

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення інтерактивного вебдодатку для фільмотеки
засобами JavaScript»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Хубулава Георгій Кахаберович
Керівник: Копішинська Олена
Петрівна
Рецензент: Яхін Сергій Валерійович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій УТКІН

«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Хубулави Георгія Кахаберовича

1. Тема кваліфікаційної роботи:
«Розроблення інтерактивного вебдодатку для фільмотеки засобами JavaScript»,
Керівник роботи: к. ф.-м. н., доцент, професор кафедри інформаційних систем та технологій Копішинська Олена Петрівна.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: інформаційні ресурси бібліотек JavaScript React <https://react.dev/>, Redux <https://redux.js.org/>, редактор вихідного коду <https://code.visualstudio.com/>, платформа обробки зображень Cloudinary <https://cloudinary.com/>, <https://www.figma.com/design/>, статистичні дані <https://mate.academy/>
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1. Огляд загальних функцій інтернет-магазинів
Розділ 2. Формування концепції та обґрунтування методів розробки інтернет-магазину спортивного одягу
Розділ 3. Практичні аспекти програмної реалізації інтернет-магазину спортивного одягу
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження
6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 19.06.2026 р.	

Здобувач вищої освіти

Георгій ХУБУЛАВА

Керівник роботи

Олена КОПШИНСЬКА

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 РОЗВИТОК ВЕБТЕХНОЛОГІЙ ТА ФУНКЦІЙ ВЕБСАЙТІВ В ІНФОРМАЦІЙНОМУ ЗАБЕЗПЕЧЕННІ ІНТЕРЕСІВ СУСПІЛЬСТВА.....	9
1.1 Роль мережі інтернет у задоволенні інформаційних потреб сучасного суспільства	9
1.2 Розвиток вебтехнологій від початкових версій до сучасних.....	12
1.3 Розширення можливостей застосування JavaScript на основі фреймворків і бібліотек	16
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА ВИБІР МЕТОДОЛОГІЇ СТВОРЕННЯ ВЕБДОДАТКУ ДЛЯ ФІЛЬМОТЕКИ.....	20
2.1 Аналіз вебресурсів кінематографічної тематики та формування вимог до вебдодатку для фільмотеки	20
2.2 Проєктування структури та архітектури вебдодатку	22
2.3 Формування стеку технологій з розроблення вебсайту фільмотеки та створення макету.....	27
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ВЕБСАЙТУ ДЛЯ ФІЛЬМОТЕКИ.....	33
3.1 Створення HTML-структури та технічні рішення інтерфейсу користувача.....	33
3.2 Роль JavaScript у реалізації функціональності вебдодатку «Фільмотека»	39
3.3 Економічне оцінювання вартості проєкту.....	44
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ.....	52

ВСТУП

Актуальність теми кваліфікаційної роботи розглядається в контексті інформаційного суспільства, коли процеси вирішення різноманітних задач, збору, обробки та передачі великої кількості інформації головним чином здійснюються на основі потужних технологій пошуку й обробки даних в мережі інтернет. Значна частина користувачів мережі інтернет використовує вебресурси для отримання новин, перегляду мультимедійного контенту, спілкування та обміну враженнями щодо різноманітних подій і продуктів. Особливе місце серед таких ресурсів займають вебсайти та вебдодатки, присвячені кіноіндустрії, оскільки кіно є важливою складовою сучасної культури та розважальної сфери.

Стрімкий розвиток технологій JavaScript, поява сучасних бібліотек і фреймворків, а також поширення концепції інтерактивних вебдодатків створюють нові можливості для розроблення функціональних та зручних інформаційних систем. Сучасні користувачі очікують не лише доступу до інформації про фільми, а й можливості взаємодії з контентом: перегляду трейлерів, ознайомлення з біографіями акторів, участі в обговореннях, публікації власних відгуків та отримання актуальних новин кіноіндустрії.

Розроблення інтерактивного вебдодатку для фільмотеки дозволяє поєднати інформаційні, мультимедійні та комунікаційні функції в єдиному середовищі, що робить тему дослідження актуальною як з практичної точки зору, так і з позицій вивчення сучасних засобів веброзробки. Виконання роботи сприяє розширенню меж уяви про тематику та напрямки інтеграції вебдодатків у різноманітні сфери діяльності. Крім того, створення такого програмного продукту на основі JavaScript демонструє ефективність використання сучасних вебтехнологій для побудови динамічних та інтерактивних вебресурсів.

Метою кваліфікаційної роботи є розроблення інтерактивного вебдодатку для фільмотеки, який забезпечує користувачам доступ до інформації про складові кіноіндустрії на основі широкого спектру вебтехнологій, зокрема мови програмування JavaScript.

Завданнями кваліфікаційної роботи є:

- вивчення інформаційної бази предметної області та користувацьких потреб цільової аудиторії;
- здійснення аналізу технологій розроблення вебсайтів на стороні клієнтів та обґрунтування вибору для подальшої роботи;
- розроблення графічного макету вебсайту і підбір графічного матеріалу з різноманітних джерел;
- програмна реалізація проєкту із забезпеченням інтерактивних та динамічних функцій, у т. ч. за рахунок програмування на JavaScript;
- забезпечення адаптивності сайту для повноцінного перегляду на екранах різного розміру, проведення технічного аудиту і оптимізації вебсайту;
- здійснення економічного аналізу розробки за відомими методами.

Об'єктом дослідження є процес створення інтерактивних вебдодатків для представлення та обробки інформації про кіноіндустрію..

Предметом дослідження є методи, засоби та технології розроблення інтерактивного вебдодатку для фільмотеки з використанням мов JavaScript, HTML та CSS в середовищах редакторів коду та графічного макетування.

Методами дослідження є: аналіз та узагальнення науково-технічної літератури з питань веброзробки, графічне проєктування, порівняльний, тестування програмного забезпечення для перевірки працездатності розробленого продукту, підключення бібліотек JavaScript тощо.

Інформаційна база представлена технічними стандартами веброзробки, науковими статтями, популярними публікаціями, матеріалами науково-практичних конференцій всіх рівнів, ресурсами мережі інтернет, узагальненими статистичними даними аналітичних компаній в галузі ІТ, перелік яких подано у списку використаних джерел.

Практична значущість кваліфікаційної роботи полягає у створенні працездатного інтерактивного вебдодатку для фільмотеки, який може використовуватися як інформаційний ресурс для шанувальників кіно. Розроблений вебдодаток забезпечує зручний доступ до новин кіноіндустрії,

трейлерів фільмів, біографій акторів та відгуків користувачів. Отримані результати можуть бути використані як основа для подальшого розширення функціональності вебресурсу, інтеграції із зовнішніми базами даних та впровадження додаткових сервісів рекомендацій і персоналізації контенту. Крім того, створений проєкт демонструє практичні навички використання сучасних вебтехнологій та може бути використаний як приклад розроблення інтерактивних вебдодатків у навчальних і професійних цілях.

Апробація результатів роботи відбувалася шляхом оприлюднення доповіді на студентській науково-практичній конференції.

Структура та обсяг кваліфікаційної роботи. Робота має 47 сторінок, складається зі змісту, вступу, трьох розділів, висновків, списку використаних джерел загальною кількістю найменувань 41 та додатків. Містить 19 рисунків і 2 таблиці.

РОЗДІЛ 1

РОЗВИТОК ВЕБТЕХНОЛОГІЙ ТА ФУНКЦІЙ ВЕБСАЙТІВ В ІНФОРМАЦІЙНОМУ ЗАБЕЗПЕЧЕННІ ІНТЕРЕСІВ СУСПІЛЬСТВА

1.1 Роль мережі інтернет у задоволенні інформаційних потреб сучасного суспільства

Інформаційне суспільство є сучасним етапом розвитку людства, у якому інформація, знання та цифрові технології виступають основними ресурсами економічного, культурного та соціального розвитку. Характерною ознакою такого суспільства є активне використання інформаційно-комунікаційних технологій для створення, поширення, зберігання та обробки інформації. У цих умовах особливого значення набуває забезпечення швидкого та зручного доступу користувачів до необхідних інформаційних ресурсів незалежно від їхнього місця перебування [1].

Одним із найважливіших досягнень інформаційної епохи стало створення глобальної мережі Інтернет та сервісу World Wide Web (WWW), що об'єднав мільйони вебресурсів у єдиний інформаційний простір. Завдяки використанню гіпертекстових технологій та вебстандартів користувачі отримали можливість швидко знаходити, переглядати та обмінюватися інформацією практично з будь-якої точки світу. Саме вебсайти стали основною формою представлення інформації в мережі та ключовим інструментом взаємодії між власниками інформаційних ресурсів і користувачами [2].

Вебсайт являє собою сукупність взаємопов'язаних вебсторінок, об'єднаних спільною тематикою, структурою та адресою. Сучасні вебресурси забезпечують цілодобовий доступ до інформації, мультимедійних матеріалів і різноманітних онлайн-сервісів. На відміну від перших вебсайтів, які переважно виконували роль електронних сховищ документів, сучасні вебресурси характеризуються високим рівнем інтерактивності та можливістю двосторонньої взаємодії між користувачами й інформаційною системою [3].

Розвиток вебтехнологій суттєво змінив способи отримання інформації та організації дозвілля. Значна частина населення використовує мережу Інтернет не лише для професійної діяльності чи навчання, а й для ознайомлення з новинами, перегляду відеоматеріалів, участі в тематичних спільнотах та обговорення культурних подій. Особливо помітною ця тенденція є у сфері кіноіндустрії, де користувачі регулярно звертаються до вебресурсів для отримання інформації про нові фільми, перегляду трейлерів, ознайомлення з біографіями акторів, рейтингами, рецензіями та відгуками інших глядачів [3].

На сьогоднішній день вебсайт є невід'ємною частиною супроводу будь-якої професії, будь-якої сфери діяльності. Підприємство, яке має вебресурс, має більшу кількість клієнтів, потужний засіб популяризації своєї діяльності, а отже й суттєві переваги серед конкурентів. Вебсайт дає можливість партнерам та клієнтам більше дізнатися про діяльність підприємства та скласти уявлення про нього. Наявність такого ресурсу в інтернеті означає, що бізнес розвивається, доступний з будь-якої точки та у будь-який час [2].

Щоб точніше зрозуміти, для чого потрібен сайт для будь-якої компанії, варто розглянути основні функції, які він може виконувати:

1) інформаційна – представлення нової інформації з деякої тематики, області чи проблеми. Клієнти та партнери будуть проінформовані про оновлення або зміни, сучасні тренди, зможуть висловити свою думку при наявності інтерактивних елементів, зареєструватися і т.ін. [1];

2) маркетингова – просування товарів чи послуг, збільшення попиту на них, статистична інформація, огляди тощо;

3) іміджева – представлення підприємства, створення образу, популяризація діяльності.

За допомогою вебсайту можна швидко отримати доступ до потрібної інформації. З кожним днем з'являються нові шляхи для стимулювання продажів з використанням інтернету, що дає можливість підвищити взаємодію всіх учасників комунікаційного процесу, знаходити необхідні обсяги інформації про наявних та потенційних споживачів-користувачів [3].

