

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 ОНЛАЙН-БІБЛІОТЕКИ ФІЛЬМІВ З
ВИКОРИСТАННЯМ CMS WORDPRESS**

**DEVELOPMENT OF AN ONLINE MOVIE LIBRARY USING CMS
WORDPRESS**

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

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

Виконав: здобувач вищої освіти
Групи ІПЗс-21
Гемерлінг Вадим Валерійович

(підпис)

Керівник:
асистент
Потейчук Михайло Іванович

(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 НАЦІОНАЛЬНИЙ 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.М. М. М. М.
« 2 » 02 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Гемерлінгу Вадиму Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка онлайн-бібліотеки фільмів з використанням CMS WordPress

Керівник роботи: асистент Потейчук Михайло Іванович

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: 13 рисунків; 8 діаграм.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Потейчук М. І.		
Специфікації вимог до розробленої системи	Потейчук М. І.		
Розробка об'єкта проєктування	Потейчук М. І.		
Нормоконтроль	Повстяна Ю. С.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		7.5 %	
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.	Виконано
2	Аналіз проблеми розробки та впровадження об'єкту проєктування	27.02.2024 р.	Виконано
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	12.03.2024 р.	Виконано
4	Розробка функціонально-структурної схеми роботи об'єкта проєктування та проєктування бази даних	17.04.2024 р.	Виконано
5	Практична реалізація об'єкта проєктування та розробка бази даних	18.05.2024 р.	Виконано
6	Тестування та налагодження об'єкта проєктування	01.06.2024 р.	Виконано
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	05.06.2024 р.	Виконано

Здобувач вищої освіти

Керівник кваліфікаційної роботи

(підпис)

Гемерлінг В. В.
(прізвище, ініціали)

Потейчук М. І.
(прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія
на кваліфікаційну роботу

Здобувач вищої освіти: Гемерлінг Вадим Валерійович

Тема: «Розробка онлайн-бібліотеки фільмів за допомогою CMS WordPress»

Коротка характеристика кваліфікаційної роботи: Кваліфікаційна робота присвячена вивченню тонкощів розробки онлайн-бібліотеки фільмів, що дає змогу розібратись в аспектах розробки, комерції та просування мультимедійного сайту.

Самостійні розробки і пропозиції автора: У роботі узагальнено методи розробки веб-ресурсів за допомогою CMS WordPress, самостійно розроблено дизайн.

Практичне значення роботи: Продукт, отриманий в результаті виконання кваліфікаційної роботи, може розміщувати в собі рекламу та використовуватись в комерційних, благодійних або інформаційних цілях.

Недоліки: В роботі варто було б детальніше проаналізувати способи просування сайту. Також, окремо слід було б розглянути тему стосовно інтернет-трафіку на сайті.

Загальний висновок: Робота виконана на високому науковому рівні, має логічну та послідовну структуру, відповідає встановленим вимогам, може бути представлена до захисту на державній екзаменаційній комісії та заслуговує позитивної оцінки.

Рецензент: Здолбіцька Ніна Василівна, к.т.н., доцент кафедри комп'ютерних наук
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи _____


(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 Гемерлінг Вадим Валерійович
група ІПЗс-21

Тема кваліфікаційної роботи: Розробка онлай-бібліотеки фільмів з використанням CMS WordPress

Керівник Потейчук Михайло Іванович

Актуальність теми: робота характеризується актуальністю та своєчасністю. Ефективне ведення бізнесу сьогодні неможливе без належної онлайн присутності, а створення веб-сайтів відіграє ключову роль у цьому процесі.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи: автор досліджує різноманітні підходи та інструменти, що використовуються для розробки функціональних веб-сайтів, визначає найефективніші серед них та надає обґрунтування свого вибору. Зміст роботи відповідає вибраній темі. До позитивних аспектів кваліфікаційної роботи включаються логічна послідовність викладу матеріалу та якісна практична реалізація його концепцій.

Зауваження та недоліки: суттєвих недоліків в роботі не виявлено.

Загальний висновок робота виконана на високому рівні та заслуговує позитивної оцінки.

Керівник 
(підпис)

(Потейчук М. І.)
(прізвище, ім'я, по батькові)

«7» червня 2024 р.

АНОТАЦІЯ

Гемерлінг В. В. Розробка онлайн-бібліотеки фільмів за допомогою CMS WordPress. 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 онлайн-платформ і сформульовано завдання. В другому розділі визначено вимоги до структури онлайн-платформи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено та проведено тестування онлайн-бібліотеки фільмів, що отримала назву «FilmsLib». У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки демонструють можливості онлайн-бібліотеки фільмів, створеної за допомогою CMS WordPress.

Ключові слова: веб-сайт, онлайн-платформа, мультимедіа, CMS WordPress, PHP, SQL.

ABSTRACT

Hemerlinh V. Development of an online movie library using WordPress CMS. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and proposals, a list of used sources and annexes.

The first section analyzes the current state of the problem of online platform development and formulates the tasks. The second section defines the requirements for the structure of the online platform, as well as the means, methods and algorithms of development. The third section develops and tests an online movie library called "FilmsLib". The conclusions summarize the information presented in the previous chapters. The results of the development demonstrate the capabilities of the online movie library created using CMS WordPress.

Keywords: website, online platform, multimedia, CMS WordPress, PHP, SQL.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ОНЛАЙН-БІБЛІОТЕК ФІЛЬМІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	15
Висновки до розділу 1	16
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	18
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	18
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	23
Висновки до розділу 2	27
РОЗДІЛ 3 РОЗРОБКА ОНЛАЙН-БІБЛІОТЕКИ ФІЛЬМІВ.....	29
3.1 Практична реалізація об'єкта проектування	29
3.2 Тестування та налагодження інформаційно-комп'ютерної системи.....	36
3.3 Захист інформаційно-комп'ютерної системи.....	38
3.4 Розробка документації для програмного продукту	41
3.5 Розробка інсталяційного пакета	42
Висновки до розділу 3	43
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

ВСТУП

Актуальність роботи. Онлайн-бібліотеки фільмів та онлайн-кінотеатри стали невід'ємною частиною сучасної індустрії розваг. Вони пропонують користувачам зручний доступ до великих колекцій фільмів, серіалів та іншого відеоконтенту через Інтернет. Ця галузь стрімко розвивається, привертаючи мільйони глядачів по всьому світу та генеруючи значні прибутки.

Розробка власного онлайн-кінотеатру є актуальним завданням, враховуючи зростаючий попит на послуги потокового відео та необхідність задовольнити потреби глядачів у зручному доступі до якісного контенту. Створення такої платформи вимагає вирішення низки технічних та функціональних завдань, таких як проектування бази даних, реалізація ефективної системи пошуку та фільтрації, забезпечення безпеки даних користувачів, розробка зручного інтерфейсу та інтеграція мультимедійного вмісту.

Метою цієї кваліфікаційної роботи бакалавра є розробка онлайн-бібліотеки фільмів з використанням системи керування вмістом WordPress. Для реалізації мети потрібно виконати наступні завдання:

- розробити функціональність для каталогізації та класифікації фільмів;
- спроектувати структуру бази даних;
- зреалізувати потужну систему пошуку;
- впровадити систему відгуків;
- інтегрувати функціональність для перегляду мультимедійного вмісту;
- розробити зручний та інтуїтивний інтерфейс користувача;
- забезпечити адміністративну панель для керування вмістом бібліотеки.

Об'єктом дослідження є онлайн-бібліотека фільмів з використанням CMS WordPress.

Предметом дослідження є засоби розробки онлайн-бібліотеки фільмів.

Результати роботи можуть бути використані для створення реального онлайн-кінотеатру, який задовольнить потреби глядачів та забезпечить зручний доступ до великої колекції фільмів.

Розроблена система також може слугувати основою для подальших досліджень та вдосконалень у галузі онлайн-бібліотек та потокового відео.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ОНЛАЙН-БІБЛІОТЕК ФІЛЬМІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Онлайн кінотеатри змінили спосіб, у який ми споживаємо фільми та телевізійний контент. Вони пропонують глядачам можливість переглядати величезні бібліотеки фільмів, шоу та серіалів через Інтернет на різних пристроях, від смартфонів до телевізорів із підтримкою потокового відео. Ця зручність і гнучкість стали ключовими факторами зростання популярності онлайн кінотеатрів, оскільки глядачі можуть насолоджуватися контентом у будь-який час і в будь-якому місці.

Окрім доступу до величезних бібліотек, онлайн кінотеатри також пропонують персоналізовані рекомендації, можливість створювати списки «до перегляду», залишати відгуки та брати участь у спільнотах з іншими кінолюбителями. Деякі сервіси, такі як Netflix, навіть інвестують значні кошти у створення власного оригінального контенту.

Роль онлайн кінотеатрів у сучасній розважальній індустрії є вирішальною. Вони зробили фільми та телебачення більш доступними для широкої аудиторії, знизивши бар'єри для входу та надавши альтернативу традиційним кінотеатрам і методам розповсюдження контенту. Крім того, онлайн кінотеатри сприяють більшій різноманітності та демократизації індустрії, відкриваючи нові можливості для незалежних творців і забезпечуючи платформу для просування їхніх робіт.

Проте, незважаючи на свої переваги, онлайн кінотеатри також стикаються з певними проблемами, такими як питання щодо авторських прав, обмеженого доступу в деяких країнах та можливого негативного впливу на традиційні кінотеатри. Однак, у цілому, вони демонструють значний вплив на розважальну індустрію, перетворюючи спосіб, у який ми споживаємо візуальний контент.

Онлайн-бібліотека фільмів – це веб-ресурс, призначений для збирання, зберігання, організації та надання доступу до інформації про фільми. Основні компоненти такої системи включають:

- база даних фільмів – центральна складова, де зберігається вся інформація про фільми, включаючи назву, рік випуску, жанр, режисерів, акторів, синопсис, рейтинги тощо. База даних повинна бути добре структурована та масштабована для зберігання великих обсягів даних;

- система каталогізації – механізм для класифікації та організації фільмів у базі даних за різними критеріями, такими як жанр, режисер, актор, рік випуску тощо. Ефективна система каталогізації полегшує пошук і фільтрацію фільмів;

- пошукова система – функціональність для користувачів, щоб шукати фільми за назвою, акторами, режисерами або іншими атрибутами. Потужний пошуковий механізм є критично важливим для легкої навігації в великій бібліотеці фільмів;

- користувацькі профілі та рекомендаційна система – можливість для користувачів створювати облікові записи, зберігати список переглянутих фільмів, рейтинги та відгуки. Рекомендаційна система може аналізувати вподобання користувачів і пропонувати релевантні фільми;

- мультимедійний вміст – інтеграція відео-контенту, такого як трейлери, уривки з фільмів або, можливо, повні версії фільмів для перегляду. Це збагатить досвід користувачів і зробить бібліотеку більш привабливою;

- система рейтингів і відгуків – механізм для користувачів, щоб оцінювати фільми та залишати відгуки. Ця функція сприяє залученню спільноти та надає іншим користувачам корисну інформацію;

- соціальна інтеграція – можливість ділитися інформацією про фільми в соціальних мережах, а також інтегрувати функції, такі як коментарі в соціальних мережах або відгуки з інших сайтів;

– адміністративна панель – зручний інтерфейс для адміністраторів для керування вмістом бібліотеки, додавання нових фільмів, редагування існуючих записів, модерації відгуків користувачів тощо.

