

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
кваліфікаційної роботи ступеня  
*бакалавра*

(назва освітньо-кваліфікаційного рівня)

студента Юрової Анастасії Кирилівни  
(ПІБ)

академічної групи 122-21зск-1  
(шифр)

спеціальності 122 Комп'ютерні науки  
(код і назва спеціальності)

освітньої програми Комп'ютерні науки  
(назва освітньої програми)

на тему: Розробка стрімінгової веб-платформи для перегляду  
японської анімації.

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Алексєєв М.О.</i>			
<b>розділів:</b>				
спеціальний	<i>проф. Алексєєв М.О.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
<b>Рецензент</b>				
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2024

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем  
(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

«    »

2024 року

**ЗАВДАННЯ  
на кваліфікаційну роботу**

*бакалавра*

(назва освітньо-кваліфікаційного рівня)

студента 122-21зск-1

Юрової А.К.

(група)

(прізвище та ініціали)

тема кваліфікаційної роботи  
*перегляду японської анімації.*

Розробка стрімінгової веб-платформи для

затверджена наказом ректора НТУ «ДП» від 26.04.2024

№ 366-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2024 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки.</i>	<i>27.05.2024 р.</i>

Завдання видав

(підпис)

проф. Алексєєв М.О.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Юрова А.К.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2024 р.

Термін подання кваліфікаційної роботи до ЕК: 08.06.2024 р

## РЕФЕРАТ

Пояснювальна записка: 92с., 22рис., 1табл., 3дод., 21 джерело.

Об'єкт розробки: Стрімінгова веб-платформа для перегляду японської анімації.

Мета кваліфікаційної роботи: розробка веб-платформи для зручного пошуку та перегляду японської анімації.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформи для розробки, виконано проектування і розробка програми, описана робота програми, алгоритм і структура її функціонування, а також виклик та завантаження програми, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення даного проекту полягає в розробці стрімінгової веб-платформи з зручним та зрозумілим інтерфейсом для перегляду японської анімації.

Актуальність даного програмного продукту визначається великим попитом на стрімінгові веб-сервіси, що спрощує процес пошуку та перегляду японської анімації та економить час на похід в кінотеатр.

Список ключових слів: СТРІМІНГОВА ВЕБ-ПЛАТФОРМА, ЯПОНСЬКА АНІМАЦІЯ, ІНТЕРФЕЙС, ВЕБЛУМ, ФІГМА, ПЕРЕГЛЯД, КОНСТРУКТОР САЙТІВ.

## ABSTRACT

Explanatory note: 92 pages, 22 figures, 1 table, 3 appendices, 21 sources.

Object of the Development: The streaming web platform for viewing Japanese animation.

Purpose of the work: development of a web platform for convenient search and viewing of Japanese animation.

The introduction analyzes the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides justification for the relevance of the topic, and clarifies the task setting.

In the first chapter, the subject area is analyzed, the relevance of the task and the purpose of the development are determined, the task setting is formulated, requirements for software implementation, technologies, and software tools are indicated.

In the second chapter, existing solutions are analyzed, platforms for development are selected, the program is designed and developed, the program's operation, algorithm, and structure of its functioning are described, as well as the call and loading of the program, input and output data are determined, and the composition of technical means parameters is characterized.

In the economic section, the complexity of developing the information system is determined, the cost calculation of the work to create the program is carried out, and the time for its creation is calculated.

The practical significance of this project lies in the development of a streaming web platform with a convenient and understandable interface for viewing Japanese animation.

The relevance of this software product is determined by the high demand for streaming web services, which simplifies the process of searching and viewing Japanese animation and saves time on going to the cinema.

List of keywords: WEB PLATFORM, JAPANESE ANIMATION, COMPUTER, INTERFACE, DEVELOPMENT, WEB, STREAMING WEB PLATFORM, DESIGN, VIEWING.

## ЗМІСТ

РЕФЕРАТ .....	3
ABSTRACT .....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 Загальні відомості з предметної галузі .....	10
1.2 Призначення розробки та галузь її застосування .....	11
1.3 Підстави для розробки.....	12
1.4 Постановка завдання.....	12
1.4.1 Ключові аспекти, актуальність платформи та можливості .....	14
1.4.2 Опис інтерфейсу користувача.....	15
1.5 Вимоги до програми або програмного виробу .....	16
1.5.1 Вимоги до функціональних характеристик.....	17
1.5.2 Вимоги до інформаційної безпеки .....	17
1.5.3 Вимоги до складу та параметрів технічних засобів .....	18
1.5.4 Вимоги до інформаційної та програмної сумісності.....	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	20
2.1 Функціональне призначення програми.....	20
2.2 Опис застосованих математичних методів.....	21
2.3 Опис застосованих технологій та мов програмування.....	21
2.4 Опис структури системи та алгоритмів її функціонування.....	36
2.5. Обґрунтування та організація вхідних та вихідних даних програми .....	40
2.6. Опис роботи розробленої системи .....	41
2.6.1. Використані технічні засоби .....	41
2.6.2 Використані програмні засоби.....	41
2.6.3. Виклик та завантаження програми.....	41
2.6.4. Опис інтерфейсу користувача.....	41

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ .....	49
3.1 Розрахунок трудомісткості та вартості розробки програмного продукту ..	49
3.2 Розрахунок витрат на створення програми.....	53
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
Додаток А. Код Програми. ....	58
Додаток Б. Відгук керівника економічного роздіду .....	91
Додаток В. Перелік документів на оптичному носії .....	92

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

HTML – мова розмітки веб-сторінок;

CSS – мова стилів, яка використовується для визначення вигляду та форматування веб-сторінок;

ПЗ – програмне забезпечення;

Аніме – японська анімація;

JavaScript – високорівнева мова програмування, яка переважно використовується для веб-розробки;

React – відкрита бібліотека JavaScript для розробки інтерфейсів користувача;

UI – графічний інтерфейс користувача;

Онгоінг – аніме, останній сезон якого ще виходить і транслюється по телебаченню в Японії.

## ВСТУП

Перегляд японської анімації, відомої як аніме, завжди був одним із улюблених способів проведення вільного часу, особливо серед підлітків, які все більше віддають перевагу іноземній культурі. Це призвело до зростання популярності та затребуваності аніме. У сучасному світі, де культура стає все більш глобалізованою, а доступ до різноманітних форм розваги стає універсальним, створення онлайн-платформи для перегляду японської анімації є актуальним та важливим завданням.

Наразі існує багато схожих платформ, які вже запуснені в Інтернеті та користуються попитом. Вони містять інформацію про аніме, режисерів, студії та статус виходу серій. Проте далеко не всі з них пропонують зручний пошук, перегляд улюбленого аніме, переклад українською мовою та можливість перегляду в оригінальній озвучці. Тому веб-платформа, що створюється, дозволить заощадити час на пошук нових анімацій, вибрати улюблену студію озвучки з перекладом на декілька популярних мов світу, а також у будь-який час отримати необхідну інформацію про аніме та додати його до «обраного» списку для перегляду пізніше або разом з іншими користувачами через особливу функцію платформи — спільний перегляд.

Для підвищення продуктивності щороку на інтернет-платформах з'являється все більше додаткових ресурсів, які прискорюють роботу та роблять сайти більш привабливими. Користувачі все більше звертають увагу не лише на функціональність сайтів, але й на їхній зовнішній вигляд. Це особливо важливо враховувати під час розробки стрімінгової веб-платформи, щоб залучити якомога більше клієнтів.

З огляду на це, слід використовувати найсучасніші та найпотужніші інструменти розробки, щоб створити сучасний, багатофункціональний та якісний продукт.



Дизайн веб-платформи відіграє ключову роль для вашого бізнесу. Це інструмент, за допомогою якого ви презентуєте свій бренд і завойовуєте довіру потенційних клієнтів. Якісний веб-дизайн має вирішальне значення у взаємодії з клієнтами та впливає на рівень їхньої конверсії. Важливо, щоб платформа була візуально приваблива та зручна для користувачів.

Дана кваліфікаційна робота присвячена створенню стрімінгової веб-платформи для комфортного та якісного перегляду японської анімації. Проект включає різноманітні аспекти, від проектування та створення дизайну користувацького інтерфейсу до реалізації функціоналу та розробки адаптивності для зручного перегляду на мобільних пристроях.

Головна мета проекту — розробка сучасної та інтуїтивно зрозумілої веб-платформи, яка відповідатиме потребам як новачків у світі японської анімації, так і досвідчених фанатів. Під час розробки враховуються не тільки технічні аспекти, але й бажання та потреби користувачів, щоб забезпечити комфортне та приємне користування платформою.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ЇЇ СТВОРЕННЯ

### 1.1. Загальні відомості з предметної галузі

Веб-платформа для перегляду японської анімації — це електронна платформа, яка дає можливість цільовій аудиторії швидко знайти та переглянути улюблене аніме різних жанрів через Інтернет.

Історія виникнення веб-платформи починається з росту популярності аніме у всьому світі. Японська анімація завоювала прихильність мільйонів глядачів завдяки унікальному стилю, різноманіттю сюжетів та високій якості анімації. У зв'язку з цим виникла потреба у створенні зручних платформ для перегляду аніме онлайн.

Перші сервіси для перегляду аніме з'явилися ще в 1990-х роках, коли Інтернет став більш доступним для широкої аудиторії. Тоді основним способом поширення аніме були піратські веб-сайти та файлообмінні мережі, де фанати могли знайти і завантажити улюблені серіали. Однак ці методи були нелегальними і часто супроводжувалися низькою якістю відео та відсутністю офіційного перекладу [18].

Ситуація почала змінюватися з появою легальних стрімінгових платформ, таких як Crunchyroll (2006) та Funimation (1994, з новим фокусом на стрімінг у 2009). Ці платформи запропонували легальний і зручний спосіб перегляду аніме з високою якістю відео, офіційними субтитрами та дубляжем. Вони уклали угоди з японськими студіями, що дозволило глядачам насолоджуватися новими серіями майже одночасно з їх виходом в Японії [18].

На сучасному етапі платформи для перегляду аніме продовжують розвиватися. З'являються нові сервіси, такі як Netflix, який активно інвестує в аніме-контент, та Amazon Prime Video, що також пропонує широкий вибір японської анімації. Водночас існуючі платформи вдосконалюють свої сервіси,

пропонуючи більше можливостей для персоналізації перегляду, соціальної взаємодії та кросплатформеного доступу.

Приклад успішно структурованої веб-платформи можна побачити на ресурсах багатьох стрімінгових сайтів, а також на сайтах для перегляду фільмів та анімацій. Кожен з цих сайтів має дуже просту систему навігації, незважаючи на велику кількість різноманітного відеоконтенту, який пропонується.

Цей простий підхід має свою логіку. Коли ви шукаєте японську анімацію, вам не потрібно починати з загальної категорії "Аніме", де представлені усі аніме. Натомість, на веб-платформах популярних студій озвучення японських анімацій ви можете просто прокрутити до відповідної основної категорії, після чого варіанти стають все більш конкретними. Таким чином, основна категорія стає своєрідним напівфільтром, що дозволяє швидко і зручно знайти потрібне аніме.

Основною перевагою веб-платформи для перегляду японської анімації є можливість доступного та зручного перегляду аніме без необхідності попереднього завантаження на ПК, а також можливість глибше зануритися в культуру Японії. Користувачі можуть легко переглянути доступні аніме, ознайомитися з анонсами та датами виходу нових серій, а також додати їх у список «обраних» для спільного перегляду.

## **1.2. Призначення розробки та галузь її застосування**

Призначенням розробки є створення стрімінгової веб-платформи для перегляду японської анімації. Коротке найменування для web-сайту – «ANIMATV». Даний сайт призначається для людей, які хочуть відпочити за переглядом аніме на свій вибір. Програмний продукт є безкоштовним, що дозволяє зекономити не лише час, а й гроші.

Основна функціональність платформи включає можливість переглядати сторінки з різноманітними аніме, додавати у список «обраних», збереження та

додавання у колекції вподобаних анімацій. Крім того, користувачі можуть залишати коментарі та оцінки до різних аніме, що сприяє активному спілкуванню між глядачами. Цільовою аудиторією даного ПЗ є студії озвучування, художники, косплеєри, фанати японської анімації, а також звичайні люди які бажають провести час за переглядом улюбленого аніме.

Ця платформа дозволить підвищити популярність японської анімаційної культури, залучити більше студій озвучування аніме та зацікавлених осіб для сумісної взаємодії.

### **1.3. Підстави для розробки**

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- Освітня програма 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 366-с від 26.04.2024 р;

Завдання на кваліфікаційну роботу на тему «Розробка стрімінгової веб-платформи для перегляду японської анімації».

### **1.4. Постановка завдання**

Розробити стрімінгову веб-платформу, що надає можливість користувачам зручно та швидко знаходити та переглядати японську анімацію як в оригіналі так і з українським озвученням. Дизайн платформи буде створений у графічному редакторі Figma та розроблений у онлайн середовищі для створення сайтів

Weblium з використанням технологій HTML, CSS, JS, React та редактору коду Visual Studio Code (рис. 1.1).

Weblium — це SaaS платформа, розумний конструктор сайтів, на якій будь-який користувач може зібрати собі сайт з блоків з можливістю додавання стороннього коду [10].

Ще Weblium працює у хмарі, тому вам ніколи не потрібно буде нічого завантажувати, встановлювати та оновлювати.

Також в інструменті доступний імпорт коду створюваного сайту та написання власного для створення стилів, ефектів анімації, інтеграції інструментів сторонніх розробників та інших налаштувань для вашої роботи [10].

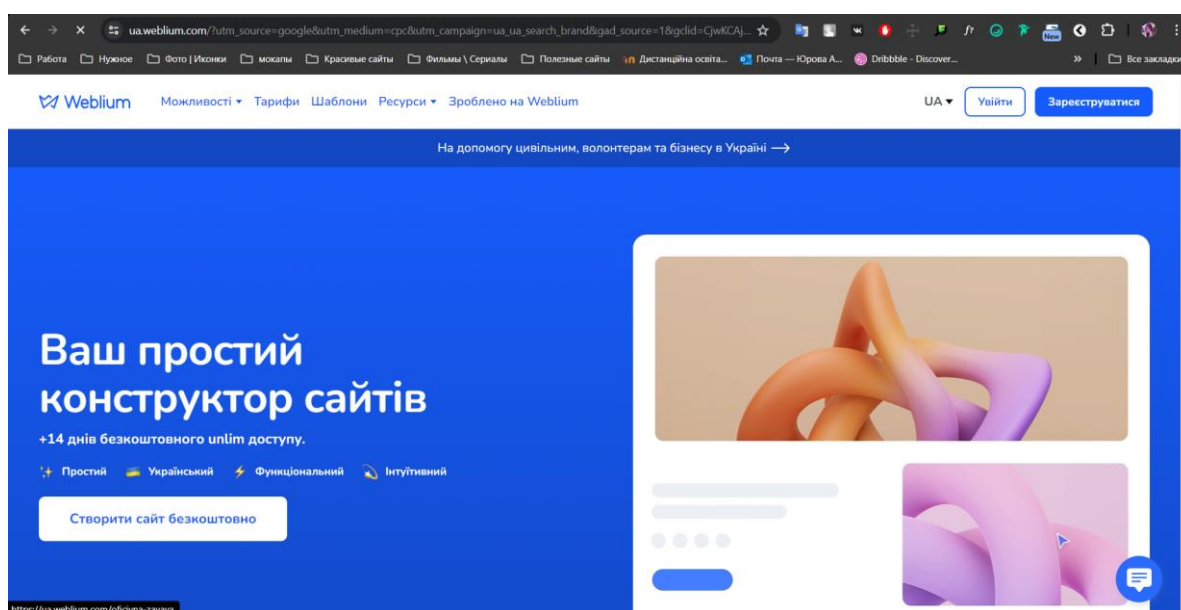


Рис. 1.1. Головна сторінка платформи Weblium

Основними завдання проекту:

1. Розробка розмітки та дизайну інтерфейсу, що дозволить користувачам легко та зручно користуватися платформою.
2. Розробка зручної навігації по сайту для переходу між сторінками.

3. Розробка функціоналу для управління каталогом аніме, включаючи фільтр пошуку та додавання в список «обраних», який буде доступно для зареєстрованих користувачів.

4. Розробка зручного управління особистим кабінетом користувача з списком останніх переглянутих аніме та колекцій зі збереженими анімаціями.

5. Створення функціоналу для спільного перегляду аніме користувачами платформи.

У якості вимог до програмного продукту можуть виступати наступні чинники:

- вивчення предметної галузі завдання;
- проведення аналізу можливостей аналогічних продуктів;
- вибір технології, платформи та архітектури для розробки;

#### **1.4.1. Ключові аспекти, актуальність платформи та можливості**

Веб-платформа для перегляду японської анімації має такі ключові аспекти:

- Каталог з усіма анімаціями: розміщення постерів з назвою аніме та кількістю сезонів та епізодів.
- Сторінка обраного аніме: розміщення повної інформації про обране користувачем аніме, а також відео програвач з можливістю обрати потрібний сезон та епізод.
- Список обраних анімацій: можливість додавати аніме до списку обраних, а також створювати колекції для перегляду пізніше.
- Реєстрація: спеціальна форма, яка дозволяє користувачам створити обліковий запис та користуватися платформою без обмежень.
- Спільний перегляд: можливість переглядати аніме спільно з іншими користувачами платформи після створення облікового запису.

Ці ключові аспекти є важливими для забезпечення зручності та ефективності користування веб-платформою для перегляду японської анімації.

Можливості платформи включають:

- Моніторинг анімацій на сайті, оновлення виходів нових аніме, епізодів та сезонів.
- Реєстрацію користувачів через базу даних.
- Вхід до особистого профілю користувача, що дозволяє додавати аніме до «обраного» списку, створювати колекції, спілкуватися з іншими користувачами на сайті та залишати коментарі під кожною анімацією після перегляду, оцінювати аніме за рейтингом, спільний перегляд анімації, редагувати особисту інформацію.
- Адміністративний доступ: можливість додавання нових японських анімацій на веб-платформу безпосередньо через базу даних, редагування та видалення аніме, епізодів та серій.

Така стрімінгова веб-платформа для перегляду японської анімації має велику можливість залучити широку аудиторію та забезпечити зручність та доступність перегляду аніме. Враховуючи зростання популярності японської культури по всьому світу, така платформа може стати успішним бізнес-рішенням.

#### **1.4.2. Опис інтерфейсу користувача**

Шлях користувача починається з переходу за посиланням до головної сторінки, де увагу привертає великий відео-банер з головним текстом та кнопкою переходу на сторінку з усіма японськими анімаціями.

У меню сайту користувач знаходить такі кнопки:

- "Аніме" – сторінка з усіма японськими анімаціями на платформі;
- "Найкраще" – блок з найкращими аніме за версією користувачів платформи на головній сторінці;

- "Жанри" – розкриваюче меню з усіма жанрами японської анімації;
- "Анонси" — блок з аносованими аніме на головній сторінці платформи;
- "Онгоінги" – блок з аніме, останній сезон яких ще виходить і транслюється по телебаченню в Японії, знаходиться на головній сторінці сайту.
- "Спільний перегляд" – сторінка, яка дає можливість спільного перегляду аніме між користувачами платформи, доступна лише зареєстрованим користувачам;
- "Увійти" – при натисканні на цю кнопку з'являється форма для авторизації в системі, якщо користувач вже має акаунт;
- "Реєстрація" – при натисканні на цю кнопку з'являється форма для реєстрації нового користувача у базі даних. Після успішної реєстрації користувач перенаправляється на головну сторінку для перегляду основного контенту.

