

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Західноукраїнський національний університет  
Факультет комп'ютерних інформаційних технологій  
Кафедра комп'ютерної інженерії

Варвянський Антон Олегович

## **Онлайн-платформа для потокового відео на основі технології React / Online video streaming platform based on React technology**

спеціальність: 123 – Комп'ютерна інженерія  
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41  
Варвянський Антон Олегович

Науковий керівник  
к.т.н. Савка Н.Я.

Кваліфікаційну роботу допущено  
до захисту:

" \_\_\_ " \_\_\_\_\_ 20\_\_\_ р.

Завідувач кафедри  
\_\_\_\_\_ О.Л. Дубчак

ТЕРНОПЛЬ- 2024

## АНОТАЦІЯ

Кваліфікаційна робота на тему «Онлайн платформа для потокового відео на основі технології React» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 58 сторінок пояснюючої записки, 10 рисунків, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка ефективної онлайн-платформи для потокового відео з використанням технології React.

Методи розробки полягають у застосуванні методів аналізу та синтезу, порівняння, моделювання, захисту інформації.

Розглянуто задачу розробки онлайн-платформи для потокового відео. Описано основні поняття та методи передачі потокового відео онлайн. Проведено дослідження технологій стиснення відео для онлайн використання та аналіз програмно-апаратних засобів відображення потокового відео.

Здійснено алгоритмічно-інформаційне проектування платформи, включаючи структуру системи, алгоритми забезпечення безпеки даних, структуру ієрархічної бази даних та методи оптимізації потокового відео.

Проведено програмно-технологічну реалізацію онлайн-платформи, включаючи розробку API, UI та UX-дизайну, а також тестування платформи.

На основі проведеного аналізу розроблено рекомендації щодо покращення продуктивності та безпеки онлайн-платформ для потокового відео. Надано план дій при типових проблемах, що виникають під час роботи з потоковим відео.

Ключові слова: ПОТОКОВЕ ВІДЕО, ОНЛАЙН-ПЛАТФОРМА, REACT, СТИСКАННЯ ВІДЕО, БЕЗПЕКА ДАНИХ, АЛГОРИТМИ ОПТИМІЗАЦІЇ.

## ANNOTATION

Qualification work on the topic “Online video streaming platform based on React technology” in the specialty 123 “Computer Engineering”, Bachelor's degree contains 58 pages of explanatory note, 10 figures, 2 appendices. The volume of graphic material is 2 sheets of A3 format.

The purpose of the qualification work is to develop an effective online video streaming platform using React technology.

Development methods include the use of methods of analysis and synthesis, comparison, modeling, and information security.

The task of developing an online video streaming platform is considered. The basic concepts and methods of online video streaming are described. The study of video compression technologies for online use and the analysis of software and hardware for displaying streaming video are carried out.

The algorithmic and information design of the platform, including the system structure, data security algorithms, hierarchical database structure, and methods for streaming video optimization, is carried out.

The software and technological implementation of the online platform was carried out, including the development of API, UI and UX design, as well as testing of the platform.

Based on the analysis, recommendations for improving the performance and security of online video streaming platforms were developed. An action plan is provided for typical problems that arise when working with streaming video.

**Keywords: STREAMING VIDEO, ONLINE PLATFORM, REACT, VIDEO COMPRESSION, DATA SECURITY, OPTIMIZATION ALGORITHMS.**

## ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1 Онлайн платформи відображення потокового відео .....	12
1.1 Аналіз методів передачі потокового відео онлайн.....	12
1.2 Дослідження технологій стиснення відео.....	14
1.3 Аналіз програмно-апаратних засобів відображення потокового відео .....	16
1.4 Постановка задачі кваліфікаційної роботи.....	19
2 Алгоритмічне та інформаційне проектування платформи .....	21
2.1 Структура системи .....	21
2.2 Алгоритми забезпечення безпеки даних.....	24
2.3 Структура ієрархічної бази даних .....	28
2.4 Методи оптимізації потокового відео .....	29
3 Програмно-технологічна реалізація онлайн-платформи .....	32
3.1 Програмна реалізація API.....	32
3.2 Розробка UI та UX-дизайну онлайн-платформи .....	34
3.3 Тестування онлайн-платформи.....	39
Висновки .....	42
Список використаних джерел .....	43
Додаток А Світлокопія публікації.....	46
Додаток Б Техніко-економічне обґрунтування розробки веб-платформи .....	50

					КР.КІ.9043911.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата	ОНЛАЙН-ПЛАТФОРМА ДЛЯ ПОТОКОВОГО ВІДЕО НА ОСНОВІ ТЕХНОЛОГІЇ REACT	Літ.	Арк.	Акрушів
						8	58	
Розробив		Варвянський А.О.				ЗУНУ.ФКІТ. КІ-41		
Перевір.		Савка Н.Я.						
Консульт.		Савка Н.Я.						
Н. Контр.		Дубчак Л.О.						
Затвердив		Дубчак Л.О.						

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – прикладний програмний інтерфейс.

MVP – мінімально життєздатний продукт

UI – дизайн інтерфейсу користувача.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

На сьогоднішній день цифрові технології значно впливають на розвиток різних сфер життя, включаючи розважальну індустрію. Потокowe відео є одним із найбільш популярних видів медіаконтенту, який забезпечує миттєвий доступ до фільмів, телепередач, музичних відео та іншого контенту в режимі реального часу. Зростання популярності онлайн-відео обумовлене не лише зручністю його використання, але й можливістю перегляду на будь-якому пристрої з доступом до Інтернету.

В умовах постійного зростання попиту на якісний відеоконтент актуальним стає створення інноваційних платформ для потокового відео. Вони повинні забезпечувати високу якість зображення та звуку, стабільність роботи, а також зручний та інтуїтивний інтерфейс для користувачів.

Сучасні технології, такі як Angular або Vue, мають свої переваги та застосовуються при розробці веб-застосунків, однак, вони можуть не завжди відповідати всім вимогам, що ставляться до сучасних платформ потокового відео. Angular є потужним інструментом для розробки великих корпоративних застосунків, але його складність і крива навчання можуть стати перепоною для нових розробників. Vue, хоч і є легким у засвоєнні та швидким у розробці, проте не забезпечує достатньої гнучкості та масштабованості.

В той же час сьогодні досить популярною є технологія React, що дозволяє створювати масштабовані та інтерактивні платформи, які відповідають вимогам сучасного ринку. React, розроблений Facebook, став вибором багатьох провідних компаній завдяки своїй гнучкості, масштабованості та активній спільноті розробників. Його компонентна архітектура дозволяє легко створювати та підтримувати складні інтерфейси користувача, забезпечуючи високу продуктивність і повторне використання коду. Завдяки віртуальному DOM, React мінімізує кількість оновлень реального DOM, що сприяє покращенню продуктивності застосунку.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, метою кваліфікаційної роботи є розробка надійної та легкої у використанні онлайн платформи для потокового відео, яка забезпечує користувачам доступ до високоякісного контенту, інтерактивні функції та можливості для персоналізації перегляду.

Для досягнення мети роботи необхідно виконати нижчевказані задачі:

- дослідити предметну область;
- провести аналіз існуючих рішень;
- спроектувати загальну структуру системи застосунку;
- описати механізми забезпечення безпеки даних та конфіденційності користувачів;
- спроектувати структуру ієрархічної бази даних;
- описати методи оптимізації потокового відео;
- реалізувати власне API для комунікації компонентів системи;
- розробити UI та UX-дизайн застосунку з врахуванням різних вимог до цільових пристроїв;
- провести тестування готового застосунку на відповідність заданим вимогам та на стабільність функціонування;
- здійснити техніко-економічне обґрунтування розробки проєкту.
- Об'єктом дослідження є процес розробки веб-ресурсів.

Предметом дослідження є технологія React для розробки онлайн-платформи потокового відео.

Практичне значення: розроблена платформа призначена для реальних користувачів для перегляду відеоконтенту, підвищення інтерактивності та задоволення зростаючих потреб у якісному онлайн-відео.

Роботу виконано згідно вимог, що описано у [4-6].

Основні результати дослідження опубліковано на IX науково-практичній конференції «Інтелектуальні комп'ютерні системи та мережі» [1]. Копії публікації наведено у додатку А.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ОНЛАЙН ПЛАТФОРМИ ВІДОБРАЖЕННЯ ПОТОКОВОГО ВІДЕО

## 1.1 Аналіз методів передачі потокового відео онлайн

Потокове відео, або стрімінг, є однією з ключових технологій сучасного інтернету, яка дозволяє користувачам переглядати відеоконтент в реальному часі без необхідності завантаження всього файлу на пристрій. Існує кілька методів та протоколів передачі потокового відео, кожен з яких має свої переваги та недоліки. Розглянемо основні методи, що використовуються в онлайн середовищі.

HTTP Live Streaming (HLS). HLS є одним з найпопулярніших протоколів для передачі потокового відео, розробленим компанією Apple. Основні характеристики HLS наведені нижче.

Сегментація відео, відео поділяється на невеликі сегменти, зазвичай тривалістю 6-10 секунд, що дозволяє забезпечити безперервний перегляд навіть при проблемах з мережею.

Адаптивна швидкість передачі, HLS підтримує різні рівні якості відео (бітрейти), що дозволяє динамічно змінювати якість залежно від доступної пропускної здатності мережі.

HLS підтримується більшістю сучасних браузерів і пристроїв, що робить його універсальним рішенням для потокової передачі.

Dynamic Adaptive Streaming over HTTP (DASH). DASH є конкурентом HLS, розробленим Moving Picture Experts Group (MPEG). Мережевий адаптивний стрімінг – подібно до HLS, DASH підтримує адаптивну зміну якості відео в залежності від мережевих умов. DASH підтримує різні відео- та аудіоформати, включаючи MP4, WebM та інші.

Мультиплатформенність – DASH є відкритим стандартом, що забезпечує його підтримку на різних платформах і пристроях.

Real-Time Messaging Protocol (RTMP) є протоколом, розробленим компанією Adobe для передачі аудіо, відео та даних через інтернет в режимі реального часу.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		



RTMP забезпечує низьку затримку передачі, що робить його популярним для живих трансляцій. Широко використовуваний у поєднанні з Adobe Flash Player, але через відмову від підтримки Flash, його популярність знизилася. Може працювати через HTTP, що дозволяє обходити обмеження мережевих брандмауерів.

Web Real-Time Communication (WebRTC) є сучасним стандартом, який дозволяє передачу аудіо та відео безпосередньо між браузерами без необхідності встановлення додаткових плагінів. Характеристики WebRTC.

1) Підтримка реального часу. WebRTC забезпечує мінімальні затримки, що робить його ідеальним для відеоконференцій та живих трансляцій.

2) Пряма передача. Відео та аудіо передаються безпосередньо між користувачами, що знижує затримки та підвищує якість передачі.

3) Широка підтримка браузерів. WebRTC підтримується більшістю сучасних веб-браузерів, включаючи Chrome, Firefox, Safari та Edge.

Адаптивний стрімінг дозволяє динамічно змінювати якість відеопотоку в залежності від умов мережі та доступних ресурсів пристрою. Основні методи включають:

– bitrate адаптація. цьому методі використовується кілька варіантів одного й того ж відео з різними бітрейтами. Клієнт динамічно перемикається між цими варіантами залежно від доступної пропускної здатності мережі;

– resolution адаптація. Відео кодується в кількох роздільних здатностях.

Клієнт вибирає найкращу роздільну здатність, яку може відтворювати без буферизації. Frame rate адаптація – зміна частоти кадрів може бути використана для зниження навантаження на мережу при збереженні відносно високої якості відео.

Для ефективною передачі відео в мережі використовуються різні стандарти кодування. Найпопулярнішими є:

1) H.264/AVC – один з найпоширеніших відеокодеків, який забезпечує високу якість при відносно низькому бітрейті;

2) H.265/HEVC – наступник H.264, який забезпечує ще кращу компресію

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

та якість, але вимагає більшої обчислювальної потужності для декодування.

3) VP9 – відеокодек, розроблений компанією Google, який забезпечує високу якість при низькому бітрейті та використовується на YouTube.

