

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня магістр**

зі спеціальності 122 «Комп'ютерні науки» \_\_\_\_\_,

освітньо-професійної програми «Інформаційні технології проектування» \_\_\_\_\_

на тему: «Web-орієнтована система музичних гуртів»

Здобувача групи ІТ.м-22 Білан Владислав Олександрович

(шифр групи)

(прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_

(підпис)

Владислав БІЛАН

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доц., Юлія ПАРФЕНЕНКО

(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

\_\_\_\_\_

(підпис)

**Суми – 2023**

**Сумський державний університет**

**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**  
**Спеціальність 122 «Комп'ютерні науки»**  
**Освітньо-професійна програма «Інформаційні технології проектування»**

**ЗАТВЕРДЖУЮ**

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу магістра студентів**

Білан Владислав Олександрович

(прізвище, ім'я, по батькові)

**1 Тема кваліфікаційної роботи** «Web-орієнтована система музичних гуртів»

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

**2 Термін здачі студентом кваліфікаційної роботи** «11» грудня 2023 р.

**3 Вхідні дані до кваліфікаційної роботи** інформація про гурти, альбоми Spotify, фото учасників та груп, текст та переклади пісень

**4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області, постановка задачі, проектування веб-орієнтованої системи музичних гуртів певного напрямку, реалізація веб-орієнтованої системи музичних гуртів певного напрямку

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації)** актуальність роботи, мета та задачі проекту, аналіз аналогів, порівняльна таблиця аналогів, функціональні вимоги, засоби реалізації, моделювання проекту (контекстна діаграма, діаграма декомпозиції), діаграма варіантів використання, модель бази даних, демонстрація проекту, результати тестування, висновки.

**6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання \_\_\_\_\_.

Керівник \_\_\_\_\_  
(підпис)

Завдання прийняв до виконання \_\_\_\_\_  
(підпис)

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Дослідження предметної області	04.09.23-19.09.23	
2	Формування мети і задач	15.09.23-19.09.23	
3	Аналіз аналогів та проблем використання	20.09.23-27.09.23	
4	Виявлення вимог до проекту	28.09.23-02.10.23	
5	Планування робіт та розробка макету	03.10.23-06.10.23	
6	Вибір засобів реалізації	07.10.23-09.10.23	
7	Проектування інформаційної системи	10.10.23-19.10.23	
8	Розробка інформаційної системи	20.10.23-09.11.23	
9	Тестування та завершення роботи	10.11.23-13.11.23	
10	Оформлення пояснювальної записки	14.11.23-01.12.23	

Магістрант \_\_\_\_\_

Владислав БІЛАН

Керівник роботи \_\_\_\_\_

к.т.н., доц. Юлія ПАРФЕНЕНКО

## Анотація

Тема кваліфікаційної роботи магістра «Web-орієнтована система музичних гуртів»

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 40 найменувань, 2 додатків. Загальний обсяг роботи – 87 сторінок, у тому числі 81 сторінок основного тексту, 5 сторінок списку використаних джерел, 22 сторінок додатків. Пояснювальна записка включає в себе 34 рисунків та 16 таблиць.

Кваліфікаційну роботу присвячено розробці веб-орієнтованої системи музичних гуртів певного напрямку. В роботі здійснено аналіз предметної області, а саме застосування веб-сайтів для популяризації музики та гуртів. Була сформована задача, визначені функціональні вимоги та обрані засоби реалізації продукту.

У роботі здійснено проектування веб-орієнтованої системи, а саме функціональне моделювання роботи з веб-сайтом, моделювання варіантів використання розробки та проектування моделі бази даних. Процес реалізації системи був описаний.

Результатом виконаної роботи є реалізована веб-орієнтована система музичних гуртів напрямку К-поп.

Практичне значення роботи полягає у розповсюдженні напрямку К-поп, популяризації гуртів, за рахунок розміщення невідомих груп поруч з відомими.

Ключові слова: інформаційні технології, веб-орієнтована система, веб-сайт, музичні гурти, К-поп, YouTube API, Spotify.

## ЗМІСТ

ВСТУП .....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Огляд останніх досліджень і публікацій .....	9
1.2 Аналіз Web-орієнтованих систем.....	13
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ .....	19
2.1 Мета та задачі дослідження .....	19
2.2 Методи дослідження.....	21
3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ .....	23
3.1 Діаграми нотації IDEF0 .....	23
3.2 Діаграма Use Case .....	33
3.3 Проектування бази даних .....	35
4 РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ МУЗИЧНИХ ГУРТІВ НАПРЯМУ К-POP .....	36
4.1 Розробка архітектури системи.....	36
4.2 Встановлення та розробка компонентів web-орієнтованої системи .....	38
4.3 Розміщення на хостингу та тестування web-орієнтованої системи музичних гуртів .....	42
4.4 Результат реалізації веб-орієнтованої системи музичних гуртів напряму К-pop .....	45
ВИСНОВКИ .....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

ДОДАТОК А.....	65
ДОДАТОК Б.....	77

## ВСТУП

**Актуальність.** Створення централізованого веб-сайту для декількох гуртів є дуже актуальним. Різноманіття жанрів та напрямів все більше, а нові гурти з'являються в них кожен рік. Тому було вирішено створити сайт на котрому буде розміщуватися інформація про декілька гуртів для їх популяризації. Замість створення окремих веб-сайтів для кожного гурту, централізована платформа дозволяє економити час та ресурси на розробку та підтримку окремих сайтів. Користувачам легше шукати інформацію, оскільки вони можуть переглядати різні гурти на одному веб-ресурсі. А також це дозволить розвивати як нові групи, так і в цілому напрям. Для приклада було обрано гурти напряму К-поп, бо він доволі новий і про багато груп мало інформації [1].

**Об'єкт дослідження:** процес інформування про музичні гурти.

**Предмет дослідження:** web-орієнтована система для інформування про музичні гурти певного напрямку.

**Мета.** Розробка web-орієнтованої системи для розміщення інформації про музичні гурти певного напрямку.

Для досягнення мети проекту необхідно виконати наступні задачі:

- виконати детальний аналіз теми інформаційної підтримки діяльності музичних гуртів та огляд сучасних публікацій, визначити актуальність та цільову аудиторію використання представленої web-орієнтованої системи музичних гуртів певного напрямку та провести аналіз аналогів;
- виконати проектування моделі та структури web-орієнтованої системи;
- реалізувати структуру web-орієнтованої системи та її функціональні можливості;

– провести тестування web-орієнтованої системи та розмістити її на хостингу.

**Практичне значення.** Популяризація починаючих музичних гуртів, з мінімальними фінансовими витратами та знанням програмування. Розміщена інформація одразу декількох гуртів на одному сайті дозволить створювати спільну фан-базу, котра потім забезпечить просування наступних нових гуртів.



# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

Із зростанням використання інтернету та цифрових технологій музичні гурти активно використовують web-платформи для залучення аудиторії, продажу музики та вирішення інших завдань. Багато досліджень і розробок в галузі музичних технологій та веб-систем фокусуються саме на цьому [2].

Музика займає велике значення в житті, за допомогою неї отримує досвід пращурів і їх історію. Також прослуховуючи її можна закріпити ідентичність певної культури [3].

На сьогодні дуже багато створюється сайтів для гуртів різного масштабу з використанням різних технологій. Вони мають інформацію про групу, привабливі для користувачів, а також популяризують їх в певному регіоні або світі [4].

Музичні журнали швидко втрачають популярність, переходячи до більш інтерактивних онлайн-форматів. Гурти вже не покладаються на них для просування, замість цього знаходять свої власні шляхи отримати визнання. Веб-сайти стають основним джерелом музичної інформації, і навіть національні платформи менш важливі для деяких, ніж місцеві ресурси та думки друзів у соціальних мережах. Демократизація засобів для створення професійного медіа дозволяє гуртам просувати себе без значної підтримки з боку основних медіа [5].

Можливість отримати популярність і визнання стає доступнішою завдяки зростанню потужності інтернету та зменшенню впливу великих лейблів. Існує багато порад і підходів, як за допомогою інтернету можливо збільшувати свою фан-базу, що описано у [6].

Цей [7] дослідницький матеріал про ірландський фолк-панк гурт Mr. Irish Bastard показує, як використання Інтернету впливає на успіх музичних колективів. Він підкреслює важливість наявності в мережі для привертання уваги фанатів, продажу музики та залучення організаторів концертів. Цей приклад демонструє, як веб-орієнтовані підходи можуть допомогти розвитку гуртів у сучасній музичній індустрії.

Розроблена веб-орієнтована система для музичних гуртів, яка використовує методи машинного навчання для автоматичної класифікації музики за жанрами, виявляється важливим інструментом для сучасних музикантів. Ця технологія дає змогу гуртам удосконалювати організацію та презентацію своєї творчості в онлайні, спрощуючи процес категоризації музики та покращуючи сприйняття аудиторією [8].

Проект, представлений у роботі [9], ілюструє застосування веб-орієнтованої системи для підтримки музичних активностей серед літніх людей. "Angklung Smart Band" використовує технології Інтернету речей для організації музичних занять з використанням ангклунгу. Ця система допомагає замінити складний метод рухів рук на технології, спрощуючи та полегшуючи доступ до музичних активностей для літніх людей. Це демонструє, як веб-технології можуть підтримувати та розширювати можливості музичних гуртів, навіть у контексті розвитку музичних занять для осіб похилого віку.

Ще одна стаття [10] на той же самий проект показує його трохи з іншого боку, з технічного. Ця система, що складається з веб-додатку, IoT пристроїв та музичних інструментів, полегшує та покращує організацію та проведення музичних заходів у групах. Такий підхід дозволяє зробити музичні активності доступнішими та ефективнішими для літніх людей, що проживають як у домашніх умовах, так і у лікарнях чи спеціальних закладах.

Також існують додатки, котрі дозволяють новим гуртам перевірити стабільність звуків без потреби в експерті. Це може бути корисним інструментом для музичних гуртів або клубів, де аматори можуть практикувати та вдосконалювати свої навички, використовуючи веб-додаток для контролю якості виконання звуків. Така система може сприяти розвитку навичок учасників музичних гуртів, особливо тих, хто грає на духових інструментах [11].

У даному випадку [12] використання веб-орієнтованих технологій виявляється у просуванні інді-гурту BAD habit через Інтернет. Цей проект використовує Joomla CMS для створення веб-сайту, який пропонує інформацію про гурт за допомогою PHP-скриптів, бази даних MySQL та веб-сервера Apache. Веб-сайт створено з метою популяризації гурту серед шанувальників музики в Індонезії та надання користувачам зручного джерела інформації про гурт BAD habit.

Використовуючи Microsoft FrontPage XP, автором статті [13] було створено домашню сторінку про популярний музичний гурт PADI. Це наукове дослідження підтримує ідею створення веб-сайту для музичного гурту, щоб надати шанувальникам актуальну інформацію та матеріали про гурт, такі як фотографії учасників, тексти пісень та посилання на різні сторінки. Це може бути важливим елементом веб-орієнтованої системи для музичних гуртів, яка забезпечує доступ до актуальної та цікавої інформації для прихильників гурту та інших користувачів, які хочуть більше дізнатися про їхню творчість та діяльність.

Також організація Drum Corps Indonesia [14] звертається до технологічних можливостей та Інтернету для підтримки своєї діяльності. У науковій роботі обговорюється створення динамічного веб-сайту для цієї організації. Цей веб-сайт має на меті просування гурту та залучення громадськості до участі в Drum Corps Indonesia, дозволяючи організації оновлювати інформацію. Використовуються мова програмування PHP та база даних MySQL для створення цієї веб-орієнтованої системи для музичних гуртів. Такий підхід дозволить Drum Corps Indonesia створити динамічний веб-сайт, який буде привабливим для користувачів та забезпечить широкий доступ до інформації про їхню діяльність та музичний гурт.

Враховуючи все вище зазначене і вибір тематики К-поп, було також проаналізовано чому цей напрям набирає популярності.

У [15] описано історію, чому корейська продукція з кожним роком все більше займає частку продукції. Спочатку це були корейські фільми та драми. Це було пов'язано з їх дешевизною, а також цінностями, які пропагують корейці. Вони завжди молоді, але демонструють важливість сім'ї. К-поп музика це суміш усіх жанрів, реп, рок, поп, мікс заходу та сходу, старого і нового.

Загалом про цей напрям описано у цій статі [16]. Головними факторами популярності є візуал і постійний зв'язок виконавців з своїми фанатами, кожна група або соло виконавець мають особливі назви для них, котрі багато про що кажуть. Також впливає те, що вони не дають засумувати своїм фанdomам випускаючи багато контенту.

В даній статі [17] розбираються особливості та відмінності к-поп. У статі розповідається, що основа це наявність великої кількості колективів, котрі живуть разом, а також продемонстровано як це важко стати «айдолом», бо треба спочатку пройти відбір на конкурсах від різних гігантів, потім роками тренуватися, і лише потім отримати можливість виступити або не отримати.

Тому тема web-орієнтованих систем музичних гуртів певного напрямку є актуальною. Розробка web-орієнтованої системи на прикладі гуртів напрямку К-поп є дуже доречною для його популяризації відносного нового напрямку, а також надання можливості просування свого гурту.

## 1.2 Аналіз Web-орієнтованих систем

Сучасні реалії вимагають повсюдне використання інформаційних технологій у всіх сферах життєдіяльності людини аби забезпечити ефективність, швидкість та якість комунікацій. Щодня відбувається користування веб-сайтами, вони зараз надаються можливість замовити їжу, речі, техніку. Також на них розміщують рекламу, новини, відео і тому подібне. Саме тому використання їх для збільшення аудиторії у певних груп або розповідь про них загалом, є дуже актуальним [18].

Web-орієнтовані системи k-поп гуртів користуються популярністю. На них розміщують основну інформацію про групу, їх останні ролики, посилання на соцмережі, музику.

Для порівняльного аналізу було відібрано дві системи. А саме веб-сайт фанатів гурту BTS «ARMY BTS USA» та фанатський ресурс Stray Kids «SKZ ARCHIVE» [19,20].

До структури «ARMY BTS USA» входять такі компоненти, як основна інформація про гурт, дискографія, новини.(рис.1.1-1.2).

Основні функціональні можливості наступні:

- інформація про гурт;
- перегляд випущених альбомів, їх склад;

- галерея фото;
- посилання на соцмережі.

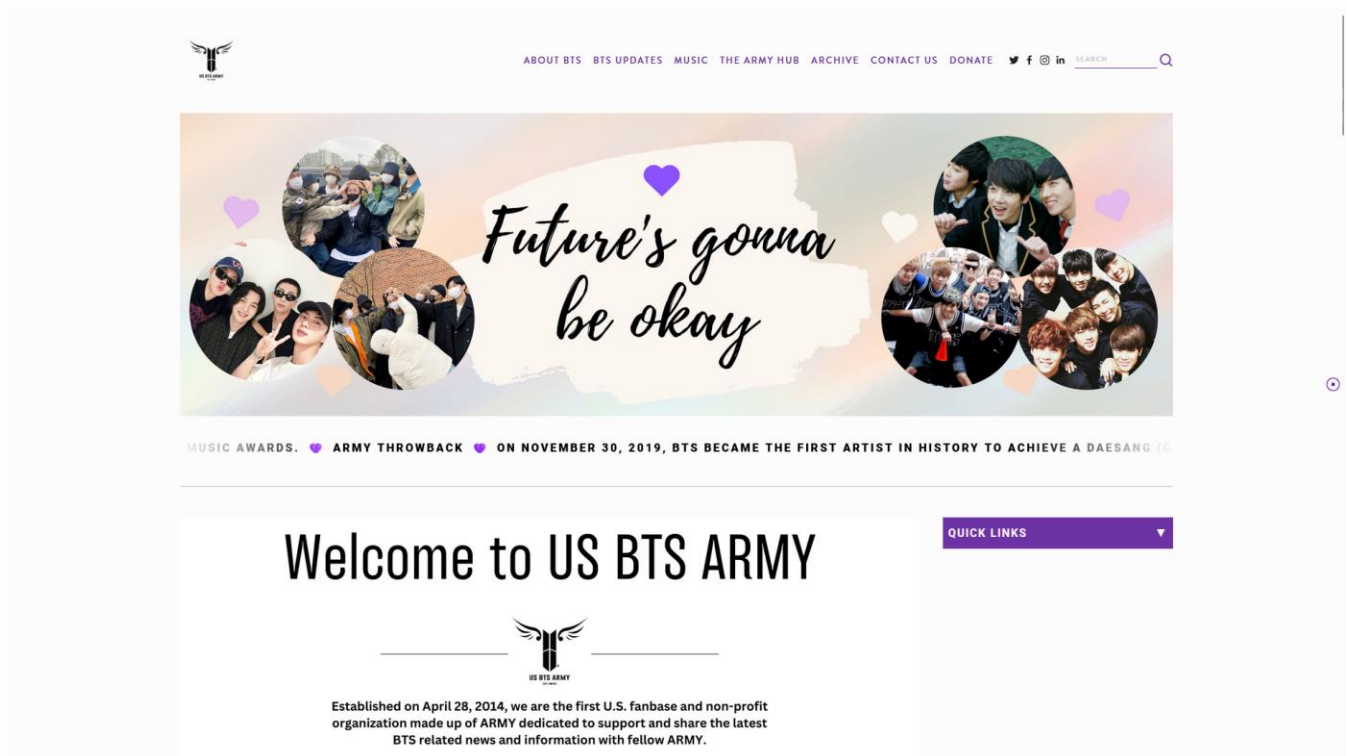


Рисунок 1.1 – Головна сторінка «ARMY BTS USA»

*Джерело: знімок з екрану [19]*

Веб-сайт має доволі простий дизайн, без кольорів, які акцентують все на себе, тому в центрі сторінки фото членів групи та їх девіз концентрують на себе увагу. Це дозволяє прибрати розфокус у відвідувача даного сайту, що в свою чергу збільшує загальне враження.

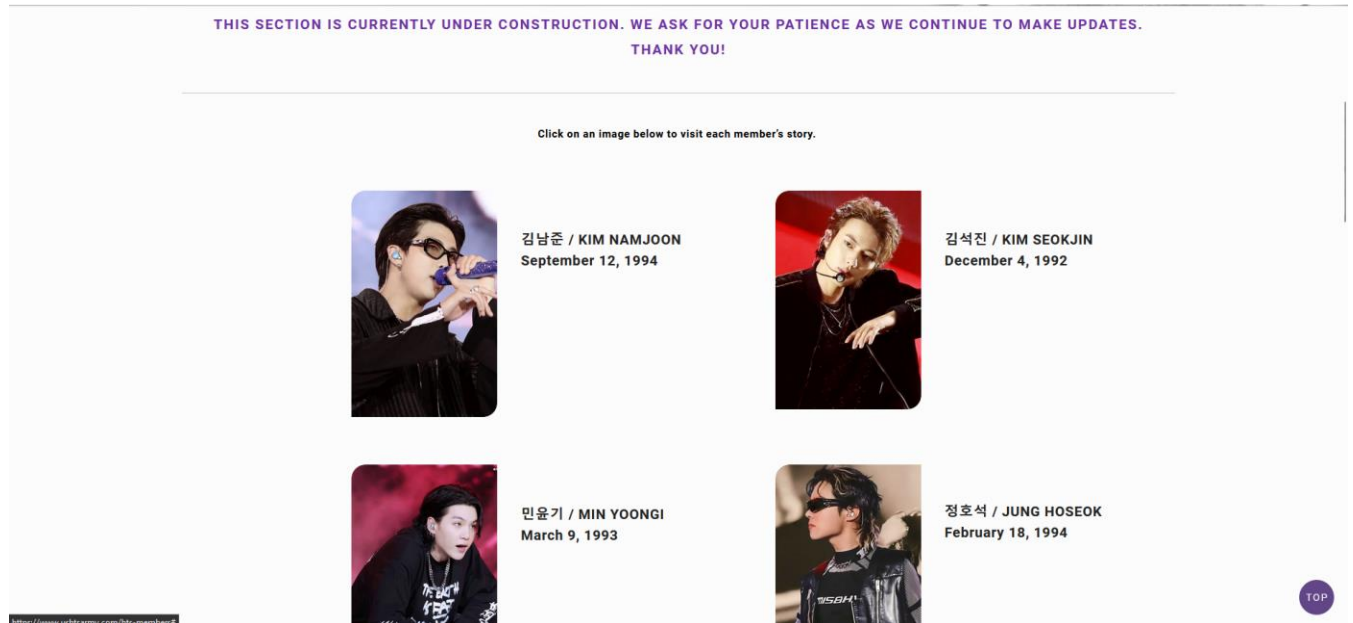


Рисунок 1.2 – Сторінка про групу веб-сайту «ARMY BTS USA»

*Джерело: знімок з екрану [19]*

Після проведення дослідження даного сайту можна зробити висновок, що з першого погляду дійсно є наявним широкий набір сторінок з різним функціоналом, але у результаті отримуємо, що немає тієї ж загальної історії групи, на сторінці дискографії не описана історія того чи іншого альбому.

