

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА САЙТУ «АВТОРСЬКІ РЕЦЕНЗІЇ ФІЛЬМІВ
ТА СЕРІАЛІВ» НА БАЗІ ANGULAR, ASP.NET CORE ТА
ENTITY FRAMEWORK**

**DEVELOPMENT OF THE SITE «AUTHO'R REVIEWS OF
FILMS AND SERIES» BASED ON ANGULAR, ASP.NET CORE
TA ENTITY FRAMEWORK**

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-41
Маркович Анатолій Сергійович


(підпис)

Керівник:
к.т.н., доцент
Ящук Андрій Анатолійович


(підпис)

Кваліфікаційну роботу
допущено до захисту

« 8 » 06 2024 р.

к.т.н., доцент

Гарант освітньої програми:

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНО ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.В. М. Микуча
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Маркович Анатолій Сергійович

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: *Розробка сайту «Авторські рецензії фільмів та серіалів» на базі Angular, ASP.NET Core та Entity Framework*

Керівник роботи: *к.т.н., доцент, Яциук Андрій Анатолійович*

затверджені наказом закладу вищої освіти від «30» грудня 2023 р. № 477/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «5» червня 2024 р.

3. Вихідні дані до роботи: *технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: *4 рисунки, 4 лістинги.*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Яцук А. А.		
Специфікації вимог до розробленої системи	Яцук А. А.		
Розробка об'єкта проєктування	Яцук А. А.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		8,0%	
Академічна доброчесність	Яцук А. А.		

7. Дата видачі завдання «2» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	07.02.2024 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	вик.
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкту проєктування та проєктування бази даних	17.04.2024 р.	вик.
5	Практична реалізація об'єкту проєктування та розробка бази даних	18.05.2024 р.	вик.
6	Тестування та налагодження об'єкту проєктування	01.06.2024 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	05.06.2024 р.	вик.

Здобувач вищої освіти

(підпис)

Керівник кваліфікаційної роботи

(підпис)

Маркович А. С.
(прізвище, ініціали)

Яцук А. А.
(прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти: *Маркович Анатолій Сергійович*

Тема: *Розробка сайту «Авторські рецензії до фільмів та серіалів» на базі Angular, ASP.Net Core та Entity Framework».*

Коротка характеристика кваліфікаційної роботи та прийнятих рішень:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проаналізовані проблематика та поставлені завдання за темою роботи, здійснено теоретичне дослідження та практичну реалізацію сайту «Авторські рецензії до фільмів та серіалів» на базі Angular, ASP.Net Core та Entity Framework», здійснено тестування розробленої системи.

Самостійні розробки і пропозиції автора: *Автор самостійно сайт «Авторські рецензії до фільмів та серіалів» на базі Angular, ASP.Net Core та Entity Framework.*

Практичне значення роботи *полягає в тому, що розроблено онлайн-ресурс, що дозволяє розміщувати переглядати авторські рецензії до фільмів та серіалів. Це допоможе користувачам підібрати цікаві фільми для перегляду.*

Недоліки: *Істотних недоліків в роботі не виявлено.*

Загальний висновок: *У кваліфікаційній роботі бакалавра здійснено аналіз предметної області, вибір методів і засобів для розробки, розроблено програмний продукт, здійснено його тестування. Усі поставлені завдання виконано в повному обсязі.*

Кваліфікаційна робота здобувача освіти Марковича Анатолія Сергійовича рекомендується до захисту екзаменаційній комісії та заслуговує позитивної оцінки.

Рецензент кваліфікаційної роботи: Саварин Павло Вікторович, к.п.н., доцент кафедри ЦОТ


(підпис)

« 07 » червня 2024 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач вищої освіти Маркович Анатолій Сергійович
(прізвище, ім'я, по батькові)

група ПЗ-41

Тема кваліфікаційної роботи: РОЗРОБКА САЙТУ «АВТОРСЬКІ РЕЦЕНЗІЇ ФІЛЬМІВ ТА СЕРІАЛІВ» НА БАЗІ ANGULAR, ASP.NET CORE ТА ENTITY FRAMEWORK.

Керівник к.т.н., доц. Ящук А. А.

Актуальність теми. В умовах сучасного цифрового світу, де інтернет став основним джерелом інформації, платформи для обміну думками та рецензіями на культурні продукти, зокрема фільми та серіали, набувають все більшої популярності. Розробка веб-сайту для авторських рецензій на базі сучасних технологій, таких як Angular, ASP.NET Core та Entity Framework, є актуальною проблемою

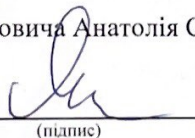
Об'єкт дослідження – створення інтерактивного веб-додатку, що забезпечує користувачам зручний доступ до інформації про фільми та серіали, а також можливість залишати власні рецензії та оцінки.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи. Було проведено аналіз предметної області, визначено вимоги до програмного продукту, здійснено розробку сайту «Авторські рецензії фільмів та серіалів» на базі Angular, asp.net core та Entity framework. Здійснено тестування системи.

Загальний висновок

Робота Марковича Анатолія Сергійовича може бути оцінена на позитивну оцінку.

Керівник


(підпис)

Ящук А. А.

(прізвище, ім'я, по батькові)

«07» 06 2024 р.

АНОТАЦІЯ

Маркович А. С. Розробка сайту «Авторські рецензії фільмів та серіалів» на базі Angular, ASP.NET core та Entity Framework. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки комп'ютерних ігор і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено повноцінний сайт. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки демонструють можливості фреймворку Angular.

Ключові слова: Angular, ASP.NET Core, Entity Framework, html, CSS.

ANNOTATION

Markovych A. S. Development of the Site "Author's Reviews of Films and Series" Based on Angular, asp.net Core and Entity Framework. Manuscript.

Bachelor's Qualification Work of the OP "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualification work consists of an introduction, three chapters, conclusions and suggestions, a list of references, and appendices. The first chapter analyzes the current state of the problem of developing computer games and formulates the task. The second chapter defines the requirements for the developed system, as well as the tools, methods, and algorithms for development. The third chapter develops a fully functional website. The conclusions summarize the information presented in the previous sections. The development results demonstrate the capabilities of the Angular framework.

Keywords: Angular, ASP.NET Core, Entity Framework, html, CSS

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ САЙТУ НА БАЗІ ANGULAR, ASP.NET CORE ТА ENTITY FRAMEWORK ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ.....	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	12
Висновки до розділу 1.....	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання.....	15
Висновки до розділу 2.....	17
РОЗДІЛ 3 РОЗРОБКА САЙТУ «АВТОРСЬКІ РЕЦЕНЗІЇ ФІЛЬМІВ ТА СЕРЕАЛІВ».....	19
3.1 Практична реалізація об'єкта проектування.....	19
3.2 Реалізація маршрутизації в проекті Angular	24
3.3 Тестування об'єкта проектування	26
Висновки до розділу 3.....	28
ВИСНОВКИ	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	32
ДОДАТКИ	34

ВСТУП

Актуальність теми. В умовах сучасного цифрового світу, де інтернет став основним джерелом інформації, платформи для обміну думками та рецензіями на культурні продукти, зокрема фільми та серіали, набувають все більшої популярності. Розробка веб-сайту для авторських рецензій на базі сучасних технологій, таких як Angular [1], ASP.NET Core та Entity Framework [2], є надзвичайно актуальною з кількох причин:

- потреба в якісному контенті;
- інтеграція та масштабованість;
- розвиток веб-технологій;
- зростання інтересу до власного контенту.