Для забезпечення високої якості потокового відео важливо обрати правильну інфраструктуру. Основні варіанти:

1) Content Delivery Network (CDN) – Використання CDN дозволяє знизити затримки та підвищити швидкість доставки відео за рахунок розміщення серверів ближче до кінцевих користувачів;

2) Cloud Streaming – використання хмарних платформ, таких як AWS, Azure або Google Cloud, для масштабування та управління відеопотоками.

Передача потокового відео супроводжується рядом викликів:

1) Буферизація та затримки – забезпечення безперебійного відтворення відео без затримок і буферизації є критичним завданням;

2) Безпека – захист відеоконтенту від піратства та несанкціонованого доступу;

3) Сумісність – забезпечення роботи на різних пристроях та браузерах з різними технічними характеристиками.

Аналіз існуючих методів передачі потокового відео в онлайн середовищі показує, що для створення ефективної онлайн платформи необхідно використовувати сучасні протоколи та технології, що забезпечують високу якість, надійність та безпеку відеоконтенту. Вибір конкретного методу залежить від конкретних вимог та умов використання, що дозволяє забезпечити найкращий користувацький досвід [7].

## 1.2 Дослідження технологій стиснення відео

Дослідження різних технологій стиснення відео для передачі в режимі онлайн дозволить визначити їхню ефективність та придатність для використання

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

в конкретному контексті. Технологія стиснення відео є невід'ємною частиною процесу запису і збереження кадрів, одержуваних за допомогою відеокамер безпеки. Вона стискає сирі файли в менший за розміром формат. Така технологія дозволяє зберігати на жорсткому диску більший обсяг відеоматеріалів. Крім того, стиснення відеофайлів дозволяє відправляти файли по мережі без істотних затримок, що особливо важливо для віддаленого перегляду за допомогою смартфона або планшета. Стислі відеофайли займають менше місця на жорсткому диску, це дозволяє зберігати більше матеріалів або зберігати файли довше.

Методи стиснення відео.

– H.264/AVC (Advanced Video Coding): цей стандарт стиснення відео є одним з найпоширеніших і використовується для передачі відео в онлайн середовищі. Він забезпечує ефективне стиснення при дотриманні високої якості відтворення.

– H.265/HEVC (High Efficiency Video Coding): наступне покоління стандарту H.264, H.265, забезпечує ще більш ефективне стиснення відео при збереженні якості. Він дозволяє передавати відео з більш високою роздільною здатністю та меншим обсягом даних.

– VP9: розроблений Google, VP9 є конкурентом H.265 та надає подібну якість відео при меншому обсягу даних. Цей формат широко використовується для відеострімів на YouTube та інших платформах.

– AV1 (AOMedia Video 1) – цей відкритий стандарт стиснення відео, розроблений AOMedia, пропонує ще більшу ефективність в порівнянні з H.265 та VP9. AV1 дозволяє зменшити обсяг даних при збереженні високої якості відтворення.

– LCEVC (Low Complexity Enhancement Video Coding) – цей підхід до стиснення відео комбінує стандартні кодеки, такі як H.264 або H.265, з елементами перехресної інтерполяції, що дозволяє покращити якість відтворення при зменшенні обсягу даних.

Комбіновані методи:

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

– Adaptive Bitrate Streaming (ABR) – цей підхід комбінує різні кодеки та роздільні здатності для забезпечення оптимальної якості відтворення в залежності від швидкості Інтернет-з'єднання користувача та характеристик пристрою.

– Multi-Rate Layered Coding – цей метод дозволяє розділити відеопотік на декілька рівнів якості, що дозволяє користувачам вибирати найбільш підходящий рівень якості відтворення в залежності від їхніх потреб та можливостей мережі.

Аналіз попередніх робіт та реалізацій в області стиснення відео дозволяє зрозуміти, які методи та технології є найбільш ефективними для онлайн використання. Він також вказує на те, які аспекти можуть бути поліпшені та доповнені у майбутніх дослідженнях [8].

Проблеми та виклики:

– баланс між якістю та обсягом даних: вибір оптимального кодеку або підходу до стиснення відео часто вимагає знаходження балансу між якістю відтворення та розміром файлу;

– підтримка різних пристроїв та мереж: забезпечення сумісності та оптимальної продуктивності на різних пристроях та у різних умовах мережі може бути складним завданням.

Відео є ще довго залишатиметься найпопулярнішим медіаформатом. Ідеальне поєднання аудіо та візуальних компонентів дозволяє йому передавати інформацію, розважати і навіть розвивати творчі здібності так, як жоден інший формат не може. У цілому, дослідження технологій стиснення відео для онлайн використання включає аналіз різних методів та кодеків з метою визначення їхньої ефективності та придатності для конкретного використання.

### 1.3 Аналіз програмно-апаратних засобів відображення потокового відео

Сучасні технології потокового відео використовують широкий спектр програмних та апаратних засобів для забезпечення якісного відтворення контенту.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Це включає різноманітні програми, бібліотеки, фреймворки, а також апаратні рішення, які забезпечують оптимальну продуктивність, сумісність та масштабованість. У цьому розділі розглянемо основні програмні та апаратні засоби, що використовуються для відображення потокового відео.

Серед програмних засобів для відображення потокового відео важливу роль відіграють відеоплеєри. HTML5 Video Player підтримується більшістю сучасних браузерів без необхідності встановлення додаткових плагінів, забезпечуючи адаптивне потокове відео через HLS і DASH, і легко інтегрується з JavaScript бібліотеками та фреймворками.

Відкритий відеоплеєр Video.js, заснований на HTML5, підтримує велику кількість плагінів для розширення функціоналу, таких як субтитри, аналізатор бітрейту, реклами, і сумісний з різними відеоформатами, включаючи HLS, DASH, MP4. Plug відзначається простотою використання, легкістю налаштування та інтеграції, мінімалістичним дизайном та підтримкою відео, аудіо, YouTube та Vimeo контенту.

Для розробки відеододатків часто використовуються JavaScript бібліотеки та фреймворки. Наприклад, React дозволяє створювати повторно використовувані компоненти для відтворення відео і має велику кількість бібліотек та плагінів для інтеграції з відеоплеєрами та сервісами. Redux забезпечує централізоване управління станом додатків, що особливо корисно для великих проектів з великою кількістю відеоконтенту, і легко інтегрується з React для створення масштабованих додатків. RxJS, завдяки реактивному програмуванню, дозволяє ефективно обробляти асинхронні події, що особливо корисно для стрімінгу та обробки відео в реальному часі.

Серверні технології також відіграють ключову роль у потоковому відео. Node.js завдяки подієвому циклу забезпечує високу продуктивність та масштабованість для обробки великої кількості одночасних підключень, а також підтримує модулі, як fluent-ffmpeg для обробки відео в реальному часі. Nginx, як сервер HTTP, забезпечує високу продуктивність та можливість використання як сервера для розповсюдження відеоконтенту з підтримкою протоколу RTMP для

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

живих трансляцій [9].

Апаратні засоби для відображення потокового відео. Клієнтські пристрої відіграють важливу роль у забезпеченні якісного відтворення відео. Сучасні смартфони та планшети оснащені мобільними процесорами, такими як Qualcomm Snapdragon та Apple A-series, які забезпечують високу продуктивність для відтворення відео високої чіткості. Екрани з високою роздільною здатністю підтримують 4K та HDR відео, що забезпечує високу якість зображення.

Комп'ютери та ноутбуки з дискретними відеокартами від NVIDIA та AMD забезпечують апаратне декодування відео, знижуючи навантаження на центральний процесор та підтримуючи одночасне відтворення відео на декількох екранах. Смарт телевізори, завдяки вбудованим додаткам для стрімінгових сервісів, таких як Netflix, YouTube та Amazon Prime Video, підтримують 4K та HDR для високої якості зображення. На серверному боці використовуються потужні сервери для відеострімінгу з процесорами Intel Xeon або AMD EPYC для обробки великої кількості одночасних підключень та трансляцій.

Графічні прискорювачі, такі як NVIDIA Tesla, використовуються для апаратного кодування та декодування відео. Мережеві комутатори та маршрутизатори з високою пропускну здатністю, підтримкою 10G та 40G Ethernet забезпечують швидку та надійну передачу відеоконтенту. QoS технології дозволяють пріоритизувати відеопотоки у мережі, забезпечуючи стабільне відтворення без затримок.

Важливу роль відіграє також CDN (Content Delivery Network), яка забезпечує глобальну мережу серверів для розподілу контенту по всьому світу, знижуючи затримки та підвищуючи швидкість доставки відео. Кешування контенту дозволяє зберігати часто запитуваний контент ближче до кінцевих користувачів, зменшуючи навантаження на основний сервер.

Сучасні технології та інновації. Сучасні технології та інновації включають нові кодеки та формати відео, такі як AV1, який забезпечує високу ефективність стиснення та відкрите ліцензування, що знижує витрати на використання. H.266/VVC пропонує ще вищу ефективність стиснення та підтримку нових

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

форматів відео, таких як 8К.

Технології доставки контенту, такі як Low Latency Streaming, включають WebRTC, який забезпечує мінімальні затримки для живих трансляцій, та CMAF (Common Media Application Format), який дозволяє зменшити затримку при використанні HLS та DASH. Edge Computing, який використовує обчислювальні потужності ближче до кінцевих користувачів, знижує затримки та підвищує продуктивність.

Сучасні програмні та апаратні засоби для відображення потокового відео пропонують широкий спектр можливостей для забезпечення високоякісного користувацького досвіду. Вибір конкретних технологій та інструментів залежить від вимог проекту, доступного бюджету та умов, що дозволяє знаходити оптимальні рішення для будь-яких завдань у сфері потокового відео [10].

#### 1.4 Постановка задачі кваліфікаційної роботи

Розглянуті вище онлайн-платформи потокового відео часто стикаються з проблемами ефективності, що виражаються в значних витратах на обробку та передачу даних, неефективному використанні мережевих ресурсів та погіршенні якості обслуговування кінцевих користувачів. Це призводить до затримок, зниження якості відео та інших негативних наслідків, які знижують задоволення користувачів. Кінцевий споживач не завжди отримує відеоконтент з потрібною якістю, а витрати на ресурси при цьому залишаються високими.

Проаналізовано зовнішні фактори, що впливають на функціонування онлайн-платформ потокового відео. Серед основних факторів виділено пропускну здатність мережі, затримки передачі даних, завантаженість серверів та кінцевих пристроїв користувачів. Виходячи з цього, обґрунтовано розробку системи керування потоковим відео, яка враховує зміни зовнішніх чинників.

Таким чином, сформульовано мету кваліфікаційної роботи, а саме створення онлайн платформи для потокового відео, яка дозволить користувачам

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

переглядати відеоконтент у високій якості, надавати інтерактивні можливості та забезпечувати персоналізацію користувацького досвіду.

Для реалізації такої платформи необхідно вирішити наступні задачі:

- спроектувати загальну структуру системи застосунку;
- описати механізми забезпечення безпеки даних та конфіденційності користувачів;
- спроектувати структуру ієрархічної бази даних;
- описати методи оптимізації потокового відео;
- реалізувати API для комунікації компонентів системи;
- розробити UI та UX-дизайн платформи з урахуванням різних вимог до цільових пристроїв;
- провести тестування готового застосунку на відповідність заданим вимогам та стабільність функціонування;
- здійснити техніко-економічне обґрунтування розробки проєкту.

Також зазначена розробка включає синтез MVP, який повинен виконувати функції та володіти характерними ознаками.

1) Веб-версія та кросплатформа. Платформа буде реалізована як PWA, що забезпечить доступність на будь-якій платформі з підтримкою сучасних веб-браузерів.

2) Авторизація. Платформа повинна мати механізми захисту від несанкціонованого доступу, включаючи реєстрацію, авторизацію та відновлення доступу у разі втрати паролю.

3) Список відео та їх аналіз. Можливість перегляду списку відео за обраний період часу та детальний перегляд кожного відео, включаючи метадані та статистику переглядів.

4) Розширений пошук. Фільтрування відео за різними критеріями, такими як жанр, тривалість, наявність субтитрів та мова оригіналу.

