

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
комп'ютеризованих систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: «Програмне забезпечення для редагування відеофайлів»

Виконав студентка IV курсу, групи ІТ-84в
(шифр групи)

Гюржи Євгенія Олегівна
(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник доцент, к.т.н., Олійник Ю.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант
з графічної
документації

Старший викладач, Вітковська І.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент доцент кафедри ІСТ, к.т.н., Ткач М.М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Київ - 2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
комп'ютеризованих систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Едуард ЖАРІКОВ
(ім'я прізвище)

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Гюржи Євгенії Олегівні

(прізвище, ім'я, по батькові)

1. Тема проєкту «Програмне забезпечення для редагування відеофайлів»

керівник проєкту Олійник Юрій Олександрович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____»квітня 2023 р. №____

2. Термін подання студентом проєкту « 19 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: загальні положення, змістовний опис і аналіз предметної області, аналіз існуючих технологій та успішних ІТ-проєктів, аналіз вимог до програмного забезпечення, постановка задачі.

2) Моделювання та конструювання програмного забезпечення: архітектура програмного забезпечення, конструювання програмного забезпечення, аналіз безпеки даних.

3) Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ, опис процесів тестування, опис контрольного приклад.

4) Впровадження та супровід програмного забезпечення: розгортання програмного забезпечення, підтримка програмного забезпечення.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань.

2) Схема структурна компонентів програмного забезпечення.

3) Схема креслення вигляду екранних форм.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «10» березня 2023 року _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2023	
2	Аналіз існуючих методів розв'язання задачі	23.03.2023	
3	Постановка та формалізація задачі	23.03.2023	
4	Розробка інформаційного забезпечення	29.03.2023	
5	Алгоритмізація задачі	29.03.2023	
6	Обґрунтування вибору використаних технічних засобів	03.04.2023	
7	Розробка програмного забезпечення	03.04.2023	
8	Налагодження програми	25.04.2023	
9	Виконання графічних документів	01.05.2023	
10	Оформлення пояснювальної записки	01.05.2023	
11	Подання ДП на попередній захист	06.06.2023	
12	Подання ДП рецензенту	12.06.2023	
13	Подання ДП на основний захист	19.06.2023	

Студент

(підпис)

Євгенія ГЮРЖИ

(ініціали, прізвище)

Керівник

(підпис)

Юрій ОЛІЙНИК

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, 1 додатку, містить 20 таблиць та 17 рисунків та 9 джерел – загалом 75 сторінок.

На сьогоднішній день такі відеохостинги як YouTube набувають все більшої популярності, та з кожним роком все більше людей прагнуть стати блогерами або використовують відеохостинги для просування свого бізнесу. Проте для того, щоб контент приваблював користувачів, він повинен відповідати деяким стандартам. Відеоконтент повинен, якнайменш, сприйматися легко. Саме для створення відповідності таким стандартам і були створені програми для редагування відео. Більшість з них мають одну особливість, яку в деяких випадках можна сприйняти за недолік – уся обробка відбувається на клієнтському комп'ютері, що зменшує період життя батареї та потребує від комп'ютера певного рівня потужності.

Дипломний проєкт присвячений темі обробки та редагування відеофайлів.

Мета роботи: зниження навантаження на пристрій користувача під час обробки відеофайлів за рахунок використання серверних технологій. Об'єкт дослідження: формати відео та аудіо.

Предмет дослідження: способи обробки відео та аудіофайлів.

У першому розділі розглянуто основну інформацію з досліджуваної теми.

Другий розділ присвячений розробці архітектури програмного забезпечення.

Третій розділ містить інформацію про контроль якості, наведені приклади тестів.

Четвертий розділ присвячено впровадженню та супроводу програмного забезпечення.

КЛЮЧОВІ СЛОВА: ВЕБ, СЕРВЕР, CLOUD, ОБРОБКА ВІДЕО.

ABSTRACT

The explanatory note of the diploma project consists of four sections, 1 Appendix, contains 20 tables and 17 figures - a total of 75 pages.

Video hosting services such as YouTube are becoming more and more popular, and more and more people are becoming bloggers or using video hosting services to promote their business. However, in order for your content to attract users, it must meet certain standards. At the very least, it should be easy to understand. It is to create compliance with such standards and created video editing programs. Most of them have one feature that in some cases can be considered a disadvantage - all processing takes place on the client computer, which reduces battery life and requires a certain level of power from the computer.

This project is dedicated to the topic of video processing.

Purpose: development of a software product for video editing on the server

Object of research: video and audio formats.

Subject of research: methods of processing video and audio files.

The first section discusses the basic information on the research topic.

The second section is devoted to the development of program architecture.

The third section contains information on quality control, examples of tests.

The fourth section is devoted to the implementation and maintenance of the software.

KEY WORDS: WEB, SERVER, CLOUD, VIDEO PROCESSING.

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

Програмне забезпечення для редагування відеофайлів

Технічне завдання

КП.ІТ-8411.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Юрій ОЛІЙНИК

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Євгенія ГЮРЖИ

Київ – 2023

ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1	Вимоги до функціональних характеристик	6
4.1.1	Користувальницького інтерфейсу	6
4.1.2	Для користувача:	10
4.1.3	Додаткові вимоги:	11
4.2	Вимоги до надійності:.....	11
4.3	Умови експлуатації:	11
4.3.1	Вид обслуговування	11
4.3.2	Обслуговуючий персонал.....	11
4.4	Вимоги до складу і параметрів технічних засобів	11
4.5	Вимоги до інформаційної та програмної сумісності	12
4.5.1	Вимоги до вхідних даних	12
4.5.2	Вимоги до вихідних даних	12
4.5.3	Вимоги до мови розробки.....	12
4.5.4	Вимоги до середовища розробки.....	12
4.5.5	Вимоги до представленню вихідних кодів	12
4.6	Вимоги до маркування та пакування.....	13
4.7	Вимоги до транспортування та зберігання	13
4.8	Спеціальні вимоги	13
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	14
5.1	Попередній склад програмної документації.....	14
5.2	Спеціальні вимоги до програмної документації	14
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	15
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	16

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: «Програмне забезпечення для редагування відеофайлів».

Галузь застосування:

Наведене технічне завдання поширюється на розробку веб-сервісу для редагування відеофайлів [045440], котрий використовується для:

- обрізання відео;
- склеювання відео;
- додавання аудіо;
- накладання відео на відео, відео на фото і навпаки.
- додавання фільтрів;
- виконання базових ефекти.

Веб-сервіс з редагування відеофайлів призначений для монтажерів — людей, які займаються професійною обробкою відео для різного типу замовників (блогери, підприємці, музичні виконавці та інші).

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки програмного забезпечення для редагування відеофайлів є завдання на дипломне проєктування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для редагування відеофайлів з перенесенням максимальної кількості обчислень на сервер, щоб зменшити навантаження на клієнтський комп'ютер.

Метою розробки є зниження навантаження на пристрій користувача під час обробки відеофайлів за рахунок використання серверних технологій.

					КПІ.ІТ-84В11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1 Користувальницького інтерфейсу

- зручність використання;
- обрізка та склеювання відео;
- додавання ефектів;
- додавання аудіо;
- зміна розміру відео та фото;
- переміщення відео в часі;
- рендерінг відео;
- завантаження та збереження результату;
- бібліотека доданих фрагментів.

На вкладці «Projects» користувач може переглядати список існуючих проєктів або створити новий за допомогою форми для створення нового проєкту (Рисунок 4.2). Користувач може обрати, з яким проєктом він хоче працювати, відзначивши потрібний проєкт за допомогою галочки. Також можна переглядати час останніх змін по кожному з проєктів.

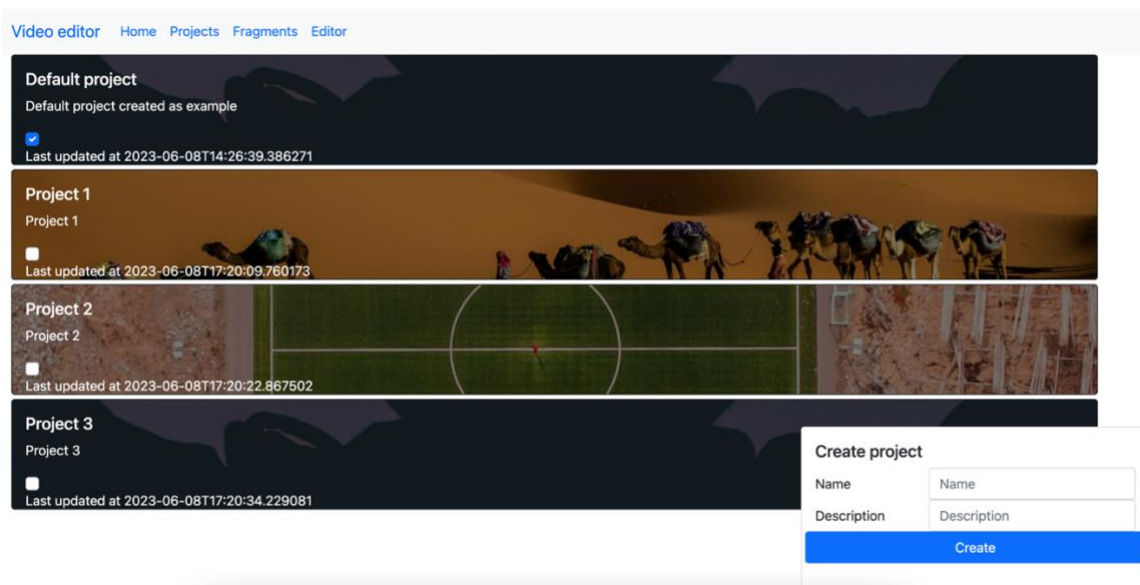


Рисунок 4.1 – Вкладка «Projects»

Створити проєкт користувач може, написавши назву проєкту та коротко описавши його. Коли користувач заповнив форму, він натискає кнопку «Create».

Рисунок 4.2 – Форма для створення нового проєкту

Створивши та обравши проєкт, з яким хоче працювати, користувач переходить на вкладку «Fragments», де він може переглядати бібліотеку фрагментів відео, а також додавати нові фрагменти, використовуючи форму для додавання фрагментів (Рисунок 4.4) та видаляти з сервера вже існуючі, використовуючи для цього кнопку «Delete». Користувач може обрати, з якими фрагментами він буде працювати у проєкті, натиснувши кнопку «Use in projeect».

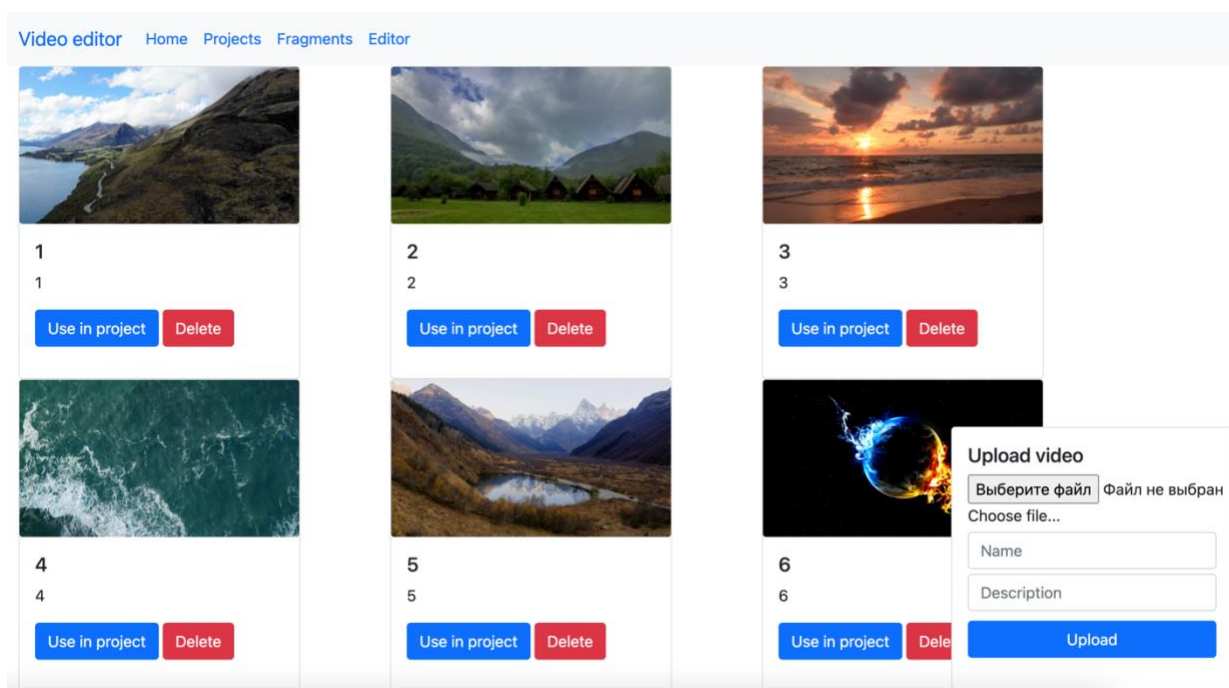


Рисунок 4.3 – Вкладка «Fragments»

Для того, щоб додати нові фрагменти, користувач використовує форму для додавання нових файлів у бібліотеку редактора. Користувач натискає кнопку «Обрати файл», обирає його з бібліотеки власного пристрою, потім вводить ім'я та опис файлу та натискає кнопку «Upload».

Upload video

Выберите файл

Файл не выбран

Choose file...

Name

Description

Upload

Рисунок 4.4 – Форма для додавання нових фрагментів

На вкладці «Editor» користувач має змогу переглядати робочу область, таймлайн з доданими фрагментами та панель інструментів (Рисунок 4.6).

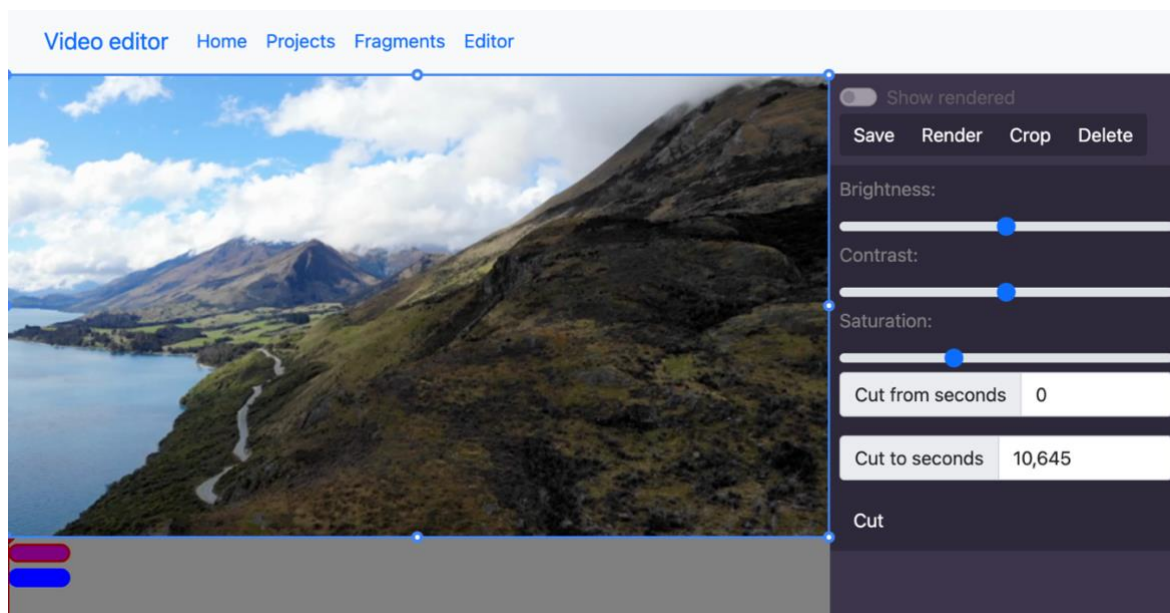


Рисунок 4.5 – Вкладка «Editor»

На панелі інструментів користувач може регулювати за допомогою повзунків такі ефекти, як яскравість відео (Brightness), контрастність відео

(Contrast) та насиченість відео (Saturation). На цій панелі також є кнопка «Cut», яка дозволяє обрізати обраний на таймлайні фрагмент в тому місці, де був встановлений бігунок. Також є другий спосіб обрізання фрагментів відео – можна визначити межі відео, вписавши, з якої по яку секунду відео повинно бути обрізано («Cut from seconds» та «Cut to seconds»). Тут користувач може зберігати свої зміни, які були застосовані до фрагментів (Save), також обрізати зображення відео (Crop), видаляти непотрібні фрагменти з таймлайну, обравши фрагмент та натиснувши кнопку «Delete». Для рендерингу фінального відео користувач натискає кнопку «Render». Переглянути отриманий результат користувач може, натиснувши перемикач «Show rendered».

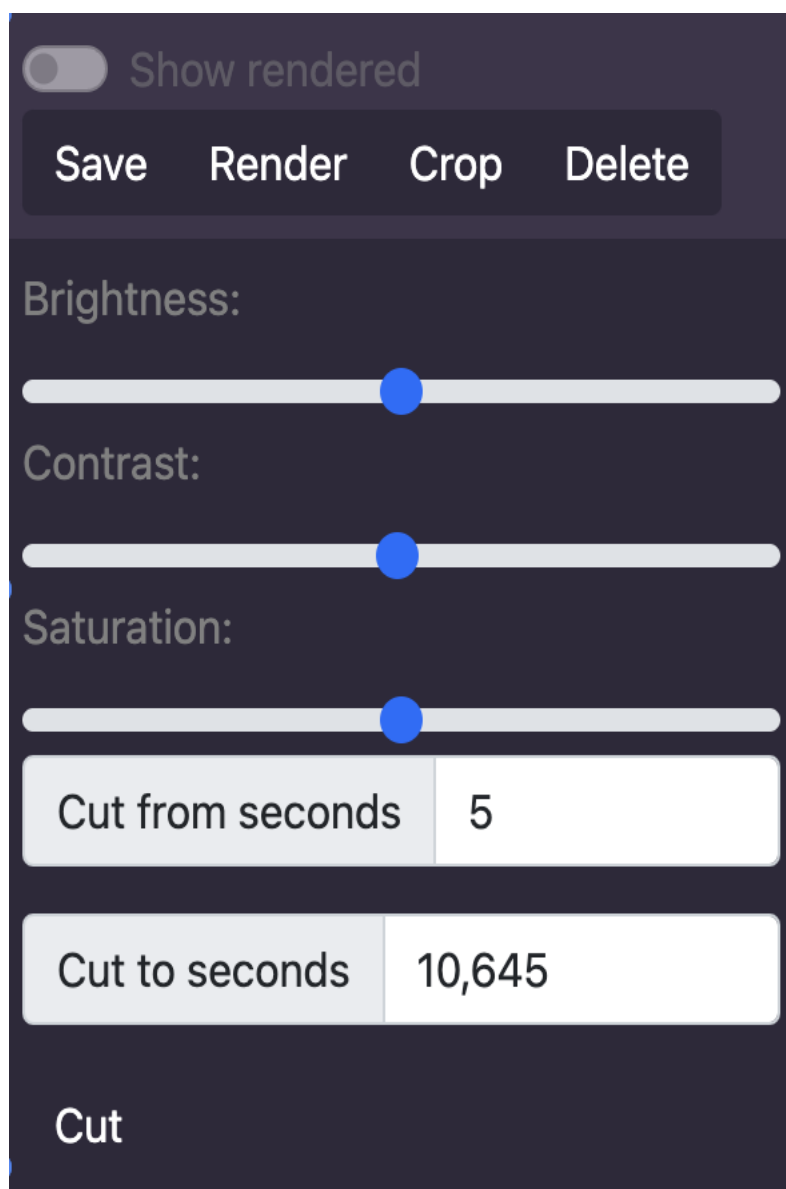


Рисунок 4.6 – Панель інструментів

Коли рендеринг завершено, користувач отримує повідомлення, яке говорить про те, що рендеринг успішно завершено.



Рисунок 4.7 – Вікно повідомлення про успішність рендеринга

Натиснувши на панелі інструментів повзунок «Show rendered», користувач переглядає отриманий результат. У плеєрі перегляду результату користувач може натиснути три крапки, які розташовані у правому нижньому куті та завантажити відео на власний пристрій.

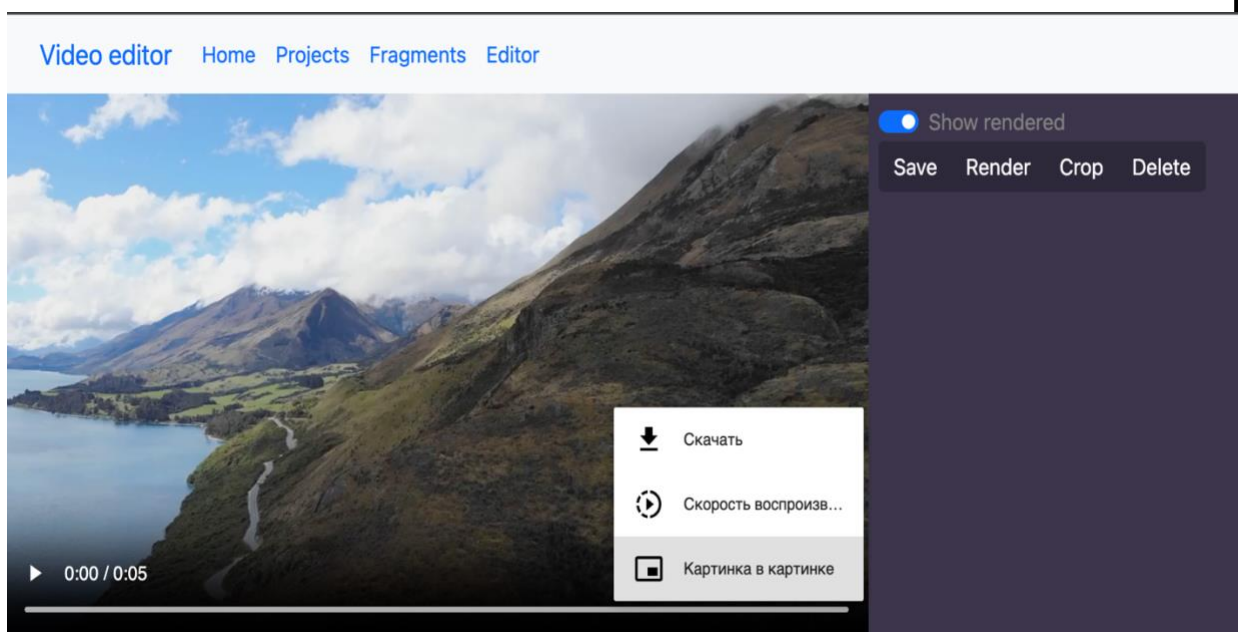


Рисунок 4.8 – Плеєр перегляду відредагованого відео

4.1.2 Для користувача:

- обрізання відео;
- склеювання відео;
- додавання фрагментів відео;
- додавання аудіо;
- зміна розміру відео та фото;

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

- переміщення відео;
- фінальний рендеринг;
- виділення кількох елементів (відео, аудіо) для подальшого застосування ефектів;

- застосування ефектів для відео, картинки чи аудіо;
- переміщення початку на кінець відео в часі;
- перегляд процесу рендерингу відео;
- завантаження результату;
- перегляд бібліотеки відео онлайн;
- завантаження відео з комп'ютера;
- завантаження відео зі сторонніх сервісів;
- реєстрація.

4.1.3 Додаткові вимоги:

- простота інтерфейсу;
- перенесення максимальної кількості обчислень на сервер;
- транслювання результатів редагування відео в режимі реального часу.

4.2 Вимоги до надійності:

- передбачити контроль введення інформації;
- передбачити захист від некоректних дій користувача;
- забезпечити цілісність інформації в базі даних;
- відновлення останніх збережених даних при збоях системи.

4.3 Умови експлуатації:

- наявність стабільного Інтернету;
- наявність ПК або ноутбуку.

4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

4.4 Вимоги до складу і параметрів технічних засобів

Програмне забезпечення повинно функціонувати на операційних

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		11

системах, для яких реалізовано віртуальну машину Java та пакет NodeJS.

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Pentium 2.2 MHz;
- об'єм ОЗП: 128 Мб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт.

Рекомендована конфігурація технічних засобів:

- тип процесору: Intel Core i7;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт.

4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно працювати під управлінням найпопулярніших операційних систем, як Windows, Linux або Mac.

4.5.1 Вимоги до вхідних даних

Вхідні дані повинні бути представлені в наступному форматі: MP4, MP3, RAW, GPEG, PNG.

4.5.2 Вимоги до вихідних даних

Результати повинні бути представлені в наступному форматі: MP4. Програмне забезпечення повинно давати доступ до усіх можливостей, використовуючи мінімальну кількість ресурсів процесора та відеокarti для продовження періоду автономної роботи комп'ютера. В той же час, результати роботи повинні транслюватися на таймлайні в режимі реального часу з сервера на клієнт, щоб монтажер мав змогу без перешкод та помилок з боку програмного забезпечення створювати якісний відеоряд.

4.5.3 Вимоги до мови розробки

Розробку виконати на мові програмування Java та JavaScript.

4.5.4 Вимоги до середовища розробки

Розробку виконати за допомогою середовища IntelliJ IDEA.

4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у текстовому вигляді за допомогою мови Java.

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не пред'являються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4.8 Спеціальні вимоги

Надання можливості видаляти відео з сервера для зменшення плати за зберігання відео.

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		13

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача;
- графічні матеріали.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна процесу створення відео;
- загальна модель архітектури бекенду;
- загальна модель архітектури фронтенду;
- діаграма компонентів програмного забезпечення;
- креслення вигляду екранних форм.