Сучасні вебсайти та вебдодатки кінематографічної тематики виконують низку важливих функцій. Насамперед вони забезпечують інформаційну підтримку користувачів шляхом публікації новин про кіноіндустрію, анонсів прем'єр, оглядів та аналітичних матеріалів. Не менш важливою є комунікаційна функція, яка реалізується через можливість залишення коментарів, оцінювання фільмів, ведення блогів та участі в дискусіях. Крім того, такі ресурси виконують культурно-просвітницьку функцію, сприяючи популяризації кіномистецтва та поширенню відомостей про його історію, жанри, режисерів і акторів [4].

Відомі аналітичні компанії, такі як Statista, Data.AI, Similarweb, Semrush здійснюють системний моніторинг та публікують регулярні звіти про популярність інформаційних технологій. Серед них найбільш повним, інтегрованим джерелом вважають щорічні огляди Digital [5]. Цей документ пропонує огляди і для українського сегменту аудиторії. Навіть під час такого потужного виклику, яким є військова агресія на території України, запровадження військового стану, розрив багатьох економічних та інформаційних зв'язків, періодична недоступність мережевих послуг, саме діяльність через інтернет демонструє стійкість, відновлення та зростання. За даними звіту Digital 2026 [5], станом на початок 2021 р. кількість українських інтернет-користувачів була 29,47 млн, що становило 67,6 % населення країни – рівень проникнення (рис. 1.1).

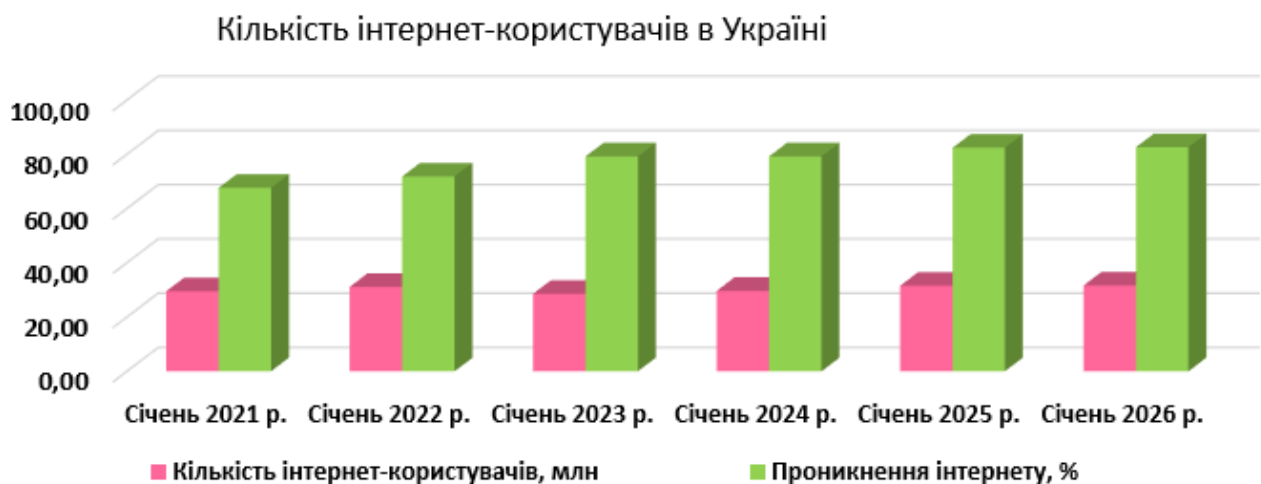


Рисунок 1.1 – Динаміка росту кількості користувачів інтернету в Україні

У 2026 р. ця цифра зростає до 31,5 млн (проникнення склало 82,5%). Отже, найбільш очевидно, щоб отримати більшу кількість нових потенційних клієнтів, будь-якому підприємству, підприємцю, одиниці суспільної діяльності необхідно використовувати ресурси інтернету.

Сучасний рівень розвитку інформаційної технології характеризується наявністю потужної комп'ютерної техніки, програмного забезпечення, розвинених комунікацій, використання вебтехнологій. Кожного дня інформаційне суспільство отримує та переробляє інформацію за допомогою комп'ютерної техніки та новітніх технологій. Для підтримки такого способу діяльності необхідним елементом стає наявність сучасного вебсайту [6].

Сучасний користувач очікує від вебресурсу не лише надання інформації, а й високої швидкодії, зручного інтерфейсу, інтерактивності та адаптивності до різних типів пристроїв. Саме тому під час розроблення вебдодатків особлива увага приділяється використанню сучасних вебтехнологій, які забезпечують комфортну взаємодію користувача з інформаційним середовищем [7].

Таким чином, мережа Інтернет стала невід'ємною складовою інформаційного суспільства та основним засобом задоволення інформаційних потреб населення. Вебсайти та вебдодатки забезпечують оперативний доступ до різноманітних інформаційних ресурсів, сприяють комунікації між користувачами та створюють нові можливості для поширення культурного контенту. У сфері кіноіндустрії вони виступають ефективним засобом інформування, популяризації кіномистецтва та формування спільнот шанувальників кіно, що обумовлює актуальність розроблення сучасних інтерактивних вебдодатків для фільмотек.

1.2 Розвиток вебтехнологій від початкових версій до сучасних

Історія сучасних вебтехнологій нерозривно пов'язана зі створенням Всесвітньої павутини (World Wide Web). У 1989 році британський науковець Tim

Berners-Lee запропонував концепцію розподіленої гіпертекстової системи документів, яка мала забезпечити швидкий обмін науковою інформацією між дослідниками. Уже в 1990 році було створено перший вебсервер, веббраузер та мову гіпертекстової розмітки HTML, що заклало основу подальшого розвитку вебпростору [8, 9].

Перші вебсторінки були переважно статичними документами, що містили текстову інформацію та прості гіперпосилання. Основною технологією їх створення стала мова HTML (HyperText Markup Language), яка дозволяла описувати структуру документа та забезпечувати навігацію між сторінками. Зі зростанням популярності мережі Інтернет виникла необхідність у стандартизації вебтехнологій, що сприяло створенню міжнародних специфікацій та рекомендацій для розробників браузерів і вебресурсів [8].

Протягом 1990-х років HTML активно розвивався, отримуючи нові можливості для відображення таблиць, графічних елементів, форм введення даних та інтеграції сценаріїв. Важливим етапом став вихід HTML 4.0, де було реалізовано принцип відокремлення структури документа від його візуального оформлення. Це стало передумовою появи каскадних таблиць стилів CSS та сучасного підходу до веброзробки [10].

Справжнім кроком уперед стало впровадження HTML5, який сьогодні є базовим стандартом розроблення вебресурсів. HTML5 значно розширив можливості створення мультимедійних та інтерактивних вебдодатків без використання сторонніх плагінів [11]. До його основних переваг належать:

- підтримка вбудованого аудіо- та відеоконтенту;
- використання семантичних елементів (<header>, <nav>, <section>, <article>, <footer>);
- підтримка локального зберігання даних у браузері;
- інтеграція графічних можливостей через елемент Canvas;
- покращена підтримка мобільних пристроїв за рахунок спеціальних функцій та елементів;
- підвищення доступності та оптимізації для пошукових систем [12-13].

Для вебдодатку фільмотеки особливо важливими є можливості HTML5 щодо роботи з мультимедійним контентом, адже вони дозволяють відображати трейлери фільмів, інтегрувати відеоматеріали та створювати зручну структуру інформаційних сторінок [13].

Візуальне оформлення вебсторінок забезпечується каскадними таблицями стилів CSS (Cascading Style Sheets). Якщо HTML відповідає за структуру документа, то CSS визначає його зовнішній вигляд. Сучасна версія CSS3 значно розширила можливості оформлення інтерфейсів завдяки підтримці анімації, трансформацій, градієнтів, тіней, гнучких макетів та адаптивного дизайну. Саме CSS3 дозволив створювати складні інтерфейси без необхідності використання додаткових графічних технологій [14-15].

Одним із найважливіших етапів еволюції вебтехнологій стало поширення мови програмування JavaScript. Спочатку вона використовувалася для реалізації нескладних інтерактивних елементів, однак сьогодні JavaScript є основою більшості сучасних вебдодатків. Завдяки цій мові реалізуються динамічне оновлення контенту, обробка дій користувача, обмін даними із сервером, валідація форм та створення інтерактивних інтерфейсів.

Сучасний JavaScript розвивається відповідно до стандарту ECMAScript, який регулярно оновлюється. Нові версії мови містять механізми асинхронного програмування, модульну архітектуру, покращені засоби роботи з даними та підтримку об'єктно-орієнтованого підходу. Завдяки цьому JavaScript використовується не лише на стороні клієнта, а й для серверної розробки за допомогою платформи Node.js [16].

Важливу роль у роботі JavaScript відіграє об'єктна модель документа DOM (Document Object Model), яка представляє HTML-документ у вигляді деревоподібної структури. DOM забезпечує програмний доступ до всіх елементів сторінки та дозволяє змінювати їхній вміст, стилі й поведінку без повторного завантаження документа. Саме цей механізм лежить в основі інтерактивності сучасних вебдодатків. Об'єктна структура сайту DOM – це об'єктне уявлення вихідного HTML-документа, для різних програм, простіше

кажучи – дерево вузлів [17]. Воно має один корінь, який розгалужується на безліч дочірніх гілок, кожна з яких може розгалужуватися сама і закінчуватися «листям». Корінь – це HTML-елемент, а гілки – вкладені елементи. Дві частини об'єктної структури є обов'язковими – елементи `<head>` (головна, керуюча частина) і `<body>` - тіло вебсайту (рис. 1.2) [18].

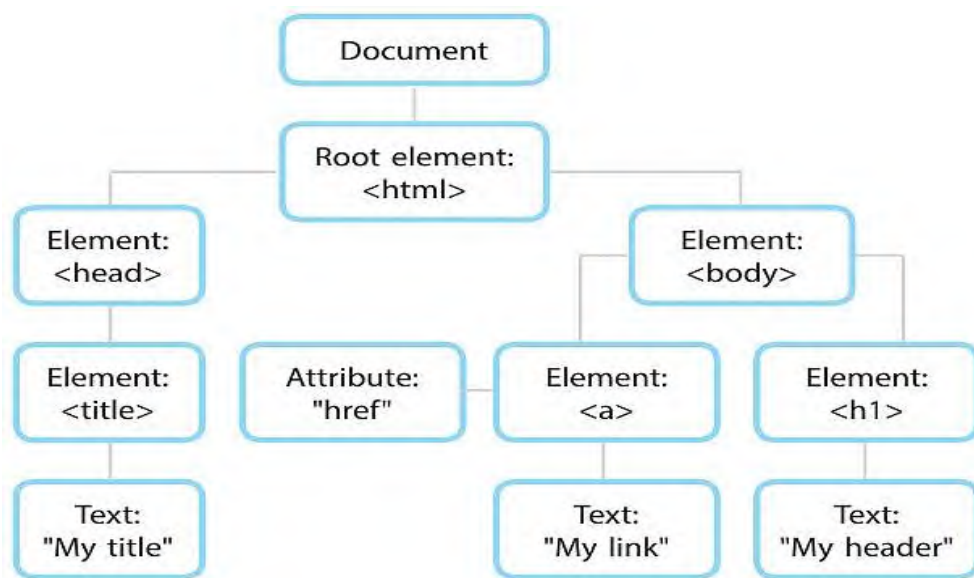


Рисунок 1.2 – Об'єктна структура документу HTML DOM

Шлях від HTML коду у файлі до оформленої сторінки у браузері називається критичним шляхом рейдерингу, його поділяють на два етапи: аналіз коду і побудова дерева рейдера та відтворення сторінки [17].