Ця система може бути реалізована з використанням різних веб-технологій та фреймворків, включаючи традиційні стеки, такі як PHP, Python, Ruby, JavaScript, а також більш сучасні фреймворки, такі як React, Angular або Vue.js для фронтенду. Вибір технологій залежатиме від вимог проекту, масштабованості, бюджету та наявних навичок розробників.

Порівняємо три відомі сервіси, такі як Netflix, Amazon Prime Video, Hulu.

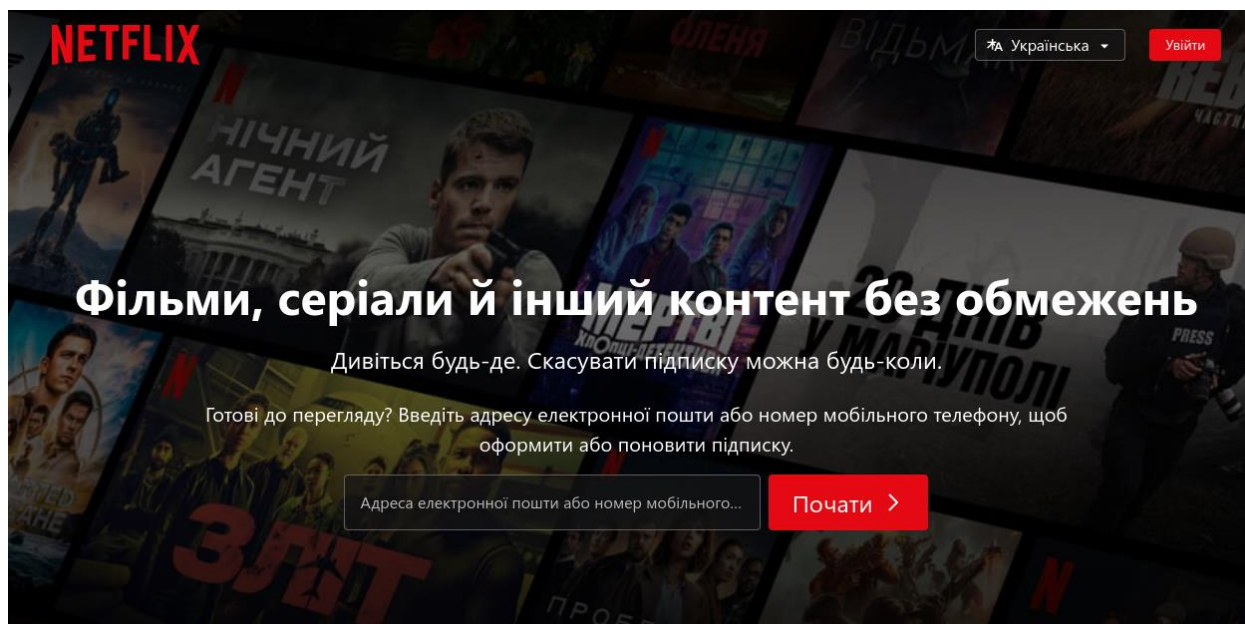


Рисунок 1.1 – Зовнішній вигляд сайту Netflix [1]

Netflix є одним з найвідоміших та найуспішніших онлайн кінотеатрів, що пропонує віртуальний перегляд фільмів [1]. Netflix, Inc. – американський провайдер медійних послуг та продюсерська компанія зі штаб-квартирою в Лос-Гатос, Каліфорнія. Компанія була заснована 1997 року Рідом Гастінгсом і Марком Рендолфом у Скотс-Валлі, Каліфорнія. Дослідження цієї платформи допоможе зрозуміти її особливості та успішні стратегії, що можуть бути використані при розробці вашого онлайн кінотеатру. Ось деякі аспекти, які можна розглянути в аналізі Netflix:

– має велику колекцію фільмів, серіалів, документальних стрічок та інших відеоматеріалів. Вивчення різноманітності та популярності їхнього контенту допоможе визначити, які жанри та типи фільмів можуть бути привабливими для вашої аудиторії;

– виробляє власний оригінальний контент, який став одним із ключових факторів його успіху. Дослідження стратегії створення оригінального контенту та його якості допоможе зрозуміти, як можна підвищити привабливість вашого кінотеатру через власні продукції;

– має потужну технологічну інфраструктуру для стрімінгу відео, обробки платежів та забезпечення безпеки даних. Аналіз їхніх технологічних рішень та інфраструктури допоможе вам побудувати надійну та ефективну систему для вашого онлайн кінотеатру.

Аналіз Amazon Prime Video як прикладної галузі для розробки та створення онлайн кінотеатру для віртуального перегляду фільмів (рис. 1.2).

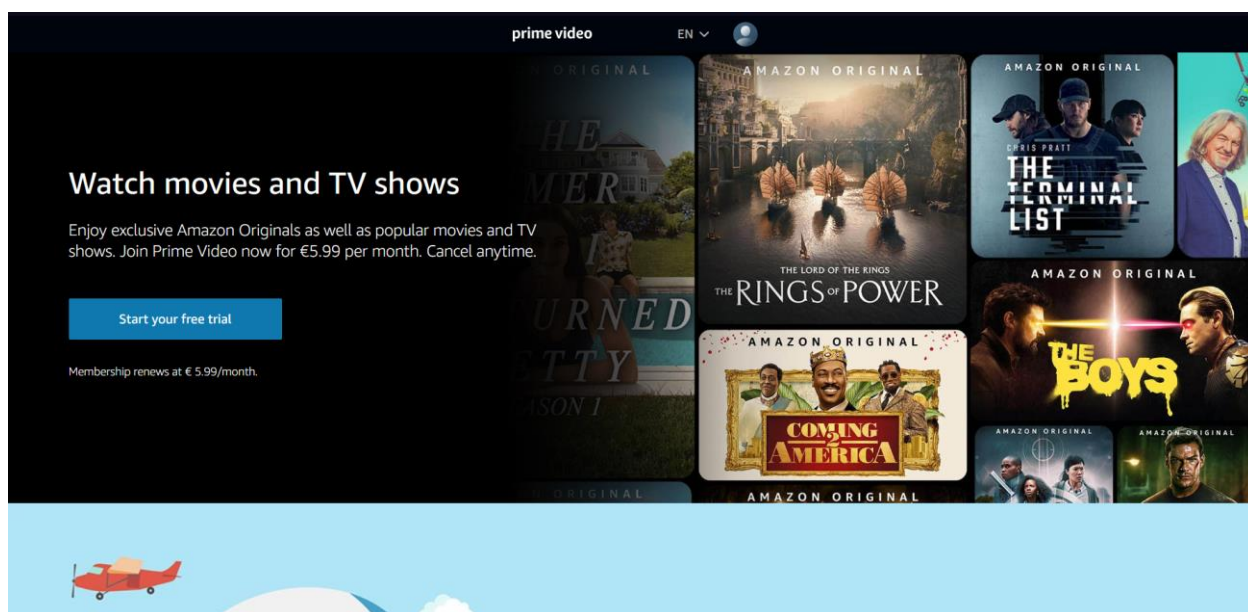


Рисунок 1.2 – Зовнішній вигляд сайту Amazon Prime Video [2]

Amazon Prime Video – це сервіс «відео за запитом», розроблений компанією Amazon.com [2]. Він пропонує телевізійні шоу та фільми в оренду або для купівлі як частину підписки Amazon Prime, деякі відео доступні для

перегляду виключно повним членам Prime або Prime Video. Як і конкуренти, Amazon прагне отримати ексклюзивний контент, щоб вирізнити свої послуги, включаючи багаторічну ліцензійну угоду з HBO в США.

Було проведено детальний огляд Amazon Prime Video як одного з провідних онлайн-кінотеатрів. Вивчено його основні функціональні можливості, включаючи стрімінг відео високої якості, доступ до широкого асортименту фільмів та серіалів, оригінального контенту та інтеграцію з іншими сервісами Amazon.

Проаналізовано популярність та успіх Amazon Prime Video в ринковому сегменті онлайн-кінотеатрів. Вивчено його аудиторію, рейтинги, відгуки користувачів та вплив на ринкову конкуренцію.

Також проведено аналіз основних функціональних можливостей Amazon Prime Video, таких як персоналізовані рекомендації, можливість завантаження контенту для перегляду офлайн, поділ підписки з родинними членами тощо. Визначено їх вплив на користувацький досвід та привабливість платформи для аудиторії.

Потенційний прибуток Amazon Prime Video та його бізнес модель також були детально вивчені (рис. 1.3).

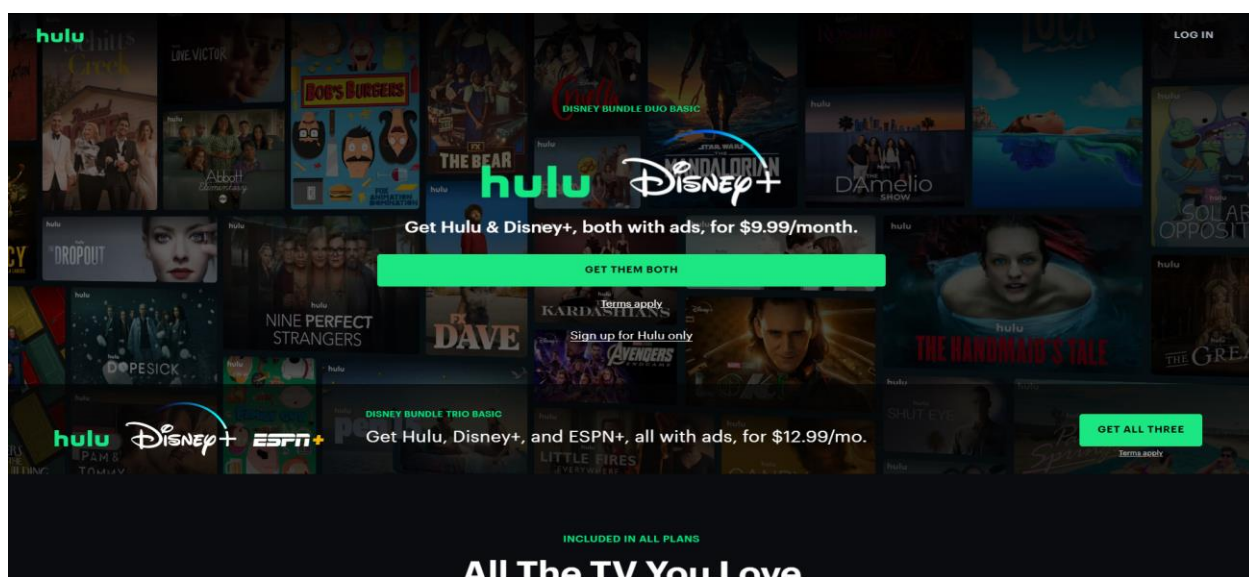


Рисунок 1.3 – Зовнішній вигляд сайту Hulu [3]

Hulu є одним з популярних онлайн кінотеатрів, який надає можливість віртуального перегляду фільмів та серіалів [3]. Hulu (укр. Гулу) – американський потоковий VOD-сервіс, що повністю контролюється і більшість акцій якої належить The Walt Disney Company, з NBCUniversal, що належить Comcast, як акціонерний капітал зацікавлених сторін. Давайте проведемо аналіз Hulu та його основних характеристик.

