібліотека HTTP на основі Promesis. Вона полегшує CRUD-діяльність та доставку асинхронних HTTP-запитів до кінцевих точок REST. Цією REST-кінцевою точкою/API може бути наш власний сервер Node.js або зовнішній API, такий як Google API, GitHub API, тощо. Найпоширенішими запитами до API є get, post та delete, тому що нам завжди потрібно отримувати дані для відображення в наших додатках, а також додавати і видаляти дані з і до нашого API;
- **react**. React - це бібліотека JavaScript, яка використовується для створення користувацьких інтерфейсів. До складу пакунку входить лише функціональність, необхідна для визначення React-компонентів компонентів міститься в пакеті react. Рендерер React рендерер, такий як react-dom для веб або react-native для нативних додатків, зазвичай використовується разом з ним. Коли дані змінюються, React швидко оновлює та рендерить відповідні компоненти;
- **react-dom**. Серверний та DOM-рендерери для React доступні через цей пакет. Він призначений для використання разом з пакетом react, який містить основні компоненти React. За необхідності, пакет react-dom пропонує специфічні для DOM функції, які можуть бути використані як чорний хід для виходу з моделі React;
- **react-router-dom**. npm-пакет під назвою React Router DOM дозволяє інтегрувати динамічну маршрутизацію у динамічну маршрутизацію у веб-додатку. Його можна використовувати для відображення сторінок та для навігації відвідувачів. Це повна клієнтська та серверна бібліотека маршрутизації для React Router DOM. Створення односторінкових додатків або програм з великою кількістю сторінок або компонентами, але без оновлення сторінок, оскільки вміст динамічно завантажується з URL-

адреси, вимагає використання React Router Dom. React Router Dom робить цю процедуру цю процедуру, яка відома як маршрутизація.

2.3. OMDb API

Веб-сервіс кінопорталу потребує надійного і повного джерела даних про фільми, щоб надавати користувачам точну і актуальну інформацію. API OMDb (Open Movie Database) є популярним вибором для таких цілей, пропонуючи величезне сховище інформації про кіно, що включає деталі про фільми, телевізійні шоу та акторів.

API OMDb пропонує кілька переваг для розробників кінопорталів:

- Вичерпні дані: API надає доступ до широкого спектру інформації, пов'язаної з фільмами, включаючи назви, жанри, дати виходу, короткий опис сюжету, рейтинги та інформацію про акторів;
- Простота використання: API OMDb розроблений таким чином, щоб бути зрозумілим, з простими HTTP-запитами та JSON-відповідями, які дозволяють легко інтегрувати його у веб-додатки;
- Економічна ефективність: API пропонує безкоштовний рівень з базовими функціями, що робить його доступним для невеликих проектів і стартапів. Для більш широкого використання доступні платні плани;
- Актуальна інформація: регулярні оновлення гарантують, що дані залишаються актуальними, дозволяючи користувачам отримувати доступ до найновішої інформації про фільми та їхні рейтинги.

Щоб використовувати API OMDb при розробці веб-сервісу, розробники повинні виконати наступні кроки:

1. Реєстрація для отримання ключа API: відвідуючи веб-сайт OMDb API зареєструватися для отримання ключа API. Цей ключ необхідний для аутентифікації запитів до API.

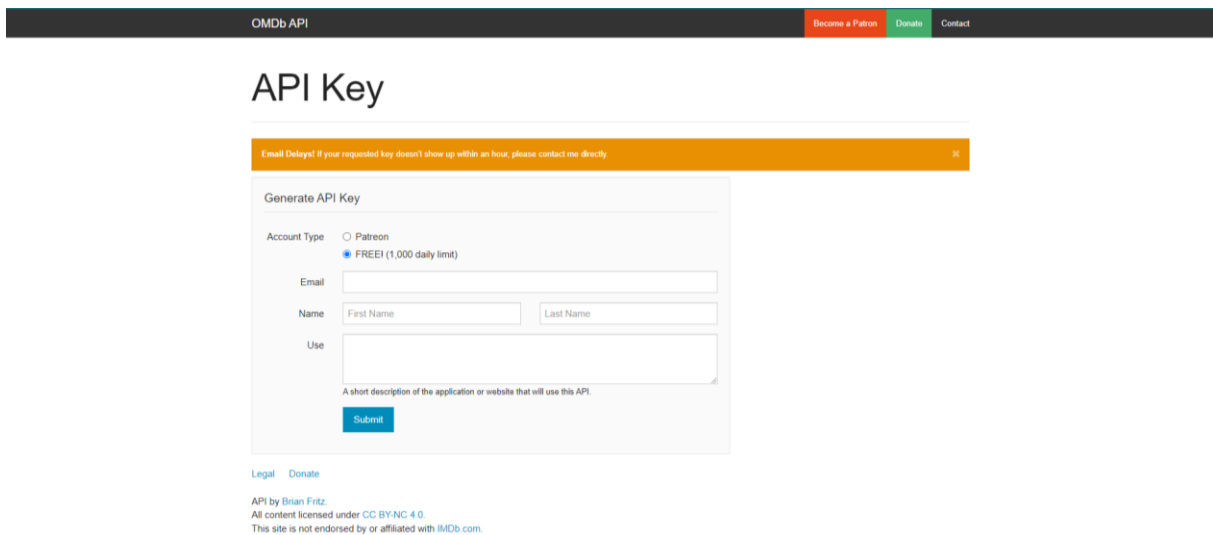


Рис. 1.9. Реєстрація для отримання ключа OMDb API

2. Документація API: ознайомлення з документацією API, щоб зрозуміти доступні кінцеві точки, параметри запитів і формати відповідей.

2.4. Середовище розробки

Visual Studio Code - це безкоштовний, легкий, але потужний редактор вихідного коду, який працює на робочому столі та в Інтернеті і доступний для Windows, macOS, Linux та Raspberry Pi OS. Він має вбудовану підтримку JavaScript, TypeScript та Node.js, а також багату екосистему розширень для інших мов програмування (таких як C++, C#, Java, Python, PHP та Go), середовищ виконання (таких як .NET та Unity), середовищ (таких як Docker та Kubernetes) та хмар (таких як Amazon Web Services, Microsoft Azure та Google Cloud Platform).