Після успішної реєстрації на стрімінговій веб-платформі, користувачу буде доступний такий функціонал:

- Додавати аніме у список «обраних»;
- Коментувати та виставляти рейтинг переглянутого аніме;
- Спілкуватися з іншими користувачами платформи;
- Переглядати відразу з декількома користувачами одразу обрану японську анімацію;
- Створювати особисті колекції з аніме для перегляду пізніше.

### **1.5.1. Вимоги до програми або програмного виробу**

В результаті роботи ми повинні отримати стрімінгову веб-платформу для перегляду японської анімації.



### **1.5.1. Вимоги до функціональних характеристик**

Стрімінгова веб-платформа для перегляду японської анімації повинна відповідати певним вимогам щодо своїх функціональних характеристик, щоб забезпечити максимальний комфорт та зручність для користувачів. Основні функціональні характеристики даної платформи є:

1. Зручний та інтуїтивно зрозумілий інтерфейс: платформа повинна мати зрозумілий у використанні інтерфейс, який дозволяє користувачам легко знаходити необхідні аніме, переглядати його та використовувати додаткові функції на платформі.

2. Інформація про кожне аніме: сайт повинен надавати детальну інформацію про кожне аніме, включаючи назву, опис, рейтинг на сайті, фотографії, трейлер та список та кількість епізодів. Це допомагає користувачам швидко та легко знайти потрібну японську анімацію для перегляду.

3. Спільний перегляд: користувачам повинна бути надана можливість зручного онлайн-перегляду аніме один з одним, проте ця функція буде доступна лише після реєстрації на платформі.

### **1.5.2. Вимоги до інформаційної безпеки**

Стрімінгова веб-платформа для перегляду японської анімації має певні вимоги до інформаційної безпеки. Однією з вимог є реєстрація та авторизація користувачів на платформі для того щоб повноцінно користуватися усім функціоналом платформи. Для цього користувачі повинні створити обліковий запис та використовувати унікальний логін та пароль або зареєструватися через Google сервіс. На сайті існують дві ролі: користувач та адміністратор. Кожна роль має свою відповідну панель керування. Користувачам надається доступ до функцій додавання аніме до списку «обраних», залишати коментарі, спілкуватися

з іншими користувачами на сайті та спільний перегляд. Адміністратор, з свого боку, має розширені права, такі як додавання, редагування та видалення аніме.

Ці моменти забезпечують безпеку інформації та обмежують доступ до функцій тільки авторизованим користувачам.

Отже, забезпечується повноцінний захист усіх важливих даних, а також запобігається недобросовісна поведінка та зловживання з боку користувачів.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для правильного функціонування кінцевого продукту необхідними технічними умовами є:

- операційна система Windows, Linux, MacOS;
- наявність будь-якого сучасного браузера, наприклад, Google Chrome, Opera, Microsoft Edge;
- підтримка високошвидкісного Інтернету;
- обсяг оперативної пам'яті (ОЗУ) не менш 2048 МБ.
- Ці вимоги дадуть оптимальну роботу веб-платформи і забезпечать зручність у користуванні.

### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Користувач має можливість зареєструватись на платформі, знайти, обрати та подивитися обрану японську анімацію.

Розробка сайту проводилась на конструкторі сайтів Weblium з використанням графічного редактору Figma та технологій HTML, CSS, Javascript та React. В якості середовища для редагування коду було обрано Visual Studio Code.

- Веб-платформа повинна бути сумісною з основними браузерами, такими як Chrome, Firefox, Safari та Edge.

- Веб-платформа повинна використовувати безпечні та надійні програмні засоби для забезпечення функціонування веб-застосунку, такі як веб-сервер, база даних, CMS, SSL-сертифікат тощо.
- Веб-платформа повинна регулярно оновлювати своє програмне забезпечення та виправляти помилки та вразливості.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

#### 2.1. Функціональне призначення програми

Головна мета даного проекту є створення стрімінгової веб-платформи для перегляду японської анімації. Стрімінгова платформа працює за принципом передачі контенту від провайдера до користувача. Весь контент вже завантажений на сторонньому сервері, користувачеві не потрібно нічого завантажувати для перегляду або прослуховування. Контент транслюється наживо, а швидкість завантаження напряму залежить від інтернет-з'єднання користувача.

Веб-платформа буде розроблена у середовищі конструктора для розробки сайтів Weblium з використанням технологій HTML, CSS, мови програмування JavaScript та бібліотеки React. Основний функціонал платформи полягає в можливості користувача знайти, обрати та подивитися обране аніме використовуючи систему фільтрації та пошук.

Для кращої і якісної взаємодії на платформі є особистий кабінет користувача, який доступний при реєстрації на сайті, він дає можливість керувати обраними аніме, шляхом створення різноманітних колекцій для перегляду пізніше, відстежувати історію перегляду тощо. У профілі користувача можна знайти таку інформацію як: рейтинг на сайті, улюблені жанри, дата реєстрації на сайті, дата народження, загальна кількість переглянутого контенту в годинах та гендер.

Також наявність облікового запису дає можливість користувач взаємодіяти один з одним, зокрема додавати один одного в «список друзів», обмінюватися повідомленнями на платформі та сумісно переглядати обрану японську анімацію.

## **2.2. Опис застосованих математичних методів**

При створенні стрімінгової веб-платформи математичні методи не використовувалися.

## **2.3. Опис застосованих технологій та мов програмування**

Перед початком створення дизайну стрімінгової веб-платформи для перегляду японської анімації, був проведений аналіз серед прямих та непрямих конкурентів, та аналіз цільової аудиторії.

Аналіз конкурентів – це процес дослідження ваших основних конкурентів у галузі. Ви можете збирати дані про продажі, маркетингові стратегії, сильні та слабкі сторони, а також продукти та послуги, щоб запропонувати щось краще для ваших цільових клієнтів [1].

Після збору достатньої інформації про конкурентів, ви зможете розробляти та впроваджувати кращі бізнес-стратегії. Ви можете зосередитися на поліпшенні послуг або запропонувати те, чого не вистачає конкурентам.

Цей аналіз також допомагає визначити останні маркетингові тенденції в галузі. Ви зможете передбачити майбутні виклики та визначити потенційні можливості, діючи попереду конкурентів [1].

Цільова аудиторія – це саме ті люди, котрим потрібен продукт чи послуга компанії. Ця група людей, об'єднаних загальними ознаками, має схожі звички, спосіб життя, пріоритети та рівень доходів. Саме на них має бути налаштована рекламна компанія підприємства [21].

Незнання цільової аудиторії – це перший крок на шляху до підвищення витрат на рекламу та зниження прибутків від продажу продукту.

Тобто цільова аудиторія — це люди, котрі не просто зацікавлені, а й з великою ймовірністю готові придбати те, що пропонує компанія. Це коло, на яке

має бути орієнтований бізнес. Адже немає ніякого сенсу витратити гроші та час на тих, хто ніколи не зацікавиться пропозицією [21].

Після проведення маркетингового аналізу розроблюваного програмного продукту, була розпочата робота над дизайном у графічному редакторі Figma.

Figma являється онлайн-сервісом для проектування дизайну інтерфейсів (рис. 2.1), за допомогою якого спрощується та прискорюється робота над створенням веб-сайтів та мобільних додатків за допомогою руху спеціальних елементів інтерфейсу - піктограм, кнопок, меню, вікон, форм зворотнього зв'язку, векторної графіки тощо. Користувачі мають можливість співпрацювати над проектами разом в режимі реального часу бо усі файли програми знаходяться в хмарі, і не потрібно завантажувати робочий простір на комп'ютер [5].

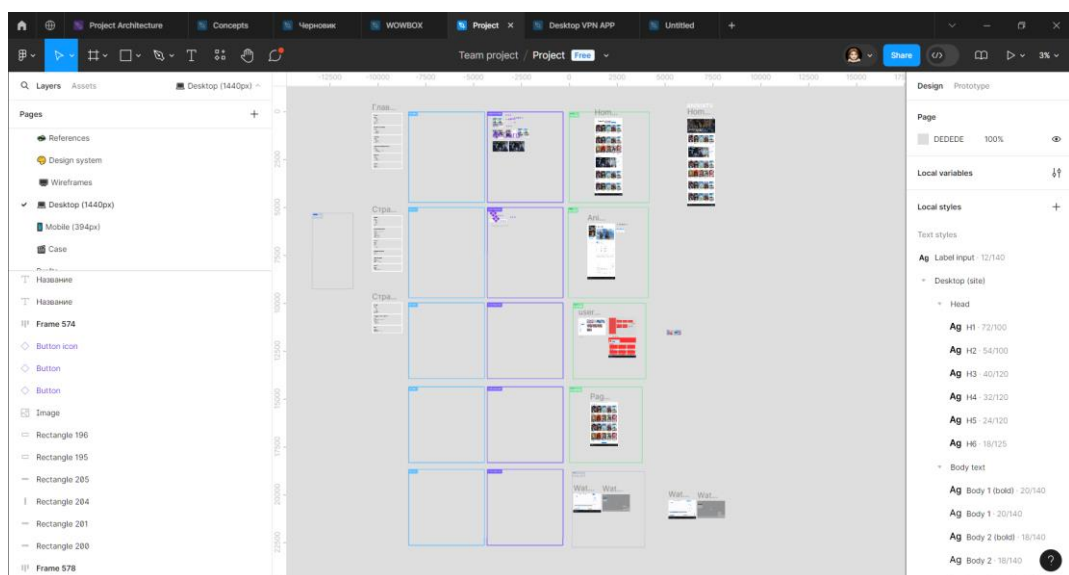


Рис. 2.1. Інтерфейс графічного редактору Figma

Ви можете використовувати онлайн сервіс в браузері або завантажити додаток на свій ПК, ви обираєте, що вам зручніше. Програмний продукт працює на різних платформах, що дає можливість не залежати від одного комп'ютера, як це буває при роботі з іншими графічними редакторами. Комп'ютерна версія дозволяє працювати без використання мережі Інтернет, а при підключенні синхронізувати робочий файл для подальшої роботи [13].

У Figma є достатня кількість переваг, які роблять її популярною та затребуваною на ринку серед користувачів, а найголовніші з них це: зберігання важливої інформації у хмарі, спільна робота над макетами в робочому просторі в режимі реального часу та доступ до них.

Щоб почати роботу, вам потрібно тільки зареєструватися. Якщо вам потрібна спільна робота, треба створити команду, додати проект у створену команду та запросити людей, але у команді може бути не більше ніж три проекта, далі лише за підпискою. Користувач має доступ до базових інструментів Figma, за допомогою яких робиться прототипування та створюється майбутній дизайн інтерфейсів, атрибутів коду, можливостей перегляду та коментування, збереження файлів без обмежень, імпорту документів з Sketch, адаптивного дизайну тощо. Для загальної роботи цього цілком вистачає та не обов'язково купляти підписку. Панель керування у Figma зручна та зрозуміла. Зліва знизу знаходиться список з елементами проекту, зверху - панель інструментів, праворуч доступна панель властивостей.

Переваги створення веб-інтерфейсу у Figma:

- За допомогою інструментів на панелі керування, можна легко та швидко створити потрібний макет.
- Створений макет буде клікабельним, є можливість протестувати кожну деталь, від кнопки, до модального вікна.
- Усі екрани розміщені на одній сторінці для зручної роботи без перемикання.
- Після завершення розробки макету, можна перевірити його вигляд та роботу на готовому мокапі. Мокап – це спеціальний макет, зображення реального предмета, на яке накладається елемент дизайну: логотип, обкладинка, скріншот та інше. Він дозволяє відразу побачити, як об'єкт буде виглядати в реальності [5].

У фігмі також існують створенні користувачем, спеціальні UI елементи, які повторюються в макеті більше ніж два рази, які називаються компонентами.

Компонент можна створити з кнопок, іконок, полів, меню, заголовків, форм та блоків. Є два типи компонентів – основні та залежні. Наприклад, якщо створити кнопку, та зробити з неї компонент, то вона буде основною, а її копія буде залежною. Усі зміни, які робляться в основному компоненті, автоматично застосовуються для залежного. Це дуже зручна функція у Figma, яка значно спрощує роботу користувачів.

Існує велика кількість інструментів у фігмі, ось деякі з них:

- Фрейми - потрібні для створення макетів, шаблонів інтерфейсів, сайтів, бувають різних розмірів, є можливість обрати вже готовий потрібний фрейм.
- Модульна сітка - потрібна для того, щоб рівномірно розмістити кожний елемент у фреймі.
- Векторні форми - для створення різних елементів інтерфейсу;
- Криві - для малювання кривих і простих векторних форм;
- Зображення – для швидкого розміщення графічних елементів у фреймі;
- Ефекти і маски - в тому числі з ефектами накладення, градієнтами і можливістю зручно працювати з декількома шарами;
- Текст - з підтримкою Google Fonts та інструментами підключення додатків.

Figma окрім базових інструментів має ще додаткові плагіни, які можна зберігати та застосовувати в роботі, наприклад:

- Vlobs – плагін для відтворення векторних об'єктів в різноманітних формах;
- Text Styles Creator - автоматизує створення та управління текстовими стилями в Figma для більш гладкого дизайн-процесу;
- Artboard Mockups – плагін з різноманітними мокапами для презентації макетів;



- Unsplash – дозволяє легко імпортувати зображення з однойменного сервісу;

- Noise - генерує ідеальну шумову текстуру для Figma;

А з урахуванням популяризації Figma кількість корисних плагінів буде тільки збільшуватися.

Як і у будь якого програмного продукту, у якого багато переваг є і ряд недоліків. Одним з найбільших недоліків є те, що для роботи в Figma потрібен вихід у мережу Інтернет, у іншому випадку ви просто не матимете доступу до хмарного сховища. Другою проблемою, є те що в Figma інтерфейс англійською мовою і для людей які не знають англійську буде складніше орієнтуватися на платформі. Є ще декілька незначних недоліків: деколи виникають проблеми при роботі з текстом, імпортовані файли можуть некоректно відображатися, ієрархія файлів недостатня і є відсутність деяких потрібних плагінів. Але це дрібниці, які на мою думку можна виправити в наступних оновленнях версії Figma.

Для створення стрімінгової веб-платформи для перегляду японської анімації, я використовувала платформу для створення сайтів Weblium та технології для розробки HTML, CSS та JavaScript.

Конструктор сайтів – це програмне забезпечення для створення сайту в візуальному редакторі без спеціальних знань програмування. Це окрема самостійна послуга, може надаватися в якості додаткової хостингових компаній. В рамках безкоштовної послуги зазвичай надається хостинг з базовими характеристиками та піддомен (наприклад, якщо говорити про конструктор weblium.com, сайт буде розміщений на ім'я `yourname.weblium.com`) [3].

Для того, щоб запустити сайт на одній із CMS потрібно спочатку придбати хостинг та зареєструвати домене ім'я, потім можна запускати процес встановлення та налаштування CMS.

Із використанням конструкторів сайтів – все просто та легко. Необхідно зареєструватися, обрати шаблон, додати потрібні елементи (блоки, модулі) на сайт та наповнити контентом. Процес написання програмного коду беруть на

себе конструктори. Для зміни дизайну створеного сайту у більшості конструкторів потрібно всього декілька кліків мишкою. Використовуючи конструктори сайтів можливо без знань HTML, CSS, PHP створити свій сайт за короткий термін часу; швидко змінити шаблон та макет сайту; із легкістю керувати контентом – додавати текстову складову, завантажувати зображення, відео та ін.; додати свій сайт до списку пошукових систем без особливих знань SEO [4].

Розрізняють онлайн-конструктори та програми-конструктори. Перші працюють повністю в браузері, а дані зберігають на власному сервері. Найчастіше, послуга зберігання платна, і користувачі сильно обмежені тарифними планами послуги. Другі встановлюються на комп'ютер, як програма. Офлайн-конструктори сайтів на кшталт графічного редактора – з тією різницею, що на виході ви отримуєте архів зі сторінками майбутнього веб-сайту. Його можна додати на будь-який хостинг, купити доменне ім'я та опублікувати в інтернеті.

Найбільш розповсюдженими конструкторами сайтів є:

1. Wix – має широкі можливості і зручність сервісу. Дозволяє налаштувати переадресацію, підключити аналітику, поставити індивідуальний фон вікон, можна налаштувати SEO, має велику кількість та високу якість готових шаблонів, розділені за типами сайтів і нішах, Окремо є огляд найбільш популярних, а також нових [4].

2. uCoz – має величезну кількість безкоштовних модулів, хостинг (об'єм якого із часом буде збільшуватися), конструктор, доступ по FTP та ін. багато іншого. Освітні сайти можуть абсолютно безкоштовно зняти рекламний банер [2].

3. A5 – можна створити власний сайт без програмування, має 150 мегабайтів дискового простору на необмежену кількість сторінок. Сайти можна створювати на HTML5 та на flash. Має безкоштовні шаблони, зручний інтерфейс,

велику кількість відео-уроків щодо користування ним, недорогі тарифи при переході на платний варіант [4].

4. Nethouse – застосовується для створення сайтів для малого бізнесу (портфолію, візитка, магазин) без знань програмного коду. Його можна прив'язати до домену та отримати необмежений дисковий простір, зручний інтерфейс, стильні шаблони, можливість реалізації системи електронних платежів, SEO оптимізацію, інтеграцію із соціальними мережами, техпідтримку та багато іншого. Редагування сторінок сайту проводиться в режимі онлайн, можна додавати всілякі блоки: статті; відгуки; коментарі; каталоги товарів; фото і відео матеріали; документацію; форму для зворотного зв'язку; блок статистики та ін. Додані блоки можна рухати в різних напрямках і міняти 284 місцями [4].

5. Weblium — це зручний для початківців конструктор веб-сайтів із редактором з функціоналом drag-and-drop, який допоможе вам створити веб-сайт, що виглядатиме справді професійно. Він спеціально розроблений для колективної роботи, тому, якщо ви оберете платний план, то ви та ваші товариші по команді зможете одночасно працювати над різними частинами того самого сайту.

6. Umi – дозволяє створити сайти, які будуть мати адаптивний дизайн. Наявний модуль автоматичного просування сайту в пошукових системах. Сервіс пропонує необмежену кількість створення сторінок, новин та постів у блогах, можливість прив'язки та покупки домену, створення магазину без обмеження по кількості товарів, синхронізацію каталогу товарів із 1С та сервісом «Мій склад» [4].

7. Jimdo – дозволяє створити сайт безкоштовно, проте – це комерційний продукт і орієнтований в основному на створення інтернет-магазину. Має велику кількість шаблонів, зрозумілий користувацький інтерфейс, можливість створення та завантаження власного шаблону, SEO інструменти, відсутність нав'язливої реклами, дозволяє підключити Google Analytics, вести синхронізацію із Twitter і Dropbox, створювати QR-коди [4].