Hulu пропонує широкий вибір контенту, включаючи фільми, серіали, оригінальні програми, анімаційні шоу та документальні фільми. Вони співпрацюють з різними студіями та кінокомпаніями, що дозволяє їм мати в асортименті відомі та актуальні фільми та серіали.

Продукт також доступний на різних платформах, включаючи веб-сайт, мобільні пристрої (смартфони та планшети), телевізори, стрімінгові приставки та інші.

Платформа дає можливість обирати різні варіанти підписок, включаючи рекламні та безрекламні плани. Рекламні плани надають доступ до контенту за більш доступну ціну, але з включеною рекламою під час перегляду. Безрекламні плани, у свою чергу, забезпечують перегляд без реклами за вищу вартість.

Не обійшлося й без своїх оригінальних програм та серіалів, які можна переглядати виключно на їх платформі. Цей оригінальний контент додає варіативності та унікальності до їх пропозицій.

Hulu використовує алгоритми рекомендацій, щоб пропонувати користувачам контент, який може їх зацікавити. Вони аналізують перегляди, вподобання та інші дані, щоб підкреслити релевантні фільми та серіали для кожного користувача.

Доступні функції стрімінгу, що включають в себе можливість призупинити, перемотати або відновити перегляд, а також створювати власні списки перегляду.

Діють географічні обмеження послуги у Сполучених Штатах та обмеженій кількості інших країн. Це може бути обмеженням для користувачів з інших регіонів.

Провівши аналіз можна відзначити, що розробка власного онлайн кінотеатру є доцільною з кількох ключових причин:

- ринкові дослідження демонструють стрімке зростання попиту на онлайн кінотеатри та можливість віртуального перегляду фільмів. Цифровий контент стає дедалі популярнішим, а користувачі активно шукають зручні способи отримати доступ до різноманітного мультимедійного вмісту;

- онлайн кінотеатри забезпечують глобальний доступ до бібліотек фільмів і серіалів, не обмежений територіальним розподілом. Це відкриває нові можливості для залучення та обслуговування аудиторії з різних куточків світу, потенційно збільшуючи базу користувачів;

- віртуальні кінотеатри надають користувачам свободу переглядати улюблені фільми та шоу у будь-який зручний для них час і місце. Така гнучкість дозволяє адаптуватися до індивідуальних графіків та вимог сучасного способу життя.

Враховуючи ці фактори, розробка власного онлайн кінотеатру може стати перспективним проєктом, здатним задовольнити зростаючі потреби споживачів у зручному доступі до відео-контенту.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Для реалізації онлайн-бібліотеки фільмів необхідно вирішити наступні завдання:

- розробити функціональність для каталогізації та класифікації фільмів у бібліотеці за різними критеріями, такими як жанр, режисер, актор, рік випуску тощо;

- спроектувати структуру бази даних для зберігання інформації про фільми, включаючи назву, рік випуску, жанр, режисерів, акторів, синопсис;

- реалізувати потужну систему пошуку, яка дозволить користувачам знаходити фільми за назвою, акторами, режисерами або іншими атрибутами бази даних;
- впровадити систему відгуків, що дозволить користувачам залишати свої коментарі;
- інтегрувати функціональність для перегляду мультимедійного вмісту, такого як потокове передавання повних версій фільмів;
- розробити зручний та інтуїтивний інтерфейс користувача для навігації по бібліотеці фільмів, перегляду деталей фільму, пошуку та фільтрації результатів;
- забезпечити адміністративну панель для керування вмістом бібліотеки, додавання нових фільмів, редагування існуючих записів, модерації відгуків користувачів тощо.

Ці завдання охоплюють основні аспекти розробки онлайн-бібліотеки фільмів, включаючи проектування бази даних, реалізацію ключових функцій, інтеграцію мультимедійного вмісту, створення інтерфейсу користувача та забезпечення інструментів для адміністрування системи.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи бакалавра було проведено аналіз предметної області онлайн бібліотек фільмів та онлайн-кінотеатрів.

Ринок онлайн-кінотеатрів є надзвичайно потужним і генерує величезні прибутки. Великі компанії, такі як Netflix, Amazon та Disney, мають значні частки ринку в цій сфері. Був проведений аналіз аналогічних продуктів, таких як Netflix, Amazon Prime Video, Hulu, з метою виявлення їхніх спільних рис. Окреслено доцільність розробки власного онлайн-кінотеатру. Сформульовано основні завдання кваліфікаційної роботи бакалавра, спираючись на проведений аналіз.

У цьому розділі було детально розглянуто поточний стан ринку онлайн-кінотеатрів та онлайн-бібліотек фільмів. Дослідження показало, що ця галузь стрімко розвивається і має величезний потенціал завдяки зростаючому попиту на цифровий контент та зручність віддаленого перегляду.

Аналіз конкурентів дозволив виявити спільні риси успішних онлайн-кінотеатрів, такі як великі бібліотеки контенту, зручний інтерфейс користувача, персоналізовані рекомендації та підтримка різних платформ і пристроїв. Це допомогло окреслити ключові вимоги та функціональні можливості, які необхідно врахувати під час розробки власного онлайн-кінотеатру.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

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

Детально сформулюємо вимоги до системи, спираючись на завдання кваліфікаційної роботи, наведені в попередньому розділі.

Проілюструємо основні функціональні вимоги у вигляді UML- діаграми прецедентів (рис. 2.1).

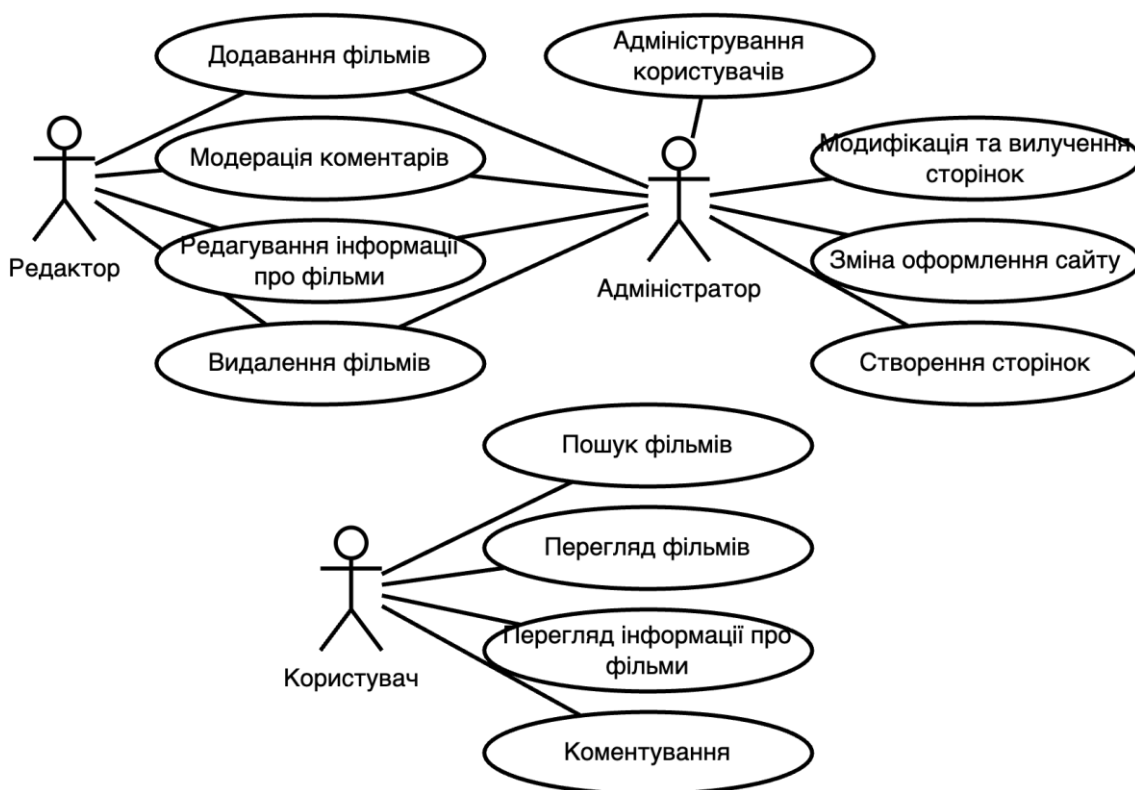


Рисунок 2.1 – Діаграма прецедентів

Ця діаграма зображує основні функціональні можливості та процеси, пов'язані з адмініструванням та використанням онлайн-бібліотеки фільмів. При цьому користувачі поділено на три ролі.

Функції, що стосуються керування контентом бібліотеки з боку редакторів включають додавання нових фільмів, редагування інформації про наявні фільми, модерацію коментарів користувачів та видалення фільмів.

Адміністратори мають всі права редакторів, а також здійснюють адміністрування користувачів, модифікацію та вилучення сторінок сайту, зміну оформлення сайту та створення нових сторінок. Звичайні користувачі можуть здійснювати пошук фільмів за різними критеріями, переглядати фільми та інформацію про них, залишати коментарі, а також переглядати та оцінювати коментарі інших користувачів.

Сформулюємо функціональні вимоги до системи:

- початкове введення інформації про фільми в базу даних, включаючи назву, рік випуску, жанр, режисерів, акторів, синопсис та інші деталі;
- зміна або редагування існуючих даних про фільми в базі даних;
- видалення застарілих або непотрібних записів про фільми;
- здійснення пошуку фільмів за запитом користувача за різними критеріями, такими як назва, актори, режисери, жанр тощо;
- забезпечення віддаленого доступу до системи через протокол TCP/IP для користувачів та адміністраторів;
- реалізація заходів безпеки та захисту даних, включаючи автентифікацію користувачів, шифрування даних та контроль доступу;
- розмежування прав доступу для різних типів користувачів, таких як адміністратори, звичайні користувачі та редактори;
- контроль введеної інформації для забезпечення її коректності та цілісності;
- забезпечення цілісності бази даних, включаючи механізми резервного копіювання та відновлення даних;
- виведення знайденої інформації про фільми в зручному для користувача форматі, включаючи деталі фільму, відгуки користувачів та можливість перегляду мультимедійного вмісту.

Інтерфейс користувача для онлайн-бібліотеки фільмів має бути зручним, інтуїтивним та естетично привабливим. Основні елементи інтерфейсу включатимуть:

- головну сторінку з популярними фільмами, новими надходженнями та рекомендаціями на основі історії перегляду користувача;
- сторінку пошуку з різними фільтрами та критеріями пошуку, такими як жанр, рік випуску, актори, режисери тощо;
- сторінку деталей фільму з інформацією про назву, рік випуску, жанр, режисерів, акторів, синопсис, відгуки користувачів та можливістю переглянути фільм;
- адміністративну панель для керування вмістом бібліотеки, додавання нових фільмів, редагування існуючих записів та модерації відгуків користувачів.

Головне меню сайту повинно включати наступні пункти (рис. 2.2).