Дерево рейдера, або дерево візуалізації – це структура, яка складається із HTML-елементів, які будуть відображатися на сторінці і пов'язані з ними стилі. Таке дерево складається з двох компонентів: об'єктної моделі CSS та DOM [18].

Тобто, починається все із загального Document, далі в нього вкладається елемент `<HTML>`, наступними слідує `<HEAD>` і `<BODY>`, а також всі наступні елементи у правильній послідовності. Саме DOM дозволяє отримати доступ до елементів сторінки за допомогою JavaScript [19].

Подальший розвиток вебтехнологій привів до появи концепції вебдодатків, які за функціональністю наближаються до традиційного програмного забезпечення. Сьогодні для створення таких систем активно

використовуються сучасні JavaScript-фреймворки та бібліотеки, зокрема React, Vue.js, Angular та платформа Node.js. Вони дозволяють значно пришвидшити процес розроблення, забезпечити модульність коду та підвищити якість програмних продуктів.

Таким чином, розвиток вебтехнологій пройшов шлях від простих статичних документів до повноцінних інтерактивних вебдодатків. Сучасний технологічний стек HTML5, CSS3 та JavaScript забезпечує створення функціональних, адаптивних і мультимедійних вебресурсів. Саме ці технології є основою розроблення інтерактивного вебдодатку для фільмотеки, що розглядається у даній роботі.

1.3 Розширення можливостей застосування JavaScript на основі фреймворків і бібліотек

Стрімкий розвиток вебтехнологій сприяв появі великої кількості інструментів, які значно спрощують процес створення сучасних вебдодатків. Якщо на початкових етапах розвитку мережі для розроблення вебресурсів було достатньо використання HTML, CSS та JavaScript, то зі збільшенням складності програмних систем виникла потреба у спеціалізованих бібліотеках та фреймворках, які дозволяють організувати код, прискорити розроблення та забезпечити підтримку масштабних проєктів.

Серед найбільш популярних інструментів екосистеми JavaScript сьогодні можна виділити React, Angular, Vue.js та jQuery. Незважаючи на появу нових технологій, вони продовжують використовуватися для створення вебресурсів різного масштабу – від невеликих інформаційних сайтів до корпоративних вебсистем та інтерактивних вебдодатків [20].

Однією з найпоширеніших технологій є React – бібліотека JavaScript для створення користувацьких інтерфейсів, розроблена компанією Meta (Facebook) у 2013 році. Основною особливістю React є компонентний підхід, відповідно до

якого інтерфейс складається з незалежних багаторазово використовуваних компонентів. Такий підхід значно спрощує супровід проєкту та дозволяє ефективно організовувати структуру коду. React використовує механізм Virtual DOM, який оптимізує оновлення сторінки та забезпечує високу швидкодію вебдодатків [21].

Іншим популярним рішенням є Angular – повноцінний фреймворк, який підтримується компанією Google. Angular містить широкий набір вбудованих інструментів для маршрутизації, роботи з даними, тестування та керування станом застосунку. Завдяки комплексному підходу Angular часто використовується для розроблення великих корпоративних інформаційних систем. Водночас складність архітектури та високий поріг входу можуть ускладнювати його використання в невеликих проєктах.

Фреймворк Vue.js поєднує простоту освоєння та високу продуктивність. Його архітектура дозволяє поступово інтегрувати функціональність у вже існуючі вебпроєкти або використовувати Vue.js як основу для створення повноцінних односторінкових застосунків (Single Page Applications, SPA). Завдяки якісній документації та активній спільноті Vue.js став одним із найпопулярніших інструментів сучасної веброзробки [22].

Окремо слід згадати бібліотеку jQuery, яка відіграла важливу роль у розвитку клієнтського програмування. Вона значно спрощувала роботу з DOM, обробку подій та виконання AJAX-запитів у період, коли браузері мали суттєві відмінності у підтримці вебстандартів. Хоча сьогодні більшість її функцій реалізовано безпосередньо засобами сучасного JavaScript, jQuery продовжує використовуватися для підтримки великої кількості існуючих вебпроєктів.

Згідно аналітичних звітів компанії W3Tech [23], 18.6% вебсайтів не використовують жодну з бібліотек JavaScript, які були в зоні опитування, jQuery використовується на 73,8 % усіх вебсайтів (додаток А, верхня смуга), що складає 90,6 % частки ринку бібліотек JavaScript (нижня смуга діаграми).

Порівняльну характеристику найбільш поширених фреймворків і бібліотек JavaScript наведено в таблиці 1.1 [20-22].

Таблиця 1.1 – Порівняння сучасних JavaScript-фреймворків та бібліотек

Технологія	Переваги	Недоліки
React	Висока продуктивність, компонентний підхід, велика екосистема, активна спільнота	Потребує додаткових бібліотек для повноцінної архітектури
Angular	Повний набір інструментів, підтримка Google, висока масштабованість	Складність освоєння, значний обсяг коду
Vue.js	Простота використання, хороша документація, висока продуктивність	Менша частка корпоративного ринку порівняно з React та Angular
jQuery	Простота інтеграції, підтримка застарілих проєктів	Обмежені можливості для сучасних SPA-додатків

Сучасна веброзробка також активно використовує серверні JavaScript-технології. Однією з найпоширеніших є платформа Node.js, яка дозволяє виконувати JavaScript-код поза браузером та створювати високопродуктивні серверні застосунки. На базі Node.js функціонує велика кількість популярних інструментів та фреймворків, серед яких Express.js, Next.js та NestJS.

Для розроблення інтерактивних вебдодатків дедалі частіше застосовується концепція Single Page Application (SPA), відповідно до якої сторінка завантажується лише один раз, а подальше оновлення контенту відбувається без повного перезавантаження браузера. Такий підхід забезпечує швидку взаємодію користувача із системою та покращує зручність використання вебресурсів.

У контексті розроблення вебдодатку для фільмотеки використання сучасних JavaScript-бібліотек та фреймворків дозволяє реалізувати інтерактивний інтерфейс користувача, динамічне відображення інформації про фільми, перегляд трейлерів, систему відгуків та інші функціональні можливості, що відповідають вимогам сучасних вебресурсів. Таким чином, застосування сучасних інструментів екосистеми JavaScript є важливою передумовою створення якісного та конкурентоспроможного вебдодатку.

Для організації та керування структурою вебдодатку можна використовувати різні інструменти, такі як Webpack, Gulp, Grunt, Parcel, які дозволяють автоматизувати процес збірки та оптимізації вебдодатків [24].

Окрім того, для розробки вебдодатків можна використовувати різноманітні інструменти тестування, які дозволяють перевіряти працездатність та функціональність додатку, такі як Jest, Mocha, Chai та інші.

JavaScript можна використовувати, наприклад, для створення форми замовлення компонентів, перевірки достовірності даних, введених користувачем, та передачі даних на сервер. Анімація, зокрема анімовані графіки, діаграми, графіки тощо, може бути створена за допомогою JavaScript. Це може допомогти у візуалізації даних і підвищити привабливість вебсайту для користувачів.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА ВИБІР МЕТОДОЛОГІЇ СТВОРЕННЯ ВЕБДОДАТКУ ДЛЯ ФІЛЬМОТЕКИ

2.1 Аналіз вебресурсів кінематографічної тематики та формування вимог до вебдодатку для фільмотеки

Стрімкий розвиток мережі Інтернет та цифрових технологій суттєво вплинув на способи поширення інформації про кіноіндустрію. Якщо раніше основними джерелами відомостей про фільми були друковані видання, телебачення та спеціалізовані журнали, то сьогодні переважна більшість користувачів отримує інформацію через вебресурси. Сучасні сайти кінематографічної тематики поєднують інформаційні, мультимедійні та комунікаційні функції, забезпечуючи швидкий доступ до новин, трейлерів, рейтингів, біографій акторів та відгуків глядачів.

На сьогодні існує велика кількість вебресурсів, присвячених кіноіндустрії. Серед найбільш відомих можна виділити IMDb, Rotten Tomatoes та Metacritic [25-26]. Ці ресурси містять великі бази даних фільмів, відомості про акторів і режисерів, рейтинги користувачів та професійних критиків, а також новини кінематографічної індустрії.

Проведений аналіз дозволив виділити основні функціональні можливості сучасних вебресурсів кінематографічної тематики:

- надання довідкової інформації про фільми;
- публікація новин та анонсів прем'єр;
- демонстрація трейлерів і рекламних матеріалів;
- формування рейтингів та рекомендацій;
- представлення інформації про акторів, режисерів і кінокомпанії;
- організація пошуку та фільтрації контенту;
- забезпечення взаємодії між користувачами через коментарі та відгуки.

Незважаючи на широкий набір функцій, багато популярних ресурсів орієнтовані, насамперед, на великі інформаційні бази даних та складні системи рекомендацій. Для звичайного користувача це може ускладнювати навігацію та сприйняття інформації. Крім того, частина ресурсів містить значну кількість рекламного контенту або вимагає реєстрації для доступу до окремих функцій.

Аналіз існуючих рішень показав, що користувачі зацікавлені не лише у пошуку інформації про фільми, але й у можливості отримувати актуальні новини кіноіндустрії, переглядати трейлери майбутніх прем'єр, знайомитися з творчими біографіями відомих акторів та ділитися власними враженнями від перегляду фільмів. Саме тому сучасний вебдодаток для фільмотеки повинен поєднувати функції інформаційного порталу та інтерактивної платформи для спілкування.

На основі проведеного аналізу було сформульовано основну мету проєктування вебдодатку – створення сучасного інформаційного ресурсу для шанувальників кіно, який забезпечуватиме доступ до актуальної інформації про кіноіндустрію та надаватиме можливості інтерактивної взаємодії користувачів із контентом.

1. Наявність головної сторінки з актуальними новинами зі світу кіно.
2. Можливість перегляду інформації про популярні та нові фільми.
3. Відображення трейлерів із використанням сучасних мультимедійних засобів.
4. Надання інформації про акторів та їхні творчі біографії.
5. Реалізація блогу або системи відгуків для публікації коментарів користувачів.
6. Забезпечення зручної навігації між розділами вебресурсу.
7. Підтримка адаптивного відображення на персональних комп'ютерах, планшетах та мобільних пристроях.
8. Забезпечення швидкого завантаження сторінок і комфортної взаємодії користувача з інтерфейсом.

Окрім функціональних вимог, важливими є також нефункціональні вимоги. Вебдодаток повинен характеризуватися зрозумілим інтерфейсом,

стабільністю роботи, привабливим дизайном та відповідністю сучасним вебстандартам. Особливу увагу необхідно приділити адаптивності інтерфейсу та зручності перегляду мультимедійного контенту, оскільки значна частина користувачів здійснює доступ до вебресурсів за допомогою мобільних пристроїв.

2.2 Проєктування структури та архітектури вебдодатку

Концепція проєктування вебдодатку для фільмотеки включає кілька важливих складових [27].

1. Проєктування структури та архітектури вебдодатку;
 - структура сторінок;
 - логічна схема взаємодії модулів;
 - обґрунтування вибору HTML5, CSS3, JavaScript;
 - структура каталогів проєкту.
2. Проєктування користувацького інтерфейсу:
 - концепція дизайну;
 - кольорова гама;
 - принципи UX/UI;
 - адаптивність інтерфейсу.
3. Реалізація навігації та взаємодії користувача з контентом;
 - меню;
 - переходи між сторінками;
 - робота з формами;
 - інтерактивні елементи JavaScript.
4. Реалізація функціональних модулів;
 - новини кіноіндустрії;
 - сторінка трейлерів;

- сторінка акторів;
- блог та система відгуків.