Web-сайт «SKZ ARCHIVE» був розроблений для фанатів для перегляду відео, розміщення фото учасників(рис.1.3-1.5). Даний сервіс користується великим попитом серед фандому Stray Kids.

Серед функцій web-сайту «SKZ ARCHIVE» варто виділити такі:

- наявність посилань на офіційні сторінки;
- велика галерея фото групи;
- можливість завантажити музику.



Рисунок 1.3 – Головна сторінка веб-сайту «SKZ ARCHIVE»

*Джерело: знімок з екрану [20]*

Серед переваг варто зазначити, що даний сайт має зрозумілий інтерфейс, великий вибір варіантів перегляду відео. Але за рахунок того, що отримання інформації не автоматизовано на сторінці, то будь-які оновлення треба проводити вручну авторам сайту.

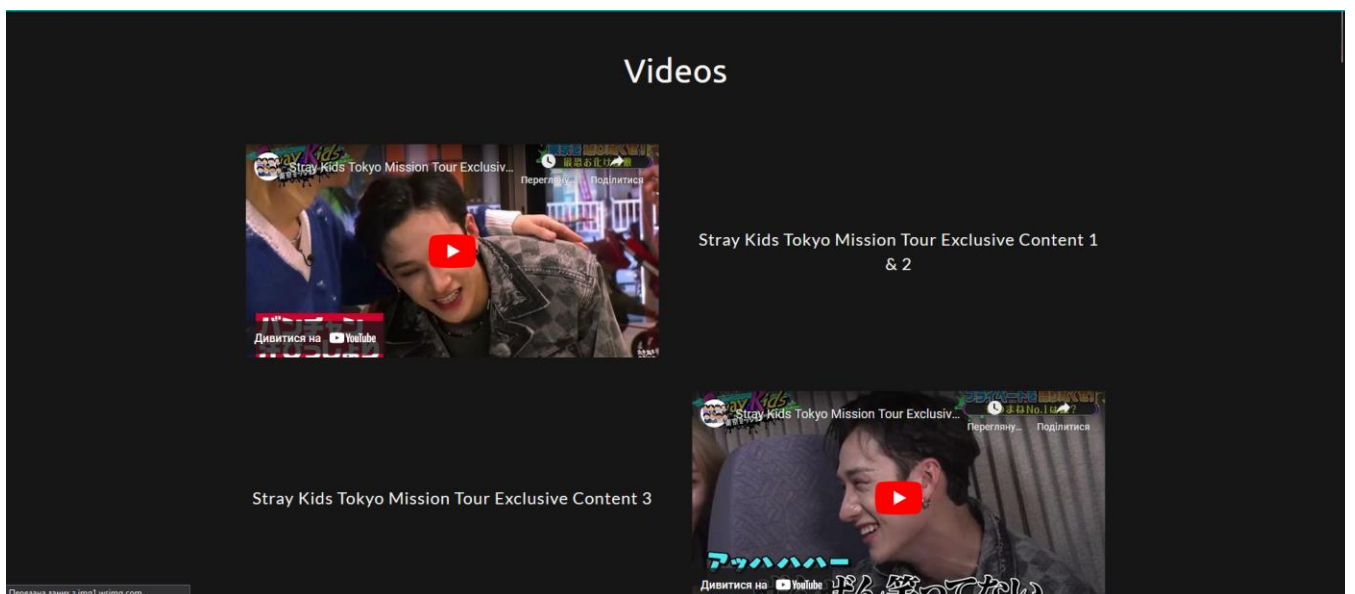




Рисунок 1.4 –Сторінка відео  
*Джерело: знімок з екрану [20]*

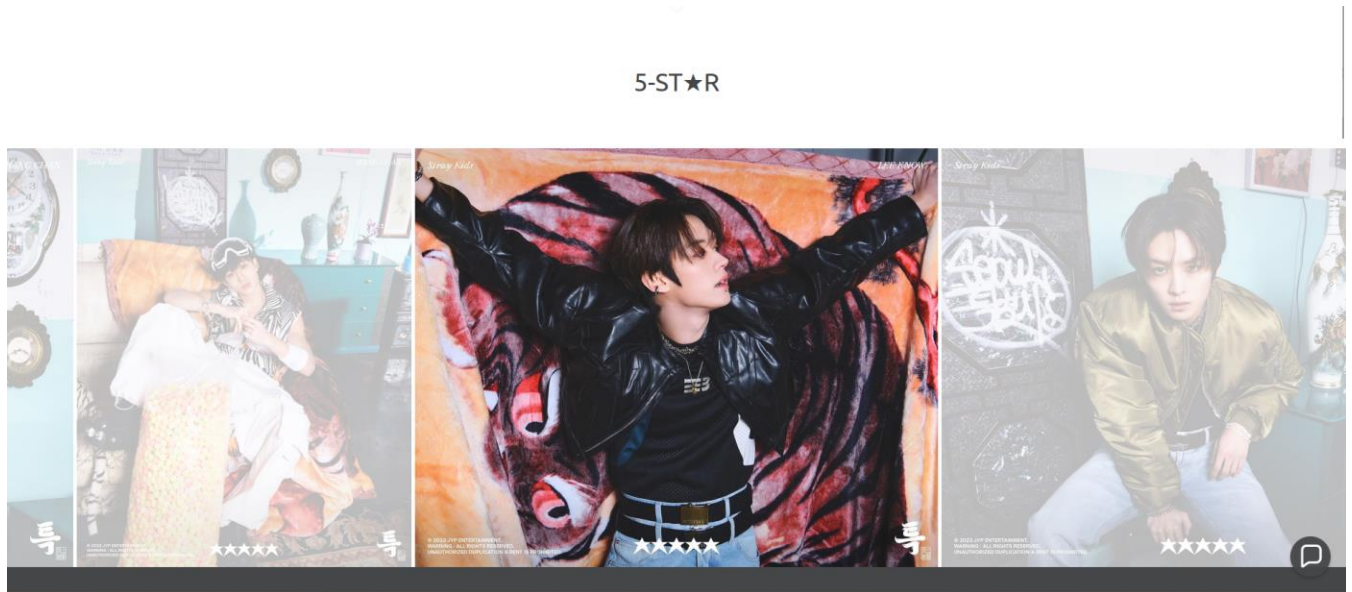


Рисунок 1.5 – Сторінка галереї фото  
*Джерело: знімок з екрану [20]*

Результатом порівняльного аналізу двох вищезазначених програмних рішень та системи, котру розробляємо є таблиця 1.1.

Таблиця 1.1 – Порівняльна таблиця аналогів інформаційних систем

<b>Характеристика/Web-орієнтована система</b>	<b>«BTS ARMY USA»</b>	<b>«SKZ ARCHIVE»</b>	<b>«K-pop bands»</b>
Сучасний дизайн	+	-	+
Зручний інтерфейс та навігація	-	-	+
Інформація про групу	+	-	+
Галерея фото	+	+	-
Посилання на сторінки в соцмережах	+	+	+
Можливість прослуховування музики	-	-	+

Автоматичне оновлення відео	+	-	+
Посилання на музичні платформи	+	-	+
Завантаження музики	-	+	-

*Джерело: побудовано автором*

Дані з таблиці 1.1 надають можливість виділити актуальні функціональні рішення, які можна використати у власній розробці, та недоліки, які варто уникнути. Отже, створюваний програмний продукт повинен мати простий, сучасний та зручний інтерфейс, забезпечувати інформацію про групи, оновлення контенту автоматично, а також інформацію про альбоми. Важливо зазначити, що розроблюваний сервіс повинен надавати інформацію про три групи одночасно.

Але окрім всього вище зазначеного розроблювана web-орієнтована система матиме можливість адміністрування її без спеціалізованих знань в веб-розробці. Буде реалізовано редагування пісень, їх перекладу, інформації про альбом, додавання нових альбомів там пісень. Перевірити чи можливо таке на досліджуваних сайтах неможливо, бо відсутній доступ бек-енду [21].

## 2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

### 2.1 Мета та задачі дослідження

Метою кваліфікаційної роботи магістра є розробка системи для інформування користувачів про популярні групи на прикладі напряму к-поп. Розроблене рішення буде мати попит серед як фанатів к-попу, так і просто меломанів або людей, котрі хочуть дізнатися більше про даний напрям музики. Також, використовуючи дану систему можна переглянути текст пісень і їх переклад українською, коротку інформацію про дані групи, їх соцмережі і навіть прослухати невеликі фрагменти альбомів, для більшого ознайомлення. Упровадження створеного сервісу дозволить збільшити кількість україномовних фанатів к-попу, а також збільшити обізнаність людей не знайомих з даним напрямом.

Розроблена система має задовольняти наступні вимоги до функціональності по категоріям користувачів:

- відвідувач:
  - перегляд останніх відео на каналах цих груп;
  - перегляд інформації про групу;
  - перегляд тексту пісень, їх перекладу, демо-прослуховування;
- адміністратор:
  - авторизація;
  - можливість редагування тексту та перекладу, інформації про альбоми, додавання нових.

Створена web-орієнтована система повинна мати простий та зрозумілий дизайн, із використанням певної кількості візуальних ефектів. Адже її призначенням

є залучення нових користувачів до цього напрямку, а також збільшення обізнаності про дані групи.

Вся інформація про альбоми та пісні буде знаходитись в базі даних у відповідних таблицях. В тій же базі даних знаходиться інформація про логіни та паролі адміністраторів. Вони передаються на сервер в зашифрованому вигляді.

Під час формалізації мети проекту було проведено планування робіт проекту, яке детально описано у Додатку А.

## 2.2 Методи дослідження

Після постановки мети проекту, визначення функціональних та нефункціональних вимог до розроблюваної web-орієнтованої системи музичних гуртів напряму k-pop, формалізації переліку обов'язкових на виконання задач проекту, треба було визначити методи даного дослідження.

Існують наступні чотири основні методи проведення дослідження:

- теоретичний метод;
- системно-функціональний метод;
- емпіричний метод;
- метод моделювання [22].

Теоретичний метод полягає у детальному аналізі особливостей та можливостей предметної області. Під час роботи над проектом теоретичний метод було використано для аналізу цільової аудиторії, її потреби. Також при виборі інструментів на розробку було застосовано теоретичний метод. Цей метод також дозволяє дізнатися переваги та недоліки інструментів розробки та технологій, їх особливості застосування для різних цілей.

Використання емпіричного методу полягає у застосуванні спостережень, вимірювань та експериментів у якості методик пізнання предметної області. Він використовувався на етапі аналізу потреб цільової аудиторії, на основі власних спостережень і досвіду, а також знайомих, котрі цікавляться даним напрямом.

До методу моделювання входить розробка схем та діаграм, що демонструють етапи та процеси під час роботи над реалізацією програмного продукту та його доля в майбутньому, де і як він буде використовуватися. Якщо розглядати роботу над проектом «Web-орієнтована система музичних гуртів напрямку K-pop», то він застосовувався на етапі побудови діаграм IDEF0 та Use Case.

Системно-функціональний метод - це підхід до дослідження та розуміння систем, який базується на аналізі їх функцій та взаємозв'язків між їх складовими частинами. Він був використаний при розробці програмного модулю проекту. А саме для створення варіантів використання, зображених на діаграмі UseCase [23].

Для власне програмної реалізації web-орієнтованої системи було обрано такі технології, як HTML – мову розмітки, для створення шаблону web-сторінок, каскадні таблиці стилів CSS для забезпечення функціонування веб-сайту на різних пристроях та налаштування зовнішнього вигляду, скриптову мову програмування JavaScript для надання front-end функціоналу web-сторінкам, на кшталт слайдеру або роботи з арі, скриптової мови програмування PHP для реалізації back-end частини даного програмного продукту. Для збереження даних представленої системи було обрано системи управління базами даних MySQL [24].

## 3 ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ

Після проведення аналізу предметної області, визначення актуальності роботи та потреб, постановки мети та задач на розробку, завершення процесу планування робіт на реалізацію проекту, було перейдено до наступного етапу – це проектування. Створено діаграми нотації IDEF0, деталізована послідовність процесів, які беруть участь у функціонуванні веб-сторінок.

### 3.1 Діаграми нотації IDEF0

Для відтворення функцій роботи створюваної web-орієнтованої системи існують різні методи об'єктно-орієнтованого та структурного підходу. IDEF0 одна із таких методологій. Є графічним описом системи та процесів роботи додатку як набір взаємопов'язаних функцій. Завдяки використанню IDEF0 досліджуються функції відокремлено від об'єктів, котрі будуть їх використовувати. Діаграма містить такі вхідні дані, керування, вихідні дані та механізми [25].

Для контекстної діаграми було виділено наступне:

1. Вхідні дані: запит на прослуховування альбому, запит на перегляд інформації про гурти, запит на перегляд останніх відео гурту, запит на перегляд тексту пісні та її перекладу для звичайного користувача, а також запит на авторизацію, запит на редагування тексту або перекладу пісні, запит на додавання пісні або альбому, запит на редагування інформації про альбом.

2. Управління: доступність арі, доступність spotify, наявність підключення до серверу.

3. Вихідні дані: прослуховування альбому, перегляд інформації про гурти, перегляд останніх відео гурту, перегляд тексту та перекладу пісні для користувача, а також авторизація, редагування тексту або перекладу пісні, додавання пісні або альбому, редагування інформації про альбом для адміністратора.

4. Механізми: web-орієнтована система, адміністратор, користувач, апаратне забезпечення.

На рисунку 3.1 зображена контекстна діаграма IDEF0 для користувача.

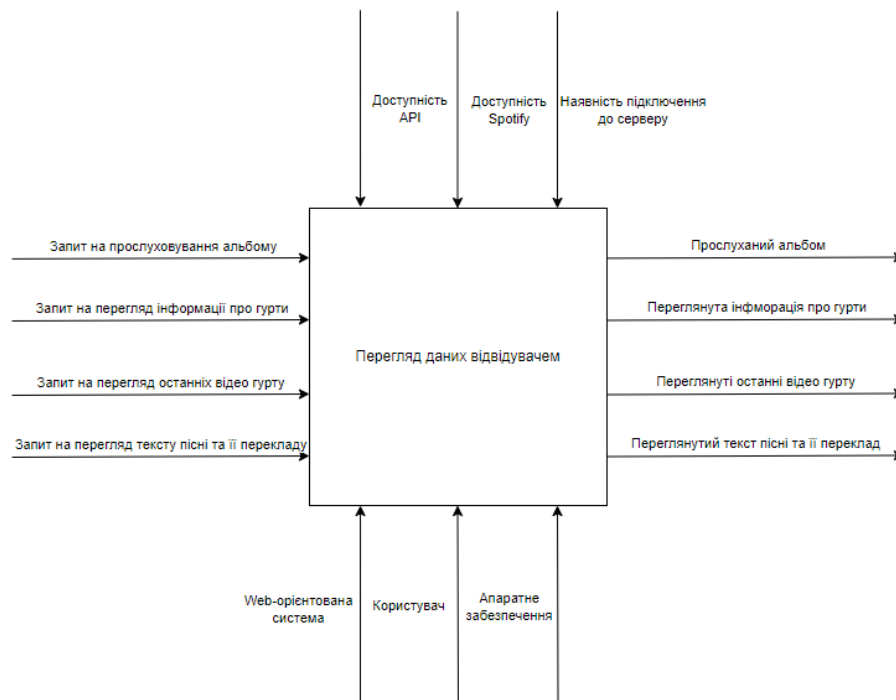


Рисунок 3.1 – Контекстна діаграма IDEF0 для користувача

*Джерело: зроблено автором (знімок з екрану)*



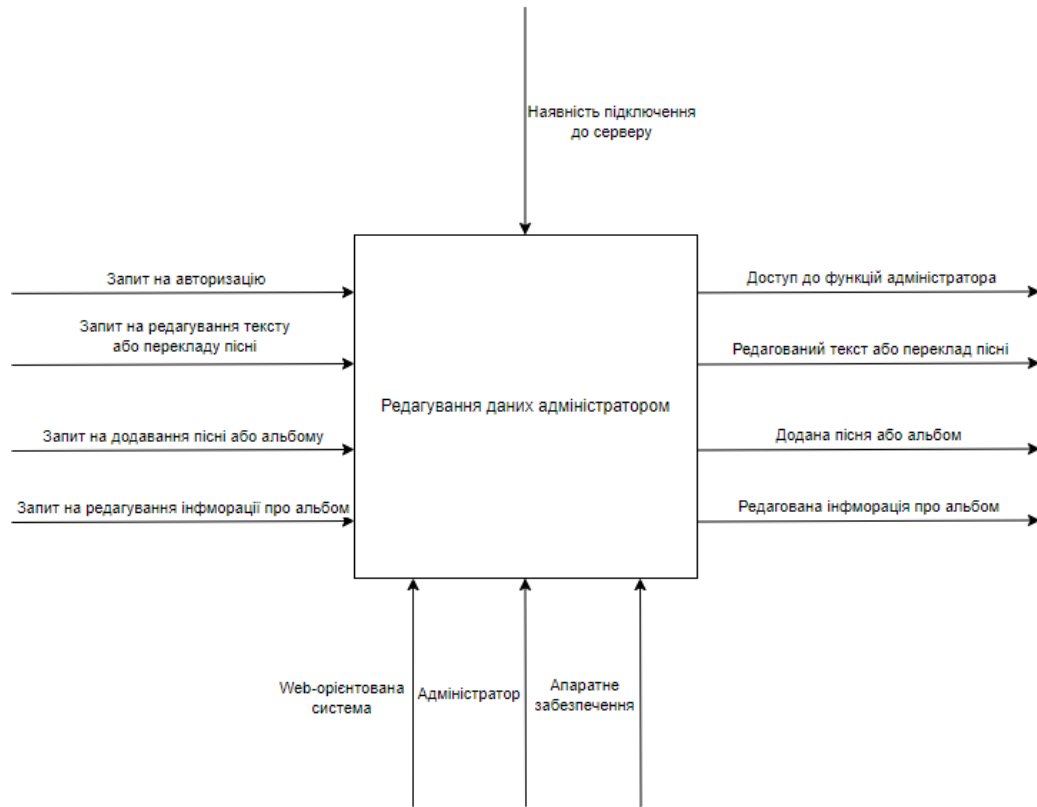


Рисунок 3.2 – Контекстна діаграма IDEF0 для адміністратора  
*Джерело: зроблено автором (знімок з екрану)*

Для деталізації процесів системи з боку користувача було виконано декомпозицію першого рівня для процесу вибору групи для відображення відео, що зображена на рисунку 3.3.

Блоки дочірньої декомпозиції діаграми IDEF0 для користувача є такі:

1. Прослуховування альбому.
2. Перегляд останніх відео обраної групи.
3. Перегляд тексту та перекладу пісні.
4. Перегляд інформації про гурти.

Для детального представлення інформації було створено таблиці входів та виходів даних для користувача (табл. 3.1-3.4) для кожного підпроцесу.

Декомпозиція діаграми прослуховування альбому представлена такими процесами:

1. Вибір групи.
2. Вибір альбому.
3. Прослуховування альбому.

Декомпозиція діаграми перегляду останніх відео обраної групи:

1. Вибір групи.
2. Пошук відео.
3. Перегляд останніх відео гурту.

Таблиця 3.1 – Вхідні та вихідні дані для діаграми прослуховування альбому

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір групи	Запит на прослуховування альбому	Доступність Spotify	Web-орієнтована система, користувач та апаратне забезпечення	Відкрита відповідна сторінка
Вибір альбому	Відкрита відповідна сторінка			Вибраний альбом
Прослуховування альбому	Вибраний альбом			Прослуханий альбом

*Джерело: побудовано автором*

Таблиця 3.2 – Вхідні та вихідні дані для діаграми перегляду останніх відео обраної групи

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір групи	Запит на перегляд останніх відео гурту	Доступність АРІ	Web-орієнтована система, користувач та апаратне забезпечення	Назва групи
Пошук відео	Назва групи			Знайдені відео
Перегляд останніх відео групи	Знайдені відео			Переглянуті останні відео групи

*Джерело: побудовано автором*

Декомпозиція діаграми перегляду інформації про гурти:

1. Вибір гурту.
2. Перегляд інформації про гурт.

Таблиця 3.3 – Вхідні та вихідні дані для діаграми перегляду інформації про гурти

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір гурту	Запит на перегляд інформації про гурт	Наявність підключення до серверу	Web-орієнтована система, користувач та апаратне забезпечення	Назва гурту
Перегляд інформації про гурт	Назва гурту			Переглянута інформація про гурт

*Джерело: побудовано автором*

Декомпозиція діаграми перегляду тексту та перекладу пісні:

1. Вибір гурту.
2. Вибір альбому.
3. Вибір пісні.
4. Відкриття тексту та перекладу.