Окрім легкого та швидкого запуску, Visual Studio Code має функцію завершення коду IntelliSense для змінних, методів та імпортованих модулів; графічне налагодження; лінкування, багатокурсорне редагування, підказки параметрів та інші потужні функції редагування; зручну навігацію коду та рефакторинг; а також вбудований контроль вихідного коду, включаючи підтримку Git'у. Багато з цього було адаптовано з технології Visual Studio.

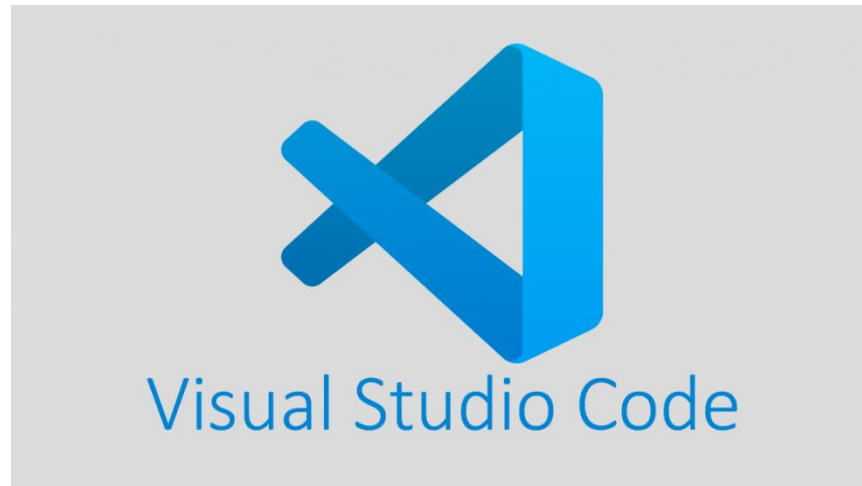


Рис. 1.10. Текстовий редактор VS Code

Власне код Visual Studio створено за допомогою оболонки Electron, Node.js, TypeScript та протоколу Language Server Protocol, який оновлюється щомісяця. Численні розширення оновлюються так само часто, як це необхідно. Підтримка різних мов програмування та їх розширень варіюється від простого підсвічування синтаксису та узгодження дужок до налагодження та рефакторингу. Ви можете додати базову підтримку для вашої улюбленої мови за допомогою колоризаторів TextMate, якщо немає мовного сервера.

До основних переваг VS Code відносять:

- Гнучкість і налаштованість: VS Code легко налаштовується за допомогою плагінів. Ви можете додати лише ті функції, які вам потрібні;
- Полегшений редактор: він починається як базовий текстовий редактор, що робить його швидшим і більш гнучким;
- Підтримка багатьох мов: легко налаштовується для різних мов, таких як C, C++ та Python, за допомогою простих конфігурацій плагінів;
- Універсальність: може використовуватися для широкого спектру завдань, таких як редагування та перегляд файлів розмітки, створення та створення документів LaTeX.

До суттєвих недоліків VS Code відносять:

- Вимагає налаштування: на відміну від Visual Studio, VS Code потребує ручного налаштування компіляторів, налагоджувачів та мовних особливостей;
- Залежність від плагінів: вам потрібно встановити плагіни для базових функцій, таких як IntelliSense, підсвічування синтаксису та налагодження.

2.5. Висновки до розділу 2

У даному розділі було проаналізовано та вивчено можливість розробки проекту за допомогою стеку MERN. Було проаналізовано основні аспекти та визначено переваги при використанні всіх компонентів стеку (MongoDB, Express, React, NodeJs) разом. Також було визначено основні аспекти безпеки під час роботи зі стеком, і визначено заходи для профілактики безпеки.

Виведено основні програмні залежності для роботи з серверними та клієнтськими рівнями, та проаналізовано їх використання разом з NodeJs.

Проаналізовано використання OMDB API під час реалізації проекту та визначено переваги його використання. Підкреслено доцільність використання Visual Studio Code та визначено переваги та недоліки його використання.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РОЗРОБКА КІНОПОРТАЛУ

3.1. Система збереження ключей

Для початку потрібно буде скористатися програмною залежністю dotenv.

Dotenv — це популярний модуль для керування змінними середовища, який завантажує конфігурацію з .env файлу в process.env. Незважаючи на важливість включення Dotenv у проект, він напрочуд простий у використанні.

Файл .env створюється наступним чином: створити файл із назвою .env у кореневому каталозі нашого проекту та додати змінні середовища, які потрібно приховати.

```
.env
1 MONGODB_URI=mongodb+srv://fintk:KBijiP8UxIoX1aqh@cluster0.kppjnx.mongodb.net/
2 OMDB_API=http://www.omdbapi.com/?apikey=da7f2d14&
3 SECRET_KEY=cher1nio
```

Рис. 1.11. Файл .env

Потім можна отримати доступ до цих ключів у своєму коді, імпортувавши модуль Dotenv.

```
1 require("dotenv").config()
```

Рис. 1.12. Приклад отримання доступу до файлу .env

Кафедра КІТ (47)				НАУ 24 33 92 000 ПЗ			
<i>Виконав</i>	<i>Камінський К.В.</i>			ПРОЕКТУВАННЯ ТА РОЗРОБКА КІНОПОРТАЛУ	<i>Літера</i>	<i>Аркуш</i>	<i>Аркуші</i>
<i>Керівник</i>	<i>Шевченко О.П.</i>				У	47	16
<i>Консулт.</i>					УС-413Б 122		
<i>Норм. контр.</i>	<i>Шевченко О.П.</i>						

В нас були створені ключі:

- MONGODB_URL – для підключення до бази даних MongoDB;
- OMDB_API – для підключення до Omdb API;
- SECRET_KEY – секретний ключ згенерований для JWT для запобігання розкриття конфіденційної інформації користувача.

3.2. Реалізація структури БД

Після підключення бази даних до проекту, можна приступити до реалізації структури БД. Насамперед, так як в нас використовується база даних MongoDB, це означає що база даних в нас не реляційна. Нереляційні бази даних відрізняються від традиційних реляційних баз даних тим, що вони зберігають дані в нетабличній формі. Натомість нереляційні бази даних можуть базуватися на таких структурах даних, як документи. Документ може бути дуже детальним, водночас містити низку різних типів інформації в різних форматах.