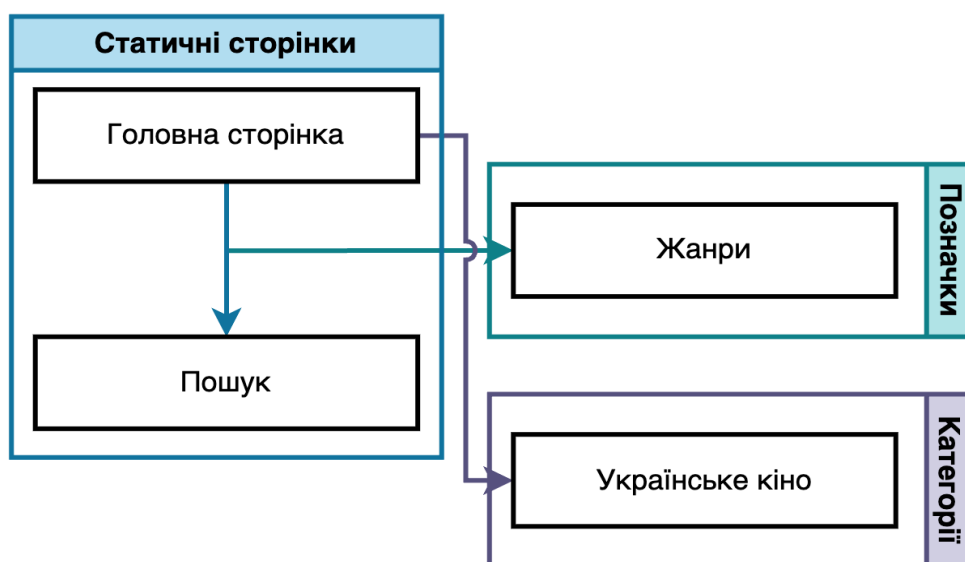


Рисунок 2.2 – Головне меню сайту

Оскільки тема кваліфікаційної роботи передбачає використання CMS WordPress, то доречно для реалізації навігації скористатись стандартними таксономіями, такими як категорії та позначки.

Статичні сторінки в меню:

- головна сторінка – основна сторінка сайту, яка відображає останні фільми, новинки тощо;
- пошук – сторінка з формою для пошуку фільмів за різними критеріями;

– жанри – це категорії WordPress, які використовуються для групування фільмів за жанрами. Кожен фільм може належати до одного або кількох жанрів. Це дозволяє користувачам легко знаходити фільми за їхніми уподобаннями;

– українське кіно – це позначка WordPress, яка використовується для виділення фільмів українського виробництва. Фільми з цією позначкою будуть відображатися на окремій сторінці, присвячений українському кінематографу.

Використання стандартних таксономій WordPress, таких як категорії та позначки, дозволяє ефективно організувати та структурувати контент сайту.

Розглянемо макети основних сторінок сайту (рис. 2.3).

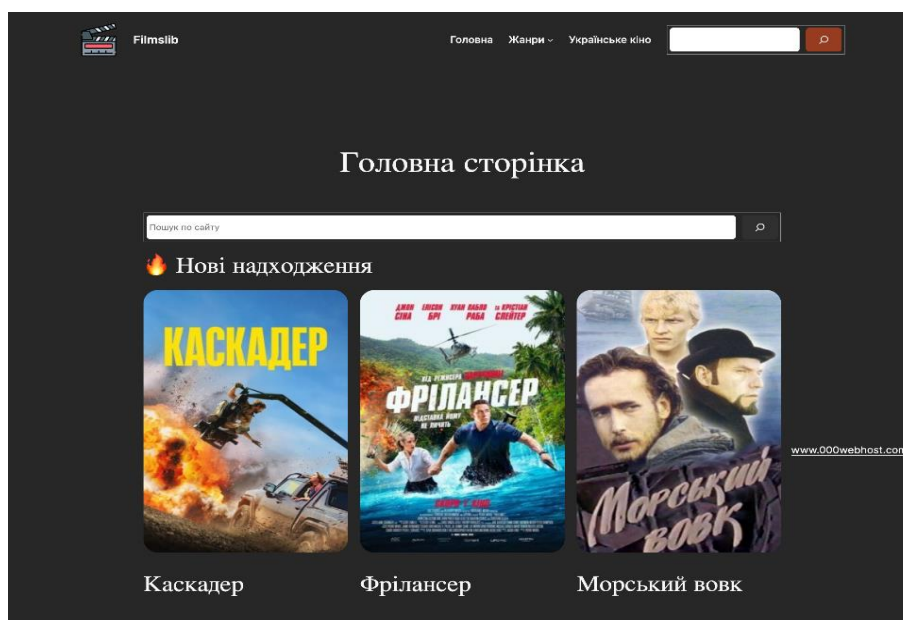


Рисунок 2.3 – Фрагмент макету головної сторінки

Макет представляє головну сторінку онлайн-бібліотеки фільмів під назвою «Filmslib». Макет має чітку структуру та зручну навігацію, що дозволяє користувачам швидко знаходити потрібну інформацію.

У верхній частині сторінки розташовано головне меню з розділами «Головна», «Жанри», «Українське кіно» та полем пошуку. Це дозволяє користувачам переходити до потрібних розділів сайту та здійснювати пошук фільмів.

Основний контент сторінки поділено на три секції: «Нові надходження», «Фільми 2024 року» та «Українські фільми».

У секції «Нові надходження» представлені останні додані фільми.. Кожен фільм супроводжується відповідним постером.

Секція «Фільми 2024 року» містить інформацію про фільми поточного року. У секції «Українські фільми» розміщено фільми українського виробництва на основі заданої категорії.

Макет покликаний надати користувачу зручний та привабливий дизайн онлайн-бібліотеки фільмів, який дозволяє легко орієнтуватися та знаходити цікаві фільми. Макет також приділяє увагу українському кінематографу, виділяючи його в окрему секцію.

Розглянемо макет сторінки, яка містить інформацію про конкретний фільм, та забезпечує його перегляд. Даний макет є спільним для всіх сторінок з фільмами (рис. 2.4).

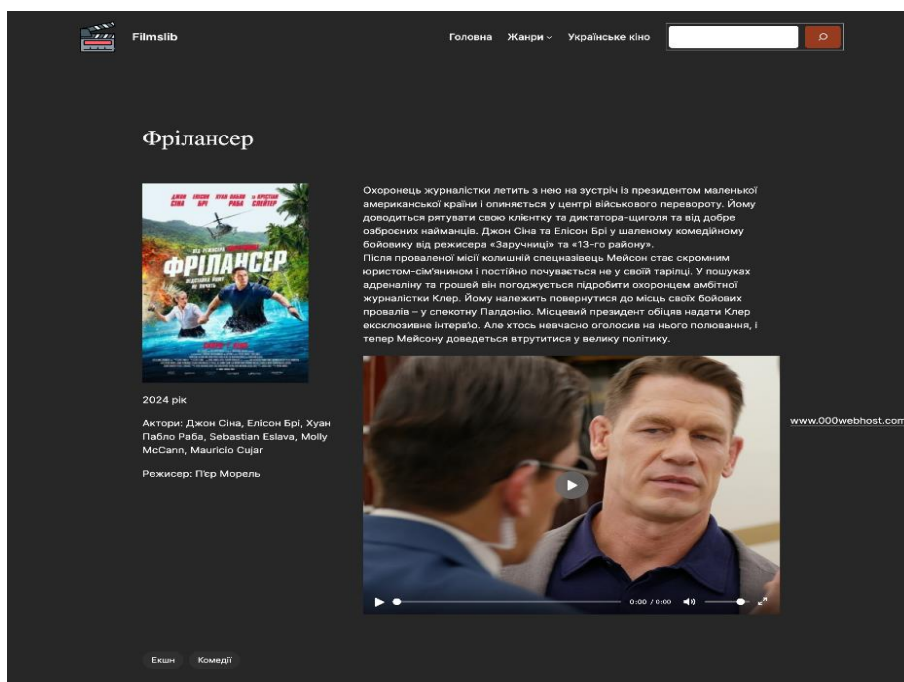


Рисунок 2.4 – Фрагмент макету сторінки з фільмом

У верхній частині сторінки розташовано головне меню. Основний контент сторінки складається з двох основних частин: інформації про фільм та

розділу для коментарів. У розділі з інформацією про фільм представлено постер фільму, а також детальний опис сюжету. Також зазначено жанр фільму, рік випуску та режисера. Справа відображається медіаплеєр для перегляду фільму.

Нижче розміщено розділ для коментарів, де користувачі можуть залишити свої відгуки про фільм. Дизайн сторінки є зручним та інтуїтивно зрозумілим, використовує темну тему, що покращує досвід користувачів під час перегляду інформації про фільм.

Важливо також забезпечити адаптивність дизайну сторінок, для того, щоб користувачі могли зручно взаємодіяти з сайтом із різних пристроїв. В першу чергу це стосується розробки адаптивного меню, форми пошуку та засобів для перегляду фільму, тобто елементів, з якими користувач взаємодіє безпосередньо.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для забезпечення необхідного функціоналу та зручного адміністрування системи було обрано систему керування вмістом Wordpress.

WordPress – це потужна система керування вмістом (CMS), яка широко використовується для створення та управління веб-сайтами, блогами, електронними магазинами та іншими типами веб-проектів [4]. Розроблена на мові програмування PHP та використовуючи базу даних MySQL, WordPress пропонує гнучку та масштабовану платформу, що поєднує простоту використання та розширені можливості для розробників.

Одна з головних переваг WordPress – це її зручний та інтуїтивний інтерфейс адміністратора. Він дозволяє легко створювати, редагувати та публікувати контент, керувати користувачами та налаштуваннями сайту без необхідності глибоких технічних знань. Завдяки вбудованій системі тем та плагінів, користувачі можуть персоналізувати зовнішній вигляд та функціональність свого сайту відповідно до своїх потреб.

Екосистема WordPress надзвичайно багата та різноманітна. Офіційний каталог плагінів містить понад 59 000 безкоштовних плагінів, які розширюють можливості WordPress у різних сферах, таких як електронна комерція, SEO, безпека, галереї та мультимедіа, соціальні мережі, аналітика та багато іншого. Користувачі також можуть створювати власні плагіни, використовуючи потужну систему хуків (hooks) та фільтрів WordPress.

Крім того, WordPress пропонує величезний вибір тем оформлення, як безкоштовних, так і преміум. Темі дозволяють змінювати зовнішній вигляд сайту, його структуру та функціональність, не торкаючись базового коду WordPress. Це забезпечує гнучкість та можливість легко оновлювати та підтримувати сайт. Розробники також можуть створювати власні теми, використовуючи стандартні шаблони WordPress та налаштовуючи їх за допомогою PHP, HTML, CSS та JavaScript.

WordPress має потужну та активну спільноту розробників, які постійно вдосконалюють та розширюють можливості платформи. Ядро WordPress регулярно оновлюється, забезпечуючи покращення продуктивності, безпеки та нові функції. Завдяки відкритому вихідному коду, будь-хто може долучитися до розвитку WordPress та внести свій внесок у проект.

Одним з найважливіших аспектів WordPress є її SEO-дружність. Платформа автоматично генерує семантичні URL, підтримує мета-теги, заголовки та налаштування сайтмапів, що полегшує індексацію сайту пошуковими системами. Крім того, доступні численні плагіни для розширеної SEO-оптимізації, такі як Yoast SEO, All in One SEO Pack та інші.