Таблиця 3.4 – Вхідні та вихідні дані для діаграми авторизація адміністратора

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір гурту	Запит на перегляд тексту пісні та її перекладу	Наявність підключення до серверу	Web-орієнтована система, користувач та апаратне забезпечення	Назва гурту
Вибір альбому	Назва гурту			Назва альбому
Вибір пісні	Назва альбому			Назва пісні
Відкриття тексту та перекладу	Назва пісні			Переглянутий текст та переклад пісні

*Джерело: побудовано автором*

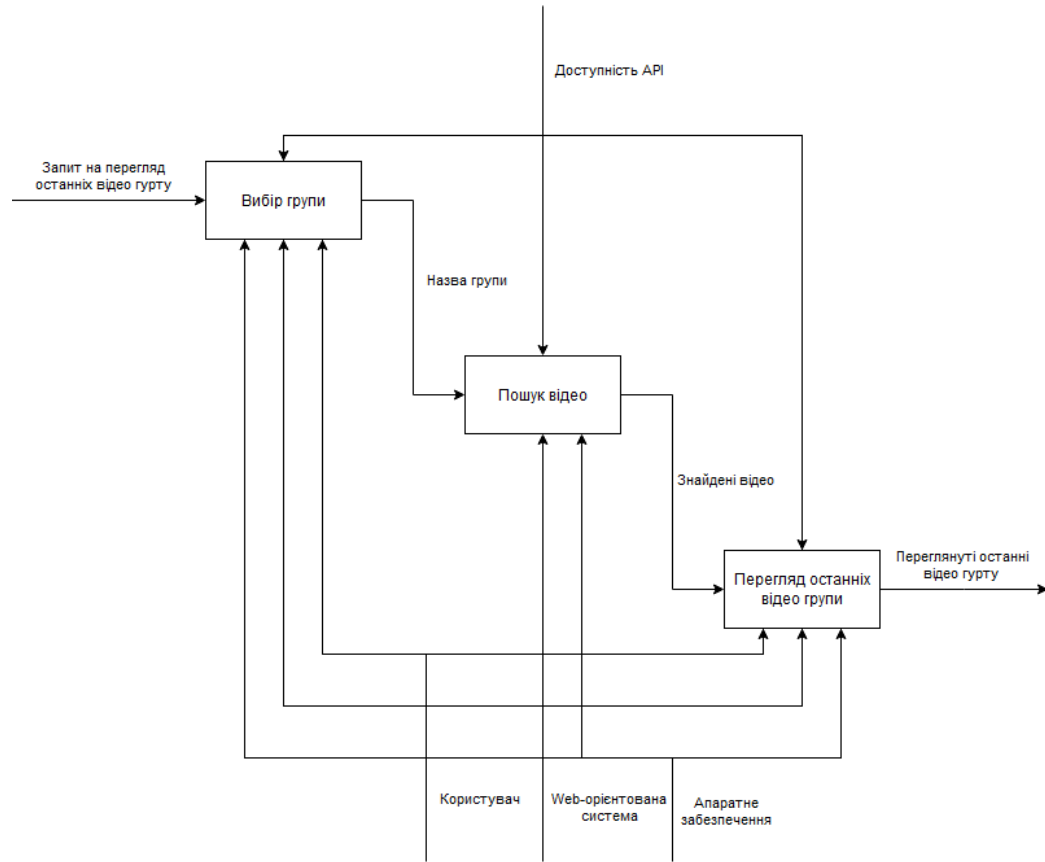


Рисунок 3.3 – Діаграма декомпозиції вибору групи для перегляду останніх відео  
*Джерело: зроблено автором (знімок з екрану)*

Для деталізації процесів системи з боку адміністратора було виконано декомпозицію першого рівня для процесу авторизації, що зображена на рисунку 3.4.

Блоки дочірньої декомпозиції діаграми IDEF0 для адміністратора є такі:

1. Авторизація.
2. Редагування тексту або перекладу пісні.
3. Додавання пісні або альбому.
4. Редагування інформації про альбоми.

Для детального представлення інформації було створено таблиці входів та виходів даних для адміністратора (табл. 3.5-3.8) для кожного підпроцесу.

Декомпозиція діаграми авторизації представлена такими процесами:

1. Перехід на сторінку.
2. Перевірка даних на сервері.
3. Авторизація.

Декомпозиція діаграми редагування тексту або перекладу пісні:

1. Вибір групи.
2. Вибір альбому.
3. Вибір пісні.
4. Вибір редагування перекладу або тексту.
5. Редагування тексту або перекладу пісні.

Таблиця 3.5 – Вхідні та вихідні дані для діаграми авторизації

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Запит на авторизацію	Перехід на сторінку	Доступність серверу	Web-орієнтована система, адміністратор та апаратне забезпечення	Введені дані користувача
Перевірка даних на сервері	Введені дані користувача			Позитивна відповідь серверу
Авторизація	Позитивна відповідь серверу			Доступ до функцій адміністратора

*Джерело: побудовано автором*

Таблиця 3.6 – Вхідні та вихідні дані для діаграми редагування тексту або перекладу пісні

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізм	Вихідні дані
Вибір групи	Запит на редагування тексту або перекладу пісні	Доступність серверу	Web-орієнтована система, адміністратор та апаратне забезпечення	Назва групи
Вибір альбому	Назва групи			Назва альбому
Вибір пісні	Назва альбому			Назва пісні
Вибір редагування тексту чи перекладу пісні	Назва пісні			Обраний переклад або текст
Редагування тексту або перекладу пісні	Обраний переклад або текст			Редагований текст або переклад пісні

*Джерело: побудовано автором*

Декомпозиція діаграми додавання пісні або альбому:

1. Вибір гурту.
2. Вибір додавання пісні чи додавання альбому.
3. Додавання пісні або альбому.

Таблиця 3.7 – Вхідні та вихідні дані для діаграми додавання пісні або альбому

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір гурту	Запит на додавання пісні або альбому	Наявність підключення до серверу	Web-орієнтована система, адміністратор та апаратне забезпечення	Назва гурту
Вибір додавання пісні чи додавання альбому	Назва гурту			Обране додавання пісні або альбому
Додавання пісні або альбому	Обране додавання пісні або альбому			Додана пісня або альбом

*Джерело: побудовано автором*

Декомпозиція діаграми редагування інформації про альбом:

1. Вибір гурту.
2. Вибір альбому.
3. Редагування інформації про альбом.

Таблиця 3.8 – Вхідні та вихідні дані для діаграми редагування інформації про альбом

Стрілка/ Підпроцес	Вхідні дані	Управління	Механізми	Вихідні дані
Вибір гурту	Запит на редагування інформації про альбом	Наявність підключення до серверу	Web-орієнтована система, адміністратор та апаратне забезпечення	Назва гурту
Вибір альбому	Назва гурту			Назва альбому
Редагування інформації про альбом	Назва альбому			Редагована інформація про альбом

*Джерело: побудовано автором*



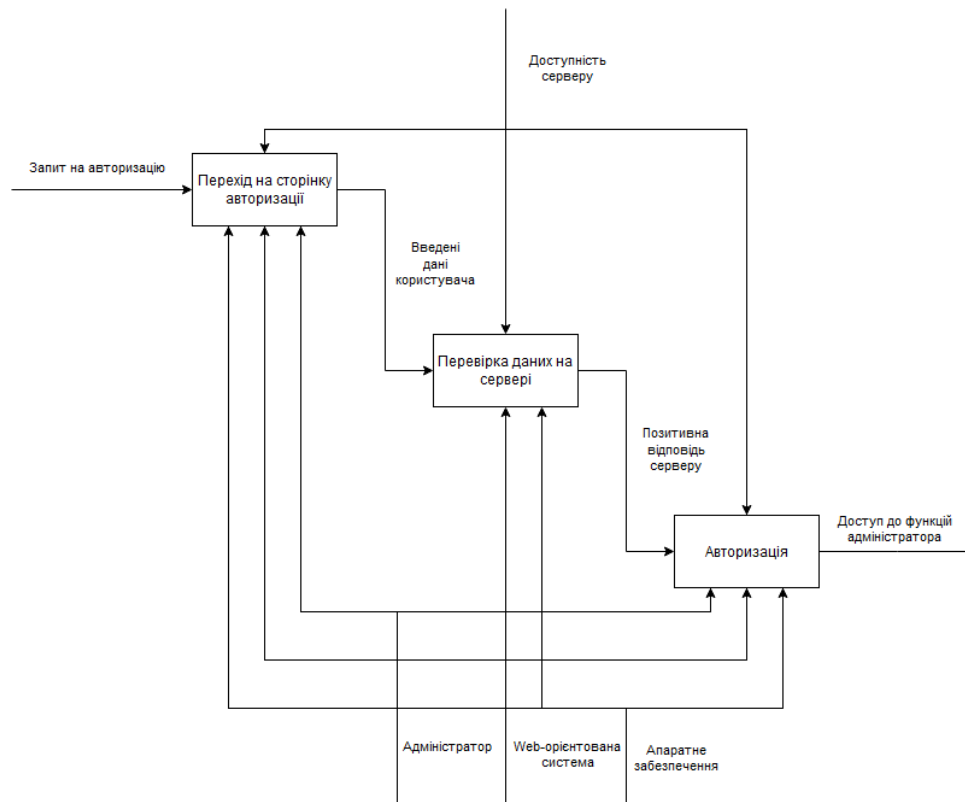


Рисунок 3.4 – Діаграма декомпозиції авторизації

*Джерело: зроблено автором (знімок з екрану)*

### 3.2 Діаграма Use Case

Після виконання моделювання та опису бізнес-процесів майбутньої web-орієнтованої системи необхідно створити діаграму варіантів використання. За допомогою Use Case узагальнюються всі процеси, які відбуваються в даній системі. Основою даної діаграми є актори, зв'язки, примітки та механізми розширення [26].

Для web-орієнтованої системи музичних гуртів напрямку К-поп було виділено такі варіанти використання:

- перегляд відео гуртів;
- редагування тексту пісні та перекладу;

- прослуховування альбому;
- авторизація адміністратора;
- перегляд інформації про групу;
- редагування опису альбому;
- додавання пісні або альбому.

Акторами діаграми Use Case є адміністратор та користувач.

Діаграма використання web-орієнтованої системи музичних гуртів напрямку К-поп зображена на рисунку 3.1

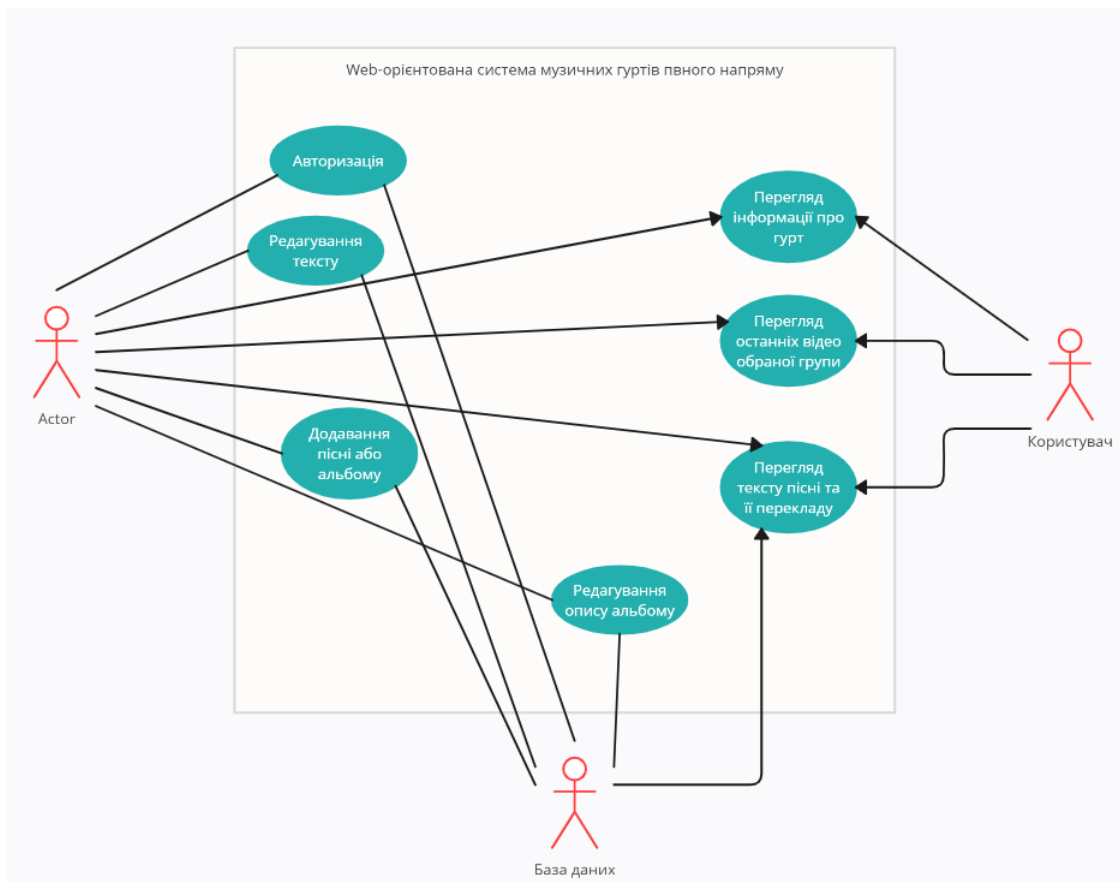


Рисунок 3.5 – Діаграма Use Case

*Джерело: зроблено автором (знімок з екрану)*

### 3.3 Проектування бази даних

База даних виступає основою складовою інформаційної системи, яка включає в себе зібрані дані з конкретної області, які вимагають управління. Для здійснення цього управління застосовуються системи управління базами даних (СУБД).

Структура бази даних інформаційної системи включає наступні таблиці:

Таблиця «Login» - таблиця, в якій зберігається інформація для авторизації.

Таблиця «album» - таблиця, в якій зберігається інформація про альбоми.

Таблиця «group\_kpop» - таблиця, в якій зберігається інформація групи.

Таблиця «song» - таблиця, в якій зберігається інформація про пісні.

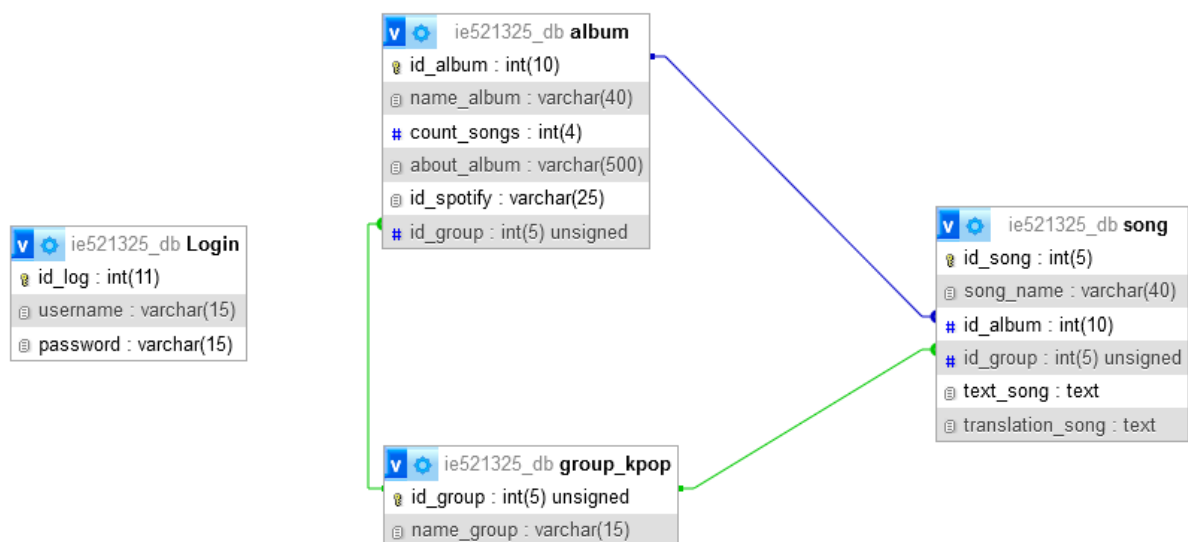


Рисунок 3.6 – Схема реалізованої бази даних засобами phpMyAdmin

*Джерело: зроблено автором (знімок з екрану)*

## 4 РЕАЛІЗАЦІЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ МУЗИЧНИХ ГУРТІВ НАПРЯМУ K-POP

### 4.1 Розробка архітектури системи

Архітектура web-сайту є основою для взаємодії користувача з ним. Було вирішено розробити веб-сайт за клієнт-серверною архітектурою [27].

Дана архітектура містить в собі такі компоненти: клієнт(браузер: JavaScript, HTML, CSS), сервери(на яких знаходиться веб-сайт, а також серверна частина коду на PHP) та сервер бази даних, яка використовується сайтом (MySQL). Обмін між клієнтом та сервером відбувається за допомогою http-запитів [28].

В браузері оброблюється клієнтська частина сторінки, структура якої написана за допомогою мови розмітки гіпертексту HTML[29]. За відображення елементів веб-сторінки відповідають каскадні таблиці стилів CSS [30]. За відправлення даних на сервер і їх обробку, а також за надання динамічності відповідає скриптова мова програмування JavaScript [31].

На сервері здійснюється обробка отриманих запитів від клієнта за допомогою серверної мови програмування PHP [32]. Вона також відповідає за надсилання відповідей на клієнт і запитів до сервера бази даних, а також обробку отриманих результатів.

Сервер бази даних представлений СУБД MySQL [33]. Він відповідає за збереження даних сайту, а також отримання запитів і надсилання результатів запиту до серверу.

На рис. 4.1 представлена архітектура веб-сайту.

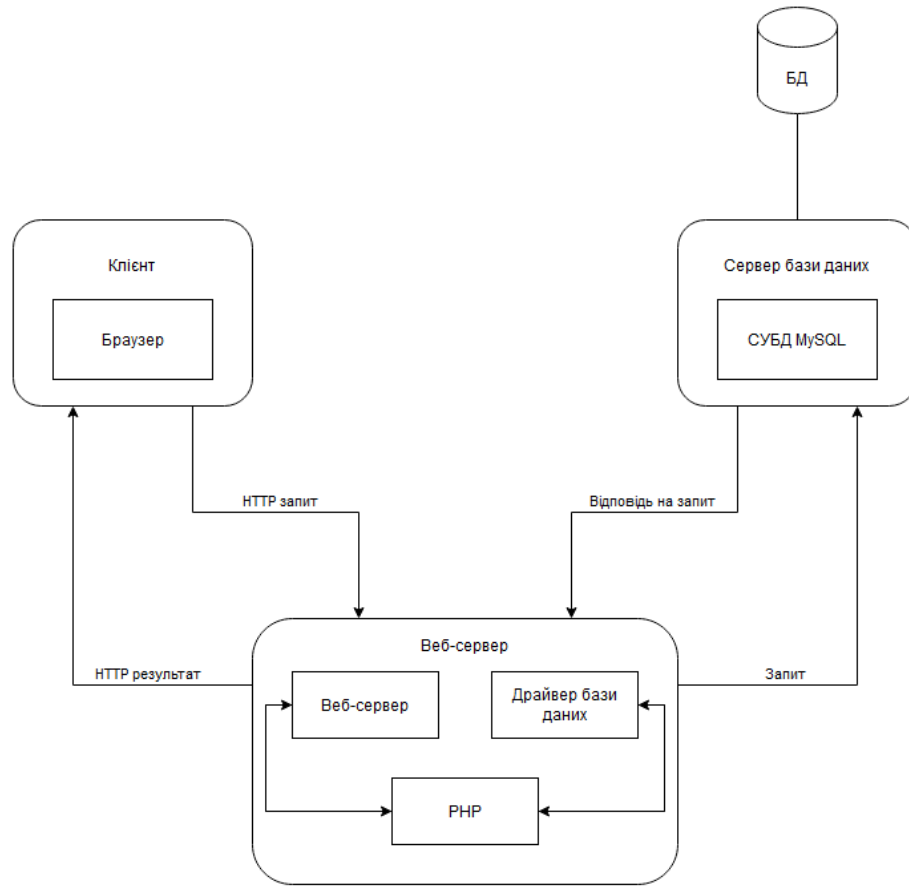


Рисунок 4.1 – Архітектура веб-орієнтованої системи

*Джерело: зроблено автором (знімок з екрану)*

## 4.2 Встановлення та розробка компонентів web-орієнтованої системи

Першим етапом в розробці web-орієнтованої системи було створення шаблону веб-сторінок, використовуючи HTML та CSS. За допомогою CSS значення media [34] було реалізовано адаптивність веб-орієнтовано системи під різні девайси, починаючи від невеликих смартфонів, закінчуючи звичайним монітором комп'ютера. На рисунку 4.2 продемонстровано зовнішній вигляд головної сторінки на телефоні.

