

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МЕНДЕЛЮК КРІСТИНА ВІКТОРІВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____ Наталія ВЕСЕЛОВСЬКА
«___» _____ 2026 р.

**ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСИСТЕМИ ПОТОКОВОГО
ПЕРЕГЛЯДУ ВІДЕОКОНТЕНТУ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:
Тетяна БАЦЕНКО, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця – 2026

АНОТАЦІЯ

Менделюк К.В. Проєктування та розробка вебсистеми потокового перегляду відеоконтенту. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

У кваліфікаційній роботі досліджено та проаналізовано сучасні підходи до розробки вебсистем відеостримінгу. Проведено аналіз існуючих платформ та обґрунтовано вибір технологій. Розроблено архітектуру системи, створено макети та прототип у Figma. Реалізовано адаптивний інтерфейс, навігацію, систему авторизації та персоналізації користувачів.

Ключові слова: вебсистема, відеостримінг, UX/UI-дизайн, веброзробка.

70 с., 26 рис., 7 табл., 0 дод., 49 джерел.

ANNOTATION

Mendeliuk K.V. Design and Development of a Web System for Video Content Streaming. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia, 2026.

In the qualification work, modern approaches to the development of video streaming web systems are studied and analyzed. Existing platforms are reviewed and the technology stack is justified. The system architecture is developed, and wireframes and a prototype are created in Figma. The system implements a responsive interface, navigation, user authentication, and personalization features.

Keywords: web system, video streaming, UX/UI design, web development.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	4
ВСТУП.....	6
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБСАЙТІВ.....	10
1.1. Аналіз існуючих платформ потокового відео: переваги та недоліки.....	10
1.2. Технічні характеристики та архітектура вебсистем потокового відео.....	13
1.3. Постановка задачі.....	15
РОЗДІЛ 2. ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБСИСТЕМИ.....	18
2.1. Інструменти розробки сайту.....	18
2.2. Figma як основний інструмент дизайну та прототипування.....	27
2.3. Принципи UX/UI-дизайну вебсайтів.....	30
2.4. Методологія прототипування та порівняльний аналіз інструментів.....	33
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСИСТЕМИ ПОТОКОВОГО ПЕРЕГЛЯДУ ВІДЕОКОНТЕНТУ.....	37
3.1. Розробка макета і прототипу сайту.....	37
3.2. Створення структури та навігації сайту.....	41
3.3. Розробка дизайну інтерфейсу в Figma.....	44
ВИСНОВКИ.....	63
СПИСОК ЛІТЕРАТУРИ.....	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ABS – адаптивне потокове передавання відео (Adaptive Bitrate Streaming)

API – програмний інтерфейс застосунку (Application Programming Interface)

ARIA – набір атрибутів доступності вебконтенту (Accessible Rich Internet Applications)

AV1 – відкритий відеокодек високої ефективності стиснення

AVC – стандарт відеокодування H.264 (Advanced Video Coding)

Back-end – серверна частина вебсистеми

CDN – мережа доставки контенту (Content Delivery Network)

CSS – каскадні таблиці стилів (Cascading Style Sheets)

CTA – заклик до дії (Call To Action)

Dev Mode – режим у Figma для передачі дизайну розробникам

DRM – система захисту цифрового контенту (Digital Rights Management)

ES – стандарт мови JavaScript (ECMAScript)

FFmpeg – інструмент для обробки мультимедіа

Front-end – клієнтська частина вебсистеми

HLS – протокол потокового передавання відео (HTTP Live Streaming)

HTML – мова розмітки вебсторінок (HyperText Markup Language)

HTTP/HTTPS – протоколи передачі даних у вебі (HyperText Transfer Protocol / Secure)

Hi-fi – високодеталізований прототип (High-Fidelity)

JWT – токен для автентифікації (JSON Web Token)

Lo-fi – низькодеталізований прототип (Low-Fidelity)

ML – машинне навчання (Machine Learning)

MPA – багатосторінковий вебзастосунок (Multi Page Application)

MPEG-DASH – стандарт адаптивного стримінгу (Dynamic Adaptive Streaming over HTTP)

Next.js – метафреймворк для React

Node.js – серверне середовище виконання JavaScript

OAuth 2.0 – протокол авторизації

OTT – сервіс доставки відеоконтенту через інтернет (Over-The-Top)

PWA – прогресивний вебзастосунок (Progressive Web App)

React – бібліотека для створення інтерфейсів

Redis – система кешування даних

REST API – архітектурний стиль взаємодії клієнт–сервер

SPA – односторінковий вебзастосунок (Single Page Application)

UI – користувацький інтерфейс (User Interface)

URL – адреса ресурсу в інтернеті (Uniform Resource Locator)

UX – користувацький досвід (User Experience)

WCAG – стандарти доступності вебконтенту (Web Content Accessibility Guidelines)

Wireframe – каркасний макет сторінки

W3C – консорціум стандартів вебу (World Wide Web Consortium)

ВСТУП

Розробка вебсистем потокового перегляду відеоконтенту набуває надзвичайної актуальності не лише з кожним роком, а й з кожним місяцем. Щодня на цифрових платформах з'являються нові відеосервіси, а конкуренція між ними досягає безпрецедентного рівня. За даними Precedence Research (лютий 2026 р.), обсяг глобального ринку відеостримінгу у 2025 році досяг 159,98 млрд доларів США, а до 2026 р. прогнозується зростання до 195,85 млрд доларів із сукупним річним темпом приросту 18,5% [33]. Це свідчить про масштабну трансформацію медіаспоживання: глядачі стрімко відмовляються від традиційного телебачення на користь гнучких, персоналізованих онлайн-платформ.

Прискорення цього переходу спричинила пандемія COVID-19: місяці карантину й перехід навчання та бізнесу в онлайн докорінно змінили звички мільярдів людей. Щомісячний час, що витрачається на перегляд відеоконтенту онлайн, зріс у 2020–2021 рр. понад 40%, а кількість підписників провідних стримінгових сервісів збільшилася на десятки відсотків [18]. Ця тенденція збережена і у 2025–2026 рр.: за даними Nielsen, стримінг у травні 2025 р. вперше перевищив 45,5% загального телевізійного перегляду у США, випередивши кабельне та ефірне мовлення разом узяті [31].

На українському ринку також спостерігається активний розвиток вітчизняних відеоплатформ: MEGOGO, Sweet.TV та Takflix формують локальний OTT-сегмент із акцентом на україномовний контент. Водночас більшість наявних рішень орієнтована на масову аудиторію і не завжди забезпечує оптимальний баланс між технічною якістю, зручністю інтерфейсу та доступністю. Це свідчить про реальну потребу у проектуванні і розробці нових вебсистем потокового відео, що відповідають сучасним стандартам якості.

Сучасна вебсистема стримінгу – це технологічно складний продукт, що поєднує адаптивне потокове передавання відео (ABS), компонентну SPA-архітектуру на базі React/Next.js, серверний API на Node.js, CDN-інфраструктуру

та алгоритми персоналізації. Процес її розробки охоплює проєктування UX/UI-дизайну в Figma, побудову інтерактивного прототипу, реалізацію клієнтської та серверної частин і тестування. Таким чином, тема дипломної роботи є актуальною як з наукової, так і з практичної точки зору.

Актуальність роботи обумовлена динамічним зростанням індустрії відеостримінгу та трансформацією способів споживання медіаконтенту, що потребує створення ефективних вебсистем із високим рівнем продуктивності, адаптивності та зручності використання. Недосконалість існуючих платформ у частині користувацького досвіду, навігації та доступності обумовлює необхідність проєктування інноваційних рішень, орієнтованих на потреби сучасного користувача.

Мета кваліфікаційної роботи – проєктування та розробка вебсистеми потокового перегляду відеоконтенту, яка поєднує сучасні технічні рішення із якісним UX/UI-інтерфейсом, відповідає актуальним стандартам веброзробки 2025–2026 рр. та забезпечує зручний, доступний і надійний перегляд відеоконтенту для кінцевого користувача.

Завдання дослідження. Для досягнення поставленої мети необхідно вирішити такі завдання:

1. вивчити та проаналізувати існуючі платформи потокового перегляду відеоконтенту – як глобальні (Netflix, YouTube, Amazon Prime Video), так і українські (MEGOGO, Sweet.TV, Takflix) – виявити їхні переваги та недоліки;
2. з'ясувати теоретичні основи поняття вебсайту, його класифікацію та технічні характеристики систем відеостримінгу (ABS, CDN, DRM, відеокодеки);
3. дослідити та обґрунтувати технологічний стек розробки вебсистеми відповідно до сучасних стандартів веброзробки;
4. вивчити дизайнерський інструмент Figma, його можливості та нові функції (Config 2025: Make, Draw, Sites, Buzz);

5. опрацювати принципи UX/UI-дизайну вебплатформ та застосувати їх при проєктуванні інтерфейсу;
6. розробити макет і прототип вебсистеми потокового відео в середовищі Figma;
7. спроектувати структуру та навігацію вебсайту (Site Map, User Flow);
8. розробити дизайн інтерфейсу ключових екранів системи — головна сторінка зі слайдером, тематичні секції фільмів, форми авторизації та сторінка профілю користувача.

Об'єкт дослідження. Методи проєктування та технології створення вебсистем потокового перегляду відеоконтенту.

Предмет дослідження. Інструменти UX/UI-проєктування вебсистеми потокового перегляду відеоконтенту: дизайн-платформа Figma, принципи побудови інтерактивного прототипу, стандарти доступності WCAG 2.1 та технологічний стек як архітектурна специфікація для подальшої програмної реалізації.

Методи дослідження. У роботі використано такі методи: порівняльний аналіз — для оцінки існуючих платформ стримінгу та інструментів розробки; системний підхід — для проєктування архітектури вебсистеми та її компонентів; метод прототипування — для розробки макетів та інтерактивних прототипів у Figma; метод UX-аудиту на основі евристичної оцінки — для перевірки відповідності інтерфейсу принципам юзабіліті; практична розробка — для безпосередньої реалізації дизайну вебсистеми.

Практичне значення полягає в тому, що її результати — технологічний стек, архітектурні рішення, компонентна система дизайну (Design System) та інтерактивний прототип — можуть бути безпосередньо використані при реалізації комерційного або освітнього відеостримінгового сервісу. Розроблений дизайн інтерфейсу у Figma слугує готовою специфікацією для команди розробників. Дослідження принципів UX/UI та стандартів WCAG 2.1 може бути використане у навчальному процесі при вивченні дисциплін «Проєктування вебзастосунків» та «UI/UX-дизайн».

Структура роботи. Робота складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі розглянуто теоретичні основи розробки вебсистем потокового відеоконтенту. Проведено аналіз існуючих платформ відеостримінгу, визначено їхні переваги та недоліки, а також досліджено технічні характеристики та архітектуру сучасних вебсистем.

У другому розділі досліджено інструменти та технології розробки вебсистеми. Розглянуто можливості платформи Figma, проаналізовано принципи UX/UI-дизайну та обґрунтовано вибір технологічного стеку для реалізації проєкту.

У третьому розділі виконано проєктування та розробку вебсистеми потокового перегляду відеоконтенту. Створено макети та інтерактивний прототип у Figma, розроблено структуру та навігацію сайту, а також реалізовано дизайн інтерфейсу ключових сторінок системи.

Загальний обсяг роботи складає 70 сторінки, містить 7 таблиць, 26 рисунків та список літератури з 49 джерел.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБСАЙТІВ

1.1. Аналіз існуючих платформ потокового відео: переваги та недоліки

Ринок потокового відеоконтенту є одним із найбільш динамічно зростаючих сегментів цифрової економіки станом на 2025–2026 роки. За даними Precedence Research (лютий 2026 р.), обсяг глобального ринку відеостримінгу у 2025 році досяг 159,98 млрд доларів США, а до 2026 року прогнозується зростання до 195,85 млрд доларів з сукупним річним темпом приросту (CAGR) 18,5% на горизонті 2026–2035 рр. [37]. Це свідчить про масштабну трансформацію медіаспоживання: глядачі відмовляються від традиційного телебачення на користь гнучких персоналізованих онлайн-платформ.

Для проєктування власної вебсистеми необхідно провести порівняльний аналіз провідних платформ – як глобальних, так і українських. Це дозволяє виявити кращі практики галузі (best practices), визначити недоліки конкурентів і сформулювати переваги нового продукту.

Netflix – найбільша у світі платформа підписного стримінгу. Станом на четвертий квартал 2025 р. компанія налічує понад 325 млн платних підписників у більш ніж 190 країнах – приріст на 7,6% порівняно з 302 млн роком раніше [33]. Квартальний дохід за Q4 2025 становив 12,05 млрд дол. (+17,6% рік до року), а річний – 45,2 млрд дол. (+16% рік до року). Ключові переваги: персоналізовані рекомендації на основі машинного навчання (ML), широка бібліотека Netflix Originals, підтримка адаптивного потокового передавання (ABS), офлайн-завантаження та впровадження ШІ для локалізації субтитрів і планування рекламних кампаній. Серед недоліків: відсутність безкоштовного рівня, регіональні обмеження бібліотеки, висока вартість підписки (від 7,99 дол./міс.).

YouTube – найбільша у світі відеоплатформа компанії Google із більш ніж 2,7 млрд щомісячних активних користувачів (MAU) у 2025 р. За даними Nielsen, у травні 2025 р. стримінг вперше перевищив 45,5% загального телеперегляду у США, і YouTube є ключовим рушієм цього зростання [49]. Переваги:

безкоштовний доступ, UGC-модель, Live Streaming, широкий інструментарій монетизації. Недоліки: нав'язлива реклама, нерівна якість контенту та алгоритмічна упередженість.

Amazon Prime Video – відеосервіс Amazon, інтегрований із глобальною екосистемою Prime. Переваги: оригінальний преміум-контент, технологія X-Ray, можливість оренди/купівлі окремих фільмів. Недоліки: складний UX та додаткова плата за частину контенту навіть при наявності підписки.

Поряд із глобальними платформами, на українському ринку активно діють вітчизняні OTT-сервіси. MEGOGO – найбільша українська OTT-платформа, заснована у 2011 р. Станом на початок 2026 р. пропонує понад 15 000 фільмів і серіалів, доступна у 14 країнах, підтримує Smart TV, iOS, Android, ігрові консолі та CarPlay. Тарифи від 99 до 599 грн/міс. Ключові переваги: найбільша україномовна контентна бібліотека, подкасти, аудіокниги, широка географія покриття та підтримка різноманітних пристроїв. Серед недоліків: окрема плата за прем'єри та частину контенту понад базову підписку, а максимальний тариф може бути недоступним для частини аудиторії.

Sweet.TV – запущена у 2017 р. як перша в Україні платформа з легальним україномовним дубляжем. Пропонує 260+ телеканалів і 10 000+ фільмів, підтримує відтворення у якості 4K, тарифи від 50 грн/міс [44]. Переваги: доступна ціна, великий вибір телеканалів, акцент на україномовній локалізації. Недолік: бібліотека VOD-контенту є вужчою порівняно з Netflix та MEGOGO, а функціональність рекомендаційної системи поступається глобальним платформам.