Концептуальну діаграму сайту представлено на рис. 2.1.



Рисунок 2.1 – Концептуальна діаграма сайту зі схемою посилань

Під час проєктування вебдодатку для фільмотеки було використано модульний підхід, відповідно до якого інтерфейс поділено на логічно незалежні функціональні блоки. Такий підхід забезпечує зручність подальшого супроводу програмного коду та спрощує розширення функціональних можливостей системи.

Структура вебдодатку побудована за принципом тематичного інформаційного порталу та включає такі основні компоненти:

- шапку сайту з навігаційним меню;
- головний інформаційний банер;
- блок опису фільму;
- розділ акторського складу;
- сторінки прем'єр;

- розділ оглядів та відгуків;
- рейтинг фільмів;
- нижній колонтитул із службовою інформацією.

Життєвий цикл створення сайту відбувається протягом основних етапів:

1. Етап ініціації (задум, визначення цілей);
2. Проєктування;
3. Розроблення (верстка);
4. Тестування, оптимізація;
5. Впровадження;
6. Адміністрування сайту.

На початковій стадії визначаються основні цілі та завдання сайту, тобто створюється опис проєкту, технічне завдання. Це досить важливий етап, адже грамотно складене технічне завдання (ТЗ) дозволить чітко визначити порядок робіт, терміни виконання, уникнути непорозумінь між замовником та виконавцем та економить час на створення сайту. ТЗ повинно вміщувати групи розробників, обсяг та графік робіт, фінансування і узгоджуватися із замовником [8, 9, 28].

Етап проєктування вміщує створення прототипу сайту, для наочного представлення розташування елементів на сторінці і узгодження із замовником. Та на цьому етапі створюють макети сторінок. Дизайн сайту відповідає за візуальне його представлення. Слід розуміти, що дизайн має забезпечувати зручність використання сайту (usability), тому всі його елементи повинні зручно розміщуватися, для того, щоб користувач міг легко знайти їх [8, 28].

Наступний етап – розроблення. При розробці сайту необхідно визначити кількість сторінок та зв'язки між ними. Виділяють ієрархічну, лінійну, ґратчасту та змішану структури. Далі вказано види побудови сайту [9, 28].

Ієрархічна – основна інформація розміщується згори вниз. В основному головна сторінка розміщується зверху, яка не має попередніх і містить посилання на всі інші сторінки. Навігація спрямована згори вниз із посиланнями вгору за ієрархією та навпаки. Цей спосіб найпоширеніший. Така структура дає

відвідувачам можливість легкого пошуку інформації на сайті від загальної інформації до більш конкретної (рис. 2.2.) [8].

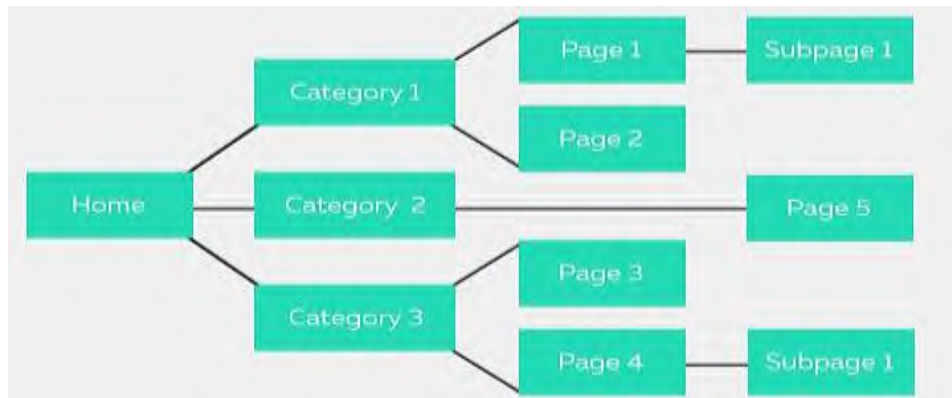


Рисунок 2.2 – Схема ієрархічної (деревоподібної) структури вебсайту

Лінійна, або послідовна, або каскадна структура складається із послідовно розміщених сторінок одна за одною у вигляді, так званого ланцюга. За даним способом користувачі можуть переміщуватися тільки в одному напрямку – послідовно, тому така структура зручна для створення невеликого сайту із декількох сторінок, або для реєстрації користувачів на інтернет-ресурсі. Таке розміщення інформації ефективно для представлення тексту у послідовному порядку, адже все відбувається за конкретним маршрутом [28]. Схематично лінійну структуру подано на рис. 2.3.

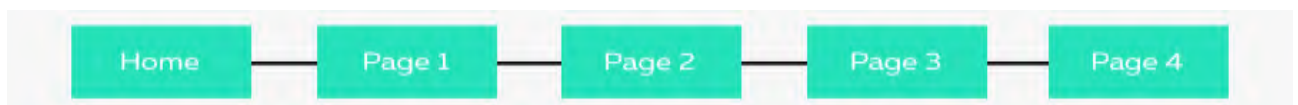


Рисунок 2.3 – Схема лінійної структури сайту

Гратчаста, або таблична, структура полягає у швидкій навігації між вертикальними та горизонтальними сторінками без переходу на проміжні сторінки, найчастіше застосовується для створення каталогів.

Змішана структура полягає у побудові системи навігації сайту, із поєднанням двох або більше компонентів, а саме ієрархічної та лінійної структур. Така структура доволі складна, адже всі сторінки розміщені на окремих розгалуженнях.

Також на етапі розробки слід підібрати матеріал для розміщення на сторінках сайту та обрати програмні засоби, за допомогою яких буде розроблено сайт. Після чого відбувається наповнення сайту матеріалами та створення гіперпосилань для зручної навігації по сайту. При цьому вся інформація повинна бути унікальною та грамотною із використанням графіки та анімацій [8]. На рис. 2.4 показана UML-діаграма варіантів використання майбутнього вебсайту.



Рисунок 2.4 – UML-діаграма варіантів використання фільмотеки

Діаграма (див. рис. 2.4) відображає функціональні можливості вебдодатку з точки зору користувача. Користувач може переглядати інформацію про фільм, акторський склад, прем'єри, огляди та рейтинг фільмів, а також здійснювати перехід до зовнішнього сервісу YouTube для перегляду трейлерів. Діаграма демонструє структуру функціональних можливостей системи та взаємозв'язки між окремими варіантами використання.

У процесі проєктування вебдодатку для фільмотеки було обрано архітектуру односторінкового вебресурсу, тобто лендінгу, в якому основні

тематичні розділи розташовані в межах одного HTML-документа та відображаються за допомогою механізмів динамічного керування інтерфейсом засобами JavaScript. Такий підхід дозволяє мінімізувати кількість переходів між сторінками та забезпечити швидку взаємодію користувача із системою.

Під час проєктування вебдодатку для фільмотеки було використано модульний підхід, відповідно до якого інтерфейс поділено на логічно незалежні функціональні блоки. Такий підхід забезпечує зручність подальшого супроводу програмного коду та спрощує розширення функціональних можливостей системи.

У результаті всіх етапів створено проєкт вебсайту, за допомогою якого можна отримати уявлення про очікувані результати, приблизний кошторис, а отже контролювати подальше створення сайту і досягти поставлених цілей.

2.3 Формування стеку технологій з розроблення вебсайту фільмотеки та створення макету

Перед розробкою вебсайту після визначення його основної ідеї, цільової аудиторії, функціональності слід обрати відповідне програмне забезпечення, на основі якого і в середовищі якого буде створено сайт. Всі методи розробки сайтів можна умовно розділити на дві групи: ручне та автоматизоване [29].

Автоматизований метод полягає у створенні вебсайту за допомогою спеціальних конструкторів або систем керування контентом. Він передбачає поділ структури сайту на дизайн та контент [29].

Конструктори сайтів – це онлайн-системи, що створюють сайт із типового набору модулів і компонентів із розміщенням його у мережі. Часто конструктори сайтів генерують HTML код, який майже не можливо відредагувати і він займає багато місця, а отже сторінки будуть довго завантажуватися. Такий метод підходить для створення простих сайтів, але він не гарантує розробку унікального сайту, а деякі можливості можуть бути обмежені, платними.

Ручне написання сайтів здійснюється у текстових або спеціальних редакторах HTML та CSS кодів із підтримкою використання JavaScript, PHP і т. ін. Текстові редактори поділяють на редактори текстового режиму і віконні редактори. Віконні редактори виглядають більш сучасно і володіють деякими корисними елементами користувацького інтерфейсу. Створення та редагування може здійснюватися у віконних редакторах, наприклад, Sublime Text, Notepad ++, TextPad тощо. Цей процес досить складний та трудомісткий, адже він вимагає навичок в області вебпрограмування і вебдизайну.

Для реалізації завдань практичної частини кваліфікаційної роботи було обрано саме ручний метод, адже він дає можливість вибору конструкції вебресурсу. В якості середовища проєктування обрано продукт компанії Microsoft Visual Studio Code, або VS Code, який є одним із найпопулярніших інтегрованих середовищ для веброзробки (IDE). Він пропонує широкий спектр функціональних можливостей та розширень, які допомагають веброзробникам підвищити продуктивність та зручність роботи. Для вибору ідеального IDE доречно ознайомитися з думками професійних розробників, аналітиків [30].

Однією з основних переваг VS Code є його інтеграція з багатьма популярними інструментами та технологіями веброзробки. Наприклад, VS Code має підтримку для мов програмування, таких як HTML, CSS, JavaScript, Python та інших [31]. Він також підтримує інтеграцію з Git, що дозволяє веброзробникам легко керувати версіями свого коду та співпрацювати з іншими розробниками. Завдяки інтеграції IDE програмісти можуть мати той самий набір можливостей в одному місці без необхідності постійно перемикати інструменти [31]. Для веброзробників і програмістів існує широкий набір програмних засобів від редакторів коду до написання складних додатків на мовах Java, PHP та ін.

Крім того, VS Code пропонує багато корисних функцій, які спрощують написання коду та зменшують кількість помилок. Наприклад, він має автодоповнення коду, що допомагає веброзробникам швидше та точніше писати код. Також VS Code має можливість відлагодження коду, що дозволяє розробникам швидко виявляти та виправляти помилки в своєму коді.

Середовище має широкий спектр розширень та плагінів, що дозволяє розширити його функціональність та адаптувати його під потреби веб-розробника. Наприклад, можна встановити розширення для роботи з фреймворками, такими як React, Angular або Vue, або для роботи з базами даних та іншими інструментами.

Для наочного представлення майбутнього сайту зручно використовувати наперед створений макет. Макет сайту – це його реалістичний прототип, створений у графічному редакторі. Він має вигляд готового сайту із всіма блоками, зображеннями, кнопками і тому подібне. Єдина відмінність у тому, що макет не має функціонального наповнення, тобто містить тільки шаблони описів. Він потрібен для скорочення процесу створення сайту, адже при наявності макету вебпрограмісту легше розробляти сайт із мінімальними дефектами [32].

Можливо виділити основні переваги наявності макету:

- допомагає скласти наочне уявлення про вигляд майбутнього сайту, побачити, як будуть виглядати елементи на сторінках;
- розробити єдиний дизайн для всіх сторінок сайту;
- продумати наповнення елементів текстом, зображенням, відео тощо;
- задати чіткі границі для роботи над створенням сайту.