5.2 Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	10.03	
2.	Розробка технічного завдання	23.03	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	19.03	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	29.03	Схема структурна програмного забезпечення та специфікація компонентів (діаграма компонентів, схема алгоритму)
5.	Програмна реалізація програмного забезпечення	03.04	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	25.04	Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту	01.05	Пояснювальна записка
8.	Розробка матеріалів графічної частини проекту	01.05	Графічний матеріал проекту
9.	Оформлення технічної документації проекту	03.06	Технічна документація

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІТ-84в11.045440.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		16

**Пояснювальна записка
до дипломного проєкту**

на тему: «Програмне забезпечення для редагування відеофайлів»

КПІ.ІТ-8411в.045440.02.81

Київ – 2023

ЗМІСТ

ВСТУП 7

1	АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	9
1.1	Загальні положення.....	9
1.2	Змістовний опис і аналіз предметної області	9
1.3	Аналіз існуючих технологій та успішних ІТ-проектів	11
1.3.1	Аналіз відомих алгоритмічних та технічних рішень.....	11
1.3.2	Аналіз допоміжних програмних засобів та засобів розробки	12
1.3.3	Аналіз відомих програмних продуктів	14
1.4	Аналіз вимог до програмного забезпечення	15
1.4.1	Розроблення функціональних вимог	21
1.4.2	Нефункціональні вимоги.....	25
1.5	Постановка задачі.....	26
	Висновки по розділу	26
2	МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.1	Моделювання та аналіз програмного забезпечення	28
2.2	Архітектура програмного забезпечення.....	33
2.3	Конструювання програмного забезпечення	36
2.4	Аналіз безпеки даних	47
	Висновки по розділу	48
3	АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	49
3.1	Аналіз якості ПЗ	49
3.2	Опис процесів тестування	49
3.3	Опис контрольного прикладу.....	58
	Висновки по розділу	63
4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .	64
4.1	Розгортання програмного забезпечення	64
4.2	Підтримка програмного забезпечення	64

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

Висновки по розділу.....	71
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТОК А ЗВІТ ПОДІБНОСТІ.....	75

					КПІ.ІТ-84В11.045440.02.81	Арк.
						3
Змін.	Арк.	№ докум.	Підп.	Дата.		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

MVC	– розшифровується як модель-уявлення-контролер (від англ. model-view-controller). Це спосіб організації коду, що передбачає виділення блоків, відповідальних за рішення різних завдань. Один блок відповідає за дані програми, інший відповідає за зовнішній вигляд, а третій контролює роботу програми.
EC2	– розшифровується як Elastic Compute Cloud. Це хмарний сервіс, що надає віртуальні сервери (Amazon EC2 Instance). За його допомогою для розробників спрощується проведення обчислень у хмарі в масштабі всього Інтернету.
S3	– розшифровується як Simple Storage Service. Це протокол передачі, розроблений компанією Amazon. Також – об'єктне сховище. Його унікальність полягає у зберіганні величезного обсягу даних у вихідному форматі без ієрархії та розбивки на окремі каталоги. У сховища S3 немає обмежень масштабування.
ALB	– розшифровується як Application Load Balancer. Це балансувальники навантаження другого покоління від комерційної публічної платформи хмарної Amazon Web Services (AWS) [7].
AWS	– розшифровується як Amazon Web Services. Це

набір хмарних сервісів компанії Amazon. На єдиній платформі користувачі можуть замовити обчислювальні ресурси, сховище, інфраструктуру, послуги з готовими для використання інструментами.

Рендеринг

- це процес створення фінального зображення або послідовності із зображень на основі двовимірних або тривимірних даних. Цей процес відбувається з використанням комп'ютерних програм і часто супроводжується важкими технічними обчисленнями, які лягають на обчислювальні потужності комп'ютера або окремі його частини.

AVI

- це мультимедійний контейнер, що має величезну популярність серед користувачів персональних комп'ютерів і ноутбуків по всьому світу. Найчастіше, AVI файли використовують як відео формат.

MPEG

ASF

- це формат файлів, що розширюється, призначений в першу чергу для зберігання та відтворення синхронізованих потоків цифрових мультимедіа та їх передачі по мережах.

BSD

- UNIX-подібна ОС, яка була створена для обміну досвідом між навчальними закладами.

Кросплатформенність

- це здатність працювати більше, ніж на одній апаратній платформі або операційній

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

системі.

Хостинг

- послуга з розміщення контенту на сервері, який зазвичай має безперервний доступ до мережі.

					КПІ.ІТ-84В11.045440.02.81	Арк.
						6
Змін.	Арк.	№ докум.	Підп.	Дата.		

ВСТУП

В наші часи все більшої популярності набувають такі відеохостинги як YouTube, Vimeo та TikTok. З ростом популярності таких платформ, збільшується і вплив відеоконтенту на потенційного користувача. Деякі люди вирішують стати блогерами, а дехто успішно використовує ці хостинги для просування власних великих та маленьких бізнесів, адже в сучасному світі ми вже не можемо уявити маркетинг без створення контенту для соціальних мереж. І саме для створення якісного контенту необхідні навички користування програмами для відеомонтажу. Проаналізувавши ринок, можна прослідкувати, що усі програми для монтажу об'єднують одне: відносно високі вимоги до техніки, на якій вони запускаються. І навіть якщо сам процес обробки відео можливо виконати на певному комп'ютері, це не означає, що цей процес буде зручним, адже найпоширеніша проблема, яка виникає у процесі розробки відеопроєкту – це те, що користувач не може переглядати результати своєї роботи в режимі реального часу. Для найбільшої продуктивності комп'ютер повинен мати потужну відеокарту, яка, ймовірно, на сьогоднішній день буде коштувати більше, ніж будь-який інший пристрій.

Проте хороша техніка для відеомонтажу все ще доступна далеко не кожному. Купити новенький макбук чи навіть звичайний ноутбук з хорошими характеристиками для редагування відео зараз можуть далеко не всі. Попри це, швидкий і надійний інтернет доступний вже в багатьох регіонах України та світу. Тому хорошим рішенням було б перенесення навантаження на бік сервера. У такому випадку бажана швидкість інтернету буде приблизно дорівнювати необхідній для перегляду відео на YouTube, а навантаження на клієнтський гаджет значно впаде. Можливі також адаптації до швидкості інтернету у вигляді зменшення кількості кадрів, що передається, чи зменшення розширення відео.

Для того, щоб отримати мінімальне розуміння того, чи буде подібний сервіс користуватися популярністю, можна звернути увагу на той факт, що вже існують подібні сервіси для ігор (Cloud gaming services). Ці сервіси займаються

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

приблизно тим же, що і онлайн відеоредактор, а саме – здаванням в оренду віртуальних машин. Більше того, перевагою з точки зору бізнесу для відеохостингу може служити той факт, що під час гри навантаження на сервер є непередбачуваним, через що доводиться виділяти віртуальну машину, можливості якої, скоріше за все, використовуватимуться не повністю для кожного користувача. І хоча cloud провайдери (такі як AWS) і пропонують рішення цієї проблеми у вигляді сервісів таких як Fargate (AWS), де оплата відбувається лише за використану потужність, у розрахунку на однакову потужність, вони все ще коштують більше, ніж орендовані сервери. У випадку відеоредактора навантаження на сервер є дещо більш передбачуваним, що дозволяє оптимізувати витрати на оренду.

Метою роботи є зниження навантаження на пристрій користувача під час обробки відеофайлів за рахунок використання серверних технологій.

Об'єкт дослідження: формати відео та аудіо.

Завданням даної роботи є аналіз існуючих рішень, створення програмного забезпечення та візуалізація процесу використання.

Практичним значенням даної роботи є можливість редагувати відео на пристроях з невеликою обчислювальною потужністю.

					КПІ.ІТ-84В11.045440.02.81	Арк.
						8
Змін.	Арк.	№ докум.	Підп.	Дата.		

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Відеоконтент є надзвичайно актуальним у наші часи, і його легко поширювати на таких платформах як Facebook, Instagram, LinkedIn і Twitter. Мільйони людей по всьому світу кожен день поглинають величезну кількість інформації на просторі соціальних мереж. Саме тому розміщення реклами на цих платформах є чудовим способом надати унікальний контент і підвищити актуальність бренду, тому цей маркетинговий хід є найбільш привабливим для більшості підприємців. Ви можете розмістити своє повідомлення перед цільовою аудиторією не встаючи з-за обіднього столу. Досить часто концептуальна відеореклама бренду допомагає клієнтам прийняти рішення про покупку певного продукту. Наявність відео, якими можна поділитися зі своїми клієнтами, на сьогоднішній день допомагає отримати значну перевагу на ринку.

Редагування відео — це процес послідовного об'єднання різних частин фільму, щоб розповісти історію або передати повідомлення.

Важливість процесу обробки відео не завжди є очевидною, оскільки робота відбувається за кадром, але саме монтаж у кінцевому результаті задає потрібний настрій рекламному ролику та згодом – потенційному покупцю, на якого з самого початку була розрахована реклама. Талановиті відеоредактори можуть перетворити звичайне відео для вашого бізнесу в чудовий маркетинговий інструмент, який допоможе залучити нових клієнтів і зберегти існуючих. Ваша команда може володіти чудовими навиками запису відео, але якщо ці кадри не з'єднуються в особливий сенс, потрібну динаміку та настрій, ваше повідомлення може бути втраченим.

1.2 Змістовний опис і аналіз предметної області

Для того, щоб зрозуміти вимоги до програмного продукту, необхідно глибше розглянути сам процес обробки відео, схема якого продемонстрована на рисунку 1.1:

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9



Рисунок 1.1 – Схема процесу створення відео

У загальному випадку, існують такі кроки:

- сортування і фільтрування – це найпростіше і найпоширеніше завдання в редагуванні. Багато відео можна значно покращити, просто позбувшись дефектних або небажаних фрагментів;
- пошук найкращого фрагменту – це коли знято набагато більше

матеріалу, ніж вам насправді потрібно, і монтажер обирає лише найкращі фрагменти для остаточного монтажу. Часто професіонали знімають декілька варіантів фрагменту і під час монтажу обирають найкращий;

- визначення порядку фрагментів – основною задачею більшості відео є розповідь історії або надання інформації. Фрагменти мають розташовуватися у порядку, який допоможе глядачу легше сприйняти історію, а не змусить його вимкнути відео;

- додавання ефектів, музики – це часто є найбільш захоплюючою частиною редагування відео. Більшість відео може бути покращена за рахунок додавання додаткових елементів;

- встановлення стилю, темпу або настрою відео – такі прийоми, як музика для настрою та візуальні ефекти, можуть впливати на реакцію аудиторії;

- фінальний рендеринг – застосування усіх налаштувань до відео і створення результату у заданій якості (формат відео, якість звуку).

Отже для відеоредакторів важливим є також перегляд проміжних результатів. Це навантаження в деяких випадках може бути перенесене з сервера на сторону клієнта.

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

Проаналізуємо відомі на сьогодні алгоритмічні забезпечення у даній області та технічні рішення, що допоможуть у реалізації програмного забезпечення для редагування відеофайлів. Далі будуть розглянуті допоміжні програмні засоби, засоби розробки та готові програмні рішення.

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

Розробка не має спеціальних алгоритмічних рішень, присутні лише загальні алгоритми, а тому необхідність в моїй розробці виконувати аналіз існуючих алгоритмічних рішень відсутня.

Щодо існуючих технічних рішень веб-сервіси для редагування відео можуть використовувати HTML5 та JavaScript. HTML5 надає багато

					КПІ.ІТ-84в11.045440.02.81	Арк.
						11
Змін.	Арк.	№ докум.	Підп.	Дата.		

можливостей для мультимедіа, включаючи підтримку відео. JavaScript дозволяє взаємодіяти з елементами відео і додавати функціональність редагування, таку як зміна розміру, обрізка або додавання ефектів. У процесі обробки відео на серверному боці можуть використовуватися мови програмування, наприклад, Python або Node.js.

Існують спеціалізовані фреймворки для розробки веб-сервісів, що дозволяють редагувати відео. Наприклад, Video.js - це потужний фреймворк, який забезпечує відтворення відео на веб-сторінці та API для керування ним. Також є Popcorn.js, який дозволяє створювати інтерактивні відео з різними ефектами та шарами змісту. Деякі фреймворки комбінують веб-інтерфейс з потужними бібліотеками для обробки відео, такими як FFmpeg або OpenCV.

Для зручності можна використовувати хмарні сервіси, які надають готові рішення для редагування відео через веб-сервіси. Наприклад, Amazon Elastic Transcoder від Amazon Web Services дозволяє змінювати формати, розміри та кодеки відео. Такі хмарні платформи, як Microsoft Azure або Google Cloud Platform, також надають схожі сервіси для редагування відео.

Крім того, існують сторонні бібліотеки та API, які можна використовувати для редагування відео через веб-сервіси. Наприклад, Video.js має плагіни для функцій редагування, таких як обрізка, зміна розміру або додавання підписів. Cloudinary надає API для маніпуляції відео, включаючи зміну розмірів, конвертацію форматів або додавання водяних знаків.

1.3.2 Аналіз допоміжних програмних засобів та засобів розробки

а) JavaScript є потужною мовою програмування для фронтенду. Вона використовується для створення динамічних інтерфейсів та взаємодії з користувачем на веб-сторінках. JavaScript має багато розширень і бібліотек, що дозволяють розробникам реалізувати різноманітні функції та ефективно працювати з веб-технологіями.

б) Java є мовою програмування бекенду, яка використовується для розробки серверних додатків та бізнес-логіки. Вона має широке застосування в

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

індустрії програмного забезпечення і відома своєю надійністю та масштабованістю. Java підтримує об'єктно-орієнтовану парадигму програмування та надає багатий набір бібліотек і фреймворків.

в) Spring є потужним фреймворком для розробки додатків на мові програмування Java. Він пропонує різноманітні модулі і компоненти для ефективної розробки програм, зокрема:

- 1) Spring Data - модуль для роботи з базами даних, який спрощує доступ і маніпулювання даними.
- 2) Spring Boot - фреймворк, який дозволяє швидко створювати і розгортати додатки з мінімальною конфігурацією.
- 3) Spring Web - модуль, який допомагає обробляти HTTP-запити і відповіді.
- 4) Spring Security - модуль для забезпечення безпеки додатків, включаючи аутентифікацію та авторизацію користувачів.

г) Amazon Web Services (AWS) - це платформа хмарних обчислень, яка надає широкий спектр послуг для розробки, випробування та розгортання додатків. Деякі з них включають:

- 1) Amazon S3 - послуга для збереження та керування файлами та об'єктами даних в хмарі.
- 2) Application Load Balancer (ALB) - послуга, яка розподіляє навантаження між різними серверами, забезпечуючи масштабованість і надійність додатків.
- 3) Amazon EC2 - послуга, яка дозволяє розміщувати програми на віртуальних серверах.
- 4) Amazon Route53 - послуга для створення та керування DNS-записами додатків, що дозволяє забезпечити доступ до них через Інтернет.

					КПІ.ІТ-84в11.045440.02.81	Арк.
						13
Змін.	Арк.	№ докум.	Підп.	Дата.		

1.3.3 Аналіз відомих програмних продуктів

Більшість існуючих рішень дозволяють редагувати відео, використовуючи клієнтське обладнання. Розглянемо наступні програмні продукти:

– Adobe Premiere Rush є популярним онлайн-редактором відео, який пропонує розширені можливості монтажу. Його інтуїтивно зрозумілий і структурований інтерфейс дозволяє зручно працювати з відео. Adobe Premiere Rush надає широкий спектр функцій, таких як обрізка, зміна розміру, додавання ефектів і переходів, що дозволяє створювати професійно виглядаючі відео.

– iMovie є інструментом редагування відео, спеціально розробленим для користувачів пристроїв Apple. Його вбудований набір інструментів дозволяє зручно редагувати відео, змінюючи його розмір, додавати ефекти і фільтри, а також працювати зі звуком. iMovie має простий у використанні інтерфейс, що робить процес редагування відео доступним для користувачів будь-якого рівня навичок.

– WeVideo є потужною інтернет-платформою для редагування відео, яка надає багато функцій і можливостей. Вона підтримує обробку відео з різних джерел, що дозволяє вам працювати зі своїми власними матеріалами або використовувати контент з бібліотеки програми. WeVideo надає різноманітні інструменти для редагування, включаючи шаблони, фільтри, ефекти та аудіо-мікшери, що дозволяє вам створювати вражаючі відеоролики онлайн.

Для порівняння розробки з дипломного проєкту з аналогами можна скористатись таблицею 1.1.

Таблиця 1.1 – Порівняння з аналогами

Назва програмного забезпечення	Функціонал
Adobe Premiere Rush	Надає широкий спектр функцій монтажу відео. Має інтуїтивний та зручний у використанні інтерфейс. Підтримує Windows, macOS та Linux.

Продовження таблиці 1.1

iMovie	Програмний продукт з вбудованим набором інструментів для обробки відео, включаючи обрізку, зміну розміру, додавання ефектів та фільтрів. Легкий у використанні інтерфейс з інтуїтивно зрозумілим навігаційним меню.
WeVideo	Надає розширені можливості, включаючи використання шаблонів, фільтрів, ефектів та аудіо-мікшерів. Зручний та простий у використанні інтерфейс.
Програмне забезпечення для редагування відеофайлів	Продукт повинен надавати можливість виконувати базові дії з редагування відео, а саме: обрізання відео; склеювання відео; додавання аудіо; накладання відео на відео, відео на фото і навпаки, додавання фільтрів; виконувати базові ефекти.

1.4 Аналіз вимог до програмного забезпечення

Даний програмний продукт повинен бути легким у користуванні та інтуїтивно зрозумілим. Головною його перевагою над конкурентами буде зміщення навантаження, пов'язаного з обробкою відео з клієнтського комп'ютера на сервер, відповідно, навантаження на клієнтський комп'ютер повинно бути мінімальним. Інтерфейс повинен давати доступ до усіх можливостей, використовуючи мінімальну кількість ресурсів процесора та відеокарти для продовження періоду автономної роботи комп'ютера [3]. В той же час, результати роботи повинні транслюватися на таймлайні в режимі

					КПІ.ІТ-84в11.045440.02.81	Арк.
						15
Змін.	Арк.	№ докум.	Підп.	Дата.		

реального часу з сервера на клієнт, щоб монтажер мав змогу без перешкод та помилок з боку програмного забезпечення створювати якісний відеоряд.

Продукт повинен надавати можливість виконувати базові дії з редагування відео, а саме:

- обрізання відео;
- склеювання відео;
- додавання аудіо;
- накладання відео на відео, відео на фото і навпаки.
- додавання фільтрів;
- виконувати базові ефекти.

Головною функцією програмного забезпечення є обробка відеоматеріалів, більше функцій можна побачити на рисунку 1.2.

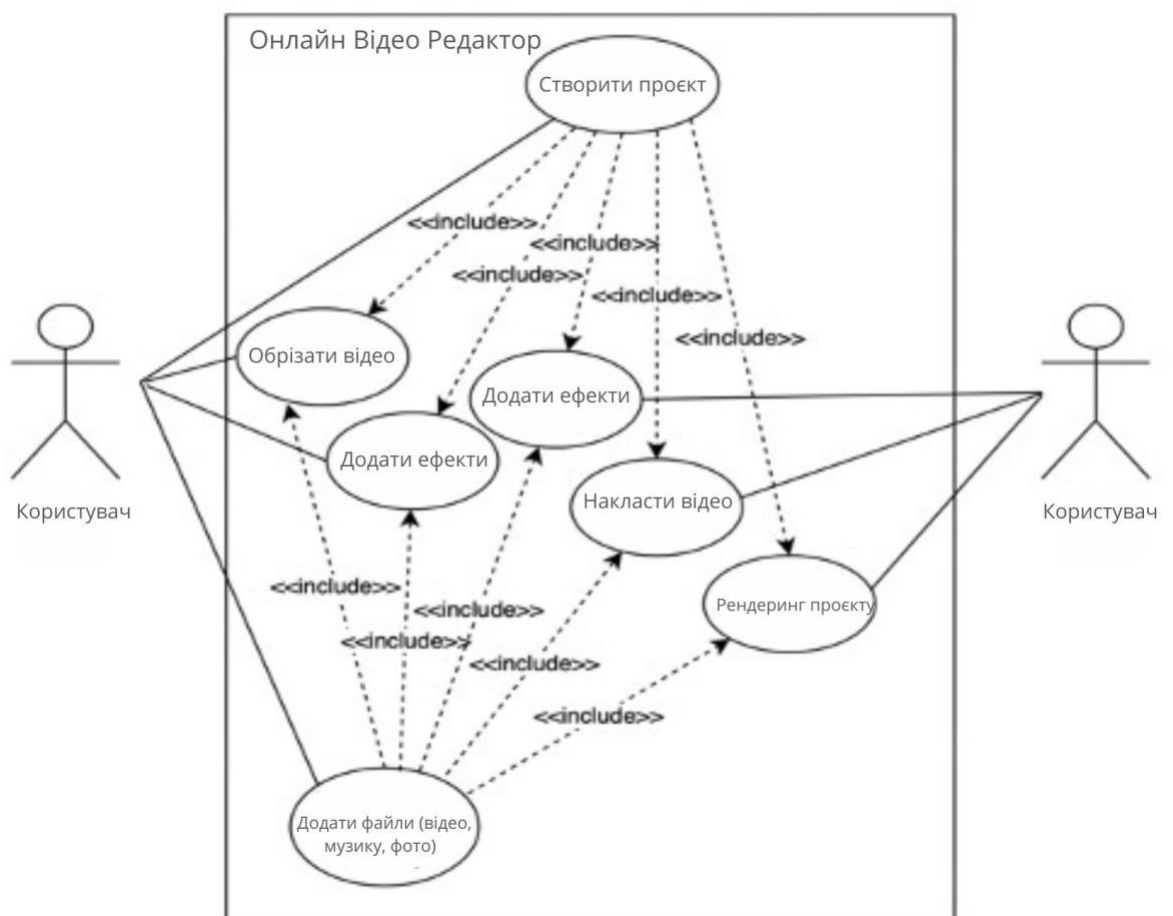


Рисунок 1.2 – Діаграма варіантів використання

В таблицях 1.2 - 1.7 наведені варіанти використання програмного забезпечення.

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		16

Таблиця 1.2 – Варіант використання UC-1

Унікальний ідентифікатор	UC-1
Назва	Реєстрація
Опис	Користувач реєструється на сервері
Учасники	Користувач
Передумови	Відкрита сторінка реєстрації на веб-сервері
Постумови	Користувач зареєструвався на веб-сервері
Сценарій	– Користувач заходить на веб-сервер – Користувач обирає форму для реєстрації – Користувач реєструється на веб-сервері

Таблиця 1.3 – Варіант використання UC-2

Унікальний ідентифікатор	UC-2
Назва	Обрізання відео

Продовження таблиці 1.3

Опис	Користувач обирає межі відео та обрізає потрібний фрагмент
Учасники	Користувач
Передумови	Відкрита сторінка редагування відео, користувач завантажив потрібний файл MP4
Постумови	Користувач обрізав потрібний йому фрагмент
Сценарій	– Користувач завантажує файл MP4 – Користувач обирає межі потрібного йому фрагменту – Користувач обрізає відео, натіснувши кнопку Cut

Таблиця 1.4 – Варіант використання UC-3

Унікальний ідентифікатор	UC-3
Назва	Склеювання відео
Опис	Користувач обирає фрагменти відео та склеює їх
Учасники	Користувач
Передумови	Відкрита сторінка редагування відео, користувач завантажив потрібні

Продовження таблиці 1.4

	файли MP4, обрав потрібні йому фрагменти
Постумови	Користувач склеїв попередньо обрані фрагменти відео
Сценарій	– Користувач завантажує файл MP4 – Користувач обирає потрібні йому фрагменти – Користувач склеює потрібні йому фрагменти

Таблиця 1.5 – Варіант використання UC-4

Унікальний ідентифікатор	UC-4
Назва	Додавання фрагментів відео/аудіо/зображення
Опис	Користувач додає потрібні йому файли MP3/MP4/JPEG/PNG
Учасники	Користувач
Передумови	Відкрита сторінка редагування відео
Постумови	Користувач додав на таймлайн потрібні йому відео/аудіо/фото

Сценарій	– Користувач натискає на пункт меню «Додати» Користувач обирає потрібні йому медіафайли з файлової системи або на сервері – Користувач натискає кнопку «Ок»
----------	--

Таблиця 1.6 – Варіант використання UC-5

Унікальний ідентифікатор	UC-5
Назва	Фінальний рендеринг
Опис	Користувач створює фінальне відео
Учасники	Користувач
Передумови	Відкрита сторінка редагування відео, користувач провів усе потрібне редагування
Постумови	Користувач створив фінальне відео
Сценарій	– Користувач редагує файли та створює відеоряд – Користувач натискає на кнопку «Render» та створює фінальне відео

Таблиця 1.7 – Варіант використання UC-6

Унікальний ідентифікатор	UC-6
Назва	Застосування ефектів для відео/фото/аудіо
Опис	Користувач додає ефекти до обраного медіафайлу
Учасники	Користувач
Передумови	Відкрита сторінка редагування відео, додані медіафайли на таймлайн
Постумови	Користувач додав на таймлайн потрібні йому відео/аудіо/фото
Сценарій	– Користувач додає медіафайли – Користувач обирає фрагмент та відмічає ефект на панелі

1.4.1 Розроблення функціональних вимог

На рисунку 1.3 наведено загальну модель вимог, а в таблицях 1.8 – 1.17 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування вимог можна побачити на рисунку 1.4.