5) Незалежність від доступу до сервісів. Платформа повинна забезпечувати можливість перегляду завантажених відео офлайн, зберігаючи дані в локальній базі даних.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		



## 2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ПРОЄКТУВАННЯ ПЛАТФОРМИ

### 2.1 Структура системи

Як було описано у параграфі 1.4, проєкт буде реалізовано як онлайн платформа потокового передавання відео з використанням фреймворку React. Онлайн платформи для потокового передавання відео мають кілька переваг порівняно з традиційними методами доставки медіаконтенту:

- миттєвий доступ до контенту. Онлайн платформи не потребують завантаження файлів на пристрій користувача, що дозволяє миттєво починати перегляд відео. Це знижує час очікування та забезпечує кращий користувацький досвід;

- доступність на різних пристроях. Платформи для потокового передавання відео можуть бути доступними на будь-якому пристрої з Інтернет-підключенням і веб-браузером. Це дозволяє користувачам переглядати відео на різних операційних системах та пристроях, таких як смартфони, планшети, комп'ютери та смарт-телевізори;

- легкість оновлення. Оновлення контенту та функціоналу онлайн платформ можуть бути зроблені безпосередньо на сервері. Це означає, що користувачі завжди мають доступ до найновіших версій без необхідності встановлення оновлень на свої пристрої.

Звичайно, є деякі недоліки онлайн платформ порівняно з традиційними методами доставки медіаконтенту. Наприклад, якість потокового передавання може залежати від швидкості Інтернет-з'єднання користувача, а також можливі затримки при великому навантаженні на сервери. Однак, для більшості користувачів переваги у вигляді доступності, зручності та зниження витрат на розробку значно переважають недоліки, що робить онлайн платформи для потокового передавання відео оптимальним вибором для багатьох застосунків [11].

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

В якості технологічного стеку для розробки онлайн-платформи обрано такий набір технологій:

- React для Frontend;
- Node.js та Nest для Backend;
- PostgreSQL в якості бази даних;
- Vercel для хостингу ресурсів застосунку;
- Render для хостингу Backend-частини застосунку.

React для Frontend. React є надзвичайно популярною бібліотекою для розробки інтерфейсів користувача, що надає можливість створювати компоненти, які можуть бути повторно використані та розширені. Вона також має широку спільноту, що полегшує процес її вивчення та підтримки. Крім того, React забезпечує роботу як зі статичними, так і з динамічними сторінками, що сприяє легкому та швидкому створенню масштабованих застосунків.

Node.js та Nest для Backend. Node.js є популярним серверним середовищем JavaScript, яке дозволяє розробникам використовувати JavaScript як мову програмування для створення серверної частини застосунку. Це забезпечує можливість застосування тих самих знань та інструментів, що використовуються для розробки фронтенд-частини. Nest, у свою чергу, є фреймворком для Node.js, який дозволяє швидко створювати API та веб-додатки. Він характеризується простим та зрозумілим API та підтримує широкий спектр плагінів та модулів, що полегшує розробку та розширення функціональності веб-додатків.

PostgreSQL є потужною реляційною базою даних з відкритим вихідним кодом, відомою своєю надійністю, стабільністю та функціональністю. Вона дозволяє ефективно зберігати, оновлювати та запитувати дані, а також забезпечує високу продуктивність і масштабованість для застосунків будь-якого масштабу. PostgreSQL має багатий набір функцій, що робить її ідеальною для складних додатків з високими вимогами до зберігання даних та аналітики.

Vercel для хостингу ресурсів застосунку: використання цього сервісу для хостингу ресурсів проекту забезпечує надійне шифрування по протоколу HTTPS, а також високу швидкість з'єднання. Vercel надає чудову платформу для хостингу

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

ресурсів вашого проєкту, забезпечуючи швидке та безпечне з'єднання.

Його можливості автоматичного розгортання та інтеграції з популярними інструментами розробки дозволяють ефективно керувати вашими проєктами, забезпечуючи високу продуктивність і зручність для користувачів.

Render для хостингу Backend-частини платформи. Render – це сучасний хмарний хостинг, який забезпечує зручний та ефективний спосіб розгортання та хостингу веб-застосунків. Він надає потужні можливості для масштабування додатків відповідно до потреб, що дозволяє збільшувати потужність сервера при збільшенні навантаження на додаток. Render також забезпечує високий рівень безпеки та захисту при розгортанні застосунків в інтернеті.

В першу чергу варто розібратись з розгортанням застосунку. Загалом платформуможна розділити на 3 компоненти, які повинні бути розгорнуті. UML-діаграму розгортання представлено на рисунку 2.1 із таким позначеннями:

- клієнтська частина, яка виконується на пристрої клієнта, використовуючи браузер як середовище виконання коду, і хоститься за допомогою сервісу Vercel;

- серверна частина, яка виконується в контейнеризованому сервері Render, використовуючи Node.js як середовище виконання коду;

база даних PostgreSQL, яка зберігається та опрацьовується за допомогою сервісу Render. Детальний приклад наведено в КР.КІ. 9043911.00.00.000 С1

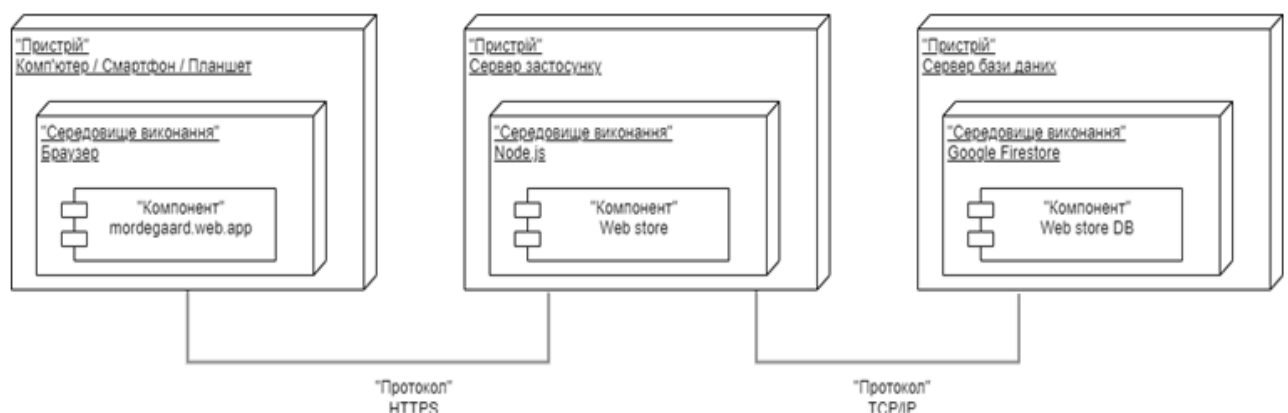


Рисунок 2.1 – UML діаграма розгортання системи

Взаємодія між серверною та клієнтською частинами здійснюватиметься за допомогою REST API. REST (Representational State Transfer) API є архітектурним стилем для розробки програмного забезпечення, який використовується для створення веб-сервісів та API. REST API використовує принципи та протоколи Інтернету, такі як HTTP (Hypertext Transfer Protocol), для забезпечення обміну даними між клієнтами та серверами.

Основними характеристиками REST API є використання URI (Uniform Resource Identifier), за допомогою яких кожен ресурс REST API представляється та ідентифікується його місцезнаходженням. Для взаємодії з ресурсами в REST API використовуються HTTP-методи. Найпоширенішими методами в REST API є GET (для отримання ресурсів), POST (для створення ресурсів), PUT (для оновлення або редагування ресурсів) та DELETE (для видалення ресурсів). Дані в REST API зазвичай передаються у форматі JSON завдяки його простоті та зручності використання [12].

Отже, обраний стек технологій дозволяє розробникам швидко створювати масштабовані та ефективні веб-застосунки без значних фінансових затрат, а проста схема розгортання застосунку забезпечує можливість оперативного та простого впровадження нових оновлень у майбутньому.

## 2.2 Алгоритми забезпечення безпеки даних

Компоненти застосунку обмінюються даними, використовуючи захищені протоколи, як-от HTTPS при обміні даними між клієнтською та серверною компонентами застосунку. Проте шифрування під час обміну даними не є достатнім, аби забезпечити необхідний рівень безпеки, тому варто попіклуватись над шифруванням даних, які зберігатимуться в базі даних, та над шифруванням чутливих даних, які передаватимуться по спроектованому HTTP REST API між серверною та клієнтською компонентами.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення високого рівня безпеки паролів користувачів буде використовуватись алгоритм хешування bcrypt. Bcrypt є потужним інструментом для хешування паролів, який робить процес розгадування паролів вкрай складним і ресурсомістким. Основні характеристики bcrypt:

- сіль (Salt): Кожен пароль хешується з унікальною сіллю. Це забезпечує унікальність хешу навіть для однакових паролів, що значно ускладнює атаки типу «rainbow table»;
- настроюваний параметр складності (Cost Factor). Bcrypt дозволяє налаштувати кількість ітерацій, необхідних для хешування. Це робить процес більш ресурсомістким і ускладнює атаки грубої сили. Параметр складності можна налаштувати відповідно до вимог безпеки;
- стійкість до атак грубої сили (Brute Force Resistance). Повільність алгоритму забезпечує стійкість до атак грубої сили, оскільки час, необхідний для перевірки кожного можливого паролю, значно збільшується.
- захист від атак зі словником (Dictionary Attacks).

Використання унікальної солі для кожного паролю робить атаки зі словником менш ефективними [13].

Процес хешування паролів включає такі етапи:

- генерація солі. Для кожного паролю генерується унікальна сіль. Сіль додається до паролю перед процесом хешування;
- хешування паролю: пароль разом із сіллю хешується за допомогою алгоритму bcrypt, з використанням визначеної кількості ітерацій.

Згенерований хеш разом із сіллю зберігається у базі даних. Важливо зазначити, що сам пароль ніколи не зберігається у відкритому вигляді. Сіль має зберігатись у змінних середовища. Це необхідно, щоб секрети були відокремлені від коду програми, щоб запобігти випадковому витоку цих секретів на хмарні системи. Блок-схему цього алгоритму представлено на рисунку 2.2.

Процес AES-CBC розшифрування даних з БД та подальшої відправки цих даних користувачу відбувається за наступним алгоритмом:

- сервер отримує зашифровані дані з бази даних;

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

- сервер розшифрує отримані дані, використовуючи ключ бази даних;
- у випадку успішної розшифровки, сервер форматує розшифровані дані та повторно їх шифрує, використовуючи вже ключ для передачі даних.

Блок-схему зазначеного алгоритму представлено на рисунку 2.3.



Рисунок 2.2 – Блок-схема алгоритму збереження чутливих даних у базі даних



Рисунок 2.3 – Блок-схема алгоритму відправки чутливих даних клієнту зі сторони сервера

Основою безпеки застосунку є використання надійних сервісів із власними механізмами захисту, такими як наявність HTTPS-сертифікатів у Vercel та SSH-шифрування при розгортанні коду сервера на платформі Render. Що стосується захисту чутливих персональних даних, то використання bcrypt-алгоритму забезпечує ефективне шифрування та дешифрування даних, додаючи додатковий рівень безпеки [14].

## 2.3 Структура ієрархічної бази даних

Як було описано вище, проєкт використовує реляційну базу даних PostgreSQL. У реляційних базах даних інформація представляється у вигляді набору взаємопов'язаних таблиць. Кожна таблиця складається з рядків і стовпців, де рядки представляють окремі записи, а стовпці – атрибути цих записів. Взаємозв'язки між таблицями встановлюються за допомогою ключів (первинних і зовнішніх), що дозволяє ефективно організувати і управляти даними [15].

В PostgreSQL модель даних реалізується у вигляді таблиць, які можуть мати різні типи взаємозв'язків, такі як один-до-одного, один-до-багатьох та багато-до-багатьох. Це забезпечує гнучкість у моделюванні реальних світових відносин між даними. PostgreSQL підтримує строгий контроль типів даних, що забезпечує цілісність даних і спрощує валідацію і маніпуляцію даними.

На вершині ієрархії БД буде 2 основні колекції: Users та Auth. В колекції Auth зберігаються токени авторизація користувача який здійснив вхід у систему. Це потрібно для того щоб при повторному запиті на закриті ендпоінти роботи валідацію чи цей токен згенерувала наша система і в залежності від результату робити вхід або відображати користувачу помилку авторизації. До кожного запису є посилання до конкретного користувача, який здійснив вхід.

В колекції Users зберігаються всі зареєстровані користувачі. Включно з їх налаштуваннями, ключами доступу та додатковими даними, які асоціюються з конкретним користувачем (улюблені відео наприклад).