Після розгляду всіх переваг можна перейти до розробки макету. Вся робота відбувається поетапно. Починається із створення ТЗ, в якому слід указати цілі сайту, короткий аналіз цільової аудиторії, користувацький сценарій, кількість сторінок та затвердити замовником [32]. Порядок розробки та зміст технічного завдання відповідає ДСТУ 34.601–90.

Кожен інтернет-ресурс, від домашньої сторінки до великого інформаційного порталу, містить декілька тематичних рубрик, сполучених між собою гіперпосиланнями. Якщо тематичні рубрики містять власні підрозділи, кожна з них також має свою стартову сторінку.

Зовнішня структура сайту містить в собі розташування видимих блоків на ньому, а саме: шапку, яка має назву ресурсу, логотип та навігаційне меню для переходів між розділами сайту, блоки контенту із текстом, відео, зображенням,

службові форми та footer (підвал сайту), в якому розміщені контактні дані та інформація про авторські права (рис. 2.5). Крім таких блоків, на сайті може бути розміщена реклама або посилання на інші інтернет-ресурси. Розробка зовнішньої структури тісно пов'язана з програмною реалізацією сайту.

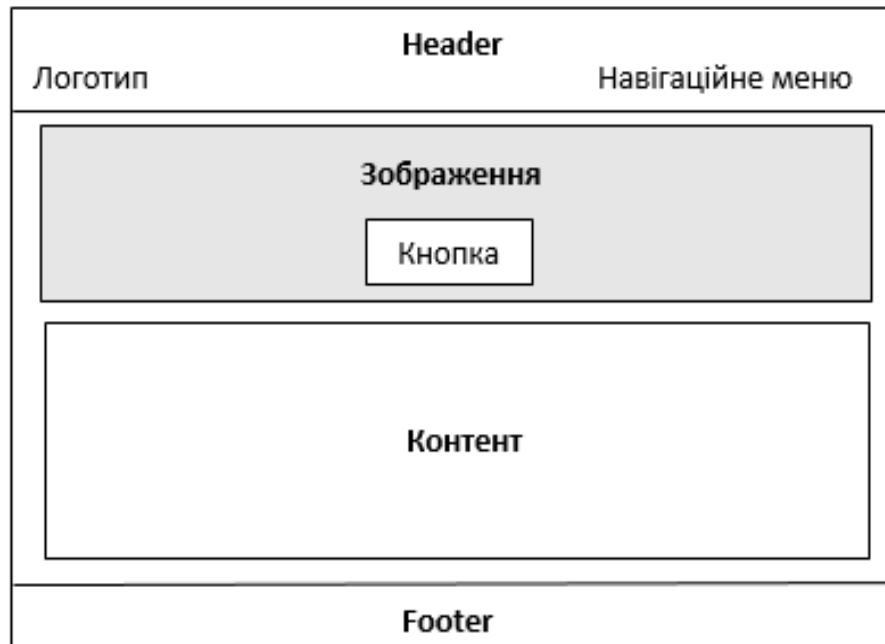


Рисунок 2.5 – Макет зовнішньої структури сайту у формі лендінгу

Макет вебсайту забезпечує зручну основу для подальшої розробки та дизайну вебсторінки. Для розробки макету використовують різноманітні графічні редактори, наприклад Adobe Photoshop, Wix, Figma та інші, за допомогою яких макет створюється досить швидко. Для макету розроблюваного вебсайту було обрано редактор Figma.

У програмі доступні різноманітні інструменти для створення фігур, ліній та інших елементів дизайну. Крім того, Figma пропонує можливість створювати векторні зображення та використовувати різні типи шрифтів.

Окрім роботи з графікою, Figma забезпечує зручну роботу зі структурою документа, дозволяє створювати різні елементи дизайну та організовувати їх у вигляді шарів. Крім того, Figma дозволяє працювати зі створеними документами в команді для роботи над проєктами великих масштабів. Фрагмент початку роботи над макетом показано на рис. 2.6.



Рисунок 2.6 – Робота над макетом у редакторі Figma

Однією з найважливіших функцій Figma є можливість відстежувати зміни, що вносяться до документу. Це дозволяє ефективно працювати в команді та швидко вирішувати проблеми, пов'язані зі змінами в дизайні. Також варто зазначити, що Figma доступний як для використання в браузері, так і в вигляді додатку для робочого столу. Це дозволяє працювати з програмою в будь-яких умовах та на будь-яких пристроях.

Програму Figma обрано завдяки тому, що вона не лише має інструментарій візуального створення елементів макету вебсторінки, але й спеціальний плагін «Figma to html» [33], який допомагає легко перетворити ваш дизайн Figma на код, не втрачаючи жодної частини дизайну чи кольорів. Вигляд головної сторінки, попередньо спроектованої у Figma, показано на рис. 2.7.

При проєктуванні інтерфейсу було обрано концепцію темного дизайну (Dark Theme), яка традиційно використовується на ресурсах кінематографічної тематики. Основний фон представлений темними відтінками сірого та чорного кольорів, тоді як золотисті елементи виконують роль акцентів та допомагають виділяти ключові елементи навігації й заголовки. Така кольорова схема асоціюється з атмосферою кінотеатрів та преміальних кінопорталів. Для головного банера обрано заставку популярно фільму «Дюна».

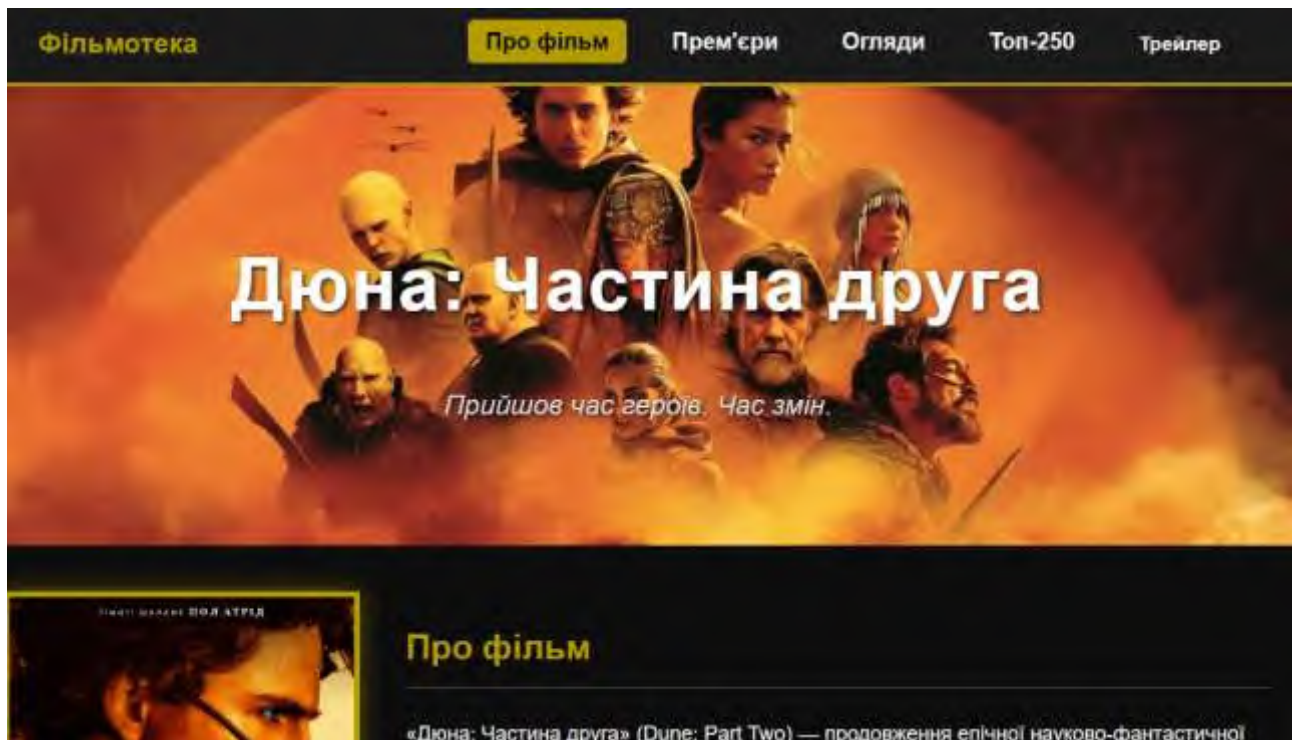


Рисунок 2.7 – Вигляд частини верхньої секції вебсайту, спроектованого у Figma

Навігаційна система побудована за принципом горизонтального меню, яке розміщується у верхній частині сторінки. Для підвищення зручності користування використано механізм фіксованої шапки (sticky navigation), завдяки чому меню залишається доступним під час прокручування сторінки.

Для покращення орієнтації користувача реалізовано:

- підсвічування активного пункту меню;
- ефекти наведення курсора;
- візуальне групування контенту за тематичними секціями;
- плавні переходи між елементами інтерфейсу.

Загалом, розроблено односторінковий сайт з описом унікальних властивостей органічної продукції, вибрані відповідні графічні елементи, розроблені логотипи. У процесі створення макету сайту було враховано його завдання, специфіку предметної області, цільову аудиторію та враження, які він буде справляти на відвідувача. Наступним етапом є його технічна, програмна реалізація.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ВЕБСАЙТУ ДЛЯ ФІЛЬМОТЕКИ

3.1 Створення HTML-структури та технічні рішення інтерфейсу користувача

Для розробки вебсайту для фільмотеки було обрано мову розмітки HTML5 та технологію CSS3, а також мову програмування JavaScript в його чистому вигляді. HTML5 використовується для формування семантичної структури та каркаса вебсторінки, тобто дозволяє створити і структурувати розділи, заголовки, абзаци, посилання та блоки вебдодатку, визначає загальну структуру та організацію контенту, що сприяє покращенню користувацького досвіду та забезпечує більш ефективну роботу з вебсайтом [34].

CSS3 застосовується для візуального оформлення, організації адаптивних сіток (Flexbox, Grid) та забезпечення плавних UX-переходів (Transitions).

JavaScript (Vanilla JS) використовується для забезпечення інтерактивності, динамічного керування станом інтерфейсу (ввімкнення/вимкнення видимості блоків) та обробки подій користувача.

Проект має модульну структуру, розділену за призначенням файлів, що спрощує його підтримку та масштабування в майбутньому (рис. 3.1).

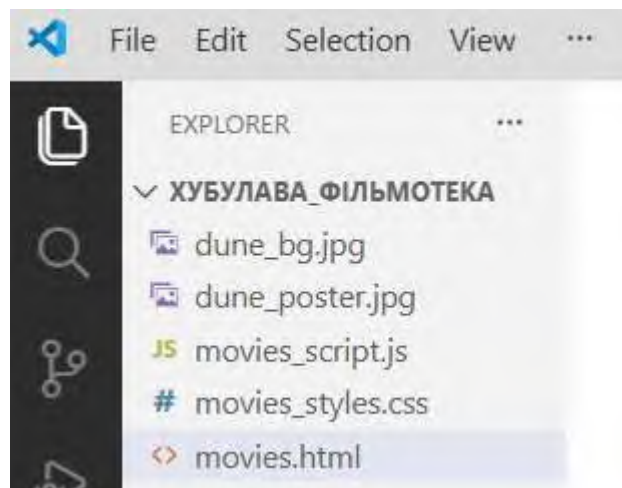


Рисунок 3.1 – Структура проєкту в середовищі Visual Studio Code

Файл `movies.html` – вхідна точка додатка, яка містить повну семантичну розмітку всіх інформаційних блоків вебсайту, визначає загальну розмітку та розташування основних елементів, таких як заголовки, текст, зображення, навігаційні меню, футер та інші (іноді називають «скелет» сайту), представлених семантичними елементами. Повний лістинг коду HTML показано в додатку Б.