Takflix – перший спеціалізований онлайн-кінотеатр українського авторського кіно, заснований у 2019 р. режисеркою Надією Парфан. Діє за моделлю «один квиток – один фільм» (95 грн), пропонує 200+ стрічок [45]. Переваги: унікальна спеціалізація на українському авторському кіно, підтримка вітчизняних кінематографістів, культурна місія. Недоліки: модель поштучної оплати є менш зручною за підписку, а вузька ніша обмежує широку аудиторію.

Узагальнений порівняльний аналіз платформ наведено у табл. 1.1.

Таблиця 1.1 – Порівняльний аналіз платформ потокового відео (2025–2026 рр.)

Платформа	Рік запуску	Переваги	Недоліки
Netflix	1997 (стримінг з 2007)	325 млн підписників (Q4 2025); персоналізація ML; оригінальний контент; ABS; офлайн; III-субтитри	Висока ціна; регіональні обмеження; відсутність безкоштовного рівня
YouTube	2005	2,7+ млрд MAU; безкоштовний доступ; UGC; Live Streaming; YouTube Premium	Нав'язлива реклама; нерівна якість UGC; алгоритмічна упередженість
Amazon Prime Video	2006	Інтеграція з Amazon; X-Ray; оренда/купівля фільмів; оригінальний контент	Складний UX; частина контенту платна понад підписку
MEGOGO (UA)	2011	Найбільша OTT в Україні; 15 000+ фільмів; 14 країн; від 99 грн/міс.; подкасти; аудіокниги	Окрема плата за прем'єри; максимальний тариф до 599 грн/міс.
Sweet.TV (UA)	2017	260+ ТВ-каналів; 10 000+ фільмів; укр. дубляж; 4K; від 50 грн/міс.	Вужча VOD-бібліотека порівняно з Netflix
Takflix (UA)	2019	Спеціалізація на укр. авторському кіно; 200+ фільмів; підтримка вітчизняних кінематографістів	Оплата за кожен фільм окремо (95 грн); вузька ніша

Результати аналізу свідчать, що більшість платформ активно застосовують ABS та алгоритми ML, а на українському ринку є стійкий попит на локалізований контент за доступними цінами. Це підтверджує доцільність розробки нової

вебсистеми з акцентом на якісний UX та підтримку сучасних стандартів відтворення відео.

1.2. Технічні характеристики та архітектура вебсистем потокового відео

Вебсайт (від англ. website) – це сукупність пов'язаних між собою вебсторінок, доступних за унікальною адресою в мережі Інтернет (URL – Uniform Resource Locator) та об'єднаних спільним доменним іменем. Відповідно до стандартів Консорціуму Всесвітньої павутини (W3C – World Wide Web Consortium), вебсайт є інформаційним ресурсом, що розміщується на вебсервері та передається клієнтам за допомогою протоколу HTTP/HTTPS [47].

З технічного погляду, вебсайт являє собою комплекс клієнтської (front-end) та серверної (back-end) складових. Клієнтська частина відповідає за відображення інформації у браузері користувача та будується на базі технологій HTML5, CSS3 та JavaScript. Серверна частина забезпечує обробку запитів, взаємодію з базою даних і формування динамічного контенту. Залежно від призначення виокремлюють такі типи вебсайтів: інформаційні (корпоративні, вікі); електронної комерції (інтернет-магазини); соціальні мережі; вебзастосунки (web applications) з розвинутою інтерактивністю; медіаплатформи – портали для поширення та перегляду аудіовізуального контенту [47].

Вебсайти для потокового відео відносяться до категорії медіаплатформ – різновиду вебзастосунків. Їх відмінна риса: необхідність обробки та передачі великих обсягів мультимедійних даних у реальному часі, що ставить підвищені вимоги до архітектури сервера, якості мережевого з'єднання та продуктивності клієнтської частини. Відеостримінгова платформа виконує такі функції: зберігання і каталогізація відеоконтенту; потокова передача відеофайлів; пошук і фільтрація; управління обліковими записами; збір і аналіз поведінкових даних аудиторії.

Сучасні медіаплатформи будуються переважно за архітектурним принципом SPA (Single Page Application) на базі фреймворків React, Vue.js або

Angular, що забезпечує плавну взаємодію без повного перезавантаження сторінки. Стримінгові платформи на базі SPA забезпечують вищі показники утримання аудиторії порівняно з традиційними МРА-сайтами завдяки плавній навігації без перезавантаження сторінки та швидшому відгуку інтерфейсу.

Визначивши архітектурну основу медіаплатформ як різновиду вебзастосунків, розглянемо ключові технічні та функціональні характеристики, що визначають якість таких систем станом на 2025–2026 роки.

Адаптивне потокове передавання (Adaptive Bitrate Streaming, ABS) – фундаментальна технологія, що забезпечує динамічний вибір якості відеопотоку залежно від пропускної здатності мережі користувача. Провідними стандартами ABS є MPEG-DASH і Apple HLS. Технологія усуває переривання відтворення та забезпечує оптимальну якість зображення для кожного з'єднання [44].

Відеокодеки суттєво впливають на якість і обсяг трафіку. Станом на початок 2026 р. застосовуються: H.264 (AVC) – найбільш сумісний стандарт; H.265 (HEVC) – вдвічі ефективніша компресія; VP9 – відкритий кодек Google на YouTube; AV1 – найсучасніший відкритий кодек з найвищою ефективністю стиснення, підтримка якого значно розширилась у 2025 р. [23].

CDN (Content Delivery Network) – розподілена мережа серверів у різних географічних точках для кешування та доставки медіаконтенту з мінімальною затримкою. Станом на 2026 р. провідні постачальники: Cloudflare, Akamai, Amazon CloudFront, Fastly [24].

DRM (Digital Rights Management) – система захисту ліцензованого контенту від несанкціонованого копіювання. Стандарти: Google Widevine, Apple FairPlay, Microsoft PlayReady. У грудні 2025 р. компанія Redflag AI у партнерстві з EZDRM запустила хмарну DRM-as-a-Service платформу, що спрощує захист контенту для нових проєктів [37].

Рекомендаційні системи є критичними для утримання аудиторії. За даними звіту Netflix Q4 2025, впровадження III-алгоритмів підвищило точність прогнозування «хітів» на 23%, а персоналізовані постери та трейлери збільшили

клікабельність на 31% [32]. Сучасні рекомендаційні системи застосовують колаборативну фільтрацію, контентну фільтрацію та гібридні алгоритми ML.

Адаптивний дизайн (Responsive Design) є обов'язковою вимогою: у 2025 р. смартфони та планшети генерують понад 55% інтернет-трафіку і забезпечують близько 35% доходів відеоплатформ [37]. Платформа має однаково якісно працювати на різних пристроях і розмірах екранів.

Аутентифікація реалізується через OAuth 2.0 для входу через Google/Apple, JWT (JSON Web Tokens) для сесійного управління та RBAC (Role-Based Access Control) для розмежування прав доступу.

HTML5-відеоплеєр (Video.js або Shaka Player) – центральний елемент UX платформи. Підтримує субтитри, вибір швидкості та якості відтворення, повноекранний режим, доступність WCAG 2.1. Якість плеєра безпосередньо визначає задоволеність користувачів.

Отже, вебсистема потокового відео є технологічно складним вебзастосунком, що поєднує клієнтську SPA-архітектуру з комплексом серверних технологій: ABS-стримінгом, CDN-інфраструктурою, DRM-захистом та ML-рекомендаціями. Сукупність цих компонентів і визначає конкурентоспроможність сучасної медіаплатформи, слугуючи теоретичною основою для постановки задачі у наступному підрозділі.

1.3. Постановка задачі

На підставі проведеного аналізу існуючих платформ відеостримінгу та вивчення технічних характеристик сучасних медіасистем визначено ключові вимоги до проєктованої вебсистеми.

Аналіз конкурентів виявив спільні слабкі місця: складний інтерфейс, відсутність україномовної локалізації, недостатня доступність для користувачів з обмеженими можливостями та нераціональна цінова політика. Враховуючи ці недоліки, нова вебсистема має забезпечити якісний користувацький досвід, технічну надійність і доступність для широкої аудиторії.

Функціональні вимоги визначають, що система повинна вміти робити:

- забезпечувати потоковий перегляд відеоконтенту з підтримкою адаптивного передавання (ABS), що гарантує безперервне відтворення незалежно від якості інтернет-з'єднання;
- надавати можливість реєстрації та автентифікації користувачів із підтримкою OAuth 2.0 та JWT;
- реалізовувати зручний пошук і фільтрацію контенту за жанром, рейтингом та роком випуску;
- забезпечувати управління особистим профілем користувача та списком переглядів;
- відтворювати відео через HTML5-плеєр із підтримкою субтитрів і вибором якості зображення.

Нефункціональні вимоги визначають стандарти якості та продуктивності системи:

- адаптивний дизайн, що забезпечує коректне відображення на мобільних, планшетних і десктопних пристроях;
- час відгуку інтерфейсу не більше 2 секунд для забезпечення комфортної взаємодії;
- підтримка сучасних відеокодеків AV1 та H.265 для оптимальної якості зображення при мінімальному обсязі трафіку;
- захищене з'єднання за протоколом HTTPS/TLS 1.3 для забезпечення безпеки даних користувачів.

Сформульовані вимоги слугують основою для проєктування архітектури системи, розробки дизайну інтерфейсу та вибору технологічного стеку у наступних розділах роботи.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз теоретичних основ розробки вебсайтів для потокового перегляду відеоконтенту з використанням актуальних даних 2025–2026 рр.

Аналіз глобальних (Netflix, YouTube, Amazon Prime Video, Plex) та українських (MEGOGO, Sweet.TV, Takflix) платформ виявив ключові тенденції: широке застосування ABS, алгоритмічних рекомендацій на основі ШІ, CDN-інфраструктури та акцент на мобільних пристроях. Підтверджено стрімке зростання ринку – з 159,98 млрд дол. у 2025 р. до 195,85 млрд дол. у 2026 р. Netflix як лідер ринку досяг 325 млн підписників (Q4 2025). На українському ринку активно розвиваються власні OTT-сервіси з акцентом на україномовний контент і доступні тарифи.

Визначено поняття вебсайту згідно зі стандартами W3C, охарактеризовано медіаплатформи як різновид SPA-вебзастосунків та розглянуто ключові технічні складові відеостримінгу: адаптивне потокове передавання (MPEG-DASH, HLS), відеокодеки (AV1, H.265, H.264), CDN, DRM і рекомендаційні алгоритми ML. Встановлено, що якість відеоплеєра, адаптивний дизайн та надійна система аутентифікації є обов'язковими складовими конкурентоспроможної платформи.

На підставі проведеного аналізу сформульовано постановку задачі з чіткими функціональними і нефункціональними вимогами до проєктованої вебсистеми, що стануть основою для проєктування архітектури та вибору технологічного стеку у наступних розділах роботи.

РОЗДІЛ 2

ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ ВЕБСИСТЕМИ

2.1. Інструменти розробки вебсайту

Вибір технологічного стеку є одним із ключових рішень при розробці сучасної вебсистеми. Правильно підібраний набір інструментів безпосередньо впливає на продуктивність системи, зручність її підтримки та швидкість розробки. Для вебсистеми потокового перегляду відеоконтенту сформовано технологічний стек, що охоплює чотири основні складові: клієнтську частину, серверну частину, базу даних та середовище розробки. Архітектуру стеку ілюструє рис. 2.1.

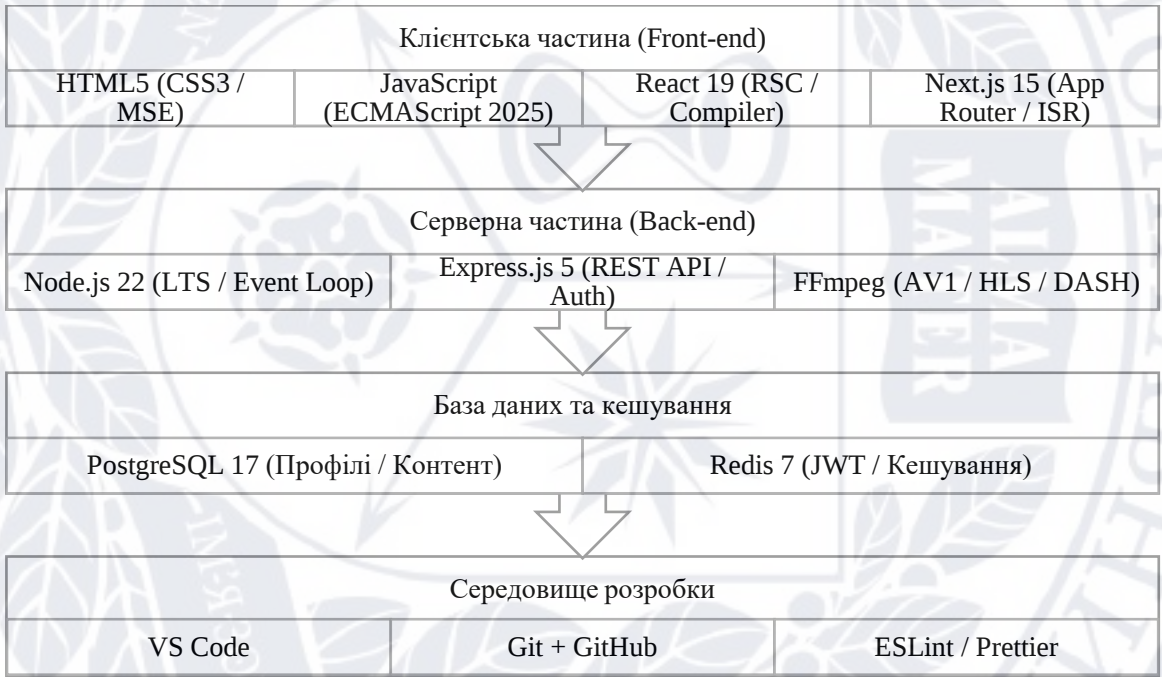


Рисунок 2.1 – Архітектура технологічного стеку вебсистеми потокового відео [1, с. 345]

Стек організовано за чотирьохрівневим принципом: клієнтська частина (HTML5, CSS3, JavaScript ES2025, React 19, Next.js 15) взаємодіє із серверною (Node.js 22 LTS, Express.js 5, FFmpeg) через REST API; серверний рівень, своєю чергою, звертається до рівня збереження даних, що складається з реляційної бази PostgreSQL 17 та кеш-сховища Redis 7. Усі рівні розробляються у єдиному середовищі Visual Studio Code з підтримкою Git і GitHub для версіонування. Така

багаторівнева організація забезпечує чітке розмежування відповідальності між компонентами системи (Separation of Concerns) та спрощує її подальше масштабування і супровід.

1. Клієнтська частина (Front-end).