В цьому проєкті був використаний Angular версії 17.3.4. Головні причини в виборі саме цього фреймворку, саме цієї версії були такі: покращення продуктивності, інструментів розробки, керування станом та підвищена безпека.

Ці удосконалення роблять Angular ще більш потужним та зручним фреймворком для розробки сучасних веб-додатків, забезпечуючи високий рівень ефективності та надійності.

Мета роботи: створення зручного та практичного джерела для отримання інформації та рецензій.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- здійснити аналіз сучасного стану проблеми розробки сайтів на базі Angular, asp.net core та Entity Framework;
- ідентифікувати та провести аналіз вимог до розроблюваного програмного забезпечення та його проектування;
- вибрати інструменти та методи для розробки вебсайту;
- реалізувати сайт на базі Angular, asp.net core та Entity Framework у відповідності до теми на практиці;
- реалізувати маршрутизацію на сайті;

– здійснити тестування розробленого сайту.

Об’єктом роботи є створення інтерактивного веб-додатку, що забезпечує користувачам зручний доступ до інформації про фільми та серіали, а також можливість залишати власні рецензії та оцінки. Дослідження включає аналіз та реалізацію інтерфейсу користувача, який повинен бути інтуїтивно зрозумілим і адаптивним для різних пристроїв, таких як настільні комп’ютери, планшети та смартфони. Важливою складовою є функціональність системи, яка охоплює реєстрацію та аутентифікацію користувачів, публікацію рецензій, коментування та оцінювання фільмів і серіалів.

Особлива увага приділяється інтеграції з базою даних для ефективного зберігання та обробки інформації, а також забезпеченню безпеки даних користувачів. Розробка передбачає використання сучасних технологій, зокрема Angular, для створення динамічного та масштабованого додатку, що здатний витримувати високі навантаження і забезпечувати високу продуктивність. Таким чином, об’єкт дослідження включає як технічні аспекти розробки, так і питання забезпечення якісного користувацького досвіду.

Предметом роботи в розробці сайту про авторські рецензії фільмів та серіалів на Angular є технічні та функціональні аспекти створення веб-додатку, який забезпечує зручний доступ до контенту, можливості взаємодії користувачів з рецензіями та забезпечення безпеки даних.

Дослідження охоплює вивчення архітектурних підходів та методик розробки на базі Angular, зокрема, реалізацію маршрутизації для забезпечення навігації між сторінками, механізмів двосторонньої прив’язки даних для динамічного оновлення контенту, і застосування стилів для створення привабливого та інтуїтивно зрозумілого інтерфейсу.

Практична цінність розробки полягає у створенні ефективного веб-додатку, який дозволяє користувачам зручно знаходити, читати та публікувати рецензії на фільми та серіали. Веб-додаток на базі Angular забезпечує швидку і зручну навігацію, що покращує користувацький досвід. Використання сучасних

технологій і методик розробки сприяє високій продуктивності та масштабованості системи, що дозволяє обслуговувати велику кількість користувачів одночасно. Інтеграція з базою даних дозволяє зберігати і обробляти великий обсяг інформації про фільми, серіали та рецензії, забезпечуючи їхню доступність у реальному часі.

Розробка також враховує аспекти безпеки, що забезпечує захист даних користувачів від несанкціонованого доступу та інших загроз. Завдяки адаптивному дизайну веб-додаток коректно відображається на різних пристроях, забезпечуючи зручний доступ до контенту незалежно від платформи. Таким чином, практична цінність розробки полягає у створенні сучасного, функціонального та безпечного ресурсу для шанувальників кіно та серіалів, який покращує їхню взаємодію з контентом і надає можливість для активної участі у спільноті кіноманів.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ САЙТУ НА БАЗІ ANGULAR, ASP.NET CORE ТА ENTITY FRAMEWORK ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Аналіз сучасного стану проблеми включає в себе два аспекти: стан ринку онлайн-рецензій та технічні вимоги та виклики.

Сучасний стан ринку онлайн-рецензій характеризується високою насиченістю та інтенсивною конкуренцією. Інтернет простір вже містить безліч платформ, що пропонують рецензії на фільми та серіали, серед яких найпопулярніші – IMDb [3], Rotten Tomatoes [4], Metacritic [5], а також численні блоги та YouTube-канали [6]. Це створює значні виклики для нових гравців, які прагнуть привернути увагу користувачів та виділитися серед конкурентів.

Окрім рецензій, багато існуючих платформ пропонують широкий спектр додаткового контенту, такого як трейлери, новини індустрії, інтерв'ю з акторами та режисерами. Це дозволяє їм приваблювати різноманітну аудиторію та утримувати її інтерес. Для нових сайтів важливо мати унікальну пропозицію або додаткові функції, які б відрізняли їх від вже існуючих ресурсів.

Якість контенту стає критичним фактором успіху. Користувачі все більше цінують професійні, добре написані та аргументовані рецензії, які пропонують глибокий аналіз і обґрунтовані думки. Це підвищує вимоги до авторів та редакторів, які повинні мати високу кваліфікацію та експертизу в кінематографі.

Технічні вимоги та виклики у розробці сайту «Авторські рецензії фільмів та серіалів» є численними та комплексними, враховуючи сучасні стандарти та очікування користувачів. Однією з ключових вимог є персоналізація та використання рекомендаційних систем. Сучасні користувачі очікують персоналізованого досвіду, коли контент та рекомендації адаптуються під їхні інтереси та вподобання. Це вимагає розробки ефективних алгоритмів машинного

навчання та рекомендаційних систем, які можуть аналізувати поведінку користувачів, їхні уподобання та історію переглядів, щоб пропонувати релевантний контент.

Важливо також забезпечити мобільну оптимізацію сайту, оскільки все більше користувачів використовують мобільні пристрої для доступу до контенту. Це означає, що сайт повинен бути адаптивним та швидким на будь-якому пристрої, незалежно від його розміру екрану чи операційної системи. Використання Angular дозволяє створювати адаптивні інтерфейси, однак, це потребує додаткових зусиль для оптимізації швидкості завантаження та інтерактивності на мобільних пристроях.

SEO [7] та видимість у пошукових системах є ще однією критичною технічною вимогою. Для успішного залучення аудиторії необхідно, щоб сайт був оптимізований для пошукових систем, таких як Google. Це включає технічну оптимізацію структури сайту, метатеги, URL-адреси, а також створення якісного контенту, який відповідає запитам користувачів. Стратегія побудови посилань також є важливим аспектом, який допоможе підвищити авторитет сайту в очах пошукових систем.

Використання технологій Angular, ASP.NET Core та Entity Framework надає потужний інструментарій для розробки високоякісних веб-додатків, але також приносить певні виклики. Angular забезпечує створення динамічних, інтерактивних користувацьких інтерфейсів, дозволяючи створювати швидкі та зручні для користувачів SPA (Single Page Application). Проте, для забезпечення належного користувацького досвіду потрібні значні навички у фронтенд розробці та оптимізації продуктивності.