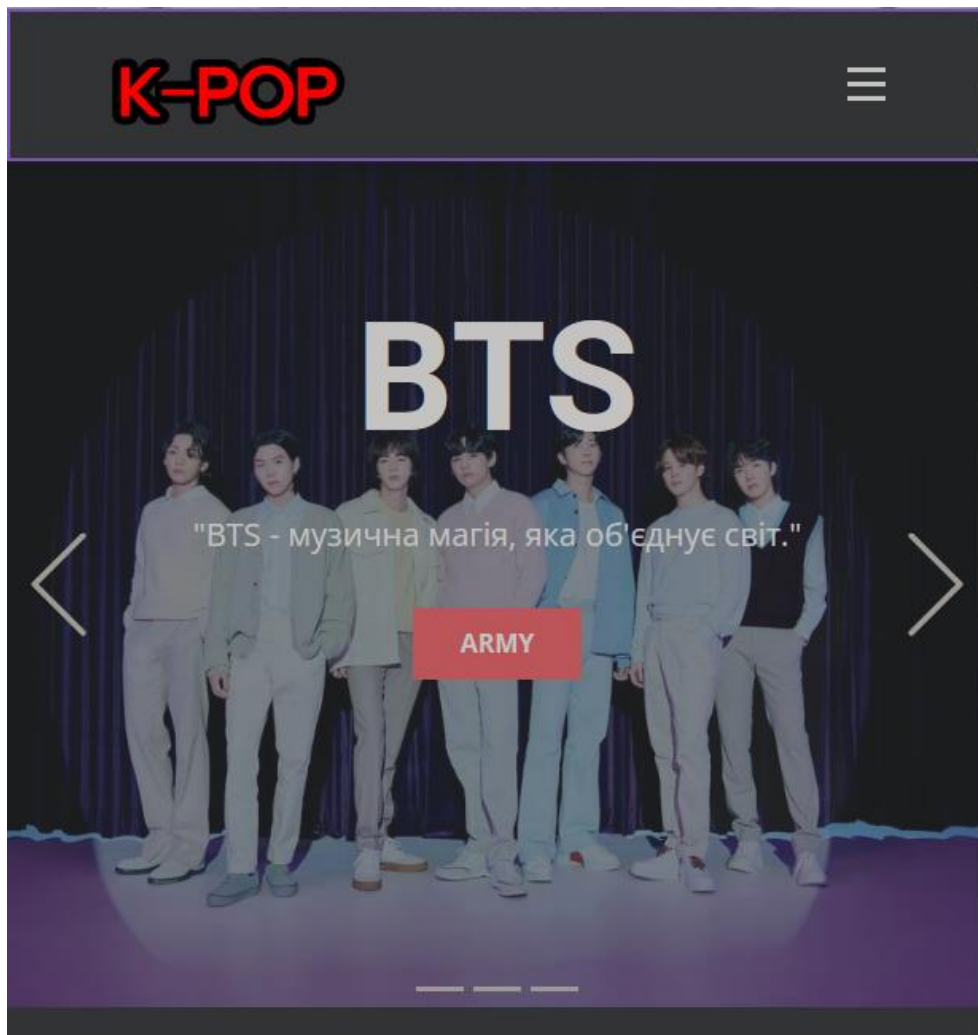


Рисунок 4.2 – Демонстрація зовнішнього вигляду головної сторінки

*Джерело: зроблено автором (знімок з екрану)*

Далі було додано функціоналу для слайдера засобами JavaScript. Наступним етапом було створення доступу до YouTube відео за допомогою YouTube-API [35]. Використовуючи його та JavaScript, було реалізовано вивід останніх відео, а також вибір групи, якої потрібно вивести на екран (рис. 4.3).і

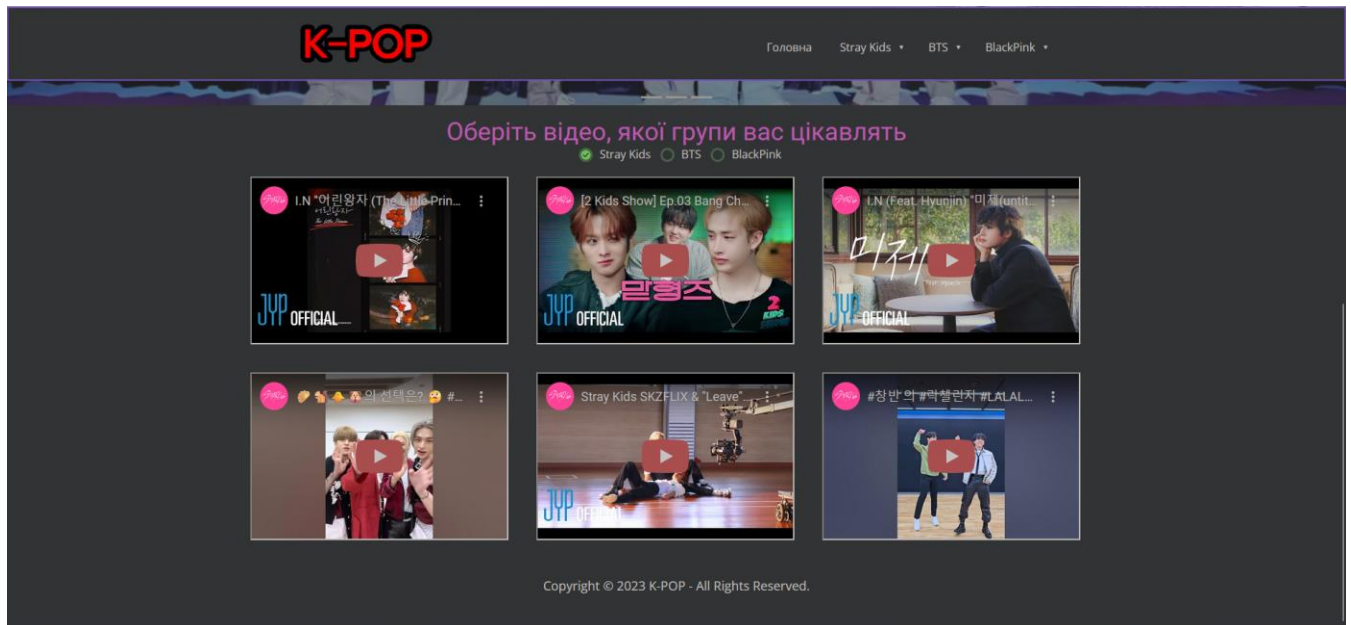


Рисунок 4.3 – Виведення останніх відео

*Джерело: зроблено автором (знімок з екрану)*

Наступним етапом є створення та заповнення бази даних. В результаті була засобами phpmyadmin була отримана структура зображена на рисунку 4.4.

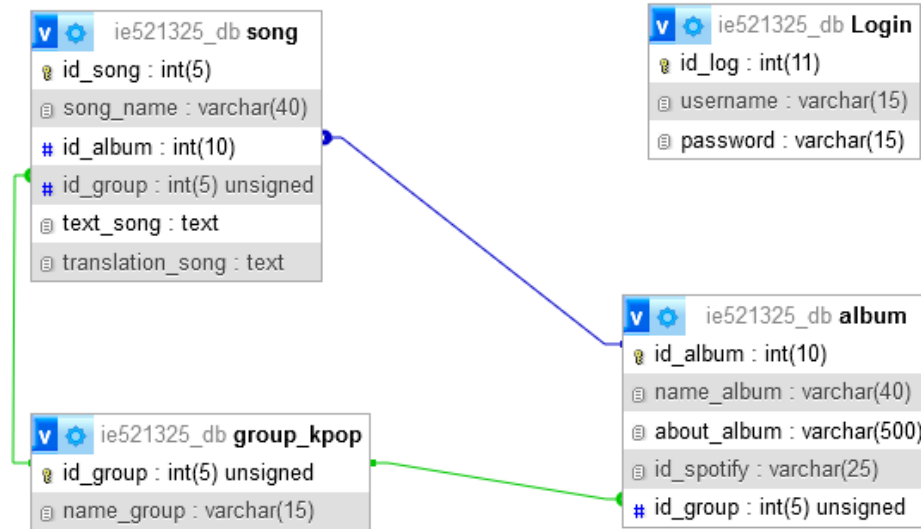


Рисунок 4.4 – Структура реалізованої бази даних засобами phpMyAdmin

*Джерело: зроблено автором (знімок з екрану)*

Для заповнення бази даних описами альбомів, текстами пісень, їх перекладами використовувався Genius [36]. Це онлайн-платформа, на котрій розміщено всі вищезгадані дані. Також для заповнення рядку SpotifyID використовувався додаток Spotify [37]. Він дозволяє отримати код для інтегрування на сторінку, де власне знаходиться ID альбому. Всі таблиці мають зв'язки один між одним, що здійснено за допомогою зовнішніх ключів [38].

Далі було створено скрипти js для взаємодії з сервером за допомогою метода fetch [39]. Цей метод дозволяє відправити дані на сервер і отримати відповідь не перезавантажуючи сторінку, що дуже спрощує користування веб-орієнтованою системою. Кожен зі створених скриптів надсилає дані відповідним php-файлам, які були створені наступними, і отримують відповідь від них.

Створені php-файли оброблюють отримані дані, бо вони надсилаються на сервер у випадку паролю та логіну в зашифрованому вигляді і у форматі json [40],



котрий потрібно розшифрувати. Розшифровані дані оброблюються, для уникнення sql-ін'єкцій. Потім в цих файлах виконується запит до MySQL, результатом якого є таблиця або рядок, котрий знову ж таки пакується до json і надсилається на клієнтську частину.

### 4.3 Розміщення на хостингу та тестування web-орієнтованої системи музичних гуртів

Перед тестування системи було розміщено на хостингу та окремому домені. Це дозволило одразу перевіряти її в умовах близьких до реального використання. Хостинг має такі параметри:

- Версія PHP: 8.3.
- Версія MySQL: 5.7.
- Максимально доступний об'єм оперативної пам'яті: 512МБ.

Тестування веб-орієнтованої системи можна розділити на тестування функцій, котрі доступні звичайному користувачеві, та функції адміністратора, тому було вирішено створити дві таблиці з результатами тестувань основних функцій (табл. 4.1-4.2).

Таблиця 4.1 – Результати функціонального тестування функцій користувача

№	Тестовий сценарій	Опис тесту	Очікуваний результат	Фактичний результат	Відмітка про пройдений тест
1	Перегляд відео обраної групи	Вибираємо необхідну групу на головній сторінці	Виведення на сторінку 6 останніх відео гурту	Виведення на сторінку 6 останніх відео гурту	Так
2	Перегляд інформації про гурт	Перехід на відповідну сторінку групи, натискання кнопок кожного з учасників	Відкриття інформації про учасників і закриття після потворного натискання	Відкриття інформації про учасників і закриття після потворного натискання	Так

Продовження таблиці 4.1

3	Перегляд інформації про альбом	Перехід на відповідну сторінку, вибір альбому	Відображення інформації про альбом	Відображення інформації про альбом	Так
4	Прослуховування альбому	Перехід на відповідну сторінку, вибір альбому, натискання клавіші прослуховування	Прослуханий альбом	Прослуханий альбом	Так
5	Перегляд тексту та перекладу пісні	Перехід на відповідну сторінку, вибір альбому, вибір пісні, натискання клавіші тексту або перекладу	Відображення тексту або перекладу відповідної пісні	Відображення тексту або перекладу відповідної пісні	Так

*Джерело: побудовано автором*

У результаті тестування був перевірений функціонал доступний звичайному користувачеві, всі функції, котрі були реалізовані, коректно працюють.

Наступними тестувались функції адміністратора.

Таблиця 4.2 – Результати функціонального тестування функцій адміністратора

№	Тестовий сценарій	Опис тесту	Очікуваний результат	Фактичний результат	Відмітка про пройдений тест
1	Авторизація	Перехід на сторінку авторизації за спеціальним посиланням, введення даних	Повідомлення про успішну авторизацію, перехід на головну сторінку	Повідомлення про успішну авторизацію, перехід на головну сторінку	Так

Продовження таблиці 4.2

2	Доступність кнопок на сторінці альбомів та пісень	Перехід на сторінку «Альбоми та пісні» будь-якої з груп	Відображення кнопок редагування та додавання або видалення різних даних	Відображення кнопок редагування та додавання або видалення різних даних	Так
3	Редагування тексту пісні або перекладу	Натискання відповідних кнопок для редагування перекладу або тексту	Редагований переклад або текст	Редагований переклад або текст	Так
4	Редагування опису альбому	Натискання відповідної кнопки	Редагований опис альбому	Редагований опис альбому	Так
5	Додавання пісні	Натискання відповідної кнопки, введення даних у відповідні поля на сторінці, натискання кнопки додати пісню	Додана нова пісня	Додана нова пісня	Так
6	Додавання альбому	Натискання відповідної кнопки, введення даних у відповідні поля на сторінці, натискання кнопки додати альбом	Доданий новий альбом	Доданий новий альбом	Так
7	Видалення пісні	Вибір альбому, вибір пісні, натискання кнопки видалення пісні	Видалення пісні з бази даних, оновлення списку	Видалення пісні з бази даних, оновлення списку	Так

*Джерело: побудовано автором*

#### 4.4 Результат реалізації веб-орієнтованої системи музичних гуртів напряму К-поп

Для початку користувач вводить URL-адресу сайту: <https://k-pop-diplom-bilan.fun/>. Відкриється головна сторінка веб-сайту – Головна (рис. 4.5). Дана сторінка є умовною візитівкою. У верхній частині знаходиться меню: Головна, Stray Kids, BTS, BlackPink, також меню містить підменю, яке здійснює перехід на сторінку Альбоми та пісні певної групи, в залежності від того, підпункт, якого гурту натиснули. Також розміщено слайдер з фото кожного із гуртів, їх назвою, девізом, назвою фан-бази і посиланням на сторінку про них.

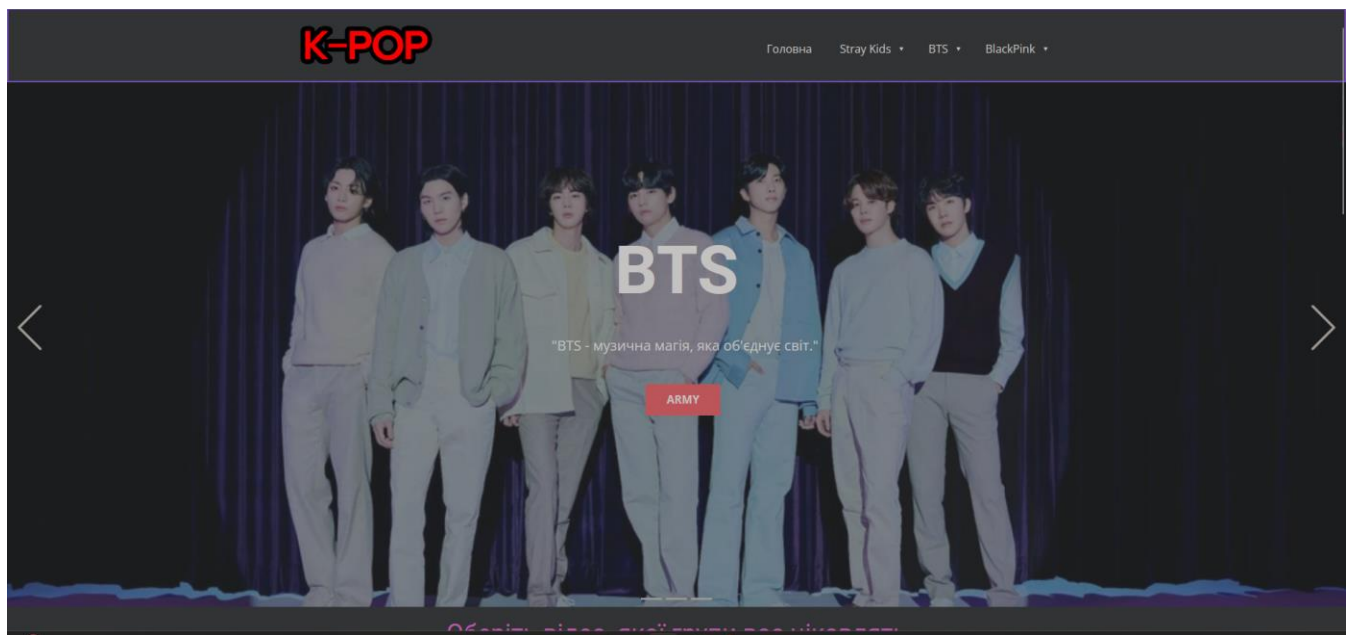


Рисунок 4.5 – Головна сторінка веб-системи

*Джерело: зроблено автором (знімок з екрану)*

Після слайдеру знаходиться вибір гурту (рис. 4.6), якщо натиснути на відповідну групу, то завантажаться останні їх відео.

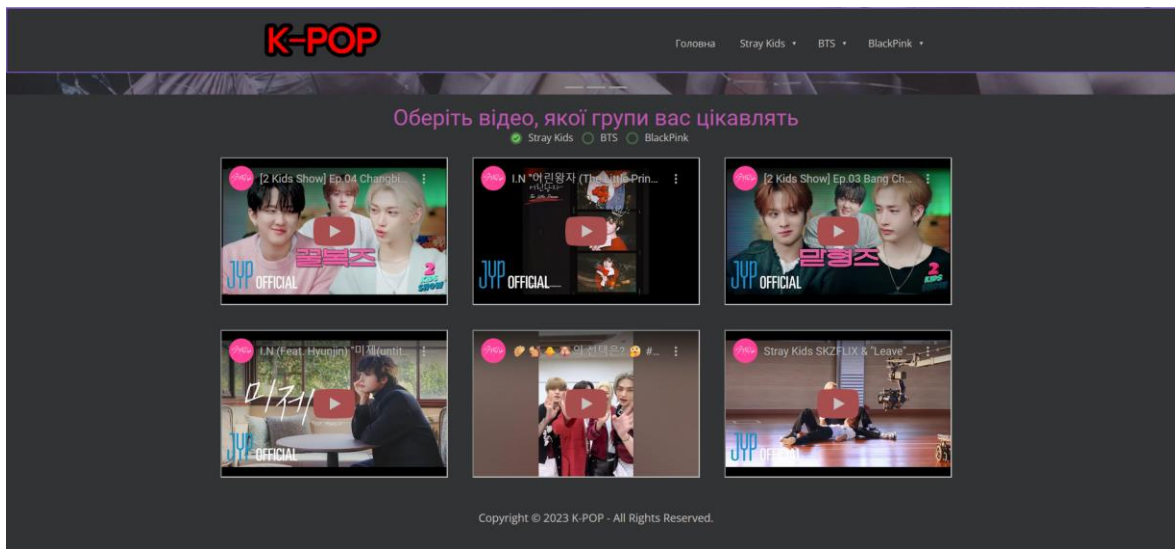


Рисунок 4.6 – Вибраний гурт і відображення відео

*Джерело: зроблено автором (знімок з екрану)*

Переходячи за посиланням Stray Kids, з'являється сторінка про гурт з основною інформацією та фото (рис. 4.7). Опускаючись нижче, бачимо кнопки, натискаючи які відображається інформація про кожного з учасників гурту і, якщо натиснути повторно, приховується, також біля футеру знаходяться посилання на соціальні мережі цієї групи (рис. 4.8).