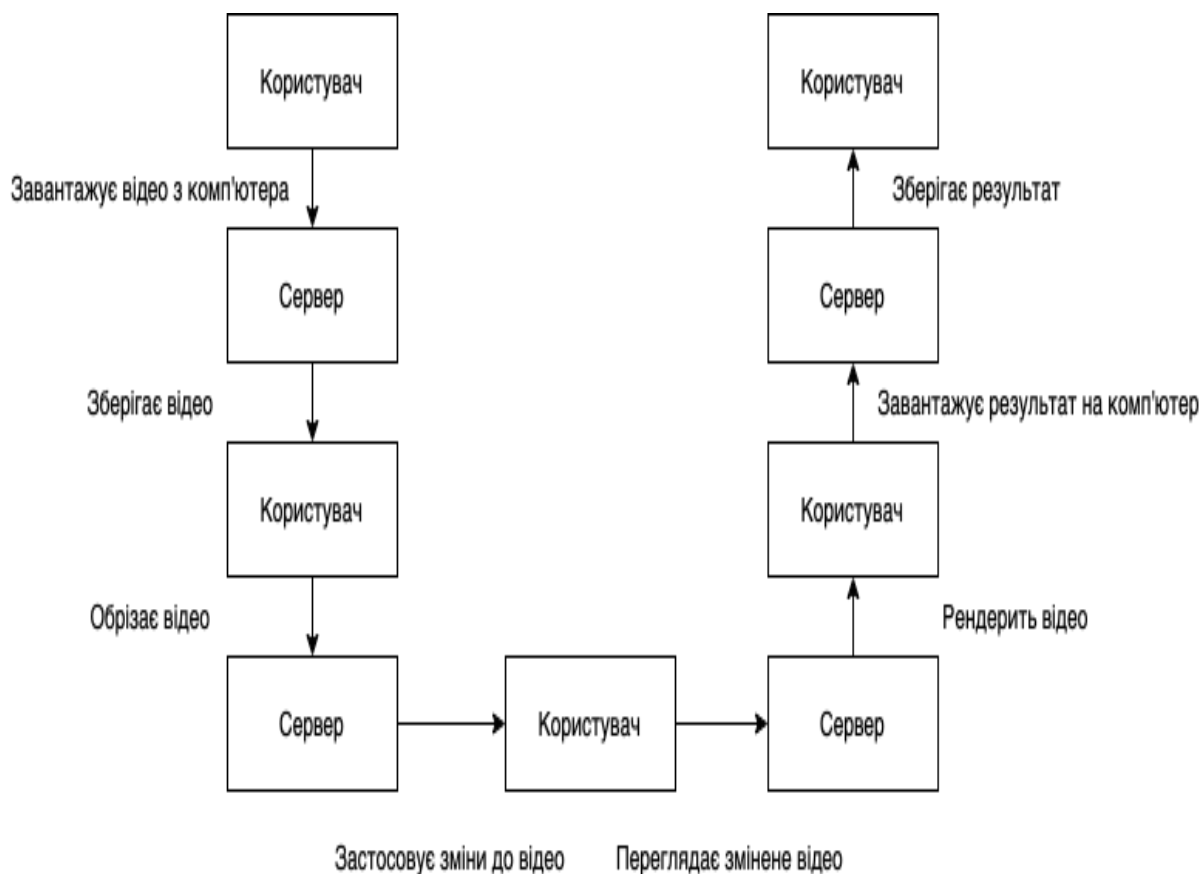


Рисунок 1.3 – Модель вимог у загальному вигляді

Таблиця 1.8 – Функціональна вимога FR-1

Назва	Реєстрація
Опис	Реєстрація необхідна для обмеження доступу до інших клієнтів.

Таблиця 1.9 – Функціональна вимога FR-2

Назва	Обрізання відео
Опис	Система повинна надавати можливість обрізання відео користувачеві, обираючи межі відео.

Таблиця 1.10 – Функціональна вимога FR-3

Назва	Склеювання відео
Опис	Система повинна надавати можливість склеювання відео користувачеві, обираючи потрібні фрагменти.

Таблиця 1.11 – Функціональна вимога FR-4

Назва	Додавання фрагментів відео
Опис	Система повинна надавати можливість додавати фрагменти відео, обираючи їх з бібліотеки доданих файлів з файлової системи комп'ютера.

Таблиця 1.12 – Функціональна вимога FR-5

Назва	Додавання аудіо
Опис	Система повинна надавати можливість додавати фрагменти аудіо, обираючи їх з бібліотеки доданих файлів з файлової системи комп'ютера.

Таблиця 1.14 – Функціональна вимога FR-7

Назва	Переміщення відео та фото
Опис	Редактор повинен надавати можливість переміщати відео та фото на робочій панелі, використовуючи курсор миші.

Таблиця 1.15 – Функціональна вимога FR-8

Назва	Фінальний рендеринг
-------	---------------------

Продовження таблиці 1.15

Опис	Редактор повинен надавати можливість створення фінального відео та його експорту на комп'ютер користувача.
------	--

Таблиця 1.16 – Функціональна вимога FR-9

Назва	Застосування ефектів для відео/фото/аудіо
Опис	Редактор повинен надавати можливість накладення ефектів на відео/фото/аудіо.

Таблиця 1.17 – Функціональна вимога FR-10

Назва	Переміщення відео та фото
Опис	Редактор повинен надавати можливість переміщати відео та фото на робочій панелі, використовуючи курсор миші.

Таблиця 1.18 – Функціональна вимога FR-11

Назва	Завантаження результату
Опис	Редактор повинен надавати можливість експортувати результат монтажу на комп'ютер користувача.

Таблиця 1.19 – Функціональна вимога FR-12

Назва	Перегляд бібліотеки відео онлайн
Опис	Редактор повинен надавати можливість перегляду доданих файлів та використовувати для проєктів.

Таблиця 1.20 – Функціональна вимога FR-13

Назва	Видалення відео з серверу
Опис	Редактор повинен надавати можливість видаляти вже додані користувачем файли.

Таблиця 1.21 – Функціональна вимога FR-14

Назва	Завантаження відео з комп'ютера
Опис	Редактор повинен надавати можливість завантажити відео з локального комп'ютера на сервер для подальшої обробки.

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8	FR-9	FR-10	FR-11	FR-12	FR-13	FR-14
UC-1	+													
UC-2		+												
UC-3			+											
UC-4				+	+									
UC-5					+			+						
UC-6						+	+		+	+				

Рисунок 1.4 – Матриця трасування вимог

1.4.2 Нефункціональні вимоги

Інтерфейс програми повинен бути максимально простим для продовження періоду життя акумулятора мобільного пристрою, програма повинна переносити максимальну кількість обчислень на сервер, щоб зменшити навантаження на клієнтський комп'ютер. Результати редагування відео повинні транслюватися в режимі реального часу, беручи до уваги швидкість передачі даних в мережі. Вимоги до операційних систем: оскільки програмний продукт розроблений на Java [1] (бекенд) та NodeJS [9] (фронтенд), його можливо запустити на будь-якій операційній системі для якої реалізовано віртуальну машину Java та пакет

					КПІ.ІТ-84в11.045440.02.81	Арк.
						25
Змін.	Арк.	№ докум.	Підп.	Дата.		

NodeJS. До цього списку входять усі найпопулярніші операційні системи, як Windows, Linux та Mac. Слід зауважити, що існують мінімальні вимоги для обладнання, на якому запускатиметься JVM (Java Virtual Machine), а саме: 128 Мб оперативної пам'яті та процесор Intel Pentium 2.2 MHz. Ці вимоги повинні виконуватися для кожної віртуальної машини і, звісно, для стабільної роботи сервісу бажано мати потужніший комп'ютер. Після деплою сервіси запускатимуться на AWS EC2 t2.micro (оскільки цей тип віртуальної машини входить до Free Tier), яка має 1 Гб оперативної пам'яті, 1 процесор та 8 Гб диску. Також існують мінімальні вимоги для MongoDB, а саме процесор на 1Ггц, 2 Гб оперативної пам'яті 512 Мб жорсткого диску. Оскільки база даних запускатиметься на AWS MongoDB Atlas, нам не потрібно думати про ці обмеження.

1.5 Постановка задачі

Метою розробки даного дипломного проєкту є полегшення обробки відеофайлів за рахунок використання серверних технологій.

Задачею розробки проєкту є розробка програмного забезпечення для редагування відеофайлів з перенесенням максимальної кількості обчислень на сервер, щоб зменшити навантаження на клієнтський комп'ютер

Поставлені наступні задачі:

- розробка повинна мати максимально простий інтерфейс для подовження життя акумулятора пристрою користувача;
- програмне забезпечення повинно надавати можливість користувачеві виконувати базові функції з обробки відеофайлів;
- наявність можливості робити рендеринг фінального відеоряду та завантажувати його на власний пристрій.

Висновки по розділу

У першому розділі було встановлено мету роботи, розглянуто наявні рішення, найбільш ефективні алгоритми і методи обробки відеофайлів, було

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		26

виявлено їх недоліки та способи усунення цих недоліків.

Також було проаналізовано наявні аналоги та рішення. Було визначено функціональні та нефункціональні вимоги до програмного продукту. Актором програмного забезпечення є користувач.

					КПІ.ІТ-84В11.045440.02.81	Арк.
						27
Змін.	Арк.	№ докум.	Підп.	Дата.		

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробку даного програмного продукту умовно можна розбити на 2 частини:

- Frontend – у якості середовища розробки використовується VS Code;
- Backend – у якості середовища розробки використовується IntelliJ

Idea та VS Code має безліч плагінів, серед яких є й такі, що дозволяють перезавантажувати програму в браузері після змін без зусиль з боку програміста, що значно спрощує процес розробки [4]. IntelliJ Idea створена для розробки програм мовою Java та має кілька досить зручних плагінів для роботи з базами даних.

2.1 Моделювання та аналіз програмного забезпечення

У загальному випадку модель архітектури фронтенду може бути зображена, як на рисунку нижче:

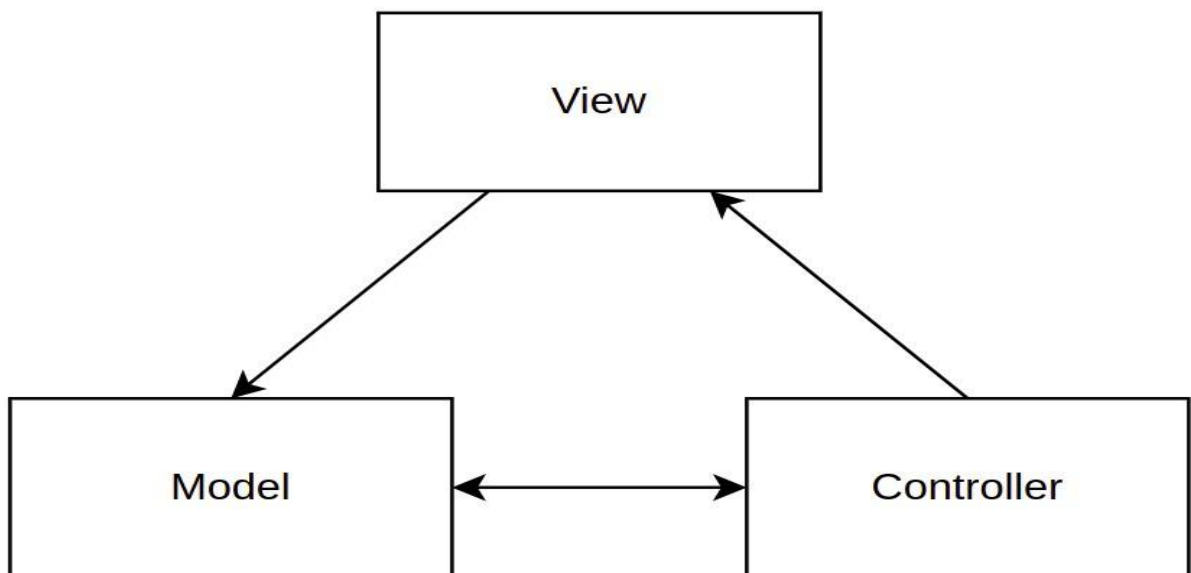


Рисунок 2.1 — Загальна модель архітектури фронтенду (MVC)

А модель архітектури бекенду виглядає, як на рисунку 2.2:

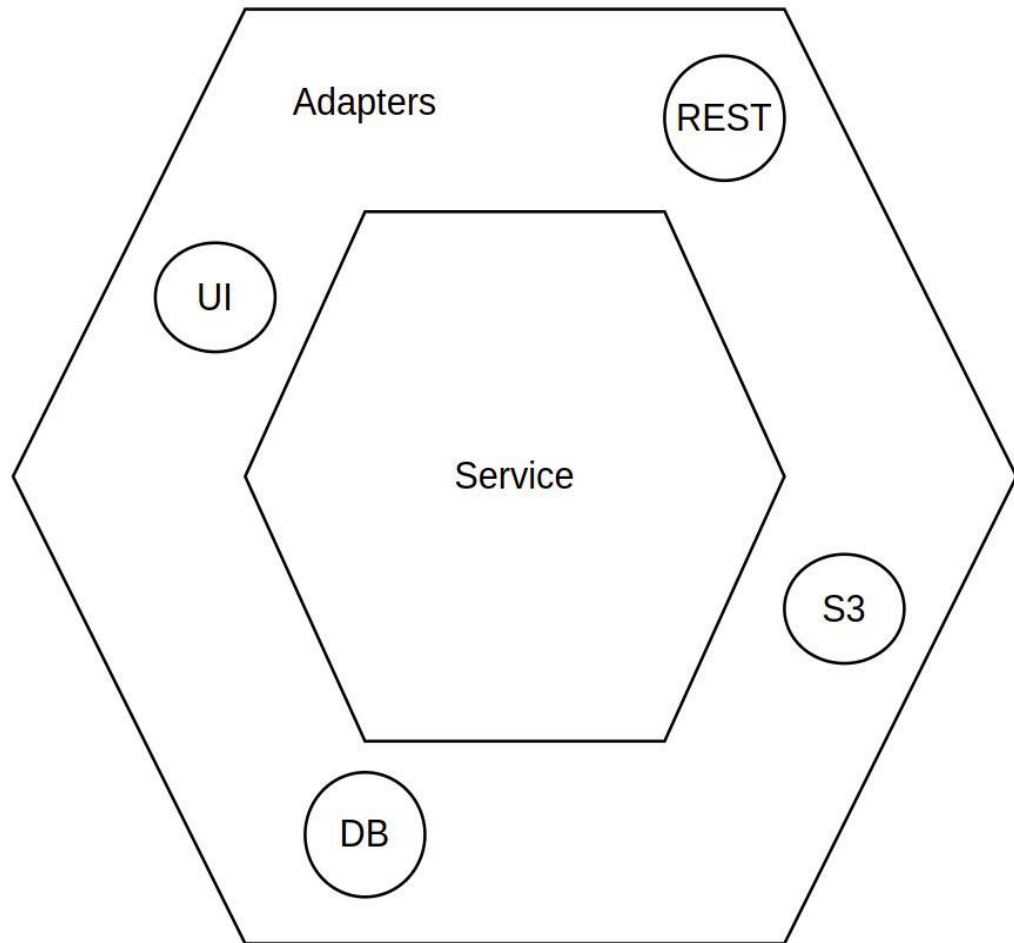


Рисунок 2.2 – Загальна модель архітектури бекенду (Hexagonal)

На комп'ютері користувача (Frontend) у браузері працює програма, яка відповідальна за передачу інструкцій серверу (Backend) і відображення результатів. Сервер ж відповідальний за обробку відео, передачу рівню завершеності процесу рендерингу, реєстрацію та завантаження відео.

Розглянемо детальну діаграму роботи програмного продукту на рисунку 2.3:

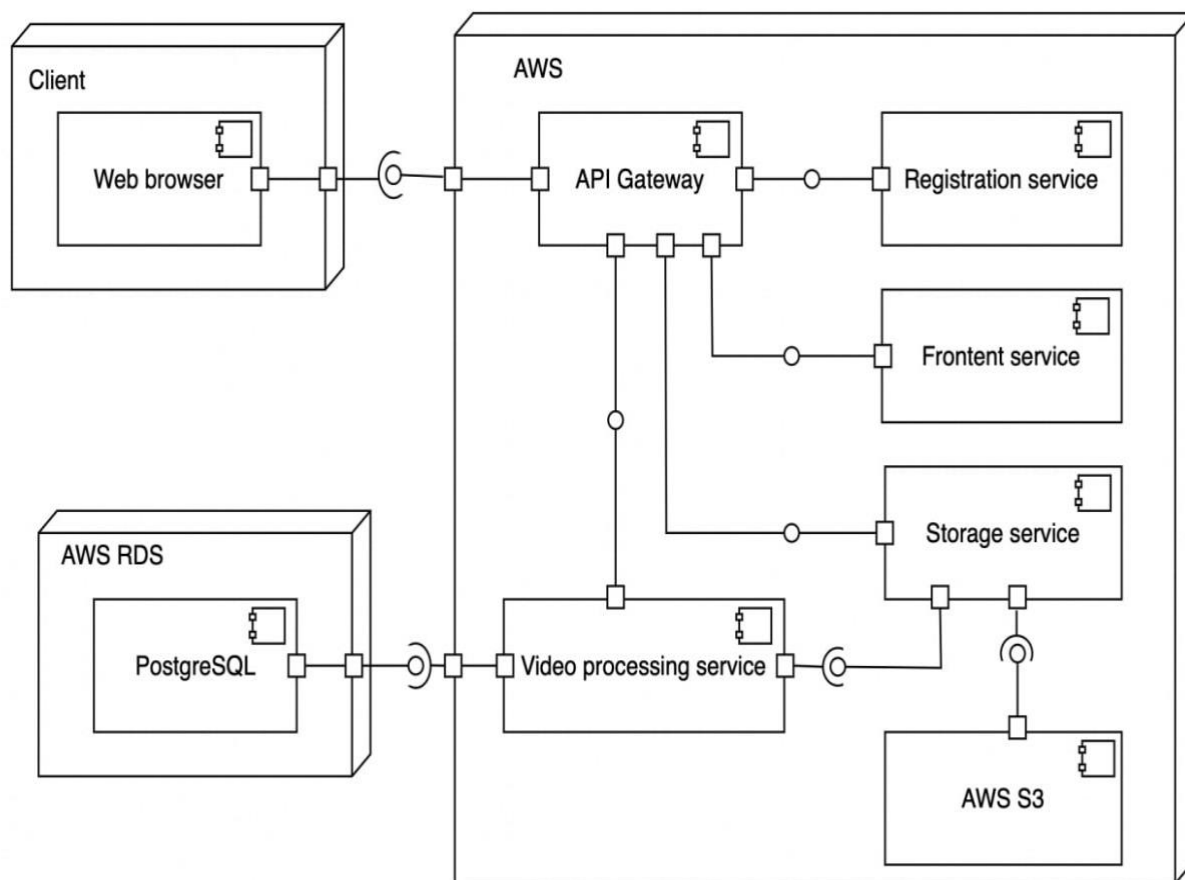


Рисунок 2.3 — Діаграма компонентів

Вона має наступні компоненти, які представлені в таблиці 2.1:

Таблиця 2.1 — Компоненти програми

Номер	Назва	Опис
1	Клієнт	Програма відповідальна за передачу інструкцій серверу і відображення результатів
2	Registration service	Сервіс реєстрації користувачів

Продовження таблиці 2.1

3	Frontend service	Сервіс видачі сайту по запиту
4	Storage service	Сервіс, відповідальний за завантаження відео з локального комп'ютера чи з Cloud
5	Video processing service	Сервіс, відповідальний за обробку відео. Має містити Load Balancer, оскільки навантаження на цей сервіс буде найбільшим
6	mongoDB	База даних користувачів, використовується для реєстрації та зберігає посилання на відео користувачів
7	S3	Тут зберігаються файли користувачів: відео, аудіо, картинки.

Для розгортання всієї інфраструктури можна використати Terraform або CloudFormation.

Послідовність роботи з програмою така:

- клієнт заходить на сайт;
- запит відправляється на Frontend service, який повертає сторінку для браузера;
- клієнт обирає спосіб реєстрації чи входу в систему;
- відбувається реєстрація чи вхід клієнта в систему за допомогою Registration service;
- дані клієнта заносяться в mongoDB;
- клієнт створює новий проєкт;
- дані про проєкт заносяться в mongoDB;
- клієнт завантажує необхідні фрагменти відео, аудіо та фото;
- завантаження відбувається за допомогою Storage service;
- дані про завантажені файли заносяться в mongoDB;
- самі файли загружаються на S3;
- клієнт працює над проєктом;
- frontend додаток відправляє запити на Video Processing service;
- video processing service виконує запити і відправляє результат клієнту, зберігаючи інформацію про виконані дії на S3;
- клієнт натискає кнопку “Render”;
- video processing service виконує обробку відео, завантажує його на S3 та відправляє посилання клієнту;
- клієнт завантажує готове відео на локальний комп’ютер.

Схема бізнес-процесу представлена на рисунку 2.4:

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		32

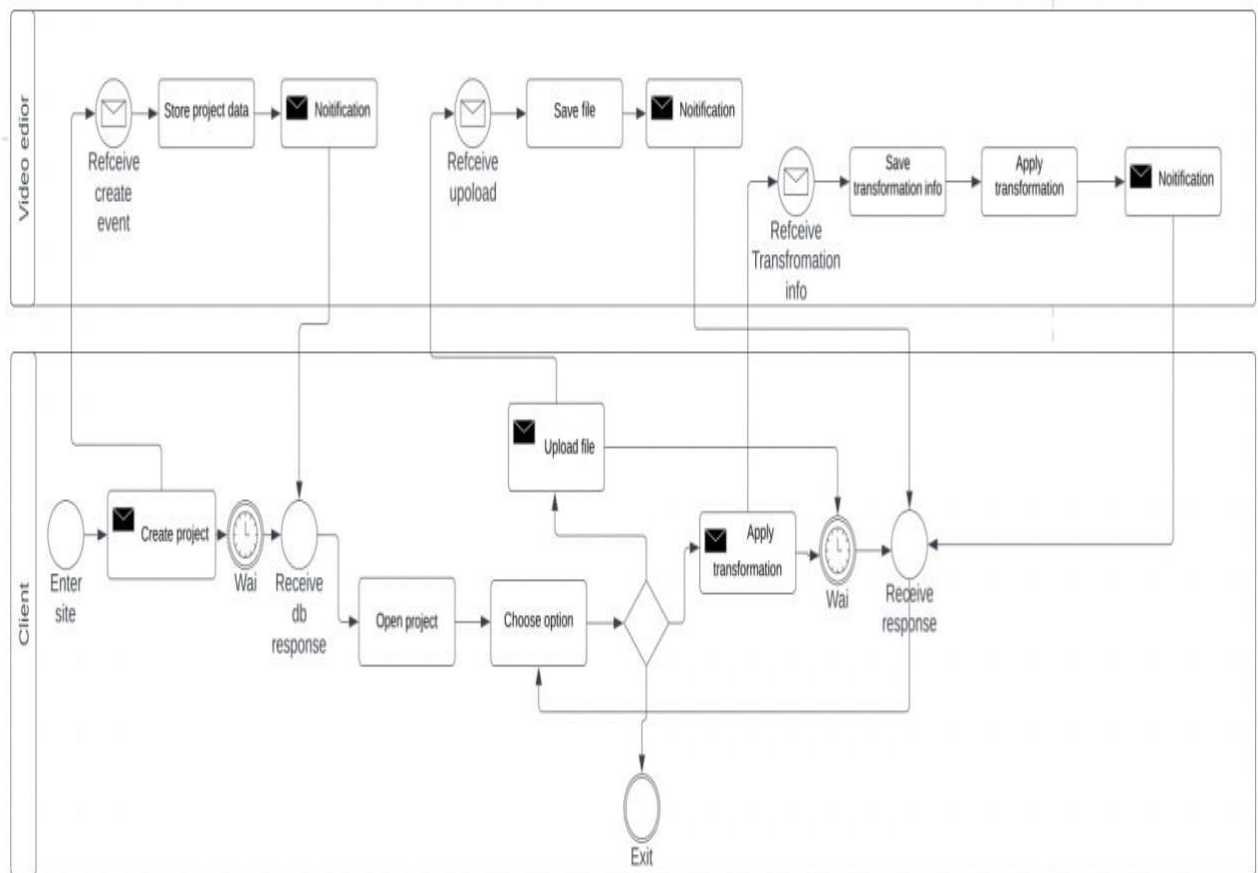


Рисунок 2.4 – Схема бізнес-процесу

Користувач вводить дані про проект і натискає кнопку «Створити», запит надходить на сервер, який зберігає дані про проект у базі і надсилає відповідь назад у браузер. Для того, щоб продовжити роботу, користувачу необхідно відкрити проект. Далі є дві опції – додати до проекту файли та застосувати трансформації до вже доданих. При виборі першої – користувач відправляє файл на сервер, який зберігає його та відправляє сповіщення про це користувачеві. При виборі другої – інформація про трансформацію відправляється на сервер, де вона зберігається, відео обробляється та результат відправляється користувачеві.

2.2 Архітектура програмного забезпечення

Під час розробки програмного забезпечення використовуються такі технології:

- а) JavaScript – мова програмування фронтенду;

- б) Java – мова програмування бекенду;
- в) Spring [8] – фреймворк для Java:
 - 1) Spring data – робота з базами даних;
 - 2) Spring boot – спрощення розробки;
 - 3) Spring web – для прийому і передачі http запитів;
 - 4) Spring security – реєстрація;
- г) AWS:
 - 1) S3 – файлова система для збереження даних;
 - 2) ALB – балансування навантаження між серверами;
 - 3) EC2 – для розіщення програм;
 - 4) Route53 – для створення DNS програми.

Опис даних

Вхідними даними у програмному забезпеченні є mp4 файли, які конвертуються в сутності та зберігаються в cloud-системі.

Опис сутностей вхідних даних:

Audio – створена для зберігання звукових доріжок, для їх подальшого модифікування та перетворення:

Таблиця 2.2 – Опис вхідних даних для Audio

Назва поля	Тип	Опис
path	String	Використовується для задання шляху в cloud систему
isPublic	Boolean	Характеристика доступу інших користувачів до цих файлів
owner	User	Користувач, кому належить певна аудіо-доріжка

Image – створена для зберігання png-файлів, які використовуються для модифікації відео та додавання спеціальних ефектів.

Таблиця 2.3 – Опис вхідних даних для Image

Назва поля	Тип	Опис
location	String	Використовується для задання шляху в cloud систему
isPublic	Boolean	Характеристика доступу інших користувачів до цих файлів
owner	User	Користувач, кому належить певна аудіо-доріжка

Video – створена для зберігання row mp4 -файлів, які використовуються для модифікації відео та додавання спеціальних ефектів.

Таблиця 2.4 – Опис вхідних даних для Video

Назва поля	Тип	Опис
isPublic	Boolean	Характеристика доступу інших користувачів до цих файлів
owner	User	Користувач, кому належить певна аудіо-доріжка

User – основна сутність для зберігання інформації певного користувача.