HTML5 забезпечує структуру сторінок і нативне відтворення відео через елемент `<video>` без використання сторонніх плагінів. Media Source Extensions (MSE) API, що входить до специфікації HTML5, є технічною основою для реалізації адаптивного потокового передавання за стандартами MPEG-DASH та HLS [41]. CSS3 відповідає за візуальне оформлення інтерфейсу: завдяки технологіям Flexbox і CSS Grid Layout забезпечується коректне відображення на пристроях з різними розмірами екрана, а CSS-змінні використовуються для реалізації темної теми оформлення. JavaScript стандарту ECMAScript 2025 є мовою програмування клієнтської логіки [37]; сучасні можливості стандарту забезпечують ефективну асинхронну обробку даних та роботу з потоками відеоконтенту.

Для побудови компонентного інтерфейсу застосовується React 19 – провідна бібліотека розробки користувацьких інтерфейсів. React Compiler автоматично оптимізує компоненти, а React Server Components (RSC) забезпечують серверний рендеринг сторінок, що позитивно впливає на швидкість завантаження та індексування пошуковими системами. Поверх React використовується метафреймворк Next.js 15, який додає файлову маршрутизацію через App Router, серверні дії, автоматичну оптимізацію зображень та інкрементну статичну регенерацію (ISR) [38].

2. Серверна частина (Back-end).

Серверна частина вебсистеми побудована на платформі Node.js 22 LTS – середовищі виконання JavaScript на сервері [46]. Подійно-орієнтована архітектура Node.js ефективно обробляє велику кількість одночасних підключень, що є критично важливим при потоковій передачі відео. Для побудови програмного інтерфейсу (REST API) використовується Express.js 5 – мінімалістичний HTTP-фреймворк, що забезпечує маршрутизацію запитів,

автентифікацію користувачів та управління каталогом відеоконтенту. Транскодування відеофайлів у формати AV1 та H.265 і їх підготовка для HLS/DASH-стрімінгу здійснюється за допомогою бібліотеки fluent-ffmpeg – програмної обгортки навколо утиліти FFmpeg.

3. База даних та кешування.

Для зберігання структурованих даних застосовується PostgreSQL 17 – реляційна база даних із відкритим кодом, у якій зберігаються профілі користувачів, метадані відеоконтенту, історія переглядів та інформація про підписки. З метою підвищення продуктивності системи впроваджено Redis 7 – систему кешування даних в оперативній пам'яті. Redis використовується для зберігання сесійних токенів (JWT) та кешування часто використовуваних запитів до каталогу, що суттєво скорочує навантаження на базу даних.

4. Середовище розробки.

Розробка вебсистеми здійснюється у редакторі Visual Studio Code із підключеними інструментами статичного аналізу коду ESLint та автоматичного форматування Prettier. Для версіонування коду та організації командної роботи використовується система Git у поєднанні з хмарним репозиторієм GitHub.

Повний перелік технологій із зазначенням їх призначення у проєкті наведено у табл. 2.1.

Таблиця 2.1 – Технологічний стек вебсистеми

Інструмент	Категорія	Версія	Призначення у проєкті
1	2	3	4
HTML5/ CSS3	Розмітка / стилі	Living Standard	Структура сторінок, адаптивний дизайн, Dark Mode
JavaScript ES2025	Мова програмування	ECMAScript 2025	Клієнтська логіка, робота з потокami даних
React 19	UI-бібліотека	19.x	Компонентний інтерфейс, серверний рендеринг
Next.js 15	Метафреймворк	15.x	Маршрутизація, SEO, оптимізація завантаження

Продовження таблиці 2.1

1	2	3	4
Node.js 22 LTS	Серверне середовище	22.x LTS	REST API, потокова передача відео
Express.js 5	HTTP-фреймворк	5.x	API endpoints, автентифікація, маршрутизація
fluent-ffmpeg	Медіаобробка	актуальна	Транскодування AV1/H.265, підготовка HLS/DASH
PostgreSQL 17	База даних	17.x	Профілі користувачів, метадані контенту
Redis 7	Система кешування	7.x	Сесійні токени, кешування запитів
Visual Studio Code	Редактор коду	1.96+	Розробка, відладка, аналіз коду
Git + GitHub	Версіонування	Git 2.x	Контроль версій, командна робота

Наведений технологічний стек є збалансованим рішенням, що відповідає сучасним стандартам веброботки 2025–2026 рр. Кожен із обраних інструментів виконує чітко визначену роль у системі та забезпечує її надійність, масштабованість і зручність у розробці.

Для наочного представлення взаємодії між рівнями вебсистеми розроблено архітектурну схему типу «клієнт–сервер–база даних» (рис. 2.2). Клієнтська частина на основі React 19 та Next.js 15 взаємодіє із серверною через REST API за захищеним протоколом HTTPS. Серверний рівень на Node.js 22 та Express.js 5 обробляє вхідні запити, виконує автентифікацію через JWT та формує відповіді. Рівень збереження даних складається з реляційної бази PostgreSQL 17 та кешу Redis 7, що забезпечує скорочення навантаження на основну базу та прискорення відгуку системи.



Рисунок 2.2 – Архітектура вебсистеми «клієнт–сервер–база даних»

Як видно з рис. 2.2, система організована за трирівневим принципом: клієнтський рівень відповідає за відображення інтерфейсу, серверний — за

обробку бізнес-логіки та автентифікацію, рівень даних — за постійне зберігання та кешування. Така архітектура забезпечує чітке розмежування відповідальності між компонентами та спрощує подальше масштабування системи.

Для зберігання даних вебсистеми спроектовано реляційну базу даних у PostgreSQL 17, що складається з чотирьох взаємопов'язаних таблиць. Структуру таблиць та зв'язки між ними відображає ER-діаграма (рис. 2.3). Таблиця Users зберігає облікові дані користувачів: ідентифікатор, ім'я, email, хеш пароля та дату реєстрації. Таблиця Movies містить метадані фільмів: назву, опис, рік, рейтинг, URL постера та URL відеофайлу. Таблиця Favorites реалізує зв'язок між користувачами та фільмами через зовнішні ключі user_id та movie_id. Таблиця Watch_History фіксує хронологію переглядів із часовою міткою watched_at.

Схема бази даних вебсистеми фільмів

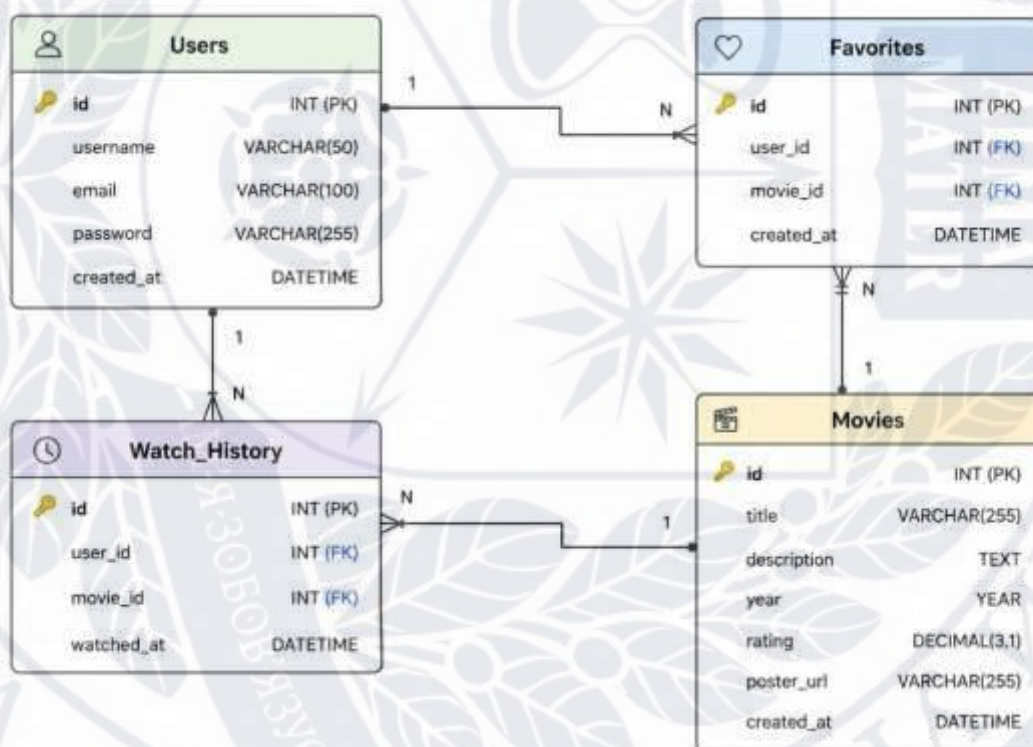


Рисунок 2.3 – ER-діаграма бази даних вебсистеми

Як видно з рис. 2.3, зв'язки між таблицями організовано за принципом 1:N — один користувач може мати довільну кількість записів у таблицях Favorites та Watch_History. Таблиця Movies є центральною сутністю схеми, на яку посилаються обидві проміжні таблиці через зовнішній ключ movie_id. Первинні

ключі усіх таблиць мають тип INT з автоінкрементом, що забезпечує унікальність кожного запису.

Взаємодія між клієнтською та серверною частинами реалізована через REST API. Повний перелік ендпоінтів із зазначенням методу, шляху та необхідності авторизації наведено у таблиці 2.2.

Таблиця 2.2 Специфікація REST API вебсистеми

Метод	Шлях	Опис	Авторизація
POST	/api/auth/register	Реєстрація нового користувача	Ні
POST	/api/auth/login	Вхід, повертає JWT-токен	Ні
GET	/api/movies	Список усіх фільмів	Ні
GET	/api/movies/:id	Дані одного фільму	Ні
GET	/api/movies?genre=Marvel	Фільтрація за жанром	Ні
POST	/api/favorites/:movieId	Додати до вподобаних	JWT
DELETE	/api/favorites/:movieId	Видалити з вподобаних	JWT
GET	/api/history	Історія переглядів	JWT
GET	/api/profile	Профіль поточного користувача	JWT

Як видно з табл. 2.2, специфікація містить дев'ять ендпоінтів, розподілених на дві групи. Перша група — публічні ендпоінти без авторизації, доступні всім відвідувачам: реєстрація, вхід та перегляд каталогу фільмів. Друга група — захищені ендпоінти з перевіркою JWT-токена у заголовку запиту Authorization: Bearer <token>, що забезпечують доступ до персональних даних користувача.

Для демонстрації технічної реалізації ключових функцій системи наведено чотири фрагменти програмного коду. Рисунок 2.4 ілюструє серверний маршрут Express.js, що реалізує отримання списку фільмів із підтримкою динамічної фільтрації за жанром та мінімальним рейтингом [15].

```

routes/movies.js

const express = require('express');
const router = express.Router();
const pool = require('../db');

router.get('/', async (req, res) => {
  const { genre, min_rating } = req.query;
  let query = 'SELECT * FROM movies WHERE 1=1';
  const params = [];

  if (genre) {
    params.push(genre);
    query += ` AND genre = ${params.length}`;
  }

  if (min_rating) {
    params.push(parseFloat(min_rating));
    query += ` AND rating >= ${params.length}`;
  }

  try {
    const result = await pool.query(query, params);
    res.json(result.rows);
  } catch (err) {
    res.status(500).json({ error: 'Помилка сервера' });
  }
});

module.exports = router;

```

Рисунок 2.4 – Express.js маршрут отримання списку фільмів із фільтрацією

Як видно з рис. 2.4, маршрут підтримує два необов'язкових параметри запиту — `genre` та `min_rating`. SQL-запит формується динамічно: базова частина доповнюється умовами лише за наявності відповідних параметрів. Параметризовані запити у форматі `$1`, `$2` захищають систему від SQL-ін'єкцій. Якщо параметри не передані, маршрут повертає повний каталог фільмів.

Рисунок 2.5 демонструє функціональний React-компонент картки фільму `MovieCard`, що відповідає за відображення постера, метаданих та кнопки перегляду.

```

components/MovieCard.jsx

import React from 'react';

const MovieCard = ({ title, genre, year, rating, posterUrl, onWatch }) => {
  return (
    <div className="movie-card">
      <img
        src={posterUrl}
        alt={`Постер фільму ${title}`}
        className="movie-card_poster"
      />
      <div className="movie-card_info">
        <h3 className="movie-card_title">{title}</h3>
        <p className="movie-card_meta">{genre} · {year}</p>
        <span className="movie-card_rating">★ {rating}/10</span>
      </div>
      <button
        className="movie-card_btn"
        onClick={() => onWatch(title)}
        aria-label={`Дивитись фільм ${title}`}
      >
        ДИВИТИСЬ
      </button>
    </div>
  );
};

export default MovieCard;

```

Рисунок 2.5 – React-компонент MovieCard [38]

Як видно з рис. 2.5, компонент MovieCard є функціональним і отримує всі необхідні дані через props без власного стану, що відповідає принципу розділення відповідальності у React-архітектурі. Атрибут aria-label на кнопці «ДИВИТИСЬ» забезпечує коректне зчитування елемента читачами екрана відповідно до стандарту WCAG 2.1.

Рисунок 2.6 наводить алгоритм фільтрації каталогу фільмів, реалізований у вигляді JavaScript-функції filterMovies. Функція підтримує одночасну фільтрацію за жанром та мінімальним рейтингом, що забезпечує гнучкий пошук у каталозі вебсистеми.

```

utils/filterMovies.js – алгоритм фільтрації каталогу

/**
 * Фільтрація масиву фільмів за жанром та мінімальним рейтингом
 * @param {Array} movies – масив об'єктів фільмів
 * @param {string} genre – жанр для фільтрації (необов'язково)
 * @param {number} minRating – мінімальний рейтинг (необов'язково)
 */

function filterMovies(movies, genre = null, minRating = null) {
  return movies.filter(movie => {
    // перевірка жанру якщо параметр передано
    if (genre && movie.genre !== genre) return false;
    // перевірка мінімального рейтингу якщо параметр передано
    if (minRating && movie.rating < minRating) return false;
    return true;
  });
}

// приклад використання
const byGenre = filterMovies(movies, 'Action');
const topRated = filterMovies(movies, null, 8.0);
const combined = filterMovies(movies, 'Drama', 7.5);

module.exports = filterMovies;

```

Рисунок 2.6 – Алгоритм фільтрації каталогу фільмів за жанром та рейтингом

Як видно з рис. 2.6, функція `filterMovies` приймає масив фільмів та два необов'язкових параметри — `genre` і `minRating`. Якщо передано лише жанр — повертаються всі фільми цього жанру незалежно від рейтингу. Якщо передано обидва параметри — застосовується комбінований фільтр. Такий підхід відповідає функціональним вимогам до пошукового модуля системи, визначеним у підрозділі 1.3.