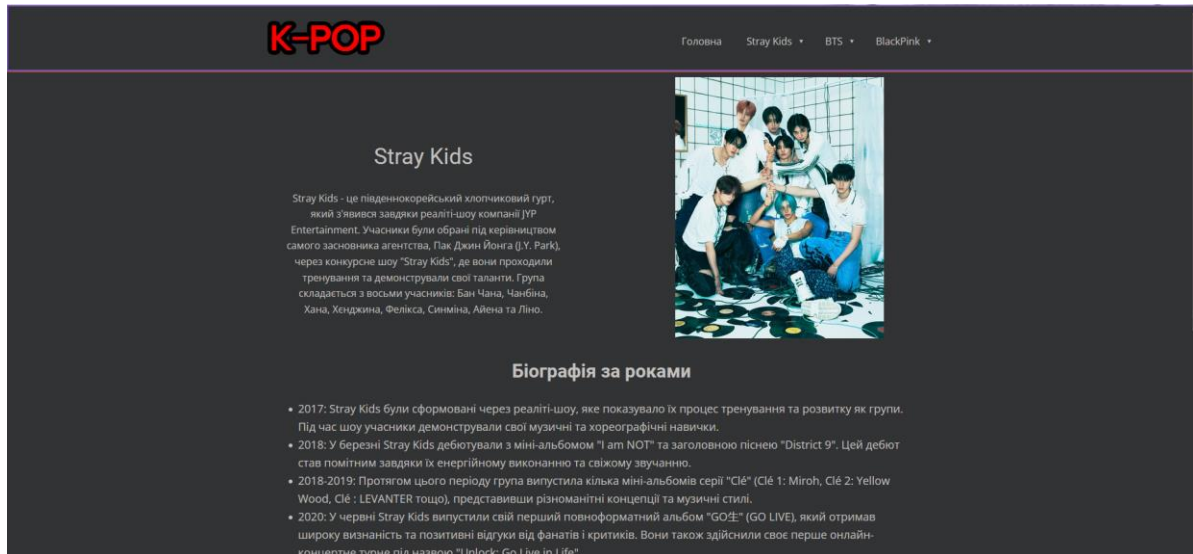


Рисунок 4.7 – Сторінка Stray Kids, відображення основної інформації  
Джерело: зроблено автором (знімок з екрану)

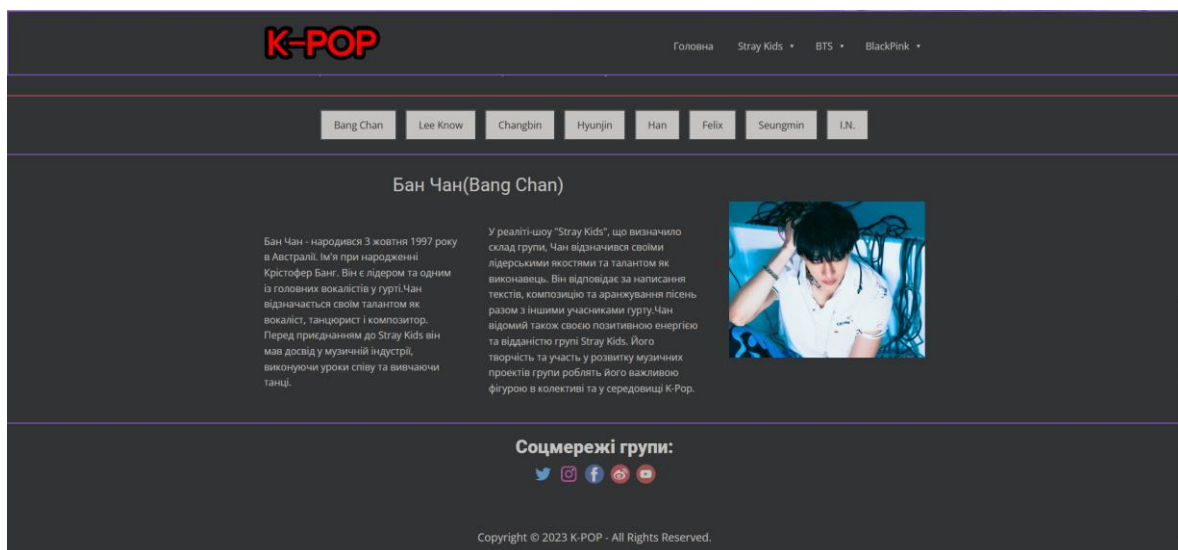


Рисунок 4.8 – Відображення інформації про учасника та соцмережі Stray Kids  
Джерело: зроблено автором (знімок з екрану)

Сторінки BTS та BlackPink мають таку саму структуру, відмінність лише у кольорі фону, він підібраний відповідно до кольору гурту (рис. 4.9-4.12).



Рисунок 4.9 – Сторінка BTS, відображення основної інформації  
*Джерело: зроблено автором (знімок з екрану)*



Рисунок 4.10 – Відображення інформації про учасника та соцмережі BTS  
*Джерело: зроблено автором (знімок з екрану)*





Рисунок 4.11 – Сторінка BlackPink, відображення основної інформації  
*Джерело: зроблено автором (знімок з екрану)*

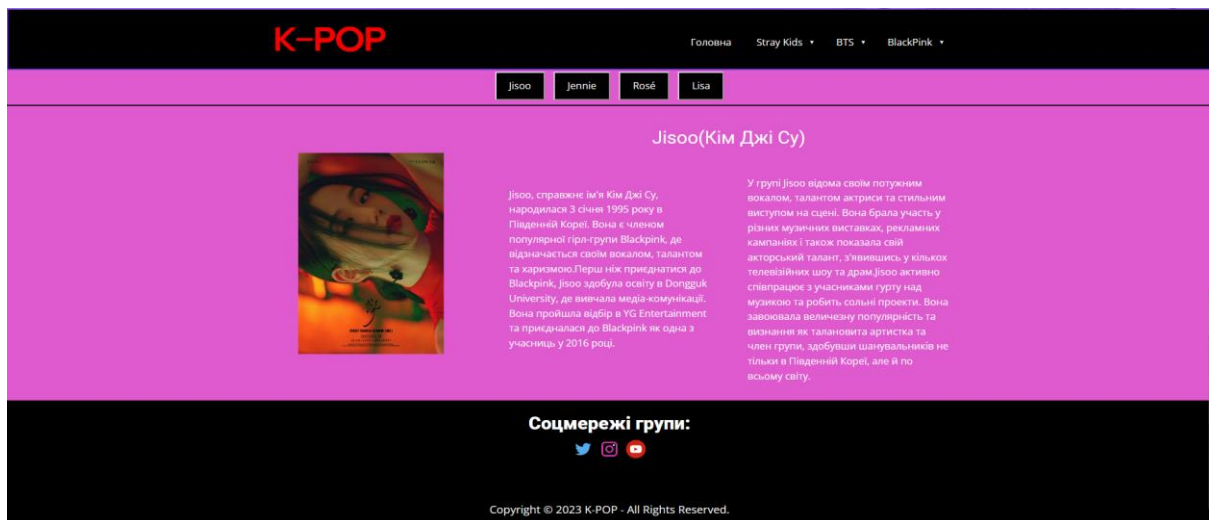


Рисунок 4.12 – Відображення інформації про учасницю та соцмережі BlackPink  
*Джерело: зроблено автором (знімок з екрану)*

Після переходу за посиланням Альбоми та пісні, потрапляємо на сторінку альбомів та пісень відповідного гурту, підпункт якого натиснули (рис. 4.13). На

даній сторінці можемо обрати альбом відповідної групи, прочитати інформацію про цей альбом, прослухати його, обрати пісню з цього альбому, прочитати її текст та переклад, котрі до натискання на відповідні тексти сховані, повторне натискання знову приховує (рис. 4.14).

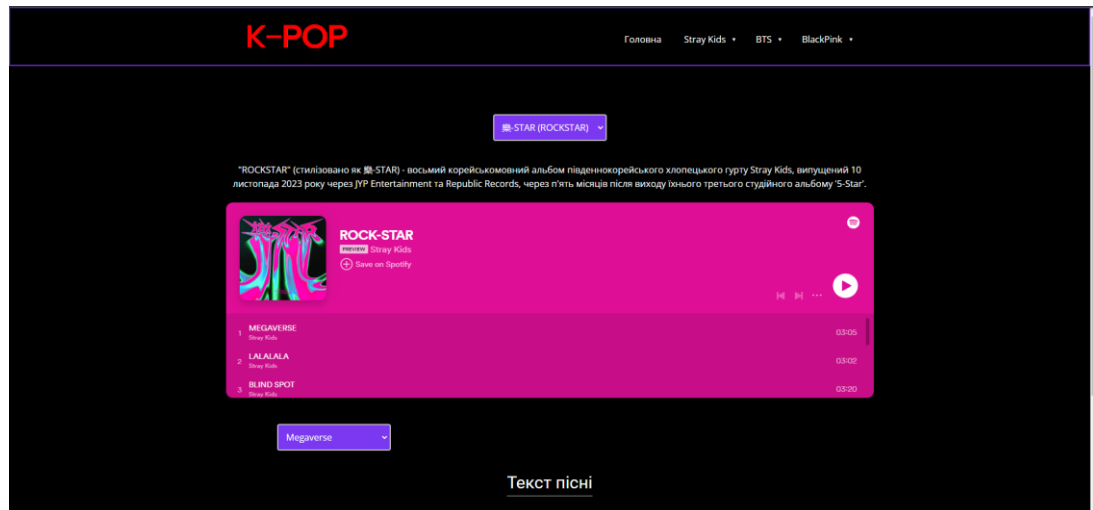


Рисунок 4.13 Сторінка Альбоми та пісні, опис альбому, його прослуховування

*Джерело: зроблено автором (знімок з екрану)*

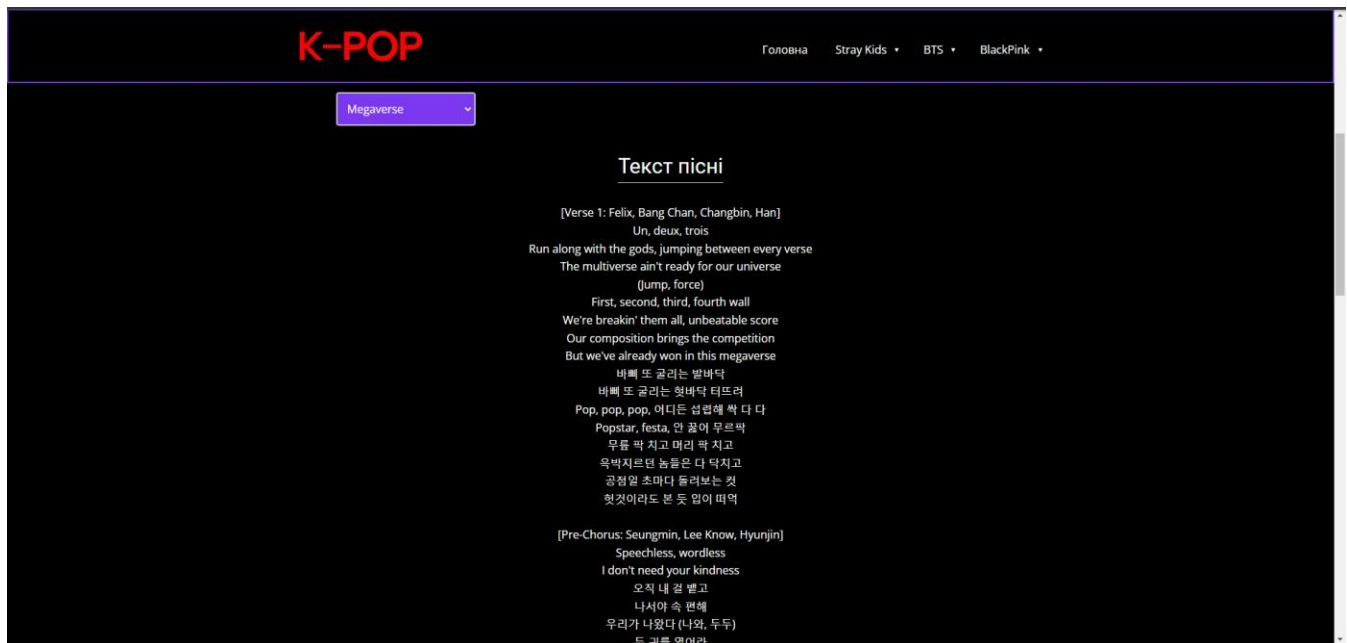


Рисунок 4.14 Перегляд тексту пісні відповідного альбому

*Джерело: зроблено автором (знімок з екрану)*

Веб-орієнтована система має різні версії в залежності від розміру екрану. Приклад головної сторінки на смартфоні розміщено на рисунку 4.15.

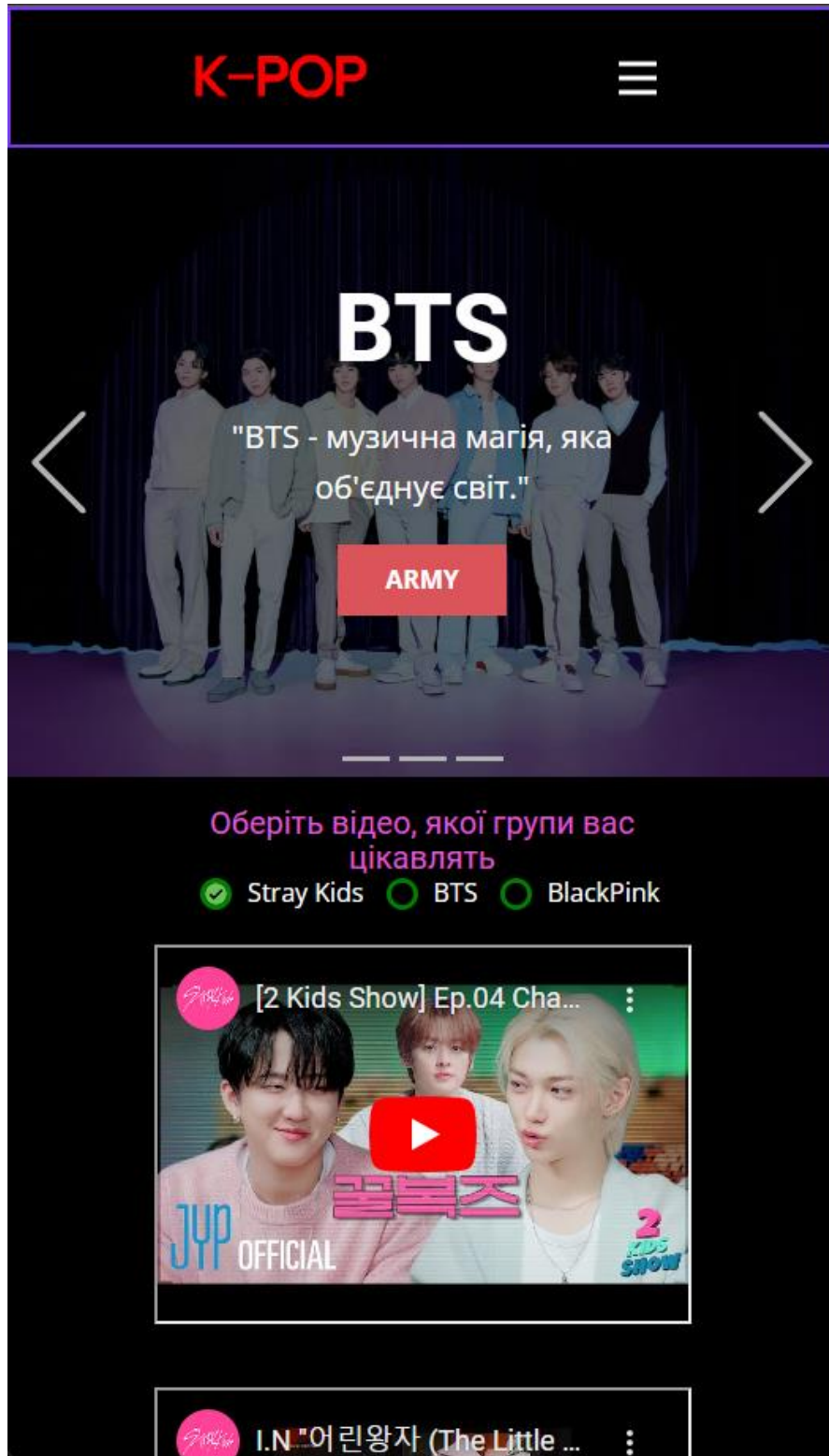


Рисунок 4.15 – Головна сторінка на мобільних пристроях

*Джерело: зроблено автором (знімок з екрану)*

Авторизація адміністратора доступна лише за спеціальним посиланням. Це дозволяє обмежити доступ до сторінки з боку інших користувачів, не адміністраторів. На сторінці знаходиться форма для вводу логіну та паролю і кнопка для підтвердження входу (рис. 4.16).

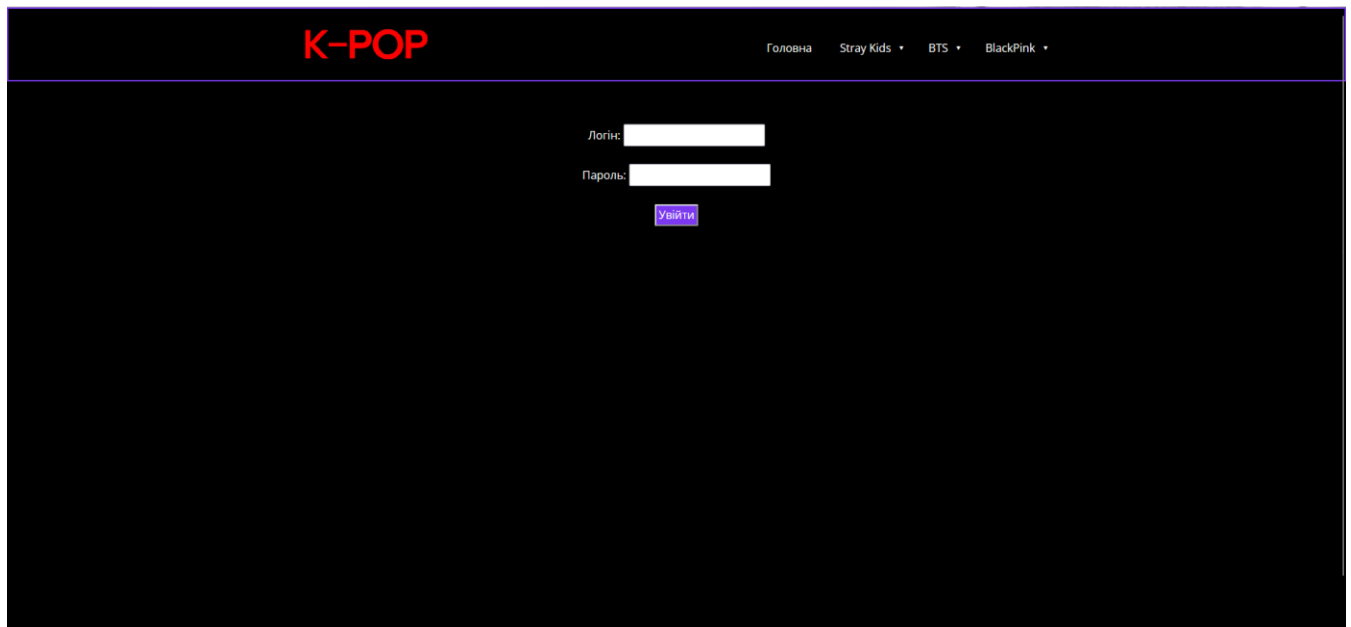


Рисунок 4.16 – Сторінка авторизації адміністратора

*Джерело: зроблено автором (знімок з екрану)*

Після успішної авторизації створюється cookies, що користувач авторизований. Тепер перейшовши на сторінку Альбоми та пісні, стають доступні кнопки редагування, додавання та видалення (рис. 4.17-4.18).