8. Setup – конструктор сайтів, який містить 5000 варіантів дизайну, зручний інтерфейс, має інтеграцію із сервісом SeoPult. Із даною системою можна автоматизувати ведення всієї роботи із рекламою, створити сайт-візитку, інтернет-магазин та ін. У розпорядженні будуть WYSIWYG-редактор, фотогалерея, каталог, імпорт із Excel, SEO-оптимізація, розкрутка сайту, великий вибір віджетів та багато іншого [4].

9. Readymag – інструмент, що дозволяє створювати сайти та онлайн-презентації у браузері без участі розробників: реалізацію проекту від прототипування до запуску можна повністю доручити дизайнеру. Ще Readymag працює у хмарі, тому вам ніколи не потрібно буде нічого завантажувати, встановлювати та оновлювати [20].

10. Webflow — це конструктор, що допомагає створити вебсайт будь-якої складності. Він виходить за рамки шаблонів та надає вам повний контроль над створенням зовнішнього вигляду та функціоналу сторінок. Незалежно від складності розробки, Webflow має багато можливостей. Сервіс використовується для створення різних типів вебсайтів, лендингів, сайти електронної комерції, блогів і сайтів b2b. Також є Webflow CMS (система керування вмістом), через що сервіс використовується як хостингова платформа [16].

11. Shopify - це платформа, яка дає інструменти для створення та ведення онлайн-бізнесу. Всі інструменти поєднуються в одному місці: інтернет-магазин, складський облік, аналітика, CRM, системи лояльності, багатоканальний маркетинг (продаж через маркетплейси, соцмережі, email, контекстну рекламу та інше) [6].

Проаналізувавши види конструкторів сайтів, можна зробити висновок, що користувачі, які не мають спеціальних знань, зможуть швидко створити свій перший інтернет-магазин, не встановлюючи жодних програм, прямо в вікні браузера. Весь процес зводиться тільки до написання тексту, завантаження своїх зображень, вибору певних компонентів, які надає система і розміщення їх в зазначеному місці.

Для досягнення найбільш вражаючого результату спочатку необхідно визначитися з типом сайту, який потрібен, а потім ознайомитися з кожною приведеною платформою, що підходить під його реалізацію та обрати оптимальне рішення під свої смаки та потреби.

Обрання Weblium для розробки стрімінгової веб-платформи для перегляду японської анімації може бути обґрунтоване рядом переваг, які надає ця платформа:

- Простота використання: Weblium має зручний інтерфейс з конструкторами "drag-and-drop", що дозволяє швидко створювати і редагувати веб-сайти без необхідності в глибоких знаннях програмування.

- Шаблони та дизайн: Платформа пропонує безліч шаблонів та дизайн-рішень, які можуть бути адаптовані під потреби аніме-платформи, забезпечуючи привабливий і професійний вигляд сайту.

- Адаптивність: Сайти, створені на Weblium, автоматично оптимізуються для різних пристроїв і розмірів екранів, що є критично важливим для забезпечення позитивного досвіду користувачів на мобільних пристроях і планшетах.

- SEO та швидкість завантаження: Weblium надає інструменти для оптимізації сайту під пошукові системи (SEO) та забезпечує швидке завантаження сторінок, що допомагає залучати більше трафіку та покращує користувацький досвід.

- Інтеграції та функціональність: Платформа підтримує інтеграцію з різними сервісами та інструментами, такими як системи управління контентом, аналітика, маркетингові інструменти тощо. Це дозволяє створити повнофункціональну платформу для перегляду аніме.

- Підтримка та безпека: Weblium забезпечує високий рівень безпеки для сайтів і надає підтримку користувачам, що може бути важливим для підтримки стабільної роботи платформи та швидкого вирішення можливих проблем.

– Економічність: Використання Weblium може бути більш економічно вигідним порівняно з розробкою сайту "з нуля", оскільки платформа пропонує різні цінові плани, які можуть відповідати бюджетним можливостям проекту.

HTML – це мова гіпертекстової розмітки, яка використовується для створення структури та вигляду веб-сторінок. Перша версія HTML була випущена в 1993 році Тімоті Бернерс-Лі, який вважається батьком Всесвітньої павутини (World Wide Web). Це стало вирішальним кроком у розвитку Інтернету, оскільки HTML дозволяє створювати зв'язані між собою документи з посиланнями [2].

Мова складається з різноманітних тегів, які визначають кожен елемент на веб-сторінці. HTML підтримує міжнародну стандартизацію та розвиток через організацію W3C (World Wide Web Consortium). W3C визначає стандарти і специфікації для HTML, щоб забезпечити єдність у веб-розробці та сумісність між різними браузерами. HTML5 є останньою версією HTML, яка була прийнята в 2014 році [2].

Версія включає багато нових функцій, таких як відео та аудіо плеєри, підтримка графіки без використання додаткових плагінів, покращена підтримка мобільних пристроїв та багато іншого. HTML-код можна переглядати та редагувати за допомогою будьякого текстового редактора. Простота синтаксису HTML робить його доступним для початківців у веб-розробці [2].

Основні та важливі теги HTML:

`<html>`: Вказує, що почався HTML-документ.

`<head>`: Заголовок: визначає місце, де поміщується різна інформація, яка не відображається в тілі документа. Тут розташовується тег назви документа та теги для пошукових машин.

`<body>`: Тіло документа: визначає видиму частину документа.

`<p>`: Визначає абзац тексту.

`<h1>` до `<h6>`: Визначає заголовки різних рівнів, де `h1` – найбільший рівень, а `h6` – найменший.

<a>: Визначає гіперпосилання.

<img>: Визначає зображення на сторінці.

<ul> і <ol>: Визначає нумерований і нумерований списки.

<table>: Визначає таблицю з інформацією.

<form>: Визначає форму HTML для користувацького вводу.

<input>: Визначає елемент управління вводом.

<div>: Визначає розділ в документі [12].

В HTML є можливість формувати текст за допомогою тегів, таких як <strong>, <sub>, <u> тощо.

Крім того, HTML дозволяє вбудовувати зображення, відео, аудіо та інші мультимедійні елементи на сторінку. Для взаємодії з користувачами, HTML дозволяє створювати форми з різноманітними елементами, такими як поля введення тексту, кнопки, чекбокси та списки. Користувачі можуть вводити дані у ці елементи та відправляти їх на сервер для подальшої обробки. Мова гіпертекстової розмітки є фундаментом для розробки і є важливою для будь-якого веб-розробника. Вона допомагає створювати зручні, доступні та організовані веб-сторінки, що робить їх доступними для широкої аудиторії користувачів.

CSS – це спеціальна мова стилів для візуального представлення вмісту HTML-документа. HTML організовує інформацію і надає структуру документу, а CSS оформляє його, прописуючи унікальні властивості кожному з елементів [11].

В процесі розробки створюється CSS-файл, в якому прописується правила оформлення всіх елементів на веб-сторінці. Завдяки цьому код містить менше повторюваних елементів, його простіше читати. Можна поширити одну таблицю стилів на безліч сторінок, істотно знизивши час на верстку. Кожне правило в файлі складається з селектора і блоку оформлення.

Функція селектора – визначити, до якої саме частини документа застосувати конкретне правило. Блок оформлення складається з пар

«властивість: значення». Вони пишуться в фігурних дужках і закінчуються крапкою з комою. При наявності на сторінці CSS, браузер кешує її, завдяки чому знижується час завантаження сайту при наступному вході [11].

Деякі основні CSS властивості:

- align-content: Вирівнювання рядків гумового контейнера по вертикалі;
- align-items: Вирівнювання вмісту рядків гумового контейнера;
- border: Скорочена властивість для border-width, border-style та border-color;
- bottom: Встановлює позицію елементів від нижньої частини батьківського елемента;
- @font-face: Правило, яке дозволяє вебсайтам завантажувати та використовувати шрифти, відмінні від "веб-безпечних" шрифтів;
- flex: Скорочена властивість для властивостей flex-grow, flex-shrink та flex-basis;
- grid-column-start: Визначає, з чого почати елемент сітки;
- justify-content: Визначає вирівнювання між елементами всередині гнучкого контейнера, коли елементи не використовують увесь доступний простір;
- margin: Встановлює всі властивості поля в одній декларації [12].

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [15].

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми



програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Основний синтаксис JS. Синтаксис JavaScript - це набір правил, як будуються програми JavaScript:

```
var x, y;    // Як оголосити змінну;  
x = 5; y = 6; // Як присвоїти значення;  
z = x + y;   // Як обчислити значення.
```

Синтаксис JavaScript визначає два типи значень: фіксовані значення і значення змінних.

Фіксовані значення називаються літералами. Змінні значення називаються змінними [15].

Найбільш важливими правилами для написання фіксованих значень є номери та рядки. Номери пишуться з десятковими знаками або без них, а рядки – це текст, написаний в подвійних або одинарних лапках.

Назва змінної має починатися з літери латиниці або символів "\_" чи "\$", містити лише літери латиниці, цифри і символи "\_" та "\$", а також не може збігатися з жодним з таких зарезервованих слів JavaScript.

Існує три основних типи змінних у JS, а саме:

Var – така змінна була найбільш популярною, так як не було ніякої альтернативи, поки не з'явилася ECMAScript 6. Змінні, оголошені як var доступні лише в межах певної функції. Якщо немає такої функції, то змінні доступні глобально.

Let - це нащадок var у сучасному JavaScript. Сфера його застосування не обмежується лише функціями, let також використовується у блоці виразів. Блок виразу – це все всередині { та }, наприклад, умова if або цикли. Перевага let: це зменшує ймовірність помилок, оскільки змінні доступні тільки в невеликих областях.

Const - технічно константа не є змінною. Особливістю константи є те, що вам потрібно присвоїти їй значення при оголошенні і не буде ніякого способу, аби переприсвоїти це значення. Константна обмежується областями блоків, як let. Константи слід використовувати всякий раз, коли значення не повинно змінюватися в процесі виконання програми, інакше буде повідомлення про помилку при спробі переприсвоїти константі значення [15].

Для підключення JS є два способи:

#### 1. Внутрішнє підключення - підключення в тіло HTML.

При внутрішньому підключенні тег `<script> </script>` використовується без атрибутів.

Раніше в нього вставлялися атрибути `type = "text / javascript»` і `language = "javascript»`. Перший вказував на тип скрипта, а друга мова написання.

В сучасних браузерах ці визначення введені за замовчуванням. Тобто, браузер побачивши тег `script` відразу перемикається в режим `javascript` і виконує код.

У середині тега `script` пишеться код скрипта:

```
<head>
<meta charset="utf-8">
<title>js</title>
</head>
<body>
<p>Початок документа...</p>
<script>
alert ('Вспливаюче вікно');
</script>
<p>...Кінець документа</p>
</body>
</html>
```

#### 2. Зовнішнє підключення - підключення з окремого файлу.

Для того щоб підключити зовнішній файл, в тег `script` вводиться атрибут `src`, який вказує шлях до файлу. Браузер знаходить файл і виконує код.

```
<script src="/js/one.js"></script>
```

```
<script src="/js/two.js"></script>
```

При цьому браузеру досить один раз завантажити і виконати код скрипта, після чого він збереже його в кеші, і при наступних завантаженнях буде брати код звідти, що значно скорочує час завантаження сторінки.

При наявності в тезі `script` атрибута `src`, браузер бачить лише значення атрибута. Якщо після нього в тег помістити будь-якої код, то цей код буде проігнорований [15].

Фреймворк `React.js` — це фреймворк і бібліотека `JavaScript` з відкритим вихідним кодом, розроблені та підтримувані `Facebook` і `Instagram`. Він використовується для швидкого та ефективного створення інтерактивних користувацьких інтерфейсів і вебдодатків із застосуванням значно меншої кількості коду, ніж під час використання звичайного `JavaScript`. У `React` ти розробляєш свої додатки та проекти, створюючи компоненти, що повторно використовуються, які можна розглядати як незалежні блоки `Lego`. Ці компоненти являють собою окремі частини остаточного інтерфейсу, які в зібраному вигляді утворюють увесь призначений для користувача інтерфейс програми [19].

Основна роль `React` у веб-платформі — керування відображенням цього застосунку так само як буква `V` у шаблоні “модель-представлення-контролер” (`MVC`), забезпечуючи оптимальне та найефективніше виконання рендерингу. Замість того щоб розглядати весь призначений для користувача інтерфейс як єдине ціле, `React.js` пропонує розробникам розділити ці складні призначені для користувача інтерфейси на окремі компоненти, які можна використовувати повторно і які утворюють будівельні блоки всього призначеного для користувача інтерфейсу. При цьому платформа `React JS` поєднує в собі швидкість і ефективність `JavaScript` з більш ефективним методом маніпулювання `DOM` для

швидкого рендерингу вебсторінок і створення високодинамічних вебдодатків [19].

Visual Studio Code — це засіб для створення, редагування та відлагодження сучасних веб-додатків і програм. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux та OS X. Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015. Це середовище розробки стало першим кросплатформовим продуктом у лінійці Visual Studio. За основу для Visual Studio Code використовуються напрацювання вільного проєкту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовують браузерний рушій Chromium і Node.js. Visual Studio Code - це редактор вихідного коду. Він підтримує ряд мов програмування, підсвічування синтаксису, рефакторинг, відлагодження, навігацію по коду, підтримку Git та інші можливості. Багато можливостей Visual Studio Code не доступні через графічний інтерфейс, найчастіше вони використовуються через систему команд або JSON файли [17].

Visual Studio також дозволяє замінювати кодову сторінку при збереженні документа, символи перекладу рядка і мову програмування поточного документа. На сьогодні за допомогою вбудованого в продукт призначеного для користувача інтерфейсу можна завантажити і встановити кілька тисяч розширень в категорії «мови програмування». Саме це середовище обрано при розробці кваліфікаційної роботи, оскільки воно надає всю необхідну для розробника функціональність [17].

## **2.4. Опис структури системи та алгоритмів її функціонування**

Коли користувач хоче знайти та подивитися будь який фільм або аніме, він прагне знайти такий сайт або платформу, який буде простим для розуміння та приємним на вигляд. В мережі існують сотні веб-платформ, і жодна людина не

буде терпіти незручності, щоб скористатись саме вашою веб-платформою або сайтом. Скоріше за все, потенційний споживач закриє вкладку і продовжить пошуки «тієї самої» веб-платформи.

Щоб такої ситуації не сталося, потрібно правильно продумати структуру сайту.

Структура сайту — це схема розташування його сторінок, категорій, підкатегорій і товарів. Це своєрідний план, в якому прослідковується логічний зв'язок між сторінками. З технічної точки зору навігація ресурсу являє собою набір URL, що розташовані в певній послідовності. Вона нерозривно пов'язана з семантичним ядром. А саме воно визначає, які папки та документи мають бути на сайті.

Важливо також розрізняти, що структура сайту є зовнішня і внутрішня. Під першою розуміють макет сторінки із зазначенням розташування на ній блоків. Друга відображає категорії, належність до них певних сторінок і матеріалів [8].

Структура веб-платформи наведена на рис. 2.2 та складається з різноманітних файлів та папок, які мають різне функціональне призначення.

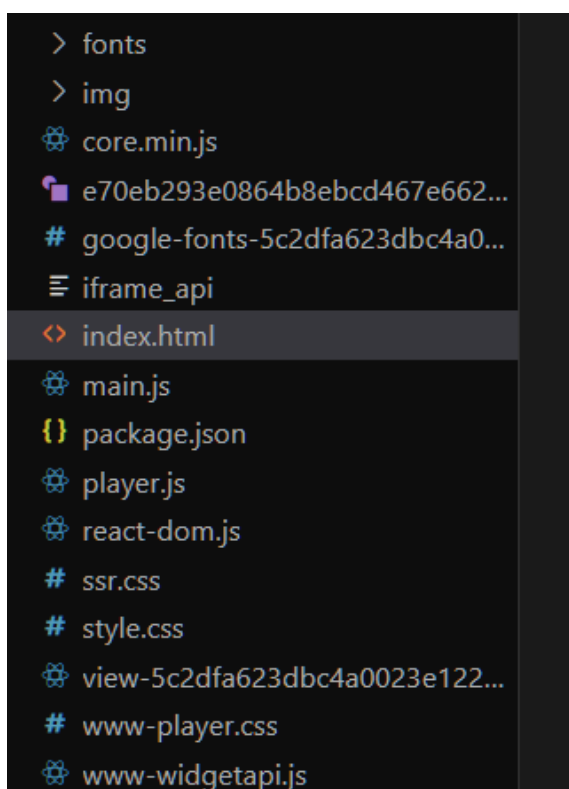
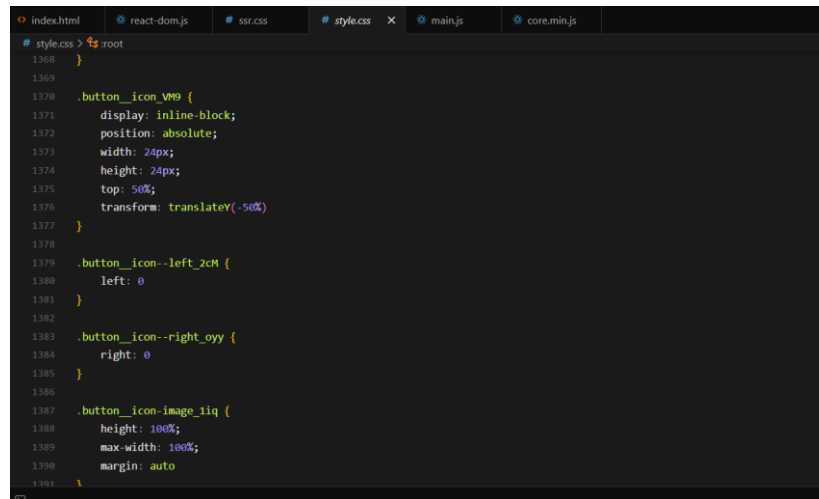




Рис. 2.3. Вміст файлу index.html

Після завантаження, виконується файл style.css, www-player.css, ssr.css для завантаження стилів.