Центральним елементом інтерфейсу вебсистеми є HTML5-відеоплеєр. Для підтримки адаптивного стримінгу за стандартом HLS застосовується бібліотека `hls.js` [43]. Рисунок 2.7 демонструє реалізацію підключення відеопотоку.

```

player.html – підключення HLS через hls.js

<video
  id="video-player"
  controls
  width="100%"
  poster="/images/spider-man-poster.jpg"
  aria-label="Відеоплеєр фільму">
  Ваш браузер не підтримує відтворення відео.
</video>

<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>

<script>
  const video = document.getElementById('video-player');
  const hlsUrl = 'https://cdn.example.com/marvel/index.m3u8';

  if (Hls.isSupported()) {
    // Chrome, Firefox – без нативної підтримки HLS
    const hls = new Hls();
    hls.loadSource(hlsUrl);
    hls.attachMedia(video);
  } else if (video.canPlayType('application/vnd.apple.mpegurl')) {
    // Safari – нативна підтримка HLS
    video.src = hlsUrl;
  }
</script>

```

Рисунок 2.7 – Підключення HLS-відеопотоку через бібліотеку hls.js [43]

Як видно з рис. 2.7, реалізація підтримує два сценарії: для браузерів без нативної підтримки HLS (Chrome, Firefox) використовується бібліотека hls.js, для Safari — нативне відтворення через атрибут src. Файл index.m3u8 є маніфестом HLS-потoku із посиланнями на відеосегменти різної якості. Бібліотека hls.js автоматично обирає оптимальну якість залежно від швидкості інтернет-з'єднання, що є практичною реалізацією адаптивного потокового передавання, описаного у підрозділі 1.2.

2.2. Figma як основний інструмент дизайну та прототипування

Figma — хмарна платформа для проектування інтерфейсів, прототипування та командної роботи над дизайном. Компанія заснована у 2012 р. Діланом

Філдом та Еваном Воллесом; штаб-квартира розташована у Сан-Франциско, США. Станом на 2025–2026 рр. Figma є провідним інструментом у галузі UI/UX-проєктування з часткою ринку понад 40%, випереджаючи Adobe XD та Sketch [42].

Ключовою технічною перевагою Figma є реалізація платформи як Progressive Web App (PWA), що дозволяє повністю працювати у браузері без встановлення додаткового програмного забезпечення. Рендеринг інтерфейсу виконується через WebGL, що забезпечує плавну роботу зі складними макетами [29]. Реальна колаборація реалізована через WebSocket – кілька учасників команди можуть працювати над одним файлом одночасно, аналогічно до Google Docs.

Платформа складається з функціонально відокремлених модулів, що охоплюють повний цикл проєктування. Design Mode є основним середовищем розробки макетів: містить векторний редактор, Auto Layout для автоматичного розміщення елементів, компонентну систему (Components та Variants) та бібліотеки стилів. Компонентна система дозволяє створювати багаторазові UI-елементи – кнопки, картки, поля введення – і застосовувати їх на всіх сторінках, забезпечуючи єдність дизайну [29]. Prototype Mode забезпечує інтерактивне прототипування з переходами між екранами, анімаціями Smart Animate та публікацією готового прототипу як посилання для тестування. Dev Mode автоматично генерує CSS-специфікації для кожного елемента макету із зазначенням відступів, розмірів шрифтів і кольорів, суттєво спрощуючи передачу дизайну команді розробників [29]. FigJam є вбудованим онлайн-вайтбордом для ранніх етапів проєктування: побудови блок-схем, карт сайту та навігаційних сценаріїв. Система Variables та Tokens забезпечує управління темами оформлення (Light/Dark Mode) через глобальні змінні кольорів, відступів і радіусів.

На конференції Config 2025 компанія представила чотири нові продукти [27]: Figma Draw – повноцінний векторний редактор для створення ілюстрацій та іконок; Figma Sites – інструмент публікації вебсайтів із автоматичною

генерацією HTML/CSS-коду з макету; Figma Buzz – засіб масового створення брендovаних маркетингових матеріалів; Figma Make – інструмент генерації інтерактивного прототипу або React-коду за текстовим описом.

У контексті даної кваліфікаційної роботи Figma використовується для: побудови каркасів сторінок та навігаційних сценаріїв у FigJam; проєктування компонентної системи дизайну з уніфікованою бібліотекою UI-елементів; розробки повноекранних макетів ключових сторінок вебсистеми; створення інтерактивного прототипу зі Smart Animate-переходами; передачі специфікацій розробнику через Dev Mode.

Порівняльний аналіз Figma з основними конкурентами наведено у таблиці 2.3.

Таблиця 2.3 – Порівняльний аналіз інструментів UI/UX-дизайну (2025–2026 рр.)

Критерій	Figma	Sketch	Adobe XD
Платформа	Браузер (PWA) + Desktop	macOS only	Windows / macOS
Колаборація	Real-time	Обмежена	Обмежена
Прототипування	Smart Animate, Variables	Базове	Auto-Animate
Dev Mode	CSS / Swift / Kotlin	Платний	Обмежено
ШІ-функції (2025)	Make, Draw, Sites, Buzz	Відсутні	Adobe Firefly
Ринкова частка	~40%	~10%	~13%
Безкоштовний план	Так	Ні	Ні
Статус (2026)	Активний розвиток	Активний (macOS)	Розробку припинено

Як видно з табл. 2.3, Figma має суттєві переваги над конкурентами за більшістю критеріїв. На відміну від Sketch, що доступний лише на macOS, Figma працює у будь-якому браузері на будь-якій операційній системі. Adobe XD станом на 2026 р. фактично виведено з активної розробки [27], тоді як Figma продовжує розширювати функціональність. Реальна колаборація, розвинена система прототипування та інтеграція ШІ-інструментів підтверджують доцільність вибору Figma як основного інструменту проєктування у даній роботі.

2.3. Принципи UX/UI-дизайну вебсайтів

UX (User Experience – досвід користувача) та UI (User Interface – користувацький інтерфейс) є взаємопов'язаними, але концептуально відмінними дисциплінами проектування цифрових продуктів. UX охоплює увесь спектр взаємодії людини із системою: її корисність, зручність та задоволення від використання. UI стосується конкретних візуальних і інтерактивних рішень інтерфейсу – того, що користувач безпосередньо бачить і з чим взаємодіє.

Розмежування між UX і UI сформулював Дон Норман, який у 1990-х роках, працюючи в компанії Apple, запровадив поняття «user experience design» як стратегічний рівень проектування, що охоплює всю систему взаємодії людини з продуктом [34]. UI є тактичною реалізацією цього досвіду – конкретними рішеннями щодо кольорів, типографіки, розміщення елементів та їх поведінки. Таким чином, UX визначає, що має робити система для користувача, а UI – як це виглядатиме та відчуватиметься.

Загальноприйнятою теоретичною основою UX/UI-проектування є «10 евристик юзабіліті» Якоба Нільсена, сформульовані у 1994 році та актуалізовані Nielsen Norman Group у 2024 році [34]. Евристики являють собою емпіричні принципи-орієнтири, що використовуються при оцінюванні якості інтерфейсних рішень (Heuristic Evaluation). Нижче розглянуто вісім найбільш релевантних для проектування вебплатформ евристик.

Евристика 1 – Видимість стану системи (Visibility of System Status). Система має постійно і своєчасно інформувати користувача про свій поточний стан. Це реалізується через індикатори завантаження, прогрес-бари, сповіщення про результат виконаних дій та візуальні підказки щодо поточного режиму роботи. Відсутність зворотного зв'язку від системи породжує невизначеність і знижує довіру користувача до продукту.

Евристика 2 – Відповідність системи й реального світу (Match Between System and the Real World). Інтерфейс має використовувати лексику, концепти та метафори, знайомі цільовій аудиторії, а не внутрішню термінологію системи. Елементи керування, іконки та підписи мають відповідати усталеним уявленням

користувачів, сформованим попереднім досвідом роботи з аналогічними продуктами.

Евристика 3 – Свобода дій та контроль (User Control and Freedom). Користувачі нерідко здійснюють дії помилково і потребують чітко позначеного «аварійного виходу» – можливості скасувати або відмінити попередню дію. Наявність функцій «Скасувати» (Undo) та «Повторити» (Redo), підтвердження перед незворотними діями і зрозуміла навігація «назад» є обов'язковими елементами будь-якого сучасного вебінтерфейсу.

Евристика 4 – Консистентність та стандарти (Consistency and Standards). Елементи інтерфейсу мають зберігати однаковий вигляд, поведінку та термінологію на всіх сторінках продукту. Досягається це через розробку єдиної дизайн-системи – бібліотеки компонентів із суворо визначеними кольорами, типографікою, відступами та станами елементів (Normal / Hover / Active / Disabled / Focus). Консистентний інтерфейс скорочує час навчання і зменшує кількість помилок.

Евристика 5 – Запобігання помилкам (Error Prevention). Продуманий дизайн має попереджати виникнення помилок ще до того, як вони відбудуться. Типові рішення включають валідацію форм у реальному часі, автодоповнення у полях введення, обмеження допустимих значень через елементи вибору замість вільного введення, а також підтвердження перед незворотними діями.

Евристика 6 – Розпізнавання, а не пригадування (Recognition Rather Than Recall). Інтерфейс має мінімізувати когнітивне навантаження: користувач не повинен утримувати в пам'яті інформацію з попереднього екрану для виконання дії на поточному. Це досягається через видимі підказки, контекстні меню, навігаційні «хлібні крихти» та збереження попередніх дій і налаштувань між сесіями.

Евристика 7 – Гнучкість та ефективність (Flexibility and Efficiency of Use). Інтерфейс має однаково добре обслуговувати як початківців, так і досвідчених користувачів. Для новачків передбачається зрозумілий онбординг і підказки; для досвідчених – гарячі клавіші, розширені фільтри та можливість налаштовувати

робочий простір під власні потреби. Такий підхід підвищує ефективність взаємодії без ускладнення інтерфейсу для менш досвідченої аудиторії.

Евристика 8 – Естетичний та мінімалістичний дизайн (Aesthetic and Minimalist Design). Відповідно до закону Хікса, час прийняття рішення зростає зі збільшенням кількості варіантів вибору – ця залежність описується логарифмічною функцією від числа альтернатив. Практичний наслідок для проєктування полягає в тому, що інтерфейс має містити лише ті елементи, які безпосередньо необхідні для поточного завдання. Зайва інформація відволікає увагу від ключових дій і знижує загальну ефективність взаємодії.

Окрім принципів юзабіліті, проєктування вебінтерфейсу передбачає прийняття конкретних рішень щодо кольорової схеми, типографіки та адаптивності.

Кольорова схема. Вибір кольорової палітри визначається характером продукту та потребами цільової аудиторії. Для медіасервісів і платформ з переважно відеоконтентом галузевим стандартом є темний режим (Dark Mode), що знижує навантаження на зорові рецептори при тривалому перегляді. Для інформаційних і ділових ресурсів, навпаки, переважає світла палітра, яка асоціюється з відкритістю та довірою. В обох випадках акцентний колір використовується для виділення кнопок дій, активних посилань та індикаторів прогресу, зберігаючи при цьому достатній контраст із фоном.

Типографіка. Чітка типографічна ієрархія дозволяє користувачеві миттєво орієнтуватися в структурі сторінки. Загальноприйнятим підходом є розподіл тексту на рівні: заголовки першого рівня (20–24 пт, напівжирне накреслення), підзаголовки (16 пт, напівжирне або середнє), основний текст (14 пт, звичайне) та допоміжні метадані (12 пт, знижена непрозорість). Для вебінтерфейсів рекомендовано використання системних або добре підтримуваних гарнітур – Inter, Noto Sans, Roboto – що забезпечує коректне відображення на всіх платформах і пристроях.

Стандарт доступності WCAG 2.1. Web Content Accessibility Guidelines 2.1 рівня AA є обов'язковою вимогою для сучасних вебплатформ. Стандарт

регламентує: коефіцієнт контрасту між текстом і фоном – не менше 4,5:1 для основного тексту та 3:1 для великого тексту (від 18 пт); підтримку читачів екрана через ARIA-атрибути (aria-label, role, aria-live); повну клавіатурну навігацію без використання миші; субтитри та аудіоопис для відеоконтенту [48]. Відповідність WCAG 2.1 є також юридичною вимогою в ЄС відповідно до Директиви 2019/882 (European Accessibility Act), що набула чинності для нових цифрових продуктів у червні 2025 року.

Принцип мобільного пріоритету. Оскільки за даними Statista (2025) понад 55% інтернет-трафіку генерується з мобільних пристроїв, дизайн вебінтерфейсів розробляється спочатку для найменшого екрану (320–375 пікселів), а потім послідовно масштабується до планшету (768–1024 пікселів) та десктопу (1280 пікселів і вище) через прогресивне вдосконалення та CSS-медіазапити [32]. Такий підхід гарантує, що основний функціонал продукту залишається доступним і зручним незалежно від розміру та типу пристрою.

Таким чином, принципи UX/UI-дизайну утворюють цілісну систему вимог до сучасного вебінтерфейсу: від загальних евристик юзабіліті до конкретних рішень щодо кольору, типографіки, доступності та адаптивності. Їх послідовне застосування забезпечує якісний користувацький досвід і відповідність актуальним галузевим та правовим стандартам.

2.4. Методологія прототипування та порівняльний аналіз інструментів

Прототипування є ключовим етапом UX-процесу, що дозволяє візуалізувати та перевірити ідеї щодо структури, навігації та зовнішнього вигляду вебсайту до початку розробки, знизити витрати на майбутні зміни та отримати зворотний зв'язок від користувачів на ранніх етапах. За визначенням Nielsen Norman Group, прототип – це представлення продукту або системи, що використовується для тестування концепцій перед фінальною реалізацією [35].

Класифікація прототипів за рівнем деталізації. Залежно від етапу проєктування та мети тестування розрізняють два основні типи прототипів. Ло-

fi прототипи (low-fidelity) – каркасні макети, що відображають лише основну структуру сторінок, їх ієрархію та навігаційні зв'язки без застосування реальних кольорів, шрифтів чи зображень. Основна мета – швидко перевірити ідеї на ранніх етапах проектування та узгодити загальні принципи навігації і структури вебсайту [35]. Hi-fi прототипи (high-fidelity) – детальні інтерактивні макети з реальними кольорами, шрифтами та анімаціями, що точно відображають фінальний вигляд і функціональність продукту. Такі прототипи використовуються для проведення юзабіліті-тестування з реальними користувачами та отримання точнішого зворотного зв'язку перед запуском [35].

Порівняльний аналіз інструментів прототипування. Для реалізації прототипування вебсистеми розглянуто три основні інструменти на додаток до Figma, детально розглянутої у п. 2.2. Balsamiq Wireframes спеціалізується на створенні lo-fi прототипів у стилі ескізів від руки, що психологічно налаштовує учасників проекту на обговорення, а не сприйняття макету як остаточного рішення. Інструмент має розширену бібліотеку готових UI-компонентів, однак не підтримує hi-fi прототипування та генерацію коду [22]. Marvel App дозволяє імпортувати макети з Figma і додавати до них інтерактивні елементи для юзабіліті-тестування. Платформа підтримує запис сесій взаємодії користувачів з прототипом і аналіз поведінки через теплові карти [36]. Adobe XD та InVision фактично виведено з активної розробки у 2023–2024 рр., що унеможлиблює їх використання у довгострокових проєктах.

Порівняльний аналіз інструментів прототипування наведено у таблиці 2.4.

Таблиця 2.4 – Порівняльний аналіз інструментів прототипування (2025–2026 рр.)