Приклад представлення цих даних в інтерфейсі Visual Studio Code (Extensions DataBase) зображено на рисунку 2.4. На ньому також можна помітити автоматично згенеровані токени доступу кожного користувача.

Незважаючи на ієрархічну структуру, PostgreSQL дозволяє робити комплексні запити – шукати документи за значенням конкретного поля, сортувати результати, об'єднувати документи, масово оновлювати документи, тощо.

Детальний приклад наведено в КР.КІ.9043911.00.00.000 С2

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		



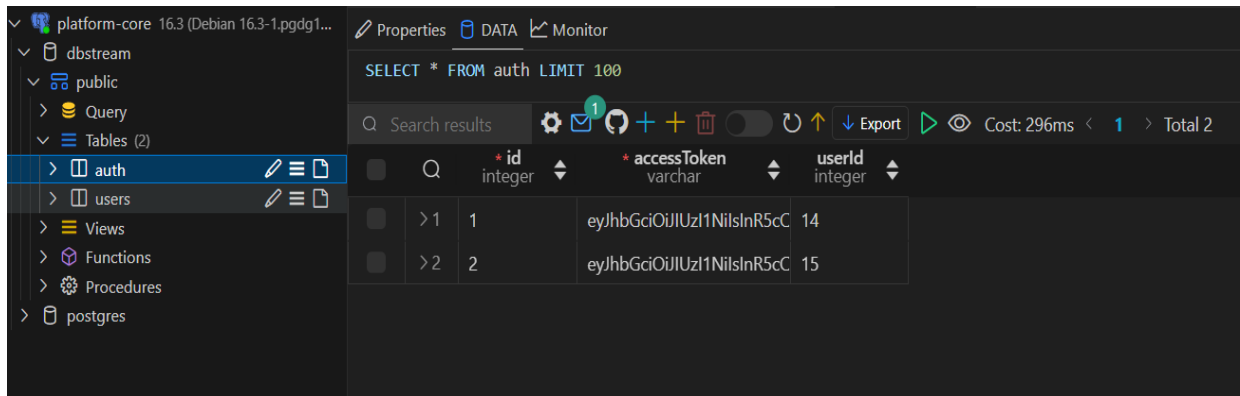


Рисунок 2.4 – Приклад бази даних в інтерфейсі Visual Studio Code

Єдина вимога – зарання провести індексацію необхідних полів перед створенням комплексних запитів. При цьому, завдяки розділенню основної інформації на 2 окремі колекції, можливо використовувати особливості реляційних баз даних для створення пошукових запитів.

Структура бази даних, яка була описана, відзначається не лише високою масштабованістю, але й спрощує процес створення запитів до бази даних, що суттєво полегшує розробку програмного забезпечення. Використання правил дозволяє забезпечити безпеку бази даних від несанкціонованого доступу та помилок зі сторони розробника [16].

## 2.4 Методи оптимізації потокового відео

Оптимізація потокового відео є важливою складовою при розробці онлайн платформ потокової передачі відео, таких як наш застосунок. Це дозволяє забезпечити високоякісний користувацький досвід, зменшити затримки, оптимізувати використання мережевих ресурсів та знизити витрати на інфраструктуру [17].

У цьому підпункті розглянемо основні методи та техніки оптимізації потокового відео, враховуючи використання API від The Movie Database (TMDb) для отримання відео. Одним із найефективніших методів оптимізації потокового

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

відео є адаптивне потокове передавання (Adaptive Bitrate Streaming, ABR).

Ця технологія дозволяє змінювати якість відеопотоку в реальному часі відповідно до поточних умов мережі та ресурсів пристрою користувача. ABR використовує кілька версій одного і того ж відео з різною якістю та швидкістю передачі. Клієнтський плеєр автоматично перемикається між цими версіями на основі таких факторів, як швидкість інтернет-з'єднання, ресурси пристрою та завантаженість сервера. Зокрема, ABR дозволяє знизити якість відео при погіршенні мережевих умов, щоб уникнути буферизації, а також враховує потужність процесора, доступну пам'ять та інші ресурси, щоб забезпечити стабільне відтворення відео. Крім того, ABR може знижувати навантаження на сервер за рахунок зменшення якості відео для окремих користувачів у разі високого трафіку. Популярні протоколи ABR включають HTTP Live Streaming (HLS) від Apple та Dynamic Adaptive Streaming over HTTP (DASH) [18].

Мережі доставки контенту (Content Delivery Networks, CDN) є ще одним важливим методом оптимізації потокового відео. CDN складається з розподіленої мережі серверів, які кешують контент і доставляють його кінцевим користувачам з найближчого географічного розташування. Це забезпечує зниження затримки, оскільки користувачі отримують відео з найближчого серверу, що зменшує затримки в передачі даних. Також, завдяки розподіленій природі CDN, навіть у разі виходу з ладу одного з серверів, контент буде доставлений з іншого доступного серверу. Це зменшує навантаження на основний сервер, який зберігає оригінальний контент, тоді як CDN забезпечує його кешування та доставку, знижуючи навантаження на головну інфраструктуру [19].

Використання API від The Movie Database (TMDb) надає широкі можливості для оптимізації контенту та покращення користувацького досвіду на нашій платформі. TMDb API дозволяє швидко отримувати метадані про фільми, серіали та акторів, що забезпечує швидке завантаження інформації про контент без необхідності зберігати великий обсяг даних на власних серверах.

Гнучкі пошукові запити, підтримувані API, дозволяють ефективно знаходити потрібний контент і доставляти його користувачам. Крім того, TMDb

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

надає доступ до високоякісних зображень і постерів, що покращує візуальну привабливість платформи [20]. Використання зображень з CDN знижує навантаження на наші сервери, а актуальність даних, які постійно оновлюються в TMDb, гарантує актуальність контенту на нашій платформі.

Кешування та передзавантаження контенту є важливими методами для покращення продуктивності платформи та зменшення часу завантаження відео. Кешування на клієнті, зберігаючи часто використовуваний контент в кеші браузера або мобільного додатку, зменшує кількість запитів до серверу та покращує швидкість доступу до відео. Передзавантаження (pre-fetching) дозволяє завантажувати наступні частини відео або ресурси до того, як вони знадобляться користувачу, що знижує затримки при відтворенні відео.

Використання ефективних відеокодеків та форматів грає ключову роль в оптимізації потокового відео. Сучасні відеокодеки, такі як H.264, H.265 (HEVC) та AV1, дозволяють знизити розмір відеофайлів без втрати якості, що зменшує обсяг переданих даних та покращує якість потоку. Формати контейнерів, такі як MP4, WebM та MKV, забезпечують ефективне зберігання і передачу відео та аудіо даних, а також метаданих [21].

Оптимізація потокового відео на нашій платформі включає комплексні методи та технології, що забезпечують високу якість користувацького досвіду, мінімізують затримки та оптимізують використання мережевих ресурсів. Використання адаптивного потокового передавання, CDN, TMDb API, кешування та передзавантаження, а також оптимізація відеокодеків і форматів сприяють створенню ефективного і масштабованого відео-сервісу.

Крім того, важливо забезпечити безперервний моніторинг і аналіз продуктивності, щоб вчасно виявляти проблеми та впроваджувати необхідні поліпшення для підтримання високої якості обслуговування користувачів.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

## 3 ПРОГРАМНО-ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ОНЛАЙН-ПЛАТФОРМИ

### 3.1 Програмна реалізація API

Для клієнт-серверної комунікації у застосунку було створено REST API, використовуючи фреймворк Nest.js та допоміжні бібліотеки та функції. Створення API включає наступні компоненти:

- розробка власної системи URL-шляхів;
- розробка проміжних маршрутів для автентифікації користувача та обробки помилок;
- розробка допоміжних класів для конструювання JSON-відповідей.

Як було описано в пункті 2.1, URI використовується для опису місцезнаходжень кінцевих точок API. URI може мати вигляд URL (Uniform Resource Locator) або URN (Uniform Resource Name). Для реалізації власного API, я використовую URL-адреси, які складаються зі схеми, доменного імені та шляху до ресурсу (це і є місцезнаходження кінцевих точок). Оскільки сервер хоститься за допомогою сервісу Render, типова URL-адреса ресурсу виглядає наступним чином: <https://platform-core.onrender.com/path/to/endpoint>, де `https` – це схема, `mrdgrd.herokuapp.com` – це доменне ім'я, а `path/to/endpoint` – це шлях до кінцевої точки API [22].

Для створення шляхів, варто користуватися принципом SEF URL (Clean URL), тобто URL-адреси мають бути зручні для сприйняття людиною та використовувати знайомі слова. Правильні URL-адреси:

- GET <https://platform-core.onrender.com/favourite-movie> (отримання вподобаних відео);
- POST <https://platform-core.onrender.com/add-favourite/:id> (додавання до списку вподобаних відео шляхом передачі `:id`);
- POST <https://platform-core.onrender.com/auth> (авторизація користувача).

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

### Неправильні URL-адреси:

- GET `https://platform-core.onrender.com/new/transactions-list` (недопустиме чередування різних прописів);
- POST `https://platform-core.onrender.com /settings/post` (не варто створювати окремі шляхи для кожного HTTP-методу).

Код, що відповідає за обробку запитів до створених маршрутів, розподіляється на окремі контролери, кожен з яких відповідає за обробку певного набору маршрутів. Наприклад, контролер `FavouriteController.ts` відповідає за функціонал пов'язаний з пошуком вподобаних відео, їх оновленням та відображенням деталей конкретного відео.

Крім маршрутів, існують також проміжні маршрути з кодом, який виконується обов'язково перед передачею у відповідний контролер. В цих проміжних маршрутах описується логіка, пов'язана з аутентифікацією користувача та обробкою помилок. Таким чином, всі маршрути мають кінцевий проміжний маршрут `ErrorHandler`, який при виникненні помилки формує зрозуміле клієнту повідомлення про помилку (див. рис. 3.1). Це дозволяє клієнтській частині програмного забезпечення отримувати детальні повідомлення про помилки та відповідно реагувати на них. Наприклад, при спробі авторизації у застосунку з відсутнім користувачем з відповідною електронною поштою та паролем, сервер повертає помилку з кодом 401 та повідомленням "Bad credentials".

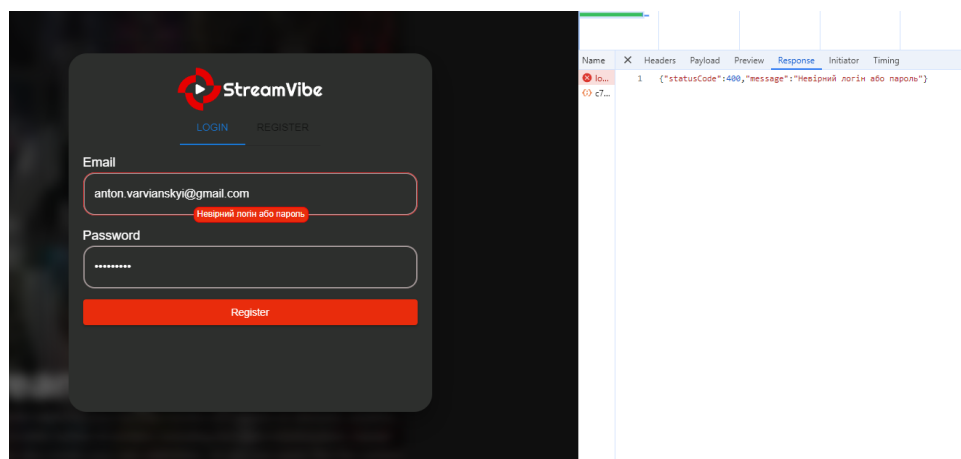


Рисунок 3.1 – Показ деталей помилки при авторизації

					КР.КІ. 9043911.00.00.000 ПЗ	Арк. 33
Змн.	Арк.	№ докум.	Підпис	Дата		

Стандартні коди помилок визначаються відповідно до стандартів статусів відповідей HTTP, що сприяє легкому впізнаванню таких помилок браузером та робить інтерфейс програмного забезпечення простішим для зовнішніх розробників. Це створює основу для взаємодії між клієнтом та сервером. Розроблене API можна легко масштабувати та додавати нові кінцеві точки, що є критичним для ефективної роботи веб-застосунків.

