


Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)


Кафедра Медіасистем та технологій  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**


рівень вищої освіти перший (бакалаврський)  
Створення SPA «Фільмотека»  
з застосуванням сучасного JS фреймворку та API  
(тема)

Виконав.  
студент 4 курсу, групи ВПВПС-20-1  
  
Волобуєв А.О.  
(прізвище, ініціали)

Спеціальність 186 Видавництво та поліграфія  
(код і повна назва спеціальності)  
Тип програми освітньо-професійна  
Освітня програма  
Видавничо-поліграфічна справа  
(повна назва освітньої програми)

Керівник   
доц. Козуб П.А.  
(посада, прізвище, ініціали)

Допускається до захисту  
Зав. кафедри МСТ

  
(підпис)

Дейнеко Ж.В.  
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
Кафедра \_\_\_\_\_ Медіасистем та технологій \_\_\_\_\_  
Рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 186 Видавництво та поліграфія \_\_\_\_\_  
Тип програми \_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
Освітня програма \_\_\_\_\_ Видавничо-поліграфічна справа \_\_\_\_\_  
(шифр і назва)

ЗАТВЕРДЖУЮ.  
Зав. кафедри МСТ \_\_\_\_\_  
(підпис)  
« 20 » травня 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові \_\_\_\_\_ *Волобуєву Артему Олексійовичу* \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ *Створення SPA «Фільмотека»* \_\_\_\_\_  
\_\_\_\_\_ *з застосуванням сучасного JS фреймворку та API* \_\_\_\_\_

Затверджена наказом по університету від \_\_\_\_\_ 20 травня 2024 р. № 558 Ст \_\_\_\_\_


2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 26 червня 2024 р. \_\_\_\_\_

3. Вихідні дані до роботи  
*Вид і призначення: web-видання – web-сайт з інформацією про фільми; мови розробки web-видання: JavaScript, HTML, стилі: SCSS, Vue.js – Frontend фреймворк*

4. Перелік питань, що потрібно опрацювати в роботі  
*Аналіз завдання на кваліфікаційну роботу, визначення цілей і задач проектування; Аналітичний огляд досягнень у виробництві та застосуванні WEB-видань; Послідовність виготовлення WEB-видання; Вибір інструментальних засобів розробки; Проектування інформаційної структури та навігації; Розробка модульної сітки; Розробка графічного дизайну; Наповнення контентом сторінок видання; Тестування і публікація; Економічна частина; Висновки*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)  
*Титульний слайд, Вступ, Вихідні данні до проекту, Етапи створення електронного видання, Вибір програмного забезпечення, Робота над макетом, Робота над сайтом, Економічна частина; Висновки.*

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Козуб П.А.		26.06.2024
Економічна частина	ас. Помогалова Н.В.		26.06.2024

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз завдання на кваліфікаційну роботу	23.05.2024	виконано
2	Аналітичний огляд досягнень у виробництві та застосуванні Web-видань	25.05.2024	виконано
3	Виготовлення веб-видання	26.05.2024	виконано
4	Вибір інструментальних засобів розробки	28.05.2024	виконано
5	Проектування інформаційної структури та навігації	31.05.2024	виконано
6	Розробка модульної сітки	31.05.2024	виконано
7	Розробка графічного дизайну	31.05.2024	виконано
8	Наповнення контентом сторінок видання	12.06.2024	виконано
9	Тестування і публікація	12.06.2024	виконано
10	Економічна частина	14.06.2024	виконано
11	Оформлення пояснювальної записки	18.06.2024	виконано
12	Оформлення графічної частини	18.06.2024	виконано

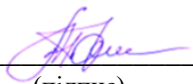
Дата видачі завдання: 20.05.2024 р.

Студент

  
\_\_\_\_\_  
(підпис)

Волобуєв А. О.

Керівник роботи

  
\_\_\_\_\_  
(підпис)

доц. Козуб П.А.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 50 с., 2 табл., 22 рис., 2 дод., 23 джерела.

ЕЛЕКТРОННЕ ВИДАННЯ, МОДУЛЬНА СІТКА, WEB-САЙТ, ВЕРСТКА САЙТУ, НАВІГАЦІЯ ВИДАННЯ, ЮЗАБІЛІТІ, ТЕХНІЧНІ ХАРАКТЕРИСТИКИ.

У представленому проєкті була створена онлайн-платформа "Фільмотека", що дозволяє користувачам знаходити та зберігати інформацію про фільми. Реалізовано можливість перегляду списку фільмів з детальною інформацією, такою як назва, рік випуску, жанр та рейтинг. Користувачі можуть використовувати функцію пошуку для зручного відбору фільмів за певними критеріями. Додатково впроваджено можливість додавати обрані фільми до особистої бібліотеки для майбутнього доступу. Крім того, користувачі можуть оцінювати фільми та залишати коментарі, створюючи активну спільноту обговорення фільмів.

Для зручності надано можливість перегляду трейлерів та інших мультимедійних матеріалів для кожного фільму, якщо вони доступні. Проєкт "Фільмотека" має на меті забезпечити користувачам зручний інструмент для знаходження та зберігання інформації про фільми.

## ABSTRACT

The explanatory note contains 50 p., 2 tabl., 22 fig., 2 app., 23 sources.

ELECTRONIC EDITION, MODULAR NETWORK, WEB SITE, SITE LAYOUT, PUBLICATION NAVIGATION, USABILITY, TECHNICAL CHARACTERISTICS.

As part of the project, an online platform called Filmoteka was created that allows users to find and store information about movies. It is possible to view a list of movies with detailed information such as title, year of release, genre, and rating.

Users can use the search function to conveniently select movies by specific criteria. Additionally, the ability to add selected movies to a personal library for future access has been introduced. In addition, users can rate movies and leave comments, creating an active community of movie discussion. For convenience, users can view trailers and other multimedia materials for each movie, if available. The Filmoteka project aims to provide users with a convenient tool for finding and storing information about movies, as well as to create a favorable environment for exchanging opinions and recommendations on watching movies.

## ЗМІСТ

	С.
ВСТУП.....	7
1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ. ВИЗНАЧЕННЯ МЕТИ ТА ЗАДАЧ ПРОЄКТУВАННЯ .....	10
2 АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ У ВИРОБНИЦТВІ ТА ЗАСТОСУВАННІ WEB-ВИДАНЬ.....	13
3 ПОСЛІДОВНІСТЬ ВИГОТОВЛЕННЯ WEB-ВИДАННЯ .....	16
4 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ .....	21
5 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ТА НАВІГАЦІЇ.....	30
6 РОЗРОБКА МОДУЛЬНОЇ СІТКИ .....	34
7 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ .....	36
8 НАПОВНЕННЯ КОНТЕНТОМ СТОРІНОК ВИДАННЯ .....	39
9 ТЕСТУВАННЯ І ПУБЛІКАЦІЯ .....	42
10 ЕКОНОМІЧНА ЧАСТИНА .....	44
ВИСНОВКИ .....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	49
ДОДАТОК А Приклади сторінок веб-сайту.....	51
ДОДАТОК Б Фрагменти коду для створення веб-сайту .....	52

## ВСТУП

У представленому проєкті було здійснено розробку онлайн-платформи під назвою «Фільмотека», яка пропонує користувачам унікальну можливість знаходити, переглядати та зберігати інформацію про різноманітні фільми. Цей проєкт націлений на створення інноваційного інструменту для кінолюбителів, що дозволяє не лише знаходити необхідні фільми, а й отримувати доступ до рейтингів, оглядів та іншої корисної інформації. З використанням сучасних технологій JS фреймворків та API, було створено інтерактивну та зручну в експлуатації платформу, що сприяє полегшенню пошуку та організації перегляду фільмів для кожного користувача. Було розглянуто деталі розробки проєкту, його функціональні можливості та перспективи в подальшому розвитку [1, 2].

Відповідно до теми індивідуального завдання «Створення SPA «Фільмотека» з застосуванням сучасного JS фреймворку та API», метою роботи є визначення інструментів для створення web-сайту, розробка модульної сітки, розробка дизайну, наповнення контентом та тестування сайту на кросбраузерність та кросплатформеність [2].

Для досягнення поставленої мети необхідно виконати кілька важливих етапів. Першим кроком є визначення інструментів для створення web-сайту. Це включає вибір відповідного JavaScript фреймворку, такого як React, Vue.js або Angular. Окрім цього, слід вибрати технології для бекенду та серверу, наприклад, Node.js та Express, а також визначити API для доступу до бази даних фільмів, як-от The Movie Database (TMDB) API.

Наступним етапом є розробка модульної сітки. Потрібно спланувати структуру компонентів SPA, визначити основні модулі, такі як пошук, перегляд фільмів та інформація про фільм. Також слід створити прототип, використовуючи модульну сітку, для візуалізації структури додатку.

Розробка дизайну є важливим кроком для створення привабливого та зручного інтерфейсу користувача. На цьому етапі визначається загальний стиль та тема веб-сайту, створюються макети сторінок, наприклад, домашньої сторінки та сторінки деталей фільму. Вибір кольорової схеми, типографії та інших елементів дизайну забезпечить естетичну привабливість сайту.

Після розробки дизайну необхідно наповнити сайт контентом. Це включає інтеграцію API для отримання даних про фільми, створення динамічного контенту на основі отриманих даних та перевірку актуальності контенту. Важливо, щоб дані були представлені користувачам у зручному та зрозумілому форматі.

Заключним етапом є тестування сайту на кросбраузерність та кросплатформеність. Це включає перевірку роботи веб-сайту в різних браузерах, таких як Chrome, Firefox, Safari та Edge, а також тестування на різних пристроях, таких як настільні комп'ютери, планшети та мобільні телефони. Важливо виправити всі виявлені баги та адаптувати сайт для різних платформ, щоб забезпечити безперебійну роботу додатку на всіх пристроях.

Таким чином, проєкт має на меті створення сучасного, зручного у використанні SPA-додатку для перегляду та пошуку інформації про фільми, який буде адаптований для різних браузерів та платформ, з використанням передових технологій та інструментів веб-розробки.

На першому етапі роботи виконується аналіз завдання для визначення вимог до проєкту та формулювання вихідних даних для проєктування.

У другому розділі проводиться аналітичний огляд досягнень у виробництві та застосуванні веб-видань.

Третій розділ містить виготовлення веб-видання.

Четвертий розділ аргументує вибір інструментів розробки для проєкту.

П'ятий розділ присвячений проєктуванню структури та навігації веб-сайту.

Шостий розділ розробка модульної сітки.

Сьомий розділ описує розробку графічного дизайну.

Восьмий розділ наповнення контентом сторінок видання.



Дев'ятий розділ описує процес тестування і публікації.

Десятий розділ містить економічні розрахунки проєкту.