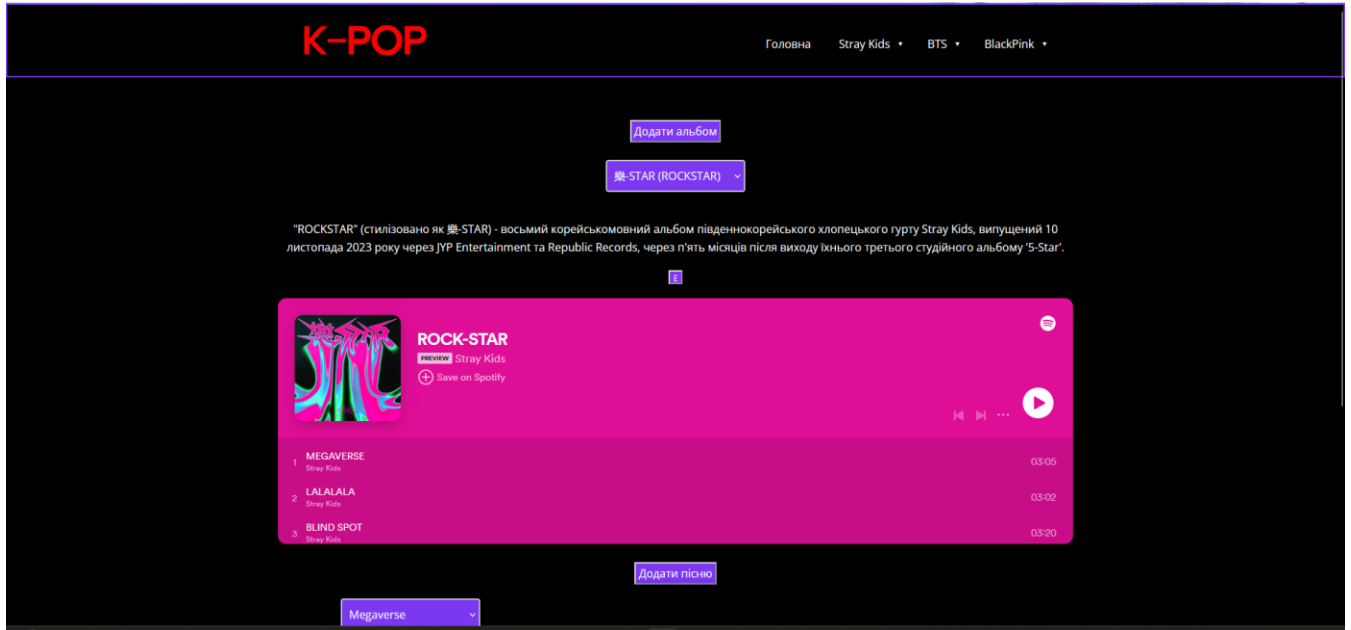


Рисунок 4.17 – Сторінка Альбоми та пісні після авторизації

*Джерело: зроблено автором (знімок з екрану)*

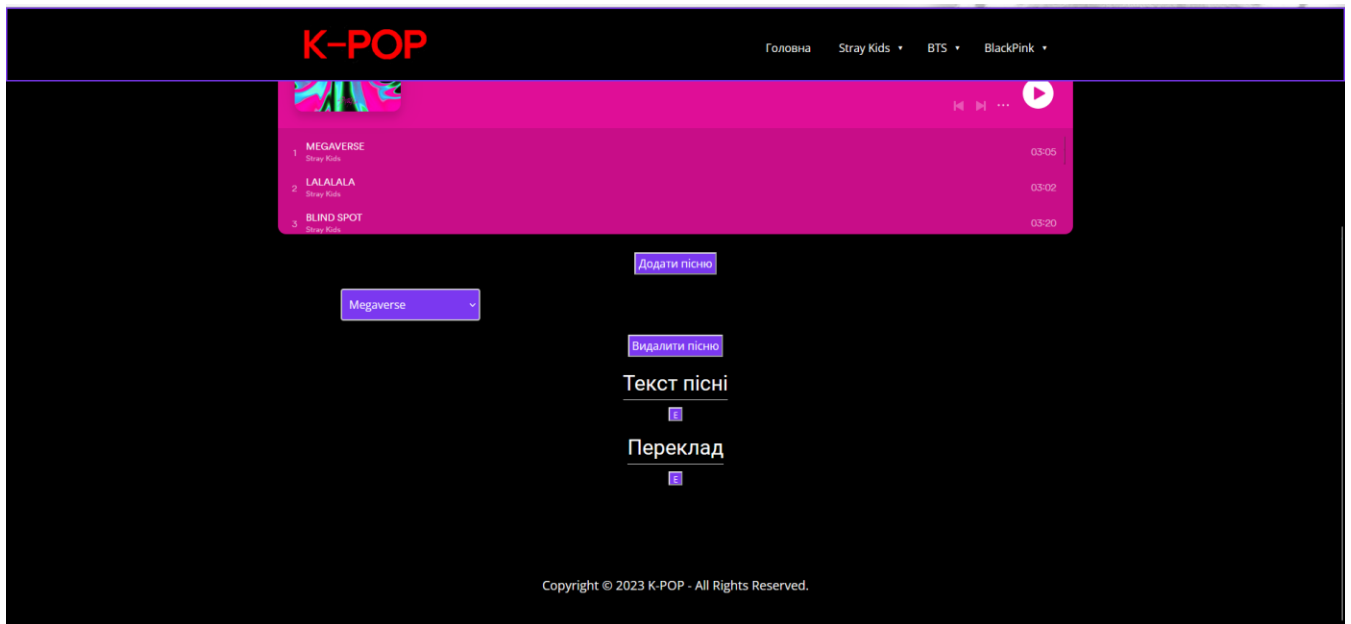


Рисунок 4.18 – Сторінка Альбоми та пісні, доступ до зміни тексту та перекладу

*Джерело: зроблено автором (знімок з екрану)*

Якщо натиснути додати альбом, перекидає на іншу сторінку, де можливо обрати зі списку гурт, для якого ми створюємо альбом, ввести назву, опис, а також SpotifyID, з інструкцією по його знаходженню (рис. 4.19).

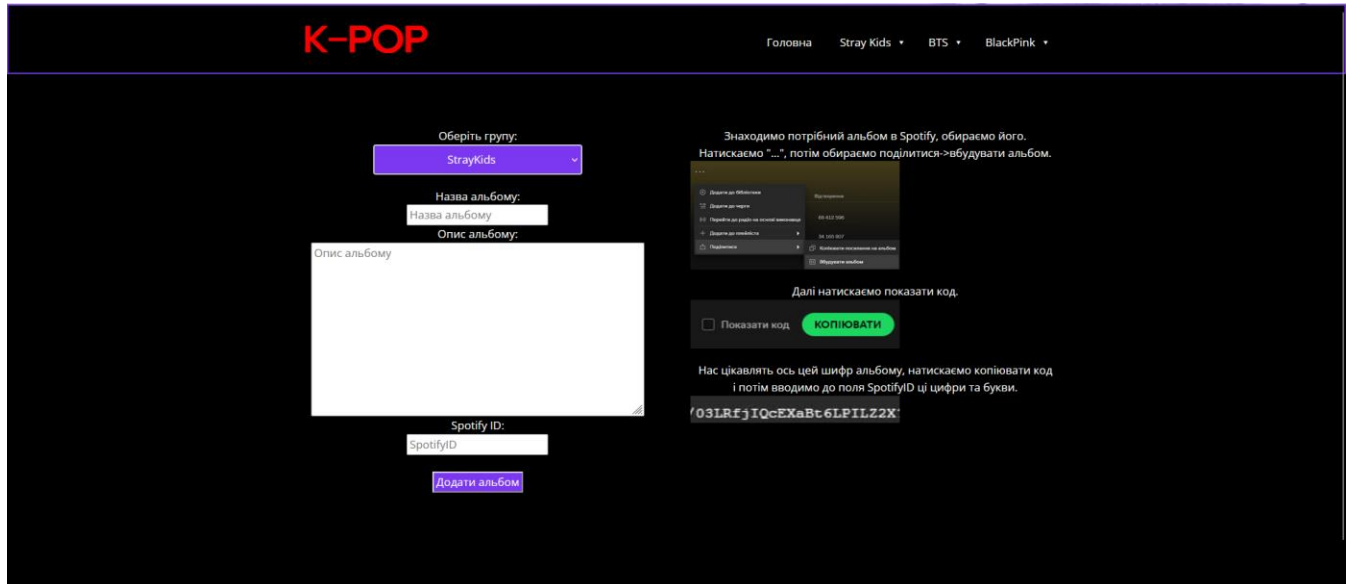
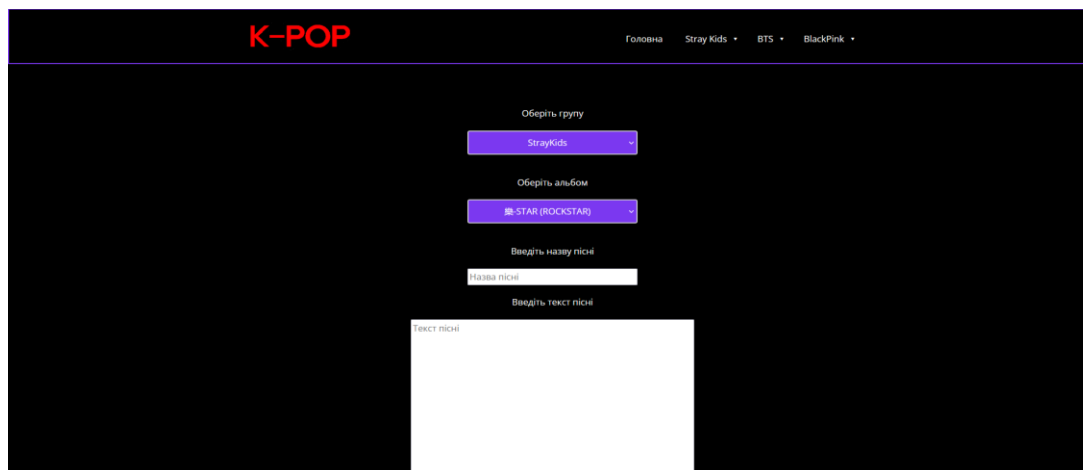


Рисунок 4.19 – Сторінка додавання альбому  
*Джерело: зроблено автором (знімок з екрану)*

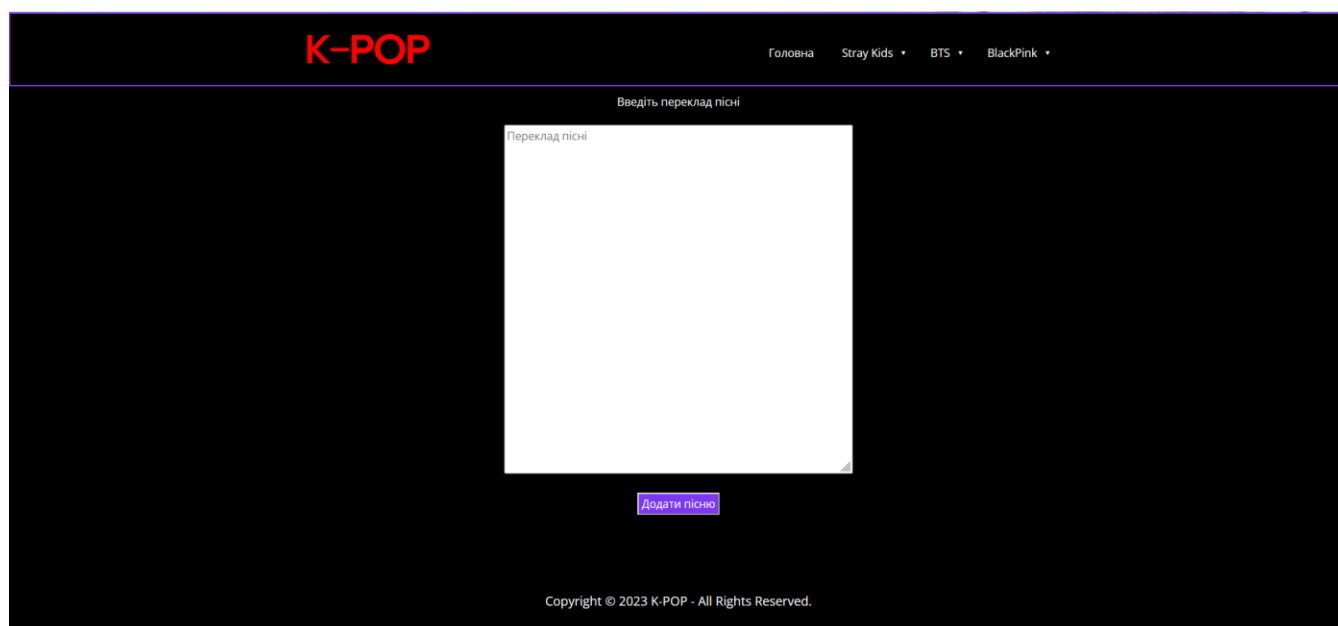
Але щоб даний альбом відобразився в списку альбомів на сторінці Альбоми та пісні, нам потрібно додати до нього пісні. Тому треба повернутися назад до неї, і натиснути кнопку додати пісню. Потрапляємо на відповідну сторінку, де в нас є можливість вибору гурту, альбомів цього гурту, вводу назви пісні, її тексту та перекладу (рис. 4.20-4.21). І лише після додавання пісні через дану форму, вона відобразиться разом з альбомом на сторінці Альбоми та пісні.



The screenshot shows the K-POP website's interface for adding a song. At the top, the logo 'K-POP' is in red, and navigation links for 'Головна', 'Stray Kids', 'BTS', and 'BlackPink' are visible. The main form is centered and includes the following elements:

- A dropdown menu labeled 'Оберіть групу' (Select group) with 'StrayKids' selected.
- A dropdown menu labeled 'Оберіть альбом' (Select album) with 'STAR (ROCKSTAR)' selected.
- A text input field labeled 'Введіть назву пісні' (Enter song name) with the placeholder 'Назва пісні'.
- A text input field labeled 'Введіть текст пісні' (Enter song lyrics) with the placeholder 'Текст пісні'.

Рисунок 4.20 – Сторінка додавання пісні: вибір гурту, альбому, ввід назви, тексту  
*Джерело: зроблено автором (знімок з екрану)*



This screenshot shows the next step in the song addition process. The form now features a large text area labeled 'Введіть переклад пісні' (Enter song translation) with the placeholder 'Переклад пісні'. Below this area is a purple button labeled 'Додати пісню' (Add song). At the bottom of the page, a copyright notice reads: 'Copyright © 2023 K-POP - All Rights Reserved.'

Рисунок 4.21 – Сторінка додавання пісні: ввід перекладу та кнопка додати пісню  
*Джерело: зроблено автором (знімок з екрану)*

Якщо при додаванні альбому буде неправильно введений SpotifyID, отримуємо ось таку помилку, котра зображена на рисунку 4.22.



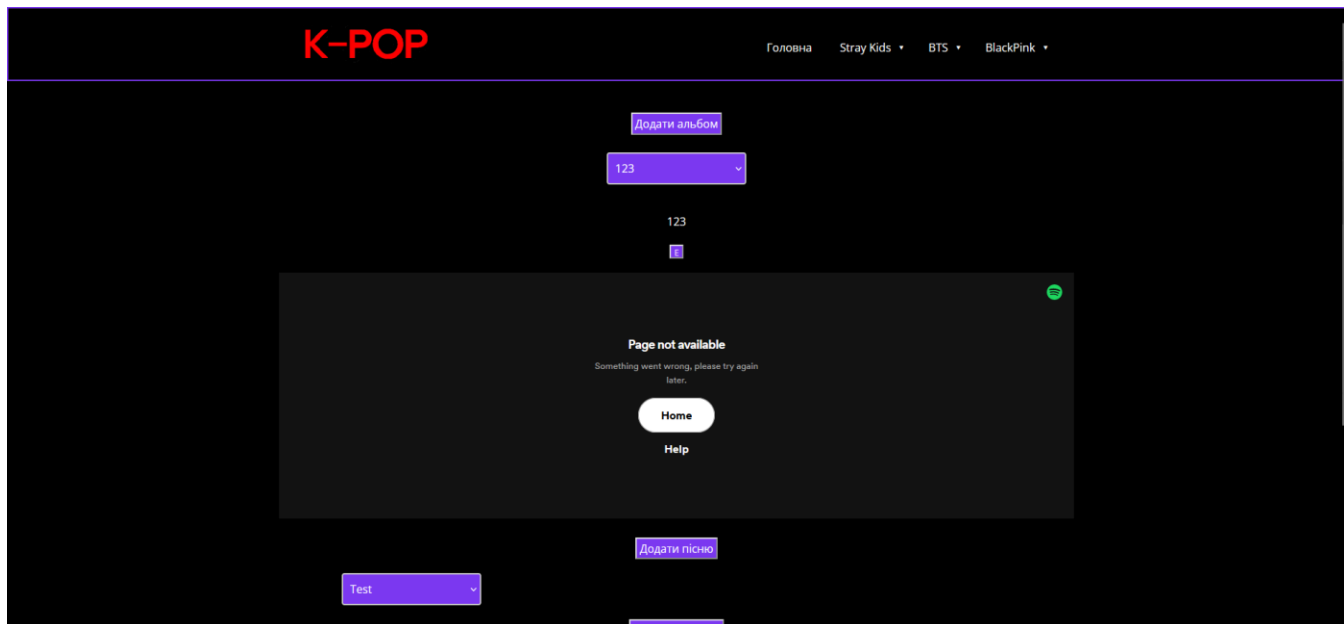


Рисунок 4.22 – Неправильно введений SpotifyID

*Джерело: зроблено автором (знімок з екрану)*

Але, якщо натиснути кнопки видалити пісню і це буде остання пісня альбому, то цей альбом перестане відображатися в списку. Для того, щоб повністю видалити цей альбом, потрібно звертатися до головного адміністратора і через phpMyAdmin він видалить його. На рисунку 4.23 зображено результат видалення пісні.

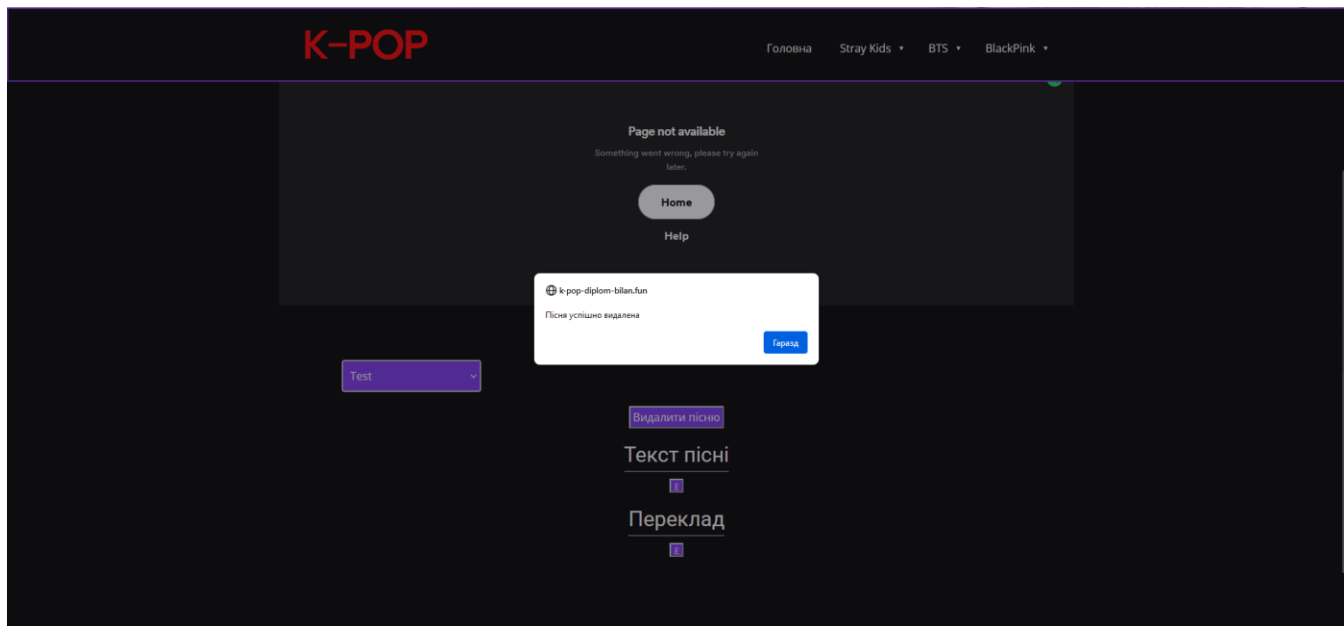


Рисунок 4.23 – Видалення пісні

*Джерело: зроблено автором (знімок з екрану)*

Після кожної дії з альбомами та піснями необхідність оновлення сторінок відсутня, всі дані динамічно змінюються.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра був проведений аналіз актуальності створення веб-орієнтованої системи музичних гуртів певного напрямку для їх просування, а також сформовано мету і задачі для проекту.

Було досягнуто мету проекту, а саме розроблено веб-орієнтовану систему музичних гуртів, яка дозволяє отримати інформацію про певні групи напрямку К-поп, про їх пісні, альбоми, нові відео на каналах, а також дозволить популяризувати даний напрям і в цілому ці гурти.

Проведено дослідження предметної області. Розглянуто аналоги розроблюваному продукту. Визначено функціональні вимоги.

У ході роботи було створено діаграми IDEF0, її декомпозицію, діаграму варіантів використання, схему бази даних, також проведено аналіз ризиків, також здійснено планування роботи.

Проведено аналіз існуючих технологій розробки веб-орієнтованої системи. Для реалізації було обрано гіпертекстову мову програмування HTML для розмітки веб-сторінок, каскадні таблиці стилів CSS для оформлення дизайну, мову JavaScript для анімації і взаємодії з серверною частиною, написаною на PHP, та СУБД MySQL для збереження даних системи.

Розроблена серверна частина, яка відповідає за обробку запитів до бази даних. Розміщено на хостингу та протестований функціонал.

Результатом кваліфікаційної роботи магістра є готова веб-орієнтована система музичних гуртів напрямку К-поп, що дозволить популяризувати дані гурти і напрям. У майбутньому система може бути вдосконалена додаванням більшої кількості гуртів, для їх популяризації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. К-поп — Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/К-поп>
2. Де і як створити свій сайт музикантові? Кращі конструктори сайтів | AREFYEV Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://arefyevstudio.com/uk/2019/08/16/de-i-yak-stvoriti-svij-sajt-muzikantovi-krashhi-konstruktori-sajtiv/>
3. Фрюгауф Т. Music in Social Life [Музика в соціальному житті]: [монографія]. Київ: Видавництво, 2021. - 300 с.
4. Halomon, S., & Wahyuni, D. (2011). Making Official Website of the Harmonic Orchestra Band Group Using Macromedia Dreamweaver 8 [Створення офіційного веб-сайту групи Harmonic Orchestra за допомогою Macromedia Dreamweaver 8]. Київ: Видавництво, 2011. - 150 с.
5. Lieb, K. (2013). Musicians and the Internet: Learning to Promote Themselves Without the Mainstream Media [Музиканти і Інтернет: Вивчення просування себе без загальних ЗМІ]. - Київ: Видавництво, 2013. - 180 с.
6. "Get More Fans: The DIY Guide to the New Music Business (2023 Edition)" [Текст]: [монографія]. - Київ: Видавництво, 2023. - 200 с.
7. Lukasik, P., & Kaszycka, I. (2013). The Use of the Internet by a Music Band: The Case Study of Mr. Irish Bastard [Використання Інтернету музичним гуртом: випадок дослідження Mr. Irish Bastard]. - Київ: Видавництво, 2013. - 250 с.
8. Chauhan, J., Shah, J., Mundhe, E., & Jain, I. (2021). Web Application for Machine Learning based Music Genre Classification [Веб-додаток для класифікації музичних жанрів на основі машинного навчання]. 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 1-6.