Інструмент	Рівень	Ключові переваги	Обмеження	Вибір
1	2	3	4	5
Figma + FigJam	Lo-fi + Hi-fi	Єдине середовище; колаборація; React-код (Make); ШІ-схеми	Потребує навчання; платний тариф для команд	Основний

Продовження табл. 2.4

1	2	3	4	5
Balsamiq	Lo-fi	Швидкість ескізів; hand-drawn стиль; 100+ компонентів	Без hi-fi; без генерації коду	Допоміжний
Marvel App	Hi-fi	User Testing; heatmaps; запис сесій; імпорт з Figma	Менший функціонал; платне тестування	За потреби
Adobe XD	Hi-fi	Інтеграція Adobe CC; Auto-Animate	Розробку припинено (2023)	Не обрано
InVision	Hi-fi	Огляди; коментарі; Inspect	Масове скорочення функцій (2024)	Не обрано

Figma є єдиним інструментом, що охоплює повний цикл прототипування – від lo-fi каркасів до hi-fi інтерактивних макетів з генерацією React-коду. Саме тому для реалізації даного проєкту обрано Figma як єдину платформу повного циклу UX-проектування: FigJam для lo-fi каркасування, Design Mode для hi-fi дизайну, Prototype Mode для інтерактивного прототипування, а Figma Make – для прямої генерації React-коду з готового макету. Такий підхід забезпечує єдине джерело правди протягом усього циклу розробки.

Висновки до розділу 2

У другому розділі визначено інструментальну та теоретичну базу для проектування і розробки вебсистеми потокового перегляду відеоконтенту.

У підрозділі 2.1 сформовано технологічний стек, що охоплює чотири складові: клієнтську частину на основі HTML5, CSS3, JavaScript ES2025, React 19 та Next.js 15; серверну частину на базі Node.js 22 LTS, Express.js 5 та fluent-ffmpeg; реляційну базу даних PostgreSQL 17 з кешуванням через Redis 7; середовище розробки Visual Studio Code з інструментами ESLint, Prettier та системою версіонування Git і GitHub. Кожен із обраних інструментів відповідає сучасним стандартам веброзробки 2025–2026 рр. і виконує чітко визначену роль у системі.

Для підтвердження технічної обґрунтованості обраного стеку розроблено архітектурну схему системи, ER-діаграму бази даних із чотирьох таблиць та специфікацію REST API з дев'ятьма ендпоінтами. Наведено чотири фрагменти програмного коду, що демонструють реалізацію серверного маршруту з динамічною фільтрацією, React-компонента картки фільму, алгоритму пошуку за жанром і рейтингом та підключення HLS-відеоплеєра. Усі наведені матеріали підтверджують відповідність архітектурних рішень функціональним вимогам, визначеним у розділі 1.

У підрозділі 2.2 розглянуто Figma як основний інструмент проектування інтерфейсу та прототипування. Охарактеризовано модульну архітектуру платформи (Design Mode, Prototype Mode, Dev Mode, FigJam, Variables), новітні ШІ-інструменти Config 2025 (Make, Draw, Sites, Buzz) та обґрунтовано переваги Figma над конкурентами – Sketch і Adobe XD – за критеріями крос-платформеності, колаборації, функціональності та актуальності розвитку.

У підрозділі 2.3 досліджено теоретичні засади UX/UI-проектування на основі евристик Nielsen Norman Group та стандарту доступності WCAG 2.1. Визначено ключові принципи побудови якісного вебінтерфейсу: консистентність, запобігання помилкам, мінімалістичність, типографічна ієрархія та підхід Mobile-First Design.

У підрозділі 2.4 систематизовано методологію прототипування (lo-fi / hi-fi) та проведено порівняльний аналіз інструментів – Figma, Balsamiq і Marvel App. Обґрунтовано вибір Figma як єдиної платформи повного циклу проектування, що забезпечує безперервний перехід від каркасів до готового React-коду без зміни середовища.

Результати другого розділу є основою для практичного проектування вебсистеми у третьому розділі.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБСИСТЕМИ ПОТОКОВОГО ПЕРЕГЛЯДУ ВІДЕОКОНТЕНТУ

У третьому розділі описано практичний процес проєктування та розробки дизайну односторінкового вебсайту, присвяченого фільмам кіновсесвіту Marvel. Розробка охоплює три послідовні етапи: створення макету та прототипу, формування структури й навігації сайту, а також детальне проєктування дизайну кожного екрану у середовищі Figma. Кожне прийняте проєктне рішення супроводжується обґрунтуванням та відповідними графічними матеріалами.

Слід зазначити, що у рамках кваліфікаційної роботи реалізацію серверної та клієнтської частин здійснено на рівні архітектурного проєктування та інтерактивного прототипування. Розроблений у Figma hi-fi прототип є повноцінним проєктним артефактом — він містить специфікації всіх компонентів через Dev Mode (кольори, відступи, типографіку, стани елементів) і безпосередньо слугує технічним завданням для програмної реалізації системи. Такий підхід відповідає сучасній галузевій практиці, за якою UX/UI-проєктування виділяється в окремий етап розробки, що передує написанню коду.

3.1. Розробка макету і прототипу сайту

Проєктування вебсайту розпочиналося зі створення концептуальної моделі, яка визначила загальний напрямок, стиль і принципи побудови інтерфейсу. На цьому підготовчому етапі сформульовано ідею продукту, окреслено перелік екранів та встановлено логіку переходів між ними.

3.1.1. Концепція проєкту

Вебсайт проєктувався як односторінковий тематичний медіаресурс, присвячений супергеройським фільмам кіновсесвіту Marvel. Головна ідея полягає у тому, щоб кожен фільм каталогу отримав власний повноекранний простір із унікальним візуальним оформленням, що відображає характер і атмосферу відповідного персонажа. Така концепція відрізняє розроблений сайт

від типових стримінгових платформ із однорідним списковим каталогом: тут глядач не переглядає список, а занурюється у світ кожного героя окремо.

Відповідно до концепції визначено склад основних екранів: головна сторінка зі слайдером фільмів; вісім індивідуальних секцій для кожного фільму каталогу; розділ постерів; форма реєстрації та форма входу; сторінка профілю авторизованого користувача з переліком вподобаних фільмів. Такий набір екранів охоплює повний цикл взаємодії користувача з сайтом – від першого знайомства з контентом до персоналізованого використання платформи.

3.1.2. Wireframes – каркасні схеми екранів

Wireframe – це схематичне зображення сторінки, що фіксує розташування функціональних блоків та їхню ієрархію без кольорового оформлення і реальних зображень. Застосування wireframes на початку роботи дозволяє зосередитися на структурі та логіці інтерфейсу, відклавши вирішення естетичних питань на наступний етап. Каркасні схеми для всіх ключових екранів розроблялися у середовищі FigJam.

Wireframe головної сторінки зафіксував трирівневу вертикальну структуру: хедер із логотипом та навігаційним меню у верхній частині; центральна зона повноекранного слайдера з hero-зображенням та назвою фільму; інформаційний блок нижче – з постером, описом, рейтингом та кадрами зі сцен, що відкривається при прокручуванні сторінки вниз (рис. 3.1). Таке рішення дозволяє зберегти сильний перший візуальний ефект від hero-зображення, одночасно надаючи користувачеві повний обсяг інформації без переходу на окрему сторінку.

Wireframe сторінки окремого фільму закріпив триблокову компоновку: hero-секція з повноекранним зображенням персонажа та назвою вгорі; центральний інформаційний блок із постером, текстовим описом, рейтингом і кнопками «Дивитись» та «Дальніше»; периферійні зони ліворуч і праворуч для декоративних зображень персонажів. Вже на рівні wireframe було закладено принцип унікального кольорового оформлення для кожного фільму (рис. 3.2).

Wireframes форм авторизації і реєстрації передбачали мінімалістичну центровану компоновку: темний фон сторінки та контейнер форми фіксованої ширини по центру екрана. Послідовне розташування елементів зверху вниз – заголовок, кнопки соціальних мереж, поля введення, кнопка підтвердження – відповідає природному напрямку читання.



Рисунок 3.1 – Wireframe головної сторінки вебсайту у середовищі FigJam

Як видно з рис. 3.1, wireframe головної сторінки фіксує трирівневу вертикальну композицію: хедер, повноекранна hero-зона та інформаційний блок. Така структура забезпечує баланс між сильним першим візуальним враженням та повнотою інформації без переходу на окрему сторінку.

Рисунок 3.2 демонструє триблокову компоновку сторінки фільму: hero-секція, центральний інформаційний блок та периферійні декоративні зони. Вже на каркасному рівні закладено асиметричну композицію, що відрізняє дизайн від типових симетричних шаблонів медіаплатформ.

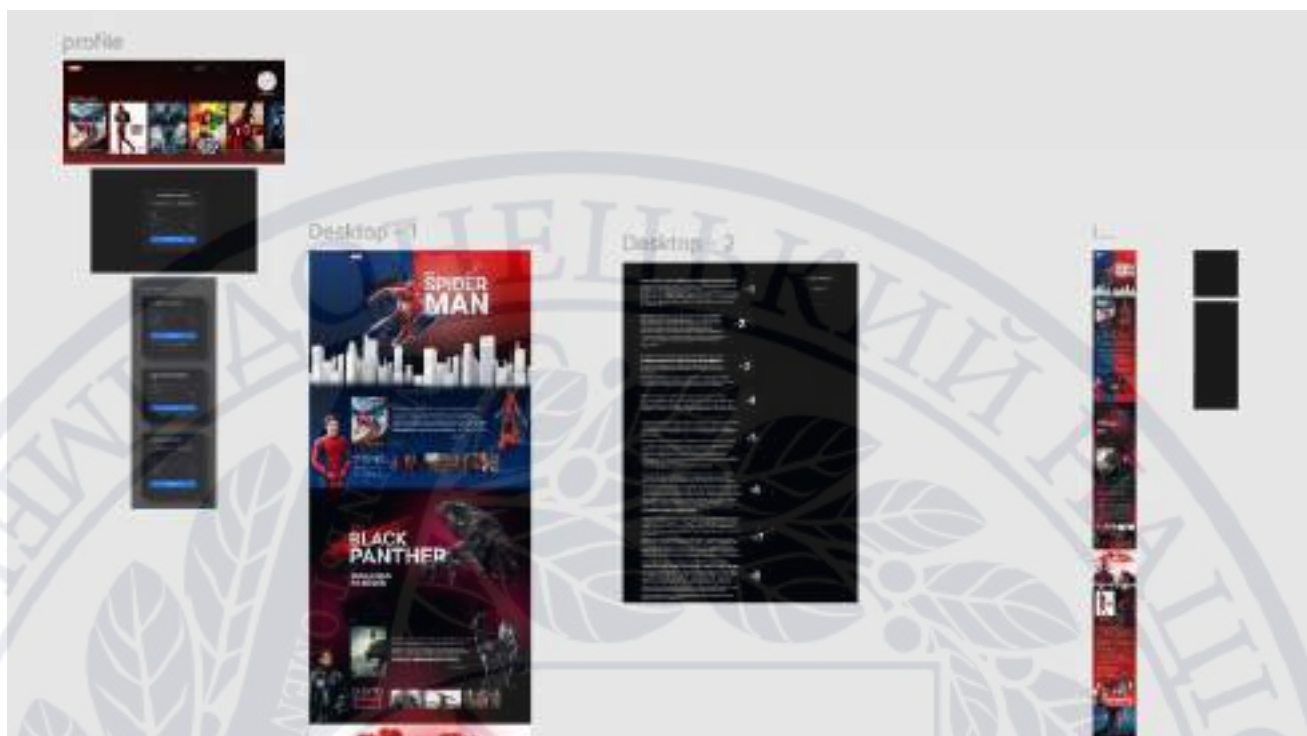


Рисунок 3.2 – Wireframe сторінки окремого фільму: функціональні блоки

3.1.3. Інтерактивний прототип у Figma

Після узгодження каркасних схем розроблено інтерактивний прототип із застосуванням функції Prototype у Figma. Між фреймами встановлено навігаційні зв'язки, що відтворюють реальну поведінку сайту: натискання на логотип Marvel повертає до головної сторінки; прокручування слайдера відкриває наступний фільм; кнопка «Дивитись» веде до відтворення відеоконтенту; посилання «Дальніше » розгортає повний опис; іконка профілю переводить авторизованого користувача до особистої сторінки, неавторизованого – до форми входу.

Побудований прототип пройшов перевірку за основними сценаріями використання: перегляд інформації про фільм, реєстрація облікового запису, додавання фільму до вподобаних та перегляд профілю. За результатами перевірки скориговано логіку переходів між формами входу та реєстрації, а також уточнено поведінку навігаційного меню. Схему навігаційних зв'язків між екранами ілюструє рис. 3.3.



Рисунок 3.3 – Схема навігаційних зв'язків між екранами у режимі Prototype (Figma)

Для наочності структури навігації на рисунку 3.3 представлено повну карту переходів між екранами системи. Центральним вузлом навігації є головна сторінка зі слайдером, від якої розгалужуються зв'язки до тематичних секцій фільмів, форм авторизації та профілю. Тестування прототипу підтвердило логічну послідовність переходів та відсутність «глухих кутів» у навігації.

3.2. Створення структури та навігації сайту

Структура вебсайту визначає організацію контенту та принципи навігації між розділами. Для розробленого проєкту обрано плоску дворівневу ієрархію, що відповідає характеру тематичного медіаресурсу з фіксованим каталогом фільмів.

3.2.1. Інформаційна архітектура

Вебсайт складається з таких основних екранів: головна сторінка – повноекранний слайдер-каталог восьми фільмів; вісім тематичних секцій фільмів – Spider-Man: Повернення додому, Black Panther: Wakanda Forever, Deadpool, Venom, Iron Man, Thor, Wonder Woman, Ant-Man; розділ Постери;

сторінка Профілю авторизованого користувача; форма реєстрації та форма входу. Структуру сайту наведено на рисунку 3.4.



Рисунок 3.4 – Карта сайту вебсистеми

Як видно з рисунку 3.4, сайт організовано за дворівневою плоскою ієрархією: головна сторінка є єдиною точкою входу до всіх тематичних секцій та допоміжних сторінок системи. Відсутність третього рівня ієрархії мінімізує глибину навігації та забезпечує доступність будь-якого розділу за не більше ніж двома кліками від головної сторінки.

Принциповим рішенням є відмова від окремої сторінки-каталогу. Замість неї роль каталогу виконує головна сторінка зі слайдером: користувач послідовно взаємодіє з одним фільмом за раз, не відволікаючись на весь список одразу. Це рішення обґрунтоване специфікою контенту: фіксована кількість фільмів дозволяє представити кожну стрічку як самостійний об'єкт із повноекранним оформленням, що підсилює емоційний ефект від знайомства з кожним персонажем.

3.2.2. Навігаційна система

Навігація реалізована через постійний хедер, розташований на всіх сторінках сайту. Хедер складається з трьох елементів: логотип Marvel у лівій частині – одночасно є посиланням на головну сторінку; горизонтальне меню «Головна / Постери / Більше» по центру; іконка профілю користувача праворуч. Мінімалістична структура хедера є свідомим проєктним рішенням: на тлі яскравих тематичних фонів надмірно насичена навігація конкурувала б із контентом за увагу глядача.