Таблиця 2.5 – Опис вхідних даних для User

Назва поля	Тип	Опис
username, email, firstName, lastName	String	Основна інформація певного користувача яка зберігається до бази даних. Використовується в подальшому для авторзації та аутентифікації
videos	List<Video>	Список відео-доріжок користувача, який використовується для подальшої модифікації.
Images	List<Audio>	Список аудіо-доріжок користувача, який використовується для подальшої модифікації.
Audios	List<Image>	Список png зображень користувача, який використовується для подальшої модифікації.

Вихідними даними в програмному забезпеченні виступає конвертований та модифікований mp4-файл, який генерується з раніше описаних сутностей.

2.3 Конструювання програмного забезпечення

Для написання фронтенду використовувалась мова програмування JavaScript.

Також було використано стилі з бібліотеки Bootstrap.

Структура директорій в сервісі фронтенду має вигляд, зображений на рисунку 2.5:

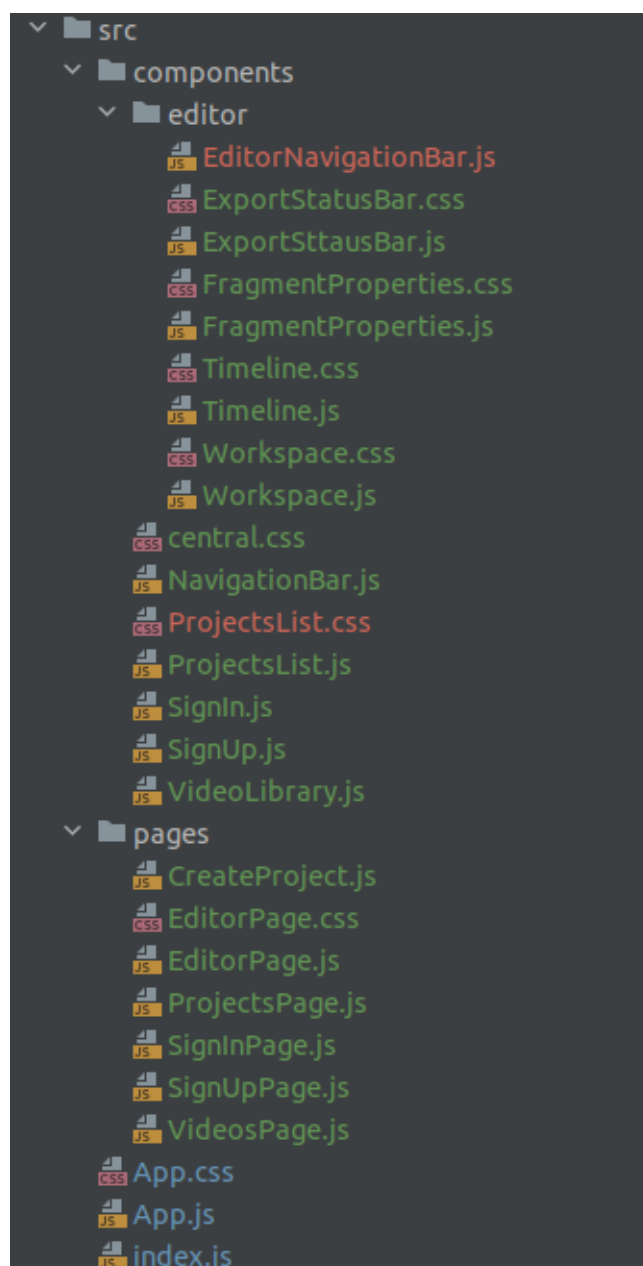


Рисунок 2.5 — Структура фронтенду проекту

Файли відсортовані по папках відповідно до їх функцій. Функції папок

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		36

проєкту описано у таблиці 2.2.

Таблиця 2.2 — Функції папок проєкту

Папка	Функція файлів
components	Містить компоненти, з яких будується інтерфейс проєкту.
components/editor	Містить компоненти, які використовується безпосередньо в онлайн-редакторі відеофайлів.
pages	Містить сторінки проєкту. Компоненти з цієї папки відповідають за відображення компонент з папки components.

Опис функцій файлів JavaScript наведений у таблиці 2.3:

Таблиця 2.3 — Опис функцій файлів JavaScript

Назва файлу	Функція
App.js	Головна компонента проєкту, відповідає за відображення усіх інших компонент.
pages/CreateProject.js	Сторінка створення проєкту, містить навігаційну панель вгорі. Дозволяє задати назву та опис проєкту.
pages/SignUpPage.js	Сторінка, що містить форму реєстрації нового користувача і навігаційну.

Продовження таблиці 2.3

	панель вгорі. Дозволяє задати ім'я, email, пароль та стаття.
pages/EditorPage.js	Головна компонента відеоредактора, відповідає за навігаційну панель вгорі та відображення усіх інших компонентів.
pages/SignInPage.js	Сторінка, що містить форму входу в існуючий акаунт і навігаційну панель вгорі. Необхідні поля: email та пароль.
pages/VideosPage.js	Сторінка, що містить список фрагментів, завантажених користувачем на сервер і навігаційну панель вгорі.
pages/ProjectsPage.js	Сторінка, що містить список проєктів, створених користувачем і навігаційну панель вгорі.
components/SignIn.js	Компонента, що містить форму входу в існуючий акаунт. Необхідні поля: email та пароль.
components/VideoLibrary.js	Компонента, що містить список фрагментів, завантажених користувачем на сервер.
components/editor/Workspace.js	Компонента, що відповідає за відображення поточного стану відеофайлу.

Продовження таблиці 2.3

components/editor/FragmentProperties.js	Компонента, що відповідає за відображення фрагментів відео, фото та аудіо, доданих до поточного проєкту.
components/editor/ExportSttausBar.js	Компонента, що відповідає за відображення поточного статусу під час фінального рендерингу відео.
components/editor/EditorNavigationBar.js	Навігаційна панель для редактора відеофайлів. Містить додаткові елементи (File, Render) у порівнянні з навігаційною панеллю інших сторінок.
components/editor/Timeline.js	Компонента, що відповідає за відображення часолінії. Містить зображення усіх відео, аудіо та фото у проєкті, відображає їх положення в часі, дозволяє переміщення в часі, розбиття відео (кнопка Cut), та обрізання. Містить панель шарів, що дозволяє змінювати порядок накладання фрагментів у робочій зоні.
components/SignUp.js	Компонента, що містить форму реєстрації нового користувача. Дозволяє задати ім'я, email, пароль та статать.
components/NavigationBar.js	Навігаційна панель. Дозволяє переміщатись між сторінками сайту

Продовження таблиці 2.3

	проекти, список фрагментів, домашня сторінка, реєстрація нового користувача, вхід у існуючий акаунт). У випадку, якщо користувач увійшов у акаунт, містить кнопку “Sign out”, що дозволяє вийти з акаунта.
components/ProjectsList.js	Компонента, що відповідає за відображення проєктів доданих користувачем.
index.js	Головний файл проєкту, задає головну компоненту, яка відображатиметься на сайті.

Розглянемо основні таблиці (бази даних) програмного продукту:

Таблиця 2.4 — Основні таблиці

Таблиця	Опис
User	Містить дані користувача (ім'я, email, пароль, стать та інші не обов'язкові при реєстрації). Всі дані зашифровані, ключ зберігається на сервері Vault.
Videos	Містить дані про відео (розміщення на S3)
Audious	Містить дані про аудіо (розміщення на S3)
Images	Містить дані про зображення (розміщення на S3)
ProcessingData	Містить дані про поточний стан роботи з відео

Розглянемо класи сервісів.

Таблиця 2.5 — Класи сервісів

ProjectEntity	Об'єкт, що відображає проєкт в базі даних, всі дані зашифровані, тому для роботи з об'єктом потрібна кастомна реалізація репозиторію
UserEntity	Об'єкт, що відображає користувача в базі даних
VideoEntity	Об'єкт, що відображає відео в базі даних
AudioEntity	Об'єкт, що відображає аудіо в базі даних
ImageEntiity	Об'єкт, що відображає зображення в базі даних
ProcessingDataEntity	Об'єкт, що відображає поточний стан обробки відео в базі даних
ProjectRepository	Репозиторій для доступу до даних про проєкти, кожен запис містить посилання на користувача, яким був створений конкретний проєкт та прапорець, який вказує, чи доступний цей проєкт іншим користувачам (включно з не зареєстрованими)
UserRepository	Репозиторій для доступу до даних користувача в базі даних
Imagerepository	Репозиторій для доступу до даних про зображення в базі даних, кожен запис містить посилання на користувача, яким він був доданий та прапорець, який вказує, чи доступний цей файл іншим користувачам (включно з не зареєстрованими)
AudioRepository	Репозиторій для доступу до даних про аудіо в базі даних, кожен запис містить посилання на користувача, яким він був доданий та прапорець,

Продовження таблиці 2.5

	який вказує, чи доступний цей файл іншим користувачам (включно з не зареєстрованими)
ProcessingDataRepository	Репозиторій для доступу до даних про поточний стан обробки відео в базі даних
Startup	Клас, з якого починають роботу всі сервіси
SecurityConfiguration	Налаштування безпеки, містить конфігурацію доступу до контролерів та ресурсів системи
WebMvcConfiguration	Налаштування MVC
UserController	REST клієнт для даних про користувача, дозволяє проводити реєстрацію нових користувачів
VideoController	REST клієнт для даних про відео, дозволяє додавати, видаляти та отримувати список публічних та приватних відео файлів
AudioController	REST клієнт для даних про аудіо, дозволяє додавати, видаляти та отримувати список публічних та приватних аудіо файлів
ImageController	REST клієнт для даних про зображення, дозволяє додавати, видаляти та отримувати список публічних та приватних зображень
ProcessingDataController	REST клієнт для даних про поточний стан обробки відео
UserService	Сервіс роботи з даними користувача
VideoService	Сервіс роботи з даними відео
ProcessingDataService	Сервіс роботи з даними про поточний стан обробки відео

Користувач взаємодіє з сервісами через REST інтерфейс. Обмін даними з деякими сервісами (наприклад, сервіс обробки відео) відбувається через Load Balancer, оскільки навантаження на ці сервіси найбільше і досить часто відбуваються стрибки навантаження, тому може знадобитись підвищення потужності. Це можна зробити двома способами:

- підвищенням характеристик сервера;
- збільшенням кількості серверів.

Перший спосіб не завжди є доцільним, адже в своєму чистому вигляді вимагає зупинки сервера. Можлива, також, модифікація, за якої попередня версія сервера знаходиться онлайн, доки не запуститься нова і не буде встановлено нове програмне забезпечення [5]. Також, при використанні цього способу, новий сервер отримує нову IP адресу, що ускладнило б імплементацію клієнта, тому у архітектурі є необхідним елемент, який приховав би зміну IP. Таким елементом може стати Load Balancer. Проте, використання балансування навантаження робить цей спосіб ускладненим варіантом другого підходу.

Другий варіант реалізації – балансувальник навантаження, який розподіляє запити між набором однакових серверів. Кількість серверів повинна мінятися в залежності від навантаження на них для того, щоб зберегти мінімальну затримку (Latency) при пікових навантаженнях та мінімізувати витрати коштів при мінімальних (більшість серверів не виконуватимуть жодних функцій, але за них все ще потрібно платити).

Існує ще одне питання, яке необхідно обдумати до реалізації другого підходу описаного вище. Розглянемо як відбуватиметься взаємодія користувача з сервером через балансувальник навантаження:

- користувач відправляє запит на Load Balancer;
- Load Balancer обирає доступний сервер на який потрібно передати запит користувача і передає його;
- запит обробляється на сервері;
- після обробки сервер передає відповідь на Load Balancer;
- Load Balancer передає відповідь клієнту;

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		43

- клієнт відправляє наступний запит;
- Load Balancer обирає доступний сервер, на який потрібно передати запит користувача і передає його.

Цей цикл продовжується і далі. Попри переваги описані вище, він має один великий недолік, який можна помітити, проаналізувавши формулювання 7-го пункту – немає гарантії, що запит піде на той же сервер, на який він пішов раніше [6]. Тому дані про поточний стан обробки відео, які зберігаються на сервері повинні бути винесені окремо, наприклад, в базу даних чи в чергу (Kinesis, kafka, sqs). Після цього 8-м пунктом можна додати “Сервер запис про поточний стан обробки відео з бази даних”, і ми отримуємо робочу версію програми. Ця архітектура описується у багатьох книгах, як те, до чого треба прагнути, оскільки сервери не зберігають стану і, відповідно, їх можна запускати і піднімати у будь-який момент часу. Проте навіть тут можна знайти недоліки:

- Велике навантаження на базу даних – кількість запитів на читання даних про поточний стан обробки відео є дуже великою, не всі бази даних можуть таке витримати. Реляційні бази зазвичай не підходять для великих навантажень, але вони й не потрібні, звичайного кешу (наприклад Redis) у цьому випадку повинно бути достатньо. У випадку використання черги цієї проблеми практично не існує, оскільки більшість черг спроектовані для великих навантажень, однак з’являється інша проблема – як дізнатись який з станів останній? Як правило, доводиться шукати компроміс між цілісністю даних і навантаженням, яке може витримати черга. На щастя, в нашому випадку не потрібно зберігати порядок даних, достатньо збереження порядку записів, внесених конкретним користувачем. Для цього дані кожного користувача можна направляти в окрему частину черги, яка зберігає внутрішній порядок (partition у випадку kafka, чи shard у випадку kinesis).

- У випадку, коли база даних чи черга знаходиться у іншому регіоні cloud, провайдер може брати плату за кількість переданих по публічній мережі даних. Цю проблему не можна вирішити не зберігаючи жодного стану на серверах (stateless application). Програми повинні зберігати дані про користувача

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		44

і про поточний стан обробки відео, проте у цьому випадку знову виникає проблема з балансувальником навантаження, який не обов'язково відправлятиме всі запити від одного і того ж самого користувача на один і той же сервер. Це може бути вирішено додаванням файлу cookie, який вказуватиме на який сервер необхідно передати запит. У цьому випадку балансувальник навантаження повинен вміти працювати з протоколом HTTP. У AWS вже існує такий сервіс – Application Load Balancer. Опція, яка нас цікавить – sticky session.

Розглянемо деякі методи конекторів, призначених для обробки REST запитів в таблиці 2.6

Таблиця 2.6 – Методи конекторів

Метод	HTTP Метод	Опис
UserController.createUser	POST	Створює нового користувача в процесі реєстрації, додає дані про користувача в реляційну базу даних
UserController.currentUser	GET	Повертає дані про поточного користувача, у випадку, коли запит був

Продовження таблиця 2.6

		зроблений незареєстрованим користувачем – повертає відповідь з кодом 404 (не знайдено)
VideoController.uploadVideo	POST	Відповідає за завантаження відео на сервер. Саме відео зберігається в S3, дані про відео (шлях, користувач, що додав, чи є публічним) записуються в реляційну базу даних
VideoController.downloadVideo	GET	Повертає відео
ImageController.uploadImage	POST	Відповідає за завантаження фото на сервер. Саме фото зберігається в S3, дані про відео (шлях,

Продовження таблиця 2.6

		користувач, що додав, чи є публічним) записуються в реляційну базу даних
ImageController.downloadImage	GET	Повертає фото
AudioController.uploadAudio	POST	Відповідає за завантаження аудіо на сервер. Саме аудіо зберігається в S3, дані про відео (шлях, користувач, що додав, чи є публічним) записуються в реляційну базу даних
AudioController.downloadAudio	GET	Повертає аудіо

2.4 Аналіз безпеки даних

Безпека даних гарантується фреймворком SpringSecurity та TLS [2], який налаштований на рівні ALB (Application Load Balancer). Проте, програма повинна гарантувати те, що у жодного користувача не буде доступу до даних інших користувачів. У разі втрати даних користувача,

вони не повинні нести жодної цінності для зловмисників. Для цього використовується шифрування даних в базі та на S3.

Висновки по розділу

У другому розділі були розглянуті вимоги до програмного продукту та висвітлена архітектура, аргументовані архітектурні рішення. Було розглянуто архітектуру фронтенду, який реалізований у вигляді Single Page Application за допомогою мови програмування JavaScript та описано основні класи фронтенду. Також були розглянуті сервіси бекенду. Сервіс обробки відео повинен витримувати значні навантаження у порівнянні з іншими сервісами (наприклад, з сервісом реєстрації), тому було детально описано можливі архітектурні рішення, розглянуті їх переваги та недоліки, описано оптимізації, призначені для зменшення навантаження на мережу, а, відповідно, й ціни при розміщенні програмного продукту на AWS.

Також, було вказано в таблицях основні класи та методи бекенду, описані їх функції.

					КПІ.ІТ-84В11.045440.02.81	Арк.
						48
Змін.	Арк.	№ докум.	Підп.	Дата.		

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Тестування програмного забезпечення - це процес перевірки, чи відповідає програмний продукт вимогам і не містить дефектів. Це включає запуск компонентів програми за допомогою ручних або автоматизованих інструментів для оцінки їх властивостей.

Метою тестування програмного забезпечення є виявлення помилок, прогалин або невідповідностей до вимог. Тестування виконується на різних етапах життєвого циклу розробки програми, оскільки помилки можуть виникнути на будь-якому етапі.

Забезпечення якості є важливим, оскільки допущені помилки можуть мати негативний вплив на продукт. Тестування допомагає виявити та виправити проблеми до того, як програмний продукт буде випущений. Правильно протестований продукт забезпечує надійність, безпеку та продуктивність, що приносить користь як розробнику, так і клієнту.

Тестування є невід'ємною частиною процесу розробки програмного забезпечення і допомагає уникнути високих витрат і проблем, пов'язаних з помилками програмного забезпечення. Цей процес дозволяє виявити та усунути потенційні проблеми, забезпечуючи якість та надійність програмного продукту перед його випуском. Важливо зауважити, що ефективне тестування передбачає ретельне планування, використання різноманітних тестових сценаріїв і врахування реальних умов використання програмного продукту. Такий підхід допомагає знизити ризик виявлення непередбачуваних проблем і забезпечити максимальну якість та задоволення користувачів.

3.2 Опис процесів тестування

Існує досить велика кількість видів тестування. Опис видів застосованих при розробці програмного продукту для даної дипломної роботи:

					КПІ.ІТ-84в11.045440.02.81	Арк.
						49
Змін.	Арк.	№ докум.	Підп.	Дата.		

– Unit tests – кількість цих тестів досить велика і вони гарантують, що ізолювані одиниці коду працюють належним чином. Ці тести повинні перевіряти одну просту вимогу і не покладатися на будь-які зовнішні залежності;

– Integration tests - тип тестування, коли програмні модулі логічно інтегровані та тестуються як група. Типовий програмний проєкт складається з кількох програмних модулів, закодованих різними програмістами. Метою цього рівня тестування є виявлення дефектів у взаємодії між цими програмними модулями, коли вони інтегровані;

– End-to-end tests – моделюють реальну роботу продукту, включаючи всі його частини.

У процесі тестування програмного забезпечення були перевірені функціональні вимоги, що наводилися в попередніх розділах, а саме:

- обрізання відео;
- склеювання відео;
- додавання аудіо;
- накладання відео на відео, відео на фото і навпаки.
- додавання фільтрів;
- виконувати базові ефекти.

Результати були оформлені в таблиці.

Таблиця 3.1 — Тест 1

Мета тесту	Перевірити функціональність обрізання відео
Початковий стан	Користувач увійшов у акаунт, відкритий проєкт, додане відео
Вхідні дані	-

Продовження таблиці 3.1

Опис проведення тесту	Необхідно обрати межі відео, використовуючи повзунок внизу екрану та натиснути кнопку “Обрізати”
Очікуваний результат	Через деякий час відео зменшилось в розмірі
Фактичний результат	Через деякий час відео зменшилось в розмірі

Таблиця 3.2 — Тест 2

Мета тесту	Склеювання відео
Початковий стан	Користувач увійшов у акаунт, відкритий проєкт, додано 2 відео
Вхідні дані	2 відео додані в проєкт
Опис проведення тесту	Необхідно обрати обидва відео на панелі знизу, затиснувши кнопку “Shift” натиснути кнопку ”Склеїти”
Очікуваний результат	Через деякий час відео об’єднались в одне на панелі знизу

Продовження таблиці 3.2

Фактичний результат	Через деякий час відео об'єднались в одне на панелі знизу
---------------------	---

Таблиця 3.3 — Тест 3

Мета тесту	Перевірити функціональність додавання фрагментів відео
Початковий стан	Користувач увійшов у акаунт, відкритий проєкт
Вхідні дані	Відео на комп'ютері
Опис проведення тесту	Необхідно обрати в меню пункт “Додати”, потім обрати потрібний файл і натиснути “Ок”
Очікуваний результат	Через деякий час відео завантажилось на сервер і відображається в проєкті
Фактичний результат	Через деякий час відео завантажилось на сервер і відображається в проєкті

Таблиця 3.4 — Тест 4

Мета тесту	Перевірити функціональність додавання аудіо
Початковий стан	Користувач увійшов у акаунт, відкритий проєкт
Вхідні дані	Аудіо на комп'ютері

Продовження таблиці 3.4

Опис проведення тесту	Необхідно обрати в меню пункт “Додати”, потім обрати потрібний файл і натиснути “Ок”
-----------------------	--

Таблиця 3.5 — Тест 5

Мета тесту	Перевірити функціональність переміщення відео
Початковий стан	Користувач увійшов у акаунт, відкритий проєкт, додане відео
Вхідні дані	—
Опис проведення тесту	Необхідно затиснути відео на панелі знизу лівою кнопкою мишки і перемістити його
Очікуваний результат	Відображається сторінка проєкту
Фактичний результат	Відображається сторінка проєкту

Таблиця 3.7 — Тест 7

Мета тесту	Перевірити функціональність входу вже зареєстрованого користувача
Початковий стан	Відкрита головна сторінка сайту
Вхідні дані	Пароль та логін вже зареєстрованого користувача
Опис проведення тесту	Необхідно ввести дані у відповідні поля, натиснути кнопку “Вхід”

Продовження таблиці 3.7

Очікуваний результат	Відображається сторінка проекту
Фактичний результат	Відображається сторінка проекту

Таблиця 3.8 — Тест 8

Мета тесту	Перевірити відображення кнопки виходу з акаунту
Початковий стан	Відкрита головна сторінка сайту
Вхідні дані	Пароль та логін вже зареєстрованого користувача
Опис проведення тесту	Необхідно ввести дані у відповідні поля, натиснути кнопку “Вхід”
Очікуваний результат	Зникли кнопки “Sign in” і “Sign up”, з’явилась кнопка “Sign out”
Фактичний результат	Зникли кнопки “Sign in” і “Sign up”, з’явилась кнопка “Sign out”

Таблиця 3.9 — Тест 9

Мета тесту	Перевірити додавання публічно доступних файлів
Початковий стан	Відкрита сторінка відеоредактора
Вхідні дані	Файл на локальній системі
Опис проведення тесту	Необхідно обрати пункт меню File потім Upload video, обрати файл та натиснути галочку “Public” після завантаження, натиснути

Продовження таблиці 3.9

	кнопку “Sign out” для виходу з акаунту, обрати пункт меню “Fragments”
Очікуваний результат	Файл доступний незареєстрованим користувачам (видимий серед фрагментів після виконання всіх кроків, описаних вище)
Фактичний результат	Файл доступний незареєстрованим користувачам (видимий серед фрагментів після виконання всіх кроків, описаних вище)

Таблиця 3.10 — Тест 10

Мета тесту	Перевірити додавання приватних файлів
Початковий стан	Відкрита сторінка відео редактора
Вхідні дані	Файл на локальній системі
Опис проведення тесту	Необхідно обрати пункт меню File потім Upload video, обрати файл, прослідкувати щоб чекбокс “Public” був вимкнений після завантаження, натиснути кнопку “Sign out” для виходу з акаунту, обрати пункт меню “Fragments”

Продовження таблиці 3.10

Очікуваний результат	Файл недоступний незареєстрованим користувачам (не видимий серед фрагментів після виконання всіх кроків описаних вище)
Фактичний результат	Файл недоступний незареєстрованим користувачам (не видимий серед фрагментів після виконання всіх кроків описаних вище)

Таблиця 3.11 — Тест 11

Мета тесту	Перевірити додавання публічно доступних проєктів
Початковий стан	Відкрита сторінка відео редактора
Вхідні дані	Файл на локальній системі
Опис проведення тесту	Необхідно обрати пункт меню File потім “Save” та натиснути галочку “Public”, потім натиснути кнопку “Sign out” для виходу з акаунту, обрати пункт меню “Projects”
Очікуваний результат	Проект доступний незареєстрованим користувачам (видимий серед фрагментів після виконання всіх кроків описаних вище)

Продовження таблиці 3.11

Фактичний результат	Проект доступний незареєстрованим користувачам (видимий серед фрагментів після виконання всіх кроків описаних вище)
---------------------	---

Таблиця 3.12 — Тест 12

Мета тесту	Перевірити додавання приватних файлів
Початковий стан	Відкрита сторінка відео редактора
Вхідні дані	Файл на локальній системі
Опис проведення тесту	Необхідно обрати пункт меню File потім “Save” прослідкувати щоб чекбокс “Public” був вимкнений, потім натиснути кнопку “Sign out” для виходу з акаунту, обрати пункт меню “Projects”
Очікуваний результат	Проект недоступний незареєстрованим користувачам (не видимий серед фрагментів після виконання всіх кроків описаних вище)
Фактичний результат	Проект недоступний незареєстрованим користувачам (не видимий серед фрагментів після виконання всіх кроків описаних вище)

3.3 Опис контрольного прикладу

Користувач створює проєкт на вкладці «Projects». Він вводить дані про назву проєкту та опис проєкту. Натискає кнопку «Create».