Отже спочатку ми реалізуємо підключення до БД через URL адрес.

```
//Connect to the DB
mongoose
  .connect(url, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then((result) => {
    console.log("You are connected to the DB")
  })
  .catch((error) => {
    console.log(error)
    console.log("Could not connect to DB")
  })
```

Рис. 1.13. Програмне підключення до БД

Далі ми визначаємо модель даних збережених для користувача, які будуть визначати його функціональні потреби.


```
const usersSchema = new mongoose.Schema({
  username: String,
  password: String,
  avatar: String,
  follows: Array,
  watched: Array,
  toWatch: Array,
})
```

Рис. 1.14. Дані користувача для БД

Тобто для користувача буде реалізовано дані його: ім'я, пароль, аватар (фото профілю), масив підписників, масив списку переглянутих та списку фільмів для перегляду пізніше.

Так само потрібно визначити набір функціональних даних для відгуків на фільми.

```
const reviewSchema = new mongoose.Schema({
  username: String,
  titleid: String,
  timestamp: String,
  stars: Number,
  content: String,
  likes: Array,
  comments: Array,
})
```

Рис. 1.15. Дані відгуків для БД

Тобто так само для відгука буде реалізовано набір даних: ім'я користувача (що зробив відгук), id назви фільму, час відгуку, кількість зірочок для відгуку, текст відгуку, кількість лайків для відгуку та коментарі до відгуку.

Моделі для збереження даних визначено і далі ми їх робимо доступними для імпорту в інші частини програми, що буде дозволяти іншим модулям взаємодіяти з даними колекціями.

3.3. Реалізація серверного рівня веб-сервісу

Серверний рівень призначений для підключення до БД, та для створення та обробки запитів на витягування інформації з БД та API. Насамперед, для створення можливості маршрутизації буде використовуватися модуль `express.Router()`, який буде відповідальним за маршрутизації по файловій структурі проекту. Щоб спростити роботу з API та зменшити витрати часу на маршрутизацію використаний модуль `fs(file system)`. Всі HTTP запити групуються в залежності від того, стосується це користувача, входу в систему чи відгуків.

Модуль `userRouter`

Модуль `userRouter` реалізований задля отримання даних про переглянуті фільми користувача з БД. Працює він таким чином, що спочатку ініціює GET запит за шляхом до вказаним даним, експрес-маршрутизатор викликає метод `User.findOne()` для отримання документу користувача з БД, і таким чином БД повертає масив даних за запитом.

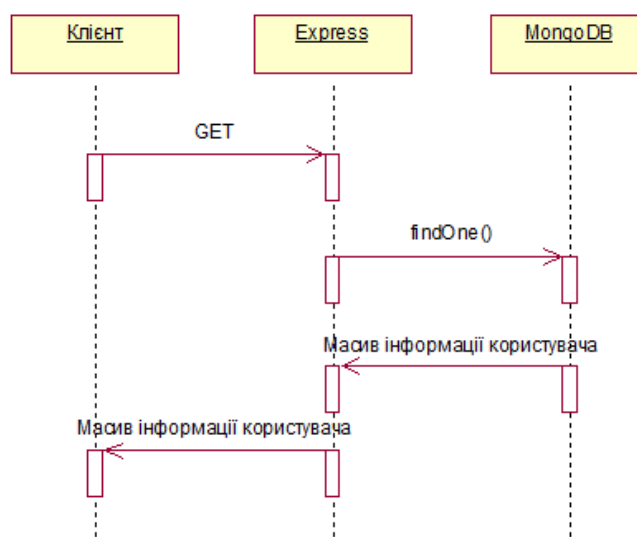


Рис. 1.16. Діаграма послідовності `userRouter()`

Модуль IMDBRouter

Схожим чином реалізовано модуль IMDBRouter, який працює з API, і слугує для маршрутизації та отримання даних про фільми. Проте в даному випадку використовується додатково бібліотека Axios задля надсилання HTTP запитів до браузера і отримання даних.

Реалізовано це таким чином, що спочатку ініціюється GET запит за певним потрібним параметром фільму, експрес-маршрутизатор обробляє цей маршрут, і вже потім axios надсилає HTTP запит до IMDB API. IMDB API обробляє запит та повертає дані фільму.

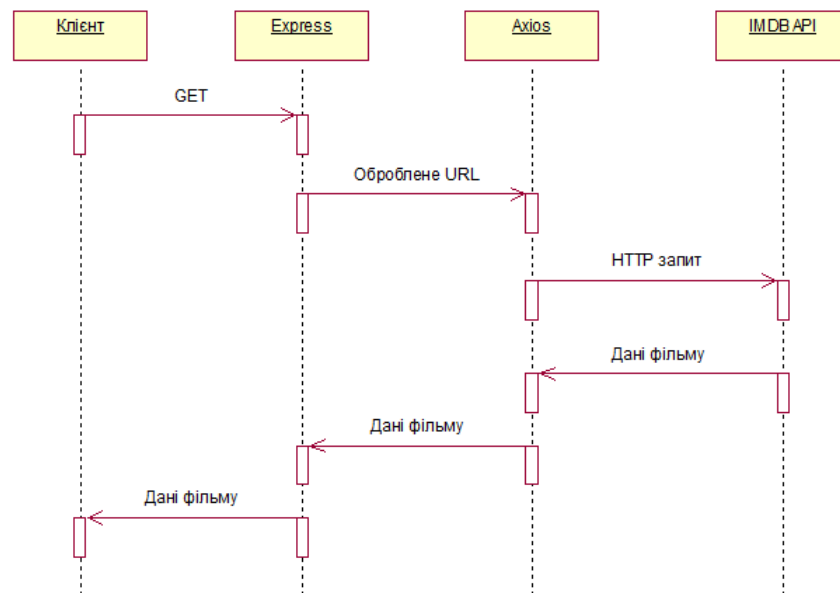


Рис. 1.17. Діаграма послідовності IMDBRouter

Модуль apiRouter

Модуль apiRouter призначений для маршрутизації та обробки різних кінцевих точок API таких як:

- Автентифікація користувача. Тут перевіряється облікові дані користувача, генерується особливий для кожного користувача JWT token після успішної авторизації та повертається цей токен користувачу для наступним авторизацій.