Активний пункт меню відображається з підкресленням, що забезпечує чітку індикацію поточного розділу. Посилання «Дальніше» на сторінках фільмів реалізує принцип поступового розкриття інформації: короткий опис видимий одразу, повний текст – на запит. Це дозволяє зберегти баланс між інформативністю та візуальною легкістю сторінки.

Навігаційні сценарії визначено окремо для двох груп користувачів. Неавторизований користувач має повний доступ до перегляду всіх сторінок фільмів та розділу постерів. При зверненні до кнопки «Дивитись» або іконки профілю сайт перенаправляє його до форми входу. Авторизований користувач отримує розширений доступ: відтворення відеоконтенту та персональна сторінка профілю зі списком вподобаних фільмів.

Зовнішній вигляд навігаційного хедера наведено на рис. 3.5.



Рисунок 3.5 – Навігаційний хедер: логотип, меню та іконка профілю

Рисунок 3.5 ілюструє мінімалістичну структуру хедера: логотип Marvel ліворуч, три пункти меню по центру та іконка профілю праворуч. Така триелементна компоновка є галузевим стандартом для медіаплатформ і відповідає евристиці Нільсена «Відповідність системи й реального світу» – користувач одразу впізнає звичну модель навігації.

3.3. Розробка дизайну інтерфейсу в Figma

Розробка дизайну здійснювалася у середовищі Figma відповідно до компонентного підходу: спочатку сформовано дизайн-систему з визначеними кольорами, типографікою та базовими компонентами, а на її основі опрацьовано кожен екран. Таке поєднане формування дизайн-системи забезпечує візуальну узгодженість усіх сторінок попри їхню суттєву індивідуальну відмінність.

3.3.1. Кольорова система та типографіка

Кольорова система розробленого сайту побудована на концепції «кольорової ідентичності»: кожна тематична сторінка фільму має власну домінантну кольорову схему, що відображає характер і атмосферу відповідного персонажа. Водночас системні елементи – хедер, форми авторизації, футер – витримані в єдиній темній монохромній палітрі, що забезпечує брендову цілісність сайту попри різноманітність тематичних фонів.

Аналіз кольорових стилів, збережених у бібліотеці Figma, виявив такі основні значення палітри: базовий білий текст #FFFFFF; акцентний червоний #E30000 для логотипу Marvel та елементів активних станів; чорний фон системних сторінок #000000; системний синій #0080A4 та #29456D для форм авторизації; темно-синій тематичний #043072 та #345C97; додаткові кольори для тематичних фонів: #420516, #1A8DFF, #E89B29 (табл. 3.1).

Таблиця 3.1 Кольорові ідентичності тематичних сторінок фільмів

Фільм	Домінантні кольори	Концепція та декоративні елементи
1	2	3
Spider-Man: Повернення додому	#043072 / #E30000	Вертикальний синьо-червоний поділ; силует міста та павутини

Продовження табл. 3.1

1	2	3
Black Panther: Wakanda Forever	#420516 / #1A0008	Темно-бордовий монохром; декоративні штрихи червоної фарби
Deadpool	#E30000 / #FFFFFF	Яскраво-червоний на білому; хаотична динамічна композиція
Venom	#000000 / #1C1C1C	Глибокий чорний монохром; назва накладена напівпрозора
Iron Man	#043072 / #29456D	Темно-синій технологічний; двоколірна назва (#E30000 + #FFFFFF)
Thor	#E30000 / #0080A4	Червоно-синій градієнт; графічні блискавки по всьому тлу
Wonder Woman	#E89B29 / #F5A800	Золотисто-жовтий; неонові геометричні лінії та овал
Ant-Man	#29456D / #345C97	Сталево-синій монохром; анфас- ракурс персонажа

Таблиця 3.1 систематизує кольорові ідентичності восьми тематичних сторінок. Аналіз палітри свідчить про послідовне застосування концепції: теплі червоно-золоті відтінки використовуються для персонажів із підвищеною емоційною енергетикою (Deadpool, Wonder Woman, Thor), тоді як холодні сині та темні тони – для технологічних або похмурих образів (Iron Man, Venom, Ant-Man). Це рішення відповідає загальноприйнятій теорії кольорової психології у UI-дизайні.

Типографічну систему сайту побудовано на гарнітурі Roboto – шрифті без зарубок, що забезпечує стабільну читабельність як на темному, так і на яскравому кольоровому тлі. Основним накресленням обрано Roboto Medium з базовим розміром 22 px та міжрядковим інтервалом 35 px – значення, зафіксовані у бібліотеці стилів Figma. Такий міжрядковий інтервал (коефіцієнт 1.59) забезпечує комфортне читання текстових блоків на насичених тематичних фонах.

Типографічна ієрархія формується через розмір і вагу шрифту: назви фільмів у hero-секціях – uppercase, 64–96 px залежно від довжини назви, що

забезпечує їх домінування на сторінці; рейтингові показники – 36–48 px Roboto Medium; основний текст описів – 15–16 px; елементи навігації хедера – 14–16 px.

Рисунок 3.6 відображає структуровану бібліотеку стилів у Figma: кольорові свотчі з HEX-значеннями та типографічні зразки з параметрами накреслення. Централізоване зберігання стилів у бібліотеці Figma забезпечує консистентність дизайну відповідно до евристики Нільсена «Консистентність та стандарти» та спрощує внесення глобальних змін на всіх сторінках одночасно.



Рисунок 3.6 – Типографічні стилі та кольорова бібліотека у Figma

3.3.2. Головна сторінка та слайдер фільмів

Головна сторінка є центральним екраном сайту та реалізована у форматі повноекранного слайдера. Кожен слайд відповідає окремому фільму і займає 100% ширини та висоти видимої частини браузера. Хедер накладається поверх слайдера – він присутній на всіх слайдах і забезпечує постійний доступ до навігації незалежно від активного фільму.

Перший слайд, присвячений фільму «Spider-Man: Повернення додому», є показовим прикладом реалізованої концепції. Тло слайда розподілено по вертикалі на два кольорових поля: ліве – темно-синє (#043072), праве – акцентно-червоне (#E30000). У точці з'єднання полів по центру розміщено постать Людини-Павука у динамічній позі стрибка. На лівому тлі схематично зображено силует міських хмарочосів та знак павутини. На правому – велика типографічна назва «SPIDER-MAN» білим uppercase-шрифтом у два рядки. Дизайн hero-секції першого слайда наведено на рис. 3.7.



Рисунок 3.7 – Головна сторінка: hero-секція слайда «Spider-Man»

На рисунку 3.7 представлено реалізацію концепції «кольорової ідентичності» на прикладі першого слайда. Вертикальний поділ тла на синє та

червоне поле відтворює кольорову гаму костюма Людини-Павука та одночасно забезпечує чіткий ритм сторінки. Розміщення постаті персонажа у точці з'єднання полів формує природний візуальний центр композиції.

При прокручуванні сторінки вниз відкривається інформаційний блок фільму. Він організований у трьохколонній структурі на тематичному тлі: ліва колонка – портрет актора у повний зріст; центральна – вертикальний постер фільму, під яким виведено рейтинг «7.9/10» великим жирним шрифтом та кнопка «ДИВИТИСЬ» з рамковим обрамленням; права частина – текстовий опис сюжету, посилання «Дальніше » та три горизонтальних кадри зі сцен фільму. Крайня права зона відведена ще одному декоративному зображенню персонажа у характерній перевернутій позі. Інформаційний блок фільму наведено на рис. 3.8.



Рисунок 3.8 – Головна сторінка: інформаційний блок фільму «Spider-Man» з постером, рейтингом та кадрами

Рисунок 3.8 ілюструє трьохколонну структуру інформаційного блоку. Розміщення рейтингу «7.9/10» великим шрифтом безпосередньо під постером є

свідомим UX-рішенням: поєднання візуального та числового оцінювання в одній зоні відповідає евристиці «Розпізнавання, а не пригадування» – користувач отримує ключову інформацію для прийняття рішення без необхідності її пошуку.

Аналогічну двочастинну структуру застосовано для всіх восьми слайдів. Повторюваність структури при різноманітності кольорового оформлення забезпечує передбачуваність навігації: ознайомившись із першим слайдом, користувач одразу розуміє, як взаємодіяти з усіма наступними.

3.3.3. Тематичні сторінки фільмів

Кожна з восьми секцій фільму реалізує спільну структурну схему через унікальний кольоровий образ. Нижче описано особливості дизайну кожної сторінки.

Сторінка «Black Panther: Wakanda Forever» витримана у глибоких темно-бордових тонах (#420516, #1A0008). Hero-секція містить постать Чорної Пантери у стрибку на тлі декоративних штрихів червоної фарби – мотив, що передає агресивну силу образу. Назва виведена у два рядки: «BLACK PANTHER» та підзаголовок «WAKANDA FOREVER». Рейтинг 9.5/10 в інформаційному блоці (рис. 3.9).



Рисунок 3.9 – Сторінка фільму «Black Panther: Wakanda Forever»: hero-секція та інформаційний блок

Сторінка «Deadpool» побудована на контрасті яскраво-червоного (#E30000) та білого (#FFFFFF). Hero-секція містить два зображення персонажа у різних позах – прийом, що підкреслює роздвоєність і непередбачуваність образу. Нестандартне типографічне рішення – логотип-маска замість літери «О» у назві «DEADPOOL» – перетворює заголовок на іконічний брендовий елемент персонажа. Дизайн сторінки наведено на рис. 3.10.



Рисунок 3.10 – Сторінка фільму «Deadpool»: hero-секція та інформаційний блок

Рисунок 3.10 ілюструє найконтрастнішу кольорову схему каталогу – червоний на білому. Використання двох зображень персонажа в різних позах та логотип-маска у назві є прикладом наративного дизайну: візуальні елементи без тексту передають роздвоєність і гумористичну природу персонажа.

Сторінка «Venom» реалізована у майже повністю чорній палітрі (#000000). Него-зображення – великий план обличчя Венома, наполовину людського, наполовину симбіота – займає весь viewport. Назва «VENOM» виведена великими напівпрозорими літерами поверх зображення, що створює ефект глибини. Портрет актора в інформаційному блоці виконано у монохромних сірих тонах для збереження загальної похмурої атмосфери.

Дизайн сторінки наведено на рис. 3.11.



Рисунок 3.11 – Сторінка фільму «Venom»: hero-секція та інформаційний блок

На рисунку 3.11 представлено найбільш мінімалістичне рішення каталогу: майже повністю чорна палітра та напівпрозора типографіка назви створюють ефект глибини і тривоги, що відповідає темному характеру персонажа.

Монохромний портрет актора в інформаційному блоці підтримує єдину тональність сторінки.

Сторінка «Iron Man» витримана у темно-синіх технологічних тонах (#043072, #29456D). Типографіка назви двоколірна: «IRON» – червоним (#E30000), «MAN» – білим (#FFFFFF), що відтворює кольорову гаму броні Тоні Старка. Рейтинг 10/10 є максимальним у всьому каталозі. Дизайн сторінки наведено на рис. 3.12.



Рисунок 3.12 – Сторінка фільму «Iron Man»: hero-секція та інформаційний блок

Рисунок 3.12 ілюструє технологічну естетику сторінки Iron Man. Двоколірна типографіка назви – червоний та білий на темно-синьому тлі –

відтворює колірну гаму броні Тоні Старка і є прикладом узгодженості дизайнерського рішення з брендовою ідентичністю персонажа.

Сторінка «Thor» побудована на переході від червоного (#E30000) у верхній частині до синього (#0080A4) знизу. Декоративні графічні блискавки, що пронизують hero-секцію, є наративним елементом дизайну: вони без тексту розповідають про природу персонажа – бога грому скандинавської міфології. В інформаційному блоці присутні два зображення актора Кріса Гемсворта в різних костюмах, що відображає еволюцію образу Тора у кіновсесвіті Marvel.

Дизайн сторінки наведено на рис. 3.13.



Рисунок 3.13 – Сторінка фільму «Thor»: hero-секція та інформаційний блок

Наративний потенціал графічних елементів проілюстровано на рисунку 3.13: використання блискавок як ключового візуального символу забезпечує

впізнаваність персонажа без додаткового текстового супроводу. Градієнт від червоного до синього формує динамічну візуальну композицію.

Сторінка «Wonder Woman» вирізняється золотисто-жовтим тлом (#E89B29), що асоціюється із силою та благородством персонажа. Нестандартним рішенням є великий напівпрозорий овал у hero-секції з назвою «Wonder Woman» всередині – геометричний прийом, що надає сторінці сучасного графічного вигляду та відрізняє її від решти каталогу. Неонові геометричні лінії на тлі підсилюють відчуття динаміки. Дизайн сторінки наведено на рис. 3.14.



Рисунок 3.14 – Сторінка фільму «Wonder Woman»: hero-секція та інформаційний блок

Нестандартне композиційне рішення каталогу проілюстровано на рисунку 3.14, де напівпрозорий овал із назвою виступає як комбінований типографічний

і геометричний елемент. Використання золотисто-жовтої палітри підкреслює відмінність сторінки та формує асоціації із силою і благородством.

Сторінка «Ant-Man» витримана у сталевих синіх тонах (#29456D, #345C97). Постать Людини-Мурахи зображена анфас у статичній позі – на відміну від динамічних поз інших героїв каталогу. Симетричний анфас-ракурс підсилює відчуття стабільності та присутності персонажа, максимально демонструючи деталізацію технологічного костюма. Рейтинг 7.9/10. Дизайн сторінки наведено на рис. 3.15.



Рисунок 3.15 – Сторінка фільму «Ant-Man»: hero-секція та інформаційний блок

Рисунок 3.15 ілюструє симетричну анфас-композицію – єдину в каталозі. На відміну від динамічних поз інших персонажів, статична фронтальна постать Людини-Мурахи акцентує увагу на деталізації технологічного костюма і

створює відчуття впевненої присутності персонажа. Сталево-синя палітра підкреслює технологічний характер образу.

3.3.4. Дизайн системи авторизації

Форми реєстрації та входу виконані у нейтральній темній палітрі (#000000, #1C1C1C), яка принципово відрізняється від яскравих тематичних сторінок фільмів. Зміна візуального середовища сигналізує користувачеві про перехід від розважального до практичного режиму взаємодії. Обидві форми розміщено по центру сторінки у контейнері фіксованої ширини (~480 px).

Форма реєстрації «Create an account» пропонує два способи створення облікового запису. Перший – через соціальні мережі: кнопки «Google» та «Facebook» у верхній частині форми дозволяють завершити реєстрацію за одне натискання. Другий – через email: поле Email із плейсхолдером «name@gmail.com» та поле Password з іконкою перемикання видимості символів і посиланням «Forgot ?». Кнопка «Create account» у синьому кольорі є головним закликом до дії на сторінці. Посилання «Already Have An Account? Log In» забезпечує перехід до форми входу. Дизайн форми реєстрації наведено на рис. 3.16.