ASP.NET Core є високопродуктивною та крос-платформною платформою для створення бекенд-систем, що забезпечує легку інтеграцію з різними базами даних, високу безпеку та масштабованість. Це робить ASP.NET Core відмінним вибором для створення масштабованих веб-додатків, однак, вимагає досвіду у бекенд розробці та розуміння архітектурних патернів. Entity Framework, як ORM

інструмент, спрощує роботу з базами даних, дозволяючи розробникам працювати з даними у вигляді об'єктів, що підвищує продуктивність розробки та знижує кількість помилок. Однак, ефективне використання Entity Framework вимагає розуміння концепцій ORM та знання специфіки роботи з реляційними базами даних.

Забезпечення продуктивності системи стає критичним завданням, особливо при збільшенні кількості користувачів та обсягу даних. Важливо забезпечити належну оптимізацію коду та архітектури для підтримки високої продуктивності та швидкого часу відгуку. Це включає використання кешування, оптимізацію запитів до бази даних та використання інструментів моніторингу продуктивності. Безпека також є ключовим аспектом розробки, що включає захист від SQL-ін'єкцій [8], XSS-атак, забезпечення безпеки автентифікації та авторизації користувачів.

Після запуску сайту необхідно забезпечити його регулярну підтримку, оновлення та масштабування у разі зростання кількості користувачів. Це вимагає добре спланованої архітектури та стратегії розвитку, що включає використання хмарних сервісів, автоматизацію процесів розгортання та моніторинг системи в режимі реального часу. Успіх проекту значною мірою залежить від здатності команди ефективно впоратися з цими технічними викликами, забезпечити безпеку, продуктивність та належну підтримку сайту після його запуску.

Стає очевидним, що розробка сайту «Авторські рецензії фільмів та серіалів» на базі Angular, ASP.NET Core та Entity Framework має значний потенціал для успіху, але вимагає ретельного планування та врахування численних технічних викликів. Висока конкуренція на ринку онлайн-рецензій підкреслює необхідність створення унікального та якісного контенту, а також використання передових технологій для забезпечення персоналізованого користувацького досвіду. Розробка ефективних рекомендаційних алгоритмів та оптимізація сайту для мобільних пристроїв є критично важливими аспектами. Окрім цього, успішна реалізація проекту залежить від здатності команди

забезпечити високий рівень продуктивності та безпеки системи, а також регулярну підтримку та масштабування у разі зростання кількості користувачів. Загалом, при належному виконанні всіх технічних вимог та врахуванні поточних тенденцій ринку, проект має всі шанси стати важливим гравцем у сфері онлайн-рецензій.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

В результаті аналізу сучасного стану проблеми було визначено наступні завдання на кваліфікаційну роботу бакалавра:

- здійснити аналіз сучасного стану проблеми розробки сайтів на базі Angular, asp.net core та Entity Framework;
- ідентифікувати та провести аналіз вимог до розроблюваного програмного забезпечення та його проектування;
- вибрати інструменти та методи для розробки вебсайту;
- реалізувати сайт на базі Angular, asp.net core та Entity Framework у відповідності до теми на практиці;
- реалізувати маршрутизацію на сайті;
- здійснити тестування розробленого сайту.

Висновки до розділу 1

Було здійснено аналіз сучасного стану ринку онлайн-рецензій, який показав високу насиченість та конкуренцію серед платформ, що пропонують огляди на фільми та серіали. Вивчено ключові технічні вимоги та виклики, зокрема персоналізацію контенту, мобільну оптимізацію, SEO та використання передових технологій таких як Angular, ASP.NET Core та Entity Framework. Виявлено, що успішна реалізація проекту залежить від здатності команди забезпечити якісний контент, високу продуктивність, безпеку та підтримку сайту

після його запуску. Підкреслено важливість створення унікальної пропозиції та додаткових функцій для залучення аудиторії в умовах жорсткої конкуренції. Таким чином, здійснений аналіз окреслив ключові напрями для подальшої розробки сайту, що дозволить забезпечити його конкурентоспроможність та привабливість для користувачів.

Крім того, було визначено, що інтеграція функцій соціальної взаємодії, таких як коментування та обговорення рецензій, а також система рекомендацій на основі вподобань користувачів, може суттєво підвищити залученість аудиторії. Впровадження таких функцій, як можливість створення персональних списків улюблених фільмів і серіалів, а також підтримка різних мовних версій сайту, сприятиме розширенню цільової аудиторії та збільшенню часу, проведеного на сайті.

Таким чином, комплексний підхід до розробки, що включає детальний аналіз ринку, технічні нововведення та орієнтацію на потреби користувачів, дозволить створити ефективний та конкурентоспроможний продукт, який задовольнить очікування кіноманів та критиків.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Аналіз вимог до розроблюваного програмного забезпечення для проекту «Авторські рецензії фільмів та серіалів» розпочинається з визначення мети проекту, його основних функцій та користувачів. Основною метою проекту є створення веб-платформи, яка дозволяє користувачам публікувати, переглядати, коментувати та оцінювати рецензії на фільми та серіали. Зацікавленими сторонами виступають адміністратори, автори рецензій, звичайні користувачі та відвідувачі сайту.

Функціональні вимоги включають реалізацію реєстрації та автентифікації користувачів, з можливістю підтвердження електронної пошти та відновлення пароля. Профіль користувача має містити можливість редагування особистих даних, таких як ім'я, прізвище, аватар та біографія. Основний функціонал для авторів передбачає створення, редагування та видалення рецензій, з можливістю вказання назви, опису, тексту рецензії, категорії (фільм або серіал), постера і рейтингу. Для звичайних користувачів передбачено можливість перегляду рецензій, їх коментування та оцінювання.

Нефункціональні вимоги охоплюють продуктивність, безпеку, масштабованість, юзабіліті та сумісність системи. Система повинна підтримувати до 1000 одночасних користувачів без значних втрат у продуктивності, а час завантаження сторінок не повинен перевищувати 3 секунд. Безпека передбачає захист від SQL – ін'єкцій, XSS та CSRF атак, а також шифрування паролів користувачів. Масштабованість забезпечить можливість горизонтального та вертикального розширення системи. Юзабіліті має забезпечити інтуїтивно зрозумілий інтерфейс з підтримкою різних мов, а сумісність – коректне відображення на різних браузерах та пристроях.

Проектування програмного забезпечення передбачає використання багаторівневої архітектури, де фронтенд реалізується на Angular, бекенд на ASP.NET Core, а для роботи з базою даних використовується Entity Framework Core з SQL Server [10]. Фронтенд відповідає за взаємодію з користувачем, бекенд – за обробку бізнес-логіки та API, база даних – за зберігання даних. Архітектура передбачає чіткий розподіл завдань між компонентами, що забезпечує модульність та легкість у підтримці та розширенні системи.