WordPress також пропонує потужні інструменти для багатомовної підтримки, що робить її ідеальною платформою для створення сайтів, доступних на кількох мовах. Плагіни, такі як WPML та Polylang, дозволяють легко переключатися між мовами, перекладати контент та керувати перекладами.

Незважаючи на всі свої переваги, WordPress не позбавлений певних недоліків. Одним з них є питання безпеки, оскільки неправильна конфігурація

або застарілі плагіни та теми можуть зробити сайт вразливим для зловмисників. Однак, регулярне оновлення WordPress, використання надійних плагінів та застосування належних заходів безпеки можуть значно зменшити ці ризики. Ще одним потенційним недоліком є продуктивність WordPress при обробці великих обсягів трафіку або складного контенту. Однак, використання кешування, оптимізації бази даних та розподілених серверних архітектур може допомогти вирішити ці проблеми.

Загалом, WordPress є потужною та гнучкою платформою, яка ідеально підходить для створення різноманітних веб-проектів, від особистих блогів до корпоративних веб-сайтів та навіть електронних магазинів [5]. Її простота використання, розширювані можливості та активна спільнота роблять WordPress одним з найпопулярніших рішень для керування вмістом в Інтернеті.

З точки зору моделі програмного забезпечення дану систему реалізовано з використанням моделі клієнт-сервер.

Модель клієнт-сервер є оптимальним вибором для розробки сайту для перегляду фільмів на WordPress, оскільки вона забезпечує чітке розмежування функціональності та відповідальності між клієнтською та серверною частинами системи. Ця модель дозволяє ефективно розподіляти навантаження та ресурси, а також досягати високої масштабованості та гнучкості. З одного боку, клієнтська частина, реалізована у вигляді веб-інтерфейсу, відповідає за зручну взаємодію з користувачем. Вона забезпечує інтуїтивний спосіб перегляду каталогу фільмів, вибору бажаного контенту та керування відтворенням відео. Завдяки гнучкості веб-технологій, користувачі можуть отримувати доступ до контенту з різних пристроїв та місць, забезпечуючи максимальну доступність сервісу. З іншого боку, серверна частина відіграє ключову роль у забезпеченні належного функціонування системи. Вона відповідає за обробку запитів, управління даними та забезпечення необхідної бізнес-логіки. У контексті сайту для перегляду фільмів, сервер зберігає та обробляє метадані фільмів, керує користувацькими даними, забезпечує авторизацію та автентифікацію, а також здійснює потокову передачу відео для відтворення на клієнтських пристроях.

Централізоване розміщення бізнес-логіки та даних на сервері забезпечує консистентність та цілісність інформації для всіх користувачів системи. Ще однією перевагою моделі клієнт-сервер є можливість масштабування системи відповідно до зростаючих потреб та навантаження. Серверну частину можна розгорнути на потужних серверах чи хмарних платформах, що дозволяє задовольнити вимоги до продуктивності та обробки великої кількості користувацьких запитів. Водночас, клієнтські додатки можуть бути оптимізовані для різних пристроїв та платформ, забезпечуючи максимальну доступність та зручність використання для користувачів.

Враховуючи специфіку можливостей Wordpress подамо загальну архітектуру системи у вигляді контекстної діаграми в нотації C4 зображено на рисунку 2.5 [6].

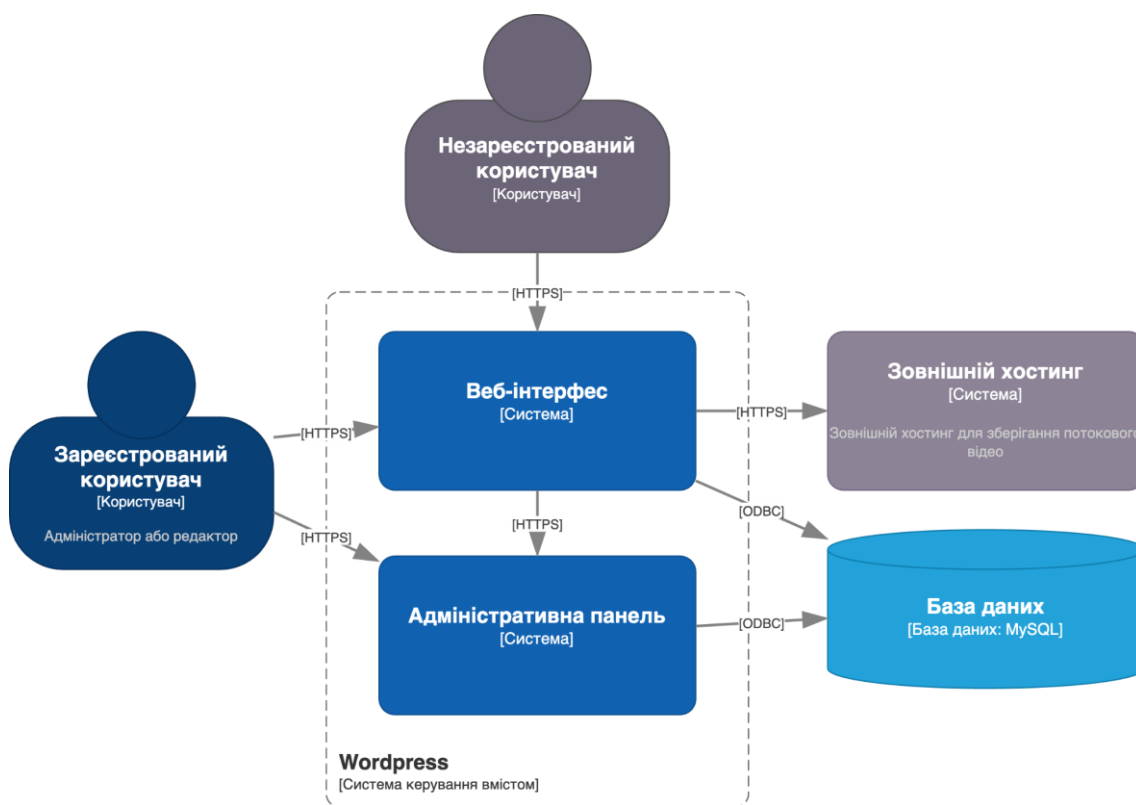


Рисунок 2.5 – Контекстна діаграма (нотація C4)

Діаграма ілюструє основні компоненти веб-системи бібліотеки фільмів, їх взаємозв'язки та способи взаємодії між клієнтами (користувачами), серверною

частиною (веб-інтерфейс та адміністративна панель), базою даних та зовнішнім хостингом для потокового відео.

У центрі діаграми розташовані два основні компоненти системи: «Веб-інтерфейс» та «Адміністративна панель». Веб-інтерфейс є клієнтською частиною системи, доступною для зареєстрованих користувачів через захищений протокол HTTPS. Адміністративна панель використовується адміністраторами або редакторами для керування вмістом та налаштуваннями бібліотеки.

Незареєстровані користувачі взаємодіють з веб-інтерфейсом через HTTPS для перегляду, пошуку та коментування фільмів.

База даних реалізована з використанням системи керування базами даних MySQL, є сховищем для зберігання інформації про фільми, користувачів та інших даних системи. База даних взаємодіє з адміністративною панеллю та веб-інтерфейсом через з'єднання ODBC.

Зовнішній хостинг представляє зовнішній сервер для зберігання потокового відео. Він взаємодіє з веб-інтерфейсом через HTTPS для передачі потокового відео фільмів. Враховуючи великі обсяги даних, які необхідно зберігати при роботі з потоковим відео, використання зовнішнього хостингу є необхідним.

Висновки до розділу 2

У даному розділі було проведено аналіз та визначено вимоги до розроблюваного програмного забезпечення онлайн-бібліотеки фільмів. Основні функціональні вимоги були представлені у вигляді UML-діаграми прецедентів, яка відображає взаємодію користувачів з системою залежно від їхніх ролей (адміністратор, редактор, звичайний користувач).

Було сформульовано детальні функціональні вимоги до системи, включаючи початкове введення та редагування інформації про фільми, пошук фільмів за різними критеріями, забезпечення віддаленого доступу, реалізацію

заходів безпеки та захисту даних, розмежування прав доступу, контроль введеної інформації, забезпечення цілісності бази даних та виведення знайденої інформації у зручному форматі.

Для реалізації обрано систему керування вмістом WordPress, яка надає потужні можливості для створення та управління веб-сайтами. Розглянуто переваги WordPress, такі як зручний інтерфейс адміністратора, широкий вибір плагінів та тем, потужна спільнота розробників та SEO-дружність.

РОЗДІЛ 3

РОЗРОБКА ОНЛАЙН-БІБЛІОТЕКИ ФІЛЬМІВ

3.1 Практична реалізація об'єкта проектування

Основна функціональність сайту реалізована засобами системи керування контентом Wordpress. Розглянемо ключові фрагменти програмного коду, які є критичними для успішного функціонування системи.

Основний програмний код написаний на PHP. PHP – це серверна мова програмування з відкритим вихідним кодом, яка широко використовується для розробки веб-додатків. PHP дозволяє розробникам створювати динамічні веб-сайти та веб-додатки, які можуть взаємодіяти з базами даних, обробляти форми, генерувати динамічний контент та забезпечувати різноманітну функціональність. Однією з головних переваг PHP є її легкість у вивченні та використанні, що робить її доступною як для початківців, так і для досвідчених розробників.

Код PHP вбудовується безпосередньо в HTML-файли, що дозволяє легко комбінувати статичний і динамічний контент на одній сторінці. Коли веб-сервер отримує запит на сторінку з PHP-кодом, він виконує код на сервері та генерує HTML, який потім відправляється до клієнта (зазвичай веб-браузера). Це робить PHP ідеальним вибором для розробки веб-додатків, де необхідно динамічно генерувати контент на основі даних з бази даних, параметрів користувача або інших факторів.

PHP тісно інтегрований з базами даних, такими як MySQL, PostgreSQL, Oracle і Microsoft SQL Server. Ця інтеграція дозволяє розробникам легко взаємодіяти з базами даних, виконувати запити, зберігати та вилучати дані. Крім того, PHP підтримує різноманітні протоколи, такі як LDAP, IMAP, SNMP, NNTP та інші, що робить його придатним для різних типів додатків.

Однією з найбільших переваг PHP є наявність величезної кількості бібліотек та фреймворків, які значно полегшують розробку веб-додатків. Деякі з найпопулярніших PHP-фреймворків включають Laravel, Symfony, CodeIgniter

та Yii. Ці фреймворки забезпечують готові рішення для багатьох типових задач, таких як маршрутизація, аутентифікація, робота з базами даних та багато іншого.

PHP також має потужну та активну спільноту розробників, яка постійно вдосконалює мову та створює нові бібліотеки та інструменти. Завдяки цій спільноті, PHP продовжує розвиватися та адаптуватися до нових тенденцій та вимог веб-розробки.

Незважаючи на свою популярність, PHP часто критикується за певні недоліки. Одним з них є проблема безпеки, оскільки багато старих проєктів на PHP можуть бути вразливими через використання застарілих практик або невідповідних бібліотек. Однак, сучасні версії PHP та фреймворки значно покращили безпеку, пропонуючи вбудовані механізми захисту, такі як фільтрація вхідних даних, шифрування та захист від вразливостей, пов'язаних з ін'єкціями SQL.