- Керування користувачами. Створюється новий користувач з новим набором даних для БД та оновлюються дані про користувача.
- Керування відгуками. Створюється новий відгук з оновленим надобром даних для БД, оновлюються дані про цей відгук та перевіряється на видалення відгуку.

Реалізація серверного рівня

Сам сервер реалізується в файлі `server.js` та використовує бібліотеку `cors` для завантаження сервером будь-якого домену чи порту з якого браузер повинен дозволити завантажувати будь-які ресурси.

За замовчування отримується номер порту 3001, і сервер запускається на цьому порту та прослуховується запуск. При успішному запуску, виводиться повідомлення.

```

require("dotenv").config()
const http = require("http")
const express = require("express")
const cors = require("cors")
const bodyParser = require("body-parser")
const apiRouter = require("./api")
const IMDBRouter = require("./imdbApi")
const userRouter = require("./userRouter")

const app = express()
app.use(cors())
app.use(bodyParser.urlencoded({ extended: true }))
app.use(bodyParser.json())
app.use(express.static("build"))

app.use(apiRouter)
app.use(IMDBRouter)
app.use(userRouter)

const path = require("path")
app.get("*", (request, response) => {
  response.sendFile(path.join(__dirname, "../build/index.html"))
})

const PORT = process.env.PORT || 3001
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`)
})

```

Рис. 1.18. Реалізація серверного рівня

3.4. Реалізація веб-рівня

Компонент веб-рівня містить маршрутизатор, який створює маршрути до кожної сторінки нашої програми. Якщо змінна стану користувача порожня або

нульова, користувач перенаправляється на сторінку входу, оскільки лише автентифіковані користувачі можуть використовувати цю програму.

Вхід/Реєстрація

Сторінка входу дозволяє користувачам увійти до свого облікового запису перед тим, як отримати доступ до сторінки. Після того, як дані введено і надіслано, вони перевіряються з базою даних, перш ніж користувач буде перенаправлений на головну сторінку. Користувач отримує сповіщення, якщо він успішно ввів правильний або неправильний пароль та ім'я користувача.

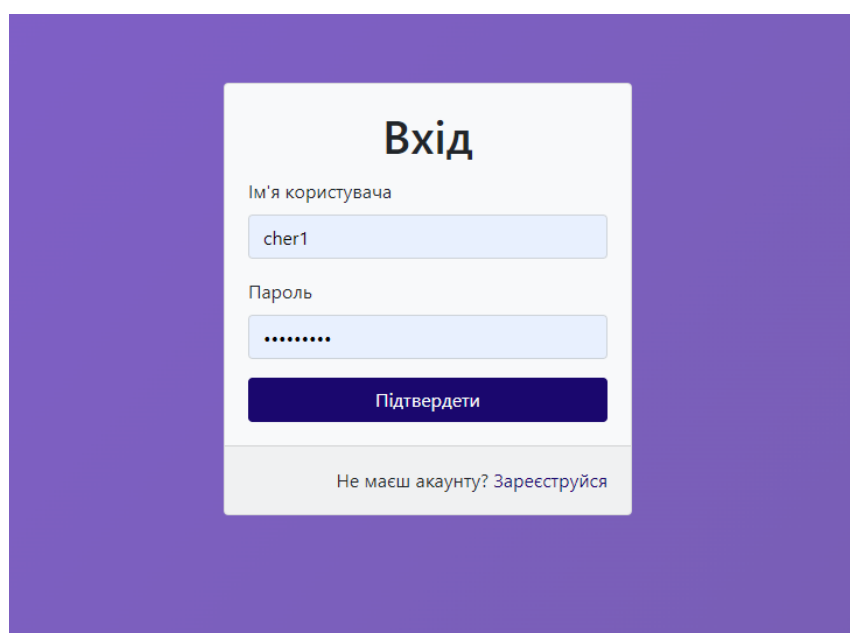
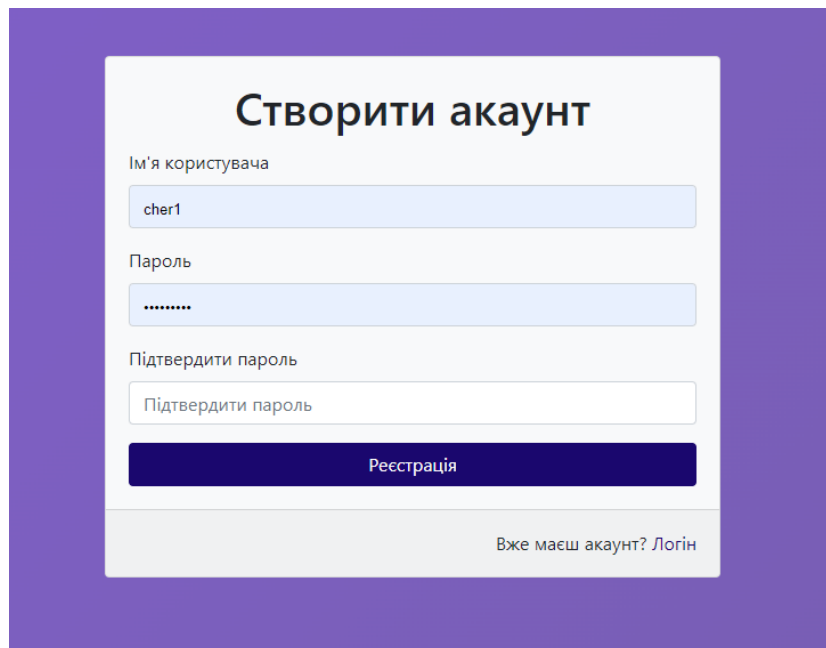


Рис. 1.19. Сторінка входу

На сторінці входу користувач також має можливість зареєструвати обліковий запис. При цьому користувач буде перенаправлений на нову сторінку з кодом з файлу Register.js, який надішле POST-запит до бази даних. Об'єкт буде розміщено лише за умови, що надіслане ім'я користувача доступне. Це досягається за допомогою наступного логічного оператора: `(users.map(u => u.username).includes(value) === true)`



Створити акаунт

Ім'я користувача
cher1

Пароль
.....