### 3.2 Розробка UI та UX-дизайну онлайн-платформи

Уся верстка застосунку створена за допомогою UI бібліотеки React для створення динамічної та «реактивної» верстки. Завдяки використанню React для верстки та Vite для збірки проекту, уся клієнтська частина застосунку реалізована за принципом all-in-one – усі компоненти інтерфейсу, їх логіка, стилі та іконки зібрані в єдиному файлі, який можна легко закешувати. В розробці зовнішнього вигляду застосунку та стилі графічних елементів використовується UI-бібліотека Material UI. Це одна з найпопулярніших UI бібліотек для розробки веб-інтерфейсів. Вона надає набір готових стилів, компонентів та JavaScript-інструментів, які допомагають швидко та зручно будувати сучасні дизайни та легко реалізовувати адаптивність за рахунок системи кінцевих точок (breakpoints), за допомогою яких задаються стилі елементів інтерфейсу для різних розмірів екрана [23].

Основою UX та UI застосунку є парадигми, описані в методичці Google Material UI, а саме:

- застосування системи сітки для розміщення об'єктів на екрані, з урахуванням пропорцій, відступів та інтервалів між елементами. Це сприяє створенню збалансованого та привабливого дизайну;
- користання яскравих, насичених кольорів та їх поєднань для виділення ключових елементів інтерфейсу, створення ієрархії та передачі настрою. Material

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- Design надає рекомендації щодо використання кольорів, тіней та їх переходів;
- використання анімацій та переходів для забезпечення плавності та просторовості в інтерфейсі. Це допомагає візуально показати зв'язок між об'єктами та станами, підкреслити важливі події та додати динамічності взаємодії;
- використання чіткої та зрозумілої типографіки для полегшення читання та передачі інформації. Material UI пропонує рекомендації щодо вибору шрифтів, їх розмірів та відступів між текстовими елементами;
- забезпечення простоти та зручності використання інтерфейсу для користувачів, що включає такі аспекти, як чітка навігація, доступність елементів управління, інтуїтивно зрозумілі жести та дії, а також консистентність стилів та взаємодії;
- рекомендація використовувати адаптивний дизайн, що пристосовується до різних розмірів екранів та пристроїв, забезпечуючи зручне використання на різних платформах;
- використання контекстних меню та дій для забезпечення швидкого доступу до відповідних опцій та функцій, які пов'язані з конкретним об'єктом або контекстом.

При плануванні та реалізації інтерфейсу платформи, потрібно дотримуватись вказаних парадигм.

У веб-розробці система сітки реалізується за допомогою використання системи Grid та Flexbox, що дозволяє зручно створювати адаптивну верстку для сторінок, на якій розміри різних елементів адаптуються до розмірів екрану.

Основу дизайну складають темні кольори в поєднанні з червоним. Наприклад, як показано на рисунку 3.2, інтерфейс використовує яскравий червоний колір для акцентування важливих елементів, таких як заголовки розділів, кнопки та іконки популярних фільмів. Цей підхід покращує візуальну привабливість та зручність користування платформою, допомагаючи користувачам швидко орієнтуватися та знаходити потрібний контент [24].

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

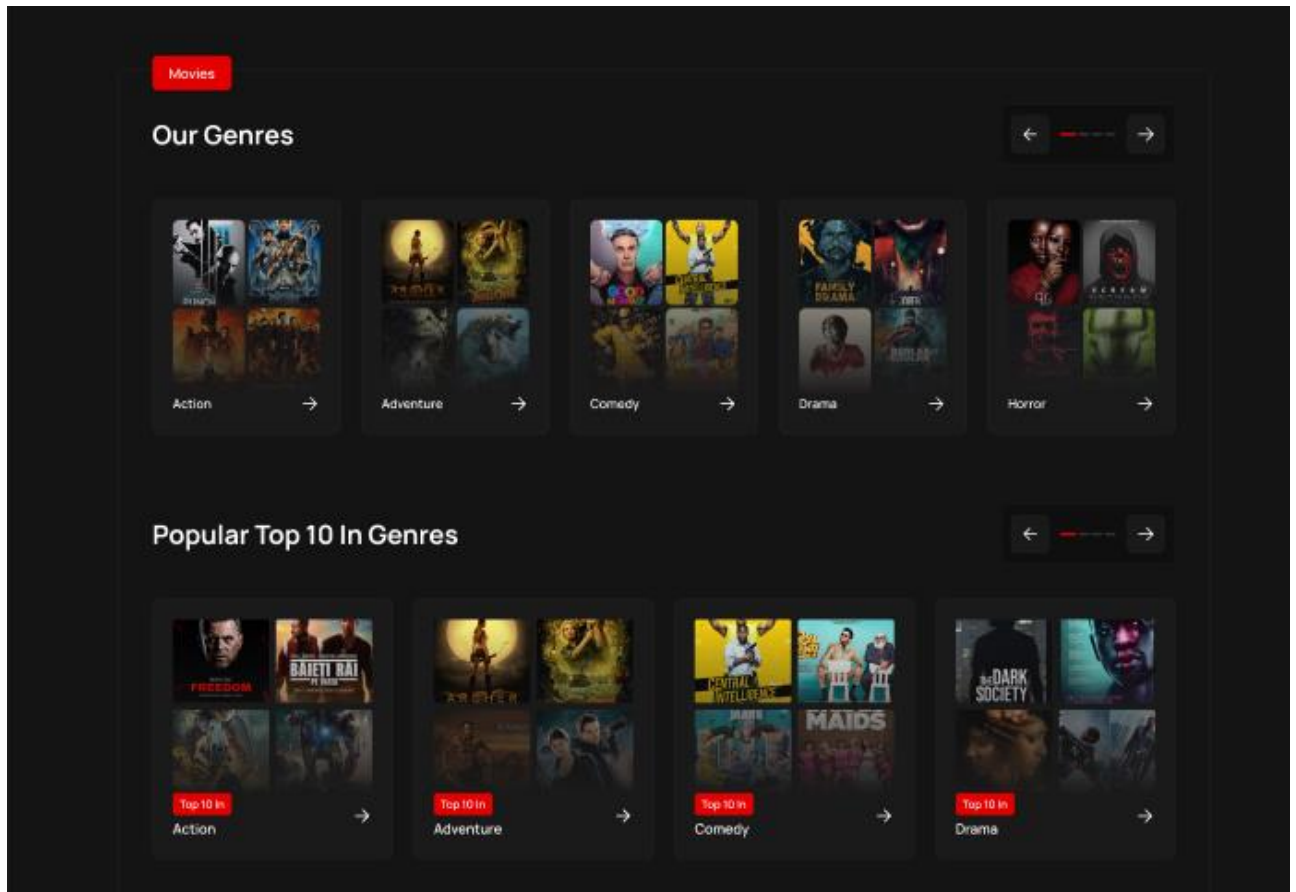


Рисунок 3.2 – Приклад використання кольорів

Користувачів онлайн-платформи можна поділити за типом пристрою, який вони використовують для взаємодії із застосунком. Різні пристрої надають різні можливості для взаємодії користувача з платформою, тому інтерфейс повинен адаптуватися до цих можливостей. Основними платформами, на яких буде працювати застосунок, є смартфони (тачскрін, малий розмір екрану, портретна орієнтація), планшети (тачскрін, великий розмір екрану, довільна орієнтація) та персональні комп'ютери (миша та клавіатура, великий розмір екрану, альбомна орієнтація).

Інтерфейс користувача має адаптуватися до вимог цільових платформ. У певних випадках вся верстка застосунку може відрізнитися залежно від платформи. Наприклад, мобільні пристрої мають значно менші екрани порівняно з ПК, тому інтерфейс мобільних застосунків повинен бути більш компактним та мінімалістичним. У мобільних застосунках меню зазвичай розташовується у верхньому, нижньому або боковому рядку, як це показано на рисунку 3.3, щоб

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		



максимально звільнити екранний простір. У застосунках для ПК меню може бути розміщене у верхній частині екрана або мати окрему панель інструментів.

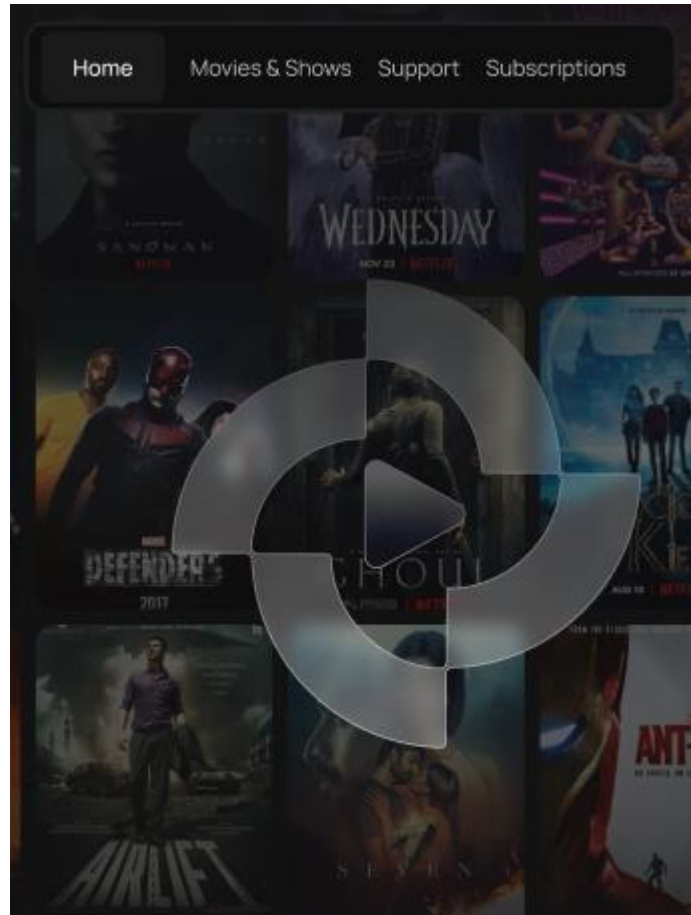


Рисунок 3.3 – Приклад навігаційного меню, розташованого зверху екрана

Для забезпечення плавних переходів між фільмами на онлайн-платформі для потокового відео використовується бібліотека *swiper/react*. Вона дозволяє створювати адаптивні слайдери з анімаційними ефектами, що гарантують приємний користувацький досвід незалежно від типу пристрою. Ця бібліотека підтримує різні методи взаємодії, такі як кліки мишею та дотики пальцем по сенсорному екрану, що полегшує навігацію контентом як на настільних комп'ютерах, так і на мобільних пристроях. Приклад того, як виглядає слайдер, створений за допомогою бібліотеки *swiper/react*, представлено на рисунку 3.4, де можна бачити розділ *Trending Now* з популярними фільмами [25].

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

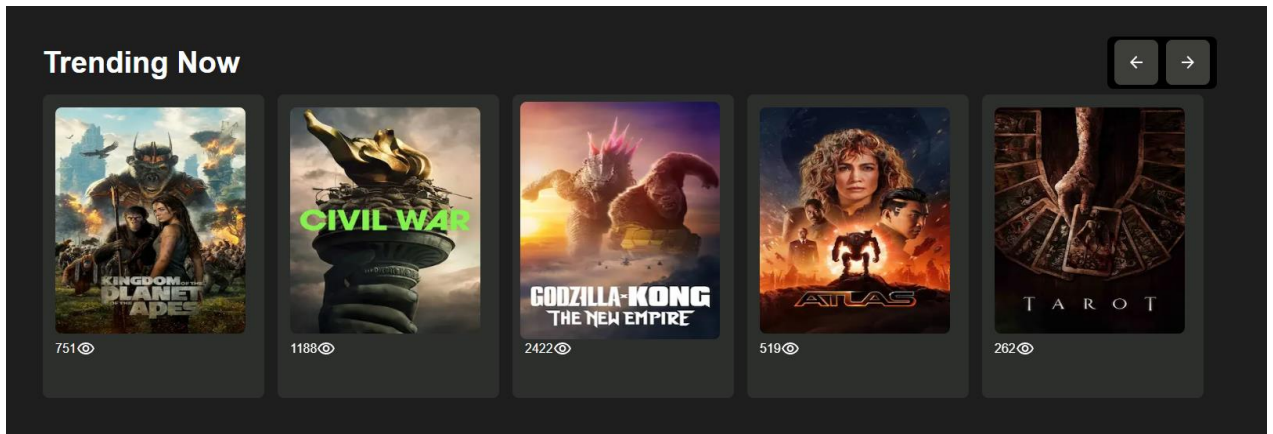


Рисунок 3.4 – Приклад слайдеру

Для відображення відео на онлайн-платформі для потокового відео використовується підхід, коли посилання на відео приходять через API, а саме відео вбудовується за допомогою тега `iframe`. Цей метод дозволяє легко інтегрувати різні відео джерела та забезпечувати стабільну роботу програвача на різних пристроях.