Для досягнення основної мети цієї роботи необхідно було вирішити наступні завдання:

- аналіз літератури по обраній темі;
- аналіз існуючих аналогів серед схожих сайтів подібного типу;
- визначення цільової аудиторії;
- обґрунтування вибору необхідного програмного забезпечення;
- розробка концепції дизайну та структури сайту;
- розробка схеми навігації;
- наповнення контентом сторінок сайту, розробка посилань на відповідні ресурси;
- перевірка та тестування сайту на різних платформах та браузерах;
- налагодження функціональності сайту.

Кваліфікаційна робота складається з вступу, десяти розділів, висновків до кожного розділу, списку використаних джерел та додатків.

## 1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ. ВИЗНАЧЕННЯ МЕТИ ТА ЗАДАЧ ПРОЄКТУВАННЯ

В цьому розділі проведено детальний аналіз завдання на кваліфікаційну роботу з тематики «Створення SPA «Фільмотека» з використанням сучасного JS фреймворку та API». Розглянуті вихідні дані та вимоги стандартів, обґрунтовані параметри видання або продукції, мета та галузь застосування виконаної роботи. Також уточнено послідовність дій та визначено цілі роботи. Розділ містить важливу інформацію щодо того, яким чином планується досягти поставлених цілей та який результат очікується в кінці проєкту.

В розділі проведено детальний аналіз вихідних даних та вимог до роботи. Виявлено потребу в створенні спеціалізованої платформи для організації та перегляду фільмів в онлайн режимі, що відповідає сучасним стандартам і вимогам користувачів. Основна проблематика проєкту полягає в розробці інструменту, який забезпечить зручний пошук, організацію та перегляд фільмів для різних категорій користувачів. Основна мета роботи полягає в створенні онлайн-платформи «Фільмотека», яка забезпечить користувачам можливість знаходити, зберігати та переглядати інформацію про фільми за допомогою сучасних веб-технологій.

В широкому сенсі цільовою аудиторією проєкту є широкий спектр користувачів, включаючи кінолюбителів будь-якого віку та активних користувачів інтернету. Основні обмеження цільової аудиторії пов'язані з їхнім технічним рівнем та можливістю роботи з сучасними веб-інструментами.

У той же час даний проєкт передбачає реалізацію інструменту у вигляді конкретного веб-додатку під визначеного споживача. Для визначення потрібної функціональності та дизайну було розглянуто два основних веб-додатку для перегляду фільмів.

Netflix. Американський відеосервіс, що належить однойменній компанії зі Сполучених Штатів, яка була заснованою в 1997 році. Спочатку вона займалася

кінопрокатом, відправляючи диски з фільмами поштою. У 1999 році був запуск інтернет-сервісу для доступу до відео за запитом. З 2013 року Netflix запустив власне виробництво телесеріалів, які доступні тільки користувачам платформи. Особливість цього сервісу - серіали тут з'являються відразу цілим сезоном і доступні для перегляду в будь-який час після релізу. Доступ до сервісу платний і обмежений територією США і кількох країн світу. Тестовий період для безкоштовного користування послугою становить 30 днів.

Megogo.net. Онлайн-кінотеатр є одним із найбільш популярних на території України. Існують різні види платних тарифів. Послуга почала працювати в листопаді 2011 року. Щомісячна аудиторія послуги – 27 мільйонів чоловік. Цей інтернет-кінотеатр заробляє на відеорекламі. Реклама відображається глядачеві перед переглядом фільму, після нього та коли ви робите паузу. Ряд нових релізів доступні лише після оплати вартості вибраного фільму чи серіалу.

Divan.tv. Менш поширена українська служба доступу до Інтернет-телебачення та легального відеовмісту. Проект розпочався в червні 2011 року. Спочатку позиціонувався як телевізійна приставка, користувач може переглядати телевізійні канали, фільми та слухати аудіоконтент. На даний момент існує Інтернет-сервіс, який можна використовувати як з придбанням приставки, так і на «розумних» телевізорах, робочих столах, смартфонах і планшетах.

Головні переваги перегляду фільмів в Інтернеті – це, широкий вибір фільмів та відсутність залежності від часу показу фільму. Але при цьому такі інтернет кінозали мають багато суттєвих недоліків – складність інтерфейсу, велика кількість непотрібних для користувача каналів та фільмів, залежність від підключення до Інтернету.

Саме тому створення більш простого веб-додатку, з функціями орієнтованими лише на конкретного споживача, є одним із останніх трендів і це можливо при розробці легих сайтів за допомогою таких таких технічних засобів, як Vue 3 для розробки інтерфейсу користувача, Vue Router для маршрутизації,

Pinia для управління станом додатка, SCSS для написання стилів, VSCode для редакції коду та TheMovieDB API для отримання даних про фільми.

Основною метою роботи є створення веб-сайту з використанням сучасного JS фреймворку та API. З цієї метою визначено конкретні цілі, включаючи розробку інтерфейсу, інтеграцію функціональності та оптимізацію продуктивності. Завдання проектування включають вибір фреймворку, розробку архітектури, інтеграцію API, створення дизайну та проведення тестування. Планування та виконання кожного етапу є ключовим для успішної реалізації проєкту та досягнення його цілей.

## 2 АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ У ВИРОБНИЦТВІ ТА ЗАСТОСУВАННІ WEB-ВИДАНЬ

В контексті аналізу досягнень у виробництві та застосуванні WEB-видань важливо визначити рівень розвитку самого інтернет-видання як частини цифрової медіа-сфери. WEB-видання, або Інтернет-видання, визначається як видання, що публікується в Інтернеті. Однією з особливостей WEB-видань є їхня інтерактивна природа, що надає користувачам можливість не лише переглядати інформацію, але й взаємодіяти з нею, активно залучаючись у процеси моделювання та виробництва [1-3, 5].

Web-видання є складовою частиною мережі та пов'язані з поняттями Web-сайту і Web-сторінки. Web-сторінка – це електронна сторінка у відповідному форматі, тоді як Web-сайт – це колекція пов'язаних Web-сторінок, що об'єднані загальними ознаками або призначенням [4, 5]. Використання WEB-видань значно розширює можливості інформаційної взаємодії, дозволяючи широкому колу аудиторії отримувати доступ до актуальної та різноманітної інформації за допомогою онлайн-ресурсів.

Аналіз розвитку WEB-видань виявляє постійне зростання їхньої популярності та розширення функціональності. Інтерактивність та можливість персоналізації контенту стають ключовими факторами успіху у цифровому медіа-просторі. Застосування сучасних технологій та розробка динамічних інтерфейсів дозволяє створювати більш ефективні та привабливі для користувачів WEB-видання.

У виробництві та застосуванні WEB-видань важливо розглядати тенденції та інновації, що впливають на розвиток цього сегмента медіа-галузі, щоб забезпечити відповідність потребам і очікуванням сучасного користувача.

Відповідно до теми індивідуального завдання «Створення SPA «Фільмотека» з застосуванням сучасного JS фреймворку та API», метою роботи є розробка сучасного веб-додатку для перегляду та пошуку інформації

про фільми. Для досягнення цієї мети необхідно виконати кілька важливих етапів, які забезпечать створення функціонального, зручного та естетично привабливого продукту.

Одним із основних завдань розробника є визначення інструментів для створення веб-сайту. Це включає вибір відповідного JavaScript фреймворку, такого як React, Vue.js або Angular, який дозволяє створювати швидкі та інтерактивні односторінкові додатки (SPA). Окрім цього, необхідно вибрати технології для бекенду та серверу, наприклад, Node.js та Express, які забезпечать обробку запитів та управління базою даних [6]. Також слід визначити API для доступу до бази даних фільмів, як-от The Movie Database (TMDB) API, що забезпечить надання актуальної інформації про фільми.

Наступним етапом є розробка модульної сітки. Потрібно спланувати структуру компонентів SPA, визначити основні модулі, такі як пошук, перегляд фільмів та інформація про фільм. Модульна сітка дозволить створити гнучку та легко масштабовану архітектуру додатку, що забезпечить його розширення та підтримку у майбутньому [7, 8].

Розробка дизайну є важливим кроком для створення привабливого та зручного інтерфейсу користувача. На цьому етапі визначається загальний стиль та тема веб-сайту, створюються макети сторінок, наприклад, домашньої сторінки та сторінки деталей фільму. Вибір кольорової схеми, типографії та інших елементів дизайну забезпечить естетичну привабливість сайту та сприятиме покращенню користувацького досвіду.

Після розробки дизайну необхідно наповнити сайт контентом. Це включає інтеграцію API для отримання даних про фільми, створення динамічного контенту на основі отриманих даних та перевірку актуальності контенту. Важливо, щоб дані були представлені користувачам у зручному та зрозумілому форматі, що сприятиме зручному пошуку та перегляду фільмів.

Заключним етапом є тестування сайту на кросбраузерність та кросплатформеність. Це включає перевірку роботи веб-сайту в різних браузерах, таких як Chrome, Firefox, Safari та Edge, а також тестування на

різних пристроях, таких як настільні комп'ютери, планшети та мобільні телефони. Важливо виправити всі виявлені баги та адаптувати сайт для різних платформ, щоб забезпечити безперебійну роботу додатку на всіх пристроях.

Додатково до раніше згаданих аспектів сучасного веб-видання, варто звернути увагу на такі ключові пункти. Мобільна оптимізація є надзвичайно важливою, з огляду на поширення мобільних пристроїв, які стали основним засобом доступу до Інтернету. Веб-видання має бути повністю оптимізоване для мобільних пристроїв, що включає адаптивний дизайн, швидкість завантаження сторінок та зручний інтерфейс на екранах різних розмірів.

Персоналізація контенту є ще одним важливим аспектом. Сучасні веб-видання часто використовують алгоритми рекомендацій для адаптації контенту до індивідуальних потреб та інтересів користувачів, що дозволяє забезпечити більш персоналізований досвід для кожного користувача. Використання мультимедійних засобів також є важливим для сучасних вебвидань. Відео, аудіо, анімація та інші мультимедійні елементи роблять контент більш привабливим та цікавим для користувачів.

Соціальна інтеграція стала невід'ємною частиною сучасних веб-видань. Інтеграція з соціальними мережами дозволяє користувачам легко ділитися цікавим контентом, коментувати та взаємодіяти з іншими користувачами, що сприяє збільшенню популярності веб-ресурсу [9]. Безпека є одним з пріоритетів сучасних веб-видань. Забезпечення захисту особистої інформації користувачів та даних включає використання різноманітних технологій шифрування та захисту даних, що гарантує конфіденційність та безпеку користувачів.

Таким чином, проєкт має на меті створення сучасного, зручного у використанні SPA-додатку для перегляду та пошуку інформації про фільми, який буде адаптований для різних браузерів та платформ, з використанням передових технологій та інструментів веб-розробки, але у той же час орієнтованим на конкретного споживача.