Підтвердити пароль
Підтвердити пароль

Реєстрація


Вже маєш акаунт? Логін

Рис. 1.20. Сторінка реєстрації

Інформаційна картка фільму

Картка фільму відображає всю інформацію про досліджуваний фільм. До цього компонента включено кнопку, яка робить PUT-запит до бази даних, що оновлює список переглянутих фільмів або список toWatch для користувачів, які увійшли до системи. Якщо фільм вже є у списку, ця кнопка відобразить це. Ви також можете вибрати кнопку «відгук», яка відобразатиме модальне вікно для написання відгуку. Також є можливість перегляду більшої кількості інформації на кінопорталі IMDb для тих користувачів яким мало інформації на інформаційній картці.

The Avengers (2012)



★★★★★ Твій рейтинг:

Переглянуто

Відгук

| 143 min | Action, Sci-Fi | 04 May 2012

Earth's mightiest heroes must come together and learn to fight as a team if they are going to stop the mischievous Loki and his alien army from enslaving humanity.

Режисери:
Joss Whedon

Сценаристи:
Joss Whedon, Zak Penn

Акторський склад:
Robert Downey Jr., Chris Evans, Scarlett Johansson

Подивитися більше інформації на IMDb

Рис. 1.21. Інформаційна картка фільму

Відгук

Подібно до картки фільму, відгук показує деяку інформацію про фільм, а також оцінку та відгуки користувачів про фільм. Інші користувачі можуть коментувати та вподобати рецензію, і ці зміни вносяться за допомогою PUT-запиту до бази даних, який оновлює властивості вподобань та коментарів до рецензії. Ви можете отримати доступ до профілю користувача, вибравши його ім'я на площині відгуку. Ви також можете додати фільм до списку «Переглянути» з картки рецензії.

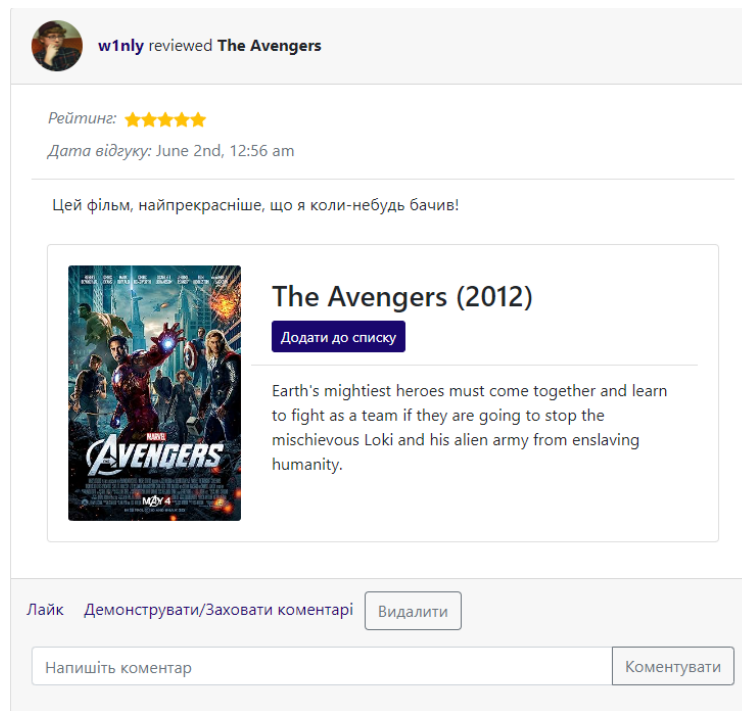


Рис. 1.22. Відгук

Навігація та пошук фільмів

З панелі навігації ви можете отримати доступ до головної стрінки, вашого профілю, списку переглянутих і запланованих до перегляду фільмів, ваших відгуків та сторінки FaQ. Панель навігації також має функціонал, що дозволяє шукати фільми в базі даних OMDb. При виконанні пошуку кнопка пошуку перенаправляє вас на порожню сторінку, що містить картку фільму з найближчою назвою, яка відповідає вашому запиту. Для доступу до бази даних OMDb виконується GET-запит. У крайньому правому куті навігаційної панелі знаходиться зображення, що містить аватар користувача. Це також працює як випадające меню, за допомогою якого можна перейти до свого профілю, написати рецензію на будь-який фільм і вийти з системи. Ця панель навігації з'являється на кожній сторінці, окрім входу та реєстрації.



Рис. 1.23. Навігаційне меню

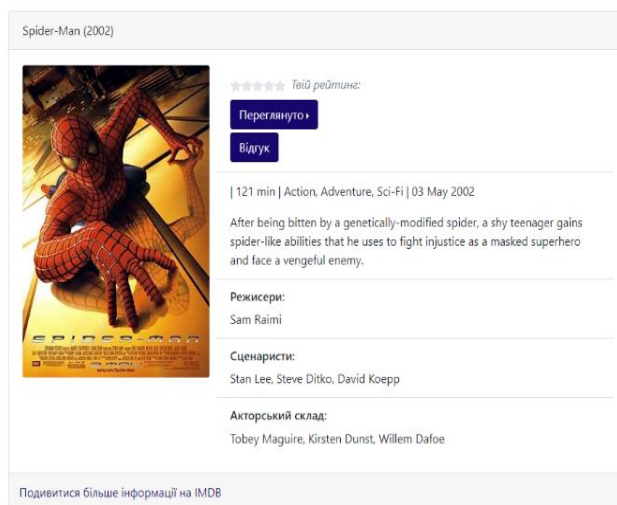


Рис. 1.24. Пошук фільму за назвою

Модальний режим написання відгуку

На сайті є різні кнопки, які демонструють модальний режим «Написати відгук», зокрема, на сторінці «Мої відгуки» та у випадяючому меню на навігаційній панелі. Кнопка «Написати відгук» має два режими. Ви або вирішили написати рецензію, не розкриваючи фільм - у цьому випадку ви можете шукати потрібний фільм у модальному вікні. Інший випадок - якщо кнопка «Відгук» вибрана на картці фільму - в цьому випадку поле «Фільм» буде автоматично заповнене фільмом з картки. Цей код виконує POST-запит до бази даних MongoDB, тільки якщо в полі рецензії є вміст.