Частина між тегами `<head>...</head>` є головною частиною з управляючими елементами: метатеги, посилання на зовнішні файли стилів, призначений для зберігання службової інформації про всю сторінку (рис. 3.2).

```

1  <!DOCTYPE html>
2  <html lang="uk">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Фільми: Дюна: Частина друга</title>
7      <link rel="stylesheet" href="movies_styles.css">
8  </head>
9  <body>
10
11      <header class="page-header">
12          <div class="logo">Фільмотека</div>
13          <nav class="main-nav">
14              <ul>
15                  <li><a href="#" id="home-button" class="active">Про фільм</a></li>
16                  <li><a href="#" id="premieres-button">Прем'єри</a></li>
17                  <li><a href="#" id="reviews-button">Огляди</a></li>
18                  <li><a href="#" id="top250-button">Топ-250</a></li>
19                  <li><button id="trailer-button">Трейлер</button></li>
20              </ul>
21          </nav>
22      </header>

```

Рисунок 3.2 – Головна частина вебсайту в `index.html` в редакторі VSC

Основними частинами, які описуються тегами HTML є:

- хедер із навігаційним горизонтальним меню - чотирма пунктами і кнопкою, описаним семантичним тегами `<header>`, `<nav>` та посиланнями `<a href="#">...</a>`;
- змістові секції у вигляді окремих блоків, які мають кілька рівнів вкладеності - `<main>` та зовнішній блок створений тегом `<section>...</section>`;
- заголовок сайту (в тезі `<header>` розташований головний банер) та кнопка виклику повідомлення;

- зона контенту, яка може включати один або більше блоків тексту та зображень;
- футер сайту, який містить інформацію про копірайт та інші важливі посилання, контактні дані, додаткові посилання.

Властивості всіх блоків винесені в окремий зовнішній файл. `movies_styles.css` - каскадна таблиця стилів, що відповідає за дизайн-систему проекту, палітру кольорів, типографіку та адаптивне розташування елементів.

Кожна секція і вкладені блоки мають відповідний стиль в селекторах `id` або `.class`. Для раціонального запису стилів сформована певна система. Наприклад, кожний вкладений блок першого рівня в секції `<main class="content-container">` описується стилем `<div id="movie-details">` та `<section class="movie-section flex-details">` та інші вкладені класи. На сторінці заплановано кілька різних класів секцій. На рис. 3.3 показана семантична ієрархічна структура в частині головного розділу. Для компактності вкладені блоки нижчого порядку згорнуті.

```

29 | <main class="content-container">
30 |   <div id="movie-details">
31 | >   <section class="movie-section flex-details">...
42 |   </section>
43 |   <hr>
44 | >   <section class="cast-section"> ...
52 |   </section>
53 |   </div>
54 | >   <div id="premieres-content"
69 |   </div>
70 | >   <div id="reviews-content" class="hidden"> ...
84 |   </div>
85 | >   <div id="top250-content" class="hidden"> ...
129 |   </div>
130 | </main>

```

A space-separated list of the classe
elements via the class selectors or
MDN Reference

Рисунок 3.3 – Частитна коду HTML, в чкому показана блочна структура сторінки та ієрархія

Перейшовши до файлу стилів, знайдемо опис властивостей вказаних блоків (рис. 3.4) в класах `.content-container`, `.hero` та інших, а також ідентифікаторах. Особливу увагу привертають каскадність в дочірніх класах.

```

30  .main-nav ul {
31      list-style: none;
32      margin: 0;
33      padding: 0;
34      display: flex;
35  }
36
37  .main-nav li {
38      margin-left: 25px;
39  }

```

```

85  .content-container {
86      max-width: 1200px;
87      margin: 0 auto;
88      padding: 0 20px;
89  }

```

Рисунок 3.4 – Визначення стилів всіх основних блоків класу .content-container і дочірніх елементів .main-nav

Повний лістинг файлу стилів наведено в додатку В. Серед інших оригінальних рішень варто зупинитися на таких:

- CSS-компонувальники: flexbox (display: flex) використано для побудови верхньої панелі навігації (.page-header), гнучкого карткового представлення інформації про фільм (.flex-details) та вирівнювання елементів у блоці "Героя";
- CSS Grid (display: grid) використано для секції акторського складу (.cast-grid). Властивість grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)) забезпечує автоматичну адаптивність: картки акторів самостійно перебудовуються в кілька рядів залежно від ширини екрана користувача;
- ефекти занурення (UX/UI): шапка сайту зафіксована при скролінгу завдяки position: sticky; top: 0.
- для інтерактивних елементів (посилань, кнопок, карток акторів) налаштовано плавний перехід властивостей (transition: 0.3s) при наведенні курсору (:hover).

Фрагмент опису зміни кнопки при наведенні показано на рис. 3.5.

```

53  .main-nav a:hover, #trailer-button:hover {
54      color: ■ #0d0d0d;
55      background-color: ■ #b39b00;
56  }

```

Рисунок 3.5 – Опис динамічних функцій CSS3 для окремих елементів у Visual Studio Code

Часто буває необхідно якимось чином трансформувати візуальне подання елемента (масштабувати, повернути, перемістити) і при цьому ніяк не торкнутися сусідніх елементів у документі. Для таких перетворень використовується властивість `transform`. Як значення виступають різні функції трансформації: `rotate`, `translate`, `scale`, `skew` та інші. Як видно з рис. 3.5, застосовано спеціальні функції трансформації блоків при наведенні мишею:

- властивість `transition: transform 0.3s;` - використовується для того, щоб плавно змінити CSS-властивості між двома станами елемента;
- функція `transform: translateY(-10px);` - використана для зміщення елемента вздовж вісі Y (по вертикалі).

Є низка важливих відмінностей: елемент позиціонується щодо відповідних сторін батька. Тобто `left: 20px` помістить елемент на 20 пікселів щодо лівої межі батька, а `translate(20px)` помістить елемент праворуч щодо того місця, де той перебував до трансформації. В цілому, `position` та `translate` чудово поєднуються один з одним. Позиціонування краще використовувати для початкового розташування елемента, а `translate` застосовувати, щоб додати анімацію.

Сайт спроектовано як комплексну інформаційну панель фільмотеки, що розділена на 4 функціональні секції:

1. Про фільм (`#movie-details`): містить постер, текстовий опис сюжету кінострічки «Дюна: Частина друга» та інтерактивну сітку з переліком акторів.
2. Прем'єри (`#premieres-content`): анонси майбутніх кіноновинок із зазначенням точних дат виходу та коротких синопсисів.

3. Огляди (#reviews-content): Модуль відображення рецензій від критиків із рейтинговою системою (картки оцінок).

4. Топ-250 (#top250-content): Табличний модуль (.top-list-table), що структурує дані про найкращі фільми за версією IMDb.

Приклад закріплення панелі навігації, зміни кольору активованої кнопки Топ-250 та при наведенні з переходом до відповідної секції, показано на рис. 3.6.

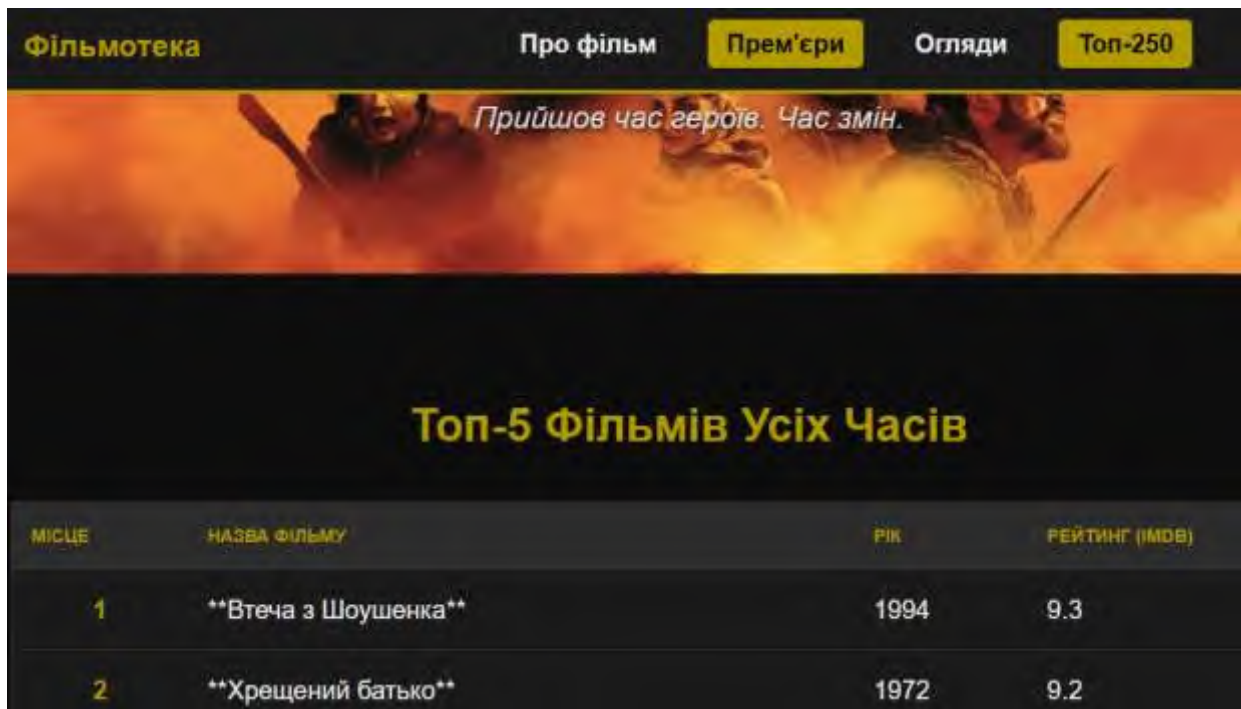


Рисунок 3.6 – Приклад перходу до середнього блоку в секції та активація кнопки

Таким чином, застосування таких засобів CSS3 як Flexbox, Grid та механізмів інтерактивної взаємодії, дозволило створити зручний користувацький інтерфейс, адаптований до вимог сучасних вебресурсів кінематографічної тематики. Стилзація сторінки є важливою частиною веброзробки, оскільки вона визначає зовнішній вигляд відображення вмісту вебсайту, досягнення дизайну інтерфейсу (UI-дизайн) та дизайну поведінки при використанні (UX-дизайн). Для цього в роботі використані такі CSS-стили, які дозволяють встановлювати різні параметри дизайну: такі як кольори, розміри, форми, типографіку та інші.

3.2 Роль JavaScript у реалізації функціональності вебдодатку «Фільмотека»

Мова програмування JavaScript є одним із ключових компонентів сучасної веброзробки та відіграє визначальну роль у створенні інтерактивних вебдодатків. На відміну від HTML, який відповідає за структуру документа, та CSS, що забезпечує його візуальне оформлення, JavaScript дозволяє реалізувати динамічну поведінку вебсторінок, реагувати на дії користувача та змінювати вміст сторінки без її повторного завантаження.