На прикладі рисунка 3.5 показано, як виглядає інтерфейс відтворення відео, вбудованого через тег `iframe`. Такий підхід дозволяє зберігати єдиний стиль інтерфейсу та забезпечувати крос-платформену сумісність.

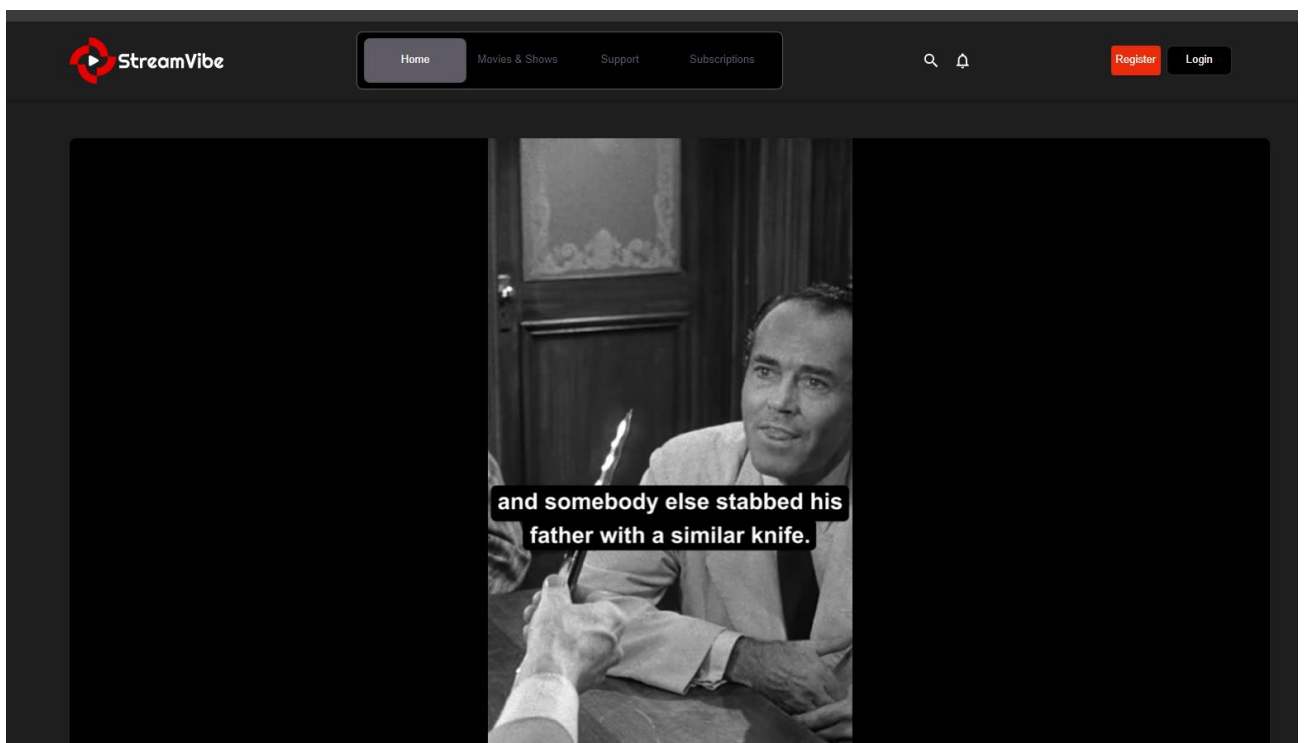


Рисунок 3.5 – Приклад показу уривку з фільму “12 Angry Man”

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Відео може бути відображене як у портретній, так і в альбомній орієнтації, залежно від типу пристрою та уподобань користувача. Інтерфейс платформи динамічно адаптується до розміру екрану, забезпечуючи зручність перегляду на будь-якому пристрої.

Цей метод вбудовування та відображення відео через iframe також дозволяє легко впроваджувати додаткові функції, такі як субтитри, налаштування якості відео та інші елементи керування, що підвищують загальну зручність користувача.

Таким чином, розроблений графічний інтерфейс користувача не лише оптимізований для різноманітних цільових пристроїв, але й естетично привабливий та, що особливо важливо, зручний у користуванні.

### 3.3 Тестування онлайн-платформи

Тестування MVP базується на кількох принципах, які допомагають перевірити припущення та отримати важливі відгуки від користувачів. В першу чергу варто дослідити, чи має MVP реалізований функціонал, який потрібен користувачу. Як було описано в параграфі 3.2, користувач очікує бачити наступні функції в MVP: веб-версія та кросплатформність, система авторизації, каталог відео, персоналізовані рекомендації, розширений пошук та незалежність від доступу до офіційних серверів.

Завдяки реалізації застосунку як прогресивного веб-додатку, користувачі можуть переглядати каталог відео, особисті плейлисти та отримувати рекомендації навіть без встановлення офіційного мобільного застосунку. Це забезпечує зручний доступ до платформи з будь-якого пристрою, включаючи ПК, планшети та смартфони.

Авторизація – платформа передбачає форму авторизації та верифікації користувача, а також використовує різні механізми шифрування для захисту

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

даних користувачів. Це гарантує безпечний доступ до персоналізованої інформації та налаштувань користувача. Каталог відео: як було описано в параграфі 3.2, каталог відео на платформі дозволяє користувачам переглядати різноманітні відео, включаючи фільми, серіали та документальні фільми. Каталог структурується за жанрами, рейтинговими оцінками та популярністю. Платформа також пропонує можливість створювати та керувати персональними плейлистами.

Персоналізовані рекомендації – завдяки алгоритмам машинного навчання, платформа пропонує користувачам індивідуальні рекомендації на основі їхньої історії переглядів та оцінок. Це сприяє користувачам у знаходженні нового контенту, що відповідає їхнім уподобанням. Приклад тестування швидкості запитів демонструє рисунок 3.6.

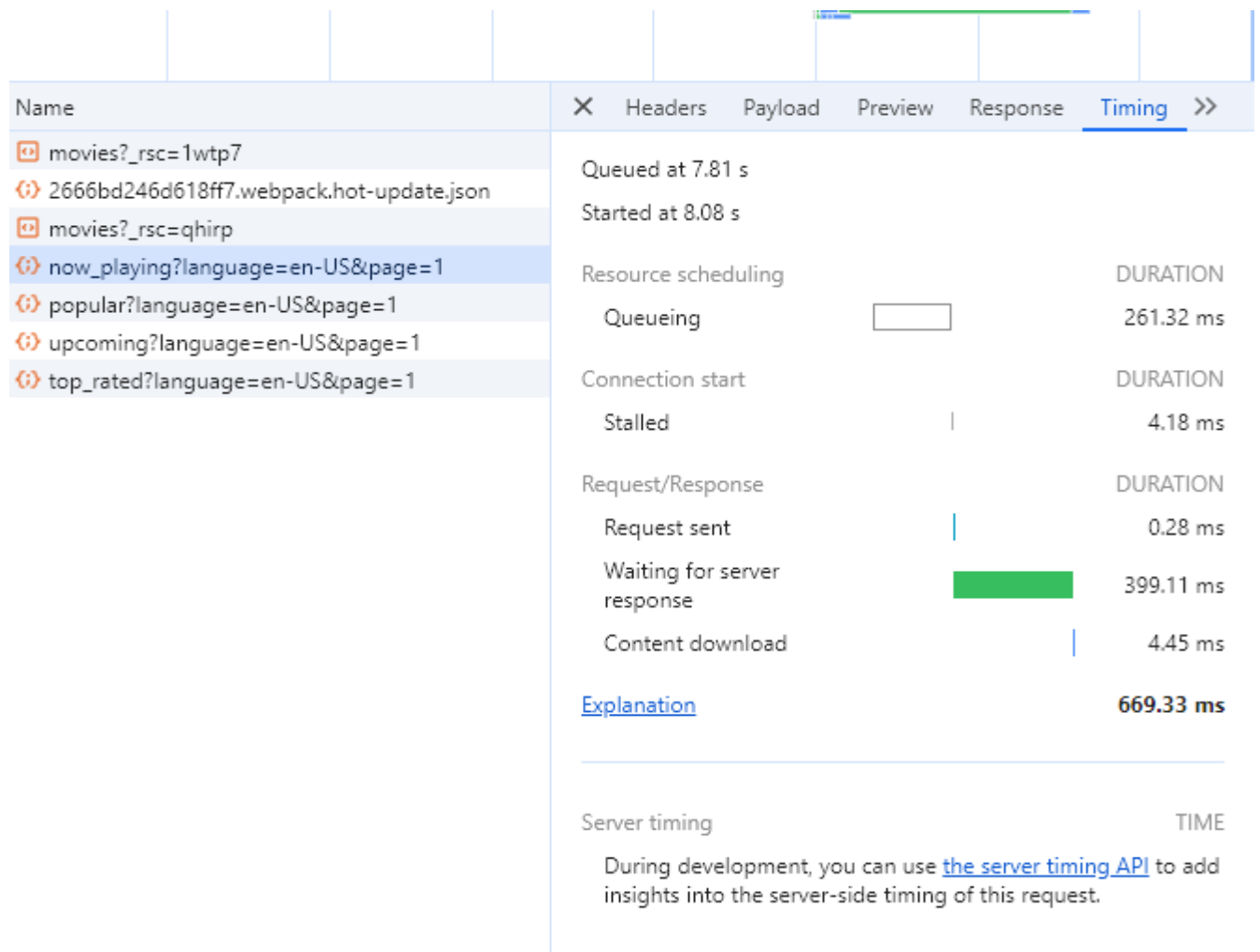


Рисунок 3.6 – Тестування швидкості HTTP-запитів

Щодо продуктивності платформи, є ще можливості для покращення. Наприклад, після відправлення запиту на отримання популярних відео (20 відео) потрібно 669 мс для отримання відповіді. Основну частину цього часу платформа очікує відповіді від сервера, тоді як передача даних займає лише незначні частки секунди. Подальша оптимізація на стороні сервера дозволить зменшити цей час і, відповідно, прискорити завантаження даних у застосунку.

Кешування не значно впливає на швидкість завантаження програми, однак саме кешування може стати проблематичним без надійного механізму очищення застарілого кешу, що ускладнює розгортання нових функцій. Це питання можна вирішити шляхом розподілу програми на окремі частини, які будуть завантажуватися асинхронно за потреби. Крім того, обробка кешування за допомогою Service Worker дозволить валідувати застарілий кеш та видаляти його.

Таким чином, можна зробити висновок, що програма виконує всі поставлені завдання, а її технічні обмеження та недоліки не є критичними для користувацького досвіду.

Основні розрахунки щодо обґрунтування ефективності розробки веб-платформи наведено у додатку Б.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

1. Проведено аналіз предметної області ринку платформ передачі потокового відео, що дозволило виділити основні технології та протоколи для забезпечення безперебійного та якісного відтворення відеоконтенту.

2. Проаналізовано основні методи адаптивного стрімінгу, які дозволяють динамічно змінювати якість відеопотоку залежно від умов мережі, що забезпечує оптимальний користувацький досвід. В результаті було сформульовано задачі для реалізації проєкту.

3. Досліджено методи забезпечення інформаційної безпеки при роботі з чутливими персональними даними користувачів та розглянуто методи аналізу фінансових транзакцій, що створило інформаційне підґрунтя для подальшої реалізації проєкту.

4. Обрано технологічний стек застосунку на основі аналізу сучасних технологій веб-розробки та методів забезпечення безпеки даних користувачів, що дозволило визначити найбільш підходящі бібліотеки та фреймворки для реалізації проєкту.

5. Спроектовано структуру системи застосунку, сформовано вимоги до методів комунікації між клієнтом та сервером, розроблено UI та UX застосунку з урахуванням різних вимог до цільових пристроїв, що забезпечило зручність та адаптивність застосунку.