Процес розробки включає планування та аналіз вимог, проектування архітектури системи, розробку фронтенду та бекенду, інтеграцію компонентів, тестування системи, її розгортання та подальшу підтримку. Тестування передбачає юніт-тести, інтеграційне, функціональне та навантажувальне тестування, а також перевірку безпеки.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для вирішення завдання розробки веб-сайту «Авторські рецензії фільмів та серіалів» необхідно обрати відповідні засоби, методи та алгоритми, що забезпечать ефективну розробку, підтримку та масштабованість системи.

Фронтенд буде реалізовано за допомогою Angular, що є сучасним фреймворком для створення динамічних веб-додатків. Angular надає можливість модульної розробки, двостороннього зв'язку даних та створення компонентів, що значно спрощує розробку та підтримку інтерфейсу користувача. Використання TypeScript, який є надмножиною JavaScript з додатковою статичною типізацією, забезпечить вищу продуктивність та знизить кількість помилок під час розробки. HTML5 та CSS3 [11] будуть використовуватися для створення сучасної розмітки та стилізації веб-сторінок, забезпечуючи адаптивний дизайн для різних пристроїв.

Бекенд буде реалізовано на базі ASP.NET Core, що є крос-платформним високопродуктивним фреймворком для створення веб-додатків. ASP.NET Core забезпечує модульність, високу продуктивність та можливість масштабування. Мова програмування C# [12] забезпечить безпеку та легкість у підтримці коду.

Для зберігання даних буде використовуватись SQL Server, реляційна система управління базами даних, яка забезпечує високу продуктивність, безпеку та надійність. Entity Framework Core, об'єктно-реляційний маппер (ORM) для .NET, забезпечить взаємодію з базою даних через об'єктно-орієнтовану модель, спрощуючи доступ до даних та їх обробку.

Для керування кодовою базою та співпраці між розробниками буде використовуватися система контролю версій Git [13]. Хмарні платформи, такі як Azure [14] або AWS [15], забезпечать розгортання, масштабування та управління веб-додатками, що дозволить швидко адаптуватися до змін у навантаженні.

Використання Agile методології, зокрема Scrum або Kanban, забезпечить гнучкість у розробці, регулярні ітерації та часті випуски функціоналу. Це дозволить адаптуватися до змін у вимогах та зменшити ризики проекту. Тестування буде здійснюватись за допомогою TDD (Test-Driven Development), що передбачає написання тестів перед реалізацією функціоналу. Це забезпечить високу якість коду та легкість у його підтримці. Юніт-тести перевірятимуть окремі модулі/компоненти системи, інтеграційні тести — взаємодію між компонентами, а функціональні тести — виконання функціональних вимог.

Для авторизації та автентифікації буде використовуватись JWT (JSON Web Token), що дозволяє створювати токени, які можуть бути використані для автентифікації запитів до API. REST API буде використовуватись для обробки HTTP-запитів, забезпечуючи зрозумілу структуру URL-адрес, використання стандартних методів HTTP (GET, POST, PUT, DELETE) та передачу даних у форматі JSON.

Для оптимізації продуктивності буде застосовуватись кешування. In-Memory Cache забезпечить тимчасове зберігання частих запитів та зменшить

навантаження на базу даних, а Distributed Cache (наприклад, Redis) забезпечить швидкий доступ до даних у розподіленому середовищі. Реалізація алгоритмів сортування (наприклад, QuickSort, MergeSort) та фільтрації забезпечить швидку та ефективну обробку списків рецензій за різними критеріями. Для обробки даних буде використовуватись LINQ [16] (Language Integrated Query), що дозволить зручно та ефективно здійснювати запити до даних у базі даних через Entity Framework Core.

Таким чином, обрані засоби, методи та алгоритми забезпечать надійну та ефективну розробку системи «Авторські рецензії фільмів та серіалів». Використання сучасних технологій та методів розробки дозволить досягти високої продуктивності, безпеки та гнучкості системи, що сприятиме її успішній реалізації та підтримці.

Висновки до розділу 2

було ретельно розглянуто вибір технологічних засобів для реалізації веб-сайту «Авторські рецензії фільмів та серіалів». Було обрано Angular для фронтенду та ASP.NET Core для бекенду, оскільки ці інструменти забезпечують високу продуктивність та гнучкість у розробці. Для зберігання даних використовується SQL Server, що забезпечує надійність та швидкодію, а Entity Framework Core спрощує доступ до бази даних через об'єктно-орієнтовану модель.

З метою забезпечення безпеки системи використовується JWT для автентифікації користувачів та кешування для оптимізації продуктивності. Впровадження Agile методології сприяє гнучкій розробці, а TDD забезпечує високу якість коду і відповідність функціональним вимогам проекту.

Обрані технології та методики розробки спрямовані на створення сучасного та ефективного веб-додатку, який надасть користувачам зручний

доступ до рецензій на фільми та серіали, а також забезпечить гнучку підтримку та масштабування системи у майбутньому.

Цей веб-додаток базується на фреймворку Angular, що забезпечує швидкодіючий інтерфейс та ефективне управління даними завдяки використанню односторонньої та двосторонньої прив'язки даних. Маршрутизація в Angular дозволяє логічно організувати навігацію по сторінках, забезпечуючи зручний перехід між розділами сайту.

Для стилізації і оформлення веб-додатку використовується CSS разом із компонентною архітектурою Angular, що дозволяє створювати чистий і легко підтримуваний код. Це сприяє створенню сучасного дизайну та візуально привабливого вигляду сторінок з рецензіями.

Під час розробки були застосовані сучасні підходи до управління станом додатку, що робить його більш ефективним та масштабовним. Використання Angular дозволяє легко впроваджувати нові функції та компоненти, що робить систему гнучкою для майбутнього розширення.

Отже, обрані технології та методики розробки сприятимуть створенню веб-додатку, який забезпечить користувачам зручний доступ до рецензій на фільми та серіали, а також зможе ефективно розвиватись і масштабуватись у майбутньому.

РОЗДІЛ 3

РОЗРОБКА САЙТУ «АВТОРСЬКІ РЕЦЕНЗІЇ ФІЛЬМІВ ТА СЕРЕАЛІВ»

3.1 Практична реалізація об'єкта проектування

Першим кроком в розробці було створення робочої області та початкову програму. Після цього запуск початкової програми для перевірки працездатності.

Далі оформлення початкової сторінки, яка дає розуміння про проект (рис. 3.1).