### 3 ПОСЛІДОВНІСТЬ ВИГОТОВЛЕННЯ WEB-ВИДАННЯ

Послідовність виготовлення веб-видання згідно з вимогами та завданнями кваліфікаційної роботи передбачає визначення цілей та завдань проєкту, планування архітектури, вибору необхідних інструментальних засобів та проєктування графічного інтерфейсу. Далі розглядаються етапи розробки дизайн-макетів сторінок, верстання, програмування та наповнення контентом. Завершальним етапом є тестування та запуск веб-видання, що дозволяє перевірити його на відповідність вимогам та готовність до публікації.

Основною метою цього етапу є конкретизація завдань та цілей проєкту з метою чіткого визначення напрямків роботи та досягнення успішного результату. Згідно з поставленими вимогами, головною метою роботи є створення веб-сайту з використанням сучасного JS фреймворку та API. Для досягнення цієї мети необхідно визначити конкретні цілі, серед яких розробка інтерфейсу, інтеграція функціональності та оптимізація продуктивності. Завдання проєктування включають в себе вибір фреймворку, розробку архітектури, інтеграцію API, створення дизайну та проведення тестування. Важливо планувати та виконувати кожен етап проєктування з урахуванням його впливу на загальний успіх проєкту та досягнення поставлених цілей [9].

Планування архітектури проєкту включає в себе визначення структури сайту, компонентів та модулів, які будуть використані у проєкті. Планування архітектури передбачає розробку схеми навігації та взаємодії користувача з веб-сайтом. Правильне планування архітектури дозволить ефективно організувати роботу над проєктом та забезпечити його успішну реалізацію [10, 11].

Вибір інструментів базується на вимогах проєкту та специфікаціях функціональності, які потрібно реалізувати згідно з вимогами замовника. При виборі інструментів для реалізації проєкту "Фільмотека" були враховані наступні критерії. Одним з основних критеріїв була сучасність та популярність інструментів, що використовуються. Це означає вибір тих інструментів, які є



актуальними на ринку та мають широку підтримку серед спільноти розробників. Популярні інструменти часто мають великі ресурси для навчання, документацію та активні форуми, що значно спрощує процес розробки та усунення помилок.

Також важливим критерієм була функціональність інструментів. Вони повинні забезпечувати всі необхідні функції для реалізації запланованого функціоналу веб-сайту. Це включає можливість створення динамічних інтерфейсів, управління станом додатка, маршрутизацію між сторінками та інші ключові аспекти розробки сучасного веб-додатка [11, 12].

Продуктивність була ще одним критерієм, який враховувався при виборі інструментів. Важливо обирати такі інструменти, які дозволяють ефективно виконувати завдання та максимально оптимізувати швидкість розробки. Це означає, що обрані технології повинні бути швидкими, легкими та добре інтегруватися між собою, що дозволяє зменшити час на розробку та тестування.

Крім того, масштабованість інструментів була важливим фактором при виборі. Інструменти повинні бути придатні для масштабування проєкту у майбутньому, забезпечуючи його розвиток та розширення [8-12]. Це включає можливість легкого додавання нових функцій, підтримку великої кількості користувачів та даних, а також адаптацію до змін у вимогах проєкту.

Легкість використання також відіграла важливу роль при виборі інструментів. Обрані технології повинні бути зрозумілими та легкими для використання, що спрощує процес розробки та підтримки проєкту.

З урахуванням цих критеріїв були обрані такі інструменти. Перш за все, Vue.js був обраний як сучасний та популярний фреймворк JavaScript для розробки користувацького інтерфейсу. Vue.js забезпечує високу продуктивність, легкість у використанні та масштабованість, що робить його ідеальним вибором для проєкту «Фільмотека».

Для маршрутизації в додатках Vue.js була обрана бібліотека Vue Router, яка дозволяє ефективно керувати навігацією між сторінками. Vue Router забезпечує гнучкість та простоту у налаштуванні маршрутизації, що є важливим для створення інтуїтивно зрозумілих та зручних для користувачів додатків.

Pinia була обрана як офіційна бібліотека управління станом для додатків Vue.js. Вона допомагає ефективно керувати станом додатка, забезпечуючи централізоване зберігання та управління даними. Pinia забезпечує високу продуктивність та простоту у використанні, що дозволяє легко додавати та управляти станом у додатку.

SCSS був обраний як препроцесор CSS, який дозволяє використовувати змінні, міксини та інші корисні функції для покращення роботи з CSS. Використання SCSS дозволяє створювати більш організований та підтримуваний CSS-код, що сприяє покращенню продуктивності та зручності у роботі з стилями.

Ці інструменти забезпечують необхідність для створення динамічного та інтерактивного веб-додатка «Фільмотека» з використанням сучасних підходів у веб-розробці. Використання цих технологій дозволяє створити зручний, продуктивний та масштабований додаток, який відповідає всім сучасним вимогам та стандартам веб-розробки.

При створення графічного дизайну розглядалось декілька варіантів колірної схеми, але рішення приймалось виходячи з вимог замовника, тому для веб-додатку передбачається використання темної колірної схеми з акцентним зеленим кольором та шрифтом Inter для забезпечення зручності та естетичного вигляду. Весь дизайн буде спроектований з фокусом на простоту, зручність та чіткість, щоб забезпечити приємний та ефективний досвід використання веб-додатку для користувачів.

Для розробки дизайну макету сторінок веб-сайту використовувалася середовище, тому що Figma є інтерактивним інструментом для дизайну та прототипування, який дозволяє розробникам, дизайнерам та іншим учасникам команди працювати над проектом у реальному часі [14, 15].

Верстка веб-сайту була виконана у середовищі Visual Studio Code. Під час верстки було застосовано принцип адаптивного дизайну, що забезпечує оптимальне відображення контенту на різних пристроях залежно від їхніх розмірів та розширення екрану [16, 17].

Адаптивна верстка дозволяє сайту оптимізувати свій вигляд для пристроїв з різними характеристиками екрану, такими як монітори, планшети та смартфони. Це забезпечує комфортне користування сайтом незалежно від пристрою, на якому він відображається. Під час верстання використовувалися Vue 3 та SCSS для створення структури та стилізації веб-сайту відповідно. Vue 3 надає зручний та потужний фреймворк для створення інтерактивних компонентів, а SCSS дозволяє використовувати змінні, міксини та інші функції для ефективного написання CSS стилів [17, 18]. Побудова користувацького інтерфейсу та односторінкового додатку (SPA) використовується Vue.js (Vue) – прогресивний фреймворк JavaScript, заснований на парадигмі компонентів, що дозволило розбити веб-додаток на окремі компоненти, які можна повторно використовувати.

Локалізація текстів зберігалася у форматі JSON файлів для кожної мови. Це дозволяє зручно організувати та управляти перекладами, а також підлаштовувати інтерфейс додатку під мову користувача.

Для динамічного імпорту текстів з JSON файлів використовувалися вбудовані засоби JavaScript та Vue, що дозволило зручно та ефективно реалізувати локалізацію та забезпечити коректне відображення контенту на будь-якій мові [19].

Маршрутизація відповідає за навігацію в веб-додатку, визначаючи, який компонент відображається для кожного URL. У цьому випадку створено 4 роути за допомогою Vue Router [16, 17].

Головна сторінка (/). Цей маршрут відповідає за головну сторінку веб-додатку, де відображається загальний контент.

Сторінка «Мої фільми» (/my-films). Цей маршрут веде на сторінку «Мої фільми», де можна переглянути фільми, які додані до черги або переглянуті.

Підмаршрути. Використовуються дочірні маршрути для сторінки «Мої фільми» – «Черга» та «Переглянуті». Кожен з цих маршрутів має свій власний компонент, який відображає відповідний контент.

Управління станом відповідає за збереження та оновлення даних в додатку.

Для цього використовується localStorage для локалізації та збережених фільмів, а дані пошуку зберігаються в URL в квері параметрах. Це ефективний спосіб зберігати та отримувати дані, необхідні для відображення на веб-сторінці.

Наповнення контенту – це важливий етап розробки, який включає в себе додавання тексту, зображень, відео та інших медіафайлів.

Інформація про фільми. Додання детальної інформації про фільми, таку як назва, рік випуску, режисер, актори, жанр, опис тощо.

Зображення та постери. Включення зображень або постерів для кожного фільму для візуального представлення.

Користувальницькі відгуки. Можливість користувачів залишати враження та рейтинги для фільмів.

Пошук і фільтрація. Забезпечення можливості користувачам швидко знаходити фільми за різними критеріями, такими як жанр, рік випуску, режисер тощо.

Локалізація контенту. Забезпечення можливості переглядати інформацію про фільми або користуватися сайтом мовою, яка відповідає вимогам користувача.

Динамічне оновлення. Постійне оновлення контенту з додаванням нових фільмів та видаленням застарілої інформації.

Тестування та запуску веб-сайту перед його передачею замовнику є одним із головних етапів розробки і передбачає використання різноманітних інструментів для розробників (Devtools) та перевірку сумісності веб-сайту з різними браузерами (Chrome, Firefox, Safari, Edge тощо), а також його коректної роботи на різних операційних системах (Windows, macOS, Linux), запуск веб-сайту локально за допомогою інструментів для керування пакетами (у цьому випадку – yarn) та локального сервера розробки (vite), щоб перевірити, як сайт веде себе в реальному часі. Розгортання веб-сайту на хмарному хостингу (Firebase) для того, щоб зробити його доступним для користувачів у мережі Інтернет.

#### 4 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ

На етапі вибору інструментальних засобів для розробки веб-видання, необхідно враховувати не лише характер інформації, що буде розміщена, але й потреби цільової аудиторії. Обґрунтований вибір програмних засобів для створення текстового, графічного, аудіо- та відео-контенту є ключовим аспектом цього етапу роботи [9, 11].

Для створення дизайну макету, логотипу та іншої графіки можна використати різноманітні програми векторної і растрової графіки, які відповідатимуть потребам та можливостям проєкту.

Характеристики програмного забезпечення для обробки текстової інформації також варто ретельно обговорити. На цьому етапі важливо обрати програми, що найбільш відповідають вимогам проєкту і забезпечать ефективну обробку текстового матеріалу.

Створення веб-сторінок передбачає верстку макету, що був розроблений на попередніх етапах. Ефективна верстка має розділяти дизайн та вміст сторінок, використовуючи мову HTML для вмісту та CSS для дизайну. Важливо мати можливість спостерігати як початковий код, так і готову сторінку, щоб забезпечити якісний результат на цьому етапі [12].

Для створення сучасних веб-інтерфейсів використовуються різні інструменти, кожен з яких має свої переваги та особливості. Один з таких інструментів – Adobe XD, який є векторним графічним редактором, спеціально розробленим для дизайну і прототипування веб-інтерфейсів. Adobe XD має зручний інтерфейс та набір інструментів, що дозволяють швидко створювати професійні макети. Цей інструмент підтримує інтерактивність прототипів, що дає змогу краще уявити, як буде виглядати та функціонувати кінцевий продукт [13].