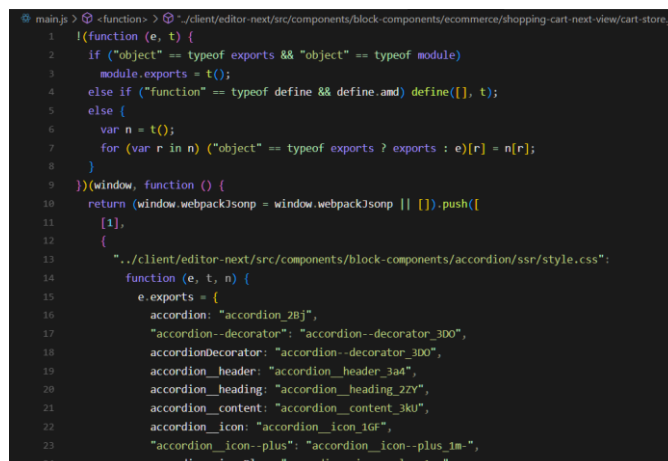
Далі завантажуються JavaScript файли core.min.js, react-dom.js, main.js, player.js. у яких обробляється весь інтерфейс та функціонал стрімінгової веб-платформи для перегляду японської анімації. (рис. 2.4).



```
# style.css > root
1368 }
1369
1370 .button_icon_VW9 {
1371   display: inline-block;
1372   position: absolute;
1373   width: 24px;
1374   height: 24px;
1375   top: 50%;
1376   transform: translateY(-50%)
1377 }
1378
1379 .button_icon--left_2cM {
1380   left: 0
1381 }
1382
1383 .button_icon--right_0yy {
1384   right: 0
1385 }
1386
1387 .button_icon-image_1iq {
1388   height: 100%;
1389   max-width: 100%;
1390   margin: auto
1391 }
```

Рис. 2.4. Вміст файлу style.css

Файл main.js є основним JavaScript файлом, де виконується основна логіка веб-платформи. Цей файл оброблює події користувача, ініціалізує запуск сайту, налаштовує конфігурацію сайту та налаштовує необхідні бібліотеки та модулі, які необхідні для завантаження сайту (рис. 2.5).



```
main.js > <function> > ~/client/editor-next/src/components/block-components/e-commerce/shopping-cart-next-view/cart-store.js
1  !function(e, t) {
2    if ("object" == typeof exports && "object" == typeof module)
3      module.exports = t();
4    else if ("function" == typeof define && define.amd) define([], t);
5    else {
6      var n = t();
7      for (var r in n) ("object" == typeof exports ? exports : e)[r] = n[r];
8    }
9  })(window, function () {
10   return (window.webpackJsonp = window.webpackJsonp || []).push([
11     [1],
12     {
13       "../client/editor-next/src/components/block-components/accordion/ssr/style.css":
14         function(e, t, n) {
15           e.exports = {
16             accordion: "accordion_2Bj",
17             "accordion--decorator": "accordion--decorator_3D0",
18             accordionDecorator: "accordion--decorator_3D0",
19             accordion_header: "accordion_header_3A4",
20             accordion_heading: "accordion_heading_2Zy",
21             accordion_content: "accordion_content_3ku",
22             accordion_icon: "accordion_icon_1Gf",
23             "accordion_icon-plus": "accordion_icon--plus_1m-",
24             accordion_iconplus: "accordion_icon--plus_1m-"
25           }
26         }
27     ]
28   ])
```

Рис. 2.5. Вміст файлу main.js

Усі елементи стрімінгової веб-платформи завантажуються з компонента App.tsx, який є головним файлом . Він також має свої дочірні компоненти, такі як Header, Input, Buttons та інші.

Директорія з різними компонентами має наступну структуру (рис. 2.6):

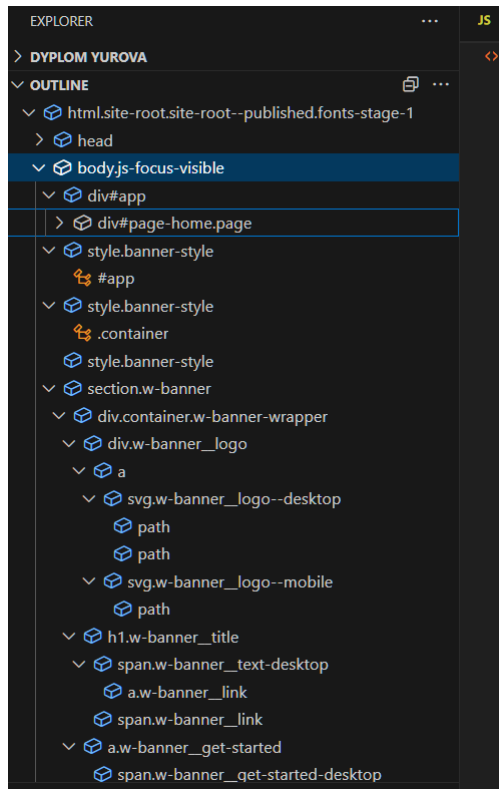


Рис. 2.6. Директорія компонентів веб-платформи

## 2.5. Обґрунтування та організація вхідних та вихідних даних програми

Вхідними даними даної кваліфікаційної роботи є:

- логін, пароль користувача;
- особисті дані користувача на його профілі на платформі;

Вихідними даними є дані про користувачів, які реєструються на веб-платформі та списки з аніме, які зберігаються у базі даних та виводяться у зручному для користувача вигляді при відправленні запиту.



## **2.6. Опис роботи розробленої системи**

### **2.6.1. Використані технічні засоби**

Для проектування веб-платформи знадобились наступні технічні засоби:

- Процесор Intel (R) Core (TM) i5-9300H CPU @ 2.40GHz
- 8 ГБ оперативної пам'яті;
- ОС Windows 10;
- Клавіатура;
- Маніпулятор «миша».

### **2.6.2. Використані програмні засоби**

Стрімінгова веб-платформа була розроблена за допомогою конструктора сайтів Weblium з використанням графічного редактора Figma, мов розмітки HTML, CSS та технологій JavaScript та React. Програмний код було написано у середовищі розробки Visual Studio Code.

### **2.6.3. Виклик та завантаження програми.**

Для відкриття стрімінгової веб-платформи потрібно в браузері вставити посилання: <https://animatv.weblium.site/>

### **2.6.4. Опис інтерфейсу користувача.**

Після того, як користувач перейшов за посиланням до веб-платформи, він одразу переходить на головну сторінку (рис. 2.7), на якій можна побачити

стартовий відео-банер з текстом та кнопкою, для переходу на сторінку з усіма аніме (рис. 2.8).

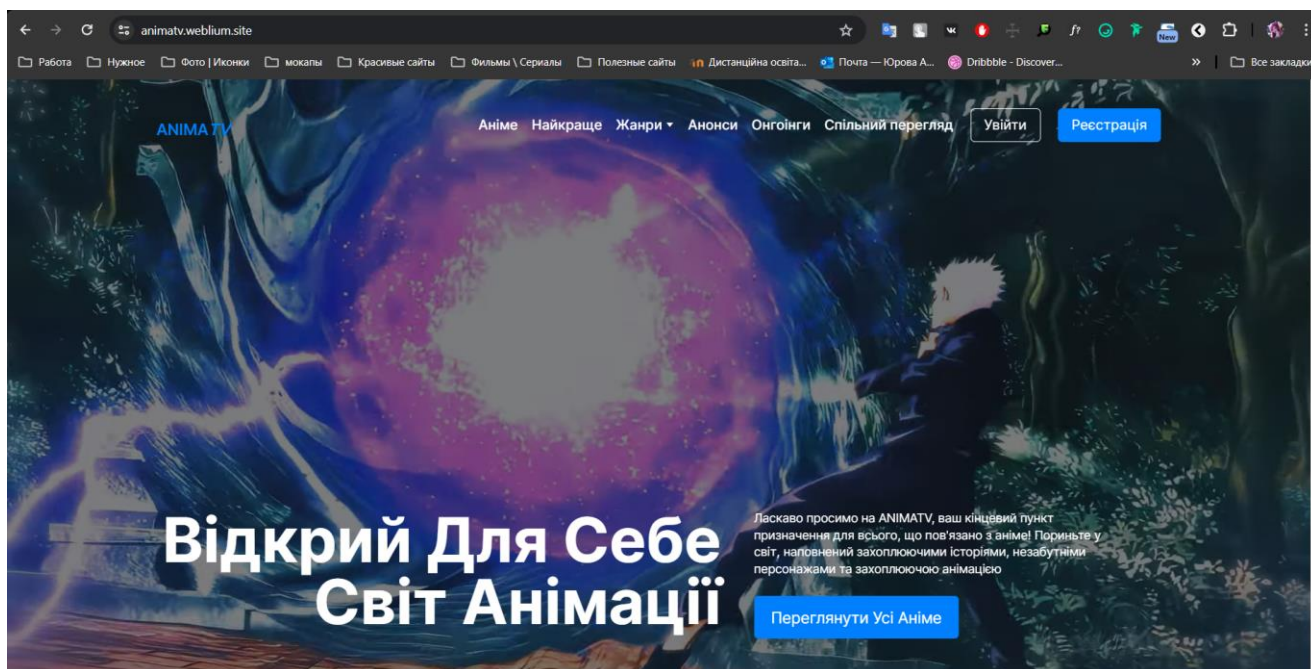


Рис. 2.7. Головна сторінка

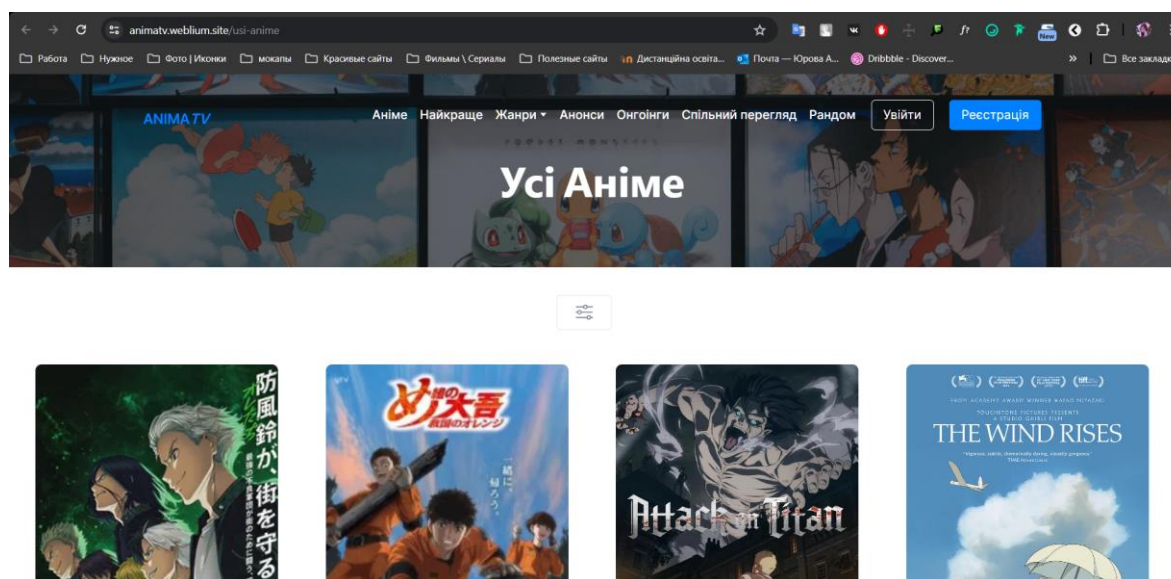
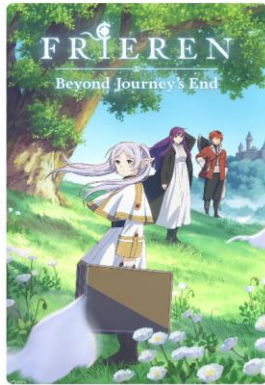


Рис. 2.8. Сторінка з усіма аніме

Якщо користувач буде гортати вниз, він зможе побачити список японських анімацій, які дивляться зараз користувачі платформи (рис. 2.9), нові епізоди, які недавно вийшли на сайті (рис. 2.10), список найкращих аніме платформи за

версією користувачів (рис. 2.11), онгоінги – це аніме-серіали, які зараз виходять на телебаченні (рис. 2.12) та анонси (рис. 2.13). Кожне аніме має власну назву та інформацію про кількість сезонів та епізодів, а також рейтинг на платформі.

## Дивляться зараз



**Фрірен, що проводить в останню путь**

Сезон 1, Епізод 28



**Клинок, який знищує демонів**

Сезон 5, Епізод 10



**Тільки я візьму новий рівень**

Сезон 1, Епізод 10



**Про моє переродження в слиз**

Сезон 3, Епізод 5

Рис. 2.9. Список аніме, які дивляться зараз

## Новинки

[Переглянути Усі >](#)



**Машл: Магія та М'язи**

Епізод 12 | Сьогодні, 18:00

[▶ Дивитися](#)



**Ласкаво просимо до класу Еліти 3**

Епізод 15 | Сьогодні, 12:00

[▶ Дивитися](#)



**Стаючи чарівницею**

Епізод 12 | Сьогодні, 16:00

[▶ Дивитися](#)



**Король шаманів: Квіти**

Епізод 555 | 18.06, 12:00

[▶ Дивитися](#)

Рис. 2.10. Нові епізоди



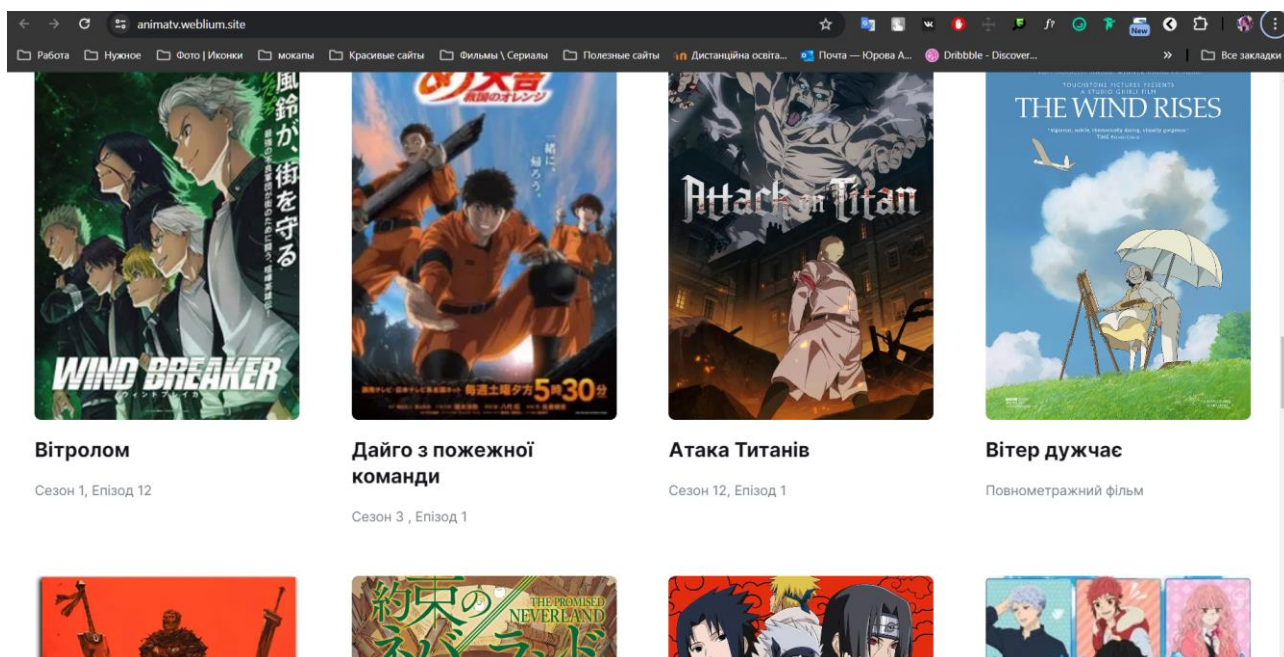


Рис. 2.11. Найкращі аніме

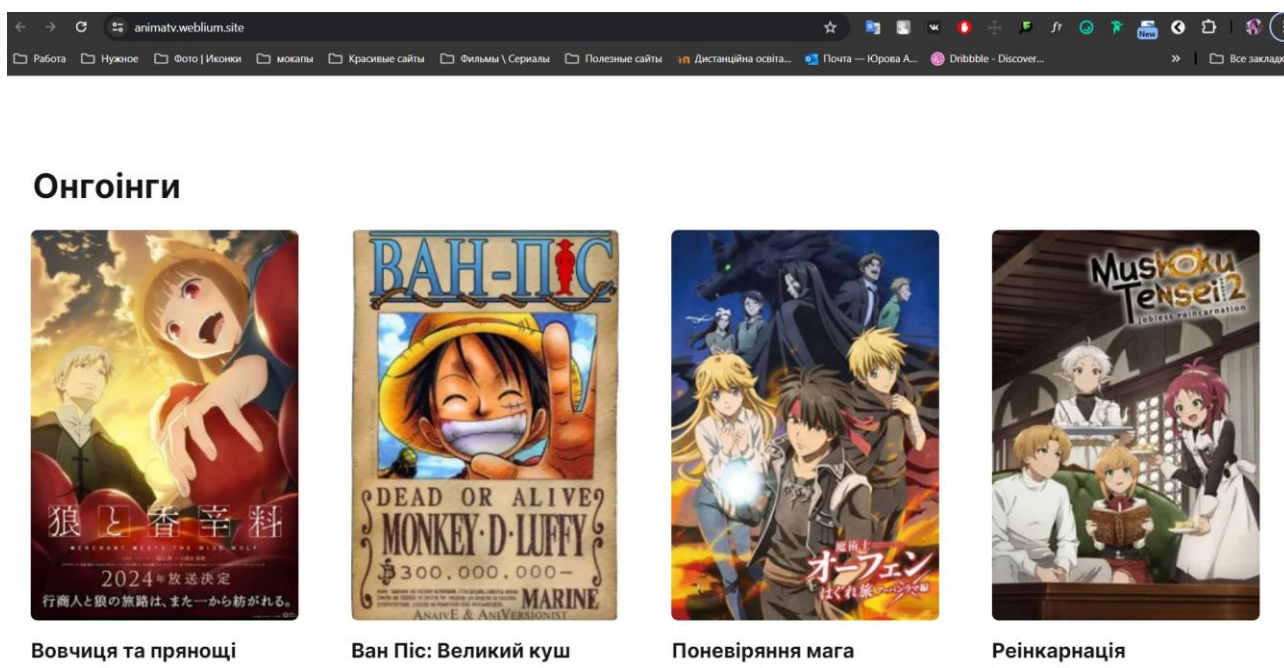


Рис. 2.12. Онгоінги

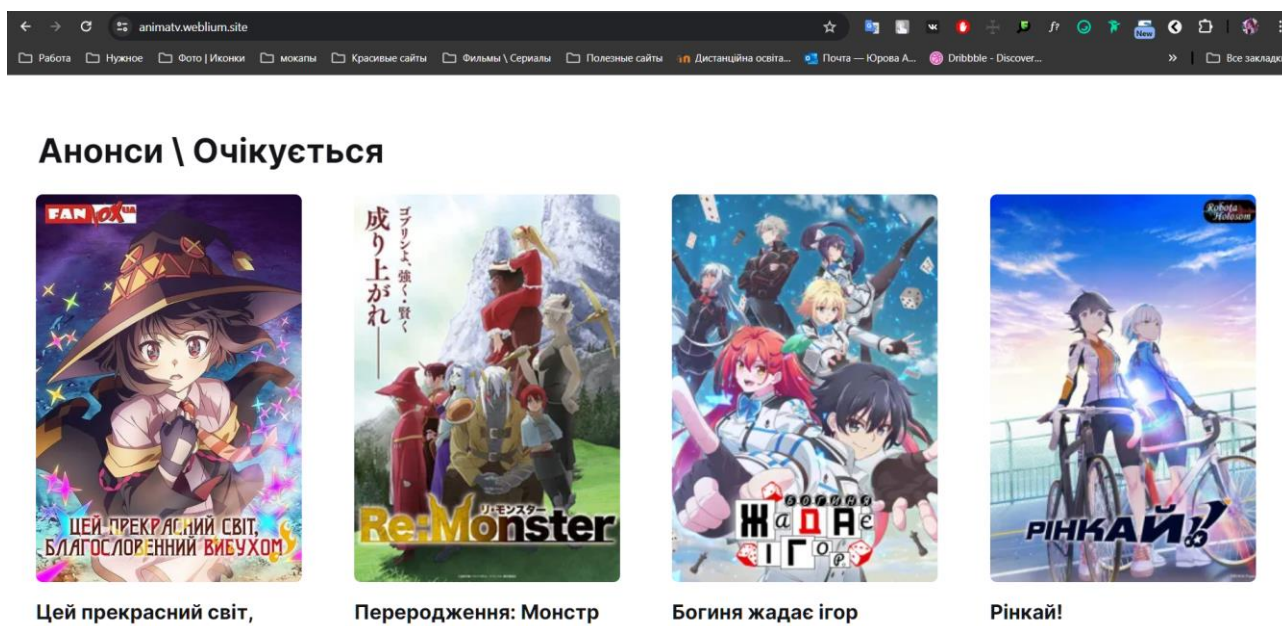


Рис. 2.13. Анонси

Якщо користувач вибере для перегляду будь яке аніме, то зможе перейти на його сторінку з усією інформацією про анімацію: переглянути список сезонів та епізодів , обрати улюблену студію озвучування та переглянути трейлер (рис. 2.14), календар з інформацією про релізи епізодів обраного користувачем аніме (рис. 2.15), дізнатися про що анімація (рис. 2.16) та коментарі від користувачів платформи (рис. 2.17).