Розглянемо реалізацію функції WP_Query для запитів до бази даних. Дана функція є ключовою для виведення контенту на сторінках за заданими параметрами. Саме за допомогою неї виводиться загальна інформація про фільми, формується головна сторінка. У лістингу 3.1 розглянуто роботу функції при виведенні останніх доданих фільмів.

Лістинг 3.1 – Функція WP_Query

```
<?php
$args = array(
    'post_type' => 'post',
    'category_name' => 'featured',
    'posts_per_page' => 3,
    'orderby' => 'date',
    'order' => 'DESC'
);

$query = new WP_Query( $args );

if ( $query->have_posts() ) :
    while ( $query->have_posts() ) :
        $query->the_post();
        // Відображення вмісту запису (посту)
        the_title();
```

```

        the_excerpt();
    endwhile;
    wp_reset_postdata();
else :
    // Відображення повідомлення, якщо записів не знайдено
    _e( 'Sorry, no posts matched your criteria.', 'textdomain' );
endif;
?>

```

Кінець лістингу 3.1

Функція `WP_Query` дозволяє створювати користувацькі запити до бази даних WordPress. Вона приймає масив аргументів `$args`, який визначає параметри запиту.

У наведеному прикладі використовуються такі аргументи:

- `'post_type' => 'post'` – вказує тип запису (post type), який потрібно отримати. У цьому випадку це звичайні записи (пости);
- `'category_name' => 'featured'` – вказує назву категорії, до якої повинні належати записи;
- `'posts_per_page' => 3` – визначає кількість записів, які потрібно отримати (в даному випадку 3);
- `'orderby' => 'date'` – визначає, за яким полем сортувати записи (в даному випадку за датою);
- `'order' => 'DESC'` – вказує порядок сортування (в даному випадку за спаданням).

Після створення об'єкту `WP_Query` з потрібними аргументами, ми перевіряємо, чи є записи, використовуючи метод `have_posts()`. Якщо записи знайдені, ми проходимо по кожному запису за допомогою циклу `while` та методу `the_post()`, який встановлює поточний запис для подальшої обробки. Всередині циклу ми можемо використовувати різні функції шаблону, такі як `the_title()` та `the_excerpt()`, для відображення вмісту поточного запису. Після завершення циклу використовується функція `wp_reset_postdata()` для скидання глобальних змінних WordPress до їх початкового стану. Якщо записів не знайдено, відображається повідомлення про відсутність результатів.

UML-діаграма діяльності функції WP_Query наведена на рисунку 3.1.

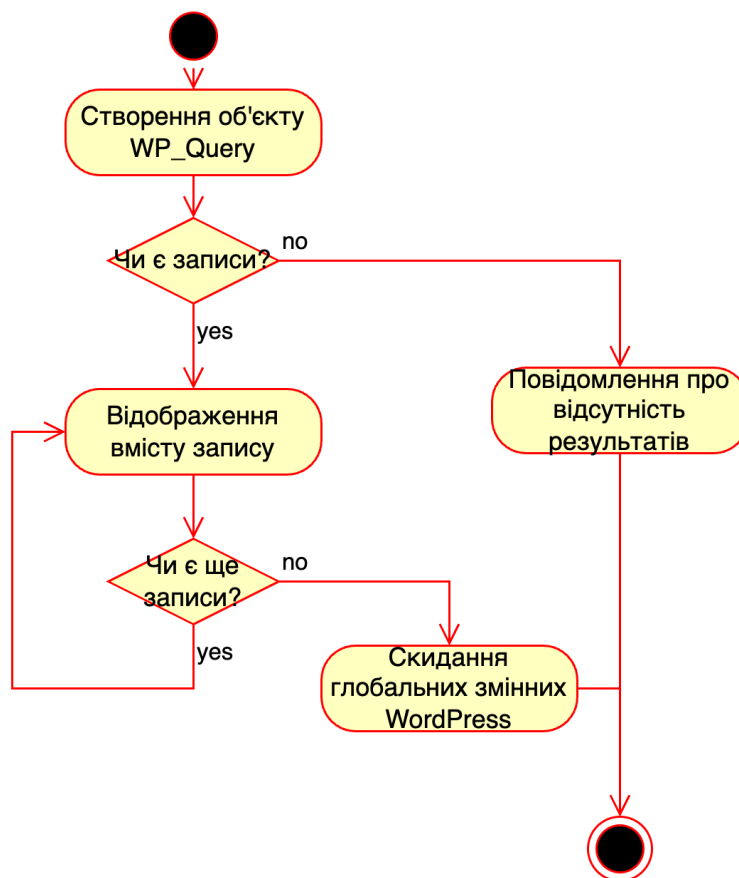


Рисунок 3.1 – UML-діаграма діяльності функції WP_Query

Розглянемо реалізацію функції пошуку. Для цього використовується комбінація функцій та шаблонів WordPress. В шаблоні створюється пошукова форма, зображена в лістингу 3.2.

Лістинг 3.2 – Створення форми пошуку в шаблоні WordPress

```

<form role="search" method="get" class="search-form" action="php
echo esc_url( home_url( '/' ) ); ?&gt;"&gt;    &lt;label&gt;
    &lt;span class="screen-reader-text"&gt;&lt;?php _x( 'Search for:',
'label', 'textdomain' ); ?&gt;&lt;/span&gt;
    &lt;input
        type="search"
        class="search-field"
placeholder="<?php echo esc_attr_x( 'Search &amp;hellip;',
'placeholder', 'textdomain' ); ?&gt;" value="<?php echo
get_search_query(); ?&gt;" name="s" /&gt;    &lt;/label&gt;
    &lt;button type="submit" class="search-submit"&gt;&lt;?php echo _x(
'Search', 'submit button', 'textdomain' ); ?&gt;&lt;/button&gt;
&lt;/form&gt;
  </pre

```

Кінець лістингу 3.2

Ця форма відправляє GET-запит з параметром `s`, який містить пошуковий запит користувача. WordPress автоматично обробляє пошуковий запит, коли параметр `s` присутній в URL. Він використовує значення цього параметра для виконання пошуку по вмісту сайту (записи, сторінки, custom post types тощо).

Для відображення результатів пошуку використовується шаблон `search.php` зображений в лістингу 3.3. Цей шаблон виконує цикл для виведення знайдених записів.

Лістинг 3.3 – Шаблон `search.php`

```
<?php
if ( have_posts() ) :
    while ( have_posts() ) :
        the_post();
        the_title();
        the_excerpt();
    endwhile;
    the_posts_pagination();
else :
    _e( 'Sorry, no posts matched your search criteria.',
    'textdomain' );
endif;
?>
```

Кінець лістингу 3.3

У цьому прикладі ми використовуємо стандартний цикл для виведення знайдених записів. Функції `the_title()` та `the_excerpt()` використовуються для відображення заголовку та уривку кожного запису. Функція `the_posts_pagination()` виводить пагінацію для навігації по сторінках результатів пошуку.

UML-діаграма діяльності шаблону `search.php` наведена на рисунку 3.2.

Для перегляду фільмів на сторінки вбудовується медіаплеєр, в якому у якості параметру передається посилання на відеопотік. Проте Wordpress не має функціоналу для вставки вбудованих елементів, тому необхідно виконати модифікації файлу `functions.php`. Функція `add_shortcode()`, яка наведена в лістингу 3.4 дозволяє створити шорткод для подальшої вставки відео в сторінку.

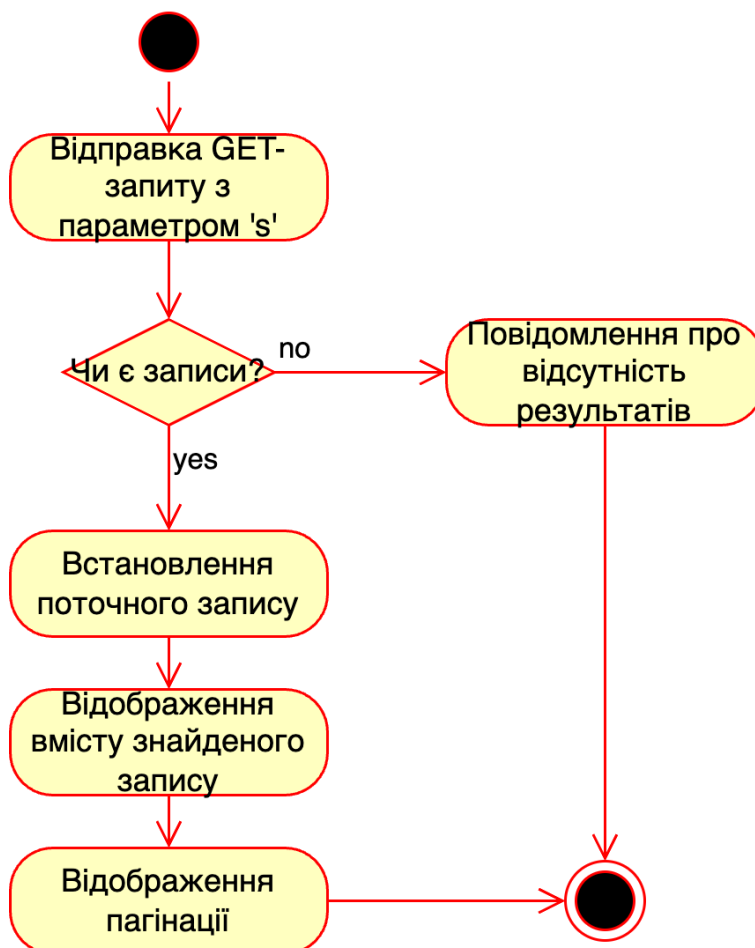


Рисунок 3.2 – UML-діаграма діяльності шаблону search.php

Лістинг 3.4 – Функція генерації шорткоду для вставки відео

```

function add_shortcode( $atts ) {
    $defaults = array(
        'width' => '100%',
        'height' => '500',
        'scrolling' => 'yes',
        'class' => 'iframe-class',
        'frameborder' => '0'
    );

    if ( ! is_array( $atts ) ) {
        $atts = array();
    }

    foreach ( $defaults as $default => $value ) { // add defaults
        if ( ! @array_key_exists( $default, $atts ) ) {
            $atts[$default] = $value;
        }
    }
    $html .= '<iframe';
    foreach( $atts as $attr => $value )
    {

```

```

        if ( strtolower($attr) == 'src' )
    {
        $value = esc_url( $value );
    }
    if ( strtolower($attr) == 'srcdoc' )
    {
        continue;
    }
    if ( strpos( strtolower( $attr ), 'on' ) === 0 )
    {
        continue;
    }

    if ($value !== '') { // adding all attributes
        $html .= ' ' . esc_attr($attr) . '="' .
esc_attr($value) . '"';
    } else { // adding empty attributes
        $html .= ' ' . esc_attr($attr);
    }
}
$html .= '></iframe>'. "\n";

return $html;
}

```

Кінець лістингу 3.4

Функція приймає параметр `$atts`, який є масивом атрибутів шорткоду. Спочатку функція визначає масив `$defaults`, який містить значення за замовчуванням для атрибутів `iframe`, таких як ширина, висота, прокрутка, клас та рамка. Якщо `$atts` не є масивом, функція створює порожній масив `$atts`. Функція проходить по масиву `$defaults` і додає значення за замовчуванням до `$atts` для тих атрибутів, які не були вказані в шорткоді. Далі функція починає формувати HTML-код `iframe`. Вона проходить по масиву `$atts` і додає кожен атрибут до тегу `<iframe>`. Якщо атрибут – це `src`, функція використовує `esc_url()` для санітизації URL. Якщо атрибут – це `srcdoc`, функція пропускає його. Також функція пропускає атрибути, які починаються з `on` (наприклад, «`onload`», «`onclick`» тощо). В кінці функція повертає сформований HTML-код `iframe`.