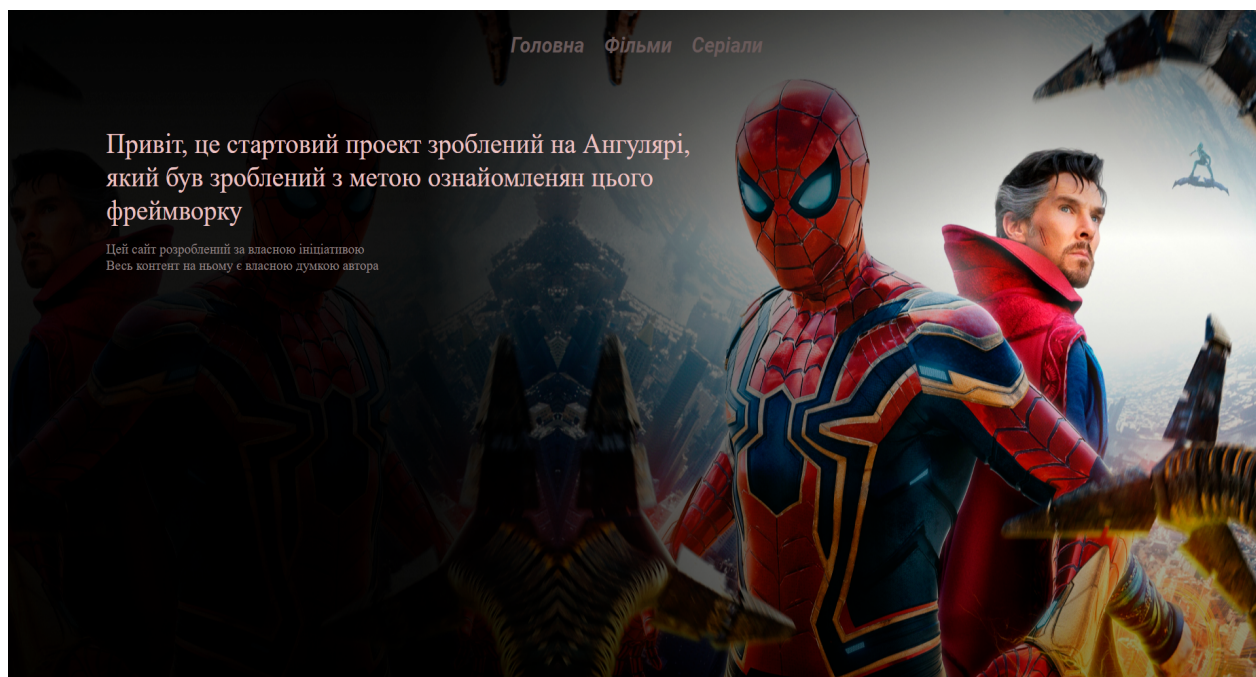


Рисунок 3.1 – Вигляд початкової сторінки

Початкова сторінка складається з двох компонентів: header та home.

Завдяки, тому що цей проект є SPA (single page application) це дає змогу відображати кожний компонент на екрані користувача динамічно. Оскільки header представляє з себе статичний компонент навігації то він відображається при переході на будь-яку сторінку.

Сторінки сайту налаштовані на динамічну заміну завдяки налаштованій маршрутизації, яка здійснюється завдяки TypeScript. Це все можливе завдяки дерективі Router Outlet, яка надається модулем Angular Router. Він діє як вікно перегляду в макеті програми, де маршрутизовані компоненти відображаються динамічно. Це дозволяє додаткам Angular мати кілька переглядів або сторінок без необхідності перезавантаження повної сторінки.

Наступним кроком була розробка сторінки фільмів, в додатку А наведено html код сторінки з фільмами (рис. 3.2).

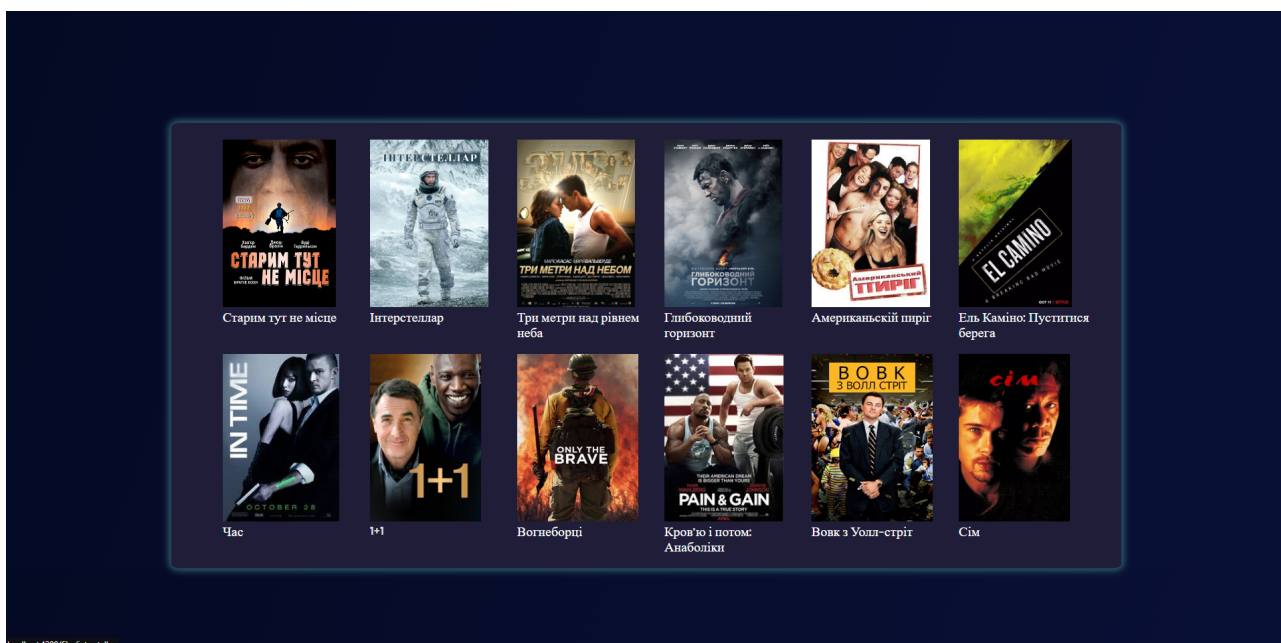


Рисунок 3.2 – Вигляд сторінки з фільмами

Контент після навігаційної панелі відображає окремий компонент `film.component`. В його розробці було використано one-way Data Binding (одностороннє прив'язування даних) для анімації використовувався селектор стилів «`hover`».

Одностороннє зв'язування даних дозволяє користувачам Інтернету керувати елементами веб-сторінки за допомогою веб-браузера. Він містить динамічний HTML і не вимагає складного програмування. Зв'язування даних використовується у веб-додатках, які містять інтерактивні компоненти, такі як

форми, калькулятори, навчальні посібники та ігри. Поступове відображення веб-сторінки робить зв'язування даних зручним, коли сторінки містять великий обсяг даних.

Angular використовує концепцію двостороннього зв'язування. Будь-які зміни, пов'язані з елементами інтерфейсу користувача, відображаються у відповідному конкретному стані моделі. І навпаки, будь-які зміни стану моделі відображаються в стані інтерфейсу користувача. Це гарантує, що структура зможе підключити DOM до даних моделі за допомогою контролера.

CSS псевдоклас «:hover» спрацьовує, коли користувач наводить на елемент мишею, але не обов'язково активує його. Цей стиль може перевизначатися іншими псевдокласами, що відносяться до посилань, такими як «:link», «:visited» і «:active», що з'являються у відповідному порядку. Щоб стилізувати посилання належним чином, вставте правило «:hover» після правил «:link» і «:visited» до :active, як визначено LVHA-порядком: «:link» - «:visited» - «:hover» - «:active».

В компоненті film.component а саме файлі film.component.ts було використане одностороннє прив'язування даних для передогляду зображень фільмів. Зроблена прив'язка даних для джерела. В лістингу 3.1 показано прив'язку даних компоненту film.component.