Ще одним популярним інструментом для дизайну макетів є Sketch. Цей інструмент широко використовується серед веб-дизайнерів завдяки своїй простоті та ефективності. Sketch пропонує простий інтерфейс та розширені

можливості для роботи з векторною графікою, що дозволяє створювати високоякісні та детальні макети. Інструмент також підтримує численні плагіни та інтеграції, що робить його ще більш гнучким та потужним.

Figma є ще одним важливим інструментом для дизайну макетів та спільної роботи над проектами. Це онлайн-інструмент, який надає можливість спільно працювати над проектами в реальному часі, що особливо корисно для команд, розподілених по різних локаціях. Figma має багатофункціональний інтерфейс для створення дизайну та підтримує всі основні функції, необхідні для професійного дизайну веб-інтерфейсів. Вона дозволяє одночасно кільком дизайнерам працювати над одним проектом, що значно покращує продуктивність та координацію команди [14, 15].

Крім того, Adobe Photoshop і Illustrator залишаються незамінними інструментами в галузі графічного дизайну. Ці програми вже давно визнані за їхній широкий функціонал для роботи зі зображеннями, векторною графікою та іншими елементами дизайну. Adobe Photoshop пропонує потужні інструменти для редагування фотографій та створення растрової графіки, тоді як Illustrator спеціалізується на векторній графіці, що є важливим для створення логотипів, іконок та інших елементів веб-дизайну [13]. Обидва ці інструменти інтегруються з іншими продуктами Adobe, що дозволяє легко переносити роботу між різними програмами та використовувати найкращі можливості кожної з них.

Таким чином, використання Adobe XD, Sketch, Figma, Adobe Photoshop та Illustrator забезпечує дизайнерам широкий вибір інструментів для створення якісних та професійних веб-інтерфейсів. Кожен з цих інструментів має свої унікальні можливості та переваги, що дозволяє дизайнерам обирати найбільш підходящий для їх конкретних потреб та завдань.

Figma обрано для створення дизайну макетів веб-сторінок через його безкоштовність та зручний інтерфейс. Цей інструмент дозволяє ефективно працювати над проектами в реальному часі та спільно редагувати макети з усіма учасниками команди. Крім того, Figma має широкий функціонал для

створення професійних макетів і дозволяє ефективно взаємодіяти з іншими програмами та сервісами для розробки веб-додатків (рис. 4.1).

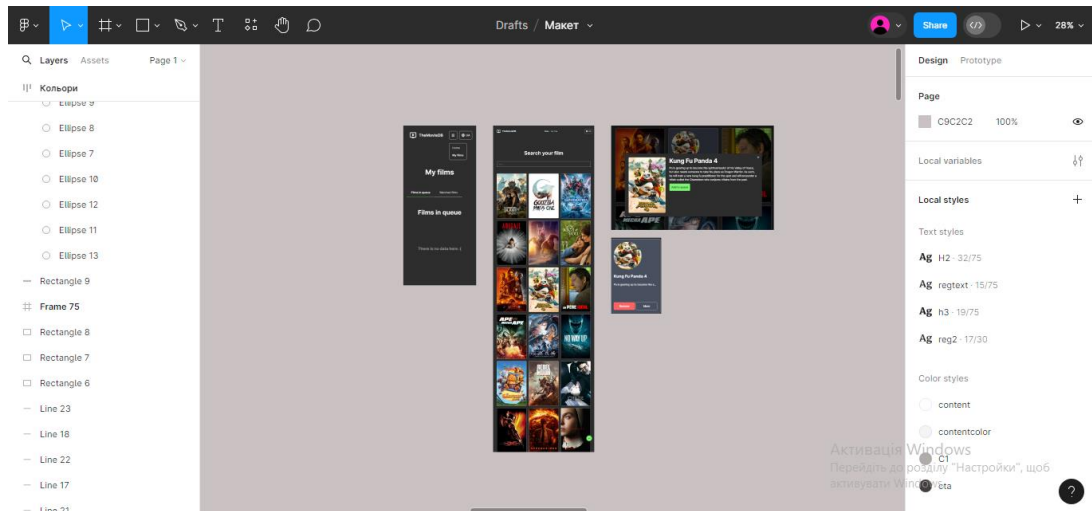


Рисунок 4.1 – Інтерфейс Figma

Figma є популярним інструментом для дизайну та прототипування завдяки ряду переваг, які роблять його привабливим для користувачів. Одна з ключових переваг Figma полягає в його інтерактивності та можливості спільної роботи. Цей інструмент дозволяє багатьом користувачам працювати над одним проєктом одночасно, надаючи можливість в реальному часі бачити зміни, вносити коментарі та взаємодіяти з іншими членами команди. Це значно полегшує координацію та підвищує ефективність роботи над проєктами.

Figma також має широкий функціонал, що включає різноманітні інструменти для створення векторних макетів, прототипування, редагування тексту, кольорів та стилів. Це дозволяє дизайнерам виконувати всі необхідні завдання в одному середовищі, не потребуючи додаткових інструментів. Ще однією важливою перевагою Figma є хмарне зберігання та доступність. Проєкти зберігаються в хмарі, що дозволяє отримувати доступ до них з будь-якого пристрою з Інтернет-підключенням, забезпечуючи таким чином зручність та гнучкість роботи [15].

Figma також характеризується крос-платформенним доступом. Цей інструмент працює у веб-браузері, а також має десктопні версії для macOS та

Windows, що робить його доступним для користувачів на різних платформах. Це особливо важливо для команд, які працюють з різними операційними системами. Крім того, Figma має простий інтерфейс, який є інтуїтивно зрозумілим. Це робить його дружнім для новачків, які можуть швидко освоїти основні функції. Широкий діапазон документації та онлайн-ресурсів також сприяє самостійному вивченню та підвищенню кваліфікації користувачів.

Проте, незважаючи на численні переваги, Figma має і деякі недоліки. Оскільки Figma базується на хмарних технологіях, вона потребує постійного Інтернет-підключення. Це може стати проблемою для користувачів у регіонах з нестабільним Інтернетом або для тих, хто працює в автономному режимі. Також у випадку дуже складних або великих проєктів Figma може сповільнити роботу або виявитися нестабільним, що є проблемою для команд, які працюють над великими проєктами з великою кількістю компонентів.

Figma має обмежений вибір вбудованих шрифтів, що може ускладнити роботу з деякими дизайнами, особливо якщо потрібен конкретний шрифт, який не доступний у Figma. Крім того, хоча Figma надає можливість спільної роботи над проєктами, в деяких випадках може бути складно керувати доступом до файлів і контролювати права доступу користувачів, особливо у великих командах або під час співпраці з клієнтами.

Для роботи з Figma необхідно мати стабільне Інтернет-підключення, оскільки цей інструмент працює у веб-браузері і всі проєкти зберігаються в хмарі. Рекомендується використовувати оновлені версії веб-браузерів, таких як Google Chrome, Mozilla Firefox, Safari або Microsoft Edge, для оптимальної підтримки функціональності Figma. Хоча Figma доступний для користувачів macOS, Windows та Linux, для роботи з векторними графічними редакторами рекомендується використовувати сучасні версії операційних систем. Відповідні характеристики комп'ютера також є важливими для оптимальної роботи з Figma. Хоча цей інструмент не вимагає великих обчислювальних потужностей, добре функціонуюча графічна підсистема сприятиме більш плавній та ефективній роботі з проєктами.



При виборі інтегрованого середовища розробки (IDE) для проєкту було обрано Visual Studio Code (VSCode) (рис. 4.2).

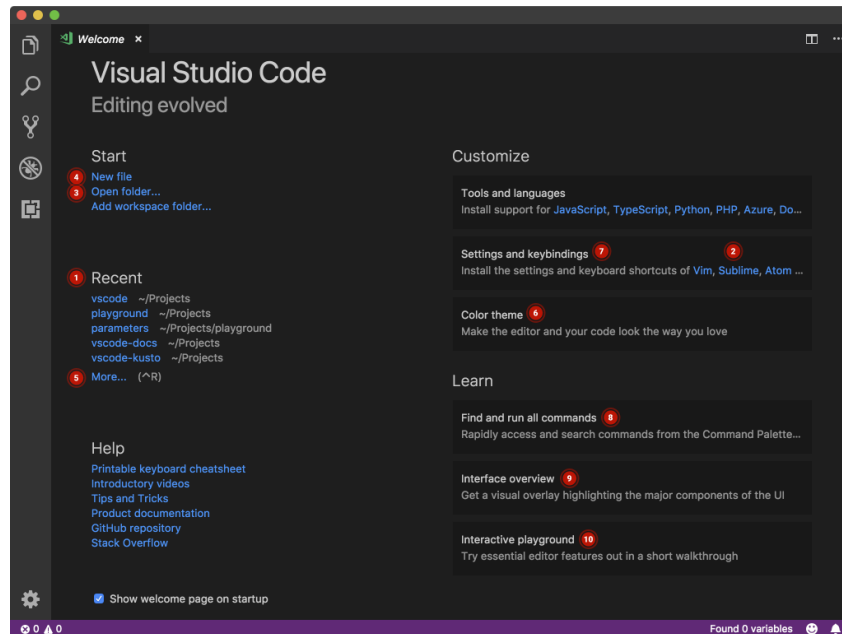


Рисунок 4.2 – Інтерфейс VSCode

Популярними інтегрованими середовищами розробки (IDE) для веб-розробки, окрім Visual Studio Code, є такі середовища, як WebStorm, Atom, Sublime Text, Brackets і Eclipse. Кожне з них має свої унікальні переваги та можливості, що робить їх привабливими для різних груп розробників.

WebStorm, розроблене компанією JetBrains, є потужним інтегрованим середовищем розробки для JavaScript, HTML і CSS. Воно пропонує широкий спектр функціональних можливостей, які сприяють комфортній роботі з веб-технологіями. Завдяки своїм можливостям, WebStorm дозволяє розробникам ефективно працювати над складними проєктами, забезпечуючи підтримку новітніх технологій та стандартів.

Atom, створений компанією GitHub, є відкритим інтегрованим середовищем розробки, яке можна налаштовувати за допомогою плагінів та тем. Це середовище надає різноманітні можливості для зручної роботи з кодом, включаючи інтелектуальне автодоповнення, інтерфейс для роботи з git та багато інших корисних функцій. Завдяки своїй відкритості та

налаштовуваності, Atom став популярним серед розробників, які цінують гнучкість та можливість підлаштовувати середовище під свої потреби.

Sublime Text є швидким, легким та потужним інтегрованим середовищем розробки, яке володіє великою кількістю корисних функцій та можливостей. Серед них варто відзначити швидке перехресне посилання, можливість розширення функціоналу за допомогою плагінів, а також підтримку великої кількості мов програмування. Завдяки своїй продуктивності та зручності використання, Sublime Text користується популярністю серед розробників, які шукають простий та ефективний інструмент для написання коду.