9. Hiranpanthaporn, S., Phoasavadi, P., & Cooharojananone, N. (2022). Angklung Smart Band: An IoT-Based System to Support Music Activities for Elderly in Thailand [Angklung Smart Band: система на основі Інтернету речей для підтримки музичних активностей для літніх людей в Таїланді]. 2022 IEEE European Technology and Engineering Management Summit (E-TEMS), 142-147.
10. Cooharojananone, N., Phoasavadi, P., Hiranpanthaporn, S., Wongratanapitak, P., & Atchariyachanvanich, K. (2022). A Study of Using the IoT-based Angklung Smart Band System to Support Music Group Activities for the Elderly in Nursing Homes [Дослідження використання системи Angklung Smart Band на основі Інтернету речей для підтримки музичних групових заходів для літніх у лікарнях для літніх].
11. Watanabe, M., Onoue, Y., Uemura, A., & Kitahara, T. (2021). Suiview: A Web-Based Application that Enables Users to Practice Wind Instrument Performance [Suiview: Веб-застосунок, що дозволяє користувачам вдосконалювати виконання на духових інструментах]. Computer Music Modeling and Retrieval. [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.scilit.net/publications/8aea9ba2e2ba6ba8d22ad8fe0a6d6216>
12. HansenRizkyK, S., & Senjaya, R. (2011). Web Applications Using the Band Bad Habit Joomla CMS [Веб-застосунки з використанням гурту Bad Habit Joomla CMS].
13. Hartadi, A., & IdaAyu, P. (2011). Making Padis Music Group Homepage Using Microsoft FrontPage XP [Створення домашньої сторінки музичної групи Padis за допомогою Microsoft FrontPage XP].
14. Quauari, H.G. (2012). Making Drum Corps Indonesia Web Using Adobe Dreamweaver, PHP and MySQL [Створення веб-сайту Drum Corps Indonesia за допомогою Adobe Dreamweaver, PHP та MySQL].

15. Корейська поп-музика - як вона захопила світ — Суспільне Культура [Електронний ресурс] – Режим доступу до ресурсу: <https://suspilne.media/culture/17899-cim-osobliva-korejska-pop-muzika-i-ak-vona-zahopila-svit/>.
16. K-POP: причини популярності | Times ZT [Електронний ресурс] – Режим доступу до ресурсу: <https://times.zt.ua/k-pop-prychyny-populiarnosti/>.
17. У кого зараз закохані всі українські школярки? — журнал | «Освіторія» [Електронний ресурс] – Режим доступу до ресурсу: <https://osvitoria.media/experience/k-pop-korejska-nyashnist-shho-zavojovuye-svit/>.
18. Веб-сайт: визначення й застосування. Статті про розробку та просування веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site>.
19. SKZ ARCHIVE [Електронний ресурс] – Режим доступу до ресурсу: <https://skzarchive.godaddysites.com/>.
20. US BTS ARMY [Електронний ресурс] – Режим доступу до ресурсу: <https://www.usbtsarmy.com/>.
21. Що таке бек-енд веб-сайту? Глосарій конструктора веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.websiterating.com/uk/website-builders/glossary/what-is-back-end/>.
22. Методи і техніка досліджень [Електронний ресурс] – Режим доступу до ресурсу: [https://elib.tsatu.edu.ua/dep/mtf/ophv\\_10/page3.html](https://elib.tsatu.edu.ua/dep/mtf/ophv_10/page3.html).
23. Інструкція, як будувати UML-діаграми | DOU [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/forums/topic/40575/>.
24. Основні переваги субд MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5607354/page:3/>.
25. Нотація IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/%D0%9A%D0%BE%D0%BD%D](https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D)

0%B4%D1%96%D1%83%D1%81%202%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%B2%D0%B0/page9.html.

26. Діаграма прецедентів - Wikiwand [Електронний ресурс] – Режим доступу до ресурсу: [https://www.wikiwand.com/uk/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0\\_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2](https://www.wikiwand.com/uk/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D1%96%D0%B2).

27. Клієнт-серверна архітектура | Онлайн-курси від компанії QATestLab [Електронний ресурс] – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>.

28. HTTP Методи GET та POST. Довідник. Уроки для початківців. W3Schools українською [Електронний ресурс] – Режим доступу до ресурсу: [https://w3schoolsua.github.io/tags/ref\\_httpmethods.html#gsc.tab=0](https://w3schoolsua.github.io/tags/ref_httpmethods.html#gsc.tab=0).

29. HTML — Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>.

30. Що таке CSS [Електронний ресурс] – Режим доступу до ресурсу: [https://css.in.ua/article/shcho-take-html\\_10](https://css.in.ua/article/shcho-take-html_10).

31. Вступ до JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.javascript.info/intro>.

32. Php що це? Чим PHP краще за інші мови програмування? — Хостинг-компанія FREEhost.UA [Електронний ресурс] – Режим доступу до ресурсу: <https://freehost.com.ua/ukr/faq/wiki/что-такое-php/>.

33. База даних MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://promoter.net.ua/articles/baza-danix-mysql.html>.

34. media CSS значення [Електронний ресурс] – Режим доступу до ресурсу: <https://css.in.ua/css/value/media>.

35. YouTube API | Інтеграція та Автоматизація за 5 хвилин [Електронний ресурс] – Режим доступу до ресурсу: <https://apix-drive.com/ua/youtube-api>.
36. Genius | Song Lyrics & Knowledge [Електронний ресурс] – Режим доступу до ресурсу: <https://genius.com/>.
37. Spotify — Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Spotify>.
38. APServer - MySQL :: Зовнішні ключі [Електронний ресурс] – Режим доступу до ресурсу: <http://apserver.org.ua/peregl.php?d=view&tid=91>.
39. Fetch [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.javascript.info/fetch>.
40. Що таке JSON | Навіщо потрібен цей формат [Електронний ресурс] – Режим доступу до ресурсу: <https://apix-drive.com/ua/blog/useful/scho-take-json>.



## ДОДАТОК А

### Планування робіт

Попит на веб-сайти для популяризації певних музичних напрямів зростає. Гуртів з кожним роком все більше, і не всім доволі легко стати популярними, тому було вирішено створити веб-сайт для популяризації їх. Як напрямок було обрано К-поп.

**Деталізація мети методом SMART.** Постановка однозначної та зрозумілої мети на етапі концептуального проектування – це ключ до успіху всього проекту. Результати деталізації мети проекту методом SMART представлено у таблиці А.1

Таблиця А.1 – Деталізація мети проекту методом SMART

Specific (Конкретна)	Створити web-орієнтовану систему гуртів напряму К-поп, котра матиме в собі інформацію про групи, їх альбоми, учасників, посилання на соцмережі, а також відео на youtube.
Measurable (Вимірювана)	Результатом роботи проекту є розроблена web-орієнтована система гуртів напряму К-поп.
Achievable (Досяжна)	Для виконання проекту наявні необхідні знання HTML, CSS, мови програмування JavaScript, PHP, баз даних MySQL та навичок написання документації. Враховуючи доступні ресурсні можливості та обмеження мета є такою, яку можливо досягти.
Relevant (Реалістична)	Розроблена web-орієнтована система дозволить проінформувати користувачів про гурти, їх альбоми, пісні, переклад, а також модифікувати деякі дані без знань мов програмування, якщо авторизований.

## Продовження таблиці А.1

Time-framed (Обмежена у часі)	Ціль має часове обмеження. Термін досягнення мети проекту визначено за домовленістю між замовником та виконавцем і складає два місяці.
-------------------------------------	--

*Джерело: побудовано автором*

Планування змісту робіт. WBS (Work Breakdown Structure) – це графічна структура елементів проекту, що ієрархічно розташовані та поєднані з продуктом проекту. Метою розробки ієрархічної моделі робіт є організація командної роботи, контроль та оцінка термінів роботи. На найвищому рівні знаходиться продукт проекту. Заходи для забезпечення досягнення мети розташовані на другому рівні. Подальші рівні складають елементи дрібної декомпозиції. Декомпозиція задач виконується до тих пір, поки вид робіт не стане елементарно простим. Елементарними роботами називають дії, виконання яких спрямовано на однозначний та чіткий результат, призначена відповідальна особа, для якої можна обчислити терміни виконання та витрати праці. На рисунку А.1 представлено WBS на розробку інформаційної системи оцінки професійних досягнень співробітників ІТ-компанії.

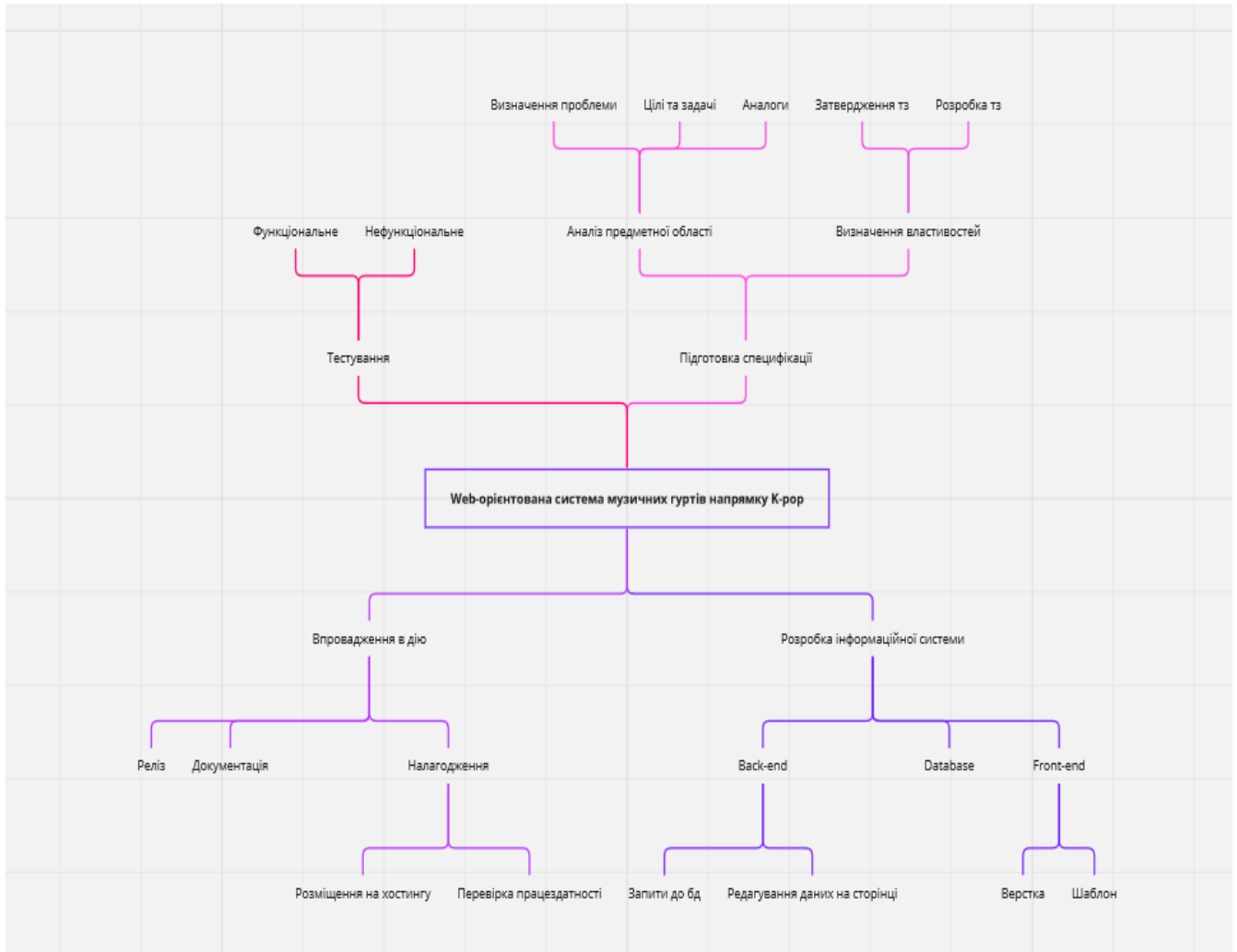


Рисунок А.1 – WBS-структура робіт проекту

*Джерело: створено автором (знімок з екрану)*

Планування структури виконавців. Розробка організаційної структури виконавців або OBS є наступним етапом після декомпозиції процесів. OBS (Organizational Breakdown Structure) – графічний вигляд структури відповідальних осіб, що беруть участь у розробці проекту та призначені на виконання елементарних робіт. На рисунку А.2 представлено організаційну структуру планування робіт проекту. Учасників проекту описано у таблиці А.2.

Таблиця А.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Білан В.О.	Виконує frond-end та back-end розробку
Проектувальник	Білан В.О.	Виконує проектування бази даних та розробляє структуру інформаційної системи
Тестувальник	Білан В.О.	Проводить тестування інформаційної системи
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту
Менеджер проекту	Білан В.О.	Відповідає за дотримання термінів, розподіл ресурсів та завдань між учасниками проекту. Виконує збір та аналіз даних.

*Джерело: побудовано автором*

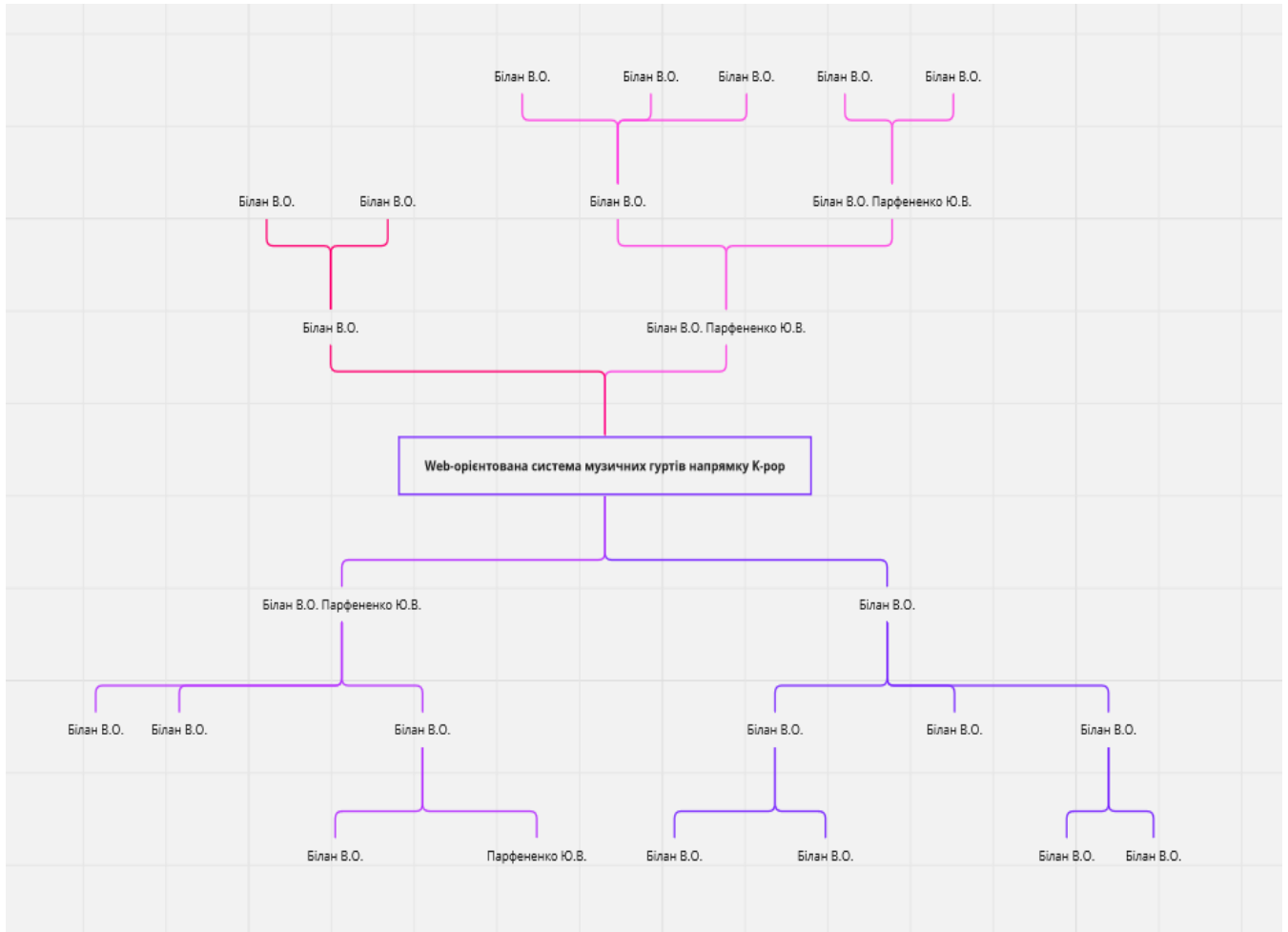


Рисунок А.2 – OBS-структура робіт проекту  
*Джерело: створено автором (знімок з екрану)*

**Діаграма Ганта.** Побудова календарного плану проекту є важливим етапом роботи над проектом та виглядає як графік виконання робіт з реальним розподілом дат. Даний календарний план надає можливість мати достовірне уявлення про тривалість виконання етапів проекту, враховуючи обмеження у ресурсах.

Календарний графік проекту представлено на рисунку А.3.

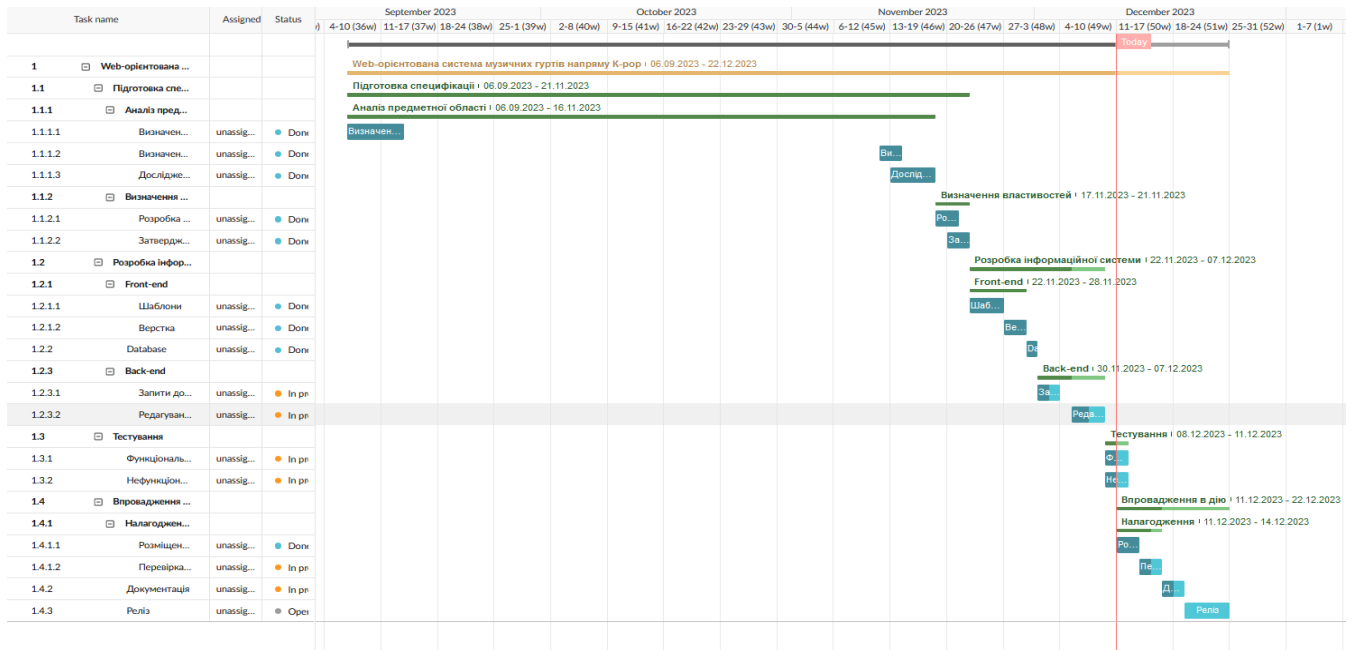


Рисунок А.3 – Календарний графік виконання робіт

*Джерело: створено автором (знімок з екрану)*

Управління ризиками проекту. Наступним етапом планування робіт проекту є оцінка та робота над ризиками: кількісне та якісне оцінювання та заходи для усунення або зменшення впливу ризиків на проект. У таблиці А.3 представлено шкалу оцінки ризиків за класифікацією, величиною впливу та ймовірністю виникнення.