UML-діаграма діяльності функції `add_shortcode` наведена на рисунку 3.3.

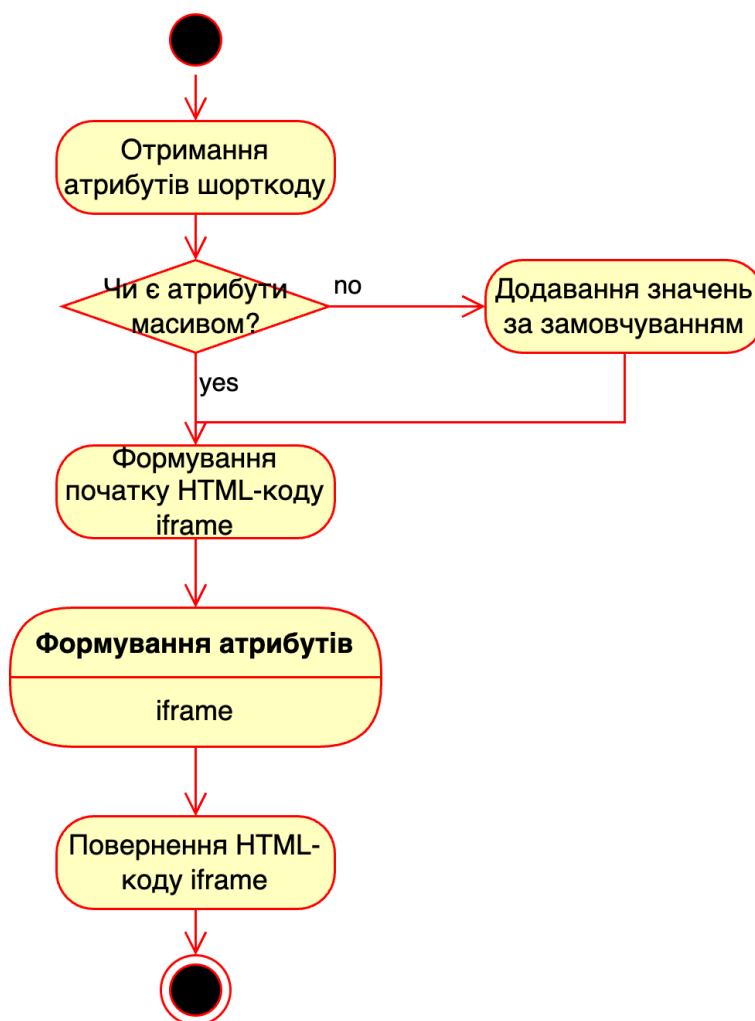


Рисунок 3.3 – UML-діаграма діяльності функції add_shortcode

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Виконаємо тестування за допомогою тест-кейсів. Використання тест-кейсів є ефективним способом тестування веб-сайту, оскільки дозволяє систематично перевірити всі важливі функції та сценарії використання. Це допомагає забезпечити якість, надійність та зручність використання веб-сайту для кінцевих користувачів.

Тестування за допомогою тест-кейсів (test case testing) – це методика тестування програмного забезпечення, яка полягає у створенні набору тестових випадків (тест-кейсів) для перевірки функціональності, надійності та продуктивності системи [7]. Тест-кейси – це документи, які описують кроки,

необхідні для перевірки конкретної функції або вимоги до системи, очікувані результати та критерії успішності тесту.

Методика тестування за допомогою тест-кейсів включає такі етапи:

- вивчення функціональних та нефункціональних вимог до системи, визначення області тестування;
- розробка тестових випадків на основі вимог, враховуючи різні сценарії використання, граничні значення, потенційні помилки тощо;
- проведення тестування системи згідно з розробленими тест-кейсами, фіксація результатів та порівняння з очікуваними;
- у разі виявлення невідповідностей або помилок, створення звітів про дефекти з детальним описом проблеми та кроками для її відтворення;
- після виправлення дефектів, проведення повторного тестування для перевірки коректності виправлень.

Для тестування сайту розробимо та виконаємо тест-кейси найважливіших функцій сайту, таких як доступність, функціональність пошуку та відтворення відео. Результати тестування опишемо нижче.

Коротко опишемо перевірку працездатності сайту. Для перевірки статусу, який використовує наступні функції Pass / Fail / Skipped, отримуємо в результаті Pass. Для тестування передумов потрібен доступ до браузера. Далі вводимо крок 1, де потрібно ввести наступне в адресному рядку: <https://filmslib.000webhostapp.com/>. В очікуваних результатах користувач бачить сайту. У фактичних результатах відображається «Як і очікувалось».

Тепер опишемо тест-кейс перевірки працездатності пошуку на сайті. Для цього статус використовує наступні кейси: Pass / Fail / Skipped, де в тестуванні функціоналу пошуку на сайті маємо отримати результат «Pass». Передумовами цього будуть наступні вимоги: доступ до браузера, і на головній сторінці сайту відразу знаходиться користувач. Опишемо деталі кроків: 1) встановимо курсор в поле пошуку, 2) введемо пошуковий запит «Комедії», 3) натиснемо кнопку «Шукати». В очікуваних результатах плануємо бачити: «пошукове поле отримує фокус», наступне «пошуковий запит введено в поле», далі «користувач

бачить результати». У фактичних результатах отримуємо «Як і очікувалось». У статусі отримуємо «Pass».

Наступним кроком є тест-кейс перевірки працездатності відтворення відео. В передумовах повинні отримувати доступ до браузера, а також користувач буде знаходитися на головній сторінці сайту. Опишемо деталі кроку: у крок 1 клацаємо по обраному фільтру, де в очікуваних результатах маємо отримати «сторінку з фільмом»; у кроці 2 натискаємо «відтворити», де в очікуваних результатах маємо отримати «фільм починає відтворюватися». У фактичних результатах отримаємо «Як і очікувалося». У статусі в обох випадках отримуємо «Pass».

3.3 Захист інформаційно-комп'ютерної системи

Для захисту інсталяції сайту необхідно вжити ряд заходів, зокрема налаштування дозволів на директорії та захист конфігураційного файлу `wp-config.php`. Для цього на веб-сервері Apache для файлу `wp-config.php` встановлено дозвіл 640 (читання для всіх, запис лише для власника і групи), на директорії дозвіл 755 (читання, запис для власника, читання та виконання для групи та всіх).

Для захисту `wp-config.php` від прямого доступу створюється файл `.htaccess` у кореневій директорії WordPress з вмістом, продемонстрованим в лістингу 3.5.

Лістинг 3.5 – Файл `.htaccess`

```
<Files wp-config.php>
  Order Allow,Deny
  Deny from all
</Files>
```

Кінець лістингу 3.5

Для захисту WordPress вжито наступні заходи:

- використовуються складні, унікальні паролі для адміністраторів та інших облікових записів WordPress;
- регулярне оновлення WordPress, плагінів та тем до останніх версій для виправлення вразливостей;
- двофакторна автентифікація для адміністраторів та інших важливих облікових записів;
- SSL-сертифікат для захисту з'єднання між веб-сервером і користувачами.

У WordPress використовуються кілька алгоритмів для генерування паролів, хешування та солі безпеки. Ці алгоритми допомагають забезпечити належний рівень захисту даних користувачів, зокрема їхніх паролів та інформації про автентифікацію. WordPress використовує функцію `wp_generate_password()` для створення випадкових паролів. Ця функція базується на `wp_rand()`, яка є обгорткою для `random_bytes()` з PHP 7.0+ або `openssl_random_pseudo_bytes()` для старіших версій PHP. Ці функції забезпечують надійне генерування випадкових даних, використовуючи криптографічно безпечний генератор псевдовипадкових чисел.

Для хешування паролів WordPress використовує алгоритм хешування на основі пам'яті `bcrypt`. Цей алгоритм є стійким до атак за словником і атак з підбором паролів, оскільки він дуже повільний і вимагає значних обчислювальних ресурсів. WordPress використовує функцію `wp_hash_password()`, яка є обгорткою для `password_hash()` з PHP 5.5+, або `wp_password_bcrypt()` для старіших версій PHP.

Солі безпеки використовуються для підвищення стійкості хешів паролів до атак за словником і райдужних таблиць. WordPress генерує унікальні солі безпеки під час встановлення за допомогою функції `wp_salt()`, яка використовує `wp_rand()` для створення випадкових рядків. Ці солі зберігаються у файлі `wp-config.php` як константи: `AUTH_KEY`, `SECURE_AUTH_KEY`, `LOGGED_IN_KEY`, `NONCE_KEY`, `AUTH_SALT`, `SECURE_AUTH_SALT`, `LOGGED_IN_SALT`, `NONCE_SALT`.

Для хешування ключів автентифікації, таких як cookie-ключі та ключі сесії, WordPress використовує алгоритм хешування HMAC з використанням алгоритму хешування SHA-256. Це забезпечує захист від підробки та зміни цих ключів.

Детальніше розглянемо роботу алгоритму хешування HMAC.

HMAC (Hash-based Message Authentication Code) – це алгоритм автентифікації повідомлень, який використовує криптографічні хеш-функції в поєднанні з секретним ключем [8].

Основна ідея HMAC полягає в тому, щоб забезпечити цілісність даних і перевірити, що повідомлення походить від очікуваного відправника, а також не було змінено під час передачі [8].

Процес створення HMAC включає такі кроки:

- секретний ключ об'єднується з повідомленням за допомогою операції XOR;
- результат роботи передається через криптографічну хеш-функцію SHA-256;
- секретний ключ знову об'єднується з хешем за допомогою операції XOR;
- результат опрацьованої роботи знову передається через криптографічну хеш-функцію;
- фінальний хеш є HMAC.

Перевага використання HMAC полягає в тому, що навіть, якщо зломисник знає повідомлення, він не зможе створити дійсний HMAC без знання секретного ключа. Це забезпечує автентифікацію повідомлення.

Діаграма діяльності алгоритму хешування HMAC зображена на рисунку 3.4.

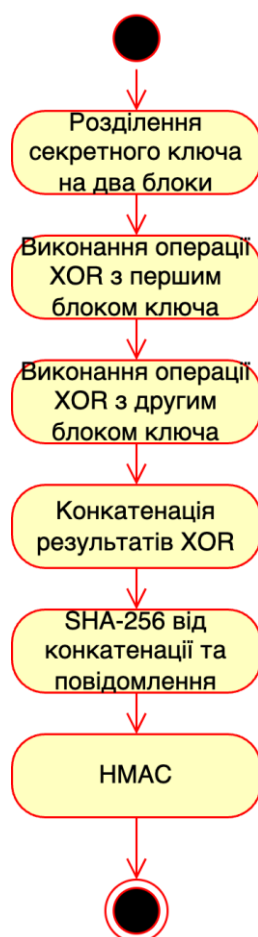


Рисунок 3.4 – UML-діаграма діяльності алгоритму хешування HMAC

3.4 Розробка документації для програмного продукту

- мінімальні системні вимоги для сайту відповідають загальним вимогам WordPress. Вони включають:
- веб-сервер Apache (рекомендовано версію 2.4 або новішу);
- СКБД MySQL версії 5.6 або новіша;
- PHP 7.4 або новіший (рекомендовано PHP 7.4 або 8.0);
- необхідні розширення PHP: mysql, mysqli, libxml, dom, simplexml, xmlreader, zlib, pcre, imagick (для роботи з зображеннями), curl, openssl;
- оперативна пам'ять мінімум 512 МБ (рекомендовано 1 ГБ або більше);
- мінімум 1 ГБ вільного дискового простору;
- рекомендовано використовувати SSL-сертифікат для забезпечення безпечного з'єднання (HTTPS).