Brackets, розроблене Adobe, є безкоштовним інтегрованим середовищем розробки, яке спеціалізується на веб-розробці. Brackets надає зручні інструменти для роботи з HTML, CSS і JavaScript, такі як попередній перегляд коду та вбудовані функції збірки. Це середовище орієнтоване на веб-розробників, які шукають простий та функціональний інструмент для створення веб-додатків.

Eclipse є популярним інтегрованим середовищем розробки, особливо популярним серед Java-розробників, але також підтримує розробку веб-додатків за допомогою плагінів, таких як Eclipse Web Tools Platform (WTP). Eclipse надає розробникам потужні інструменти для написання коду, налагодження та тестування додатків. Завдяки своїй гнучкості та можливостям розширення, Eclipse залишається популярним вибором серед розробників, які працюють з різними мовами програмування та технологіями.

Порівняння Visual Studio Code з іншими популярними інтегрованими середовищами розробки (IDE) дозволяє краще зрозуміти їхні особливості та визначити найбільш підходящий інструмент для конкретних потреб розробника. Visual Studio Code відомий своєю швидкодією та продуктивністю, а також широким спектром розширень, які дозволяють налаштувати середовище розробки для власних потреб. Завдяки активній спільноті користувачів і постійним оновленням, VSCode залишається одним з

найпопулярніших інструментів серед розробників. Це середовище забезпечує підтримку багатьох мов програмування, інтеграцію з системами контролю версій, інтелектуальне автодоповнення та багато інших функцій, що роблять його універсальним інструментом для веб-розробки.

Недоліками VSCode можуть бути обмежені можливості для роботи з великими проєктами порівняно з іншими IDE, а також виснажливе використання ресурсів системи у порівнянні з легковаговими альтернативами. Однак, з точки зору загальної продуктивності та розширюваності, VSCode зазвичай виявляється більш перевагою. Хоча інші IDE, такі як WebStorm, Atom, Sublime Text, Brackets та Eclipse, мають свої сильні сторони, Visual Studio Code залишається одним з найбільш зручних та універсальних інструментів для багатьох видів розробки програмного забезпечення.

Visual Studio Code (VSCode) володіє низкою переваг, які спричинили його вибір як основного інтегрованого середовища розробки для багатьох розробників. Однією з головних переваг є те, що VSCode є безкоштовним програмним забезпеченням. Це робить його доступним для широкого кола користувачів, незалежно від операційної системи, яку вони використовують, будь то Windows, macOS чи Linux.

Ще однією важливою перевагою VSCode є його розширюваність. Велика кількість доступних розширень та плагінів дозволяє налаштувати середовище розробки під конкретні потреби та завдання. Це робить VSCode відмінним вибором для розробників різних напрямків, оскільки за допомогою розширень можна додавати нові функції та покращувати існуючі.

Підтримка мов програмування є ще однією значущою перевагою VSCode. Це середовище розробки надає підтримку для широкого спектру мов програмування, включаючи JavaScript, TypeScript, HTML, CSS та інші популярні мови. Така багатомовність дозволяє розробникам працювати з різними мовами у межах одного середовища, що значно спрощує робочий процес.

Вбудована підтримка Git у VSCode дозволяє зручно керувати версіями коду. Це включає можливість робити коміти, злиття та інші операції з Git

безпосередньо з середовища розробки. Така інтеграція значно підвищує продуктивність роботи та знижує потребу у використанні додаткових інструментів для керування версіями.

Легкість використання є ще однією перевагою VSCode. Простий та інтуїтивно зрозумілий інтерфейс робить його доступним навіть для початківців у програмуванні. Відсутність складних налаштувань та зручна навігація дозволяють новачкам швидко освоїтися та розпочати роботу з кодом.

Активна спільнота користувачів є ще однією важливою перевагою VSCode. Велика та активна спільнота, яка постійно розвивається, надає підтримку через форуми, блоги та інші ресурси. Це дозволяє користувачам швидко знайти відповіді на свої питання та отримати допомогу у вирішенні проблем.

Нарешті, швидкодія та продуктивність є одними з ключових аспектів, які роблять VSCode відмінним вибором для розробників з будь-яким рівнем досвіду. Швидке завантаження, ефективна робота з великими проєктами та мінімальні затримки під час редагування коду забезпечують високий рівень продуктивності та комфорту під час роботи.

При виборі пакетного менеджера для розробки веб-сайту важливо враховувати його зручність, швидкість та сумісність з використовуваними інструментами. Декілька з найпопулярніших пакетних менеджерів для веб-розробки включають npm, Yarn та pnpm. Кожен з них має свої переваги та особливості, які варто врахувати перед вибором.

npm – це стандартний пакетний менеджер для JavaScript, який включений у Node.js. Він має велику кількість пакетів та підтримується активною спільнотою розробників. Однак, у деяких випадках npm може виявитися повільним при встановленні пакетів.

pnpm – це ще один альтернативний пакетний менеджер, який вирізняється тим, що встановлює пакети в один екземпляр для всього проєкту, що дозволяє економити місце на диску та прискорює встановлення пакетів.

Yarn – це альтернативний пакетний менеджер, розроблений Facebook, який ставить за мету поліпшення швидкості та надійності встановлення

пакетів порівняно з npm. Він також має кешування, що дозволяє зменшити час завантаження пакетів у майбутньому.

Yarn має кілька переваг, які роблять його привабливим для використання в проєкті. Однією з ключових переваг є його швидкість. Yarn відомий своєю швидкістю завантаження та встановлення пакетів, завдяки використанню кешування для збереження завантажених пакетів та їхніх версій. Це дозволяє значно прискорити процес встановлення пакетів, які вже були завантажені раніше, що є великою перевагою при роботі з великими проєктами.

Ще однією важливою перевагою Yarn є його надійність. Yarn забезпечує надійність установки пакетів, оскільки він гарантує однаковість встановлення пакетів на різних системах та середовищах. Це означає, що незалежно від того, на якій платформі чи в якому середовищі працює розробник, Yarn гарантує, що всі пакети будуть встановлені однаково і без помилок, що значно спрощує процес розробки та підтримки проєкту.

Зручний інтерфейс командного рядка є ще однією значущою перевагою Yarn. Він має зручний та інтуїтивно зрозумілий інтерфейс командного рядка, що спрощує використання та налаштування пакетів у проєкті. Завдяки цьому, розробники можуть легко виконувати всі необхідні команди та операції, не витрачаючи багато часу на освоєння інтерфейсу.

Підтримка локального кешування є ще однією перевагою Yarn. Він дозволяє зберігати завантажені пакети у локальному кеші, що зменшує час завантаження пакетів та економить мережні ресурси. Це особливо корисно в умовах обмеженого доступу до Інтернету або при роботі в автономному режимі, оскільки дозволяє продовжувати роботу з проєктом без необхідності постійного завантаження одних і тих самих пакетів.

Активна спільнота користувачів є перевагою Yarn. Він користується великою популярністю серед розробників та має активну спільноту, яка постійно вносить внески у його розвиток та підтримку. Це забезпечує постійне оновлення Yarn, додавання нових функцій та виправлення помилок, що робить його ще більш надійним та зручним інструментом для розробки веб-проєктів.

## 5 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ТА НАВІГАЦІЇ

Вибір структури сайту має вирішальне значення для забезпечення зручного та ефективного використання користувачами. Це є ключовим етапом у процесі проєктування веб-сайту, який визначає спосіб організації та навігації на веб-платформі. Нижче представлено чотири приклади структур сайту, кожен з яких має свої переваги та недоліки. Вибір оптимальної структури варто здійснювати з урахуванням потреб та очікувань цільової аудиторії.

Нижче наведено чотири приклади структур сайту. Кожна з них має свої переваги та недоліки у проєктуванні веб-сайту.

Стандартна структура: Головна веб-сторінка містить посилання на інші сторінки веб-сайту, а сторінки в свою чергу містять посилання на головну. Це є найпростіший і найпоширеніший спосіб організації веб-сайту (рис. 5.1).

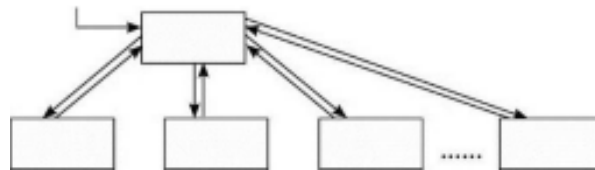


Рисунок 5.1 – Стандартна схема структури сайту

Каскадна структура: У цьому випадку посилання на сайті задані таким чином, що існує тільки один шлях обходу сторінок веб-сайту. Користувачі можуть переміщуватися тільки в одному з напрямків вперед або назад (рис. 5.2).



Рисунок 5.2 – Каскадна схема структури сайту

Хмарочосна структура: У цій моделі користувач може опинитися на певних сторінках, за умовою, що вони йдуть певним (правильним) шляхом (рис. 5.3).

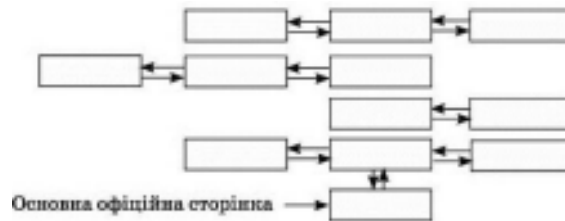


Рисунок 5.3 – Хмарочосна схема структури сайту

Павутина. У цьому випадку всі сторінки веб-сайту містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу [9]. Ця схема може заплутати користувача, але вона частіше використовується в тих випадках, коли посиланнями на документи користуються не занадто часто (рис. 5.4).

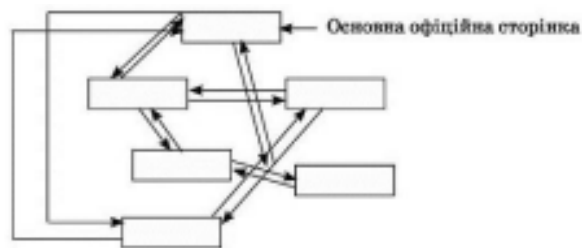


Рисунок 5.4 – Павутина схема структури сайту

Запропонована в роботі структурна організація веб-додатка, дозволяє зробити розподіл функцій, компонентів та модулів, а також визначення взаємозв'язків між ними для забезпечення ефективної роботи та відповідності вимогам замовника.

Головна сторінка фільмотеки передбачає швидкий доступ до основних функцій додатку.

Пошук фільмів. Відображає форму для пошуку фільмів та результати пошуку.

Перегляд деталей. Показує детальну інформацію про фільм, таку як назва, рік випуску, опис тощо.

Архітектура головної сторінки фільмотеки включає динамічну пагінацію, яка зберігає запити у URL для зручності навігації користувачів між сторінками результатів пошуку. Кожен запит зберігається у URL, щоб користувачі могли легко поділитися посиланнями або зберегти стан сторінки. Крім того, доступ до локалізації, яка дозволяє користувачам змінити мову на українську або англійську, доступний на будь-якій сторінці веб-сайту. При зміні мови весь контент автоматично переключається на обрану мову, забезпечуючи зручність та доступність сайту для користувачів.