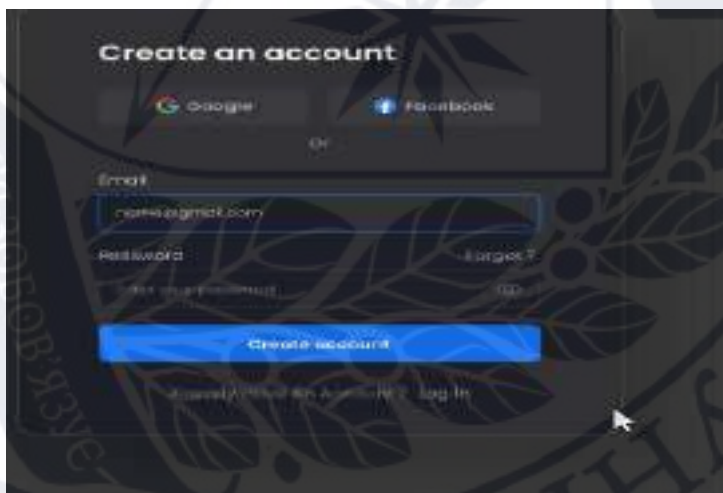


Рисунок 3.16 – Форма реєстрації «Create an account»: кнопки OAuth та поля введення

На рисунку 3.16 представлено дворівневу структуру форми реєстрації: у верхній частині розміщено соціальну авторизацію, а нижче — класичні поля

введення email і пароля. Така організація відповідає евристиці «Гнучкість та ефективність», дозволяючи досвідченим користувачам скористатися швидким

входом, а новим — пройти повну реєстрацію. Форма входу «Login to your account» має спрощену структуру: поля Email та Password та кнопка «Login now». Посилання «Don't Have An Account? Sign Up» забезпечує перехід до реєстрації. Поля введення реалізують два стани: нейтральний за замовчуванням та активний при фокусі – із синьою рамкою, що чітко позначає поточний елемент введення. Дизайн форми входу наведено на рис. 3.17.

Рисунок 3.17 – Форма входу «Login to your account»

Рисунок 3.17 ілюструє спрощену структуру форми входу порівняно з реєстрацією. Візуальна зміна стану поля при фокусі (синя рамка) реалізує евристику «Видимість стану системи» – користувач завжди бачить, який елемент активний. Нейтральна темна палітра обох форм авторизації свідомо контрастує з яскравими тематичними сторінками, сигналізуючи про перехід до функціонального режиму взаємодії.

3.3.5. Дизайн сторінки профілю користувача

Сторінка профілю є персоналізованим простором авторизованого користувача. Оформлення у темно-червоних тонах (#420516, #5A0000) пов'язує профіль із загальною Marvel-тематикою сайту та відрізняє його від нейтральних форм авторизації. Навігаційне меню у хедері оновлює активний стан – підкреслюється пункт «Профіль».

Сторінка містить два основних блоки. Перший – аватар із підписом «Змінити фото» у правій верхній частині. Другий – секція «Вподобані вами фільми» ліворуч: горизонтальний ряд вертикальних постерів фільмів, доданих користувачем до улюблених. Постери виводяться у стандартизованому форматі з однаковою висотою, формуючи рівномірну сітку. Список формується динамічно на основі дій користувача та відображає актуальний стан вподобань. Дизайн сторінки профілю наведено на рис. 3.18.

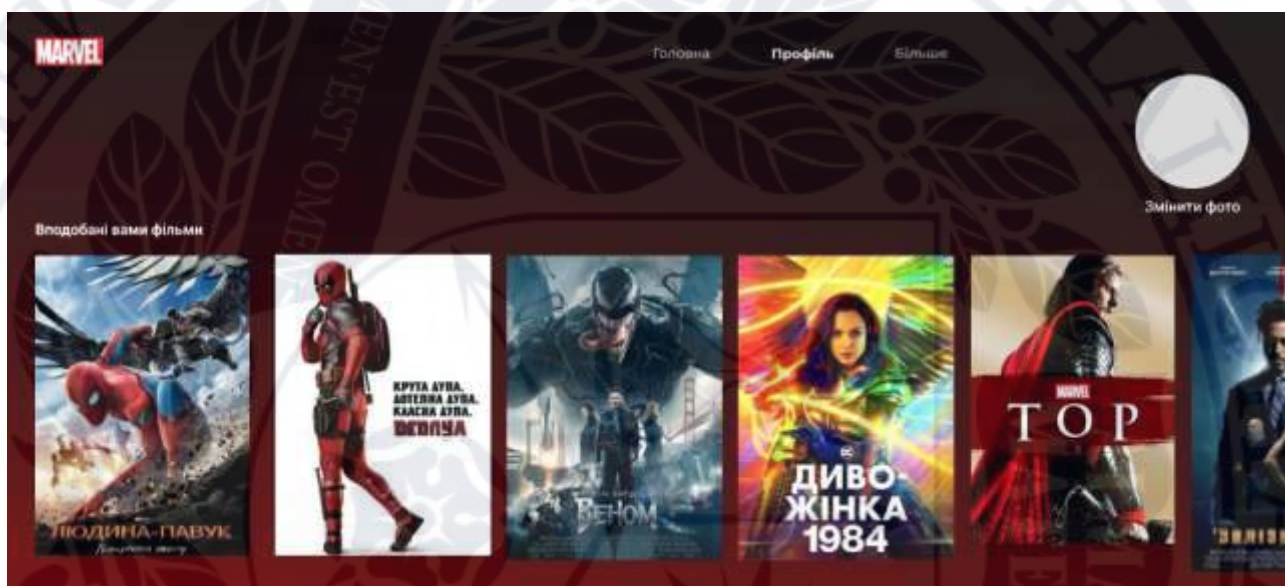


Рисунок 3.18 – Сторінка профілю: аватар та список «Вподобані вами фільми»

Персоналізований інтерфейс авторизованого користувача проілюстровано на рисунку 3.18. Використання сітки постерів у секції «Вподобані вами фільми» відповідає принципу «Розпізнавання, а не пригадування», забезпечуючи швидку ідентифікацію контенту. Темно-червона кольорова гама підтримує стилістичну єдність із тематикою ресурсу.

3.3.6. Дизайн футера

Футер виконано у чорному (#000000) і присутній на всіх сторінках сайту. Монохромне оформлення утворює чітку нижню межу будь-якого тематичного фону та виконує роль візуального завершення сторінки. Структурно футер містить секцію «Контакти» з фізичною адресою (площа Гагаріна 2, Вінниця), номером телефону та електронною поштою, а також рядок монохромних білих

іконок соціальних мереж: Instagram, Telegram, Facebook, YouTube. Мінімалістичний стиль іконок відповідає загальній темній естетиці базових елементів системи. Дизайн футера наведено на рис. 3.19.

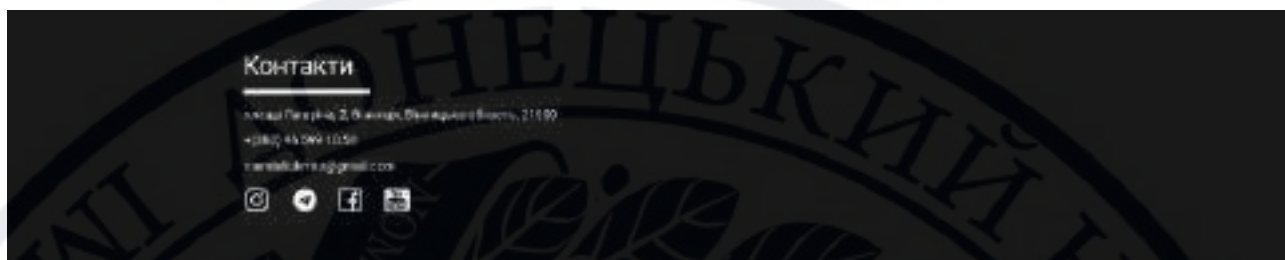


Рисунок 3.19 – Футер сайту: секція «Контакти» та іконки соціальних мереж

Рисунок 3.19 ілюструє монохромний футер як візуальний «якір» усіх сторінок сайту. Єдине чорне оформлення футера на тлі різноманітних тематичних кольорів виконує важливу UX-функцію: чітко позначає нижню межу контенту та забезпечує передбачуваний фінал взаємодії з кожною сторінкою незалежно від її кольорової теми.

Для більш системного аналізу прийнятих дизайнерських рішень було здійснено їх співставлення з евристиками юзабіліті Nielsen Norman Group. Узагальнені результати наведено в таблиці 3.2.

Таблиця 3.2 – Відповідність дизайнерських рішень евристикам юзабіліті Nielsen Norman Group

№	Евристика Нільсена	Реалізація у проєкті
1	2	3
1	Видимість стану системи	Синя рамка активного поля форм; підкреслення активного пункту меню; кнопка «ДИВИТИСЬ» як індикатор доступного контенту
2	Відповідність системи й реального світу	Постери фільмів, рейтинги, кнопки «Дивитись» – звична модель медіаплатформи; іконки соціальних мереж у футері
3	Свобода дій та контроль	Логотип Marvel як постійне посилання «додому»; навігаційне меню на всіх сторінках; посилання між формами входу та реєстрації
4	Консистентність та стандарти	Єдина дизайн-система Figma: гарнітура Roboto, спільна структура слайдів, однотипний хедер і футер на всіх сторінках

Продовження табл. 3.2

1	2	3
5	Запобігання помилкам	Два способи авторизації (OAuth + email); плейсхолдери у полях форм; іконка перемикавання видимості пароля
6	Розпізнавання, а не пригадування	Постери замість текстового списку у профілі; hero-зображення персонажів як візуальні ідентифікатори фільмів
7	Гнучкість та ефективність	OAuth для швидкої авторизації; посилання «Дальніше »» для поглибленого перегляду; слайдер як прискорений спосіб перегляду каталогу
8	Естетичний та мінімалістичний дизайн	Три пункти навігаційного меню; мінімалістичний хедер; темний футер без зайвих елементів

Як видно з таблиці 3.2, більшість дизайнерських рішень відповідають евристикам юзабіліті Nielsen Norman Group. У проекті забезпечено зрозумілу навігацію, візуальний зворотний зв'язок та консистентність елементів інтерфейсу. Це сприяє зручності користування, зниженню кількості помилок і формуванню позитивного користувацького досвіду.

3.3.7. Адаптивність інтерфейсу та відповідність стандарту WCAG 2.1

Відповідно до принципу Mobile-First Design, обґрунтованого у підрозділі 2.3, інтерфейс вебсистеми спроектовано з урахуванням коректного відображення на трьох групах пристроїв: мобільних (320–375 px), планшетних (768–1024 px) та десктопних (1280 px і вище).

На мобільних пристроях повноекранний слайдер зберігає hero-зображення та назву фільму, тоді як інформаційний блок перебудовується з трьохколонної у одноколонну структуру – постер, рейтинг, опис та кадри розміщуються послідовно вертикально. Навігаційне меню хедера на вузьких екранах трансформується у компактний варіант з іконкою. Форми авторизації при ширині менше 480 px розтягуються на всю ширину екрана з відступами 16 px з кожного боку.

Щодо відповідності стандарту WCAG 2.1 рівня AA: усі кольорові пари перевірено на достатній контраст – білий текст (#FFFFFF) на темних фонах (#043072, #000000, #420516) забезпечує коефіцієнт контрасту не менше 7:1, що

перевищує мінімальну вимогу 4.5:1. Поля форм мають коректні атрибути `aria-label` для підтримки читачів екрана. Кнопки та посилання мають зону кліку не менше 44×44 px відповідно до вимог стандарту.

Висновки до розділу 3

У третьому розділі виконано повний цикл проєктування дизайну односторінкового вебсайту для потокового перегляду відеоконтенту на тематику кіновсесвіту Marvel у середовищі Figma.

У підрозділі 3.1 сформовано концепцію медіаресурсу, розроблено каркасні схеми у FigJam та створено інтерактивний прототип у Figma з відпрацьованими навігаційними переходами. Тестування дозволило уточнити логіку взаємодії між екранами ще до етапу візуального дизайну.

У підрозділі 3.2 спроектовано дворівневу інформаційну архітектуру з вісьмома тематичними секціями. Обґрунтовано використання повноекранного слайдера замість класичного каталогу. Навігація реалізована через мінімалістичний хедер із трьома пунктами меню та різними сценаріями для авторизованих і неавторизованих користувачів.

У підрозділі 3.3 описано дизайн екранів системи. Реалізовано концепцію «кольорової ідентичності» для кожної сторінки, типографіку на основі Roboto та ключові елементи інтерфейсу. Розроблено форми авторизації через Google та Meta, сторінку профілю користувача та мінімалістичний футер. У підрозділі 3.3.7 перевірено відповідність інтерфейсу стандарту WCAG 2.1 рівня AA: коефіцієнт контрасту для всіх кольорових пар перевищує мінімально допустиме значення 4,5:1, а поля форм забезпечені коректними ARIA-атрибутами.

ВИСНОВКИ

У кваліфікаційній роботі вирішено комплекс теоретичних і практичних завдань, пов'язаних із проєктуванням та розробкою вебсистеми потокового перегляду відеоконтенту відповідно до сучасних стандартів веброзробки та UX/UI-проєктування 2025–2026 рр. За результатами проведеного дослідження сформульовано такі висновки.

Систематизовано актуальні дані щодо стану та динаміки розвитку глобального ринку відеостримінгу. На підставі статистичних матеріалів Precedence Research (лютий 2026 р.) встановлено, що сукупний обсяг ринку досяг 159,98 млрд доларів США у 2025 р. із прогнозованим зростанням до 195,85 млрд доларів у 2026 р. та середньорічним темпом приросту 18,5% на горизонті 2026–2035 рр. Виявлено стійку трансформацію моделей медіаспоживання: за даними Nielsen (травень 2025 р.), частка стримінгу у структурі телеперегляду у США вперше перевищила 45,5%, що підтверджує стратегічну актуальність обраної теми дослідження та обґрунтовує необхідність розробки нових конкурентоспроможних вебплатформ.

Проведено порівняльний аналіз шести провідних платформ відеостримінгу – глобальних (Netflix, YouTube, Amazon Prime Video) та українських (MEGOGO, Sweet.TV, Takflix) – за критеріями функціональності, цінової доступності, якості інтерфейсу та рівня локалізації контенту. Виявлено типові недоліки існуючих рішень: надмірна складність навігаційного інтерфейсу, недостатній рівень доступності для користувачів з обмеженими можливостями, регіональні обмеження контентної бібліотеки та нераціональна цінова політика. Встановлено наявність незадоволеного попиту на україномовні платформи з якісним UX та доступними тарифами, що обґрунтовує доцільність розробки нової вебсистеми.

Уточнено поняття вебсайту відповідно до стандартів W3C та надано класифікацію медіаплатформ як різновиду SPA-вебзастосунків із компонентною архітектурою на базі React. Систематизовано ключові технічні складові сучасної

відеостримінгової системи: адаптивне потокове передавання за стандартами MPEG-DASH та HLS, відеокодеки AV1, H.265 та H.264, CDN-інфраструктура, DRM-захист цифрового контенту та рекомендаційні алгоритми на основі методів машинного навчання. На підставі узагальнення аналізу сформульовано постановку задачі з чіткими функціональними і нефункціональними вимогами до проєктованої вебсистеми, що склали методологічну основу подальшого проєктування.