Щоб розгорнути WordPress на хостингу необхідно завантажити дистрибутив WordPress FTP або файловий менеджер. Далі необхідно створити нову базу даних MySQL через панель керування хостингом (наприклад, cPanel).

Конфігурування здійснюється через файл wp-config.php. Основні налаштування наведено в лістингу 3.6.

Лістинг 3.6 – Файл .wp-config.php

```
define( 'DB_NAME', 'ім'я_бази_даних' );
define( 'DB_USER', 'ім'я_користувача_бази_даних' );
define( 'DB_PASSWORD', 'пароль_бази_даних' );
define( 'AUTH_KEY', 'унікальне_значення' );
define( 'SECURE_AUTH_KEY', 'унікальне_значення' );
define( 'LOGGED_IN_KEY', 'унікальне_значення' );
define( 'NONCE_KEY', 'унікальне_значення' );
define( 'AUTH_SALT', 'унікальне_значення' );
define( 'SECURE_AUTH_SALT', 'унікальне_значення' );
define( 'LOGGED_IN_SALT', 'унікальне_значення' );
define( 'NONCE_SALT', 'унікальне_значення' );
```

Кінець лістингу 3.6

Після цього потрібно відкрити веб-браузер та перейти до URL-адреси сайту. WordPress автоматично перенаправить на сторінку встановлення. На ній потрібно обрати мову, назву сайту, ім'я користувача адміністратора, пароль та адресу електронної пошти.

3.5 Розробка інсталяційного пакета

Інсталяційний пакет системи представляє собою архів, що містить всі необхідні файли та папки для встановлення. Цей пакет включає в себе основні файли, такі як index.php, wp-config-sample.php, а також цілу структуру папок, серед яких найважливіші – wp-content, wp-includes, wp-admin. Кожна з цих папок виконує певну функцію. Наприклад, wp-content містить всі плагіни, теми та завантажені медіафайли. Папка wp-admin відповідає за адміністративну частину сайту, тобто за панель управління, в якій адміністратори можуть

налаштовувати сайт, додавати новий контент та виконувати інші важливі дії. Папка `wp-includes` містить основні файли ядра WordPress.

Щоб розпочати роботу з цим пакетом, користувачеві потрібно розпакувати архів у кореневу директорію свого веб-сервера, після чого перейти до процесу налаштування через веб-інтерфейс.

Діаграму розгортання сайту на хостингу наведено на рисунку 3.5.

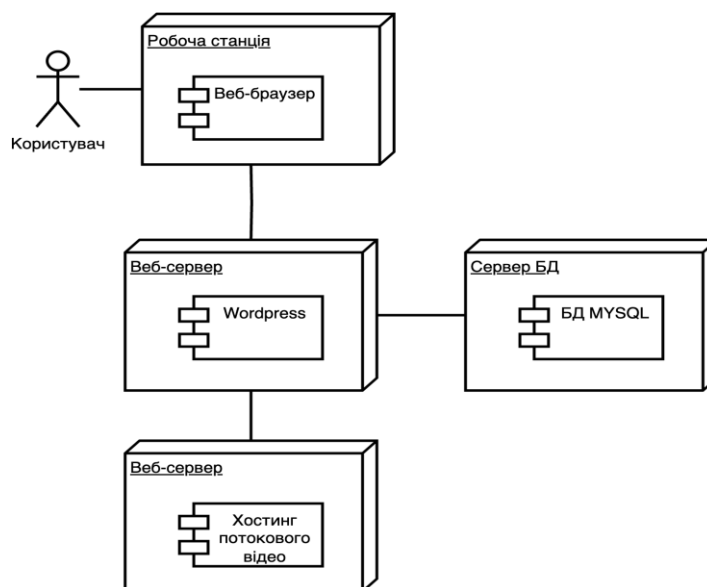


Рисунок 3.5 – UML-діаграма розгортання сайту

Висновки до розділу 3

У даному розділі було детально розглянуто практичну реалізацію онлайн-бібліотеки фільмів на основі системи керування вмістом WordPress. Основну функціональність сайту реалізовано засобами PHP, який є серверною мовою програмування з відкритим вихідним кодом і широко використовується для розробки веб-додатків.

Розглянуто реалізацію ключових функцій сайту, таких як виведення контенту за допомогою функції `WP_Query`, реалізація пошуку з використанням шаблону `search.php` та вбудовування медіаплеєра для перегляду фільмів через створення шорткоду за допомогою функції `add_shortcode`.

Проведено тестування сайту за допомогою методики тест-кейсів, яка дозволяє систематично перевірити всі важливі функції та сценарії використання. Розроблено та успішно виконано тест-кейси для перевірки працездатності сайту, функціоналу пошуку та відтворення відео.

Описано мінімальні системні вимоги для розгортання сайту на WordPress, процес ручного розгортання та конфігурування через файл `wp-config.php`. Створено інсталяційний пакет системи у вигляді архіву з необхідними файлами та папками для встановлення.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було успішно розроблено онлайн-бібліотеку фільмів з використанням системи керування вмістом WordPress. Проведений аналіз предметної області дозволив визначити актуальність та доцільність створення такої платформи, враховуючи зростаючий попит на послуги потокового відео та необхідність задовольнити потреби глядачів у зручному доступі до якісного контенту.

Розроблена система відповідає поставленим вимогам та забезпечує повноцінний функціонал для перегляду інформації про фільми, пошуку за різними критеріями, залишення відгуків та оцінок, а також безпосереднього перегляду відеоконтенту на сайті. Використання WordPress дозволило створити гнучку та масштабовану платформу, яка може бути легко розширена та модифікована відповідно до майбутніх потреб.

Практична реалізація онлайн-бібліотеки фільмів включала розробку ключових функцій, таких як виведення контенту за допомогою функції `WP_Query`, реалізація пошуку з використанням шаблону `search.php` та вбудовування медіаплеєра для перегляду фільмів через створення шорткоду. Проведене тестування підтвердило коректність роботи основних функцій системи.

Завдання, поставлене в першому розділі, було виконано у наслідку чого, ми отримали готовий продукт «FilmsLib». Для функціональності, каталогізації та класифікації фільмів у бібліотеці було використано утиліти та плагіни, запозичені з внутрішнього маркетплейсу WordPress.

Для зберігання інформації про фільми було реалізовано структуру бази даних за допомогою SQL. Бази даних включають в себе наступні дані: рік випуску, жанр, режисер, актори, синопсис та інше. База даних може бути відфільтрована для оптимізації пошукового процесу всередині неї.

Систему пошуку було реалізовано за допомогою комбінації функцій та шаблонів WordPress. В шаблоні створюється форма, яка продемонстрована в

лістингу 3.2. Для відображення результатів пошуку використовується шаблон `search.php`, зображений в лістингу 3.3.

Коментрі реалізовані за допомогою стандартних функцій WordPress, таких як: `comment_form()`, `wp_list_comments()`, `comments_number()` та інших. Ці функції зазвичай використовуються в файлі темплейту коментарів `comments.php` для виведення форми коментарів та відображення вже існуючих.

Було інтегровано медіаплеєр, за допомогою якого користувач може переглядати файли мультимедіа. Медіаплеєр піддається налаштуванням, де користувач може обрати швидкість відтворення, якість, субтитри та інші індивідуальні параметри.

Розроблена адміністративна панель, що дозволяє здійснювати модерацію всієї платформи, включно з профілями користувачів, коментарів та файлів мультимедіа. Доступ до адміністративної панелі має лише користувач, який отримав спеціальний дозвіл до панелі.

Було також розроблено зручний та інтуїтивний інтерфейс користувача для навігації по бібліотеці фільмів, перегляду деталей фільму, пошуку та фільтрації результатів.

Також здійснено розробку адміністративної панелі для керування вмістом бібліотеки, додавання нових фільмів, редагування існуючих записів, модерації відгуків користувачів тощо.

Особливу увагу було приділено питанням безпеки, зокрема захисту конфігураційного файлу `wp-config.php`, налаштуванню дозволів на директорії та файли, використанню складних паролів, регулярному оновленню WordPress і плагінів, застосуванню двофакторної автентифікації та SSL-сертифікату.

Результати роботи мають практичну цінність і можуть бути використані для створення реального онлайн-кінотеатру. Розроблена система демонструє можливості WordPress як потужної платформи для розробки веб-додатків та може слугувати основою для подальших досліджень та вдосконалень у галузі онлайн-бібліотек та потокового відео.

Подальші шляхи розвитку проекту можуть включати розширення функціоналу, додавання персоналізованих рекомендацій, інтеграцію з соціальними мережами, оптимізацію продуктивності та масштабованості системи, а також адаптацію для різних пристроїв та платформ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Netflix. URL: <https://www.netflix.com/ua/> (дата звернення: 03.02.2024).
2. Amazon.de Sign up for Prime Video. URL: <https://www.primevideo.com/> (дата звернення: 01.02.2024).
3. Hulu. Watch Thousands of Shows and Movies URL: <https://www.hulu.com/> (дата звернення: 04.02.2024).
4. Taxonomies. Wordpress. URL: <https://wordpress.org/documentation/article/taxonomies/> (дата звернення: 01.03.2024).
5. Blog Tool, Publishing Platform, and CMS – WordPress.org. URL: <https://wordpress.org/> (дата звернення: 05.03.2024).
6. The C4 model for visualising software architecture. URL: <https://c4model.com/> (дата звернення: 05.03.2024).
7. How to Write Test Cases with Examples. Guru99. URL: <https://www.guru99.com/test-case.html> (дата звернення: 24.05.2024).
8. RFC 2104: HMAC: Keyed-Hashing for Message Authentication. IETF Datatracker. URL: <https://datatracker.ietf.org/doc/html/rfc2104> (дата звернення: 24.05.2024).
9. "WordPress Database Optimization with WP-Optimize". TechCrunch. John Doe. URL: <https://techcrunch.com/wordpress-database-optimization-wp-optimize> (дата звернення: 24.05.2024).
10. PHP 8: A Look Into the Latest Features and Improvements" – SitePoint. Jane Smith. URL: <https://www.sitepoint.com/php-8-latest-features-improvements> (дата звернення: 24.05.2024).
11. "SQL Injection: Understanding and Preventing It". OWASP. URL: <https://owasp.org/sql-injection> (дата звернення: 24.05.2024).
12. "Best Practices for Securing WordPress Websites". Sucuri Blog. Mary Johnson. URL: <https://blog.sucuri.net/wordpress-security-best-practices> (дата звернення: 24.05.2024).