Сторінка доданих фільмів створена для можливості роботи з вже переглянутими або спеціально відібраними фільмами.

Фільми в черзі. Показує фільми, які користувач додав у чергу для подальшого перегляду.

Переглянуті фільми. Відображає фільми, які користувач вже переглянув.

Модальне вікно для перегляду інформації про фільм створює можливість працювати з конкретними фільмами, показує розширену інформацію про обраний фільм, таку як рейтинг, актори, жанр тощо.

Архітектура фільмотеки базується на клієнт-серверній архітектурі, де клієнтська частина (фронтенд) взаємодіє з серверною (бекенд) за допомогою API для отримання та відображення інформації про фільми. Фронтенд побудований з використанням сучасного JS фреймворку для створення динамічних та інтерактивних елементів, таких як модальні вікна та форми пошуку.

Ця структура дозволяє користувачам зручно взаємодіяти з веб-додатком, знаходити та організовувати фільми за їхніми індивідуальними потребами та вподобаннями. Нижче наведено навігаційні елементи сайту (рис. 5.5-5.9).

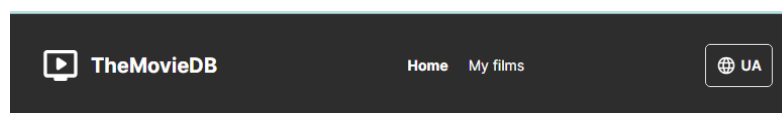


Рисунок 5.5 – Навігація в хедері сайту (Desktop)



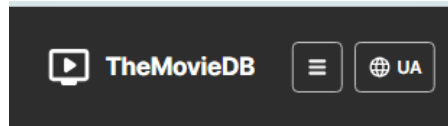


Рисунок 5.6 – Навігація в хедері сайту (Mobile)

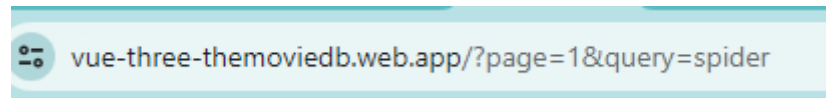


Рисунок 5.7 – Збереження стану прокрутки в URL (пагінація)

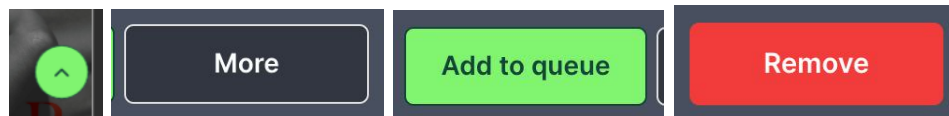


Рисунок 5.8 – Приклади зображень UI

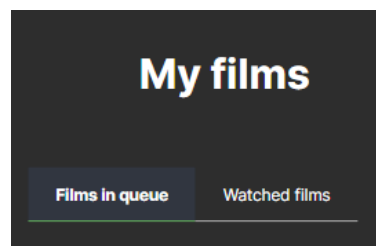


Рисунок 5.9 – Навігація на підсторінки

Структура веб-сайту є основою, на якій ґрунтується весь функціонал та навігація користувача. Вона визначає, як будуть організовані різні розділи та сторінки сайту, як вони будуть взаємодіяти між собою та яким чином користувачі будуть переміщатися в межах веб-платформи. Оптимально спроектована структура дозволяє забезпечити зручний доступ до контенту, спрощує пошук необхідної інформації та підвищує загальний рівень задоволення від використання сайту. Правильно обрана структура є важливою передумовою успіху будь-якого веб-проєкту, незалежно від його масштабів та функціональності.

## 6 РОЗРОБКА МОДУЛЬНОЇ СІТКИ

Під час проєктування модульної сітки мається на меті створити систему розміщення елементів і блоків, яка буде однорідною та легко адаптовуватися до різних екранних розмірів. Це дозволить забезпечити оптимальне використання простору, забезпечити правильне розташування контенту та зробити веб-сайт привабливим та зручним для відвідувачів.

У даному розділі розглядаються основні принципи та методики розробки модульної сітки, визначимо параметри та структуру сітки, а також розглянемо особливості її впровадження в проєкті. Ретельне проєктування та налагодження модульної сітки є важливим етапом у створенні веб-сайту, що впливає на його вигляд, функціональність та користувацький досвід.

У проєкті використовується стандартна 4-колоночна модульна сітка, що є відомим і ефективним інструментом для організації структури веб-сайтів. Кожна сторінка розділена на 4 колонки, що дозволяє розміщувати елементи з точністю та гнучкістю. Крім того, для хедеру використовується флюїдний контейнер, який адаптується до різних розмірів екранів, забезпечуючи щільне розташування контенту і зручну навігацію для користувачів. Використання такої сітки дозволяє забезпечити зручне розташування елементів та оптимальне використання доступного простору на всіх типах пристроїв.

Для створення модульної сітки було застосовано середовище редактору Figma (рис. 6.1).

Для сайту було обрано модифіковану модульну сітку 4. Для налаштування стовпців було обрано наступні параметри (рис. 6.2).

- кількість стовпців 4;
- вирівнювання по центру;
- ширина модулю 91 px;
- виїмка 21 px.

Додатково була створена Grid-сітка (рис. 6.3).

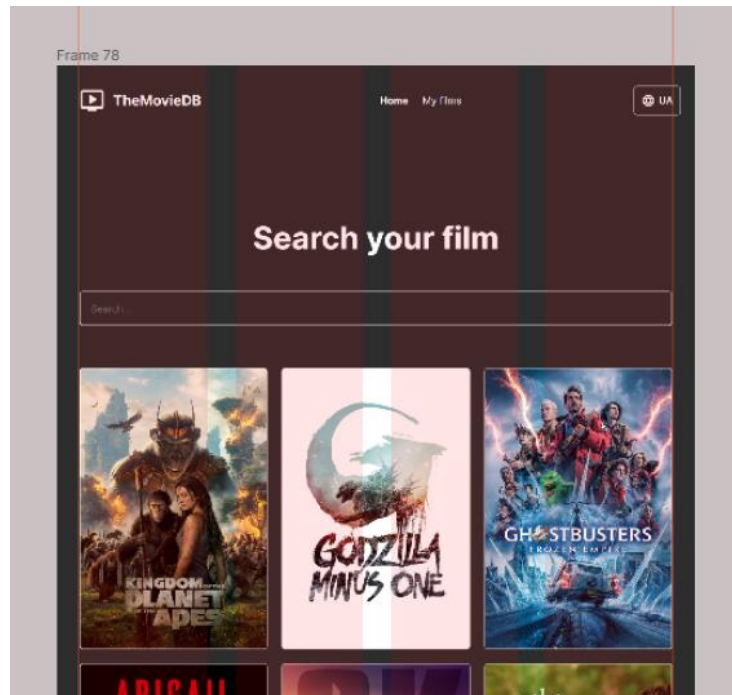


Рисунок 6.1 – Модульна сітка у середовищі Figma

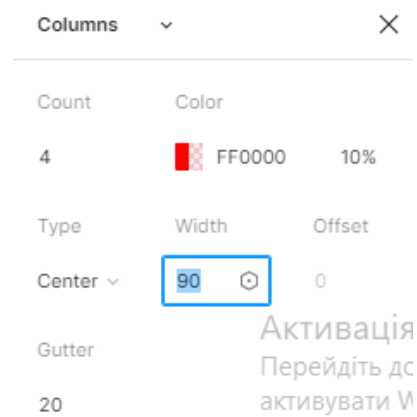


Рисунок 6.2 – Налаштування стовпців

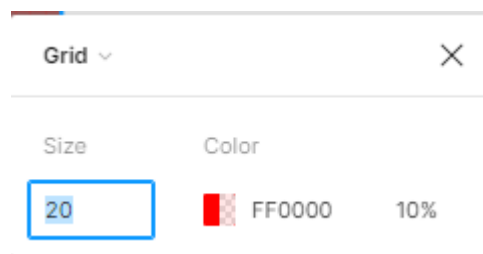


Рисунок 6.3 – Налаштування Grid-сітки

## 7 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ

Графічний дизайн відіграє ключову роль у створенні приємного та зручного для користувачів інтерфейсу, який привертає увагу та сприяє ефективному взаємодії з продуктом.

Дуже важливим є вибір колірної схеми, типографіки, іконок та інших графічних елементів, які визначатимуть стиль та атмосферу веб-додатку. Крім того, необхідно враховувати практичні аспекти роботи з різними інструментами та програмами для створення графічного контенту.

Перш ніж приступати до розробки, було вивчено зауваження замовника веб-додатку, що дозволило врахувати його потреби, вподобання та звички та створити дизайн, який привабливий та зручний саме для цього користувача. При цьому враховувалися такі основні вимоги до дизайну.

Простота та зручність. Дизайн повинен бути простим та зрозумілим для користувачів без перевантаження інформацією та складних елементів у дизайні.

Консистентність. Стиль та дизайн повинен бути однаковою на всіх сторінках веб-додатку задля створення єдиного вигляду для всього додатку та сприяння легкому використанню.

Колірна схема та типографіка, повинна в першу чергу відповідати уподобанням користувача, але враховувати атмосферу та цілі додатку. Шрифти для текстових елементів повинні бути чіткі та читабельні.

Адаптивний дизайн. Хоча основне призначення додатку є використання на стаціонарних комп'ютерах, але все ж таки можливість використання на мобільних пристроїв повинна бути врахована.

Тестування та зворотній зв'язок. Перед передачею додатку замовнику було проведено тестування дизайну у реальних умовах та його обладнанні. Обробка вражень та пропозицій замовника щодо поліпшення додатку, та регулярний зворотний зв'язок допоміг спочатку обрати, а після і вдосконалити дизайн додатку.

Дотримуючись цих рекомендацій, було розроблено графічний дизайн, який задовільняє побажання замовника, який передбачає використання темної колірної схеми з акцентним зеленим кольором та шрифтом Inter. Вибір темної палітри допомагає створити сприятливі умови для тривалого перегляду, а зелений колір використовується для виділення важливих елементів.

Щодо типографіки, шрифт Inter обраний з метою забезпечення читабельності та зручності. Він використовуватиметься для текстових елементів і заголовків, щоб забезпечити консистентність та гармонію у всьому веб-додатку.

Іконки та графічні елементи будуть виконані в мінімалістичному стилі з контрастними кольорами для полегшення сприйняття та навігації. Це дозволить користувачам швидко зорієнтуватися на сторінці та легко знаходити необхідну інформацію.