Create project

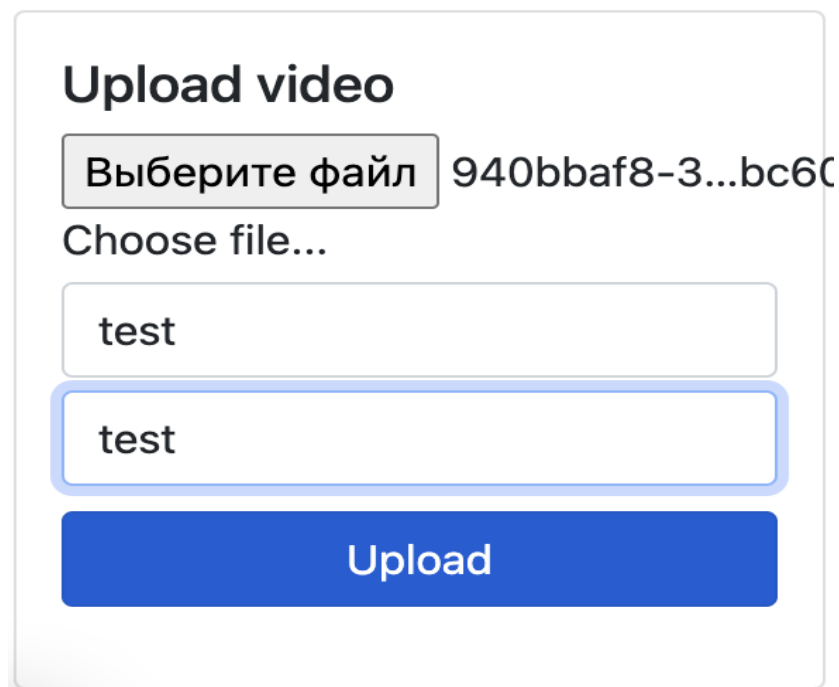
Name test

Description test

Create

Рисунок 3.1 – Форма створення нового проєкту

На вкладці «Fragments» користувач додає файли в бібліотеку, які будуть використовуватись в новому проєкті. Він вводить дані про назву файлу та опис файлу. Натискає кнопку «Upload».



Upload video

Выберите файл 940bbaf8-3...bc60

Choose file...

test

test

Upload

Рисунок 3.2 – Форма додавання в бібліотеку нових файлів

Після того, як користувач додав файл, він відображається в бібліотеці відеоредактора, та користувач може використати його у проєкті, натиснувши кнопку «Use in project». Також користувач може скористатись кнопкою «Delete» для того, щоб видалити файл з бібліотеки відеоредактора. Користувач не може видалити фрагмент, якщо він використовується в проєкті.

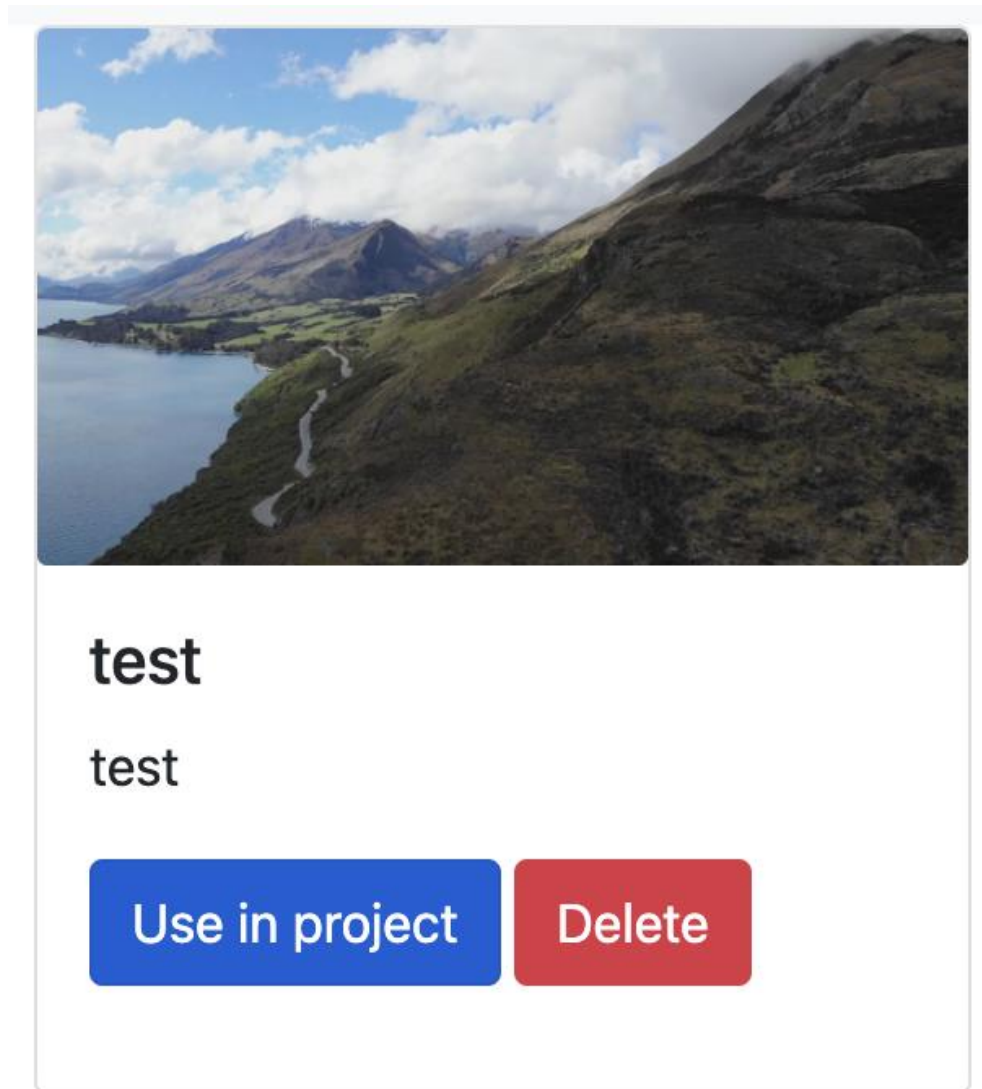


Рисунок 3.3 – Вікно відображення доданого файлу

Після того, як користувач обирає фрагмент, який хоче використати у проєкті, він переходить на вкладку «Editor» та бачить перед собою робочу область для редагування відеофайлів. Справа – інструменти для обробки відео (обрізання відео, склеювання відео, застосування ефектів, зберігання результатів та рендеринг відео). Зліва – таймлайн з повзунком для переміщення у часі та вікно попереднього перегляду.

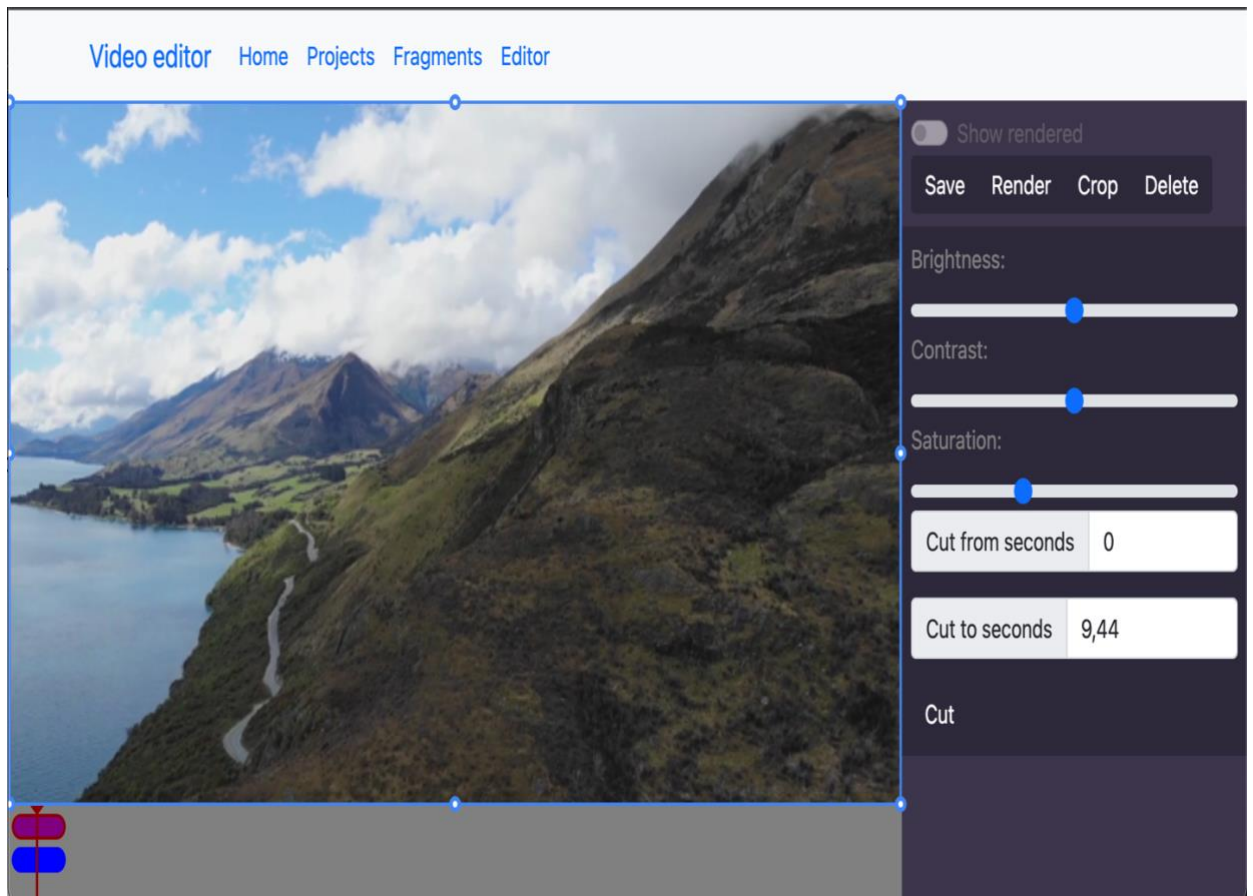


Рисунок 3.4 – Робоча область

Користувач може обрізати фрагмент відео двома способами. Перший спосіб – вручну задати межі, використовуючи кнопки «Cut from seconds» та «Cut to seconds». Другий спосіб – визначити місце розділення відео за допомогою повзунка на таймлайні і натиснути кнопку «Cut»

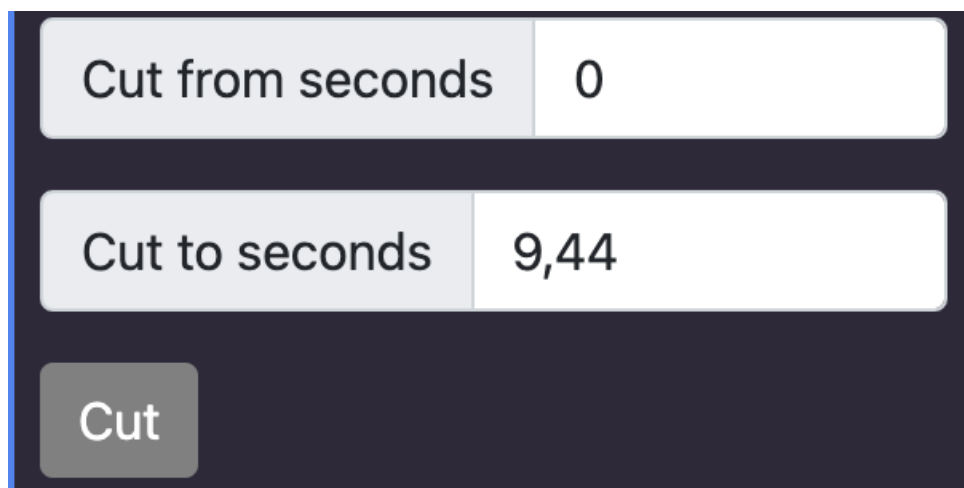
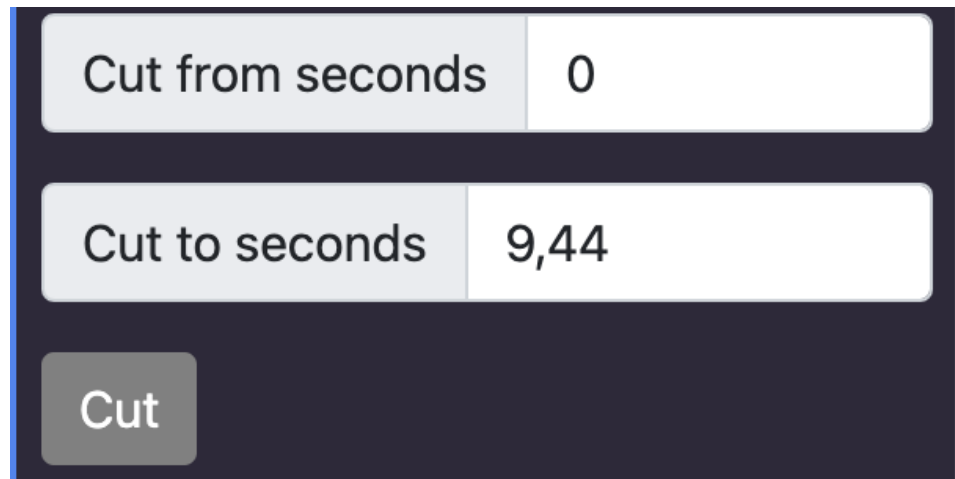


Рисунок 3.5 – Інструменти для обрізання відео

Користувач обрізає відео та може видалити з таймлайну непотрібні йому елементи за допомогою кнопки «Delete».



Cut from seconds 0

Cut to seconds 9,44

Cut

Рисунок 3.5 – Інструменти для обрізання відео

Користувач обрізає відео та може видалити з таймлайну непотрібні йому елементи за допомогою кнопки «Delete».



Рисунок 3.6 – Таймлайн

Потім користувач натискає кнопку «Render» та очікує на результат.



Рисунок 3.7 – Кнопка рендерингу

Коли рендеринг завершено, користувач отримує повідомлення про це.



Рисунок 3.8 – Вікно повідомлення про завершення рендерингу

Натиснувши перемикач «Show rendered», користувач може передивитись одержаний результат та завантажити його на свій пристрій.



Рисунок 3.9 – Перемикач «Show rendered»

Користувач передивляється одержаний результат у вікні перегляду.

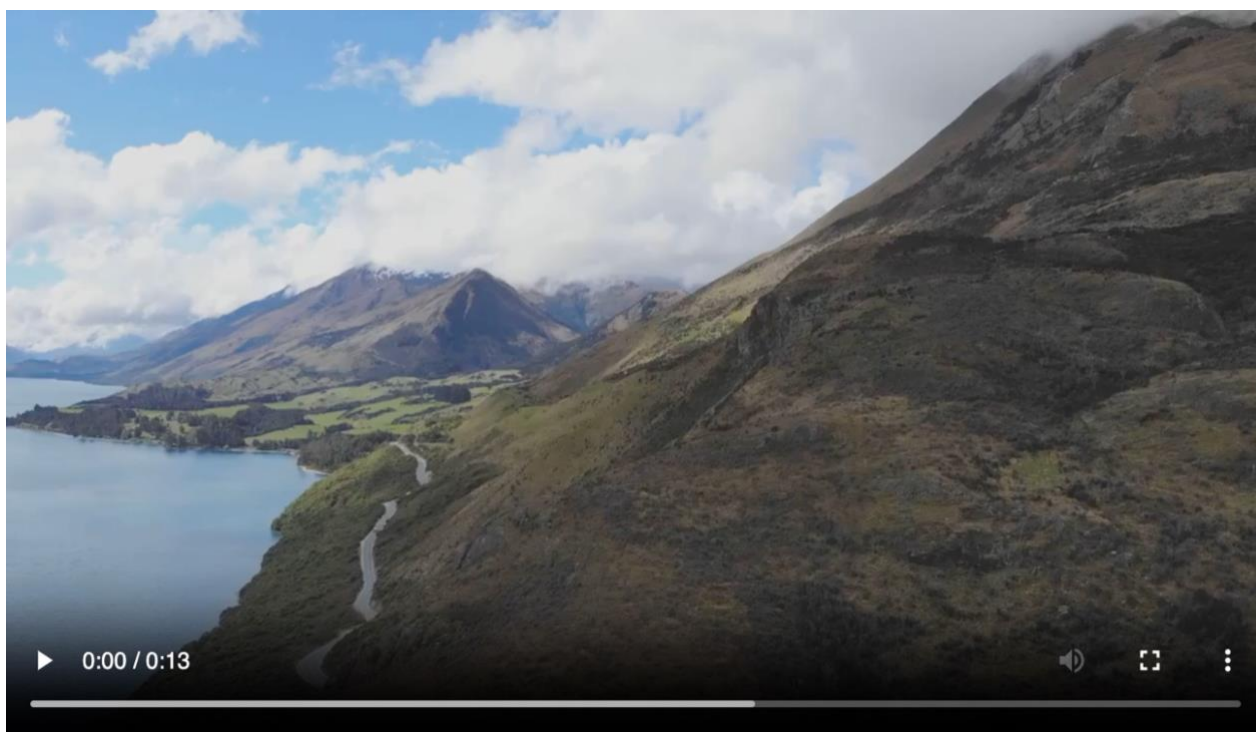


Рисунок 3.10 – Вікно перегляду одержаного результату

					КПІ.ІТ-84в11.045440.02.81	Арк.
						62
Змін.	Арк.	№ докум.	Підп.	Дата.		

Висновки по розділу

У цьому розділі було проаналізовано процес тестування в загальному та специфіку тестування програми, розробленої для даної дипломної роботи. Було описано основні види тестів, які використовуються в сучасному ІТ та їх призначення. Оскільки, кількість модульних та інтеграційних тестів, як правило, є значною, були вказані лише end-to-end тести.

Всього цих тестів 12 і вони призначені для перевірки правильності взаємодії всіх частин системи. Їх можна розбити на 3 основні групи:

- Перевірка правильності роботи відеоредактора (тести 1-5);
- Перевірка роботи з файлами, проєктами та авторизації доступу до них (тести 6-8);
- Перевірка процесів реєстрації на автентифікації (тести 9-12).

					КПІ.ІТ-84В11.045440.02.81	Арк.
						63
Змін.	Арк.	№ докум.	Підп.	Дата.		

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

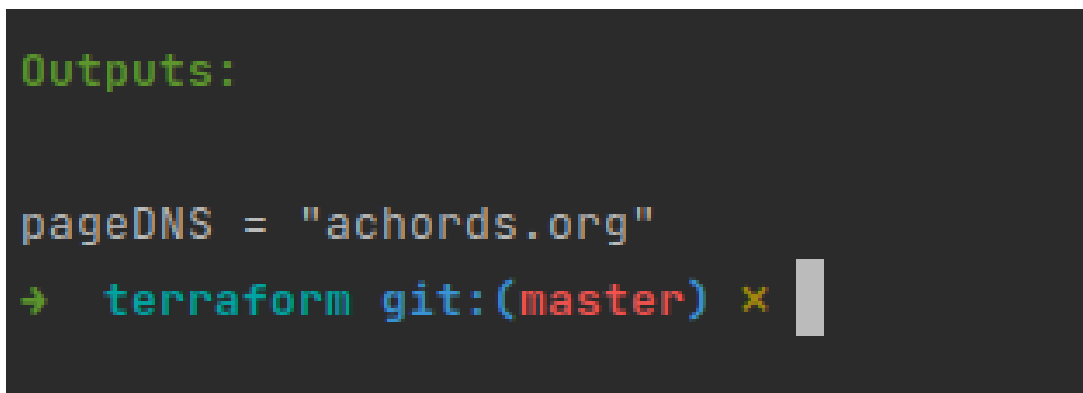
Дане програмне забезпечення розроблене для запуску на AWS Cloud. Необхідно встановити AWS CLI і Terraform, слідуючи інструкції для вашої операційної системи, створити IAM роль на AWS та налаштувати локальний aws клієнт для використання цієї ролі. Після чого в папці `infra` з репозиторію програми запуснути наступні команди:

- `terraform init` – ініціалізація робочого місця terraform;
- `terraform validate` – перевірка коду IaC на помилки;
- `terraform apply -auto-approve` – розгортання програми на AWS.

Для локального запуску програми (у випадку дебагу) можна використати файл `docker-compose.yml`, який лежить в корені репозиторія. Для цього необхідно встановити на локальний комп'ютер програми `docker` та `docker-compose` та запуснути команду `docker-compose up` в корені репозиторія.

4.2 Підтримка програмного забезпечення

Після успішного запуску програми через Terraform, у секції `Outputs` буде виведено DNS ім'я, яке можна використати для доступу до веб-інтерфейсу, яке продемонстровано на рисунку нижче:



```
Outputs:  
  
pageDNS = "achords.org"  
→ terraform git:(master) x
```

Рисунок 4.1 – Ім'я для доступу до веб-серверу

Перше, що ми можемо побачити після входу на веб-сайт – сторінка з описом веб-додатку та пропозицією увійти чи зареєструватись. Натиснувши на кнопку “Sign up”, яка знаходиться в меню вгорі, яке можна побачити на рисунку, що наведено нижче:

Video editor Home Projects Fragments Editor

Рисунок 4.2 – Меню веб-додатку

Переходимо на сторінку реєстрації, вводимо дані (логін, пароль) та тиснемо кнопку “Sign up”, яка зображена на рисунку 4.3:

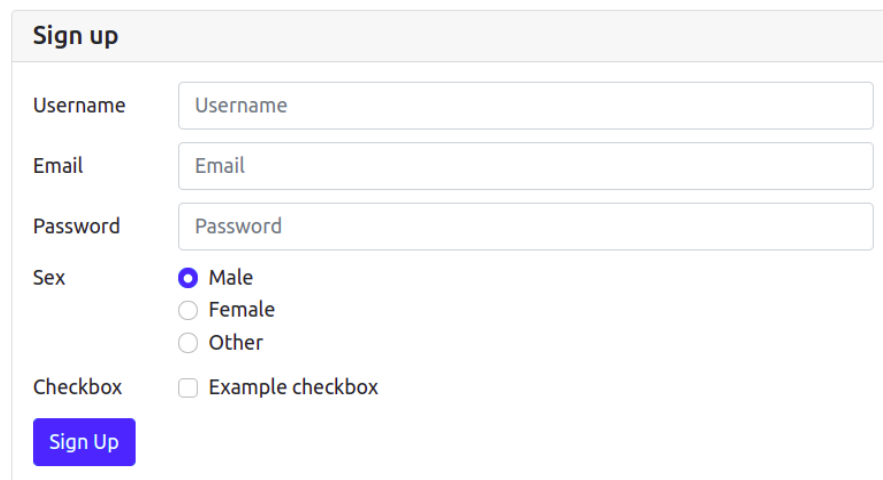
A registration form titled "Sign up" with a light gray header. It contains five input fields: "Username", "Email", "Password", "Sex" (with radio buttons for "Male", "Female", and "Other"), and "Example checkbox". A blue "Sign Up" button is at the bottom left.

Рисунок 4.3 – Форма реєстрації на сайті

У випадку, коли у користувача вже існує акаунт, необхідно натиснути кнопку «Sign in», ввести дані та знову натиснути кнопку «Sign in»:

A login form titled "Sign in" with a light gray header. It contains two input fields: "Email" and "Password". A blue "Sign In" button is at the bottom left.

Рисунок 4.4 – Форма для входу в існуючий акаунт

Для того, щоб розпочати роботу після входу, необхідно створити новий проєкт, написавши його назву та опис, та натиснути кнопку «Create».

Рисунок 4.5 – Панель створення нового проєкту

Також можна обрати існуючий, натиснувши на його віджет на екрані. Після чого відкриється сам проєкт. Список проєктів можна знайти в пункті меню «Projects» вгорі екрану, що продемонстровано на рисунку 4.6:

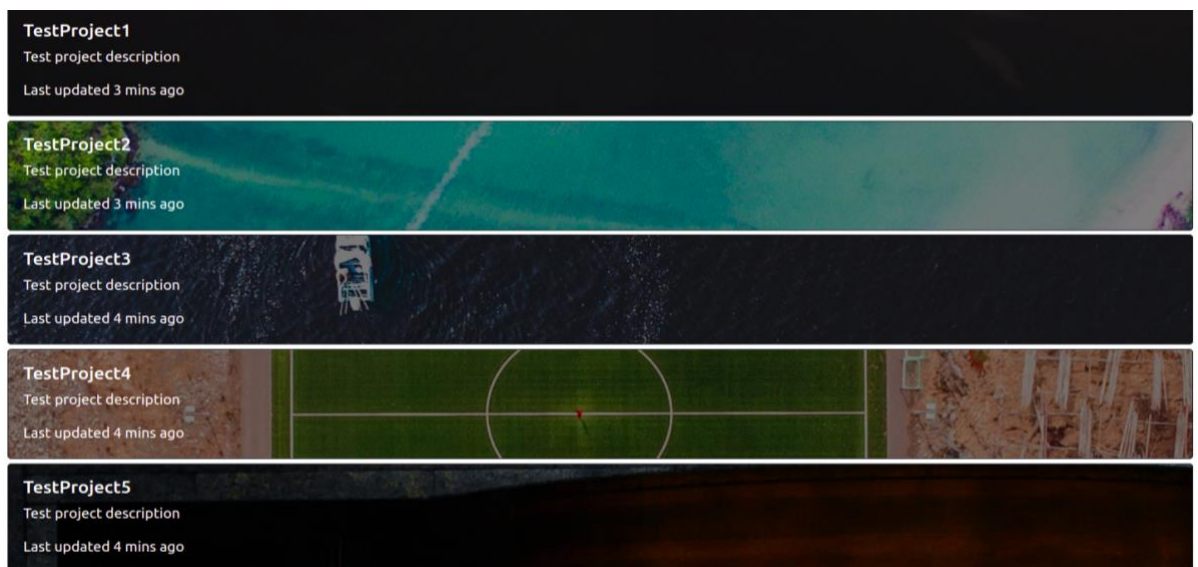


Рисунок 4.6 – Список існуючих проєктів

В центрі екрану – робоча область, яка продемонстрована на рисунку 4.7. Тут знаходиться зображення готового відео, а також фрагменти, з яких воно було створене. Для перемикання відео/фрагмент можна використати кнопку “Video/Fragments” в правому верхньому куті екрану. У випадку активного “Video” на екрані буде зображений готовий відеофайл (трансляція з сервера), у випадку активного “Fragments” відображатимуться фрагменти, з яких змонтоване відео. Для переміщення фрагменту необхідно його обрати подвійним

натисненням лівої кнопки мишки, затиснути ліву кнопку мишки та тягнути до необхідної позиції.