У розробленому вебдодатку «Фільмотека» JavaScript використовується для реалізації механізмів навігації, керування відображенням інформаційних блоків, обробки подій користувача та забезпечення інтерактивної взаємодії з контентом. Застосування сценаріїв JavaScript дозволило перетворити статичний вебсайт на інтерактивний інформаційний ресурс, який оперативно реагує на дії відвідувача та забезпечує більш комфортний користувацький досвід (рис. 3.7).

```
function toggleSections(sectionToShow, activeLink) {  
  movieDetailsSection.classList.add('hidden');  
  premieresSection.classList.add('hidden');  
  reviewsSection.classList.add('hidden');  
  top250Section.classList.add('hidden');  
  
  sectionToShow.classList.remove('hidden');  
  
  document.querySelectorAll('.main-nav a').forEach(a => a.classList.remove('active'));  
  if (activeLink) {  
    activeLink.classList.add('active');  
  }  
}
```

Рисунок 3.7 – Частина коду JS для динамічної навігації між розділами

Одним із основних завдань JavaScript у проєкті є реалізація системи навігації між функціональними розділами вебдодатку. Для цього використовується механізм динамічного керування видимістю окремих секцій сторінки. Після вибору користувачем певного пункту меню сценарій приховує всі інші розділи та відображає лише необхідний контент. Такий підхід дозволяє

організувати роботу вебдодатку за принципом односторінкового застосунку (Single Page Application), де взаємодія з інформацією здійснюється без перезавантаження сторінки.

Важливою перевагою використання JavaScript є можливість безпосередньої роботи з об'єктною моделлю документа DOM (Document Object Model). Завдяки цьому сценарії отримують доступ до елементів HTML-структури, можуть змінювати їхній вміст, зовнішній вигляд та стан. У вебдодатку «Фільмотека» ця можливість використовується для керування класами елементів, зміни активних пунктів меню та динамічного відображення інформаційних секцій (рис. 3.8).

```
navLinks.forEach(link => {  
    link.addEventListener('click', (e) => {  
        e.preventDefault();  
  
        const sectionId = link.getAttribute('data-section');  
        toggleSections(sectionId);  
    });  
});
```

Рисунок 3.8 – Код обробки подій навігаційного меню

Значну роль JavaScript відіграє в реалізації інтерактивних елементів інтерфейсу. Для всіх навігаційних пунктів меню реалізовано обробку подій натискання, що забезпечує миттєвий перехід між розділами вебресурсу. Крім того, сценарії контролюють активний стан меню, що дозволяє користувачеві легко визначити поточний розділ вебдодатку.

Окремим напрямом використання JavaScript є реалізація взаємодії з мультимедійним контентом. У розробленому вебдодатку передбачено можливість перегляду трейлера фільму. Після натискання відповідної кнопки користувач отримує інформаційне повідомлення про перехід до відеоматеріалу, після чого автоматично відкривається зовнішній відеосервіс YouTube із

трейлером стрічки. Такий механізм забезпечує інтеграцію вебдодатку із зовнішніми мультимедійними платформами та розширює його функціональні можливості (рис. 3.9).

```
//Обробник для кнопки "Трейлер"
trailerButton.addEventListener('click', function () {
    alert('Зараз відбудеться перенаправлення на сторінку з трейлером "Дюна: Частина друга"!');
    window.location.href = 'https://www.youtube.com/watch?v=DtR76pz517E';
});
```

Рисунок 3.9 - приклад коду інтеграції із зовнішніми сервісами

За допомогою JavaScript реалізовано інтеграцію вебдодатку із зовнішнім відеосервісом. Після натискання кнопки користувач отримує повідомлення та автоматично переходить до перегляду трейлера на платформі YouTube.

Ще одним прикладом використання JavaScript є реалізація подій взаємодії з елементами акторського складу. Для кожного елемента передбачено обробку наведення курсора та натискання миші. У результаті користувач отримує додатковий зворотний зв'язок від системи у вигляді повідомлень про вибраного актора. Подібний підхід демонструє можливість створення інтерактивних інформаційних карток та покращує сприйняття контенту.

```
actorCards.forEach(actor => {

    actor.addEventListener('mouseenter', () => {
        console.log(`Наведено на ${actor.textContent}`);
    });

    actor.addEventListener('click', () => {
        alert(`Ви вибрали ${actor.textContent}`);
    });

});
```

Рисунок 3.10 – Приклад роботи з DOM та подіями користувача.

Наведений код демонструє використання JavaScript для обробки подій наведення курсора та натискання миші. Завдяки цьому реалізовано інтерактивну

взаємодію користувача з інформаційними картками акторів. Використання JavaScript також сприяє підвищенню продуктивності вебдодатку. Оскільки більшість операцій виконується без повного перезавантаження сторінки, скорочується обсяг передаваних даних та прискорюється відображення інформації. Це позитивно впливає на швидкодію системи та зручність роботи користувача.

З точки зору архітектури програмного забезпечення застосування JavaScript дозволило реалізувати розподіл відповідальності між структурою, оформленням та логікою роботи вебдодатку. Такий підхід відповідає сучасним принципам веброзробки та забезпечує спрощення подальшої підтримки і модернізації програмного коду.

Таким чином, JavaScript виступає центральним інструментом реалізації інтерактивності вебдодатку «Фільмотека». Саме завдяки використанню цієї мови програмування забезпечується динамічна навігація між розділами, обробка дій користувача, інтеграція мультимедійного контенту та створення зручного користувацького інтерфейсу. Застосування JavaScript дозволило значно підвищити функціональність вебресурсу та забезпечити його відповідність сучасним вимогам до інтерактивних вебдодатків.

Готовий сайт вимагає тестування для перевірки функціонування та виправлення помилок. На цьому етапі необхідно перевірити роботу всіх гіперпосилань, зручність використання, відображення всіх графічних та анімаційних елементів тощо.

Для цього існує декілька видів тестування вебсайтів [35]:

1. тестування функціональності;
2. перевірка зручності використання;
3. Тестування продуктивності;
4. Тестування безпеки.

Функціональне тестування спрямоване на перевірку коректної роботи функціональності сайту, тобто коректність реалізації функціональних вимог. Включає перевірку роботи всіх посилань, форм, різноманітних функцій [35].

Тестування зручності використання спрямоване на дослідження того, наскільки зручно і зрозуміло користуватися сайтом кінцевим клієнтам. Основними характеристиками є: простота навчання, навігація, задоволення користувача, загальний вигляд. Тобто, сайт повинен бути зручним і логічним у використанні, послідовним, головне меню розміщувалося на всіх сторінках [36].

Тестування інтерфейсу користувача спрямоване на перевірку графічного інтерфейсу або його компонентів і включає перевірку інтерфейсу кнопок, формат і шрифт тексту, зображень та інтерфейсу всього сайту. На цьому етапі здійснюється перевірка на сумісність із різними інтернет-браузерами [36].

Тестування продуктивності полягає у дослідженні показників швидкості реакції сайту на різні навантаження. При такому тестуванні перевіряється яка кількість користувачів може взаємодіяти із однією сторінкою, на скільки сайт може обробляти одночасні запити користувачів і проводиться перевірка поведінки сайту при змінах навантаження та інше [37].

Тестування безпеки спрямоване на перевірку здатності сайту протистояти зловмисним спробам отримання доступу до даних без прав доступу. Полягає у пошуку всіх слабких сторін, які можуть призвести до втрати даних.

У результаті проведення всіх тестувань повинен бути створений документ.

Наступний етап – впровадження. На даному етапі необхідно обрати домен сайту, який буде легко запам'ятовуватися, та обрати компанію, яка надає послуги з хостингу. Вже готовий сайт може розміщуватися на: власному сервері, сервері провайдера, на сервері організації з якою співпрацює підприємство, або безкоштовному сервері [38]. Процес розміщення сайту в інтернеті передбачає надання йому адреси.

Реєстрація доменного імені дозволяє створити не лише бренд, але і дає можливість просування в мережі. Отже, для якісного просування сайту воно повинно легко запам'ятовуватися, бути коротким, або означати сферу діяльності [39].

Останній етап створення сайту це його підтримка. Для ефективної роботи сайту необхідно слідкувати за актуальністю контенту на сайті, вносити зміни до

його оформлення. Також на цьому етапі відбувається оптимізація сайту після внесених змін та під пошукові запити користувачів.

3.3 Економічне оцінювання вартості проєкту

Проведемо економічну оцінку і розрахунки щодо вартості виконання робіт з проєктування та розроблення такого типу вебсайту, а також хостингу. Зазвичай, оцінюють вартість створення програмного продукту через трудомісткість робіт та витрати на оплату праці розробника [40]. Оскільки у вашому проєкті використовуються безкоштовні технології (HTML5, CSS3, JavaScript, Visual Studio Code, GitHub Pages або інший безкоштовний хостинг) і прямі витрати на придбання програмного забезпечення відсутні, основною статтею витрат буде робота розробника. Основні етапи виконання робіт наведено в табл. 3.1.

Таблиця 3.1 – Оцінювання трудомісткості розробки

№ з/п	Етап робіт	Тривалість, год
1	Аналіз предметної області	10
2	Проєктування структури вебдодатку	12
3	Розробка HTML-розмітки	16
4	Розробка CSS-оформлення	18
5	Програмування JavaScript-функціоналу	20
6	Наповнення контентом	8
7	Тестування та виправлення помилок	10
	Документування проєкту	6
	Разом тудомісткість, год	100

Оцінка вартості розробки сайту у спеціалізованих компаніях залежить від кількості робочих годин і тарифу, встановленого за 1 годину роботи [41]. Приймемо середню погодинну оплату праці веброзробника на рівні 350 грн/год.

Тариф залежить від складності сайту і кваліфікації розробника. Наприклад, створення інтерактивного сайту-лендінгу із 6 секцій із використанням

динамічних ефектів може займати приблизно 60 годин роботи. Вартість години роботи фрілансерів або вебстудій в Україні оцінюється в діапазоні 10-20 доларів США [41].

Основна заробітна плата буде розрахована за формулою:

$$Z_{осн} = T \times C_{год}, \quad (3.1)$$

де $Z_{осн}$ - заробітна плата основного працівника;

T – трудомісткість;

$C_{год}$ – погодинна ставка оплати праці.

З урахуванням трудомісткості основна заробітна плата становитиме:

$$Z_{осн} = 100 \times 350 = 35000 \text{ грн.}$$

Накладні витрати $B_{накл}$ приймаються на рівні 20 % від основної заробітної плати:

$$B_{накл} = 35000 \times 0,2 = 7000 \text{ грн.}$$

Загальна собівартість розробки становитиме:

$$C = 35000 + 7000 = 42000 \text{ грн.}$$

За наведеним алгоритмом до собівартості розробки можуть бути додані витрати за електроенергію, амортизацію обладнання, рекламу та інші. Це збільшить вартість на 25% і отже становитиме: $C = 42000 \times 1,25 = 52500 \text{ грн.}$

Розраховані дані носять оціночний наближений характер, виконані за прийнятими методиками, але можуть бути уточнені в конкретних випадках.