Весь дизайн буде спроектований з фокусом на простоту, зручність та чіткість, щоб забезпечити приємний та ефективний досвід використання веб-додатку для користувачів. Ретельне вивчення та впровадження цих дизайнерських рішень допоможе нам створити привабливий та функціональний веб-додаток, який задовольнить потреби наших користувачів. Нижче наведені основні UI-елементи веб-сайту (рис. 7.1-7.4)

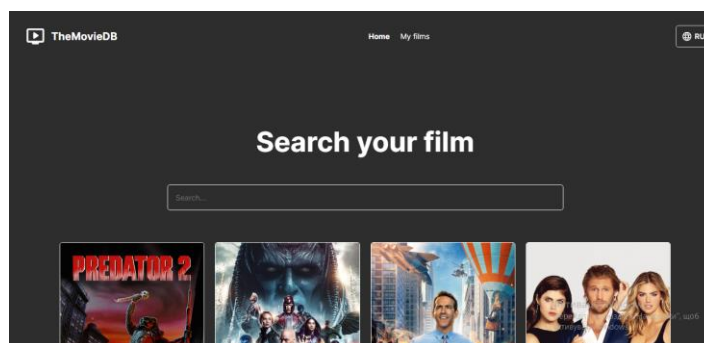


Рисунок 7.1 – Головна сторінка сайту



Рисунок 7.2 – Основна палітра сайту



Рисунок 7.3 – Модальне вікно

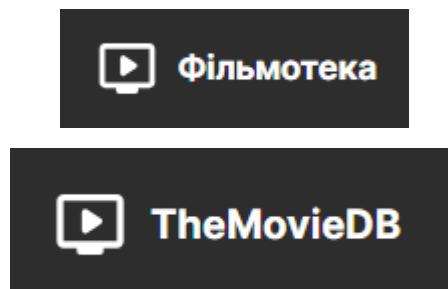


Рисунок 7.4 – Логотипи сайту «Фільмотека»

## 8 НАПОВНЕННЯ КОНТЕНТОМ СТОРІНОК ВИДАННЯ

Гарне наповнення контентом сайту є критично важливим для його успіху та привабливості для користувачів. Перш за все, чітка структура і організація відіграють вирішальну роль. Ретельно розроблена структура сайту допомагає користувачам легко зорієнтуватися та швидко знайти потрібну інформацію. Це означає, що контент повинен бути розділений на логічні блоки та групи, з дотриманням принципу послідовності та логічності, щоб забезпечити зручність використання. Процес наповнення контентом дозволяє ефективно організувати інформацію, що представлена користувачам, забезпечуючи логічність та зручність користування.

При цьому необхідно розглянути різні типи контенту, його розміщення на головній сторінці, сторінці бібліотеки та сторінці з детальною інформацією про фільм. Розробити зручний та інтуїтивно зрозумілий інтерфейс, що задовольнятиме потреби користувачів та сприятиме їхньому комфортному використанню веб-додатку.

Візуальне оформлення також має значення. Використання візуальних елементів, таких як фотографії, відео, ілюстрації та інші матеріали, може значно підвищити привабливість контенту. Візуальні елементи не лише привертають увагу, але й підкреслюють ключові моменти, роблячи інформацію більш зрозумілою та запам'ятовуваною. Послідовна навігація на сайті є важливою для забезпечення зручності користування. Важливо, щоб користувачі легко могли переміщатися по сайту та знаходити необхідну інформацію. Це досягається завдяки зрозумілим міткам та підказкам, які допомагають в орієнтуванні та роблять навігацію інтуїтивно зрозумілою.

Забезпечення можливості взаємодії з додатком через коментарі, форми зворотного зв'язку, соціальні мережі та інші засоби комунікації створює додаткові зручності у використанні додатку.

Нарешті, адаптивність контенту для різних пристроїв та екранних розмірів є важливою для використання додатку в особливих умовах, наприклад у подорожі, коли відсутні стандартні засоби відображення інформації. Переконайтеся, що контент добре відображається на мобільних пристроях, є обов'язковою умовою, оскільки все більше користувачів звертаються до Інтернету через смартфони та планшети. Це означає, що сайт повинен бути розроблений з урахуванням адаптивного дизайну, щоб забезпечити оптимальний досвід користувачам незалежно від пристрою, що використовується.

Щоб забезпечити ефективне та зручне використання веб-додатку, інформація на сторінках організована у вигляді різних блоків, які логічно групують функціонал та контент. На головному екрані розміщено поле пошуку фільмів з кнопкою пошуку та відображенням результатів у формі карток або списку. У блоку «Моя бібліотека» – дві підкатегорії: «Черга» та «Переглянуті фільми», кожна з яких міститиме список фільмів з можливістю додавання або видалення. На сторінці з детальною інформацією про фільм вказано опис, рейтинг, акторів та трейлер, а також опцію для додавання фільму до черги або переглянутих. Прототип додатку розроблений у Figma або аналогічному інструменті для візуалізації розміщення елементів та навігації перед подальшою реалізацією, забезпечуючи зручний та інтуїтивно зрозумілий дизайн для користувачів. Нижче наведені скріншоти контенту сайту (рис. 8.1-8.4)



Рисунок 8.1 – Трьомірна анімація FlipCard



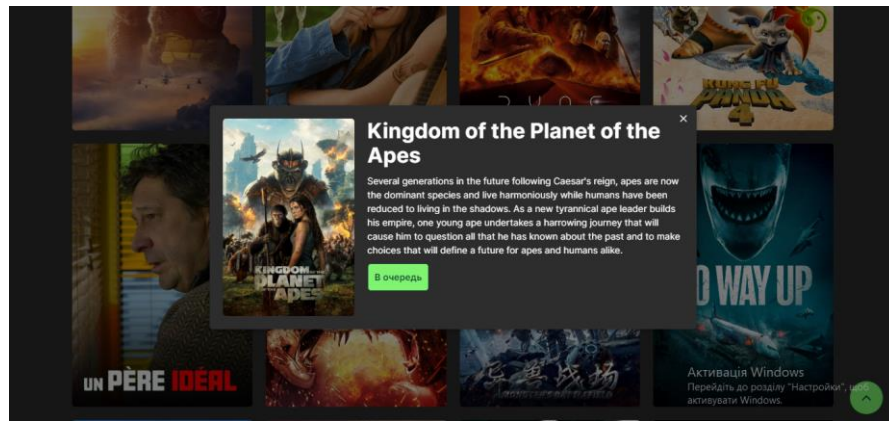


Рисунок 8.2 – Перегляд інформації про фільм

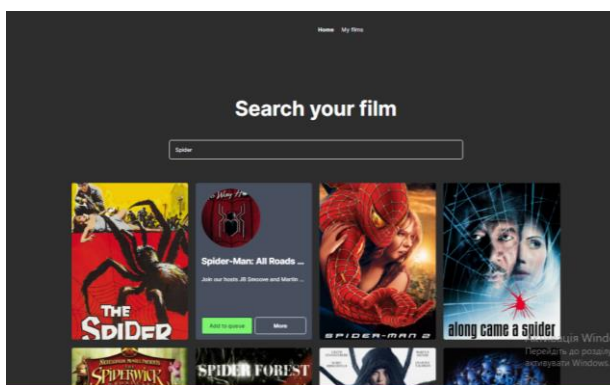


Рисунок 8.3 – Пошук фільмів на сторінці

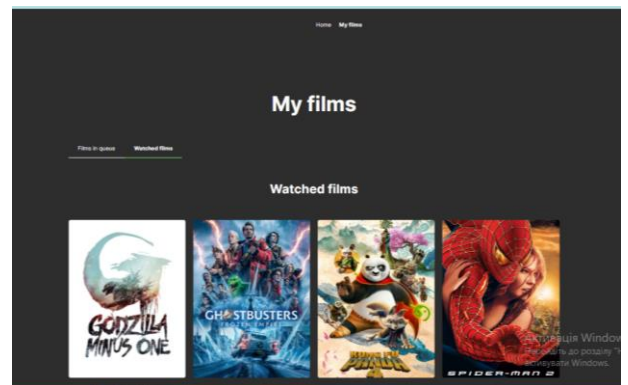


Рисунок 8.4 – Сторінка зі збереженими фільмами

Завдяки підходу, де в URL зберігається стан прокрутки та пошуковий запит – можна ділитися конкретним результатом пошуку з іншими за допомогою посилання. Так наприклад, перейшовши за посиланням: <https://vue-three-themoviedb.web.app/?page=2&query=Spider> можна перейти одразу на другу сторінку результатів за запитом Spider.

Крім цього, debounce оптимізує систему пошуку роблячи невелику затримку під час швидкого прогортання та вводу тексту, що зменшує кількість запитів на сервер.

## 9 ТЕСТУВАННЯ І ПУБЛІКАЦІЯ

Тестування відіграє важливу роль у забезпеченні якості та надійності додатку перед його публікацією. Відповідно, цей процес включає перевірку різних аспектів, таких як функціональність, сумісність та продуктивність.

Після завершення розробки веб-додатку було здійснено тестування для переконання в його коректній роботі на різних пристроях та в різних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Safari, Microsoft Edge та інші. Під час тестування кросбраузерності зверталась особлива увага на коректне відображення контенту, правильну роботу функціоналу та відсутність помилок або несправностей.

Після тестування кросбраузерності та кросплатформеності виявлено, що веб-додаток працює стабільно та безперебійно на всіх перевірених платформах та веб-браузерах.

Перевірка проводилась на хостингу Firebase, що забезпечує надійне хостингове середовище та швидку публікацію веб-додатків, та дозволяє забезпечити доступ користувачам з усього світу до розробленого продукту.

Нижче наведено процес деплою веб-додатку, створеного з використанням Vue 3 та Vite, на хостинг Firebase:

а) створення проєкту у Firebase:

- 1) увійдіть або зареєструйтеся на консолі Firebase;
- 2) створіть новий проєкт і дайте йому назву;
- 3) встановлення і налаштування Firebase CLI;

б) встановіть Firebase CLI за допомогою npm: `npm install -g firebase-tools`:

- 1) увійдіть у свій обліковий запис Firebase за допомогою команди `firebase login`;
- 2) ініціалізуйте ваш проєкт Firebase у кореневій папці вашого додатку командою `firebase init`;

в) налаштування Firebase проєкту:

1) оберіть опцію "Hosting" у процесі ініціалізації інструменту Firebase;

2) виберіть проєкт, який ви створили на кроці 1;

3) слідуйте інструкціям для налаштування опцій деплою, таких як папка, куди буде завантажено ваш додаток;

г) збірка та деплой Vue 3 додатку:

1) збудуйте додаток командою `npm run build` або `yarn build`. Це згенерує оптимізовані файли вашого додатку у папці `dist`;

2) виконайте команду `firebase deploy` для розгортання вашого додатку на Firebase Hosting;

3) після успішного деплою ви отримаєте URL-адресу вашого веб-додатку, на якій він буде доступний.

Сайт доступний за посиланням: <https://vue-three-themoviedb.web.app/>.