Також у користувача буде можливість переглянути обране аніме разом з іншим користувачем (рис. 2.18) після реєстрації на платформі.

Окрім вищезазначених функцій, користувачеві буде доступна реєстрація на платформі для того, щоб отримати особистий кабінет з інформацією про переглянуті аніме та спілкування з користувачами платформи. Макет особистого кабінету користувача представлений на рис. 2.19.



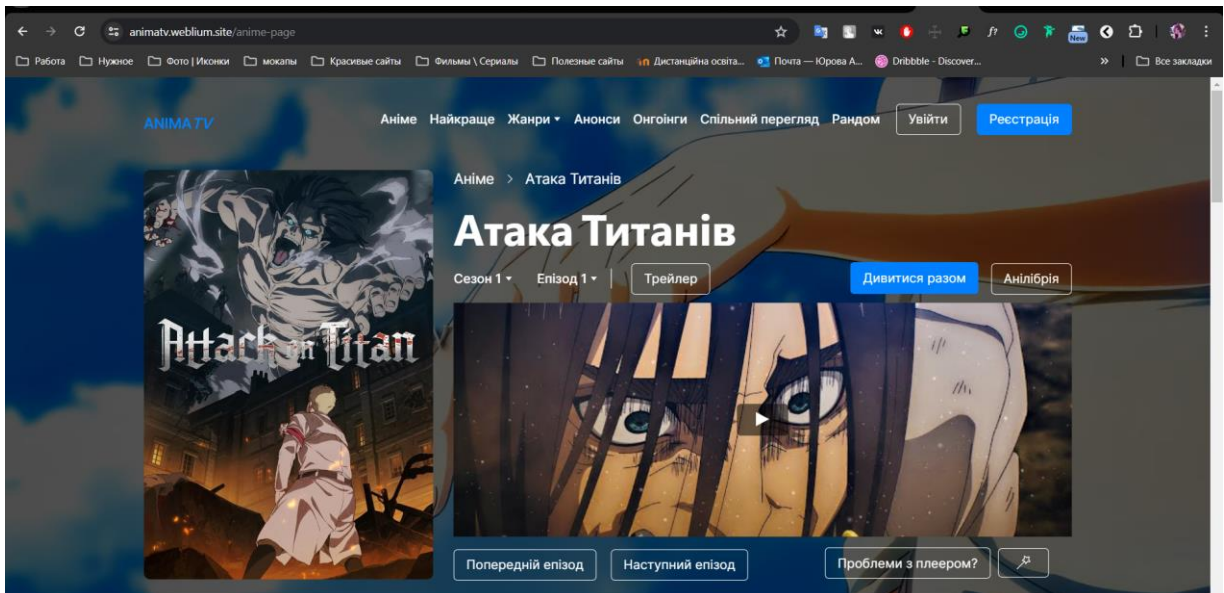


Рис. 2.14. Інформація про обране аніме та програвач з епізодами та сезонами

Сезон 1 Сезон 2 Сезон 3 **Сезон 4**

### Сезон 4 - Усі Епізоди, Дати Релізу

Номер	Епізод	Дата	Вихід
08	Епізод 8	12 Серпня, 2024	Очікується
07	Епізод 7	05 Липня, 2024	Очікується
06	Епізод 6	15 Червня, 2024	Очікується
05	Епізод 5	26 Травня, 2024	Очікується
04	Епізод 4	12 Травня, 2024	<b>Вийшов</b>

Рис. 2.15. Календар з інформацією про релізи епізодів

### Про Аніме

Вже багато років людство бореться з титанами - величезними істотами, які не мають особливого розуму, але поїдають людей і отримують від цього задоволення. Після довгої боротьби залишки людства побудували високу стіну, яка оточила країну людей, через яку титани не змогли пройти. Відтоді минуло сто років, люди мирно живуть під захистом стіни. Але одного разу підлітки Ерен і Мікаса стають свідками жахливої події - ділянку стіни руйнує супертитан, який з'являється з повітря. Титани нападають на місто, і діти з жахом бачать, як один з монстрів живцем з'їдає матір Ерена. Хлопчик присягається, що вб'є всіх титанів і помститься за людство.



Рис. 2.16. Коротка інформація про що аніме

## Коментарі (120)

Останні

Популярні

Коментар



Відправити



Ігор Шевченко



Lorem ipsum dolor sit amet consectetur. Pulvinar erat proin pretium leo a eget fusce morbi nisi. Tincidunt turpis scelerisque dui consectetur. Dignissim nam mauris augue nec. Neque faucibus sed molestie vel et tristique in. Lorem ipsum dolor sit amet consectetur. Pulvinar erat proin pretium leo a eget fusce morbi nisi. Tincidunt turpis scelerisque dui consectetur. Dignissim nam mauris augue nec. Neque faucibus sed molestie vel et tristique in.



4



1

Відповісти

4 години назад



Ігор Шевченко



Lorem ipsum dolor sit amet consectetur. Pulvinar erat proin pretium leo a eget fusce morbi nisi. Tincidunt turpis scelerisque dui consectetur. Dignissim nam mauris augue nec. Neque faucibus sed molestie vel et tristique in. Lorem ipsum dolor sit amet consectetur. Pulvinar erat proin pretium leo a eget fusce morbi nisi. Tincidunt turpis scelerisque dui consectetur. Dignissim nam mauris augue nec. Neque faucibus sed molestie vel et tristique in.



4



1

Відповісти

6 годин назад

Рис. 2.17. Коментар від користувача платформи

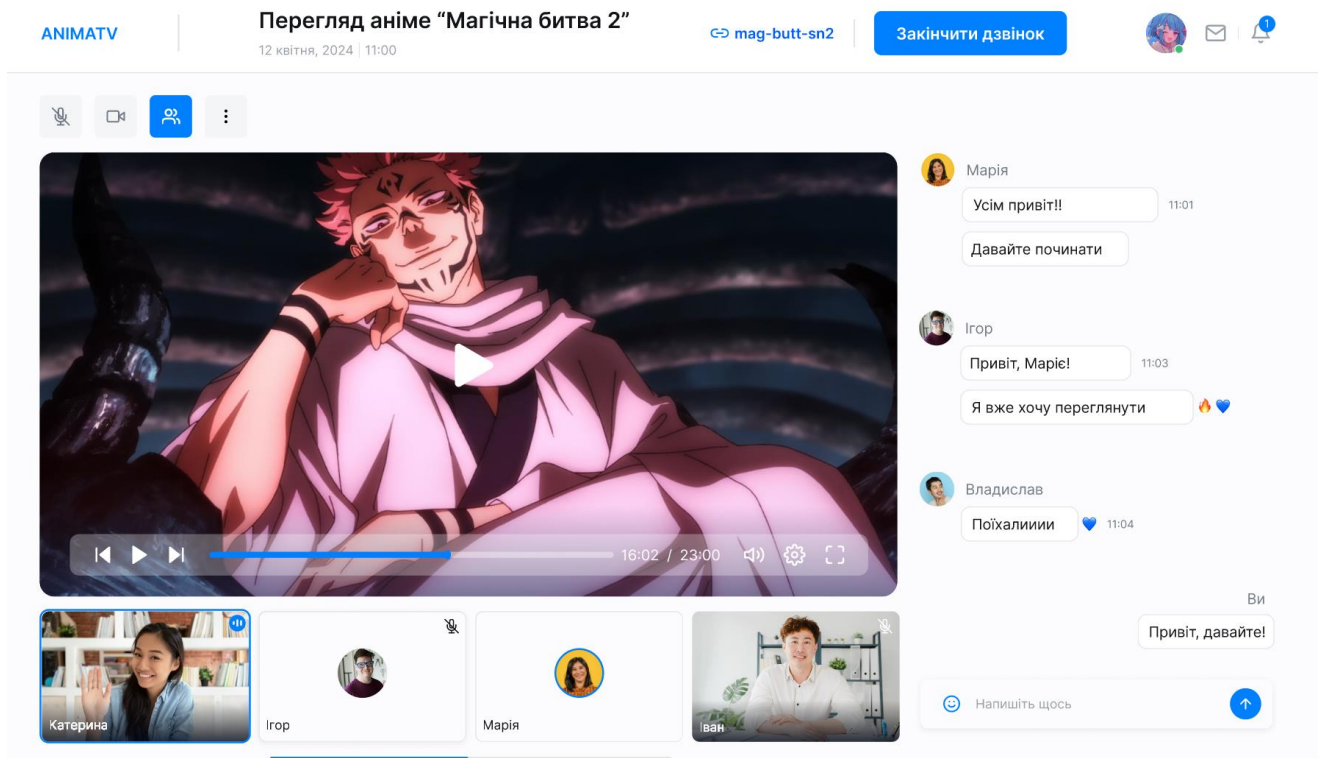


Рис. 2.18. Макет сторінки «Спільний перегляд»

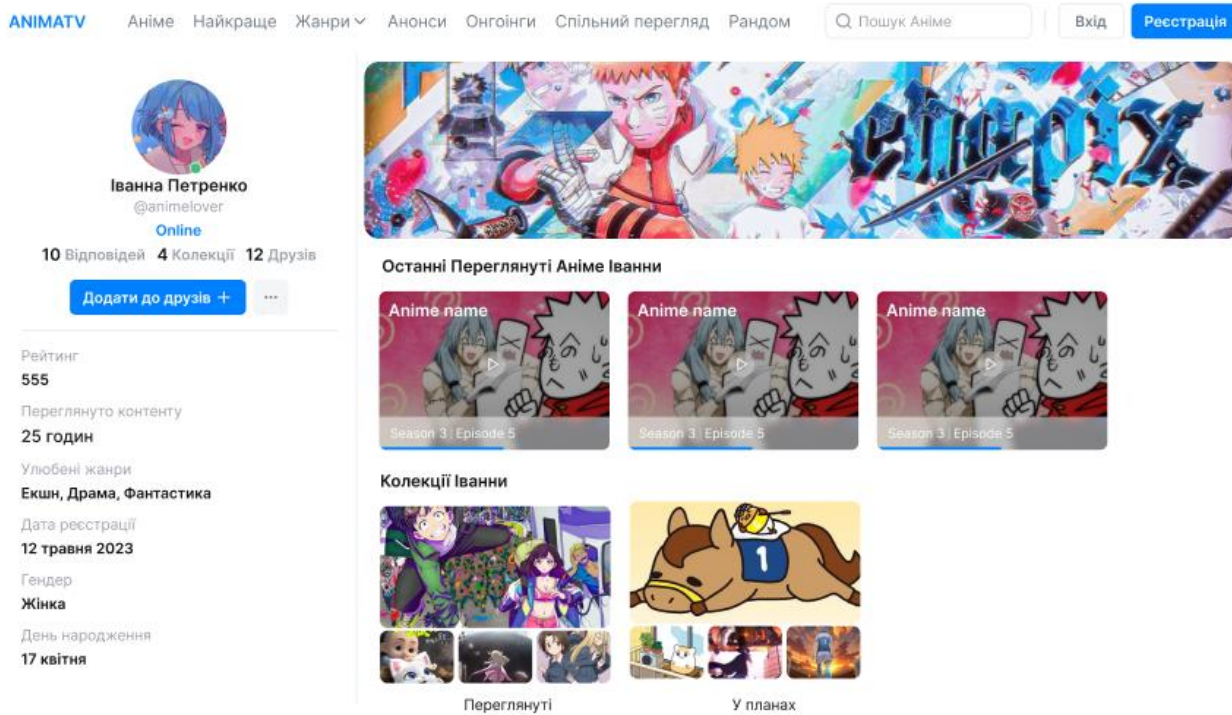


Рис. 2.19. Макет сторінки «Особистий кабінет користувача»

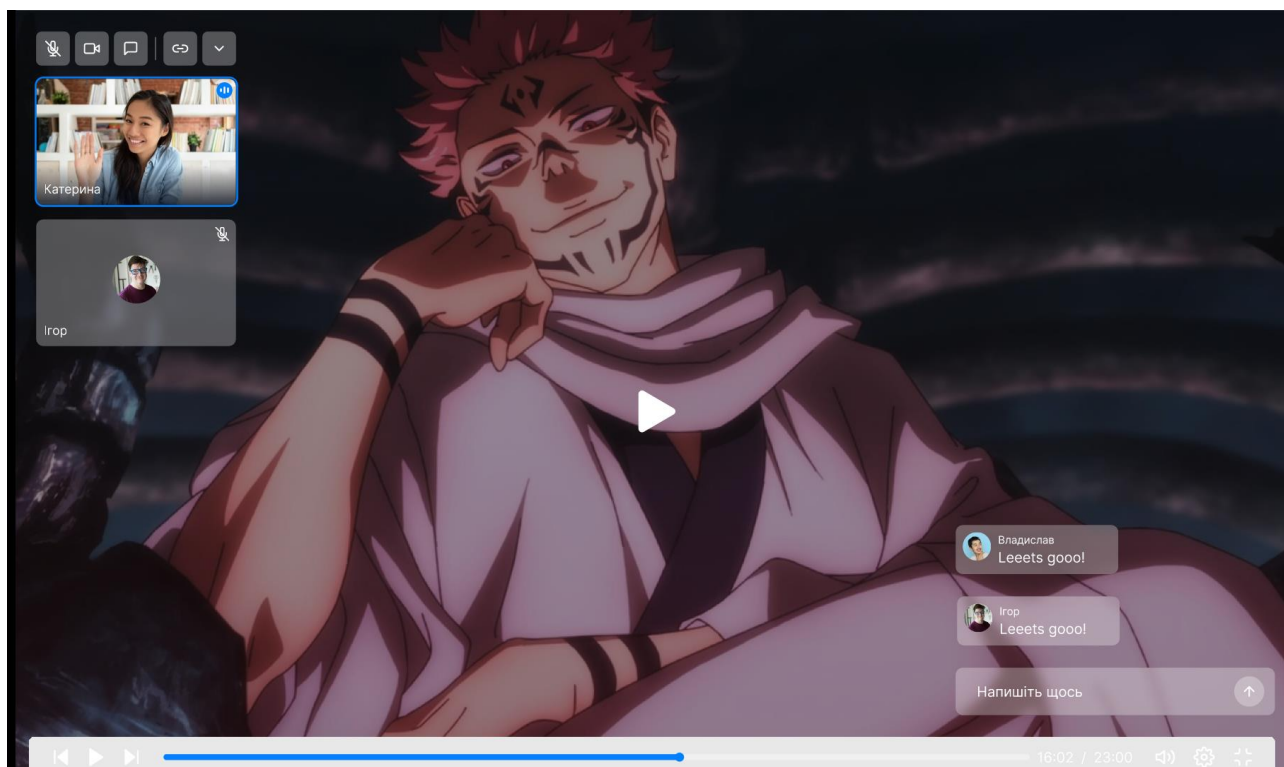


Рис. 2.20. Вигляд програвача в повноекранному режимі



## РОЗДІЛ 3

### ЕКОНОМІЧНИЙ РОЗДІЛ

#### **3.1. Розрахунок трудомісткості та вартості розробки програмного продукту**

Під час розробки стрімінгової веб-платформи для перегляду японської анімації проведено розрахунок трудомісткості та вартості з урахуванням наступних вихідних даних:

1. Передбачуване число операторів програми: 2000;
2. Коефіцієнт складності програми: 1,3;
3. Коефіцієнт корекції програмного продукту в ході його розробки: 0,1;
4. Годинна заробітна плата розробника: 188 грн/год (згідно зі статистикою з сайту WorkUA для України);
5. Коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі: 1,25;
6. Коефіцієнт кваліфікації програміста, залежний від стажу роботи: 0,8;
7. Вартість машино-години ЕОМ: 20 грн/год.

Цінова політика використання графічного редактору Figma за годину залежить від обраного тарифного плану. У фігмі є безкоштовний тариф для базового використання без функціональних обмежень. Якщо є потреба в командній роботі, то для цього потрібно придбати платний тарифний план, який коштує 476 грн. за користувача на місяць. Припустимо, що ви використовуєте програму стандартні 8 годин на день протягом 22 робочих днів на місяць (що є середнім числом робочих днів у місяці), то погодинна вартість користування програмою при місячній вартості 476 грн становить приблизно 2,70 грн/год.

Однак, не потрібно забувати, що ви також маєте сплачувати витрати за Інтернет, електроенергію, обладнання та інші ресурси, які необхідні для створення дизайну веб-платформи.

Заробітну плату розробника сайтів за годину було розраховано за допомогою даних, узятих з сайту для пошуку роботи «WORK.UA». Середня заробітна плата розробника сайтів по Україні складає приблизно 32500 грн. на місяць станом на травень 2024 рік [7].

Припустимо, стандартний робочий тиждень становить 40 годин. Кількість робочих годин на місяць приблизно виходить 188 грн/год.

Нормування праці при створенні програмного продукту значно ускладнено через силу творчого характеру розробника. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки програмного продукту можна розрахувати за допомогою формули:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \text{ людино} - \text{годин} \quad (3.1)$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$  – витрати праці на налагодження програми на ЕОМ;

$t_d$  – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де  $q$  – передбачуване число операторів (2000);

$C$  – коефіцієнт складності програми (1,3);

$p$  – коефіцієнт корекції програмного продукту в ході його розробки (0,1).