Розроблений вебдодаток дозволяє створити сучасний інформаційний ресурс для популяризації кінематографічного контенту та організації взаємодії користувачів із ним. Використання безкоштовних технологій дало змогу суттєво знизити витрати на розроблення програмного продукту. При цьому створений вебдодаток має завершену структуру, реалізує інтерактивну навігацію, підтримує мультимедійний контент та може бути використаний як основа для подальшого розвитку повноцінного кінематографічного вебпорталу. Таким чином, отримана договірна вартість близько 52,5 тис. грн є економічно обґрунтованою та відповідає обсягу виконаних робіт і сучасним умовам веброботки.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено функціональний оригінальний вебдодаток для фільмотеки і отримано такі результати:

1. Проведено аналіз сучасного стану розвитку інформаційного суспільства та визначено роль вебтехнологій у задоволенні інформаційних потреб користувачів. Встановлено, що вебресурси кінематографічної тематики є важливим засобом поширення інформації про кіноіндустрію, популяризації кіномистецтва та організації взаємодії між шанувальниками кіно.

2. Досліджено еволюцію базових вебтехнологій від перших версій HTML до сучасного технологічного стеку HTML5, CSS3 та JavaScript. Показано їхню роль у створенні адаптивних та інтерактивних вебдодатків.

3. Виконано аналіз сучасних JavaScript-бібліотек та фреймворків, визначено їхні переваги та особливості використання у веброзробці. Встановлено, що для створення інтерактивних вебресурсів ключове значення мають засоби динамічної взаємодії з користувачем, реалізовані мовою JavaScript.

4. Проведено аналіз функціональних можливостей існуючих вебресурсів кінематографічної тематики та сформовано вимоги до розроблюваного вебдодатку. Визначено основні функціональні модулі системи, орієнтовані на забезпечення комплексного доступу до інформації про кіноіндустрію.

5. Розроблено структуру інтерактивного вебдодатку для фільмотеки, яка поєднує інформаційні, мультимедійні та комунікаційні можливості в межах єдиного вебресурсу. Запропонована структура забезпечує логічну організацію контенту та зручну навігацію між функціональними розділами.

6. Спроектовано користувацький інтерфейс вебдодатку на основі принципів сучасного UX/UI-дизайну. Використання тематичного темного оформлення з контрастними акцентами дозволило сформувати візуальне середовище, що відповідає тематиці кіноіндустрії та покращує сприйняття контенту користувачами.

7. Розроблено інтерактивний вебдодаток «Фільмотека» із використанням HTML5, CSS3 та JavaScript, який забезпечує доступ до інформації про фільми, акторський склад, майбутні прем'єри, огляди та рейтинг популярних стрічок.

8. Реалізовано динамічну навігацію між функціональними розділами без перезавантаження сторінки, що дозволило застосувати принципи односторінкового вебдодатку та підвищити швидкодію користувацького інтерфейсу.

9. Розроблено механізми інтерактивної взаємодії користувача з контентом, зокрема обробку подій навігаційного меню, інтерактивні елементи акторського складу та інтеграцію із зовнішніми мультимедійними ресурсами для перегляду трейлерів.

До оригінальних результатів роботи слід віднести створення компактного інтерактивного вебдодатку, який поєднує в єдиному інформаційному просторі декілька різних категорій кінематографічного контенту: відомості про фільми, інформацію про акторів, новини прем'єр, огляди, рейтинги та мультимедійні матеріали. На відміну від багатьох великих кінопорталів, розроблений вебресурс орієнтований на простоту використання, швидкий доступ до інформації та відсутність перевантаження інтерфейсу другорядними функціями.

Виконано економічне оцінювання розробленого програмного продукту. Встановлено, що використання безкоштовних технологій та середовищ розробки дозволяє суттєво знизити витрати на створення вебдодатку при збереженні необхідного рівня функціональності та якості реалізації.

Отже, поставлена мета роботи досягнута повністю, а розроблений вебдодаток для фільмотеки може використовуватися як самостійний інформаційний ресурс для шанувальників кіно, а також як основа для подальшого розвитку та розширення функціональних можливостей із використанням сучасних вебтехнологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штанько В. І., Бордюгова Т. Г. Інформаційне суспільство: соціально-філософські проблеми становлення: навч. посібник. Харків: ХНУРЕ, 2012. 172 с.
2. Актуальність наявності вебсайту. URL: <http://marianna2017.blogspot.com/2017/05/blog-post.html> (дата звернення: 27.04.2026).
3. Щедріна О.І., Агутін М.М. Інтернет-технології в бізнесі: навч. посіб. Київ: КНЕУ, 2019. 303 с.
4. Інтернет, соцмережі, стримінги та відео. Найцікавіше зі звіту Digital 2025 про взаємодію з цифровими технологіями. *Медіамейкер*. URL: <https://mediamaker.me/najczikavishe-zi-zvitu-digital-2025-pro-vzayemodiyu-z-czyfrovymy-tehnologiyamy-16257/> (дата звернення: 27.04.2026).
5. Digital 2025: Global Overview Report – DataReportal – Global Digital Insights. *DataReportal*. URL: <https://datareportal.com/reports/digital-2025-global-overview-report> (дата звернення: 27.04.2026).
6. Зоріна О.І., Фадєєнко В.Ю. Вплив маркетингових комунікацій на діяльність підприємств в сучасних умовах. *Економіка та управління підприємствами*. 2018. Вип. 21. С. 284-287.
7. О. А. Головка. Головка, О. А. Сайт вищого навчального закладу як елемент сучасної масової комунікації. *Обрії друкарства*. 2019. №1(7). С. 45–61. [https://doi.org/10.20535/2522-1078.2019.1\(7\).169546](https://doi.org/10.20535/2522-1078.2019.1(7).169546).
8. Трофименко О. Г., Козін О. Б., Задерейко О. В., Плачінда О. Є. Вебтехнології та веб-дизайн: навч. посіб. Одеса: Фенікс, 2019. 284 с.
9. Андріана Легкодух. Етапи розробки вебсайту. 10.04.2023. *FISH DIGITAL*: вебсайт. URL: <https://www.fishdigital.agency/etapi-rozrobki-saytu> (дата звернення: 27.04.2026).
10. Правила побудови HTML документів. URL: http://www.webnav.ua/books/html4/html_rules/ (дата звернення: 27.04.2026).

11. Історія розвитку HTML. URL: <https://www.ua5.org/web/40-storja-rozvitku-html.html> (дата звернення: 27.04.2026).
12. Memorandum of Understanding Between W3C and WHATWG. URL: <https://www.w3.org/2019/04/WHATWG-W3C-MOU.html> (дата звернення: 27.04.2026).
13. Що таке HTML5. *Hostinger*: вебсайт. URL: <https://www.hostinger.com.ua/rukovodstva/chto-takoe-html-i-ih-razlichiya> (дата звернення: 27.04.2026).
14. Що таке CSS.: вебсайт. URL: https://css.in.ua/article/shcho-take-html_10 (дата звернення: 27.04.2026).
15. CSS проти CSS3.: вебсайт. URL: <https://ru.photo-555.com/8189942-css-vs-css3> (дата звернення: 27.04.2026).
16. Що таке JavaScript.: вебсайт. URL: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript (дата звернення: 27.04.2026).
17. Що таке DOM.: вебсайт. URL: <https://proglib.io/p/what-is-dom/> (дата звернення: 27.04.2026).
18. Що таке HTML.: вебсайт. URL: <https://futurenow.com.ua/shho-take-html/> (дата звернення: 27.04.2026).
19. O. Kopishynska, I. Sliusar, V. Slyusar, Y. Utkin, O. Halych and O. Skryl. Features of Using Frameworks and Artificial Intelligence Language Models for JavaScript Code Optimization in Web Application Development. 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, 2024. pp. 1-7, doi: 10.1109/DESSERT65323.2024.11122152.
20. Framework vs Library: Full Comparison. *InterviewBit*. URL: <https://cutt.ly/ywT5tTHE> (дата звернення: 27.04.2026).
21. Majida Laaziri, Khaoula Benmoussa, Samira Khouli, Kerkeb Mohamed Larbi, Abir El Yamami. Analyzing bootstrap and foundation front–end frameworks: a

comparative study. *International Journal of Electrical and Computer Engineering (IJECE)*. 2019. Vol. 9, №. 1. P. 713–722.

22. Vue vs React vs... Svelte?! *Medium*. URL: <https://medium.com/@faulknerproject/vue-vs-react-vs-svelte-5f93d70d2618> (дата звернення: 27.04.2026).

23. Usage statistics and market shares of JavaScript libraries. *W3Techs*: вебсайт. URL: https://w3techs.com/technologies/overview/javascript_library (дата звернення: 27.04.2026).

24. Андріана Легкодух. Етапи розробки вебсайту. 10.04.2023. *FISH DIGITAL*: вебсайт. URL: <https://www.fishdigital.agency/etapi-rozrobki-saytu> (дата звернення: дата звернення: 27.04.2026).

25. Metacritic. URL: <https://www.metacritic.com/>.

26. Rotten Tomatoes. URL: https://www.rottentomatoes.com/browse/movies_at_home/ (дата звернення: 27.04.2026).

27. Поняття, структура та різновиди вебсайтів. Автоматизоване розроблення вебсайтів. URL: <http://www.ndu.edu.ua/liceum/web.pdf> (дата звернення: 27.04.2026).

28. Методи розробки Web-сайтів. URL: <https://sites.google.com/site/tz5103voinovakateryna/metodi-rozrobki-web-sajtiv> (дата звернення: 27.04.2026).

29. Розробка сайтів для бізнесу : вебсайт. URL: <https://pbb.lviv.ua/statti-i-novyny/statti-po-seo/prosuvannja-saytu-v-interneti-dlja-pochatkivciv/> (дата звернення: 27.04.2026).

30. Top IDE Index. URL: <https://pypl.github.io/IDE.html> (дата звернення: 27.04.2026).

31. IDE in Software: Definition & Examples. URL: <https://study.com/academy/lesson/ide-in-software-definition-examples.html> (дата звернення: 27.04.2026).

32. Покрокова інструкція, як створити макет сайту самостійно: вебсайт. URL: <https://sendpulse.com/ru/blog/site-layout> (дата звернення: 27.04.2026).
33. StoryBrain Figma to html. URL : <https://www.figma.com/community/plugin/851183094275736358> (дата звернення: 27.04.2026).
34. W3S. URL https://www.w3schools.com/html/html_responsive.asp/ (дата звернення: 27.04.2026)
35. Тестування сайту : вебсайт. URL: <https://polyarix.com/blog/testirovanie-sajta/> (дата звернення: 27.04.2026).
36. Що таке аудит сайту?.: вебсайт. URL: <https://polyarix.com/blog/chto-takoe-audit-sajta/> (дата звернення: 27.04.2026).
37. Аудит сайту: вебсайт. URL: https://marketing-mmz.bitrix24.site/audyt_saytu/ (дата звернення: 27.04.2026).
38. Що таке url адресу сайту? <https://www.ukraine.com.ua/blog/seo-optimization/chto-takoe-url-adres-sajta.html> (дата звернення: 27.04.2026).
39. Некіт К.Г. Доменне ім'я як об'єкт цивільних прав. *Часопис цивілістики*. 2018. Вип. 23. С.40-44.
40. Вартість створення сайту. *Lemarbet*: вебсайт. URL: <https://lemarbet.com/ua/razvitie-internet-magazina/stoimost-sozdaniya-sajta-primery/> (дата звернення: 27.04.2026).
41. Оцінка економічної ефективності Інтернет-ресурсів : вебсайт. URL: http://dipplus.com.ua/metodicheskiye-ukazaniya-i-informatsiya/article_post/otsinka-yekonomichnoi-yefektivnosti-internet-resursiv (дата звернення: 27.04.2026).