## 10 ЕКОНОМІЧНА ЧАСТИНА

У результаті виконання кваліфікаційної роботи створено онлайн-платформу створення SPA «Фільмотека», яка дозволяє забезпечити кінолюбителів інноваційним інструментом для пошуку та перегляду фільмів, з можливістю більш детального ознайомлення з фільмами за допомогою інтерактивних елементів. Мета даного дипломного проєкту – створити на її основі веб-додаток під визначеного замовника за його вимогами.

Економічна ефективність проєкту розраховувалась перед розробкою веб-додатка, але після визначення вимог до кінцевого продукту, у результаті чого можливо спрогнозувати його ціну для замовника [23]. Розрахунок надає можливість замовнику провести аналіз використання різних рішень при проєктуванні порівняно з аналогами.

Платформа «Фільмотека» створена за допомогою інструментальних засобів, які широко використовуються в індустрії. Це дозволяє скоротити витрати часу на розробку. Інтерфейс максимально оптимізований, в результаті чого досягається зручність для користувача та економія ресурсів при розробці додатку. Оптимізація коду та структури даних дозволяє підвищити швидкість завантаження сторінок та зручність користування.

Сайт не має високих системних вимог, що дозволить користувачам використовувати додаток на різних пристроях.

Порядок проєктування платформи містить у собі наступні етапи:

- початковий етап. Формулюються основні вимоги до платформи, створюється технічне завдання, розробляються специфікації. Проводиться пошук референсів для знаходження цікавих дизайнів та функціональних рішень;
- етап підготовки до розробки, моделювання базових форм інтерфейсу, які в подальшому стануть основою для високодеталізованих версій дизайну;
- етап розробки, виконання основної роботи з програмування фронтенду та бекенду, включаючи створення взаємодії з API;

– етап створення графічних компонентів, на якому проводиться розробка іконок, банерів та інших графічних елементів;

– етап тестування, на якому проводиться тестування платформи на різних пристроях та в різних браузерах для забезпечення сумісності та коректної роботи.

У собівартість розробки платформи входять наступні статті витрат:

- основна заробітна плата;
- додаткова заробітна плата;
- єдиний соціальний внесок;
- інші витрати.

Заробітна плата виконавців: дизайнера – 311,00 грн/год, фронтенд-розробника – 611,00 грн/год, devOps – 311,00 грн/год, QA – 211,00 грн/год. Тривалість робочого дня кожного становить 8 годин, розробка сайту тривала 11 днів. Розрахунок заробітної платні наведено в таблиці 10.1.

Таблиця 10.1– Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконуючий		Ставка, грн/год	Час на виконання, дні	Заробітна плата, грн
		кількість	посада			
1	Формулювання мети веб-сайту	1	дизайнер	311,00	1,5	3732,00
2	Розробка графічного дизайну	1	дизайнер	311,00	2	4976,00
3	Розробка кожного модуля, верстка	1	фронтенд	611,00	4	19552,00
4	Тестування окремих компонентів	1	QA	211,00	1	1688,00
	Комплексне тестування сайту	1	QA	211,00	1	1688,00
	Завантаження на хостинг	1	DevOps	311,00	1,5	3732,00
Разом					11	35368,00
Додаткова заробітна плата (21 %)						7427,28
Усього						42795,28

Надбавки – це додаткова заробітна плата у вигляді премії за понаднормову працю, трудові успіхи тощо. Надбавка у даному випадку становить 21 % від основної зарплати.

$$35368,00 \times 0,21 = 7427,28 \text{ грн.}$$

Єдиний соціальний внесок дорівнює 22 % від величини основної заробітної плати та надбавки та дорівнює 9414,96 грн.

До інших витрат слід віднести витрати на обслуговування комп'ютерної техніки, що використовується виконавцями проекту, а саме плату за електроенергію.

Витрати на електроенергію розраховуються виходячи зі споживаної потужності пристрою і тарифу на електроенергію. У даному випадку передбачається використання моноблоку і ноутбуку з потужністю 0,143 кВт/год і 0,23 кВт/год відповідно. Вартість 1 кВт/год електроенергії прийнято у розмірі 4,32 грн. Оскільки моделі розроблялися 11 днів, час використання електроенергії в процесі розробки складає 88 годин.

Розрахунок плати за електроенергію:

$$(0,143 + 0,23) \times 4,32 \times 88 = 141,80 \text{ грн.}$$

Проект реалізується для одного замовника, тому собівартість розробленого веб-сайту становить:

$$35368,00 + 7427,28 + 9414,96 + 141,80 = 52352,04 \text{ грн.}$$

Суму прибутку від реалізації проекту при рентабельності 31 %:

$$52352,04 * 0,31 = 16229,13 \text{ грн.}$$

Ціна розробки сайту без податку на додану вартість (ПДВ):

$$52352,04 + 16229,13 = 68581,17 \text{ грн.}$$

Ціна сайту з урахуванням ПДВ, що дорівнює 20 % від ціни без ПДВ:

$$68581,17 + (68581,17 * 0,2) = 68581,17 + 13716,23 = 82297,41 \text{ грн.}$$

Результати розрахунків наведено у таблиці 10.2.

Таблиця 10.2 – Розрахунок витрат на розробку та ціни веб-сайту

Витрати	Сума, грн
Основна заробітна плата	35368,00
Додаткова заробітна плата	7427,28
Єдиний соціальний внесок	9414,96
Витрати на електроенергію	141,80
Собівартість розробки сайту	52352,04
Прибуток (рівень рентабельності 31 %)	16229,13
Ціна без ПДВ	68581,17
ПДВ	13716,23
Ціна з урахуванням ПДВ	82297,41

Отже, повна вартість розробки сайту складає 82297,41 грн. Термін реалізації повного обсягу роботи становить 11 днів для команди з чотирьох виконавців. Розрахована сума прибутку становить 16229,13 грн.

## ВИСНОВКИ

Під час розробки веб-сайту було виявлено, що використання інструментів, таких як Figma для дизайну та Visual Studio Code для верстки, сприяло швидкій та ефективній реалізації проєкту. Застосування Vue 3 разом із SCSS дозволило розробити ефективну та добре структуровану фронтенд частину. Користування пакетним менеджером Yarn сприяло кращій управлінню залежностями проєкту.

Структура веб-сайту була проєктована таким чином, щоб забезпечити легкий доступ до основних функцій, таких як пошук фільмів та керування бібліотекою, забезпечуючи при цьому зручну навігацію для користувачів. Застосування модульної сітки допомогло створити систему розміщення елементів і блоків сайту, що сприяє легкому сприйняттю контенту.

Графічний дизайн веб-додатку був розроблений з врахуванням зручності та естетичності, використовуючи темну колірну схему та мінімалістичний стиль. Це дозволило створити приємний та привабливий інтерфейс для користувачів.

Після успішного завершення тестування та деплою на хостинг Firebase було виявлено, що веб-додаток працює стабільно на різних платформах і веб-браузерах, що свідчить про високу якість та готовність до використання.

Розрахована економічна частина проєкту показала, що повна вартість розробки сайту становить 82297,41 грн, при цьому розрахований прибуток складає 16229,13 грн. Термін виконання проєкту склав 11 днів для команди з одного програміста, дизайнера та фронтенд-інженера.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Молчанов В.П. Основи проектування WEB-видань : навчальний посібник. Харків: ХНЕУ ім. С. Кузнеця, 2017. 159 с.
2. Бородкіна І.Л., Бородкін Г.О. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів: навчальний посібник. К.: Видавництво Ліра-К, 2020. 212 с.
3. Веб-технології та веб-дизайн: навч. посібник / О.Г. Трофименко, О.Б. Козін, О.В. Задерейко, О.Є. Плачінда. Одеса: Фенікс, 2019. 284 с.
4. Пасічник В.В., Пасічник О.В., Угрин Д.І. Веб-технології. Львів: Магнолія 2006, 2018. 336 с.
5. Пономарьов І.В. Технології розробки кросплатформених веб-додатків: конспект лекцій. Дніпро: ДНУ, 2023. 113 с.
6. Матвєєва Н.О., Пономарьов І.В. Паралельне програмування на платформі .NET: навчальний посібник. Дніпро, 2023. 238 с.
7. Нільсен Я. Веб-дизайн. Зручність використання веб-сайтів. Київ: ТОВ В.Д. Вільямс, 2009. 368 с.
8. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну. К.: Вид. ВНУ. 2009. 336 с.
9. Басюк Т.М., Жежнич П.І. Методи та засоби мультимедійних інформаційних систем. Навчальний посібник. Львів: Львівська політехніка, 2015. 428 с.
10. Тренди веб-розробки 2021. URL: <https://katerinka1211.github.io/left-sidebar.html/> (дата звернення: 28.05.2024).
11. Етапи створення веб-сайтів. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/> (дата звернення: 29.05.2024).
12. Архітектура веб-сайту. URL: <https://uk.theastrologypage.com/website-architecture>. (дата звернення: 29.05.2024).
13. Що таке Adobe Illustrator. URL: <https://uk.education-wiki.com/8613599-what-is-adobe-illustrator> (дата звернення: 30.05.2024).

14. Можливості Figma. URL: <https://hyperhost.ua/info/uk/mozhливosti-figma-servis-dlya-rozrobki-veb-dizaynu/> (дата звернення: 30.05.2024).
15. Що таке Figma. Функції, інструменти и переваги. URL: <https://wezom.academy/chto-takoe-figma-funksii-instrumenty-ipreimuschestva/> (дата звернення: 30.05.2024).
16. Visual Studio Code. URL: <https://www.white-windows.ua/visual-studio-code-bolshe-chem-prosto-kodovuj-redaktor/> (дата звернення: 30.05.2024).
17. Чому VSCode такий відомий? URL: <https://open.zeba.academy/-rochemu-vscode-populyaren/> (дата звернення: 30.05.2024).
18. Поняття структура та різновиди веб-сайтів. Автоматизована розробка веб-сайтів. URL: <http://www.ndu.edu.ua/liceum/web.pdf> (дата звернення: 30.05.2024).
19. Проєктування веб-сайта. URL: <https://www.4stud.in-fo/web-programming/designing-the-site-structure.html/> (дата звернення: 30.05.2024).
20. Каркас сайта, чи модульна сітка в веб-дизайне. URL: <https://webformyself.com/karkas-sajta-ili-modulnaya-setka-v-veb-dizajne/> (дата звернення: 30.05.2024).
21. Тестування і публікація Web-сайта. URL: <https://studoped-ia.ue/1-141512.html> (дата звернення: 30.05.2024).
22. Методичні вказівки з виконання кваліфікаційної роботи для студентів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти спеціальності 186 «Видавництво та поліграфія» за освітньою програмою «Видавничо-поліграфічна справа» / В.П. Ткаченко, А.В. Бізюк, О.В. Вовк, І.М. Єгорова, В.Ф. Челомбійко. Харків: ХНУРЕ, 2020. 68 с.
23. Полозова Т.В. Методичні вказівки до виконання економічної частини кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 186 Видавництво та поліграфія усіх форм навчання. Харків: ХНУРЕ, 2022. 47 с.