Таким чином, умовне число операторів становить:

$$Q = 2000 \cdot 1,3 \cdot (1 + 0,1) = 2860$$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \cdot 85) \cdot k}, \text{ людино – годин} \quad (3.3)$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (1,25);

$k$  – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності (0,8);

З урахуванням коефіцієнта кваліфікації  $k = 0,8$ , отримуємо витрати праці на вивчення опису завдання:

$$t_u = \frac{2860 \cdot 1,25}{85 \cdot 0,8} = 52,57 \text{ людино – годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \cdot 25) \cdot k}, \text{ людино – годин} \quad (3.4)$$

де  $Q$  – умовне число операторів програми;

$k$  – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.4), отримаємо:

$$t_a = \frac{2860}{20 \cdot 0,8} = 178,75 \text{ людино – годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_a = \frac{Q}{(20 \cdot 25) \cdot k}, \text{ людино – годин} \quad (3.5)$$

$$t_a = \frac{2050}{25 \cdot 0,8} = 102,5 \text{ людино – годин};$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4..5) \cdot k}, \text{ людино – годин} \quad (3.6)$$

$$t_{отл} = \frac{2860}{5 \cdot 0,8} = 715 \text{ людино – годин}$$

– за умови комплексного налагодження завдання:

$$t_{отл}^k = 1,5 \cdot t_{отл}, \text{ людино – годин} \quad (3.7)$$

$$t_{отл}^k = 1,5 \cdot 715 = 1072 \text{ людино – годин}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial\rho} + t_{\partial o}, \text{ людино – годин} \quad (3.8)$$

де  $t_{\partial\rho}$  – трудомісткість підготовки матеріалів і рукопису;

$$t_{\partial\rho} = \frac{Q}{(15..20) \cdot k}, \text{ людино – годин} \quad (3.9)$$

$$t_{\partial o} = 0,75 \cdot t_{\partial\rho}, \text{ людино – годин} \quad (3.10)$$

$t_{\partial o}$  – трудомісткість редагування, печатки й оформлення документації;

Підставляючи відповідні значення, отримаємо:

$$t_{\partial\rho} = \frac{2860}{18 \cdot 0,8} = 198,61 \text{ людино – годин};$$

$$t_{\partial o} = 0,75 \cdot 198,61 = 148,95 \text{ людино – годин};$$

$$t_{\partial} = 198,61 + 148,95 = 347,56 \text{ людино – годин};$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 52,57 + 178,75 + 143 + 715 + 347,56 = 1486,88 \text{ людино-годин.}$$

За результатами розрахунків, загальна трудомісткість розробки даного програмного продукту складає 1486,88 людино-годин.

### 3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ  $K_{ПО}$  включають витрати на заробітну плату виконавця програми  $Z_{ЗП}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн} \quad (3.11)$$

де  $Z_{ЗП}$  заробітна плата виконавців, яка визначається за формулою:

$$Z_{ЗП} = t \cdot C_{пр}, \text{ грн} \quad (3.12)$$

де  $t$  – загальна трудомісткість людино-годин,

$C_{пр}$  – середня годинна заробітна плата програміста, грн/год.

З урахуванням того, що середня годинна зарплата розробника становить 188 грн / год, отримуємо:

$$Z_{ЗП} = 1486,88 \cdot 188 = 279533,44 \text{ грн}$$

Вартість машинного часу  $Z_{МВ}$ , необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн} \quad (3.13)$$

де  $t_{отл}$  – трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$  – вартість машино-години ЕОМ, грн/год.

$$З_{\text{МВ}} = 715 \cdot 20 = 14300 \text{ грн}$$

Звідси витрати на створення програмного продукту:

$$K_{\text{по}} = 279533,44 + 14300 = 293833,44 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс} \quad (3.14)$$

де  $B_k$  – число виконавців,

$F_p$  – місячний фонд робочого часу (при 40 годинному робочому тижні  $F_p = 176$  годин.)

Звідси, за формулою (3.14) витрати на створення програмного продукту:

$$T = \frac{1486,88}{1 \cdot 176} = 8,44 \text{ міс.}$$

**Висновок:** Стрімінгову веб-платформу для розробки японської анімації створено для того, щоб користувачі змогли провести вільний час та відпочити за переглядом улюбленого аніме. Вартість даного програмного продукту становить близько 293833,44 грн. Приблизний час, потрібний для розробки, складає 8,44 місяців при стандартному 40-годинному робочому тижні і 176-годинному робочому місяці. Цей термін включає час, необхідний для проведення досліджень, розробки алгоритму, дизайну та проектування. Загальна кількість людино-годин, яку буде витрачено на розробку, складає 1486,88.

## ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено стрімінгову веб-платформу для перегляду японської анімації. Головною метою проекту було створення сучасного та інтуїтивно зрозумілого програмного продукту, який задовольняв би потреби як новачків у світі японської анімації, так і досвідчених фанатів.

Розробка здійснювалася на конструкторі сайтів Weblium за допомогою технологій HTML, CSS, JavaScript та React, а також з використанням середовища розробки Microsoft Visual Studio Code. Дизайн платформи був створений у середовищі Figma, що дозволило розробити детальний макет і візуальне оформлення веб-платформи.

Актуальність створення такої платформи підтверджується високим попитом на стрімінгові сервіси, які спрощують процес пошуку та перегляду аніме, а також забезпечують зручний доступ до улюблених анімацій без необхідності попереднього завантажування на пристрій. Крім того, веб-платформа сприяє популяризації японської культури, надаючи користувачам можливість пізнавального та розважального перегляду анімацій.

Під час роботи над проектом було проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки. Було обрано відповідну платформу для розробки, виконано проектування та реалізацію функціоналу програмного продукту.

В «Економічному розділі» визначено трудомісткість розробленої стрімінгової веб-платформи (1486,88 люд-год), проведений підрахунок вартості роботи по створенню програми (293833,44 грн.) та розраховано час на його створення (8,44 міс).

Таким чином, розробка стрімінгової веб-платформи для перегляду японської анімації є важливим і своєчасним завданням, що відповідає сучасним тенденціям ринку і потребам користувачів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аналіз конкурентів. URL: <https://im4udma.com.ua/blog/whats-a-competitive-analysis-how-do-you-conduct-one/> (дата звернення: 03.05.2024).
2. Історія розвитку Html | MarkupUA. MarkupUA. URL: <https://markup-ua.com/istoriya-razvitiya-html/> (дата звернення: 16.01.2024).
3. Конструктор сайтів – що це таке | HOSTiQ Wiki. HOSTiQ Wiki. URL: <https://hostiq.ua/wiki/about-sitebuilder/> (дата звернення: 05.05.2024).
4. Огляд конструкторів для створення власного сайту. Огляд програм, корисні поради по налаштуванню програм. URL: <https://sovety.pp.ua/index.php/ua/statti/windows/internet/2828-oglyad-konstruktoriv-dlya-stvorenniya-vlasnogo-sajtu> (дата звернення: 08.04.2024).
5. Омельчук Е. Що таке Figma: функції, інструменти і переваги - академія Wezom. Академія Wezom - Навчаємо ІТ технологіям з нуля. URL: <https://wezom.academy/chto-takoe-figma-funktsii-instrumenty-ipreimuschestva/> (дата звернення: 05.03.2024).
6. Платформа shopify: що це таке та як працює, огляд 2024. Розробка та реклама інтернет-магазинів під ключ на Shopify. URL: <https://wallnut.digital/shops/shopify/review/> (дата звернення: 27.05.2024).
7. Статистика зарплат в Україні. Work.ua – сайт пошуку роботи №1 в Україні. URL: <https://www.work.ua/salary> (дата звернення: 20.05.2024).
8. Структура сайту. URL: <https://webtune.com.ua/statti/web-rozrobka/struktura-sajtu/> (дата звернення: 07.03.2024).
9. Трубочова, В. "КОНСТРУКТОР САЙТІВ: АНАЛІЗ РОБОТИ.". URL: [https://repository.mu.edu.ua/jspui/bitstream/123456789/1558/1/mat\\_metody\\_2020.pdf#page=282](https://repository.mu.edu.ua/jspui/bitstream/123456789/1558/1/mat_metody_2020.pdf#page=282) (дата звернення: 24.03.2024).
10. Create a site for free: weblium site builder. Create a site for free: Weblium site builder. URL: <https://weblium.com/> (дата звернення: 12.03.2024).



11. CSS Довідник. Уроки для початківців. W3Schools українською. W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів. URL: <https://w3schoolsua.github.io/cssref/index.html#gsc.tab=0> (дата звернення: 17.04.2024).
12. Duckett J. HTML and CSS: design and build web sites. Wiley & Sons, Incorporated, John, 2012.
13. Figma: the collaborative interface design tool. Figma. URL: <https://www.figma.com/> (дата звернення: 01.03.2024).
14. Home. HTML For Beginners The Easy Way: Start Learning HTML & CSS Today ». URL: <https://html.com/> (дата звернення: 04.04.2024).
15. JavaScript переменные. Веб Учебники онлайн. URL: [https://schoolsw3.com/js/js\\_variables.php](https://schoolsw3.com/js/js_variables.php) (дата звернення: 22.05.2024).
16. Merge Academy – курси IT з працевлаштуванням від кращих менторів агенції Merge. Merge Academy – курси IT з працевлаштуванням від кращих менторів агенції Merge. URL: <https://merge.academy/blog/shcho-takie-webflow> (дата звернення: 27.04.2024).
17. Microsoft. Documentation for visual studio code. Visual Studio Code - Code Editing. Redefined. URL: <https://code.visualstudio.com/docs/> (дата звернення: 14.04.2024).
18. Novielli M. R. Floating worlds: a short history of japanese animation. Taylor & Francis Group, 2018. 172 с.
19. React. URL: <https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/> (дата звернення: 19.01.2024).
20. Readymag. Readymag. URL: <https://readymag.com/readymag/companies/what-is-readymag/> (дата звернення: 28.04.2024).
21. Портрет ідеального клієнта: як правильно аналізувати аудиторію. Школа бізнесу. URL: <https://online.novaposhta.education/blog/portret-idealnogo-klienta-yak-pravilno-analizuvati-auditoriyu> (дата звернення: 10.04.2024).

## КОД ПРОГРАМИ

**index.html**

```
<!DOCTYPE html>
<html lang="en" class="site-root site-root--published fonts-stage-1"
      data-scrapbook-source="https://animatv.weblium.site/" data-scrapbook-create="20240527134330423">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta property="og:type" content="website">
  <meta property="og:url" content="https://animatv.weblium.site/">
  <meta name="color-scheme" content="light only">
  <title>ANIMATV | Home</title>
  <meta name="generator" content="Weblium, effortless website builder">
  <meta name="robots" content="index, follow">
  <link rel="canonical" href="https://animatv.weblium.site/">
  <meta name="theme-color" content="#0080ff">
  <meta property="og:title" content="ANIMATV | Home">

  <link rel="stylesheet" href="google-fonts-5c2dfa623dbc4a0023e12261.css">

  <style type="text/css">
    a,
    abbr,
    acronym,
    address,
    applet,
    article,
    aside,
    audio,
    b,
    big,
    blockquote,
    body,
    canvas,
    caption,
    center,
    cite,
    code,
    dd,
    del,
    details,
    dfn,
    div,
    dl,
    dt,
    embed,
    fieldset,
    figcaption,
    figure,
    footer,
    form,
    h1,
```

h2,  
h3,  
h4,  
h5,  
h6,  
header,  
hgroup,  
html,  
i,  
iframe,  
img,  
ins,  
kbd,  
label,  
legend,  
li,  
mark,  
menu,  
nav,  
object,  
ol,  
output,  
p,  
pre,  
q,  
ruby,  
s,  
samp,  
section,  
small,  
span,  
strike,  
strong,  
sub,  
summary,  
sup,  
table,  
tbody,  
td,  
tfoot,  
th,  
thead,  
time,  
tr,  
tt,  
u,  
ul,  
var,  
video {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    vertical-align: baseline  
}  
  
em {  
    margin: 0;  
    padding: 0;  
    border: 0;

```

        font-size: 100%;
        vertical-align: baseline
    }

    article,
    aside,
    details,
    figcaption,
    figure,
    footer,
    header,
    hgroup,
    menu,
    nav,
    section {
        display: block
    }

    body {
        line-height: 1
    }

    ol,
    ul {
        list-style: none
    }

    blockquote,
    q {
        quotes: none
    }

    blockquote:after,
    blockquote:before,
    q:after,
    q:before {
        content: "";
        content: none
    }

    table {
        border-collapse: collapse;
        border-spacing: 0
    }

    html {
        width: 100%;
        overflow-x: hidden;
        -webkit-font-smoothing: antialiased;
        -moz-osx-font-smoothing: grayscale
    }

    *,
    :after,
    :before,
    html {
        -webkit-box-sizing: border-box;
        -moz-box-sizing: border-box;
        box-sizing: border-box
    }

```

```

button {
    background: none;
    border: 0;
    margin: 0;
    padding: 0;
    font-size: inherit;
    font-family: inherit
}

.slick-track:after,
.slick-track:before {
    content: "";
    display: table;
    clear: both
}

img:not([src]) {
    visibility: hidden
}

img[src] {
    background-color: transparent !important
}

/*--vars-start--*/
:root {
    --w-accordion-offset: 30px;
    --w-accordion-padding: 20px
}

:root {
    --w-apps-buttons-border-color: #a6a6a6;
    --w-apps-buttons-padding-top-as: 33.3333%;
    --w-apps-buttons-padding-top-gp: 29.5496%;
    --w-apps-buttons-padding-top-as-fre: 31.496%;
    --w-apps-buttons-padding-top-as-tur: 26.4901%;
    --w-apps-buttons-padding-top-as-chi: 36.6972%;
    --w-apps-buttons-sm-max-width-as: 120px;
    --w-apps-buttons-sm-max-width-gp: 135px;
    --w-apps-buttons-sm-max-width-as-fre: 127px;
    --w-apps-buttons-sm-max-width-as-tur: 151px;
    --w-apps-buttons-sm-max-width-as-chi: 109px;
    --w-apps-buttons-sm-border-radius: 5px;
    --w-apps-buttons-offset: 35px;
    --w-apps-buttons-md-max-width-as: 120px;
    --w-apps-buttons-lg-max-width-as: 150px;
    --w-apps-buttons-md-max-width-gp: 135px;
    --w-apps-buttons-lg-max-width-gp: 169px;
    --w-apps-buttons-sm-max-width-as-fre: 127px;
    --w-apps-buttons-md-max-width-as-fre: 127px;
    --w-apps-buttons-lg-max-width-as-fre: 159px;
    --w-apps-buttons-sm-max-width-as-tur: 151px;
    --w-apps-buttons-md-max-width-as-tur: 151px;
    --w-apps-buttons-lg-max-width-as-tur: 189px;
    --w-apps-buttons-sm-max-width-as-chi: 109px;
    --w-apps-buttons-md-max-width-as-chi: 109px;
    --w-apps-buttons-lg-max-width-as-chi: 136px;
    --w-apps-buttons-sm-border-radius: 5px;
    --w-apps-buttons-md-border-radius: 5px;
    --w-apps-buttons-lg-border-radius: 5px
}

```

```

@media (min-width: 480px) {
  :root:not(.disable-media-queries) {
    --w-apps-buttons-md-max-width-as: 150px;
    --w-apps-buttons-lg-max-width-as: 180px;
    --w-apps-buttons-md-max-width-gp: 169px;
    --w-apps-buttons-lg-max-width-gp: 203px;
    --w-apps-buttons-md-max-width-as-fre: 159px;
    --w-apps-buttons-lg-max-width-as-fre: 191px;
    --w-apps-buttons-md-max-width-as-tur: 189px;
    --w-apps-buttons-lg-max-width-as-tur: 227px;
    --w-apps-buttons-md-max-width-as-chi: 136px;
    --w-apps-buttons-lg-max-width-as-chi: 164px
  }
}

@media (min-width: 0px) {
  .html-editor-md,
  .html-editor {
    --w-apps-buttons-md-max-width-as: 150px;
    --w-apps-buttons-lg-max-width-as: 180px;
    --w-apps-buttons-md-max-width-gp: 169px;
    --w-apps-buttons-lg-max-width-gp: 203px;
    --w-apps-buttons-md-max-width-as-fre: 159px;
    --w-apps-buttons-lg-max-width-as-fre: 191px;
    --w-apps-buttons-md-max-width-as-tur: 189px;
    --w-apps-buttons-lg-max-width-as-tur: 227px;
    --w-apps-buttons-md-max-width-as-chi: 136px;
    --w-apps-buttons-lg-max-width-as-chi: 164px
  }
}

@media (min-width: 768px) {
  :root:not(.disable-media-queries) {
    --w-apps-buttons-md-border-radius: 7px;
    --w-apps-buttons-lg-border-radius: 8px
  }
}

@media (min-width: 0px) {
  .html-editor-md,
  .html-editor {
    --w-apps-buttons-md-border-radius: 7px;
    --w-apps-buttons-lg-border-radius: 8px
  }
}

@media (min-width: 992px) {
  :root:not(.disable-media-queries) {
    --w-apps-buttons-offset: 40px
  }
}

@media (min-width: 0px) {
  .html-editor {
    --w-apps-buttons-offset: 40px
  }
}

```

```

:root {
  --w-gap: 15px
}

:root {
  --w-base-v-space: 4px;
  --w-base-h-space: 5px;
  --w-v-spacer1: calc(var(--w-base-v-space) * 1);
  --w-v-spacer2: calc(var(--w-base-v-space) * 2);
  --w-v-spacer3: calc(var(--w-base-v-space) * 3);
  --w-v-spacer4: calc(var(--w-base-v-space) * 5);
  --w-v-spacer5: calc(var(--w-base-v-space) * 8);
  --w-v-spacer6: calc(var(--w-base-v-space) * 12);
  --w-v-spacer7: calc(var(--w-base-v-space) * 15);
  --w-h-spacer1: calc(var(--w-base-h-space) * 1);
  --w-h-spacer2: calc(var(--w-base-h-space) * 2);
  --w-h-spacer3: calc(var(--w-base-h-space) * 4);
  --w-h-spacer4: calc(var(--w-base-h-space) * 8)
}

:root {
  --w-hero-title-offset: 35px;
  --w-title-offset: 25px;
  --w-subtitle-offset: 25px;
  --w-heading-offset: 20px;
  --w-subheading-offset: 25px;
  --w-heading-lg-offset: 25px;
  --w-caption-offset: 20px;
  --w-body-offset: 20px;
  --w-small-offset: 20px;
  --w-quote-offset: 20px;
  --w-hero-title-h-offset: 15px;
  --w-title-h-offset: 5px;
  --w-subtitle-h-offset: 15px;
  --w-heading-h-offset: 15px;
  --w-subheading-h-offset: 15px;
  --w-heading-lg-h-offset: 15px;
  --w-caption-h-offset: 10px;
  --w-body-h-offset: 10px;
  --w-small-h-offset: 10px;
  --w-quote-h-offset: 10px
}

@media (min-width: 992px) {
  :root:not(.disable-media-queries) {
    --w-hero-title-offset: 40px
  }
}

@media (min-width: 0px) {
  .html-editor {
    --w-hero-title-offset: 40px
  }
}

:root {
  --w-layout-item-top-offset: calc(var(--w-gap) * 2);
  --w-layout-gap: var(--w-gap)
}

:root {