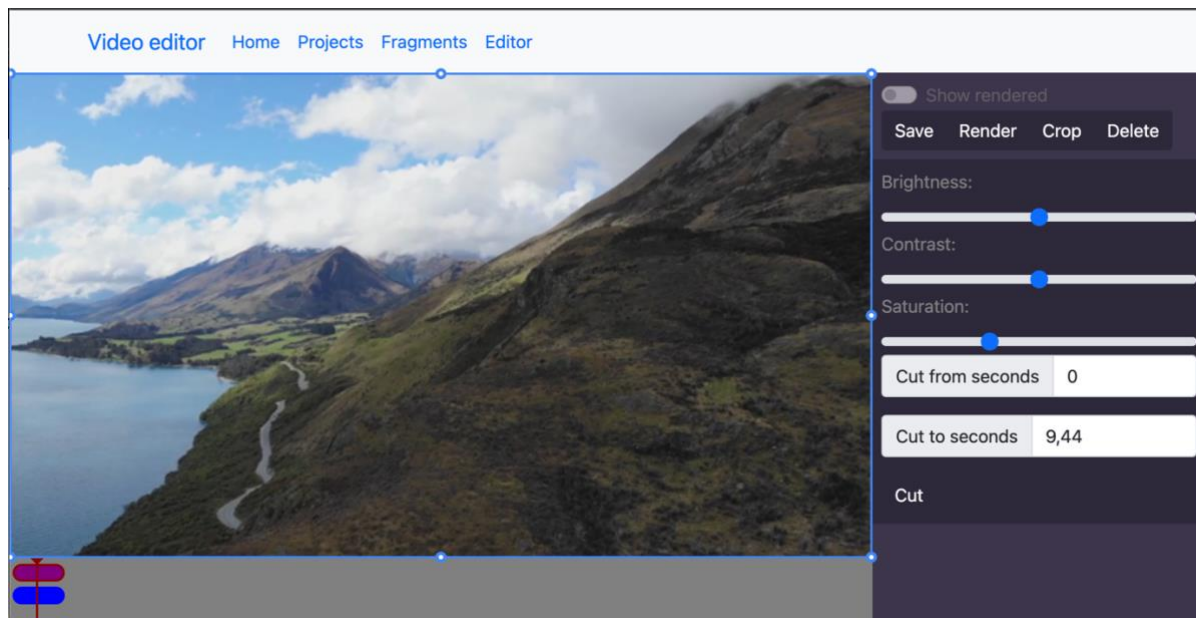


Рисунок 4.7 – Робоча область

Справа знаходиться панель налаштувань, внизу зліва - панель шарів відео, що зображені на рисунку 4.8. Панель налаштувань дозволяє змінювати параметри відео (розміри, позиція) більш точно, а також додавати ефекти (наприклад, контрастність). Оскільки відео можуть накладатися, не завжди зручно їх обирати в робочій області. Для вирішення цієї проблеми була запрограмована панель шарів. Кожен елемент має свій шар, які відображаються на панелі у вигляді списку, тому їх зручно обирати. Відео та фото (далі - фрагменти), які знаходяться на панелі вище мають вищий пріоритет, тому перекриватимуть фрагменти, які знаходяться в тій зоні в робочій області.

Внизу знаходиться таймлайн, який також являє собою список всіх фрагментів. Його особливістю є те, що він, також, відображає розміщення елементу в часі відносно початку відео. Тут можливо проводити такі маніпуляції як обрізання відео, розбиття відео на фрагменти, склеювання фрагментів.

Для розбиття відео на фрагменти необхідно обрати потрібне відео (фрагмент) на часолінії або на панелі шарів і натиснути кнопку «Cut».



Рисунок 4.8 – Панель інструментів для редагування відео

Після чого в обох формах (на часолінії та на панелі шарів) з'являться фрагменти.



Рисунок 4.9 – Панель шарів відео

Останній елемент інтерфейсу – вкладка «Fragments». Вона дозволяє додавати елементи (відео, фото, аудіо) в проєкт (вони будуть завантажені на сервер), зберігати та експортувати проєкт.

Для того, щоб загрузити фрагмент відео, аудіо чи фото на сервер, необхідно у панелі завантаження файлів, яка продемонстровано на рисунку 4.10,

натиснути «Оберіть файл» та обрати з файлової системи пристрою потрібний фрагмент, написати назву та опис файлу та натиснути кнопку «Upload».

Upload video

Выберите файл

940bbaf8-3...bc6

Choose file...

test

test

Upload

Рисунок 4.10 – Панель завантаження файлів

Після чого відкриється меню (специфічне для файлової системи), в якому необхідно обрати потрібний файл і натиснути кнопку «Open», що зображено на рисунку нижче:

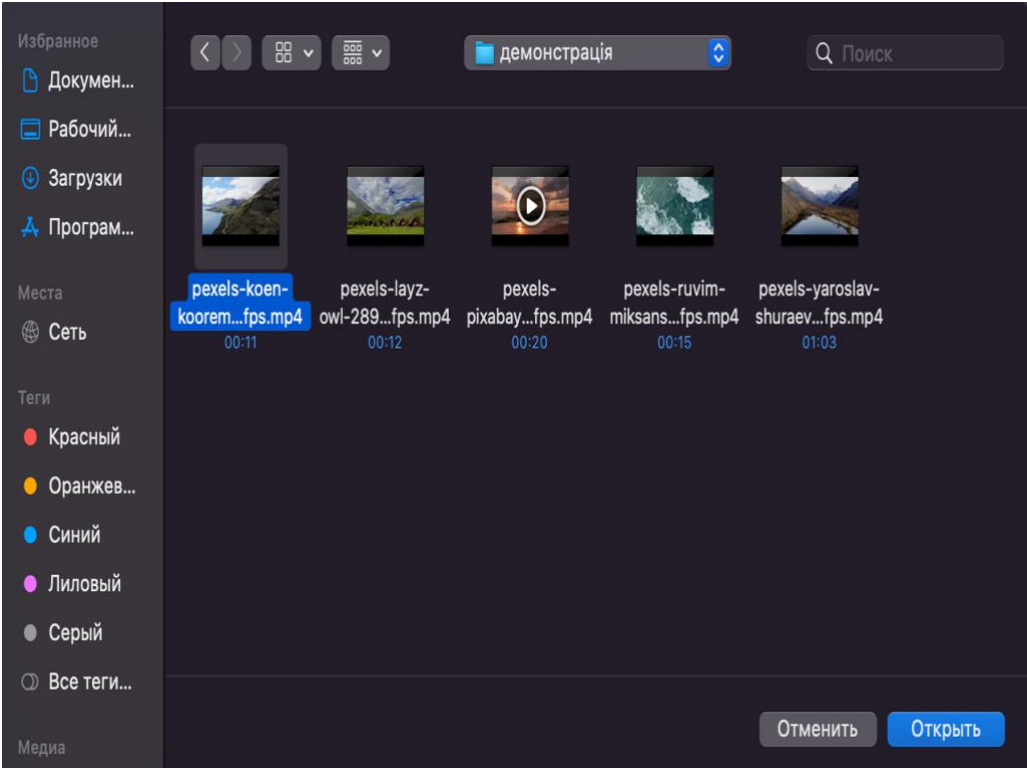


Рисунок 4.11 – Меню для вибору потрібного файлу

Для того, щоб сформувати відео необхідно обрати пункт меню «Render», який зображено на рисунку нижче, і натиснути кнопку.

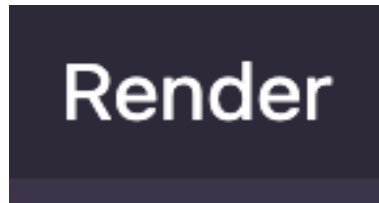


Рисунок 4.12 – Кнопка «Render»

Після чого розпочнеться рендеринг відео та по завершенню цього процесу, користувач отримує повідомлення про те, що рендеринг завершено.



Рисунок 4.13 – Повідомлення про закінчення рендерингу

Після закінчення експорту можна натиснути на перемикач «Show rendered» та передивитись отриманий результат. У програвачі можна натиснути на три крапки та завантажити результат на пристрій користувача.

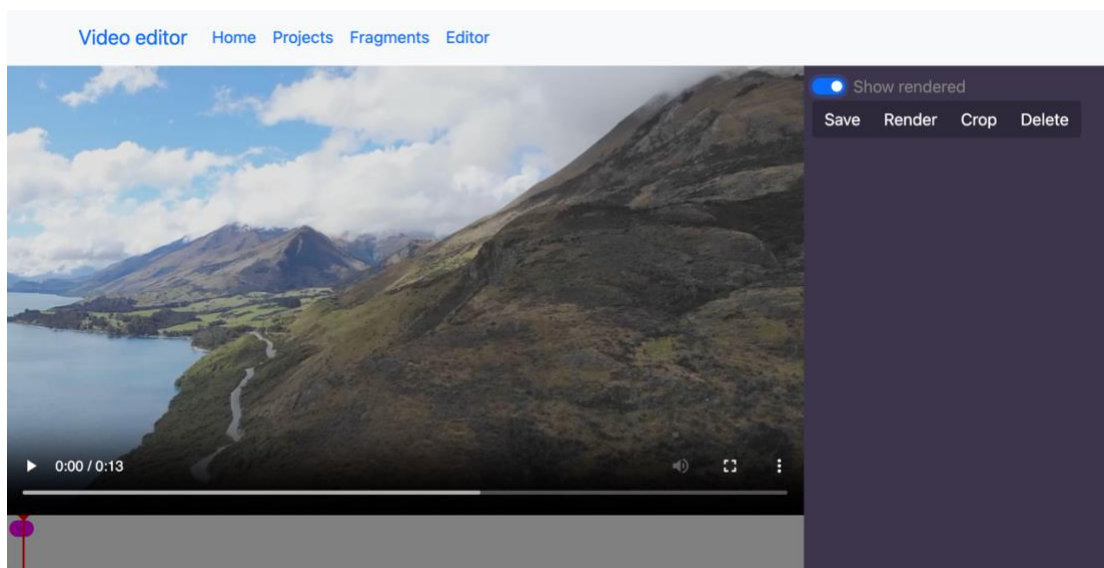


Рисунок 4.14 – Вікно перегляду отриманого результату

					КПІ.ІТ-84в11.045440.02.81	Арк.
						70
Змін.	Арк.	№ докум.	Підп.	Дата.		

Висновки по розділу

У четвертому розділі було розглянуто способи розгортання додатку (локально, на AWS з використанням Terraform). Продемонстровано можливості розробленого програмного продукту, а саме:

- реєстрацію користувачів;
- додавання відео, фото та аудіо файлів;
- створення проєктів;
- редагування відео;
- рендеринг готового відео.

Було наведено екранні форми результату процесу деплою (DNS ім'я програми відображається після завершення розгортання програми на AWS у секції “Outputs”) та екранні форми процесу роботи з програмою.

					КПІ.ІТ-84В11.045440.02.81	Арк.
						71
Змін.	Арк.	№ докум.	Підп.	Дата.		

ВИСНОВКИ

Під час виконання дипломного проєкту було розглянуто сферу редагування відео, наведено приклади готових програмних продуктів призначених для редагування відео, розглянуто їх особливості, переваги та недоліки. Спільним для більшості таких редакторів є те, що вони виконують рендеринг відео на клієнтському комп'ютері, тому мінімальні вимоги до характеристик комп'ютера для нормальної роботи є дещо завищеними. Якщо ж перенести частину програми відповідальну за обробку відео на сервер, а на клієнтській стороні залишити лише стрімінг відео та передачу команд на сервер, програму можна буде запускати на значно старіших моделях комп'ютерів. У такому випадку сам клієнтський комп'ютер буде виконувати функцію демонстрації відео, таких сервісів існує немало і, провівши аналогію з ними, можна оцінити наскільки збільшиться час автономної роботи пристрою у порівнянні з класичними відеоредакторами.

В результаті виконання дипломного проєкту було розроблено програмний продукт, що відповідає описаним вимогам. В якості середовища розробки браузерної частини було використано WebStorm, який має досить багато можливостей для зручного редагування JavaScript, html, css та перегляду результату в браузері в режимі реального часу. Сама браузерна частина була розроблена з використанням мови програмування JavaScript для реалізації Single Page Application та Bootstrap, який має велику кількість віджетів та дозволяє кастомізувати їх, встановлюючи теми.

В якості середовища розробки серверної частини було використано IntelliJ Idea, що має чудову підтримку мови програмування Java. Фреймворк

Spring використовувався для веб інтерфейсів, роботи з базою даних та підвищення якості коду (стимулює використання інверсії залежностей). Сервіси бекенду розгортаються в хмарі на AWS (ті що написані на Java – на EC2). Оскільки навантаження на сервіс редагування відео є досить великим, було вирішено використовувати балансувальник навантаження, який розподіляє

					КПІ.ІТ-84в11.045440.02.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		72

запити між усіма серверами. Переваги та недоліки цього підходу були описані у пояснювальній записці роботи.

У якості бази даних було використано `mongoDB`. Ця база добре масштабується.

У якості місця зберігання файлів завантажених користувачем було використано сервіс `AWS S3`, який дозволяє завантажувати файли розміром до 5ТБ, чого більше, ніж достатньо для даної роботи.

У третьому розділі було розглянуто процес тестування у загальному та конкретно цього програмного продукту. Було наведено 12 end-to-end тестів, а саме для перевірки:

- роботи відеоредактора;
- роботи з файлами, проєктами та авторизації доступу до них;
- процесів реєстрації на автентифікації.

У четвертому розділі було описано процес розгортання програмного продукту та розглянуто його основні можливості з наведеними екранними формами.

					КПІ.ІТ-84в11.045440.02.81	Арк.
						73
Змін.	Арк.	№ докум.	Підп.	Дата.		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Chris Richardson. Microservices Patterns: With examples in Java / Ch. Richardson. – Shelter Island: Publishing house Manning Publications Co, 2018. – 520 p.
- 2) John Carrel. Spring Microservices in Action / J. Carrel – Shelter Island Publishing house Manning Publications, 2017 – 384 p.
- 3) Sam Newman. Building Microservices: Designing Fine-Grained Systems / S. Newman – Newton: Publishing house O'Reilly Media, 2015 – 500 p.
- 4) Sam Newman – Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith / S. Newman – Newton: Publishing house O'Reilly Media, 2019 – 272 p.
- 5) Microservices: Yesterday, Today, and Tomorrow [Електронний ресурс]: – (Стаття) / Present and Ulterior Software Engineering – Електрон. дан. (1 файл). – 2017. – Режим доступу до ресурсу: <https://www.springer.com/series/558>. - Назва з екрану.
- 6) The Design and Architecture of Microservices [Електронний ресурс]: (Стаття) / IEEE Cloud Computing – Електрон. дан. (1 файл). – 2016. – Режим доступу: <https://ieeexplore.ieee.org/abstract/document/7742259>. – Назва з екрану.
- 7) Amazon Web Services Documentation [Електронний ресурс]:– Режим доступу до ресурсу: <https://docs.aws.amazon.com>
- 8) Spring Framework Documentation [Електронний ресурс]: – Режим доступу до ресурсу: <https://docs.spring.io>
- 9) Node.js Documentation [Електронний ресурс]: – Режим доступу до ресурсу: <https://nodejs.org/api/documentation.html>

ДОДАТОК А ЗВІТ ПОДІБНОСТІ



Ім'я користувача:
Лісовиченко Олег Іванович

ID перевірки:
1015544201

Дата перевірки:
10.06.2023 21:00:10 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2023 21:00:25 EEST

ID користувача:
76913

Назва документа: ІТ-84в_Гюржи_ПЗ

Кількість сторінок: 70 Кількість слів: 8266 Кількість символів: 61973 Розмір файлу: 8.12 MB ID файлу: 1015196829

4.5% Схожість

Найбільша схожість: 1.02% з Інтернет-джерелом (<https://ela.kpi.ua/handle/123456789/31087>)

2.2% Джерела з Інтернету	30	Сторінка 72
3.94% Джерела з Бібліотеки	176	Сторінка 72

0% Цитат

- Вилучення цитат вимкнене
- Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	1
------------------	---

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕДАГУВАННЯ ВІДЕОФАЙЛІВ

Текст програми

КП.ІТ-84в11.045440.03.12

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Юрій ОЛІЙНИК

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Євгенія ГЮРЖИ

Київ – 2023

VideoThumbnailService.java

```
public Path createVideoThumbnail(Path videoPath, VideoInfo info) {
    try {
        final Path thumbnailPath =
            Path.of(storageProperties.getTempFilePrefix())
                .resolve(UUID.randomUUID() + ".png");
        String formatted = COMMAND_TEMPLATE.formatted(properties.getFfmpeg(),
            videoPath, thumbnailPath);
        final Process process = Runtime.getRuntime().exec(formatted);
        ProcessUtil.throwOnError(process);
        return thumbnailPath;
    } catch (IOException e) {
        throw new StorageException(e);
    }
}
```

VideoRenderingService.java

```
private void downloadVideoFromS3(ProjectVideoInfo projectVideoInfo, VideoInfo
videoInfo, Path tempPath) {
    Path uncutPath =
        createTempFile(VideoUtil.getExtension(tempPath.getFileName().toString(), false));
    s3Client.getObject(builder -> builder
        .bucket(storageProperties.getS3().getBucketName())
        .key(videoInfo.getStorageInfo().getVideoS3Key()),
        uncutPath);
    if (videoInfo.getFormat().isPicture()) {
        final Path oldUncutPath = uncutPath;
        uncutPath = createVideoFromImage(uncutPath,
            projectVideoInfo.getDurationSeconds());
        Files.delete(oldUncutPath);
    }
    cutVideo(uncutPath, tempPath, projectVideoInfo.getCutFromSecond(),
        projectVideoInfo.getCutToSecond());
    Files.delete(uncutPath);
}

private void downloadAudioFromS3(ProjectAudioInfo projectAudioInfo, VideoInfo
videoInfo, Path tempPath) {
    final Path uncutPath =
        createTempFile(VideoUtil.getExtension(tempPath.getFileName().toString(), false));
    s3Client.getObject(builder -> builder
        .bucket(storageProperties.getS3().getBucketName())
        .key(videoInfo.getStorageInfo().getVideoS3Key()),
        uncutPath);
    cutVideo(uncutPath, tempPath, projectAudioInfo.getCutFromSecond(),
        projectAudioInfo.getCutToSecond());
    Files.delete(uncutPath);
}

private void cutVideo(Path fromPath, Path toPath, double cutFromSecond, double
cutToSecond) {
    final String cmd = "%s -y -i %s -ss %f -to %f %s".formatted(
        ffmpegProperties.getFfmpeg(), fromPath, cutFromSecond, cutToSecond,
        toPath);
}
```

					КПІ.ІТ-84В11.045440.03.12	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

        ProcessUtil.throwOnError(Runtime.getRuntime().exec(cmd));
    }

    private Path createVideoFromImage(Path imagePath, double videoDuration) {
        final Path resultPath = Path.of(storageProperties.getTempFilePrefix(),
            UUID.randomUUID() + ".mp4");
        final String cmd = "%s -framerate 1 -loop 1 -t %f -i %s -vf
pad=ceil(iw/2)*2:ceil(ih/2)*2 -c:v libx264 -r 30 %s".formatted(
            ffmpegProperties.getFfmpeg(), videoDuration, imagePath, resultPath);
        log.info("Generating video from image: {}", cmd);
        final Process process = Runtime.getRuntime().exec(cmd);
        ProcessUtil.throwOnError(process);

        return resultPath;
    }

    public String render(UUID projectId) {
        var projectInfo = projectInfoRepository.findAll().get(0);
        final Map<ProjectVideoInfo, VideoInfo> videoInfoMapping =
            projectInfo.getVideoInfos().stream()
                .collect(toOrderedMap(Function.identity(), videoInfo ->
                    videoInfoRepository.findById(videoInfo.getVideoInfoId()).orElseThrow()));

        final Map<ProjectAudioInfo, VideoInfo> audioInfoMapping =
            projectInfo.getAudioInfos().stream()
                .collect(toOrderedMap(Function.identity(), audioInfo ->
                    videoInfoRepository.findById(audioInfo.getVideoInfoId()).orElseThrow()));

        final Map<ProjectAudioInfo, Path> projectAudioInfoToTempPath =
            audioInfoMapping.entrySet().stream()
                .collect(toOrderedMap(
                    Map.Entry::getKey,
                    entry -> createTempFile(VideoUtil.getExtension(
                        entry.getValue().getStorageInfo().getVideoS3Key(),
                        entry.getValue().getFormat().isPicture()))));

        final Map<ProjectVideoInfo, Path> projectVideoInfoToTempPath =
            videoInfoMapping.entrySet().stream()
                .collect(toOrderedMap(
                    Map.Entry::getKey,
                    entry -> createTempFile(VideoUtil.getExtension(
                        entry.getValue().getStorageInfo().getVideoS3Key(),
                        entry.getValue().getFormat().isPicture()))));

        videoInfoMapping.forEach((projectVideoInfo, videoInfo) ->
            downloadVideoFromS3(projectVideoInfo, videoInfo,
                projectVideoInfoToTempPath.get(projectVideoInfo)));

        audioInfoMapping.forEach((projectAudioInfo, videoInfo) ->
            downloadAudioFromS3(projectAudioInfo, videoInfo,
                projectAudioInfoToTempPath.get(projectAudioInfo)));

        final Path renderedLocalLocation = Path.of(ffmpegService.render(
            videoInfoMapping, audioInfoMapping, projectVideoInfoToTempPath,
            projectAudioInfoToTempPath));
        final String renderedS3Key = storageProperties.getS3().getRenderedKeyPrefix()
            + "/" + UUID.randomUUID() + ".mp4";

        // Upload rendered video to s3
    }

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

```

        s3Client.putObject(
            builder ->
builder.bucket(storageProperties.getS3().getBucketName()).key(renderedS3Key),
            renderedLocalLocation);

        // Update projectInfo
        final String renderedS3Url = storageProperties.getS3().getEndpointUrl() + "/"
+ renderedS3Key;
        final ProjectInfo project =
projectInfoRepository.findById(projectId).orElseThrow();
        final String lastRenderedS3Key = project.getLastRenderedS3Key();
        project.setLastRenderedS3Key(renderedS3Key);
        project.setLastRenderedUrl(renderedS3Url);
        projectInfoRepository.save(project);

        // Remove previously rendered project to reduce storage size
        if (Objects.nonNull(lastRenderedS3Key)) {
            s3Client.deleteObject(builder ->
builder.bucket(storageProperties.getS3().getBucketName()).key(lastRenderedS3Key));
        }

        // Delete local files
        cleanAll(projectVideoInfoToTempPath.values());
        cleanAll(projectAudioInfoToTempPath.values());
        Files.delete(renderedLocalLocation);

        return presignKey(renderedS3Key);
    }

    public String presignKey(String lastRenderedS3Key) {
        if (Objects.nonNull(lastRenderedS3Key)) {
            return s3Presigner.presignGetObject(pr -> pr.getObjectRequest(gr -> gr
                .bucket(storageProperties.getS3().getBucketName())
                .key(lastRenderedS3Key))
                .signatureDuration(Duration.ofHours(2)))
                .url().toString();
        }

        return null;
    }
}

```

ProjectAudioInfo.java

```

public class ProjectAudioInfo {
    @Id
    @EqualsAndHashCode.Include
    private int id;
    private String displayColor;
    private double cutFromSecond;
    private double cutToSecond;
    private double startSeconds;
    private double durationSeconds;

    // Storate location
    private UUID videoInfoId;
    private long streamId;
}

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

ProjectInfo.java

```
public class ProjectInfo {
    @Id
    private UUID id;

    /**
     * User-specified project name
     */
    private String name;

    /**
     * User-specified project description
     */
    private String description;

    /**
     * Randomly selected project background image
     */
    private String backgroundImageLocation;

    private ProjectSize projectSize;

    private List<ProjectVideoInfo> videoInfos;
    private List<ProjectAudioInfo> audioInfos;

    private String lastRenderedUrl;
    private String lastRenderedS3Key;
    private String lastUpdated;
    public UUID owner; // public if null
}
```

ProjectInfoRepository.java

```
public interface ProjectInfoRepository extends MongoRepository<ProjectInfo, UUID> {
}
```

UserRepository.java

```
public interface UserRepository extends MongoRepository<UserInfo, UUID> {
    UserInfo findByUsername(String username);
}
```

VideoInfoRepository.java

```
public interface VideoInfoRepository extends MongoRepository<VideoInfo, UUID> {
}
```

ProjectService.java

```
public String render(UUID projectId) {
    String renderedVideoLocation = videoRenderingService.render(projectId);
    log.info("Rendered: " + videoRenderingService);
    return renderedVideoLocation;
}
```

					КПІ.ІТ-84В11.045440.03.12	Арк.
						5
Змін.	Арк.	№ докум.	Підп.	Дата.		

FFMpegService.java

```
public String render(Map<ProjectVideoInfo, VideoInfo> videoInfoMapping,
                    Map<ProjectAudioInfo, VideoInfo> audioInfoMapping,
                    Map<ProjectVideoInfo, Path> projectVideoInfoToTempPath,
                    Map<ProjectAudioInfo, Path> projectAudioInfoToTempPath) {
    final String outputPath = Path.of(storageProperties.getTempFilePrefix(),
    UUID.randomUUID() + ".mp4").toString();
    final FFMpeg ff = new FFMpeg(ffMpegProperties);
    FFMpeg.Marker lastOverlay = ff.addVideoInput(backgroundPath);
    double maxSecondVideo = videoInfoMapping.keySet().stream()
        .mapToDouble(projectVideoInfo ->
projectVideoInfo.getStartSeconds() +
                    (projectVideoInfo.getCutToSecond() -
projectVideoInfo.getCutFromSecond()))
        .max()
        .orElse(0);

    double maxSecondAudio = audioInfoMapping.keySet().stream()
        .mapToDouble(projectAudioInfo ->
projectAudioInfo.getStartSeconds() +
                    (projectAudioInfo.getCutToSecond() -
projectAudioInfo.getCutFromSecond()))
        .max()
        .orElse(0);

    double maxSecond = Math.max(maxSecondVideo, maxSecondAudio);