6. Проведено тестування готової реалізації застосунку на відповідність заданим вимогам та стабільність функціонування, що дозволило виявити та виправити поточні недоліки. Результат можна переглянути та протестувати за наступною адресою: <https://stream-platform-nine.vercel.app/>.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Варвянський А.О. Онлайн платформа потокового відео на основі технології React. ІХ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 21 травня 2024 р. Тернопіль. Україна. с.16.

2. ДСТУ 3008:2015 Національний стандарт України. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Введ. 01.07.2017. К.: ДП “УкрНДНЦ”, 2016. 25 с.

3. ДСТУ 8302:2015 Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Введ. 01.07.2016. К.: ДП “УкрНДНЦ”, 2017. 16 с.

4. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько, О.Й. Піцун / Під ред. Л.О.Дубчак. Тернопіль: ЗУНУ, 2024. 54 с.

5. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем» / Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

6. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

7. Emergen Research. Progressive Web Application Market By Offering (Services, Tools & Libraries), By Application (Dynamic Web Application, Static Web Application), By End-user (Small- & Medium-sized Enterprises, Large Enterprises), By Industry Vertical (Retail & E-commerce, Media & Entertainment), and By Region Forecast to 2027. URL: <https://www.emergenresearch.com/industry-report/progressive-web-application-market> (дата звернення: 07.06.2024).

					КР.КІ. 9043911.00.00.000 ПЗ	Арк. 43
Змн.	Арк.	№ докум.	Підпис	Дата		

8. 8.Twitter Lite PWA Significantly Increases Engagement and Reduces Data Usage. Web.dev. URL: <https://web.dev/twitter/> (дата звернення: 07.06.2024).

9. Material Design Guidelines. URL: <https://m2.material.io/> (дата звернення: 07.06.2024).

10. TMDb API. URL: <https://developer.themoviedb.org/> (дата звернення: 07.06.2024).

11. 4 Most Used REST API Authentication Methods. RestCase. URL: <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/> (дата звернення: 07.06.2024).

12. What is cipher block chaining? Techtarget. URL: <https://www.techtarget.com/searchsecurity/definition/cipher-block-chaining> (дата звернення: 07.06.2024).

13. Timing vulnerabilities with CBC-mode symmetric decryption using padding. URL: <https://learn.microsoft.com/en-us/dotnet/standard/security/vulnerabilities-cbc-mode> (дата звернення: 07.06.2024).

14. Block cipher mode of operation. Wikipedia. URL: [https://en.wikipedia.org/wiki/Block\\_cipher\\_mode\\_of\\_operation](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation) (дата звернення: 07.06.2024).

15. HTTPS. Wikipedia. URL: <https://uk.wikipedia.org/wiki/HTTPS> (дата звернення: 07.06.2024).

16. Setting up a remote notification server. Apple Developer. URL: [https://developer.apple.com/documentation/usernotifications/setting\\_up\\_a\\_remote\\_notification\\_server](https://developer.apple.com/documentation/usernotifications/setting_up_a_remote_notification_server) (дата звернення: 07.06.2024).

17. Cloud Render Documentation. URL: <https://docs.render.com/> (дата звернення: 07.06.2024).

18. Cloud Vercel Documentation. URL: <https://vercel.com/docs> (дата звернення: 07.06.2024).

19. Кубічний сплайн. Wikipedia. URL: [https://uk.wikipedia.org/wiki/Кубічний\\_сплайн](https://uk.wikipedia.org/wiki/Кубічний_сплайн) (дата звернення: 07.06.2024).

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

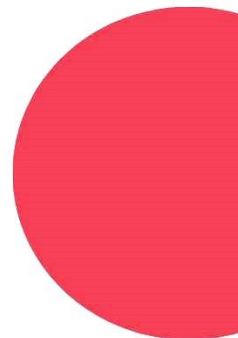


20. Linear regression. Wikipedia. URL: [https://en.wikipedia.org/wiki/Linear\\_regression](https://en.wikipedia.org/wiki/Linear_regression) (дата звернення: 07.06.2024).
21. Ordinary least squares. Wikipedia. URL: [https://en.wikipedia.org/wiki/Ordinary\\_least\\_squares](https://en.wikipedia.org/wiki/Ordinary_least_squares) (дата звернення: 07.06.2024).
22. Understanding Polynomial Regression Model. Analytics Vidhya. URL: <https://www.analyticsvidhya.com/blog/2021/10/understanding-polynomial-regression-model/> (дата звернення: 07.06.2024).
23. Logistic regression. Wikipedia. URL: [https://en.wikipedia.org/wiki/Logistic\\_regression](https://en.wikipedia.org/wiki/Logistic_regression) (дата звернення: 07.06.2024).
24. Fitting a Trend Line for Linearly Dependent Data Values. TIBCO. URL: [https://docs.tibco.com/pub/sfire-dsc/6.5.0/doc/html/TIB\\_sfire-dsc\\_user-guide/GUID-141E4F08-1478-451C-AAFF-BFF4279E2BBE.html](https://docs.tibco.com/pub/sfire-dsc/6.5.0/doc/html/TIB_sfire-dsc_user-guide/GUID-141E4F08-1478-451C-AAFF-BFF4279E2BBE.html) (дата звернення: 07.06.2024).
25. Graham D., Veenendaal V. E., Evans I., Rex B., "Foundations of Software Testing." Cengage Learning, 2018. 48-51 с.
26. Node.js – Run JavaScript Everywhere: веб-сайт. URL: <https://nodejs.org/en> (дата звернення 02.05.2024 р.).
27. React: веб-сайт. URL: <https://uk.legacy.reactjs.org> (дата звернення 12.04.2024 р.).
28. Дудник О. Методи та засоби технічної організації онлайн трансляцій відеоігор: квал. робота. Харків:ХНУРЕ, 2023. 93 с.
29. Рудніченко М.Д., Шибасєва Н.О. Навчальний посібник з дисципліни "Сучасна теорія управління ІТ-проектами" для студентів спеціальності - 126 Інформаційні системи і технології.Одеса: ОНПУ, 2020. 132 с.
30. Козловський В.О. Техніко-економічне обґрунтування та економічні розрахунки в дипломних проектах та роботах: навч. посібник. Вінниця:ВДТУ, 2003. 75 с.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А  
СВІТЛОКОПІЯ ПУБЛІКАЦІЇ

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ



ІХ НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ  
МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ  
«ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА  
МЕРЕЖІ»

21 ТРАВНЯ 2024



[KI.WUNU.EDU.UA/CONFERENCE/](http://KI.WUNU.EDU.UA/CONFERENCE/)

ТЕРНОПІЛЬ

2024



					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

<i>Мотрук Г.Ю.</i>	
Вебінтерфейс студентського гуртка «DevOps» .....	7
<i>Матіяш Ю.Р.</i>	
Розробка DevOps пайплайну для програми обробки зображень .....	8
<i>Березький М. О.</i>	
Аналіз метрик порівняння зображень .....	9
<i>Рутецький Ю.О.</i>	
Інформаційна система оптимізації витрат енергоносіїв розумного будинку .....	10
<i>Медвідь А. Я.</i>	
Маніпуляція об'єктами з використанням роботуки в реальному середовищі на прикладі відкривання дверей вбиралень .....	11
<i>Теслюк С. В.</i>	
Моделі та засоби IT-рекрутингу з використанням методів багатокритеріальної оптимізації.....	13
<i>Боровий О.О., Ілечко Р.І.</i>	
Засоби забезпечення автономності БПЛА з використанням методів комп'ютерного зору.....	14
<i>Ілечко Р.І., Боровий О.О.</i>	
Методи квантування нейромерж для платформ БПЛА.....	15
<i>Варвянський А.О</i>	
Онлайн платформа для потокового відео на основі технології React .....	16
<i>Чирська Т. В.</i>	
Алгоритми порівняння скелетонів .....	17
<i>Сич Т. А.</i>	
Аналіз програмних засобів для генерації тексту в текст .....	18
<i>Бадзь В. М.</i>	
Розроблення моделі визначення авторства англomовних текстів з використанням згорткових нейронних мерж.....	19
<i>Каштальян О.В.</i>	
Побудова інформаційної системи віртуальної примірки прикрас .....	20
<i>Рудик В.В.</i>	
Вебінтерфейс модуля обліку абітурієнтів кафедри комп'ютерної інженерії .....	21
<i>Копія А.В.</i>	
Аналіз методів і алгоритмів тестування та діагностики комп'ютерної техніки .....	22
<i>Мацюк М. І.</i>	
Модель управління крокуючими платформами на базі штучних нейронних мерж.....	23
<i>Коваль А. В.</i>	
Методи і засоби оптимізації ресурсів у хмарних системах з мікросервісною архітектурою.....	24
<i>Глод С. І.</i>	
Інтегрування можливостей Edge AI через використання Edge-enabled безпілотників у Військові операції.....	25
<i>Андрухі Б., Плекан В.</i>	
Моделі та технології штучних нейронних мерж для програмних додатків з елементами штучного інтелекту.....	27
<i>Воротній В., Ігнатєв Т.</i>	
Аналіз алгоритмів розпізнавання цифрових зображень та сфери їх застосування .....	28
<i>Мірецький В.В.</i>	
Системи автоматизації розумного будинку .....	29

<i>Газда Н. Б.</i>	Інтеграція віртуальної та доповненої реальності у симулятори-тренажери .....	30
<i>Свистун А. А.</i>	Динамічне керування багатоядерними комп'ютерними системами .....	31
<i>Красниця В., Батько Ю.М.</i>	Технології автоматизованої реєстрації та передачі показників лічильників побутових споживачів.....	32
<i>Домбровський М. О.</i>	Синтез біомедичних зображень на основі дифузійно – імовірнісних моделей .....	34
<i>Голубець В.М.</i>	Засоби підвищення ефективності технічної діагностики з використанням туманних обчислень.....	35
<i>Ляцинський П.Б.</i>	Аналіз медичних систем комп'ютерної діагностики .....	36
<i>Сагаль М.В.</i>	Розробка бази даних для зберігання та аналізу метаданих зображень .....	38
<i>Фольварочний Д.А.</i>	Аналіз переваг розумних будинків .....	39
<i>Мамчур Т.Б.</i>	Засоби реалізації обміну даними для мобільних платформ .....	40
<i>Квітень Д.О.</i>	Проектування вебінтерфейсу для управління ресурсами інтернету речей .....	41
<i>Пасько В.В., Стеренчак І.В.</i>	Основні етапи 3D-моделювання для створення реалістичних зображень .....	42
<i>Мудрик А.Р., Заяць І.В.</i>	Підвищення ефективності та якості автоматизації процесів тестування програмного забезпечення.....	43
<i>Мельник М.В.</i>	Інтеграція методів машинного навчання в системи прогнозування генерації сонячних електростанцій.....	44
<i>Юзефович В.І.</i>	Програмні засоби захисту корпоративних мереж на основі фаєрволів .....	46
<i>Івануха В.Р., Береза Б.І.</i>	Системи автоматизації розумного будинку .....	47
<i>Палка М. В.</i>	Аналіз програмних засобів генеративного штучного інтелекту .....	48

## ОНЛАЙН ПЛАТФОРМА ДЛЯ ПОТОКОВОГО ВІДЕО НА ОСНОВІ ТЕХНОЛОГІЇ REACT

**Вступ.** У сучасному світі цифрових технологій потокове відео займає одну з ключових позицій серед розважальних та освітніх індустрій. Зростання популярності онлайн-відео пояснюється зручністю доступу до контенту в будь-який час і на будь-якому пристрої. Актуальність розробки онлайн платформи для потокового відео обумовлена потребою в задоволенні цих вимог, а також можливістю забезпечити високу якість відеоконтенту, інтерактивність та персоналізацію користувацького досвіду. [1].

**Постановка задачі.** Об'єкт дослідження – онлайн платформа для потокового відео. Предмет дослідження – використання бібліотеки React як інструмент для розробки. Мета дослідження – розробка онлайн платформи для потокового відео за допомогою бібліотеки React.