```

```

        --w-btns-group-item-offset: 20px;
        --w-btns-group-offset: 35px
    }

    @media (min-width: 992px) {
        :root:not(.disable-media-queries) {
            --w-btns-group-offset: 40px
        }
    }

    @media (min-width: 0px) {

        .html-editor,
        .html-editor-lg {
            --w-btns-group-offset: 40px
        }
    }

    :root {
        --w-socials-top-offset: 30px;
        --w-socials-h-offset: 15px;
        --w-socials-bottom-offset: 30px
    }

    :root {
        --w-list-item-v-offset: 30px;
        --w-list-item-h-offset: 30px;
        --w-list-item-v-padding: 15px;
        --w-list-item-h-padding: 15px
    }

    :root {
        --w-list-with-media-offset: 30px;
        --w-list-with-media-padding: 20px
    }

    :root {
        --ui-content-gutter-1200: 15px;
        --w-content-container-size: 1200px;
        --w-content-gutter: var(--ui-content-gutter-320)
    }

    @media (min-width: 480px) {
        :root:not(.disable-media-queries) {
            --w-content-gutter: var(--ui-content-gutter-480)
        }
    }

    @media (min-width: 0px) {
        .html-editor-xs {
            --w-content-gutter: var(--ui-content-gutter-480)
        }
    }

    @media (min-width: 768px) {
        :root:not(.disable-media-queries) {
            --w-content-gutter: var(--ui-content-gutter-768)
        }
    }

    @media (min-width: 0px) {

```



```

        .html-editor-md {
            --w-content-gutter: var(--ui-content-gutter-768)
        }
    }

    @media (min-width: 992px) {
        :root:not(.disable-media-queries) {
            --w-content-gutter: var(--ui-content-gutter-992)
        }
    }

    @media (min-width: 1280px) {
        :root:not(.disable-media-queries) {
            --w-content-gutter: var(--ui-content-gutter-1200)
        }
    }

    @media (min-width: 0px) {

        .html-editor,
        .html-editor-lg {
            --w-content-gutter: var(--ui-content-gutter-1200)
        }
    }

    :root {
        --w-section-vertical-s-offset: 20px;
        --w-section-basement-offset: 30px;
        --w-section-vertical-offset: 90px;
        --w-section-vertical-xxl-offset: 130px;
        --w-section-header-offset: 45px;
        --w-section-footer-offset: 50px
    }

    @media (min-width: 768px) {
        :root:not(.disable-media-queries) {
            --w-section-vertical-offset: 95px;
            --w-section-vertical-xxl-offset: 250px;
            --w-section-header-offset: 45px;
            --w-section-footer-offset: 50px
        }
    }

    @media (min-width: 0px) {
        .html-editor-md {
            --w-section-vertical-offset: 95px;
            --w-section-vertical-xxl-offset: 250px;
            --w-section-header-offset: 45px;
            --w-section-footer-offset: 50px
        }
    }

    @media (min-width: 992px) {
        :root:not(.disable-media-queries) {
            --w-section-vertical-offset: 100px;
            --w-section-vertical-xxl-offset: 250px;
            --w-section-header-offset: 50px;
            --w-section-footer-offset: 55px
        }
    }
}

```

```

@media (min-width: 0px) {
  .html-editor {
    --w-section-vertical-offset: 100px;
    --w-section-vertical-xxl-offset: 250px;
    --w-section-header-offset: 50px;
    --w-section-footer-offset: 55px
  }
}

:root {
  --w-image-offset: 20px;
  --w-image-mask-sm-width: 70px;
  --w-image-mask-sm-height: 70px;
  --w-image-mask-md-width: 70px;
  --w-image-mask-md-height: 70px;
  --w-image-mask-lg-width: 120px;
  --w-image-mask-lg-height: 120px
}

@media (min-width: 768px) {
  :root:not(.disable-media-queries) {
    --w-image-aspect-ratio: 56.25%;
    --w-image-mask-md-width: 120px;
    --w-image-mask-md-height: 120px;
    --w-image-mask-lg-width: 170px;
    --w-image-mask-lg-height: 170px
  }
}

@media (min-width: 0px) {
  .html-editor,
  .html-editor-md {
    --w-image-aspect-ratio: 56.25%;
    --w-image-mask-md-width: 120px;
    --w-image-mask-md-height: 120px;
    --w-image-mask-lg-width: 170px;
    --w-image-mask-lg-height: 170px
  }
}

:root {
  --w-quote-offset: 30px;
  --w-quote-gap: 20px
}

:root {
  --w-logo-offset: 30px;
  --w-slogan-offset: 5px
}

:root {
  --w-nav-offset: 30px;
  --w-nav-item-gap: 15px;
  --w-nav-item-top-offset: 10px;
  --w-nav-item-column-gap: 10px;
  --w-nav-item-column-offset: 20px
}

:root {
  --w-burger-padding: 12px 8px;

```

```

        --w-burger-width: 24px;
        --w-burger-height: 16px;
        --w-burger-line-size: 2px;
        --w-burger-line-pos-transform: translateY(7px) rotate(45deg);
        --w-burger-line-neg-transform: translateY(-7px) rotate(-45deg);
    }

:root {
    --w-icon-panel-inner-offset: 0
}

:root:not(.disable-media-queries) {
    --w-icon-panel-sm-width: 50px;
    --w-icon-panel-sm-height: 50px;
    --w-icon-panel-width: 50px;
    --w-icon-panel-height: 50px;
    --w-icon-panel-xl-width: 70px;
    --w-icon-panel-xl-height: 70px
}

@media (min-width: 0px) {
    .html-editor-xs {
        --w-icon-panel-sm-width: 50px;
        --w-icon-panel-sm-height: 50px;
        --w-icon-panel-width: 50px;
        --w-icon-panel-height: 50px;
        --w-icon-panel-xl-width: 70px;
        --w-icon-panel-xl-height: 70px
    }
}

@media (min-width: 768px) {
    :root:not(.disable-media-queries) {
        --w-icon-panel-sm-width: 50px;
        --w-icon-panel-sm-height: 50px;
        --w-icon-panel-width: 70px;
        --w-icon-panel-height: 70px;
        --w-icon-panel-xl-width: 100px;
        --w-icon-panel-xl-height: 100px
    }
}

@media (min-width: 0px) {
    .html-editor,
    .html-editor-md {
        --w-icon-panel-sm-width: 50px;
        --w-icon-panel-sm-height: 50px;
        --w-icon-panel-width: 70px;
        --w-icon-panel-height: 70px;
        --w-icon-panel-xl-width: 100px;
        --w-icon-panel-xl-height: 100px
    }
}

:root {
    --w-icon-width: 24px;
    --w-icon-height: 24px;
    --w-icon-offset: 20px;
    --w-icon-h-offset: 10px;
    --w-icon-sm-inner-offset: 12px;

```

```

        --w-icon-inner-offset: 12px;
        --w-icon-numbers-inner-offset: 5px;
        --w-icon-xl-inner-offset: 15px
    }

    @media (min-width: 768px) {
        :root:not(.disable-media-queries) {
            --w-icon-inner-sm-offset: 12px;
            --w-icon-inner-offset: 15px;
            --w-icon-xl-inner-offset: 20px
        }
    }

    @media (min-width: 0px) {

        .html-editor,
        .html-editor-md {
            --w-icon-inner-sm-offset: 12px;
            --w-icon-inner-offset: 15px;
            --w-icon-xl-inner-offset: 20px
        }
    }

    :root {
        --w-icon-numbers-sm-inner-offset: 4px
    }

    :root:not(.disable-media-queries) {
        --w-icon-numbers-xl-inner-offset: 8px;
        --w-icon-numbers-inner-offset: 4px
    }

    @media (min-width: 0px) {
        .html-editor-xs {
            --w-icon-numbers-xl-inner-offset: 8px;
            --w-icon-numbers-inner-offset: 4px
        }
    }

    @media (min-width: 768px) {
        :root:not(.disable-media-queries) {
            --w-icon-numbers-xl-inner-offset: 10px;
            --w-icon-numbers-inner-offset: 8px
        }
    }

    @media (min-width: 0px) {

        .html-editor,
        .html-editor-md {
            --w-icon-numbers-xl-inner-offset: 10px;
            --w-icon-numbers-inner-offset: 8px
        }
    }

    :root {
        --w-map-wrapper-height: 450px;
        --w-map-wrapper-offset: 30px
    }

    :root:not(.disable-media-queries) {

```

```

        --w-counter-font-size: 24px;
        --w-counter-xl-font-size: 30px
    }

    @media (min-width: 0px) {
        .html-editor-xs {
            --w-counter-font-size: 24px;
            --w-counter-xl-font-size: 30px
        }
    }

    @media (min-width: 768px) {
        :root:not(.disable-media-queries) {
            --w-counter-font-size: 30px;
            --w-counter-xl-font-size: 44px
        }
    }

    @media (min-width: 0px) {

        .html-editor,
        .html-editor-md {
            --w-counter-font-size: 30px;
            --w-counter-xl-font-size: 44px
        }
    }

    :root {
        --w-contacts-list-offset: 30px;
        --w-contacts-list-item-v-offset: 16px;
        --w-contacts-list-element-v-offset: 5px
    }

    :root {
        --w-icon-list-width: 20px;
        --w-icon-list-height: 20px;
        --w-icon-list-offset: 20px;
        --w-icon-list-item-offset: 16px;
        --w-icon-list-item-h-offset: 24px
    }

    :root {
        --w-follow-us-offset: 15px
    }

    :root {
        --w-form-offset: 32px;
        --w-text-area-height: 150px;
        --w-form-field-offset: 24px;
        --w-form-button-offset: 56px;
        --w-form-input-offset: 12px;
        --w-form-description-offset: 4px;
        --w-form-multiple-group-offset: 12px;
        --w-form-multiple-item-offset: 12px
    }

    :root {
        --w-captions-group-offset: 15px;
        --w-captions-group-item-offset: 5px;
        --w-captions-group-separator-width: 1px
    }

```

```

:root {
  --w-user-message-offset: 20px;
  --w-user-photo-size: 70px
}

:root {
  --w-grid-item-top-offset: 30px;
  --w-grid-gap: 30px;
  --w-100vw: 100vw;
  --w-grid-container-size-md: calc(var(--w-100vw) + var(--w-grid-gap) - var(--ui-content-
gutter-768) * 2);
  --w-grid-container-size-xl: calc(var(--w-content-container-size) + var(--w-grid-gap) - var(--ui-
content-gutter-1200) * 2);
  --w-grid-container-size-lg: var(--w-grid-container-size-xl)
}

:root:not(.disable-media-queries) {
  --w-100vw: 100vw
}

.html-editor-xs,
.html-editor-md {
  --w-100vw: var(--w-editor-width)
}

:root:not(.disable-media-queries) {
  --w-grid-item-v-padding: 31px;
  --w-grid-item-h-padding: 27px
}

@media (min-width: 0px) {
  .html-editor-xs {
    --w-grid-item-v-padding: 31px;
    --w-grid-item-h-padding: 27px
  }
}

@media (min-width: 768px) {
  :root:not(.disable-media-queries) {
    --w-grid-item-v-padding: 34px;
    --w-grid-item-h-padding: 28px
  }
}

@media (min-width: 0px) {
  .html-editor-md {
    --w-grid-item-v-padding: 34px;
    --w-grid-item-h-padding: 28px
  }
}

@media (min-width: 992px) {
  :root:not(.disable-media-queries) {
    --w-grid-item-v-padding: 40px;
    --w-grid-item-h-padding: 30px
  }
}

@media (min-width: 0px) {

```

```

        .html-editor,
        .html-editor-lg {
            --w-grid-item-v-padding: 40px;
            --w-grid-item-h-padding: 30px
        }
    }

:root {
    --w-divider-component-offset: 0px
}

:root {
    --w-gallery-offset: 30px
}

:root {
    --w-image-with-hover-offset: 20px;
    --w-image-with-hover-button-offset: var(--w-btns-group-offset)
}

:root {
    --w-try-it-free-offset: 32px;
    --w-try-it-free-input-offset-sm: 8px;
    --w-try-it-free-input-offset-md: 8px;
    --w-try-it-free-input-offset-lg: 12px
}

:root {
    --w-language-offset: 20px
}

/*--vars-end--*/

.media-placeholder_1Ay {
    display: flex !important;
    position: absolute;
    height: 100%;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    justify-content: center !important;
    align-items: center;
    overflow: hidden
}

.icon_1X9 {
    height: calc(100% - 160px);
    min-height: 50px;
    max-height: 140px
}

.ai-visual-container_3T8 {
    position: absolute;
    top: 5px;
    right: 5px;
    z-index: 2
}

.ai-visual-button_vk_ {

```

```

        display: flex;
        padding: 8px 12px;
        border-radius: 4px;
        border: 0;
        box-shadow: 0 2px 4px 0 rgba(33, 39, 54, .16);
        -webkit-user-select: none;
        -moz-user-select: none;
        user-select: none
    }

    .ai-visual-button-image_1V1 {
        width: 24px;
        height: 24px
    }

    .ai-visual-button-label_Uim {
        font-size: 14px;
        font-weight: 600;
        line-height: 20px;
        color: var(--brand)
    }

    .ai-visual-button_vk_:hover {
        box-shadow: 0 0 1px 0 rgba(33, 39, 54, .04)
    }

    a,
    abbr,
    acronym,
    address,
    applet,
    article,
    aside,
    audio,
    b,
    big,
    blockquote,
    body,
    canvas,
    caption,
    center,
    cite,
    code,
    dd,
    del,
    details,
    dfn,
    div,
    dl,
    dt,
    embed,
    fieldset,
    figcaption,
    figure,
    footer,
    form,
    h1,
    h2,
    h3,
    h4,
    h5,

```



h6,  
header,  
hgroup,  
html,  
i,  
iframe,  
img,  
ins,  
kbd,  
label,  
legend,  
li,  
mark,  
menu,  
nav,  
object,  
ol,  
output,  
p,  
pre,  
q,  
ruby,  
s,  
samp,  
section,  
small,  
span,  
strike,  
strong,  
sub,  
summary,  
sup,  
table,  
tbody,  
td,  
tfoot,  
th,  
thead,  
time,  
tr,  
tt,  
u,  
ul,  
var,  
video {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    font: inherit;  
    vertical-align: baseline  
}  
  
em {  
    margin: 0;  
    padding: 0;  
    border: 0;  
    font-size: 100%;  
    vertical-align: baseline  
}

```

article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
menu,
nav,
section {
    display: block
}

blockquote,
q {
    quotes: none
}

blockquote:after,
blockquote:before,
q:after,
q:before {
    content: ""
}

table {
    border-collapse: collapse;
    border-spacing: 0
}

*,
:after,
:before {
    box-sizing: border-box
}

button {
    margin: 0;
    padding: 0;
    background: none;
    border: 0;
    font-size: inherit;
    font-family: inherit
}

html {
    font-size: var(--font-base-size);
    font-family: var(--font-family);
    line-height: var(--line-height-base);
    color: var(--ink);
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale
}

body,
html {
    margin: 0
}

button,

```

```

    figure,
    iframe {
        margin: 0;
        padding: 0;
        background: none;
        border: 0;
        cursor: pointer;
        border-radius: 0;
        font-size: inherit;
        outline: none
    }

    input {
        border: 0;
        box-shadow: none
    }

    input:focus {
        outline: none
    }

```

## style.css

```

.editor-form-field_17x {
    display: flex;
    max-width: 100%;
    flex-wrap: wrap
}

.editor-form-field_17x:not(:last-child) {
    margin-bottom: calc(var(--ui-text-size)*var(--ui-text-line-height)/2)
}

.editor-form-field_17x .checkbox {
    position: relative;
    width: 24px;
    height: 24px;
    margin-right: 6px;
    flex-shrink: 0
}

.editor-form-field_17x .checkbox input {
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    margin: 0;
    padding: 0;
    z-index: 1;
    opacity: 0;
    cursor: pointer
}

.editor-form-field_17x .checkbox input:checked+.mask {
    background-color: #757575
}

.editor-form-field_17x .checkbox input:checked+.mask svg {
    opacity: 1
}

```

```

}

.editor-form-field_17x .checkbox input:focus+.mask {
    outline: auto;
    outline-color: #3b99fc
}

.editor-form-field_17x .mask {
    position: absolute;
    width: calc(100% - 6px);
    height: calc(100% - 6px);
    top: 3px;
    left: 3px;
    z-index: 0;
    border-radius: 2px;
    border: 2px solid #757575;
    background-color: transparent;
    transition: background-color .2s ease-in
}

.editor-form-field_17x .mask svg {
    display: block;
    width: 80%;
    height: 100%;
    margin: auto;
    fill: #fff;
    opacity: 0;
    transition: opacity .2s ease-in
}

.editor-form-field_17x .label {
    max-width: calc(100% - 30px);
    flex-basis: calc(100% - 30px);
    cursor: pointer
}

.editor-form-field_Uzk {
    max-width: 100%
}

.editor-form-field_Uzk .select {
    position: relative;
    margin-top: 15px
}

.editor-form-field_Uzk .label {
    cursor: pointer
}

.editor-form-field_Uzk .description {
    display: block;
    text-align: left
}

.editor-form-field_Uzk .select__arrow {
    position: absolute;
    width: 18px;
    height: 18px;
    top: 50%;
    right: 15px;
    transform: translateY(-50%);
}

```

```

        color: #505050;
        pointer-events: none
    }

.editor-form-field_Uzk .select__arrow svg {
    display: block;
    width: 100%;
    height: 100%;
    fill: currentColor
}

.editor-form-field_Uzk .default-option {
    opacity: .8
}

.editor-form-field_Uzk select {
    width: 100%;
    padding-right: 40px;
    background-color: hsla(0, 0%, 40.4%, .2);
    border: 0;
    color: inherit;
    cursor: pointer;
    transition: box-shadow .2s ease-out;
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    -webkit-user-select: none;
    -moz-user-select: none;
    user-select: none;
    font-family: inherit
}

.editor-form-field_Uzk select:focus {
    outline: auto;
    outline-color: #3b99fc
}

.button_24J {
    display: inline-block;
    position: relative;
    max-width: 100%;
    text-decoration: none;
    text-align: center;
    background-position: 50%;
    overflow: visible;
    cursor: pointer;
    transition: color .2s ease-in, background-color .2s ease-in, border-color .2s ease-in, box-shadow .2s ease-in;
    word-wrap: break-word
}

.button--font-weight-bold_3_w .button__content_1on {
    font-weight: 700
}

.button--font-weight-normal_IGi .button__content_1on {
    font-weight: 400
}

.button--font-weight-inherit_5Oh .button__content_1on {
    font-weight: inherit
}

```

```

.button--font-style-italic_16q .button__content_1on {
    font-style: italic
}

.button--font-style-initial_Etk .button__content_1on {
    font-style: normal
}

.button--text-decoration-underline_yD2 .button__content_1on {
    text-decoration: underline
}

.button--left-icon_3Ep .button__content_1on {
    padding-left: 32px
}

.button--right-icon_26q .button__content_1on {
    padding-left: 0;
    padding-right: 32px
}

.button__inner_2mC {
    display: flex;
    position: relative;
    justify-content: center;
    align-items: center
}

.button__border_17U {
    position: absolute;
    width: 100%;
    height: 100%;
    top: 0;
    left: 0;
    transition: inherit
}

.button__content_1on {
    display: inline-block;
    position: relative;
    max-width: 100%;
    word-wrap: break-word
}

.button--with-icon_1uQ .button__content_1on {
    max-width: 100%
}

.button__icon_VM9 {
    display: inline-block;
    position: absolute;
    width: 24px;
    height: 24px;
    top: 50%;
    transform: translateY(-50%)
}

.button__icon--left_2cM {
    left: 0
}