    FFMpeg.Marker background = ff.timePad(lastOverlay, maxSecond,
"#00000011");

    for (Map.Entry<ProjectVideoInfo, VideoInfo> entry :
videoInfoMapping.entrySet()) {
        final ProjectVideoInfo projectVideoInfo = entry.getKey();
        FFMpeg.Output scaled = ff.scale(

ff.addVideoInput(projectVideoInfoToTempPath.get(projectVideoInfo).toString()),
                    projectVideoInfo.getWidth(),
                    projectVideoInfo.getHeight());

        FFMpeg.Output cropped = ff.crop(
                    scaled, projectVideoInfo.getCropX(),
projectVideoInfo.getCropY(),
                    Math.min(projectVideoInfo.getCropWidth(),
projectVideoInfo.getWidth()),
                    Math.min(projectVideoInfo.getCropHeight(),
projectVideoInfo.getHeight()));

        FFMpeg.Output eq = ff.eq(cropped,
                    projectVideoInfo.getBrightness(),
                    projectVideoInfo.getContrast(),
                    projectVideoInfo.getSaturation());

        // FFMpeg.Output trim = ff.trim(eq,
projectVideoInfo.getCutFromSecond(), projectVideoInfo.getCutToSecond());

        FFMpeg.Output pad = (0 != projectVideoInfo.getStartSeconds())
            ? ff.timePad(eq, projectVideoInfo.getStartSeconds())
            : eq;

        background = ff.overlay(background, pad,
                    projectVideoInfo.getX() + projectVideoInfo.getCropX(),
                    projectVideoInfo.getY() + projectVideoInfo.getCropY());
    }
}
```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

```

        FFMpeg.Marker audioBackground = null;

        for (Map.Entry<ProjectAudioInfo, VideoInfo> entry :
audioInfoMapping.entrySet()) {
            final ProjectAudioInfo projectAudioInfo = entry.getKey();

            FFMpeg.Output delayed = ff.aDelay(

ff.addAudioInput(projectAudioInfoToTempPath.get(projectAudioInfo).toString()),
            projectAudioInfo.getStartSeconds());

            FFMpeg.Output mixed = Objects.isNull(audioBackground)
                ? delayed
                : ff.aMix(audioBackground, delayed);

            audioBackground = mixed;
        }

        final String cmd = ff.buildCmd(background, audioBackground, outputPath);
        log.info("Executing FFMpeg: {}", cmd);
        execute(cmd);

        return outputPath;
    }

    private void execute(String cmd) {
        final Process process = Runtime.getRuntime().exec(cmd);
        int exitCode = process.waitFor();
        if (exitCode != 0) {
            throw new IllegalStateException("FFMpeg exited with code "
                + exitCode
                + ", command: '"
                + cmd
                + "', error: "
                + new
String(process.getErrorStream().readAllBytes()));
        }
    }
}

```

VideoStorageService.java

```

public void delete(UUID id) {
    record ProjectAndId(ProjectInfo projectInfo, UUID videoInfoId) {}
    record ProjectIdAndName(UUID projectId, String projectName) {}

    final List<ProjectIdAndName> usedInProjects =
projectInfoRepository.findAll().stream()
        .flatMap(projectInfo -> Stream.concat(
            projectInfo.getAudioInfos().stream()
                .map(ProjectAudioInfo::getVideoInfoId)
                .map(videoInfoId -> new ProjectAndId(projectInfo,
videoInfoId)),
            projectInfo.getVideoInfos().stream()
                .map(ProjectVideoInfo::getVideoInfoId)
                .map(videoInfoId -> new ProjectAndId(projectInfo,
videoInfoId))))
        .filter(projectAndId -> id.equals(projectAndId.videoInfoId()))

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

```

        .map(projectAndId -> new ProjectIdAndName(
            projectAndId.projectInfo().getId(),
            projectAndId.projectInfo().getName()))
        .distinct()
        .toList();
    if (!usedInProjects.isEmpty()) {
        throw new CannotDeleteFragmentException(
            "Fragment %s is used in the following projects:
%s".formatted(id, usedInProjects));
    }
    videoInfoRepository.deleteById(id);
}

public VideoInfo upload(MultipartFile file, String name, String description) {
    log.info("Uploading: {}", name);
    try {
        final String extension =
VideoUtil.getExtension(file.getOriginalFilename(), false);

        final String videoRandomName = UUID.randomUUID().toString();
        final String videoRandomNameWithExtension = videoRandomName + "." +
extension;
        final String videoS3Key =
videoKeyPrefixPath.resolve(videoRandomNameWithExtension).toString();
        final String thumbnailS3Key =
thumbnailKeyPrefixPath.resolve(videoRandomName + ".png").toString();
        try {
            // Write video to disk (temporarily)
            final Path videoPath =
Files.createTempFile(Path.of(storageProperties.getTempFilePrefix()), null, null);
            Files.write(videoPath, file.getBytes());

            // Extract video info
            final VideoInfo videoInfo =
videoInfoService.readVideoInfo(videoPath);
            if (Objects.isNull(videoInfo.getFormat().getStartTime())) {
                videoInfo.getFormat().setStartTime("0");
            }

            if (Objects.isNull(videoInfo.getFormat().getDuration())) {
                videoInfo.getFormat().setDuration("10");
                videoInfo.getFormat().setPicture(true);
            }
            videoInfo.setId(UUID.randomUUID());

            // Create thumbnail for the video

            final Path thumbnailPath = tryGetThumbnailPath(videoPath,
videoInfo);

            // Write video and thumbnail to S3
            final String bucket = storageProperties.getS3().getBucketName();
            client.putObject(rb -> rb.bucket(bucket).key(videoS3Key),
videoPath);
            client.putObject(rb -> rb.bucket(bucket).key(thumbnailS3Key),
thumbnailPath);

            // Enrich video info and write it to MongoDB
            videoInfo.setName(name);
            videoInfo.setDescription(description);
            videoInfo.setStorageInfo(VideoStorageInfo.builder()
                .videoS3Key(videoS3Key)
                .thumbnailS3Key(thumbnailS3Key)
                .build());

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

```

        videoInfoRepository.save(videoInfo);

        // Generate frames for preview
        try {
            final Path framesPrefixPath = splitIntoFrames(videoPath);
            copySecondScreenshotsToS3(framesPrefixPath,
videoInfo.getId());
            removeDirectory(framesPrefixPath);
        } catch (Exception e) {
            log.warn("No frames found: " + e.getMessage());
        }

        // Delete files from file system
        Files.delete(videoPath);
        Files.delete(thumbnailPath);

        return videoInfo;
    } catch (IOException e) {
        throw new StorageException(e);
    }
} catch (ArrayIndexOutOfBoundsException e) {
    throw new IllegalStateException("Cannot find extension in the file
name: " + name);
}
}

private Path tryGetThumbnailPath(Path videoPath, VideoInfo videoInfo) {
    try {
        return videoThumbnailService.createVideoThumbnail(videoPath,
videoInfo);
    } catch (Exception e) {
        final Path black = Path.of("./src/main/resources/black.jpeg");
        final Path tempPathStr =
Path.of(storageProperties.getTempFilePrefix(), UUID.randomUUID() + ".jpeg");
        Files.copy(black, tempPathStr);
        return tempPathStr;
    }
}

private Path splitIntoFrames(Path videoPath) {
    log.info("Splitting into frames: {}", videoPath);
    final Path videosDirectoryPath = Files.createTempDirectory(null);
    final String cmd = "%s -i %s -r 1 -vf scale=600:-1 -f image2 %s/%d.png"
        .formatted(ffMpegProperties.getFfmpeg(), videoPath,
videosDirectoryPath);
    log.info("Executing: {}", cmd);

    ProcessUtil.throwOnError(Runtime.getRuntime().exec(cmd));

    return videosDirectoryPath;
}

private void copySecondScreenshotsToS3(Path screenshotDir, UUID videoInfoId) {
    log.info("Copying screenshots to s3: {}", screenshotDir);
    try (Stream<Path> screenshotPathStream = Files.list(screenshotDir)) {
        screenshotPathStream.forEach(screenshotPath -> {
            final String s3Key =
storageProperties.getS3().getSecondScreenshotsPrefix()
                + "/" + videoInfoId + "/" + screenshotPath.getFileName();
            client.putObject(builder -> builder
                .bucket(storageProperties.getS3().getBucketName())
                .key(s3Key),
                screenshotPath);
        });
    }
}

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

```

    }
}

private void removeDirectory(Path directoryPath) {
    try (Stream<Path> walk = Files.walk(directoryPath)) {
        walk.forEach(path -> {
            try {
                if (!Files.isSameFile(path, directoryPath)) {
                    Files.deleteIfExists(path);
                }
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        });
    }
    Files.delete(directoryPath);
}
}

```

					КПІ.ІТ-84В11.045440.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕДАГУВАННЯ ВІДЕОФАЙЛІВ

Програма та методика тестування

КПІ. 84в11.045440.03.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Юрій ОЛІЙНИК

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Євгенія ГЮРЖИ

Київ – 2023

ЗМІСТ

1 ОБ'ЄКТ ВИПРОБУВАНЬ	3
2 МЕТА ТЕСТУВАННЯ	4
3 МЕТОДИ ТЕСТУВАННЯ	5
4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ	7

					КПІ.ІТ-84В11.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є програмне забезпечення для редагування відеофайлів представляє собою веб-сервіс, створений із використанням мови JavaScript для фронтенду та Spring фреймворку із використанням мови Java.

					КПІ.ІТ-84В11.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка збереження даних;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Chrome, Opera, Firefox, ...);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірити функціональність обрізання відео;
- перевірити функціональність додавання фрагментів відео;
- перевірити функціональність додавання аудіо;
- перевірити функціональність переміщення відео;
- перевірка зручності графічного інтерфейсу.

					КПІ.IT-84В11.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

3 МЕТОДИ ТЕСТУВАННЯ

Опис видів застосованих при розробці програмного продукту для даної дипломної роботи:

- Unit tests – кількість цих тестів досить велика і вони гарантують, що ізольовані одиниці коду працюють належним чином. Ці тести повинні перевіряти одну просту вимогу і не покладатися на будь-які зовнішні залежності;
- Integration tests - тип тестування, коли програмні модулі логічно інтегровані та тестуються як група. Типовий програмний проєкт складається з кількох програмних модулів, закодованих різними програмістами. Метою цього рівня тестування є виявлення дефектів у взаємодії між цими програмними модулями, коли вони інтегровані;
- End-to-end tests – моделюють реальну роботу продукту, включаючи всі його частини.

Маємо протестувати кожний можливий варіант (test case), як ще кажуть, прецедент. Результат кожного тесту та інформація про його перебіг буде збережена до сервісу. Людина, що тестує, проводить кожний із тестів і позначає результат його виконання за кожною із умов.

Після того, як тестування було завершено, буде проведено перегляд звіту із тестування. У разі знаходження будь-якої із помилок - їх перелік передається розробнику сервісу для їх подальшого виправлення.

Основна функціональність системи та всі її складові мають функціонувати як очікувалося та бути окресленими в окремих випадках тестування. Не повинно бути виявлено критичних дефектів, тобто дефектів, при яких основна функціональність системи не виконує, або тільки частково виконує свою задачу.

Кінцевий користувач має мати змогу успішно вибрати метод введення великих даних, ввести їх, вибрати алгоритм внутрішнього перетворення сховища, виконати алгоритм внутрішнього перетворення сховища, побачити результат

виконання чи помилку, побачити список доступних інтерфейсів перетвореного сховища даних, завантажити перетворені дані одним методом

					КПІ.ІТ-84В11.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально з використанням наскрізного тестування (End-to-end) та написанням unit-тестів, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування.

Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на виведення повідомлень про помилку, коли це необхідно;
- тестування інтерфейсу користувача;
- тестування зручності використання;
- тестування функціональності обрізання відео;
- тестування функціональності склеювання відео
- тестування функціональності додавання фрагментів відео;
- тестування функціональності додавання аудіо;
- тестування функціональності переміщення відео.

					КПІ.IT-84В11.045440.04.51	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

Програмне забезпечення для редагування відеофайлів

Керівництво користувача

КПІ.ІТ-8411.045440.01.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Юрій ОЛІЙНИК

Нормоконтроль:

_____ Ірина ВІТКОВСЬКА

Виконавець:

_____ Євгенія ГЮРЖИ

Київ – 2023

ЗМІСТ

1	ПРИЗНАЧЕННЯ ПРОГРАМИ	3
2	ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ...	4
2.1	Системні вимоги для коректної роботи	4
2.2	Завантаження застосунку.....	4
2.3	Перевірка коректної роботи	5
3	ВИКОНАННЯ ПРОГРАМИ.....	6

					КПІ.ІТ-84в.045440.05.34	
Змін.	Арк.	№ докум.	Підп.	Дата.		2

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

«Video Editor» - це веб-застосунок для редагування відеофайлів, який призначений для полегшення обробки відеофайлів за рахунок серверних технологій.

					КПІ.ІТ-84в.045440.05.34	Арк.
						3
Змін.	Арк.	№ докум.	Підп.	Дата.		

2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- наявність комп'ютеру або ноутбуку під управлінням операційних систем, як Windows, Linux або Mac;
- Тип процесору Intel Pentium 2.2 MHz.
- Об'єм ОЗП 128 Мб.
- Програмне забезпечення повинно функціонувати на операційних системах, для яких реалізовано віртуальну машину Java та пакет NodeJS.
- наявність стабільного доступу до Інтернету;
- для встановлення додатку на мобільному пристрої повинно бути не менше 70 МБ вільної пам'яті.

2.2 Завантаження застосунку

Дане програмне забезпечення розроблене для запуску на AWS Cloud, тому необхідно встановити AWS CLI і Terraform, слідуючи інструкції для вашої операційної системи, створити IAM роль на AWS та налаштувати локальний AWS клієнт для використання цієї ролі. Після чого в папці infra з репозиторію програми запуснути наступні команди:

- terraform init – ініціалізація робочого місця terraform;
- terraform validate – перевірка коду IaC на помилки;
- terraform apply -auto-approve – розгортання програми на AWS.

Для локального запуску програми (у випадку дебагу) можна використати файл docker-compose.yml, який лежить в корені репозиторія. Для цього необхідно встановити на локальний комп'ютер програми docker та docker-compose та запуснути команду docker-compose up в корені репозиторія.

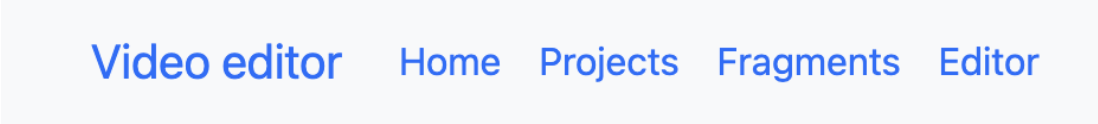
2.3 Перевірка коректної роботи

Після успішного запуску програми через Terraform, у секції Outputs буде виведено DNS ім'я, яке можна використати для доступу до веб-інтерфейсу.

					КПІ.ІТ-84в.045440.05.34	
Змін.	Арк.	№ докум.	Підп.	Дата.		5

3 ВИКОНАННЯ ПРОГРАМИ

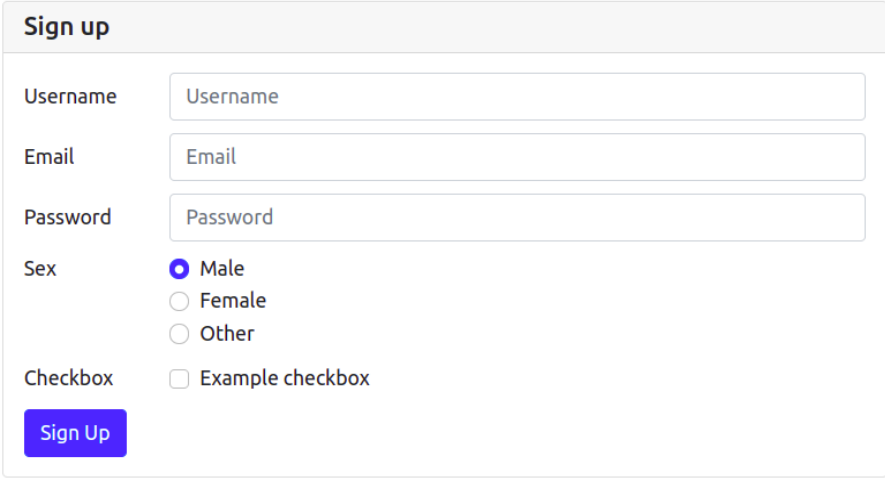
Перше, що ми можемо побачити після входу на веб-сайт – сторінка з описом веб-додатку та пропозицією увійти чи зареєструватись. Натиснувши на кнопку “Sign up”, яка знаходиться в меню вгорі, яке можна побачити на рисунку, що наведено нижче:



Video editor Home Projects Fragments Editor

Рисунок 4.2 – Меню веб-додатку

Переходимо на сторінку реєстрації, вводимо дані (логін, пароль) та тиснемо кнопку “Sign up”, яка зображена на рисунку 4.3:



Sign up

Username

Email

Password

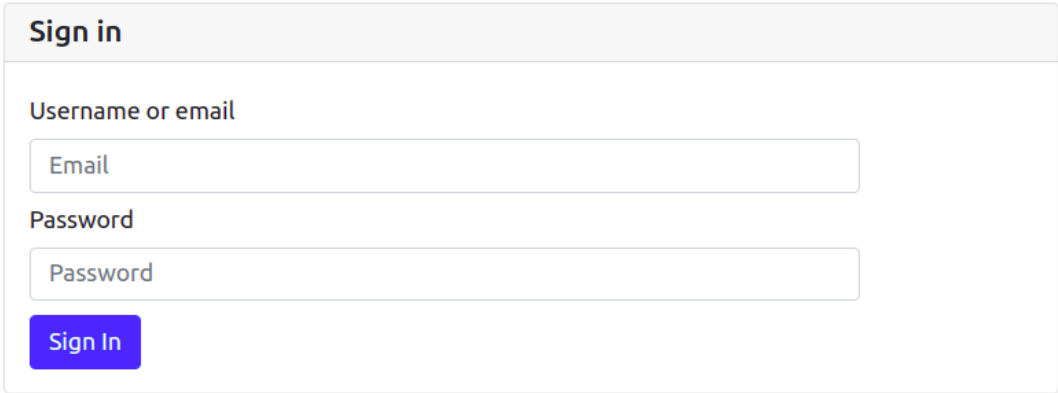
Sex ☒ Male ☐ Female ☐ Other

Checkbox ☐ Example checkbox

Sign Up

Рисунок 4.3 – Форма реєстрації на сайті

У випадку, коли у користувача вже існує акаунт, необхідно натиснути кнопку «Sign in», ввести дані та знову натиснути кнопку «Sign in»:



Sign in

Username or email

Password

Sign In

Рисунок 4.4 – Форма для входу в існуючий акаунт

Для того, щоб розпочати роботу після входу, необхідно створити новий проєкт, написавши його назву та опис, та натиснути кнопку «Create».

Рисунок 4.5 – Панель створення нового проєкту

Також можна обрати існуючий, натиснувши на його віджет на екрані. Після чого відкриється сам проєкт. Список проєктів можна знайти в пункті меню «Projects» вгорі екрану, що продемонстровано на рисунку 4.6:

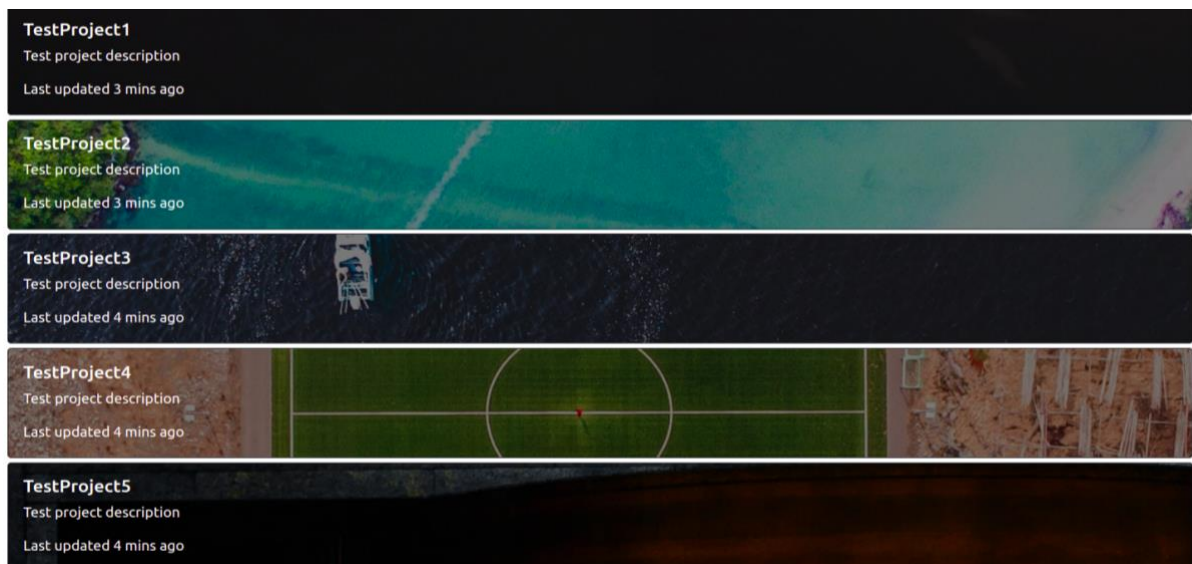


Рисунок 4.6 – Список існуючих проєктів

В центрі екрану – робоча область, яка продемонстрована на рисунку 4.7. Тут знаходиться зображення готового відео, а також фрагменти, з яких воно було створене. Для перемикання відео/фрагмент можна використати кнопку “Video/Fragments” в правому верхньому куті екрану. У випадку активного “Video”

					КПІ.ІТ-84в.045440.05.34	7
Змін.	Арк.	№ докум.	Підп.	Дата.		

на екрані буде зображений готовий відеофайл (трансляція з сервера), у випадку активного “Fragments” відображатимуться фрагменти, з яких змонтоване відео. Для переміщення фрагменту необхідно його обрати подвійним натисненням лівої кнопки мишки, затиснути ліву кнопку мишки та тягнути до необхідної позиції.

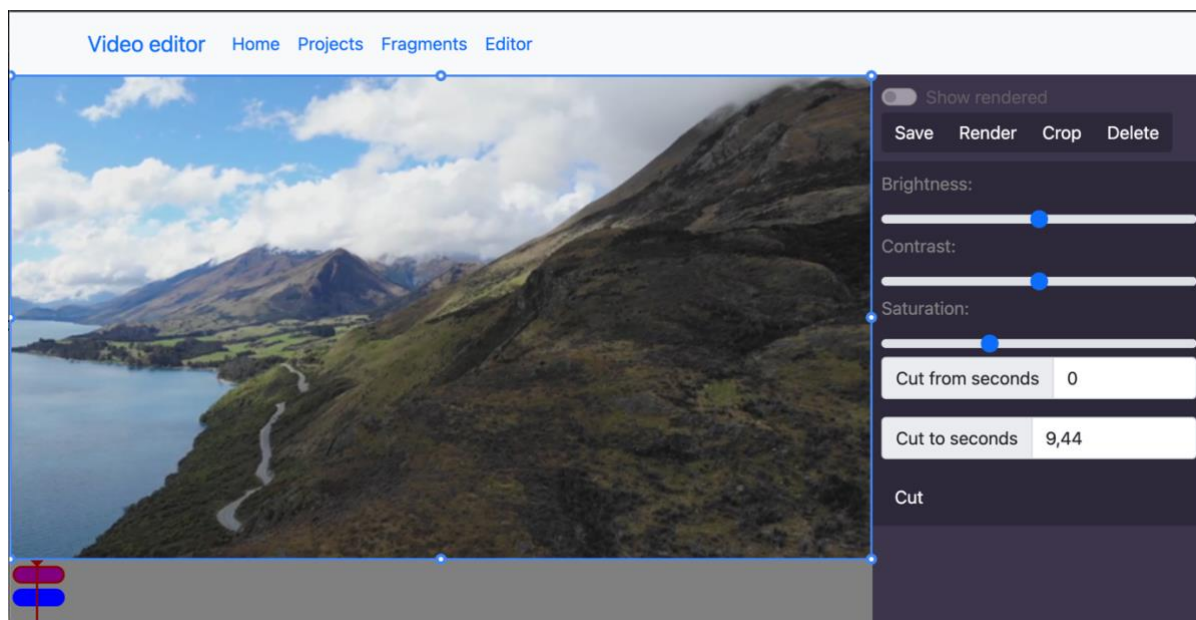


Рисунок 4.7 – Робоча область

Справа знаходиться панель налаштувань, внизу зліва - панель шарів відео, що зображені на рисунку 4.8. Панель налаштувань дозволяє змінювати параметри відео (розміри, позиція) більш точно, а також додавати ефекти (наприклад, контрастність). Оскільки відео можуть накладатися, не завжди зручно їх обирати в робочій області. Для вирішення цієї проблеми була запрограмована панель шарів. Кожен елемент має свій шар, які відображаються на панелі у вигляді списку, тому їх зручно обирати. Відео та фото (далі - фрагменти), які знаходяться на панелі вище мають вищий пріоритет, тому перекриватимуть фрагменти, які знаходяться в тій зоні в робочій області.

Внизу знаходиться таймлайн, який також являє собою список всіх фрагментів. Його особливістю є те, що він, також, відображає розміщення елементу в часі відносно початку відео. Тут можливо проводити такі маніпуляції як обрізання відео, розбиття відео на фрагменти, склеювання фрагментів.

Для розбиття відео на фрагменти необхідно обрати потрібне відео (фрагмент) на часолінії або на панелі шарів і натиснути кнопку «Cut».



Рисунок 4.8 – Панель інструментів для редагування відео

Після чого в обох формах (на часолінії та на панелі шарів) з'являться фрагменти.



Рисунок 4.9 – Панель шарів відео

Останній елемент інтерфейсу – вкладка «Fragments». Вона дозволяє додавати елементи (відео, фото, аудіо) в проєкт (вони будуть завантажені на сервер), зберігати та експортувати проєкт.