**Основний матеріал.** На сьогодні наявність ефективної онлайн платформи для потокового відео є ключовим фактором успіху для багатьох компаній та стартапів. Основні аспекти системи включають:

1) алгоритм адаптивного потокового відео. Система використовує алгоритми для адаптивного потокового відео, що забезпечують високу якість зображення та плавний стрімінг, незалежно від якості інтернет-з'єднання користувача. Цей підхід дозволяє автоматично змінювати роздільну здатність і бітрейт відео в залежності від швидкості інтернету, зменшуючи буферизацію і забезпечуючи безперервний перегляд. [2].

2) інтерактивний користувацький інтерфейс. Використання React дозволяє створювати інтуїтивно зрозумілий та швидкий інтерфейс, який підлаштовується під потреби користувачів і надає можливості для кастомізації та персоналізації [3].

3) алгоритм рекомендацій контенту. На основі даних про перегляди та вподобання користувачів впроваджується алгоритм рекомендацій, який покращуватиме користувацький досвід шляхом пропозиції релевантного контенту. Використовуючи машинне навчання та аналітику даних, система може виявляти інтереси користувачів та пропонувати їм нові відео, які відповідатимуть їхнім уподобанням, збільшуючи час взаємодії з платформою та утримання користувачів. [4].

**Висновки.** Практичне значення розробки онлайн платформи для потокового відео полягає у забезпеченні високоякісного, інтерактивного та масштабованого рішення, яке сприяє зростанню бізнесу у цій галузі. Створення такої платформи дозволяє покращити якість надання відео контенту та забезпечує можливість ефективного управління та монетизації цього контенту.

### Список літератури

1. The future of video streaming: why React is a game changer. URL: <https://www.techradar.com/features/the-future-of-video-streaming-why-react-is-a-game-changer>

2. Building a Scalable Video Streaming Platform with React. URL: <https://www.smashingmagazine.com/2021/10/building-scalable-video-streaming-platform-react/>

3. Understanding Adaptive Bitrate Streaming. URL: <https://www.streamingmedia.com/Articles/Editorial/What-Is-.../Understanding-Adaptive-Bitrate-Streaming-102367.aspx>

4. Digital Rights Management (DRM) in Streaming Media. URL: <https://www.wowza.com/blog/digital-rights-management-drm-streaming-media>

									Арк.
									49
Змн.	Арк.	№ докум.	Підпис	Дата					

## ДОДАТОК Б

### ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ВЕБ-ПЛАТФОРМИ

#### Б.1 Визначення витрат на розробку програмного забезпечення

Розроблена система реалізує веб-платформу для потокового відео на основі технології React. Дослідження базується на технології React, яка забезпечує прості та зручні механізми розробки клієнтської частини платформи.

Витрати на розробку і впровадження програмних засобів ( $K$ ) включають [5]:

$$K = K_1 + K_2, \quad (\text{Б.1})$$

де  $K_1$  – витрати на розробку програмних засобів, грн.;

$K_2$  – витрати на відлагодження і дослідну експлуатацію програми вирішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників;
- витрати на покупні вироби;
- витрати на придбання спецобладнання для проведення експериментальних робіт;
- накладні витрати;
- інші витрати.

Витрати на оплату праці включають заробітну плату всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування програмного засобу. Перелік необхідної програмної документації визначено відповідно до ДСТУ 3008-95 та включає:

- текст програми;
- керівництво користувача, яке включає інструкцію користувача;
- опис програми – відомості про логічну і фізичну модель, відомості щодо функціонування програми;

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

– пояснювальна записка – схема алгоритму, загальний опис алгоритму або функціонування програми, а також обґрунтування прийнятих технічних і технічно-економічних рішень.

Перш за все визначаємо стадії розробки веб-платформи. У таблиці Б.1 відображено інформацію щодо етапів технологічного процесу розробки проекту.

Таблиця Б.1 – Стадії розробки програмного засобу

№ п/п	Назва операції (стадії)	Середній час виконання операції, год.
1	Підготовка, складання ТЗ	3
2	Розробка прототипу системи	2
3	Створення алгоритму розробки системи	3
4	Реалізація онлайн платформи	140
5	Тестування веб-платформи	10
6	Попереднє представлення результатів	4
7	Представлення реалізованої системи	1
Всього		163

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{ОП} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (Б.2)$$

де  $n_{ij}$  – чисельність розробників  $i$ -ої спеціальності  $j$ -го тарифного розряду, осіб;

$t_{ij}$  – затрачений час на розробку проекту співробітником  $i$ -ої спеціальності  $j$ -го тарифного розряду, год.;

$C_{ij}$  – годинна ставка працівника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

За умов, якщо середньогодинну ставку розробника не відомо, її можна розрахувати за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (\text{Б.3})$$

де  $C_{ij}^0$  – основна місячна заробітна плата розробника  $i$ -ої спеціальності  $j$ -го тарифного розряду, грн.;

$h$  – коефіцієнт, що визначає розмір додаткової заробітної плати;

$PЧ_i$  – місячний фонд робочого часу працівника  $i$ -ої спеціальності  $j$ -го розряду, год.

Оскільки виконавцем кваліфікаційної роботи є студент, то він виконує всі етапи роботи, тому виступає у ролі, як розробника, так і тестувальника. Таким чином, оплата його праці – це стипендія без додаткових нарахувань, надбавок та премій. Стипендія студента становить 2000 грн. Зважаючи на це, вартість проєкту включає стипендію студента, а також витрати керівника на керівництво розробкою проєкту та консультанта із написання техніко-економічного розділу. У таблицю Б.2 записуємо витрати на розробку онлайн платформи.

Таблиця Б.2 – Розрахунок витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год (за весь рік)	Витрати на розробку, грн.
1	Керівник КР, доцент	8	800	6400
2	Консультант з техніко-економічного розділу, доцент	1	800	800
3	Студент	150	12	1800
Разом				9000

У таблиці Б.3 наведено витрати на матеріали та комплектуючі вироби при виконанні кваліфікаційної роботи.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		



Загальна сума витрат на матеріальні ресурси  $B_M$  визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (\text{Б.4})$$

де  $K_i$  - витрата  $i$ -го типу матеріалу, натуральні одиниці вимірювання;

$C_i$  - ціна за одиницю  $i$ -го типу матеріалу, грн.;

$i$  - тип матеріального ресурсу;

$n$  - кількість типів матеріальних ресурсів.

Таблиця Б.3 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп	180,0	1	180,00	18,0	220,0
2	Ручка кулькова	шт	10,0	1	10,00	1,0	11,0
3	Олівець простий	шт	8	1	8,00	0,8	8,8
4	Зошит, 24 арк.	шт	10	1	10	1	11
5	Тонер для принтера	уп	100	1	100	10,0	110
Разом							360,8

Накладні витрати включають витрати на управління, утримання приміщень та невиробничі витрати. Вони розраховуються за встановленими відсотками від витрат на оплату праці. При цьому накладні витрати складають 70% від заробітної плати:

$$H = 9000 \cdot 0,7 = 6300 \text{ грн.}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

становлять 10% від заробітної плати:

$$I = 9000 \cdot 0,1 = 900 \text{ грн.}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{оп} + B_M + I. \quad (\text{Б.5})$$

$$K_1 = 9000 + 360,80 + 900 = 10261 \text{ грн.}$$

До амортизації основних фондів включається сума амортизаційних відрахувань від вартості обладнання і приладів, що використовуються при розробці онлайн платформи. Амортизаційні відрахування розраховуємо за формулою:

$$B_{AM} = \sum_{i=1}^n \frac{B_i \cdot H_i \cdot T_i}{100 \cdot T_{E\Phi i}}, \quad (\text{Б.6})$$

де  $B_i$  – вартість  $i$ -го устаткування, грн.;

$H_i$  – річна норма амортизації  $i$ -го устаткування, %;

$T_i$  – час роботи  $i$ -го устаткування за весь період розробки, год.;

$T_{E\Phi i}$  – ефективний фонд часу роботи  $i$ -го устаткування за рік, год / рік;

$i$  – тип устаткування;

$n$  – кількість устаткування.

Усі розрахунки, що стосуються амортизаційних відрахувань, заносимо у таблицю Б.4.

На основі отриманих даних в результаті обчислень складемо кошторис витрат на розробку онлайн платформи і заносимо їх у таблицю Б.5.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця Б.4 – Амортизація основних фондів

Найменування устаткування	Вартість устаткування, грн	Річна норма амортизації,%	Ефективний фонд часу роботи обладнання, год / рік	Час роботи обладнання для розробки системи, год	Сума, грн.
Комп'ютер	20000	60	1700	163	1151
Разом амортизаційні відрахування					1151

Таблиця Б.5 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	9000
2	Витрати на куповані вироби	361
3	Амортизаційні відрахування	1151
4	Інші витрати	900
5	Витрати на відлагодження і дослідну експлуатацію програмного продукту	150
Разом		11562

Розрахувавши капітальні вкладення у розробку проекту (собівартість), перейдемо до обчислення можливої ціни.

Б. 2 Ціна програмного продукту – це витрати на його придбання і експлуатацію за весь період служби:

$$C_{II} = K \cdot \left(1 + \frac{P_p}{100}\right) + K_0 + K_k, \quad (\text{Б.7})$$

де  $K$  – кошторисна вартість;

$P_p$  – рентабельність;

$K_0$  – витрати на встановлення та освоєння веб-ресурсу, грн.;

$K_k$  – витрати на доукомплектування технічних засобів на об'єкті, грн.

Зважаючи на вищеописане, розрахуємо ціну програмного засобу

									Арк.
									55
Змн.	Арк.	№ докум.	Підпис	Дата					

$$Ц_{\Pi_a} = 101710 \cdot (1 + 0,3) = 134233 \text{ грн.}$$

$$Ц_{\Pi} = 11562 \cdot (1 + 0,3) = 15031 \text{ грн.}$$

У наступному підрозділі проведемо аналіз економічної ефективності розробки онлайн-платформи.

### Б. 3 Визначення економічної ефективності розробки проєкту

Для того, щоб побудувати таблицю показників економічної ефективності розробки програмного продукту, проведемо розрахунки необхідних показників. Економічний ефект в сфері розробки проєктного рішення розраховуємо за формулою:

$$E_{\text{ПР}} = Ц_{\Pi} - Ц_A, \quad (\text{Б.8})$$

$$E_{\text{ПР}} = 134223 - 15031 = 119192 \text{ грн.}$$

Дохід від розробки програмного продукту у  $i$ -му періоді розраховуємо за формулою:

$$D_i = J_i (B_i - C_i), \quad (\text{Б.9})$$

де  $B_i$  – ціна продажу програмного продукту в  $i$ -му періоді;

$C_i$  – собівартість розробки (фактично дорівнює сумі витрат на розробку);

$J_i$  – кількість.

$$D_i = 1 \cdot (15031 - 9000) = 6031 \text{ грн.}$$

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Економічний ефект полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів та розраховується за формулою:

$$E = \frac{D_i}{B_{заг}} \quad (Б.10)$$

$$E = \frac{6031}{9000} = 0,7.$$

Тоді термін окупності обчислюємо за такою формулою:

$$T = \frac{1}{E} \quad (Б.11)$$

$$T = \frac{1}{0,7} = 1,4 \text{ р.}$$

Обчислені значення показників економічної ефективності занесемо в таблицю Б.6.

Таблиця Б.6 – Показники економічної ефективності

№	Найменування	Значення показників	
		Аналог	Новий варіант
1	Капітальні вкладення	70425	9000
2	Ціна придбання	134233	15031
3	Економічний ефект	-	119192
5	Дохід від розробки	-	6031
7	Термін окупності проекту	1,4	

Отже, у цьому розділі проведено розрахунок витрат на розробку онлайн платформи для потокового відео. Показники, що характеризують витрати на розробку веб-ресурсу порівняно із показниками, які характеризують онлайн ресурс із аналогічним функціональним призначенням.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Розроблена програмна система має переваги у порівнянні із аналогами, зокрема, простота використання, адаптивність, масштабованість, гнучкість.

Згідно із проведеними розрахунками, що обґрунтовують економічну ефективність розробки веб-ресурсу, можна зробити висновок, що розроблена оеланій система є суттєво дешевшою, оскільки у ролі розробника виступає студент. Прогнозований економічний ефект отримано у розмірі 119192 грн., що свідчить про доцільність розробки онлайн платформи для стрімінгу відео із застосуванням технології React.

					КР.КІ. 9043911.00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		