```

```

.button__icon--right_ooy {
    right: 0
}

.button__icon-image_liq {
    height: 100%;
    max-width: 100%;
    margin: auto
}

.button_24J.disabled {
    cursor: not-allowed;
    pointer-events: none !important;
    filter: grayscale(.6)
}

.hiddenText_1oO {
    height: 1em;
    visibility: hidden
}

.button-wrapper_1g5 {
    display: block;
    width: 100%;
    height: 100%
}

.modal_33M {
    --overlay-bg-color: #f2f4f7;
    --box-shadow: 0 1px 20px 0 rgba(100, 110, 130, 0.16);
    display: flex;
    position: fixed;
    justify-content: center;
    align-items: center;
    z-index: var(--max-index, 9999)
}

.modal_33M,
.modal__overlay_2eY {
    top: 0;
    right: 0;
    bottom: 0;
    left: 0
}

.modal__overlay_2eY {
    position: absolute;
    background-color: var(--overlay-bg-color);
    opacity: .5
}

.modal__content_1zJ {
    position: relative;
    min-width: 344px;
    margin: 0 auto;
    padding: 43px 32px 65px;
    border-radius: 3px;
    box-shadow: var(--box-shadow);
    animation: fadeIn_2AA .2s ease-out forwards
}

```

```

.modal--editable_3Os {
    z-index: auto
}

.modal--editable_3Os .message__close-btn_1fi {
    cursor: auto
}

.svg_2MK {
    width: 100%;
    height: 100%;
    max-width: 100%;
    max-height: 100%;
    fill: currentColor
}

.message_2kG {
    text-align: center
}

.message--editable_1iK .message__btn_1xl {
    cursor: text
}

.message__status-ico_3jr {
    display: flex;
    width: 34px;
    height: 34px;
    margin: 0 auto 33px;
    align-items: center;
    justify-content: center;
    color: var(--ui-brand-color)
}

.message__close-btn_1fi {
    position: absolute;
    width: 16px;
    height: 16px;
    top: 16px;
    right: 24px;
    padding: 4px;
    opacity: .8;
    transition: opacity .2s ease-out
}

.message__close-btn_1fi:hover {
    opacity: 1
}

.message__close-btn_1fi:after,
.message__close-btn_1fi:before {
    content: "";
    position: absolute;
    width: 100%;
    height: 2px;
    left: 0;
    background-color: var(--ui-dark-shade-color, #212736)
}

.message__close-btn_1fi:after {

```



```

        transform: rotate(45deg)
    }

.message__close-btn_1fi:before {
    transform: rotate(-45deg)
}

.message__title_3jF {
    margin: 0 0 8px;
    font-weight: 700;
    font-size: 16px;
    line-height: 1.5;
    font-family: var(--ui-block-title-font-family)
}

.message__text_3DR,
.message__title_3jF {
    position: relative;
    padding: 0;
    z-index: 0
}

.message__text_3DR {
    margin: 0 0 42px;
    font-size: 14px;
    text-align: center;
    line-height: 1.43;
    font-family: var(--ui-text-font-family)
}

.message__btn_1xl {
    width: 210px;
    margin: 0 auto;
    padding: 0;
    border: 0;
    overflow: hidden;
    box-shadow: none;
    border-radius: var(--ui-btn-md-border-radius);
    z-index: 1;
    text-transform: uppercase;
    text-align: center;
    background-color: var(--ui-brand-color, #50e3c2)
}

.message__btn_1xl,
.message__inner-text_OOT {
    position: relative;
    line-height: 36px;
    font-family: var(--ui-block-title-font-family);
    font-size: 12px;
    color: #fff
}

.message__inner-text_OOT {
    border-radius: var(--ui-btn-md-border-radius)
}

.theme-dark_3FJ .modal__content_1zJ {
    background-color: var(--ui-dark-shade-color)
}

```

```

.theme-dark_3FJ .message__text_3DR,
.theme-dark_3FJ .message__title_3jF {
    color: var(--ui-light-shade-color, #fff)
}

.theme-dark_3FJ .message__close-btn_1fi:after,
.theme-dark_3FJ .message__close-btn_1fi:before {
    background-color: var(--ui-light-shade-color, #fff)
}

.theme-light_1j6 .modal__content_1zJ {
    background-color: var(--ui-light-shade-color)
}

.theme-light_1j6 .message__text_3DR,
.theme-light_1j6 .message__title_3jF {
    color: var(--ui-dark-shade-color, #212736)
}

.theme-light_1j6 .message__close-btn_1fi:after,
.theme-light_1j6 .message__close-btn_1fi:before {
    background-color: var(--ui-dark-shade-color, #212736)
}

.preview_1ua {
    position: relative;
    height: 420px;
    transform: translateZ(0)
}

.preview_1ua .modal__content_1zJ {
    animation: none
}

.preview__button_12C {
    position: absolute;
    top: 8px;
    right: 8px;
    padding: 5px 8px;
    z-index: 1
}

@keyframes fadeIn_2AA {
    0% {
        transform: scale(0);
        opacity: 0
    }

    to {
        transform: scale(1);
        opacity: 1
    }
}

.ui-contactForm_1f9 {
    font-size: var(--ui-text-font-size);
    line-height: var(--ui-text-line-height);
    font-weight: var(--ui-text-font-weight);
    font-family: var(--ui-text-font-family);
    color: var(--ui-text-color)
}

```

```

.ui-contactForm--light_18j {
    color: var(--ui-text-color-light)
}

.ui-input--size-sm_30Y {
    padding: var(--ui-btn-sm-padding);
    font-size: var(--ui-btn-sm-font-size);
    border-radius: var(--ui-btn-sm-border-radius);
    color: inherit;
    line-height: var(--ui-btn-sm-line-height)
}

.ui-input--size-md_1wC {
    padding: var(--ui-btn-md-padding);
    font-size: var(--ui-btn-md-font-size);
    border-radius: var(--ui-btn-md-border-radius);
    color: inherit;
    line-height: var(--ui-btn-md-line-height)
}

.ui-input--size-lg_3Ef {
    padding: var(--ui-btn-lg-padding);
    font-size: var(--ui-btn-lg-font-size);
    border-radius: var(--ui-btn-lg-border-radius);
    color: inherit;
    line-height: var(--ui-btn-lg-line-height)
}

.input_GmU:focus::-webkit-input-placeholder,
.textarea_1Lw:focus::-webkit-input-placeholder {
    opacity: 0
}

.input_GmU:focus::-moz-placeholder,
.textarea_1Lw:focus::-moz-placeholder {
    opacity: 0
}

.input_GmU:focus:-ms-input-placeholder,
.textarea_1Lw:focus:-ms-input-placeholder {
    opacity: 0
}

.input_GmU:focus:-moz-placeholder,
.textarea_1Lw:focus:-moz-placeholder {
    opacity: 0
}

.input_GmU::-webkit-input-placeholder,
.textarea_1Lw::-webkit-input-placeholder {
    -webkit-transition: opacity .2s ease-out;
    transition: opacity .2s ease-out
}

.input_GmU::-moz-placeholder,
.textarea_1Lw::-moz-placeholder {
    -moz-transition: opacity .2s ease-out;
    transition: opacity .2s ease-out
}

```

```
.input_GmU:-ms-input-placeholder,
.textarea_1Lw:-ms-input-placeholder {
  -ms-transition: opacity .2s ease-out;
  transition: opacity .2s ease-out
}
```

## Main.js

```
!(function (e, t) {
  if ("object" === typeof exports && "object" === typeof module)
    module.exports = t();
  else if ("function" === typeof define && define.amd) define([], t);
  else {
    var n = t();
    for (var r in n) ("object" === typeof exports ? exports : e)[r] = n[r];
  }
})(window, function () {
  return (window.webpackJsonp = window.webpackJsonp || []).push([
    [1],
    {
      "../client/editor-next/src/components/block-components/accordion/ssr/style.css":
        function (e, t, n) {
          e.exports = {
            accordion: "accordion_2Bj",
            "accordion--decorator": "accordion--decorator_3DO",
            accordionDecorator: "accordion--decorator_3DO",
            accordion__header: "accordion__header_3a4",
            accordion__heading: "accordion__heading_2ZY",
            accordion__content: "accordion__content_3kU",
            accordion__icon: "accordion__icon_1GF",
            "accordion__icon--plus": "accordion__icon--plus_1m-",
            accordion__iconPlus: "accordion__icon--plus_1m-",
            "accordion__icon--arrow": "accordion__icon--arrow_2-j",
            accordion__iconArrow: "accordion__icon--arrow_2-j",
            "accordion--active": "accordion--active_2ER",
            accordionActive: "accordion--active_2ER",
            "background--dark": "background--dark_3oK",
            backgroundDark: "background--dark_3oK",
            "accordion--first": "accordion--first_xWW",
            accordionFirst: "accordion--first_xWW",
            "background--light": "background--light_3W_",
            backgroundLight: "background--light_3W_",
            "align--left": "align--left_3S7",
            alignLeft: "align--left_3S7",
          };
        },
      "../client/editor-next/src/components/block-components/accordion/view/index.js":
        function (e, t, n) {
          "use strict";
          Object.defineProperty(t, "__esModule", { value: !0 });
          (t.init = void 0);
          var r = n(
            "../client/editor-next/src/components/block-components/collection/view/slider/utlis.js"
          );
          function o(e, t) {
            e.classList.add("w-disable-offset"),
              (e.dataset.state = "closed"),
              t.setAttribute("style", "height: 0");
          }
          n(

```

```

"/client/editor-next/src/components/block-components/accordion/ssr/style.css"
);
var i = function (e) {
  var t = e.querySelectorAll("[data-component=accordion]");
  !(function (e) {
    for (
      var t = function () {
        var t = e[n];
        t.onclick = function (e) {
          for (
            var n,
                r = e.target,
                i = function () {
                  if (r.hasAttribute("data-accordion-content"))
                    return { v: void 0 };
                  if (r.hasAttribute("data-accordion-item")) {
                    var e = r.dataset.accordionItem;
                    t.querySelectorAll(
                      "[data-accordion-item]"
                    ).forEach(function (t) {
                      var n = t.querySelector(
                        "[data-accordion-inner]"
                      ),
                          r = t.querySelector(
                            "[data-accordion-content]"
                          ),
                          i = n.clientHeight;
                      r.dataset.accordionContent === e
                        ? "opened" === t.dataset.state
                        ? o(t, r)
                        : (function (e, t, n) {
                            e.classList.remove("w-disable-offset"),
                              (e.dataset.state = "opened"),
                              t.setAttribute(
                                "style",
                                "height: ".concat(n, "px")
                              ),
                              (t.dataset.noAnimated = "false");
                            })(t, r, i)
                          : o(t, r);
                    });
                  }
                  r = r.parentNode;
                };
            r !== t;

          )
            if ((n = i())) return n.v;
          };
        },
        n = 0;
        n < e.length;
        ++n
      )
        t();
    })(t),
    (function (e) {
      document.addEventListener(
        "slider.onBeforeChange",
        function (t) {
          var n = t.detail,

```

```

    r = (void 0 === n ? {} : n).currentSlide;
    if (r instanceof HTMLElement) {
      var i = r.parentNode;
      e.forEach(function (e) {
        i.contains(e) &&
          e
            .querySelectorAll("[data-accordion-item]")
            .forEach(function (e) {
              var t = e.querySelector(
                "[data-accordion-content]"
              );
              (t.dataset.noAnimated = "true"), o(e, t);
            });
      });
    }
  });
  })(t);
};
(t.init = i), (0, r.addSliderMountListener)(i);
},

```

## player.js

```

(window.webpackJsonp = window.webpackJsonp || []).push([
  [13, 15],
  {
    "../client/editor-next/src/components/block-components/player/index.js":
    function (e, t, n) {
      "use strict";
      Object.defineProperty(t, "__esModule", { value: !0 });
      (t.default = void 0);
      var r = (function (e) {
        return e && e.__esModule ? e : { default: e };
      })(n("../client/editor-next/src/components/player/index.js")).default;
      t.default = r;
    },
    "../client/editor-next/src/components/player/enhancer.js": function (
      e,
      t,
      n
    ) {
      "use strict";
      (function (e, r, o) {
        Object.defineProperty(t, "__esModule", { value: !0 });
        (t.playerEnhancer = void 0);
        var i = (function (e) {
          return e && e.__esModule ? e : { default: e };
        })(n("../client/node_modules/can-use-dom/index.js")),
          a = n("../client/editor-next/src/components/player/utils.js");
        function l(e) {
          "@babel/helpers - typeof";
          return l =
            "function" == typeof Symbol && "symbol" == typeof Symbol.iterator
              ? function (e) {
                  return typeof e;
                }
              : function (e) {

```

```

    return e &&
      "function" === typeof Symbol &&
      e.constructor === Symbol &&
      e !== Symbol.prototype
      ? "symbol"
      : typeof e;
  })(e);
}
function u() {
  return (u = Object.assign
    ? Object.assign.bind()
    : function (e) {
      for (var t = 1; t < arguments.length; t++) {
        var n = arguments[t];
        for (var r in n)
          Object.prototype.hasOwnProperty.call(n, r) && (e[r] = n[r]);
      }
      return e;
    }).apply(this, arguments);
}
function c(e, t) {
  for (var n = 0; n < t.length; n++) {
    var r = t[n];
    (r.enumerable = r.enumerable || !1),
    (r.configurable = !0),
    "value" in r && (r.writable = !0),
    Object.defineProperty(e, h(r.key), r);
  }
}
function s(e, t) {
  return (s = Object.setPrototypeOf
    ? Object.setPrototypeOf.bind()
    : function (e, t) {
      return (e.__proto__ = t), e;
    })(e, t);
}
function f(e) {
  var t = (function () {
    if ("undefined" === typeof Reflect || !Reflect.construct) return !1;
    if (Reflect.construct.sham) return !1;
    if ("function" !== typeof Proxy) return !0;
    try {
      return (
        Boolean.prototype.valueOf.call(
          Reflect.construct(Boolean, [], function () {}
        )
      ),
      !0
    );
  } catch (e) {
    return !1;
  }
})();
return function () {
  var n,
      r = p(e);
  if (t) {
    var o = p(this).constructor;
    n = Reflect.construct(r, arguments, o);
  } else n = r.apply(this, arguments);
  return (function (e, t) {
    if (t && ("object" === l(t) || "function" === typeof t)) return t;
  })(n, r);
}

```

```

    if (void 0 !== t)
      throw new TypeError(
        "Derived constructors may only return object or undefined"
      );
    return d(e);
  })(this, n);
};
}
function d(e) {
  if (void 0 === e)
    throw new ReferenceError(
      "this hasn't been initialised - super() hasn't been called"
    );
  return e;
}
function p(e) {
  return (p = Object.setPrototypeOf
    ? Object.getPrototypeOf.bind()
    : function (e) {
        return e.__proto__ || Object.getPrototypeOf(e);
      })(e);
}
function y(e, t, n) {
  return (
    (t = h(t)) in e
    ? Object.defineProperty(e, t, {
        value: n,
        enumerable: !0,
        configurable: !0,
        writable: !0,
      })
    : (e[t] = n),
    e
  );
}
function h(e) {
  var t = (function (e, t) {
    if ("object" !== l(e) || null === e) return e;
    var n = e[Symbol.toPrimitive];
    if (void 0 !== n) {
      var r = n.call(e, t || "default");
      if ("object" !== l(r)) return r;
      throw new TypeError(
        "@@toPrimitive must return a primitive value."
      );
    }
  })(e, "string");
  return "string" === t ? String : Number)(e);
}
return "symbol" === l(t) ? t : String(t);
}
t.playerEnhancer = function (t) {
  var n;
  return (
    y(
      (n = (function (n) {
        !(function (e, t) {
          if ("function" !== typeof t && null !== t)
            throw new TypeError(
              "Super expression must either be null or a function"
            );
          (e.prototype = Object.create(t && t.prototype, {

```



```

    constructor: { value: e, writable: !0, configurable: !0 },
  )),
  Object.defineProperty(e, "prototype", { writable: !1 }),
  t && s(e, t);
})(l, e.Component);
var o = f(l);
function l() {
  var e;
  !(function (e, t) {
    if (!(e instanceof t))
      throw new TypeError("Cannot call a class as a function");
  })(this, l);
  for (
    var t = arguments.length, n = new Array(t), u = 0;
    u < t;
    u++
  )
    n[u] = arguments[u];
  return (
    y(d((e = o.call.apply(o, [this].concat(n)))), "state", {
      videoData: {},
      playerName: "",
    })),
    y(d(e, "fetchData", function () {
      return new Promise(function (t, n) {
        i.default || n();
        var o = e.props,
            l = o.value,
            u = l.url,
            c = l.hd,
            s = o.background,
            f = o.value,
            d = e.state.playerName,
            p = r.get("thumbnail", f);
        if (
          (d || t({ videoData: {} })), r.has("thumbnail", f)
        ) {
          if (r.pipe(r.get("thumbnail"), r.isNil)(f)) return;
          t({ videoData: { thumbnail: p } });
        }
        (0, a.videoDataRequest)({
          playerName: d,
          url: u,
          hd: c || s,
          background: s,
        })
          .then(function (e) {
            if (e.thumbnail) {
              var n = new Image();
              (n.src = e.thumbnail),
                (n.onload = function () {
                  return t({ videoData: e });
                });
            } else s && t({ videoData: e });
          })
          .catch(function (e) {
            e instanceof Error && t({ videoData: {} });
          });
      });
    })),
    y(

```

```

d(e),
"updateData",
r.debounce(300, function () {
  var t = e.props,
      n = t.value,
      o = n.url,
      i = n.hd,
      a = n.thumbnail,
      l = t.background;
  e.fetchData().then(function (t) {
    var n = e.props.value;
    r.isEqual(
      { thumbnail: a, url: o, hd: i, background: l },
      {
        thumbnail: n.thumbnail,
        url: n.url,
        hd: n.hd,
        background: e.props.background,
      }
    ) && e.setState(t);
  });
});
),
e
);
}
return (
(function (e, t, n) {
  t && c(e.prototype, t),
  n && c(e, n),
  Object.defineProperty(e, "prototype", { writable: !1 });
})(l, [
  {
    key: "componentDidMount",
    value: function () {
      this.props.inViewport && this.updateData();
    },
  },
  {
    key: "componentDidUpdate",
    value: function (e) {
      var t = this.props,
          n = t.value,
          r = n.url,
          o = n.hd,
          i = n.thumbnail,
          a = t.inViewport && !e.inViewport,
          l =
            r !== e.value.url ||
            o !== e.value.hd ||
            i !== e.value.thumbnail;
      (a || l) && this.updateData();
    },
  },
],

```

**ВІДГУК**

**Керівника економічного розділу**

**на кваліфікаційну роботу бакалавра на тему:**

**«Розробка стрімінгової веб-платформи для перегляду японської анімації»**

**студентки групи 122-21зск-1 Юрової Анастасії Кирилівни**

**Керівник економічного розділу  
доц. каф. ПЕП та ПУ, к.е.н**

**Л.В. Касьяненко**

## ПЕРЕЛІК ДОКУМЕНТІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна_робота_Юрова А.К.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна_робота_Юрова А.К.pdf	Пояснювальна записка до кваліфікаційної роботи. Документ PDF.
Програма	
Website.zip	Архів. Містить коди програми.
Презентація	
Презентація_Юрова.ppt	Презентація кваліфікаційної роботи.