Лістинг 3.1 – Демонстрація підключення шрифтів

```
export class FilmsComponent {
  no_country='https://planetakino.ua/res/get-
poster/000000000000000000000000000000003936/vend1.jpg'
  inter='https://s3.vcdn.biz/static/f/6755876391/image.jpg'
  three='https://upload.wikimedia.org/wikipedia/uk/5/5b/%D0%A2%D1%80%
D0%B8_%D0%BC%D0%B5%D1%82%D1%80%D0%B8_%D0%BD%D0%B0%D0%B4_%D1%80%D1%96%D0%B
2%D0%BD%D0%B5%D0%BC_%D0%BD%D0%B5%D0%B1%D0%B0.png'
  deepwater_horizon='https://upload.wikimedia.org/wikipedia/uk/c/c0/%
D0%93%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D0%BD%D0%B8%D
0%B9_%D0%B3%D0%BE%D1%80%D0%B8%D0%B7%D0%BE%D0%BD%D1%82.jpg'
```

```

    american='https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQqjjfv8gCbAJB0Z51jdmjxsI54zltRKgHucQ&
s'

    el_camino='https://upload.wikimedia.org/wikipedia/uk/thumb/4/4e/El_
camino_bb_film_poster.jpg/225px-El_camino_bb_film_poster.jpg'

    time='https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQ_Z14PV0cqFoila1IwuxlLPiAYhabxbfBd1y1
beuWF7ksZ_IfqLsqQwqpZ9ycLuYNBUKQ&usqp=CAU'

    one='https://static.hdrezka.ac/i/2024/1/17/z60772d98bc31ey18b52p.jp
eg'

    fireman='https://uaserials.pro/posters/3893.jpg'

    pain='https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcSCijWAQHMHpIQ_c26SAhqPPG9CI3QUPC3eEw&
s'

    wolf='https://uaserials.pro/posters/2068.jpg'

    seven='https://uaseial.club/images/serials/64/64148ee6897850835724
63.webp'

    }

```

Кінець лістингу 3.1

Були підключені шрифти Google Fonts [17], а саме:

- Roboto Bold Italic;
- Jersey 20.

Для підключення шрифтів в Angular був обраний спосіб завантаження них та зберігання їх в кореневій папці ассетів. Підключення відбувалось в головному файлі стилю проекту Angular завдяки функції стилів «@font-face», завдяки цьому шрифти доступні для використання в будь якому файлі стилю проекту. В лістингу 3.2 наведено приклад підключення шрифтів.

Лістинг 3.2 – Демонстрація підключення шрифтів

```
@font-face{
```

```

font-family: 'myfont';
src: url('assets/fonts/Jersey20-Regular.ttf');
}
@font-face {
font-family: 'titlefont';
src: url('assets/fonts/Roboto-BoldItalic.ttf');
}

```

Кінець лістингу 3.2

Сторінку кожного фільму відображали окремі компоненти відповідно. Для зручності та можливості розширення в майбутньому кількості сторінок без зайвих проблем було вирішено створити та використовувати окремий файл стилів для кожної сторінки з фільмами. На будь якій сторінці розташування та стильові властивості елементів були ідентичні, змінювався тільки контент. Завдяки цьому на створення нової сторінки для контенту витрачалось мінімум часу (рис. 3.3).

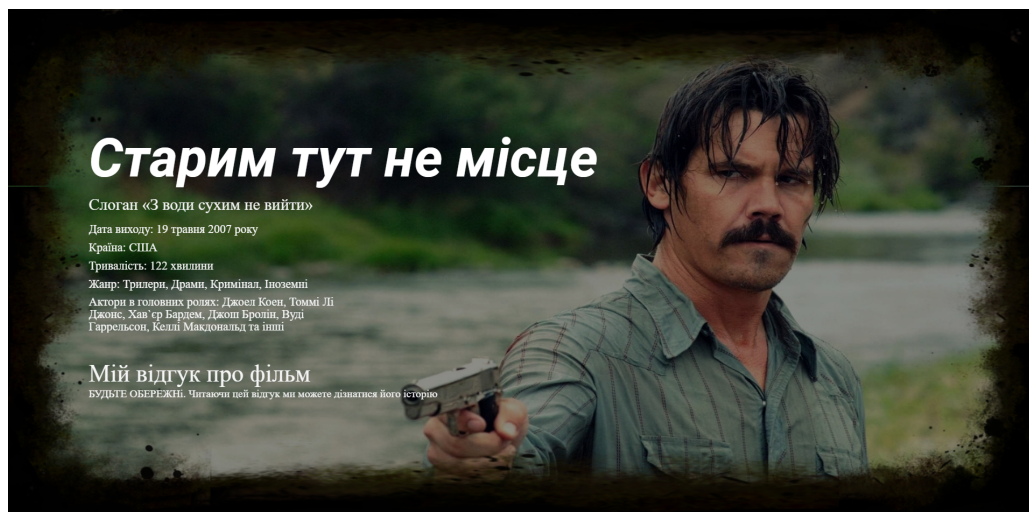


Рисунок 3.3 – Приклад сторінки рецензії

На сторінках рецензій були використані підключенні шрифти, привязка даних та анімації за допомогою стилів.

Прив'язка даних була виконано для всієї інформації про контент крім рецензій, в лістингу 3.3 наведено прив'язку даних.

Лістинг 3.3 – Демонстрація привязки даних

```
export class NoCountryForOldMenComponent {  
  backgroundimg="./assets/img/films/no_country.png"  
  title='Старим тут не місце'  
  slogan='З води сухим не вийти'  
  info = {  
    date: '19 травня 2007 року',  
    country: 'США',  
    duration: '122 хвилини',  
    genre: 'Трилери, Драми, Кримінал, Іноземні',  
    actor: 'Джоел Кoen, Томмі Лі Джонс, Хав`єр Бардем, Джош Бrolін,  
Вуді Гаррельсон, Келлі Макдональд та інші'}  
}
```

Кінець лістингу 3.3

Це дозволяє записувати дані в TypeScript файлі залежного компоненту та залишати «чистішим» файл розмітки сторінки.

3.2 Реалізація маршрутизації в проекті Angular

Маршрутизація в Angular є важливою частиною створення односторінкових додатків (SPA), дозволяючи переходити між різними компонентами, не перезавантажуючи сторінку. Angular використовує бібліотеку `@angular/router` для реалізації маршрутизації. Основними поняттями маршрутизації є роутер, маршрут та роутинг-модуль. Роутер (Router) є головним сервісом для маршрутизації, який відповідає за визначення поточного маршруту і вирішення його за допомогою відповідного компонента. Маршрут (Route) є об'єктом, що визначає шлях (URL) і відповідний компонент для відображення.

Роутинг-модуль (RouterModule) є модулем, який надає сервіси маршрутизації та директиви для Angular додатків.

Дочірні маршрути в Angular дозволяють організувати вкладену структуру компонентів, забезпечуючи гнучкі та логічно структуровані шляхи в односторінковому додатку. Дочірні маршрути використовуються для визначення підмаршрутів всередині основного маршруту, що дозволяє відображати компоненти всередині інших компонентів.