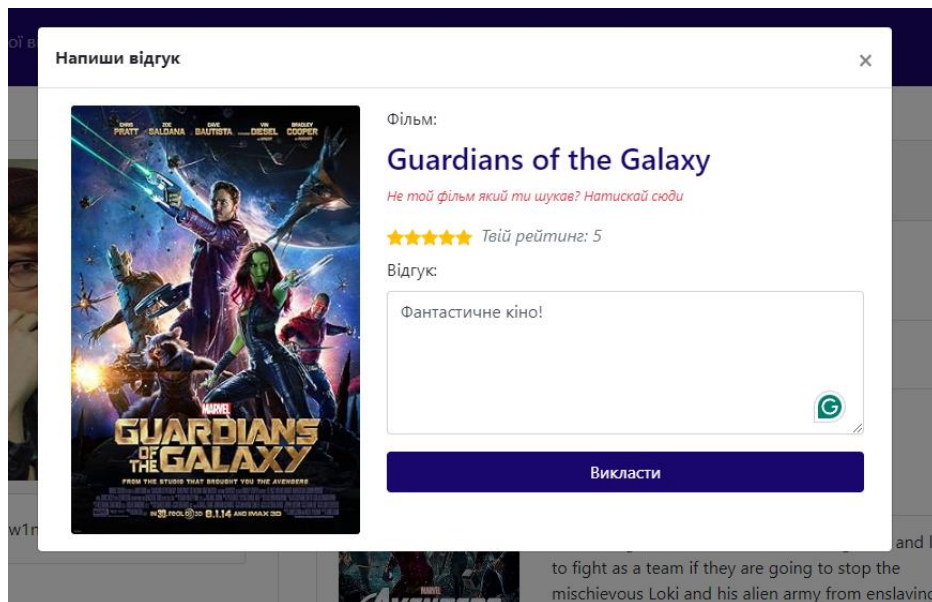


Рис. 1.25. Модальний режим написання відгуку

Головна сторінка

Головна стрічка містить картки всіх відгуків користувачів, за якими ви стежите. Вони впорядковані за датою.

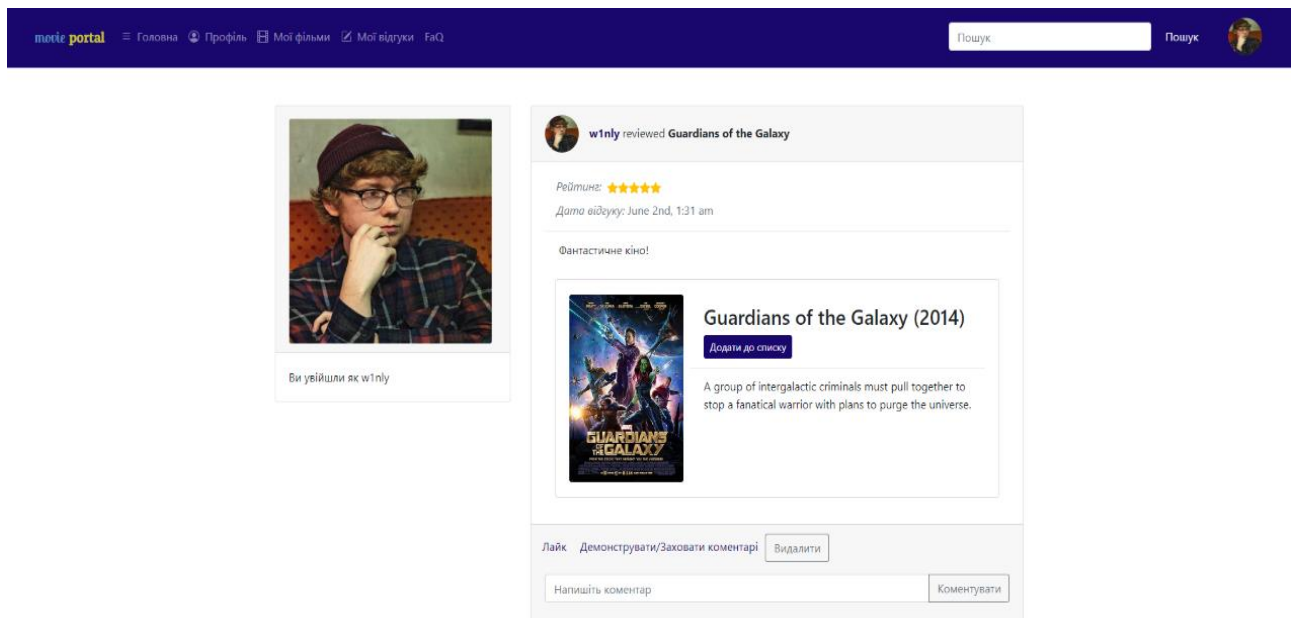


Рис. 1.26. Головна сторінка

Профіль

Сторінка профілю має основний режим перегляду власного профілю. На цій сторінці ви можете переглянути інформацію про свій профіль, а також переглянути колаж постерів фільмів зі списку переглянутих користувачем.

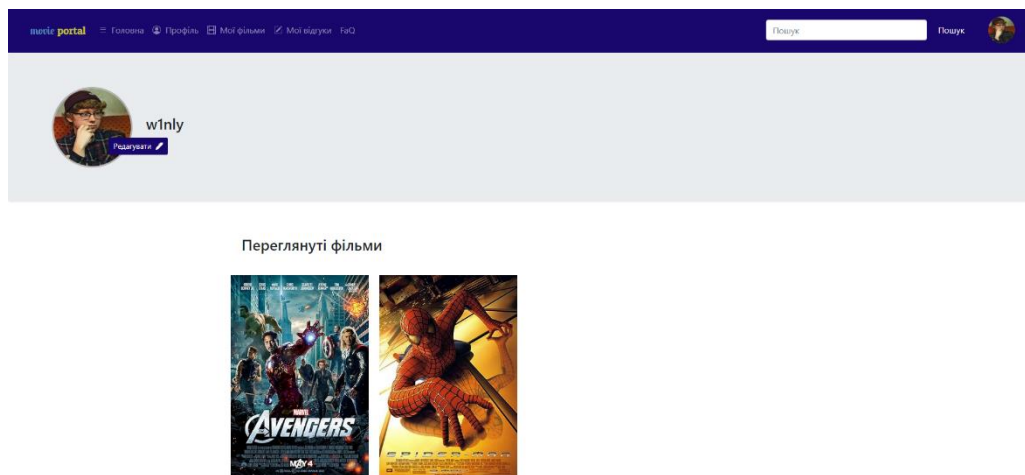


Рис. 1.27. Профіль користувача

Коли ви виберете будь-який з цих постерів, з'явиться модальне вікно, що містить додаткову інформацію про цей фільм.