Для того, щоб загрузити фрагмент відео, аудіо чи фото на сервер, необхідно у панелі завантаження файлів, яка продемонстровано на рисунку 4.10, натиснути

«Оберіть файл» та обрати з файлової системи пристрою потрібний фрагмент, написати назву та опис файлу та натиснути кнопку «Upload».

Upload video

Выберите файл

940bbaf8-3...bc6

Choose file...

test

test

Upload

Рисунок 4.10 – Панель завантаження файлів

Після чого відкриється меню (специфічне для файлової системи), в якому необхідно обрати потрібний файл і натиснути кнопку «Open», що зображено на рисунку нижче:

					КПІ.ІТ-84в.045440.05.34	
Змін.	Арк.	№ докум.	Підп.	Дата.		

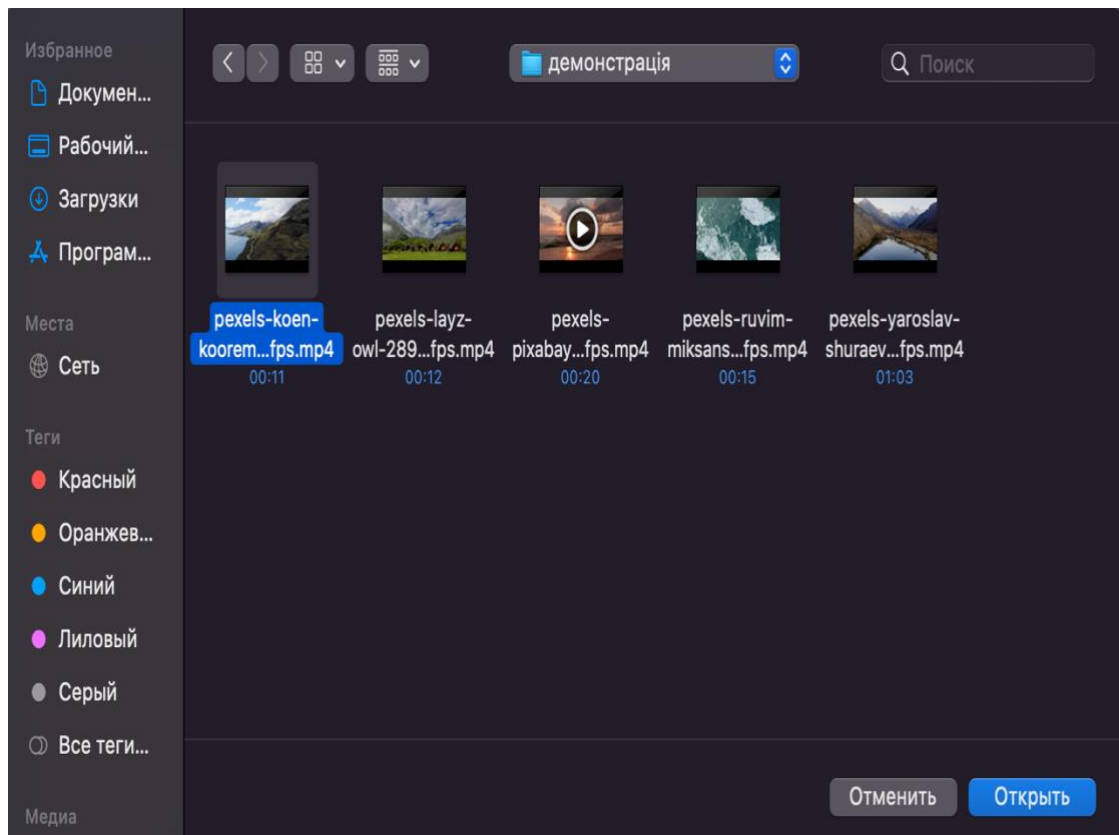


Рисунок 4.11 – Меню для вибору потрібного файлу.

Для того, щоб сформувати відео необхідно обрати пункт меню «Render», який зображено на рисунку нижче, і натиснути кнопку.

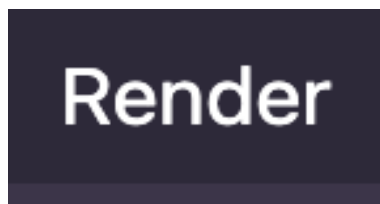


Рисунок 4.12 – Кнопка «Render»

Після чого розпочнеться рендеринг відео та по завершенню цього процесу, користувач отримує повідомлення про те, що рендеринг завершено.



Рисунок 4.13 – Повідомлення про закінчення рендерингу

Після закінчення експорту можна натиснути на перемикач «Show rendered» та передивитись отриманий результат. У програвачі можна натиснути на три крапки в правому нижньому куті та завантажити результат на пристрій користувача.

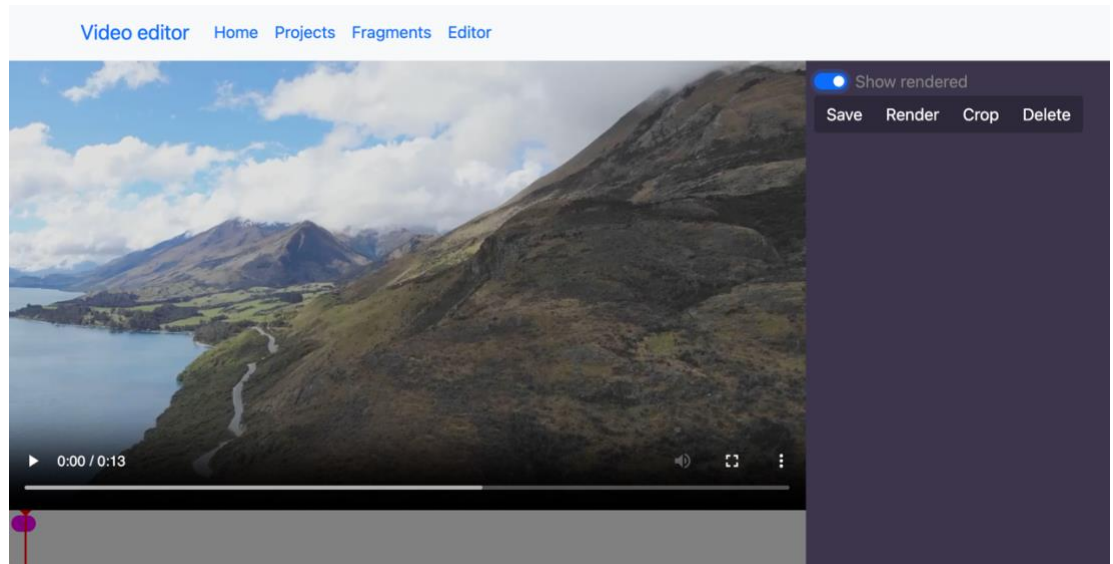
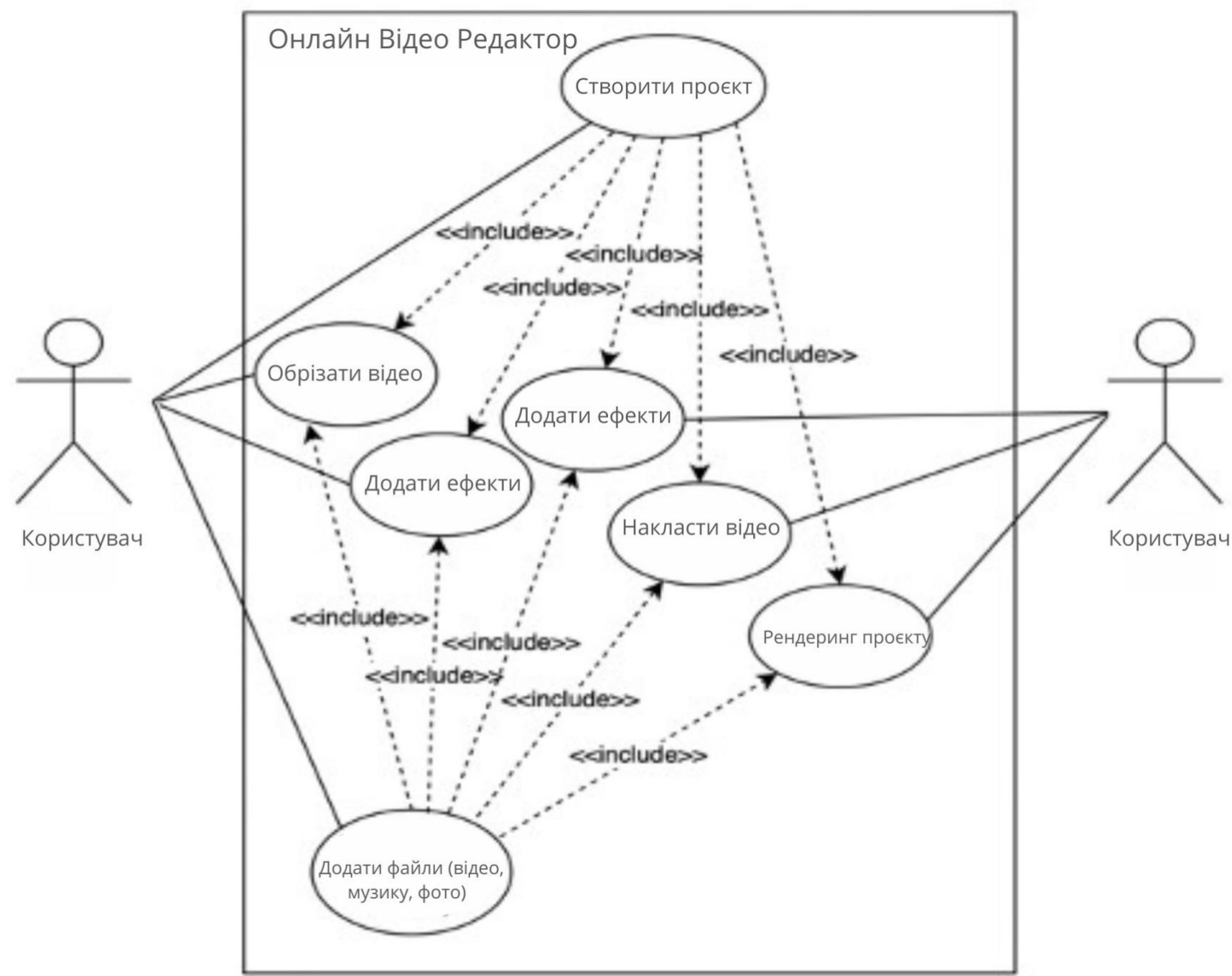
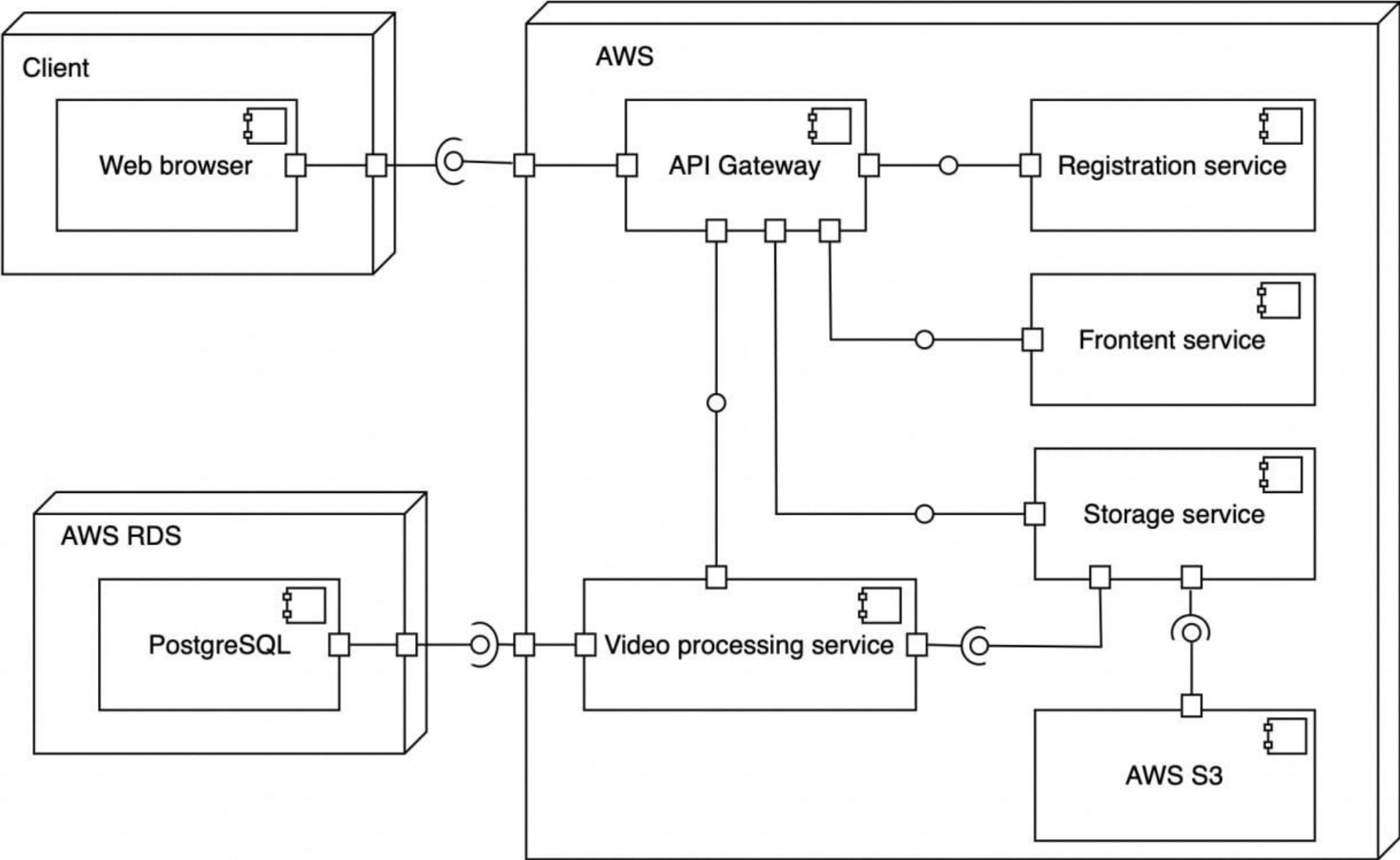


Рисунок 4.14 – Вікно перегляду отриманого результату



					КПІ.ІТ-84є11.045440.06.99.ССВ				
					Схема структурна варіантів використання				
Зм.	Арк.	№ документа	Підпис	Дата	Програмне забезпечення для редагування відеофайлів				
Розробив		Гюржи Є.О.							
Перевірив		Олійник Ю.О.							
Т. кон.					КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-84				
Н. кон.		Вітовська І.І.							
Затвердив		Олійник Ю.О.							



					КПІ.ІТ-84в11.045440.07.99.ССМ								
					Схема структурна компонентів програмного забезпечення				Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив	Гюржи Є.О.												
Перевішив	Олійник Ю.О.												
Т. кон.					Програмне забезпечення для редагування відеофайлів				Аркуш 1		Аркушів 1		
									КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-84в				
Н. кон.		Вітовська І.І.											
Затвердив		Олійник Ю.О.											

Меню веб-додатку

Sign up

Username

Username

Email

Email

Password

Password

Sex

☒ Male

☐ Female

☐ Other

Checkbox

☐ Example checkbox

Sign Up

Форма реєстрації на сайті

TestProject1

Test project description

Last updated 3 mins ago

TestProject2

Test project description

Last updated 3 mins ago

TestProject3

Test project description

Last updated 4 mins ago

TestProject4

Test project description

Last updated 4 mins ago

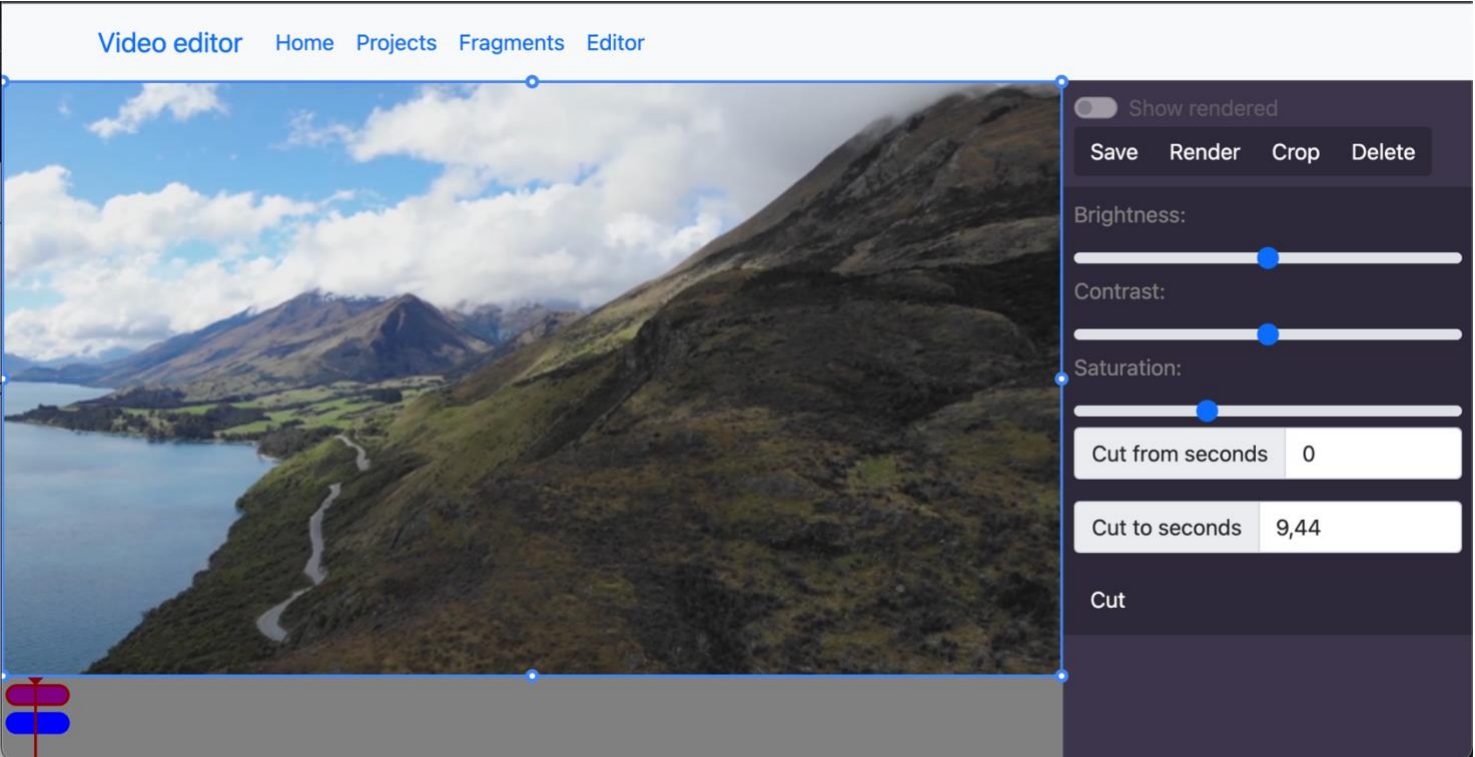
TestProject5

Test project description

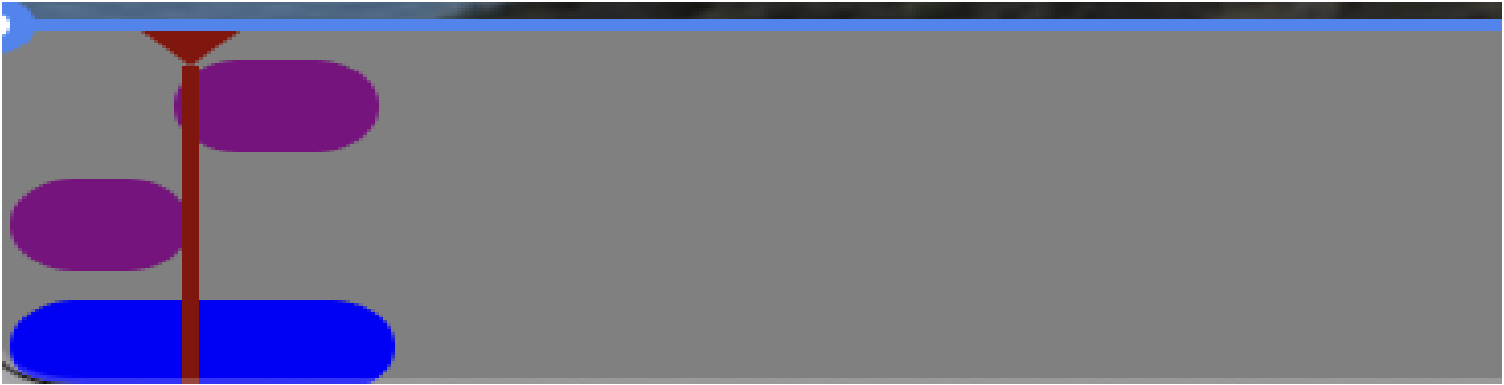
Last updated 4 mins ago

Список існуючих проєктів

					КПІ.IT-84в11.045440.08.99.KE				
					Схема вигляду креслення екранних форм		Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Гюржи Є.О.							
Перевірив		Олійник Ю.О.							
Т. кон.							Аркуш 1		Аркушів 3
					Програмне забезпечення для редагування відеофайлів		КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. IT-84в		
Н. кон.		Вітовська І.І.							
Затвердив		Олійник Ю.О.							



Робоча область

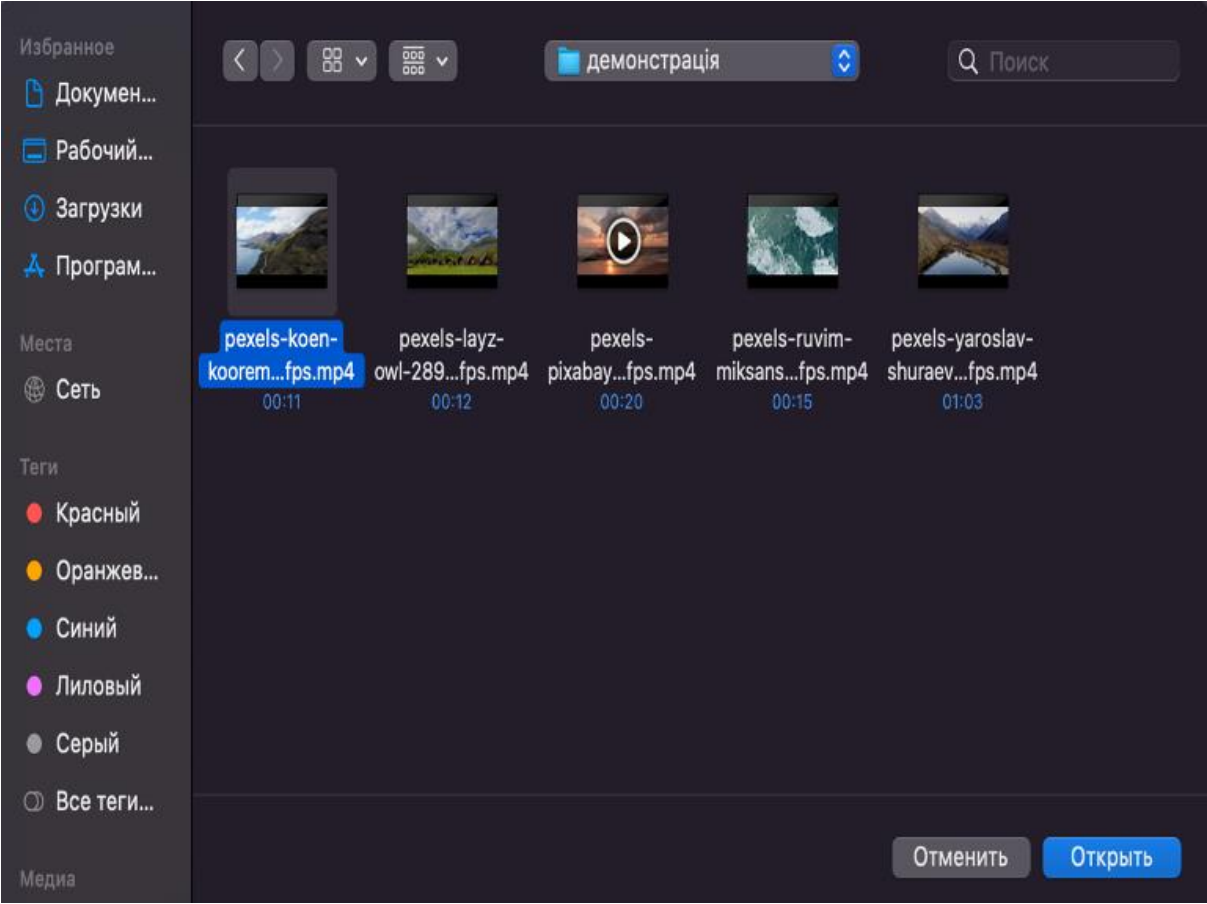


Панель шарів відео



Панель інструментів

					КПІ.ІТ-84в11.045440.08.99.КЕ				
					Схема вигляду креслення екранних форм		Літера	Маса	Масштаб
							Аркуш 2		Аркушів 3
Зм.	Арк.	№ документа	Підпис	Дата	Програмне забезпечення для редагування відеофайлів		КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-84в		
Розробив		Гюржи Є.О.							
Перевірів		Олійник Ю.О.							
Т. кон.									
Н. кон.		Вітовська І.І.							
Затвердив		Олійник Ю.О.							



Меню для вибору потрібного файлу



Доданий в бібліотеку файл



Вікно повідомлення про завершення



Перемикач «Show rendered»

					КПІ.ІТ-84в11.045440.08.99.KE							
					Схема вигляду креслення екранних форм	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Гюржи Є.О.										
Перевірів		Олійник Ю.О.										
Т. кон.												
					Програмне забезпечення для редагування відеофайлів	Аркуш 3			Аркушів 3			
Н. кон.		Вітовська І.І				КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-84в						
Затвердив		Олійник Ю.О.										