Таблиця А.3 – Шкала оцінювання ризиків за ймовірністю виникнення на величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

*Джерело: побудовано автором*

Для того, щоб знизити негативний вплив ризиків на проект треба виконати планування реагування на них. До нього входить визначення ефективності розробки та оцінка наслідків впливу на проект. Оцінювання виконується за показниками, що описані в таблиці А.3. У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку А.4. Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

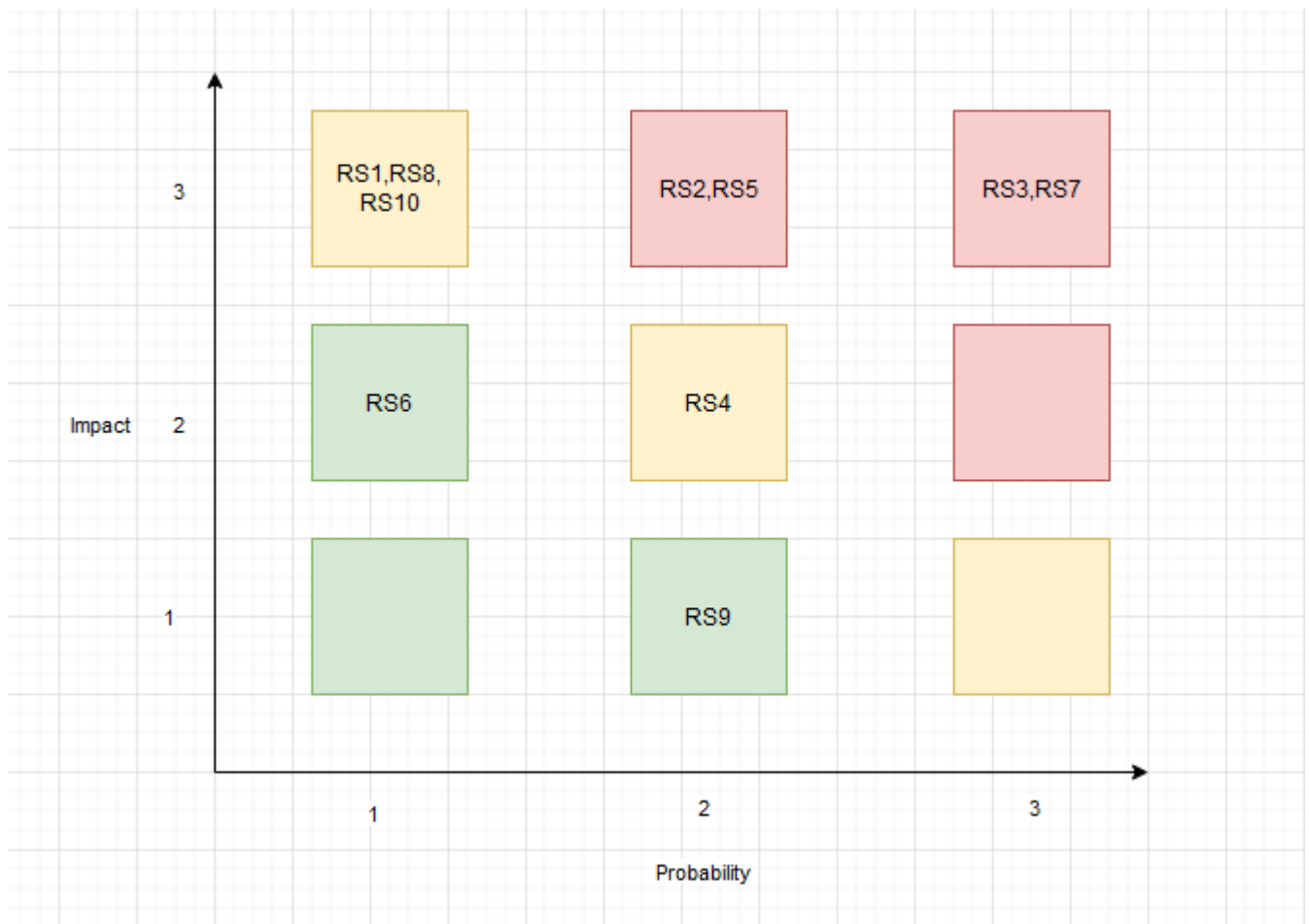


Рисунок А.4. – Матриця ймовірності

*Джерело: створено автором (знімок з екрану)*

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці А.4. У таблиці А.5 описано ризики та стратегії реагування на кожен з них.

Таблиця А.4 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики
1	Прийнятні	$1 \leq R \leq 2$	6,9
2	Виправдані	$3 \leq R \leq 4$	1,4,8,10
3	Недопустимі	$6 \leq R \leq 9$	2,3,5,7

*Джерело: побудовано автором*



Таблиця А.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Поява продукту аналогічного розробленого	Низька	Високий	3	1.Аналіз всіх аналогів, їх плюсів та мінусів.	Прийняття	
RS_2	Відкритий	Неповністю сформульоване завдання	Середня	Високий	6	Періодичне обговорення результатів керівником. 3	Попередження	При виявленні невідповідностей деяких характеристик продукту заявленим вимогам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.

Продовження таблиці А.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_3	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Визначити важливість того чи іншого етапу розробки. Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Дотримуватися календарного плану.	Пом'якшення	Змінити порядок виконання роботи, перерозподілити час на найбільш важливі завдання.
RS_4	Відкритий	Часте внесення змін у ТЗ	Середня	Середній	4	1.Виділити всі необхідні параметри проекту 2.Чітко описати вимоги до проекту.	Перенесення	Узгодити все з замовником, всі правки.

Продовження таблиці А.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_5	Відкритий	Вибір технології розробки, котра не задовольняє вимоги	Середня	Високий	6	1.Проаналізувати методи та засоби, для виконання проекту. 2.Обрати зрозумілу та легку в використанні технологію розробки.	Пом'якшення	Виділити час та ресурси на пошуки покращення обраної технології. Застосувати допоміжні ресурси.
RS_6	Відкритий	Неправильна оцінка в масштабі проекту	Низька	Середній	2	1.Провести детальний аналіз проекту. 2.Визначити основні етапи проекту, розподілити час на їх виконання.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.
RS_7	Відкритий	Помилки проектування	Висока	Високий	9	При проектуванні обговорювати всі моменти з замовником, а потім демонструвати результати.	Пом'якшення	Контролювати проміжні результати.

Продовження таблиці А.5

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_8	Відкритий	Збої в роботі програмного забезпечення	Низька	Високий	3	Підготувати план дій, для усунення збоїв	Попередження	Замінити програмне забезпечення.
RS_9	Відкритий	Невиконання моніторингу проекту	Середня	Низький	2	Здійснювати проміжний контроль результатів в ході виконання проекту. Здійснювати моніторинг проекту.	Перенесення	Контроль результатів проекту керівником і звітування перед ним на кожному з етапів.

## Продовження таблиці А.5

RS_10	Відкритий	Зміна вимог в процесі розробки проекту	Низька	Високий	3	Узгодити всі питання на початкових етапах, щоб мінімізувати кількість змін під час розробки.	Пом'якшення	Переоцінка проекту, кожного разу, коли вимоги змінюються
-------	-----------	--	--------	---------	---	--	-------------	--

*Джерело: побудовано автором*

## ДОДАТОК Б

### Скрипт файлу Альбоми та пісні.html

```

document.addEventListener('DOMContentLoaded', function () {
  const editButtons = document.querySelectorAll('.edit-button');
  const addSongButton = document.querySelector('.add-song-button');
  const addAlbumButton = document.querySelector('.add-album-button');
  const deleteSongButton = document.querySelector('.delete-song-button');

  function checkAuthorizationAndScreenSize() {
    // Перевірка авторизації
    const cookies = document.cookie.split(';').reduce((cookiesObject,
cookie) => {
      const [name, value] = cookie.split('=').map(c => c.trim());
      cookiesObject[name] = value;
      return cookiesObject;
    }, {});

    const isAuthorized = cookies['authorized'];
    // Перевірка розміру екрану
    const windowWidth = window.innerWidth;
    const maxWidthForButtons = 992;

    editButtons.forEach(button => {
      // Зміни відображення кнопок в залежності від авторизації та
розміру екрану
      if (isAuthorized === 'true' && windowWidth > maxWidthForButtons)
      {
        addSongButton.style.display = 'block';
        button.style.display = 'block';
        addAlbumButton.style.display = 'block';
        deleteSongButton.style.display = 'block';
      } else {
        button.style.display = 'none';
        addSongButton.style.display = 'none';
        addAlbumButton.style.display = 'none';
        deleteSongButton.style.display = 'none';
      }
    });
  }

  // Викликати функцію для перевірки авторизації та розміру екрану при
завантаженні сторінки
  checkAuthorizationAndScreenSize();

  // Викликати функцію при зміні розміру вікна
  window.addEventListener('resize', checkAuthorizationAndScreenSize);
  const urlParams = new URLSearchParams(window.location.search);

```

```

const groupParam = urlParams.get('group');

if (!groupParam || (groupParam !== "StrayKids" && groupParam !== "BTS" &&
groupParam !== "BlackPink")) {
    alert('Параметр group відсутній у URL. Будь ласка, вкажіть необхідний
параметр.');
```

window.location.href = 'index.html'; // Перенаправлення на сторінку з помилкою

```

}

let albumsData = {};

// Функція для отримання та оновлення даних у масиві albumsData
function updateAlbumsData(groupName, shouldSetSelected = false) {
    fetch('load.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify({ groupParam: groupName })
    })
    .then(response => response.json())
    .then(updatedData => {
        let currentSelectedValue;
        // Оновлення albumsData з отриманими даними з сервера для певної
групи
        if (shouldSetSelected) {
            currentSelectedValue = albumList.value;
        }
        albumsData = updatedData;
        albumList.innerHTML = '';
        initializeAlbumList();

        if (shouldSetSelected && currentSelectedValue) {
            const options = albumList.options;
            for (let i = 0; i < options.length; i++) {
                if (options[i].value === currentSelectedValue) {
                    albumList.selectedIndex = i;
                    break;
                }
            }
        }
    })
    .catch(error => {
        console.error('Помилка отримання даних з сервера:', error);
    });
}

// Виклик функції для отримання даних для групи та оновлення albumsData
updateAlbumsData(groupParam, false);
const albumList = document.getElementById('album-list');
const songList = document.getElementById('song-list');
```

```

const albumDescription = document.querySelector('.album-description-text');
const songText = document.querySelector('.song-text');
const translationText = document.querySelector('.translation-text');
const editorContainer = document.querySelector('.editor-container');
const editorTextarea = document.querySelector('.editor-textarea');
const updateButton = document.querySelector('.update-button');
const cancelButton = document.querySelector('.cancel-button');
const spotifyFrame = document.getElementById('spotify-frame');

// Функція для оновлення тексту блоку
function updateTextContent(blockType, newText) {
  const block = document.querySelector(`.${blockType}-text`);
  block.innerHTML = newText.replace(/\n/g, '<br>');
}

// Функція для конвертації <br> на \n
function convertToNewlines(text) {
  return text.replace(/<br\s*\/?>/g, '\n');
}

// Функція для конвертації \n на <br>
function convertToBreaks(text) {
  return text.replace(/\n/g, '<br>');
}

let lastSelectedType = null; // Змінна для зберігання останнього вибору

function editContent(blockType) {
  let blockText;
  if (blockType === 'album') {
    const block = document.querySelector('.album-description-text');
    blockText = block.innerHTML;
    editorTextarea.value = convertToNewlines(blockText);
  } else {
    const block = document.querySelector(`.${blockType}-text`);
    blockText = block.innerHTML;
    editorTextarea.value = convertToNewlines(blockText);
  }
  editorContainer.style.display = 'block';
  cancelButton.addEventListener('click', function () {
    editorTextarea.value = ''; // Очищення поля редагування тексту
    editorContainer.style.display = 'none';
  });
  return blockText;
}

// Функція для обробки кліку на кнопку редагування
function handleEditButtonClick(event) {
  const blockType = event.target.getAttribute('data-type');
  lastSelectedType = blockType; // Зберегти останній вибір
  editContent(blockType);
}

```



```

// Додати обробник події для кнопки оновлення тільки один раз
updateButton.removeEventListener('click', handleUpdateButtonClick);
updateButton.addEventListener('click', handleUpdateButtonClick);
}

// Функція для оновлення списку альбомів
function updateAlbumList(albumsData) {
  albumList.innerHTML = '';
  for (const albumKey in albumsData) {
    if (albumsData.hasOwnProperty(albumKey)) {
      const option = document.createElement('option');
      option.value = albumKey;
      option.textContent = albumsData[albumKey].title;
      albumList.appendChild(option);
    }
  }
}

// Функція для обробки кліку на кнопку оновлення
function handleUpdateButtonClick() {
  const editedText = editorTextarea.value; // Отримати змінений текст після
  редагування

  if (lastSelectedType === 'album') {
    // Логіка оновлення опису альбому
    const selectedAlbumName = albumList.value;
    fetch('update_album_description.php', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ editedDescription: editedText, albumName:
selectedAlbumName })
    })
    .then(response => {
      if (response.ok) {
        const albumDescriptionBlock = document.querySelector('.album-
description-text');
        albumDescriptionBlock.innerHTML =
convertToBreaks(editedText);
        hideEditor();
        // Оновлення списку альбомів після отримання нових даних
        updateAlbumsData(groupParam, true); // Виклик функції для
отримання оновлених даних
      } else {
        throw new Error('Network response was not ok.');
```

```

// Логіка оновлення тексту пісні чи перекладу
const selectedAlbumName = albumList.value;
const selectedSongId = songList.value;
fetch('update.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({ dataType: lastSelectedType, editedText:
editedText, songId: selectedSongId })
})
.then(response => {
  if (response.ok) {
    updateTextContent(lastSelectedType, editedText);
    hideEditor();
    updateAlbumsData(groupParam, true);
  } else {
    throw new Error('Network response was not ok.');
```

```

    songText.innerHTML = songData.lyrics.replace(/\n/g, '<br>');
    translationText.innerHTML = songData.translation.replace(/\n/g, '<br>');
}

// Функція для ініціалізації пісень для обраного альбому
function initializeSongs(selectedAlbum) {
    songList.innerHTML = '';
    selectedAlbum.songs.forEach(song => {
        const option = document.createElement('option');
        option.value = song.id;
        option.textContent = song.title;
        songList.appendChild(option);
    });
    songList.addEventListener('change', function () {
        const selectedSong = selectedAlbum.songs.find(song => song.id ===
songList.value);
        updateSongText(selectedSong);
    });
}

// Обробник події при зміні обраного альбому
albumList.addEventListener('change', function () {
    const selectedAlbumKey = this.value;
    const selectedAlbum = albumsData[selectedAlbumKey];
    albumDescription.textContent = selectedAlbum.description;
    spotifyFrame.innerHTML = `<iframe
src="https://open.spotify.com/embed/album/${selectedAlbum.spotifyId}"
width="100%" height="352" frameborder="0" allowfullscreen="" allow="autoplay;
clipboard-write; encrypted-media; fullscreen; picture-in-picture"
loading="lazy"></iframe>`;
    initializeSongs(selectedAlbum);
    const firstSong = selectedAlbum.songs[0]; // Отримання першої пісні для
обраного альбому
    updateSongText(firstSong);
});

// Логіка для зміни видимості тексту пісні та перекладу
function toggleTextVisibility(element) {
    element.classList.toggle('active');
}

const songTextHeader = document.querySelector('.song-text-header');
songTextHeader.addEventListener('click', function () {
    const songText = document.querySelector('.song-text');
    toggleTextVisibility(songText);
});

const translationHeader = document.querySelector('.translation-header');
translationHeader.addEventListener('click', function () {
    const translation = document.querySelector('.translation-text');
    toggleTextVisibility(translation);
});

```

```

function initializeSongsForFirstAlbums() {
  for (const albumKey in albumsData) {
    if (albumsData.hasOwnProperty(albumKey)) {
      const firstSong = albumsData[albumKey].songs[0];
      updateSongText(firstSong);
      break; // Зупиняємо цикл після обробки першого альбому
    }
  }
}

// Оновлення тексту перших пісень
function updateFirstSongs() {
  for (const albumKey in albumsData) {
    if (albumsData.hasOwnProperty(albumKey)) {
      const firstSong = albumsData[albumKey].songs[0];
      updateSongText(firstSong);
    }
  }
}

// Отримання посилання на кнопку видалення пісні
const deleteSongButton = document.querySelector('.delete-song-button');

// Обробник події для видалення пісні
deleteSongButton.addEventListener('click', function () {
  const selectedSongId = songList.value; // Отримання ID обраної пісні
  if (selectedSongId) {
    fetch('delete_song.php', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({ songId: selectedSongId })
    })
    .then(response => {
      if (response.ok) {
        // Оновлення списку пісень після видалення
        updateAlbumsData(groupParam, true);
        alert('Пісня успішно видалена');
      } else {
        throw new Error('Помилка видалення пісні');
      }
    })
    .catch(error => {
      console.error('Помилка:', error);
    });
  } else {
    alert('Будь ласка, оберіть пісню для видалення');
  }
});
});

```

## Скрипт сторінки Addalbum.html

```

document.addEventListener('DOMContentLoaded', function () {
  const groupSelect = document.getElementById('group-select');

  fetch('get_groups.php')
    .then(response => response.json())
    .then(groups => {
      groups.forEach(group => {
        const option = document.createElement('option');
        option.value = group.id_group;
        option.textContent = group.name_group;
        groupSelect.appendChild(option);
      });
    })
    .catch(error => {
      console.error('Помилка отримання списку груп:', error);
    });

  const addAlbumForm = document.getElementById('add-album-form');

  addAlbumForm.addEventListener('submit', function (event) {
    event.preventDefault();

    const groupSelect = document.getElementById('group-select');
    const selectedGroupId = groupSelect.value;
    const albumName = document.getElementById('album-name').value;
    const albumDescription = document.getElementById('album-
description').value;
    const spotifyId = document.getElementById('spotify-id').value;

    fetch('add_album.php', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        groupId: selectedGroupId,
        albumName: albumName,
        albumDescription: albumDescription,
        spotifyId: spotifyId
      })
    })
    .then(response => response.text())
    .then(result => {
      addAlbumForm.reset();
      console.log(result);
    })
    .catch(error => {
      console.error('Помилка додавання альбому:', error);
    });
  });
});

```

```
});
```

## Скрипт сторінки Addsong.html

```
function fetchGroups() {
  fetch('load_groups.php')
    .then(response => response.json())
    .then(groups => {
      const groupSelect = document.getElementById('group-select');
      groups.forEach(group => {
        const option = document.createElement('option');
        option.value = group.id_group;
        option.textContent = group.name_group;
        groupSelect.appendChild(option);
      });
      groupSelect.addEventListener('change', loadAlbums);
      loadAlbums();
    })
    .catch(error => {
      console.error('Помилка отримання списку груп:', error);
    });
}

function loadAlbums() {
  const groupSelect = document.getElementById('group-select');
  const selectedGroupId = groupSelect.value;

  fetch(`load_albums.php?group=${selectedGroupId}`)
    .then(response => response.json())
    .then(albums => {
      const albumSelect = document.getElementById('album-select');
      albumSelect.innerHTML = '';

      albums.forEach(album => {
        const option = document.createElement('option');
        option.value = album.id_album;
        option.textContent = album.name_album;
        albumSelect.appendChild(option);
      });
    })
    .catch(error => {
      console.error('Помилка отримання списку альбомів:', error);
    });
}

function addSongToAlbum() {
  const albumSelect = document.getElementById('album-select');
  const selectedAlbumId = albumSelect.value;
  const songTitle = document.getElementById('song-title').value;
  const songLyrics = document.getElementById('song-lyrics').value;
  const songTranslation = document.getElementById('song-translation').value;
```

```
const groupSelect = document.getElementById('group-select');
const selectedGroupId = groupSelect.value;

fetch('add_song.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    albumId: selectedAlbumId,
    title: songTitle,
    lyrics: songLyrics,
    translation: songTranslation,
    id_group: selectedGroupId
  })
})
.then(response => {
  if (response.ok) {
    console.log('Пісня успішно додана до альбому!');
    document.getElementById('song-title').value = '';
    document.getElementById('song-lyrics').value = '';
    document.getElementById('song-translation').value = '';
  } else {
    throw new Error('Помилка при додаванні пісні до альбому.');
  }
})
.catch(error => {
  console.error('Помилка:', error);
});
}

document.addEventListener('DOMContentLoaded', function () {
  fetchGroups();

  const addSongButton = document.getElementById('add-song-button');
  addSongButton.addEventListener('click', addSongToAlbum);
});
```