Щоб забезпечити навігацію між дочірніми маршрутами, потрібно використовувати директиву routerLink у шаблонах компонентів. Приклад шаблону для filmComponent. У лістингу 3.4 показано налаштування дочірніх маршрутів.

Лістинг 3.4 – Приклад дочірніх маршрутів

```
{path: 'film', children: [
    {path: 'no-country', title: 'Старим тут не
місце', component: NoCountryForOldMenComponent},
    {path: 'interstellar', title: 'Інтерстеллар', component: IterstellarComponent}
    ,
    {path: 'sem', title: 'СІМ', component: SevenmovieComponent},
    {path: 'tri-metra-nadurovnem-neba',
title: 'Три метри над рівнем неба',
component: ThreemetrComponent},
    {path: 'glubokovodnyy-gorizont',
title: 'Глибоководний горизонт',
component: GorizontComponent},
    {path: 'american-pie',
title: 'Американський пиріг', component: AmericanpieComponent},
    {path: 'el-camino',
title: 'Ель Каміно: Пуститися',
component: ElcaminoComponent},
    {path: 'chas', title: 'Час',
```

```

component:TimemovieComponent},
    {path:'one-plus-one',title:'1+1',
component:OneplusoneComponent},
    {path:'fire-man',title:'Вогнеборці',component:FiremovieComponent}
  ]}

```

Кінець лістингу 3.4

Щоб відобразити дочірні маршрути, необхідно додати додатковий `router-outlet` у шаблоні основного компонента, як показано вище. Це дозволяє вкладеним маршрутам відображатися всередині основного компонента, забезпечуючи вкладену навігацію.

Таким чином, дочірні маршрути в Angular дозволяють організувати складні структури компонентів і маршрутизації, забезпечуючи чітку і логічну навігацію в односторінкових додатках.

3.3 Тестування об'єкту проектування

Тестування є невід'ємною частиною процесу розробки веб-додатків, забезпечуючи високу якість коду і надійність програмного забезпечення. В Angular для тестування застосовуються різні інструменти, серед яких «ng test» є одним із основних.

«ng test» – це команда Angular CLI, яка запускає модульні тести, написані за допомогою Jasmine [18] і виконувані за допомогою Karma. Jasmine є бібліотекою для написання тестів, тоді як Karma є тестовим раннером, який дозволяє запускати тести в різних браузерах.

Основні кроки тестування в Angular за допомогою «ng test»:

- під час створення нового компонента, сервісу або модуля за допомогою Angular CLI, автоматично генерується файл тесту з розширенням `.spec.ts`. Ці

файли містять початкові приклади тестів, які можна розширювати відповідно до потреб;

- тести пишуться за допомогою Jasmine. Основні елементи Jasmine включають «describe» (описує групу тестів), «it» (описує конкретний тест), «beforeEach» (запускає код перед кожним тестом) і «expect» (виконує перевірку умов);

- для запуску тестів використовується команда «ng test», яка запускає Karma і відкриває браузер для виконання тестів. Результати тестування відображаються в консолі;

- після виконання тестів, Karma надає звіт про успішність тестів. У разі виявлення помилок або невідповідностей, розробники можуть швидко ідентифікувати проблеми і внести необхідні виправлення.

Використання «ng test» у процесі розробки Angular-додатків сприяє забезпеченню високої якості коду, підвищенню надійності та зменшенню часу на виправлення помилок. Це є важливим кроком до створення стабільних і ефективних веб-додатків.

Переваги використання «ng test»:

- команда «ng test» автоматизує процес запуску тестів, що робить його швидким та зручним;

- Karma автоматично перезапускає тести після внесення змін у код, що забезпечує швидкий зворотний зв'язок;

- регулярне тестування допомагає виявляти помилки на ранніх стадіях розробки, що сприяє підвищенню якості коду.

Тестування розробленого сайту пройшло успішно. На рисунку 3.4 показано результати тесту.

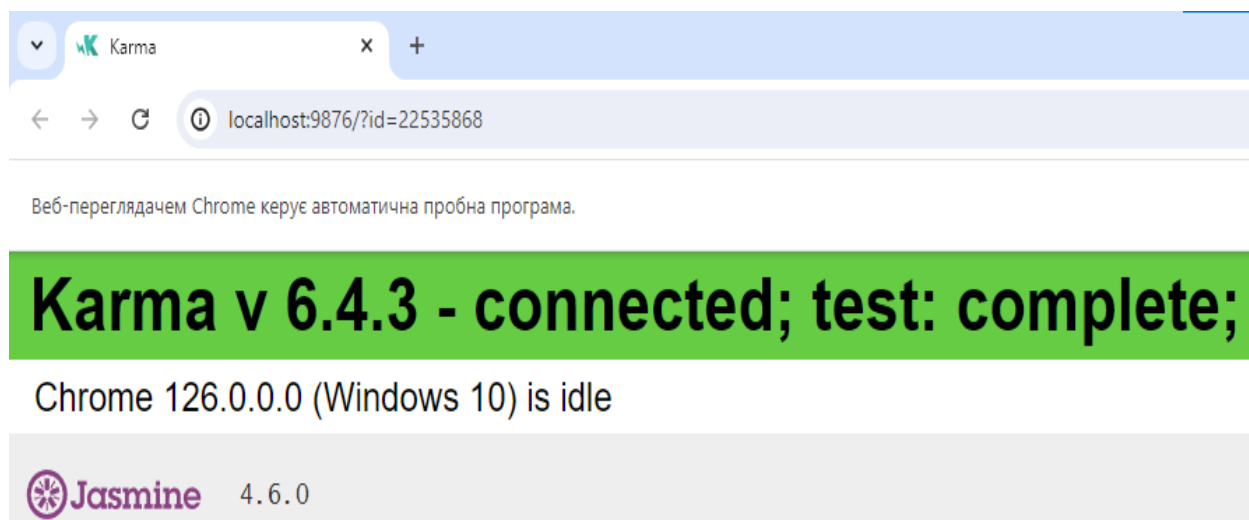


Рисунок 3.4 – результати модульного тесту

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи був розроблений повноцінний веб-сайт на основі фреймворку Angular. Цей сайт «Авторські рецензії на фільми та серіали» був створений з метою забезпечення зручного інтерфейсу для перегляду та взаємодії з рецензіями.

Маршрутизація в Angular була використана для створення плавного переходу між сторінками сайту. Були визначені різні маршрути для різних розділів, таких як домашня сторінка, сторінки з рецензіями та окремі сторінки рецензій. Використання маршрутизації сприяло не лише ефективній навігації між сторінками, але й організації загальної структури сайту.

Прив'язка даних відіграла ключову роль у динамічному оновленні вмісту, що відображався на веб-сайті. За допомогою односторонньої та двосторонньої прив'язки даних сайт мав можливість миттєво оновлювати вміст рецензій та взаємодію користувачів без необхідності оновлення сторінок. Це значно підвищило реактивність та інтерактивність веб-сайту, забезпечуючи більш залучене взаємодію користувачів.