Обґрунтовано та сформовано чотирирівневий технологічний стек вебсистеми: клієнтська частина на основі HTML5, CSS3, JavaScript ES2025, React 19 та Next.js 15; серверна частина на базі Node.js 22 LTS, Express.js 5 та FFmpeg; рівень збереження даних із реляційною базою PostgreSQL 17 та кеш-сховищем Redis 7; середовище розробки Visual Studio Code із системою версіонування Git і GitHub. Підготовлено порівняльну характеристику обраних технологій та обґрунтовано відповідність кожного компонента актуальним стандартам веброботки 2025–2026 рр. і функціональним вимогам системи. Для підтвердження технічної обґрунтованості стеку розроблено архітектурну схему системи, ER-діаграму бази даних, специфікацію REST API та наведено чотири фрагменти програмного коду, що демонструють реалізацію ключових функцій вебсистеми.

Досліджено архітектуру та функціональні можливості платформи Figma як основного інструменту UX/UI-проєктування. Охарактеризовано модульну структуру платформи (Design Mode, Prototype Mode, Dev Mode, FigJam, Variables/Tokens) та новітні III-інструменти, презентовані на конференції Config 2025 (Make, Draw, Sites, Buzz). За результатами порівняльного аналізу встановлено переваги Figma над конкурентами – Sketch і Adobe XD – за критеріями крос-платформеності, реального колаборативного редагування, глибини функціоналу прототипування та актуальності розвитку продукту, що обґрунтовує вибір Figma як єдиної платформи повного циклу проєктування.

Розроблено концепцію, каркасні схеми та інтерактивний прототип вебсистеми потокового перегляду відеоконтенту у середовищі Figma.

Спроектовано плоску дворівневу інформаційну архітектуру, що включає вісім тематичних секцій фільмів кіновсесвіту Marvel, розділ постерів, підсистему авторизації та сторінку персонального профілю користувача. Встановлено, що відмова від класичного каталогу на користь повноекранного слайдера як основного способу подачі контенту забезпечує вищий рівень емоційного залучення аудиторії та відповідає сучасним тенденціям проєктування медіаплатформ.

Розроблено дизайн інтерфейсу 19 екранів вебсистеми на основі оригінальної концепції «кольорової ідентичності»: кожна тематична сторінка фільму отримала унікальну кольорову схему, що психологічно відповідає образу й атмосфері відповідного персонажа. Типографічну систему побудовано на гарнітурі Roboto Medium (22 px, міжрядковий інтервал 35 px), що забезпечує стабільну читабельність на різноманітних кольорових фонах. Спроектовано інтерфейс OAuth-авторизації через Google і Facebook, сторінку профілю з відображенням вподобаних фільмів — у вигляді повноцінного hi-fi прототипу, готового до програмної реалізації.

СПИСОК ЛІТЕРАТУРИ

1. Архітектурні підходи до розробки масштабованих веб-застосунків. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка». 2024. Том 4, № 24. С. 341–350. DOI: <https://doi.org/10.28925/2663-4023.2024.24.341350> (дата звернення: 23.04.2026).
2. Баутіна М. Розробка надійних систем LLM: принципи проектування та підходи до впровадження. Інформаційні технології та суспільство. 2025. № 3 (18). С. 8–15. DOI: <https://doi.org/10.32689/maup.it.2025.3.1> (дата звернення: 20.04.2026).
3. Богдан Н. І. Розробка веб-додатка генерації творчих ідей. Радіoeлектроніка та молодь у ХХІ столітті : матеріали 28-го Міжнар. молодіж. форуму, 16–18 квітня 2024 р. Харків : ХНУРЕ, 2024. Т. 7. URL: <https://openarchive.nure.ua/entities/publication/96ee12f9-5f44-48f9-83c9-f43d220ecf15> (дата звернення: 22.04.2026).
4. Бродкевич В., Черніченко О. Розробка веб-додатку для управління задачами з реалізацією алгоритмів пріоритизації та сортування. Інформаційні технології та суспільство. 2025. № 4 (19). С. 13–22. DOI: <https://doi.org/10.32689/maup.it.2025.4.2> (дата звернення: 22.04.2026).
5. Вовк О. В., Чеботарьова І. Б., Дейнеко Ж. В. (ред.) Поліграфічні, мультимедійні та web-технології. Інновації та розвиток : монографія. Харків : Друкарня Мадрид, 2024. 265 с.
6. Гуменюк К. В., Січко Т. В. Передові методи анімації вебінтерфейсів: вплив на користувацький досвід. *Прикладні інформаційні технології* : матеріали Всеукр. наук.-практ. конф. Донецький національний університет імені Василя Стуса, м. Вінниця, 2025. С. 13–15.
7. Козуб В. Ю., Семенов М. А., Осадчук С. І. Аналіз впливу WebAssembly на продуктивність складних вебзастосунків. Вісник Херсонського національного технічного університету. 2025. Том 2, № 3(94). DOI: <https://doi.org/10.35546/kntu2078-4481.2025.3.2.28> (дата звернення: 25.04.2026).
8. Масштабований веб-застосунок: морфологічний аналіз та архітектурні рішення : пояснювальна записка до кваліфікаційної роботи. Харків : ХНУРЕ, 2024. URL:

<https://openarchive.nure.ua/server/api/core/bitstreams/a84a9a8d-e3cb-4c7c-8a74-8fdb7732622/content> (дата звернення: 25.04.2026).

9. Михайлишин Р. С. Оновлення програмного коду інтернет-магазину «CyberTochka» : кваліфікаційна робота на здобуття освітнього ступеня «Бакалавр» за спец. 123 Комп'ютерна інженерія. Тернопіль : ВСП «ТФК ТНТУ», 2024. URL: https://elartu.tntu.edu.ua/bitstream/lib/45940/1/Mykhailyshyn_R.pdf (дата звернення: 25.04.2026).

10. Мишківська Я. В., Січко Т. В. Аналіз продуктивності хуків у порівнянні з класовими компонентами. *Комп'ютерні технології обробки даних* : матеріали Всеукр. наук.-практ. конф. Донецький національний університет імені Василя Стуса, м. Вінниця, 2025. С. 132–134.

11. Нескородева Т. В., Федоров Є. Є., Січко Т. В., Нескородева А. Р. Експертні та рекомендаційні системи : навч. посіб. для здобувачів вищої освіти спеціальностей 122 «Комп'ютерні науки», 125 «Кібербезпека», 113 «Прикладна математика». Вінниця : Донецький національний університет імені Василя Стуса, 2022. 208 с.

12. Онлайн застосунок для перегляду фільмів та серіалів з персональними рекомендаціями (ASP.NET Core, React, AWS S3) : пояснювальна записка до кваліфікаційної роботи. Харків : ХНУРЕ, 2024. 66 с. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/fb794722-c91f-469c-af01-2b4c5a227873/content> (дата звернення: 25.04.2026).

13. Павлов Д. Л., Січко Т. В. Принцип роботи Web API та його застосування. *Прикладні інформаційні технології* : матеріали Всеукр. наук.-практ. конф. Донецький національний університет імені Василя Стуса, м. Вінниця, 2023. С. 166–168.

14. Попович В. В. Розробка веб-сайту магазину «Сімейний шопінг» : кваліфікаційна робота на здобуття освітнього ступеня «Бакалавр» за спец. 122 Комп'ютерні науки. Тернопіль : ТНТУ ім. Івана Пулюя, 2024. URL: https://elartu.tntu.edu.ua/bitstream/lib/45882/1/2024_KRB_SNc-41_Попович_В_В.pdf (дата звернення: 23.04.2026).

15. Розробка веб-додатків на основі фреймворків React та NestJS. Матеріали студентської науково-технічної конференції. Вінниця : ВНТУ, 2024. URL: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/49242/23640.pdf> (дата звернення: 19.04.2026).

16. Розробка системи реєстрації та авторизації користувачів у клієнт-серверному web-додатку. Вісник Херсонського національного технічного університету. 2025. Том 2, № 3(94). DOI: <https://doi.org/10.35546/kntu2078-4481.2025.3.2.50> (дата звернення: 25.04.2026).
17. Розробка та оптимізація веб-застосунків з використанням Server-side rendering на базі технологій React та Next.js : пояснювальна записка до кваліфікаційної роботи. Харків : ХНУРЕ, 2024. 41 с. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/1fbd949f-6ca5-4e0a-b717-9c231a81600d/content> (дата звернення: 25.04.2026).
18. Соболенко М. І. Розробка програми (API та вебсайт) для автоматизації отримання та перегляду повідомлень від інших інформаційних сервісів : атестаційна робота здобувача вищої освіти за спец. 151 – Автоматизація та комп'ютерно-інтегровані технології. Харків : ХНУРЕ, 2024. URL: <https://openarchive.nure.ua/entities/publication/5aebda51-5ffe-4646-95c5-f0c5fac6f593> (дата звернення: 25.04.2026).
19. Тихович А. Розробка веб-застосунку музична платформа з використанням Next.js : кваліфікаційна робота на здобуття освітнього ступеня «Бакалавр» за спец. 121 – Програмна інженерія. Тернопіль : ТНТУ ім. Івана Пулюя, 2024. 61 с. URL: https://elartu.tntu.edu.ua/bitstream/lib/45445/1/dyplom_Tyhovych_2024.pdf (дата звернення: 25.04.2026).
20. Чеботарьова І. Б., Черкашина Г. І. Основні тренди UI/UX дизайну 2024 року. Поліграфічні, мультимедійні та web-технології : матеріали Молодіжної школи-семінару IX Міжнар. наук.-практ. конф. Харків : ХНУРЕ, 2024. URL: <https://openarchive.nure.ua/entities/publication/15a90510-29ff-4db3-96de-3567362a8bbb> (дата звернення: 25.04.2026).
21. Ярош В., Лагута В., Смоляков М., Тарасяк І., Футиляк С., Рутар Р. Вплив інноваційного дизайну вебсайту на конверсію та користувацький досвід в електронній торгівлі. Академічні візії. 2024. № 27. С. 1–12. URL: <https://academy-vision.org/index.php/av/article/view/1049/964> (дата звернення: 25.04.2026).
22. Alliance for Open Media. AV1 Bitstream & Decoding Process Specification. 2023. URL: <https://aomediacodec.github.io/av1-spec> (дата звернення: 10.03.2026).

23. AIN.ua. Українські стримінгові платформи: які вони та що можна подивитися. 2026. URL: <https://ain.ua/tech/> (дата звернення: 15.03.2026).
24. Amazon Prime Video. Офіційний сайт. URL: <https://www.primevideo.com> (дата звернення: 10.03.2026).
25. Balsamiq. Balsamiq Wireframes Documentation. URL: <https://balsamiq.com/wireframes> (дата звернення: 02.04.2026).
26. Cloudflare. What is a CDN? How does a CDN work? 2025. URL: <https://www.cloudflare.com/learning/cdn/what-is-a-cdn> (дата звернення: 12.03.2026).
27. Figma Blog. Config 2025 – Figma Make, Draw, Sites, Buzz. Figma. 2025. URL: <https://www.figma.com/blog/config-2025-recap> (дата звернення: 05.04.2026).
28. Garrett J. J. The Elements of User Experience: User-Centered Design for the Web and Beyond. 2nd ed. New Riders Publishing, 2011. 192 p.
29. InspiringApps. Figma in 2025: Human Craft in the Era of AI Tools. URL: <https://www.inspiringapps.com/blog/figma-in-2025-human-craft-in-the-era-of-ai-tools> (дата звернення: 07.04.2026).
30. Marvel App. Marvel Platform Overview. URL: <https://marvelapp.com> (дата звернення: 12.03.2026).
31. MEGOGO. Офіційний сайт. URL: <https://megogo.net> (дата звернення: 15.03.2026).
32. Netflix, Inc. Q4 2025 Shareholder Letter. Netflix Investor Relations. 2026. URL: <https://ir.netflix.net> (дата звернення: 15.03.2026).
33. Netflix. Офіційний сайт. URL: <https://www.netflix.com> (дата звернення: 15.03.2026).
34. Nielsen J. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. 1994 (оновлено 2024). URL: <https://www.nngroup.com/articles/ten-usability-heuristics> (дата звернення: 07.03.2026).
35. Nielsen Norman Group. Prototyping for Usability Testing. URL: <https://www.nngroup.com/> (дата звернення: 03.04.2026).
36. Norman D. The Design of Everyday Things. Revised and Expanded Edition. Basic Books, 2013. 368 p.
37. Precedence Research. Video Streaming Market Size, Share & Trends Analysis Report, 2026–2035. Globe Newswire. 2026. URL: <https://www.globenewswire.com/news->

[release/2026/02/23/3242499/0/en/Video-Streaming-Market-Size-to-Exceed-USD-873-21-Billion-by-2035.html](https://www.researchandmarkets.com/research/2026/02/23/3242499/0/en/Video-Streaming-Market-Size-to-Exceed-USD-873-21-Billion-by-2035.html) (дата звернення: 04.04.2026).

38. React Blog. React 19 Release – React Compiler, Server Components. Meta. 2025. URL: <https://react.dev/blog/2024/12/05/react-19> (дата звернення: 04.04.2026).

39. SPEKA Media. Найпопулярніші стримінгові сервіси в Україні: як дивитись кіно легально. 2025. URL: <https://speka.ua/life/yak-divitisya-kino-legalno-top-8-strimingovix-servisiv-shho-dostupni-v-ukrayini-pk5x1p> (дата звернення: 21.03.2026).

40. SQ Magazine. Figma Statistics 2026: Growth, AI, Global Use. 2025. URL: <https://sqmagazine.co.uk/figma-statistics> (дата звернення: 28.03.2026).

41. Stack Overflow. Developer Survey 2025. URL: <https://survey.stackoverflow.co/2025> (дата звернення: 24.03.2026).

42. Statista. Share of internet traffic from mobile devices worldwide, 2025. URL: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web> (дата звернення: 05.04.2026).

43. Stockhammer T. MPEG-DASH: The Enabler of Adaptive Bitrate Streaming. IEEE Consumer Electronics Magazine. 2022. URL: <https://ieeexplore.ieee.org/document/10032440> (дата звернення: 01.04.2026).

44. Sweet.TV. Офіційний сайт. URL: <https://sweet.tv> (дата звернення: 15.03.2026).

45. Takflix. Офіційний сайт. URL: <https://takflix.com> (дата звернення: 15.03.2026).

46. TC39. ECMAScript 2025 Language Specification. ECMA International. 2025. URL: <https://tc39.es/ecma262> (дата звернення: 05.04.2026).

47. W3C. Architecture of the World Wide Web, Volume One. 2004. URL: <https://www.w3.org/TR/webarch> (дата звернення: 30.03.2026).

48. W3C. Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation. 2018 (оновлено 2023). URL: <https://www.w3.org/TR/WCAG21> (дата звернення: 30.03.2026).

49. YouTube. Офіційний сайт. URL: <https://www.youtube.com> (дата звернення: 15.03.2026).