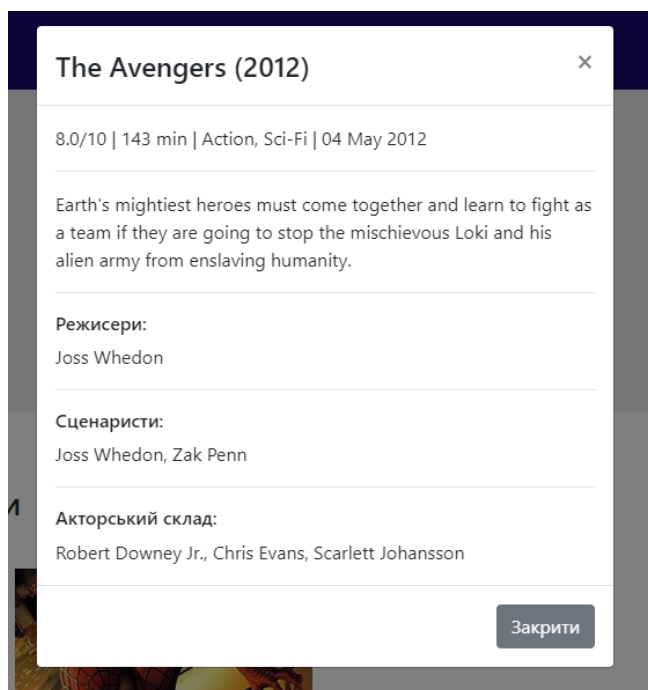


Рис. 1.28. Модальне вікно фільму в профілі користувача

На сторінці профілю також є функція редагування, яка повинна була відкрити модальну форму для оновлення аватарки та біографії користувача. На жаль, через брак часу ця функція не була включена.

Сторінка «Мої фільми»

«Мої фільми» - це сторінка, що містить стрічку карток фільмів, які належать до списків переглянутих або запланованих до перегляду. У верхній частині сторінки є перемикач, який дозволяє перемикатися між цими двома режимами. Змінна стану watched містить значення true або false, залежно від того, який список ви хочете переглянути.

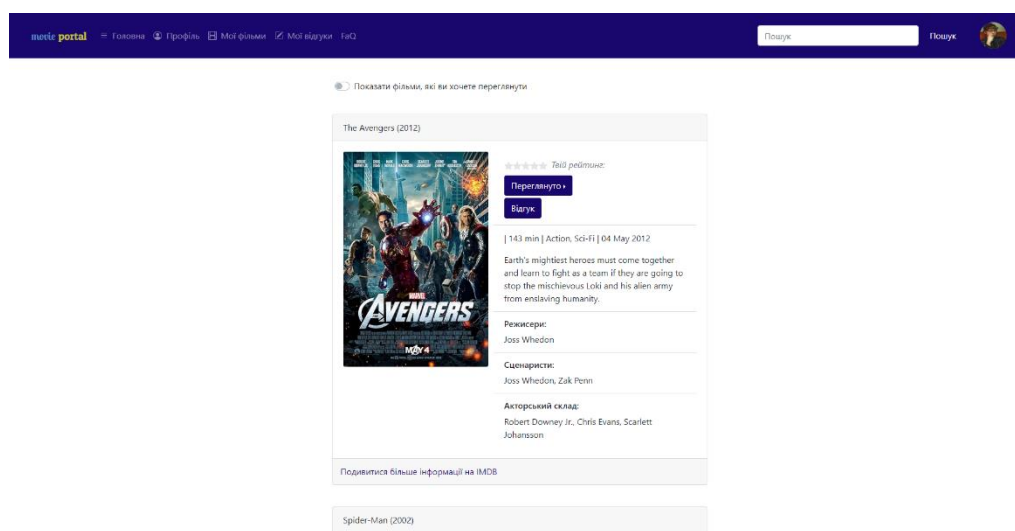


Рис. 1.29. Сторінка «Мої фільми»

Сторінка «Мої відгуки»

«Мої відгуки» - це сторінка, що містить стрічку карток усіх відгуків, залишених користувачем, який увійшов до системи.

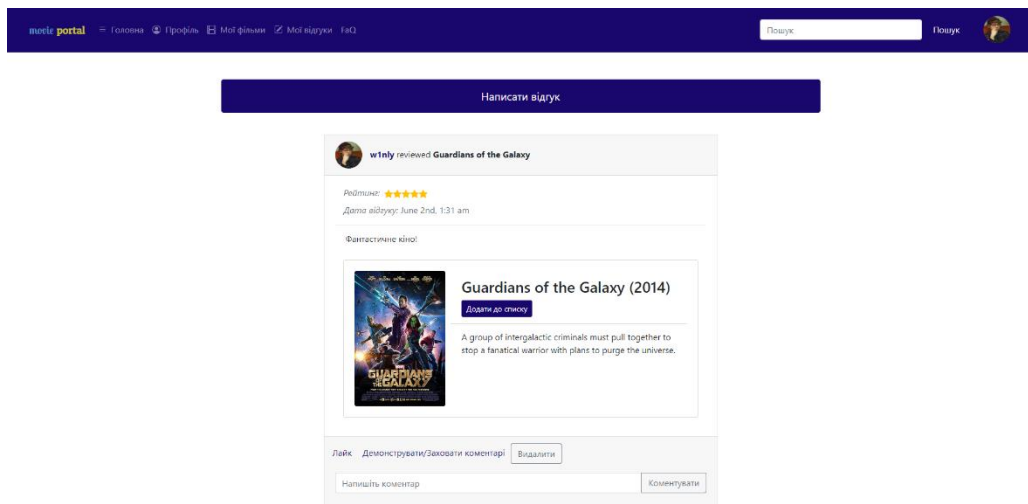


Рис. 1.30. Сторінка «Мої відгуки»

FaQ

Сторінка FaQ містить невеликий опис мого сайту та карусель react-bootstrap, що містить деякі з моїх улюблених фільмів з улюбленими акторами та актрисами.