Стилізація веб-сайту була реалізована за допомогою CSS та компонентного підходу до стилів в Angular. CSS використовувався для визначення загальної композиції, колірних схем та типографіки, що забезпечило візуально привабливе відображення контенту. Компонентна стилізація Angular дозволила створювати стилі, які були специфічні для кожного компонента, зберігаючи при цьому їхню перевикористовуваність та легкість в підтримці.

Отже, розроблений у третьому розділі веб-сайт «Авторські рецензії на фільми та серіали» на базі Angular демонструє успішне використання сучасних підходів до розробки веб-додатків, що сприяло покращенню його функціональності та користувацького досвіду.

ВИСНОВКИ

Розробка сайту на тему авторських рецензій на фільми та серіали на базі сучасних технологій, таких як Angular, ASP.NET Core та Entity Framework, є обґрунтованим і перспективним рішенням, яке відповідає сучасним вимогам та тенденціям в області веб-розробки. Використання цих технологій дозволяє створити високоякісний, швидкий, масштабований та безпечний веб-додаток, який забезпечує зручність користувачів та ефективну роботу з контентом.

Здійснено аналіз сучасного стану проблеми розробки сайтів на базі Angular, asp.net core та Entity Framework, який виявив конкурентний ринок онлайн-платформ, що пропонують рецензії на фільми та серіали, підкреслюючи насиченість та інтенсивну конкуренцію.

Аналіз вимог включав визначення мети проекту, його основних функцій та зацікавлених сторін. Він окреслював як функціональні, так і нефункціональні вимоги, включаючи продуктивність, безпеку, масштабованість, юзабіліті та сумісність. Процес проектування використовував багаторівневу архітектуру з Angular для фронтенду, ASP.NET Core для бекенду та Entity Framework Core з SQL Server для управління базами даних, забезпечуючи чітке розподілення завдань між компонентами для модульності та легкості обслуговування.

Вибрано інструменти та методи для розробки вебсайту. Для фронтенду було обрано Angular, а для бекенду – ASP.NET Core. SQL Server використовувався для зберігання даних через його надійність та швидкість, а Entity Framework Core використовувався для спрощення обробки баз даних.

Реалізовано сайт на базі Angular, asp.net core та Entity Framework. Практична реалізація проекту почалася з налаштування робочого середовища і початкової настройки програми. Розробка включала створення динамічного користувацького інтерфейсу з використанням маршрутизації та прив'язки даних Angular, забезпечуючи відгуковий та зручний досвід користувача на різних пристроях.

Реалізовано маршрутизацію на сайті. Для обробки навігації в межах сайту використовувався Angular Router, що дозволяє динамічно відображати вміст без перезавантаження сторінки. Ця частина реалізації підкреслювала природу SPA додатка, що покращує користувацький досвід, роблячи навігацію безшовною та інтуїтивно зрозумілою.

Було проведено всебічне тестування для забезпечення відповідності функціональності та продуктивності сайту визначеним вимогам. Це включало юніт-тести, інтеграційні тести та тести продуктивності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація Angular та TS. URL: <https://angular.io/docs> (дата звернення: 07.04.2024).
2. Документація ASP.NET Core та Entity Framework. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата звернення 10.04.2024).
3. IMDb. URL: <https://www.imdb.com> (дата звернення: 15.04.2024).
4. Rotten Tomatoes. URL: <https://www.rottentomatoes.com> (дата звернення: 25.04.2024).
5. Metacritic. URL: <https://www.metacritic.com> (дата звернення 15.05.2024).
6. YouTube. URL: <https://www.youtube.com> (дата звернення 15.05.2024).
7. SEO. URL: https://en.wikipedia.org/wiki/Search_engine_optimization (дата звернення 15.05.2024).
8. SQL-ін'єкції. URL: <https://uk.wikipedia.org/wiki/SQL-ін%27єкція> (дата звернення 15.05.2024).
9. SQL Server. URL: <https://www.microsoft.com/uk-ua/sql-server> (дата звернення 17.05.2024).
10. Документація HTML5 та CSS3. URL: <https://html-css.co.ua> (дата звернення 18.05.2024).
11. Документація C#. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp> (дата звернення 19.05.2024).
12. Git. URL: <https://git-scm.com> (дата звернення 20.05.2024).
13. Azure. URL: <https://azure.microsoft.com/en-us> (дата звернення 20.05.2024).
14. AWS. URL: <https://aws.amazon.com> (дата звернення 20.05.2024).
15. LINQ. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/linq> (дата звернення 15.05.2024).

16. Google Fonts. URL: <https://fonts.google.com> (дата звернення 15.05.2024).

17. Документація Jasmine. URL: <https://jasmine.github.io> (дата звернення 15.05.2024).

ДОДАТКИ

Додаток А

Код html елемента компоненту film.component

```

<html>
  <body>
    <div class="container">
      <div class="item">
        <a routerLink="no-country">
          
        </a>
        <div class="name">Старим тут не місце</div>
      </div>
      <div class="item">
        <a routerLink="interstellar">
          
        </a>
        <div class="name">Інтерстеллар</div>
      </div>
      <div class="item">
        <a routerLink="tri-metra-nadurovnm-neba">
          
        </a>
        <div class="name">Три метри над рівнем неба</div>
      </div>
      <div class="item">
        <a routerLink="glubokovodnyy-gorizont">
          
        </a>
        <div class="name">Глибоководний горизонт</div>
      </div>
      <div class="item">
        <a routerLink="american-pie">
          

```

```

        </a>
        <div class="name">Американський пиріг</div>
    </div>
    <div class="item">
        <a routerLink="el-camino">
            
        </a>
        <div class="name">Ель Каміно: Пуститися берега</div>
    </div>
    <div class="item">
        <a routerLink="time">
            
        </a>
        <div class="name">Час</div>
    </div>
    <div class="item">
        <a routerLink="one-plus-one">
            
        </a>
        <div class="name">1+1</div>
    </div>
    <div class="item">
        <a routerLink="fire-man">
            
        </a>
        <div class="name">Вогнеборці</div>
    </div>
    <div class="item">
        <a routerLink="no_country">
            
        </a>
        <div class="name">Кров'ю і потом: Анаболіки</div>
    </div>

```

```

    <div class="item">
      <a routerLink="no_country">
        
      </a>
      <div class="name">Бовк з Уолл-стріт</div>
    </div>
    <div class="item">
      <a routerLink="no_country">
        
      </a>
      <div class="name">Сім</div>
    </div>
  </div>
</body>
</html>

```

Код css элемента компоненту film.component

```

body{
  background: rgb(5,10,36) !important;
  background: linear-gradient(108deg, rgba(5,10,36,1) 0%,
  rgba(9,16,53,1) 82%) !important;
}

.container{
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  background-color: rgb(31, 29, 56);
  margin: 100px 200px 0px 250px;
  height: fit-content;
  width: fit-content;
  padding: 25px 0px 20px 4%;
}

```

```
border-radius: 1%;  
box-shadow: 0px 0px 10px 4px rgb(48, 97, 112);  
}
```

```
.item{  
  width: 200px;  
  height: 250px;  
  padding-bottom: 50px;  
}
```

```
img {max-width:100%;height:100%;}
```

```
.name{  
  color: rgb(255, 255, 255);  
  font-size: 20px;  
  font-family: myfont;  
}
```