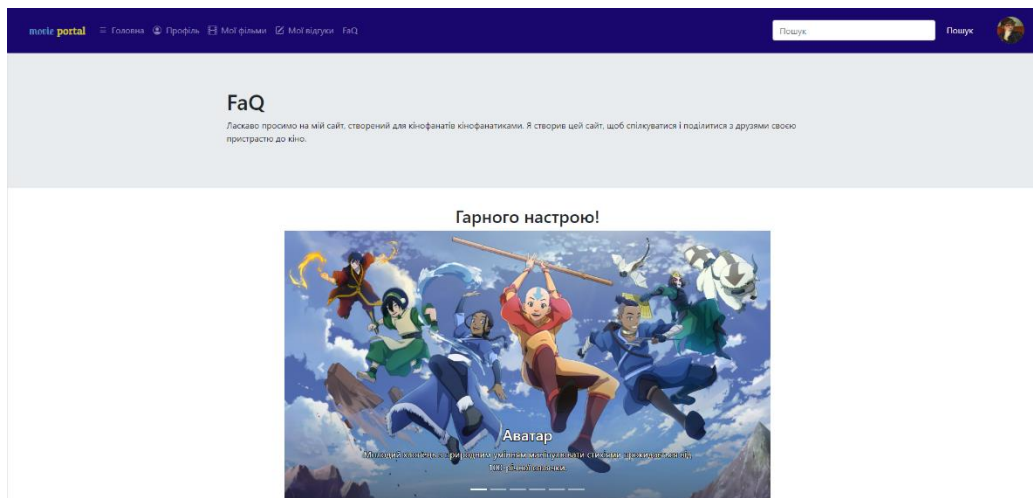


Рис. 1.31. Сторінка «FaQ»

3.5. Висновки до розділу 3

У даному розділі дипломної роботи розроблено проектування та реалізацію кінопорталу з допомогою визначених технологічних компонентів.

Було визначено доцільність використання окремого файлу програмних залежностей для забезпечення безпеки та комфорту кінопорталу.

Було визначено конструкційні особливості бази даних та розроблено структуру бази даних та кожного її компонента. Базу даних було інтегровано до проекту і визначено готовність роботи бази даних в проекті.

Структуру серверного рівня було розділено на декілька частин і приведено механізми їх роботи завдяки діаграмам послідовності. Було сконцтруйовано програмний код запуску серверного рівня.

Під час реалізації проекту було створено дизайн і визначено основні аспекти роботи веб-сторінок веб-рівня кінопорталу з майже повними визначеними функціональними можливостями.

ВИСНОВКИ

Дипломна робота була присвячена глибокому аналізу предметної області та розробці кінопорталу з великим обсягом функціональних та соціальних можливостей користувача. Було проаналізовано доцільність створення такого кінопорталу в Україні та прикладах популярності інших гігантів в предметному середовищі. На базі аналізу предметної області, було визначено базовий функціонал веб-сервісу, а також було проаналізовано популярні технологічні бек-енд, фронт-енд рішення та бази даних для створення веб-сервісу.

Було розглянуто доволі популярне рішення розробки на базі JavaScript стек MERN, аналіз якого показав доцільність використання всіх його компонентів в одній гнучкій структурі, яка дуже продуктивно пов'язується між собою. Кожен компонент було проаналізовано і визначено переваги їх використання як в сукупності з іншими так і окремо по одному. Підкреслено, що даний стек потребує інтеграції додаткових компонентів для реалізації захисних заходів, а також було проаналізовано інтеграцію інших програмних залежностей для використання при розробці. Було також визначено доцільність саме OMDb API в проекті, та проаналізовано використання саме VS Code.

При реалізації програмної частини кінопорталу було вирішено створити окремий файл програмних залежностей з метою покращення безпеки та комфорту. Розроблено детальну схему бази даних, включаючи структуру та опис кожного її компоненту. Базу даних успішно інтегровано до проекту, і тепер вона готова до повноцінної роботи. Архітектура серверного рівня розділена на декілька модулів, взаємодія яких чітко описана за допомогою діаграм послідовності. Написано програмний код для запуску серверної частини. Для веб-рівня кінопорталу розроблено дизайн та визначено ключові аспекти роботи веб-сторінок. Функціональні можливості сайту практично повністю реалізовані.

Загалом, отримані результати та розроблені рекомендації, будуть корисними в подальших розробках веб-сервісів та кінопорталів загалом.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Flanagan David. JavaScript: The Definitive Guide / David Flanagan. – O'Reilly Media, 2020. 704 с.
2. Freeman Adam, Clements David Mark. Node.js Design Patterns / Adam Freeman, David Mark Clements. – Packt Publishing, 2016. 520 с.
3. Grider Stephen. Node with React: Full Stack Web Development / Stephen Grider. – Udemу, 2020. 121 с.
4. Hallett Eric. React Native in Action / Eric Hallett. – Manning Publications, 2019. 375 с.
5. MongoDB Documentation. URL: <https://docs.mongodb.com/> (дата звернення: 15.04.2024).
6. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 15.04.2024).
7. Express.js Documentation. Express.js. URL: <https://expressjs.com/> (дата звернення: 20.04.2024).
8. React.js Documentation. React.js. URL: <https://reactjs.org/> (дата звернення: 20.04.2024).
9. Angular Documentation. Angular. URL: <https://angular.io/> (дата звернення: 20.04.2024).
10. Comparing JavaScript Frameworks: Angular vs React vs Vue. DigitalOcean, 2021. URL: <https://www.digitalocean.com/community/tutorials/comparing-javascript-frameworks-angular-react-and-vue> (дата звернення: 25.04.2024).
11. Top 10 Node.js Frameworks for Web App Development in 2021. AltexSoft, 2021. URL: <https://www.altexsoft.com/blog/engineering/top-10-node-js-frameworks-for-web-app-development-in-2021/> (дата звернення: 25.04.2024).
12. Офіційний веб-сайт IMDb. URL: <https://www.imdb.com/> (дата звернення: 15.04.2024).