

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота бакалавра

на тему: «Розробка Web-додатку для розповсюдження та зберігання цифрової
звукової інформації»

Виконав: студент групи ПЗ319-2

Спеціальність 121 «Інженерія програмного
забезпечення»

Шутка М. О.

(прізвище та ініціали)

Керівник: к. ф.-м. н., доц. Лебідь О. Ю.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент: Спеціалізоване управління

розробки та супроводження програмного

забезпечення Департаменту з питань

цифрового розвитку, цифрових

трансформацій та цифровізації ДМСУ

(місце роботи)

головний державний інспектор відділу

розробки програмного забезпечення

(посада)

Бахтін О. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

АНОТАЦІЯ

Шутка М. О. Розробка Web-додатку для розповсюдження та зберігання цифрової звукової інформації.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2023.

У роботі розглядається процес реєстрації та входу в особистий аккаунт, де користувачі матимуть можливість створити свій профіль без необхідності підтвердження пошти. Система також надає можливість змінювати особисті дані, такі як нікнейм, локація та справжнє ім'я, а також завантажувати та змінювати аватар.

Одним з ключових функціональних елементів сайту є можливість пошуку по глобальним композиціям. Користувачі зможуть знайти потрібні треки за допомогою різних критеріїв, таких як назва виконавця або назва пісні.

Система також пропонує функцію формування плейлисту. Користувачі зможуть створювати власні плейлисти шляхом натискання кнопки "лайк" або додавати пісні до вже існуючих списків відтворення.

Окрім того, web-застосунок дозволяє прослуховувати музику в режимі онлайн. Користувачі зможуть стрімити пісні безпосередньо з сайту, не завантажуючи їх на свій пристрій.

Результатом роботи є функціональний web-застосунок, який забезпечує зручний доступ до цифрової звукової інформації.

Ключові слова: WEB-ЗАСТОСУНОК, ЦИФРОВА ЗВУКОВА ІНФОРМАЦІЯ, РОЗПОВСЮДЖЕННЯ, ЗБЕРІГАННЯ, РЕЄСТРАЦІЯ, ПОШУК, ПЛЕЙЛИСТ, ОНЛАЙН ПРОСЛУХОВУВАННЯ.

ABSTRACT

Shutka M.O. Development of a web application for distribution and storage of digital sound information.

Qualification work for obtaining a bachelor's degree in specialty 121 "Software engineering". – University of Customs and Finance, Dnipro, 2023.

The work examines the process of registration and logging into a personal account, where users will have the opportunity to create their profile without the need for email confirmation. The system also provides the ability to change personal data, such as nickname, location and real name, as well as upload and change an avatar.

One of the key functional elements of the site is the ability to search for global compositions. Users will be able to find the desired tracks using various criteria such as artist name or song title.

The system also offers the function of creating a playlist. Users will be able to create their own playlists by clicking the "like" button or add songs to existing playlists.

In addition, the web application allows you to listen to music online. Users will be able to stream songs directly from the site without downloading them to their device.

The result of the work is a functional web application that provides convenient access to digital sound information.

Keywords: WEB APPLICATION, DIGITAL SOUND INFORMATION, DISTRIBUTION, STORAGE, REGISTRATION, SEARCH, PLAYLIST, ONLINE LISTENING.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ	9
1.1 Аналіз предметної області	9
1.2 Аналіз існуючих рішень	11
1.2.1 Apple Music	12
1.2.2 YouTube Music	14
1.3 Постанова задачі	16
1.4 Висновки до першого розділу	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	18
2.1 Розробка бази даних веб-сайту	18
2.1.1 Побудова концептуальної моделі	18
2.1.2 Побудова логічної моделі	20
2.1.3 Розробка фізичної моделі бази даних	21
2.2 Розробка макету веб-сайту	23
2.3 Висновки до другого розділу	28
РОЗДІЛ 3. РЕЗУЛЬТАТИ РОЗРОБКИ WEB-ДОДАТКУ ДЛЯ РОЗПОВСЮДЖЕННЯ ТА ЗБЕРІГАННЯ ЦИФРОВОЇ ЗВУКОВОЇ ІНФОРМАЦІЇ	29
3.1 Технічне завдання та етапи розробки	29
3.2 Інтерфейс користувача	32
3.3 Інструкція користувача	37
3.4 Висновки до третього розділу	47
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК	54

ВСТУП

З поширенням Інтернету та швидкого розвитку цифрових технологій, розповсюдження та зберігання цифрової звукової інформації стають надзвичайно важливими аспектами сучасного суспільства.

Відкритість та доступність музичного контенту на веб-платформах стали визнаними нормами, що змінюють спосіб, яким користувачі сприймають та споживають музику. З цим зв'язана потреба в розробці функціональних web-застосунків, які забезпечують зручний доступ до цифрової звукової інформації та надають користувачам широкі можливості пошуку, прослуховування та зберігання музичних композицій.

Об'єктом дослідження є технології розробка Web-додатків.

Предмет дослідження: розробка Web-додатку для розповсюдження та зберігання цифрової звукової інформації.

Метою даного дослідження є розробка web-застосунку, який надасть користувачам зручний спосіб знаходження, прослуховування та зберігання цифрової звукової інформації.

Проект спрямований на створення функціонального сайту, який задовольнятиме потреби користувачів у пошуку та відтворенні музичних композицій, а також забезпечить можливість створення та управління плейлистами.

Завдання кваліфікаційної роботи:

1. Проаналізувати предметну область.
2. Провести аналіз існуючих рішень в сфері збереження та відтворення музичного контенту.
3. Сформулювати постановку завдання.
4. Розробити базу даних для роботи веб-додатку.
5. Побудувати концептуальну модель бази даних.

6. Побудувати логічну модель бази даних.
7. Розробити фізичну модель бази даних.
8. Розробити макет веб-сайту.
9. Сформулювати технічне завдання та етапи розробки.
10. Розробити інтерфейс користувача.
11. Створити інструкцію користувача.
12. Протестувати створений програмний продукт та надати можливі сценарії його використання.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Розробити систему реєстрації та входу в особистий акаунт, яка надасть користувачам можливість створення профілю та змінювати особисті дані. Детально розглянути процеси аутентифікації та авторизації для забезпечення безпеки і конфіденційності користувачів.

2. Розробити функціонал для пошуку музичних композицій.

3. Розробити можливість формування плейлистів, що дозволить користувачам створювати власні списки відтворення та додавати пісні до існуючих плейлистів. Забезпечити зручну навігацію по плейлистам та можливість редагування, видалення та переміщення пісень у списку.

4. Розробити функцію онлайн прослуховування музики, що дозволить користувачам стрімити пісні безпосередньо з сайту, не завантажуючи їх на свій пристрій. Забезпечити оптимальну якість звуку та мінімальні затримки при програванні.

5. Забезпечити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що сприятиме зручній навігації та використанню функціоналу web-застосунку.

6. Розглянути принципи дизайну і взаємодії, що сприятимуть покращенню користувацького досвіду.

Для досягнення мети дослідження будуть використані такі методи:

1. Аналіз літературних джерел та веб-ресурсів, що стосуються розробки web-застосунків та технологій зберігання та розповсюдження цифрової звукової інформації. Вивчення сучасних тенденцій, стандартів та кращих практик у сфері розробки музичних веб-платформ.

2. Вивчення сучасних практик реалізації функціональності, подібної до тієї, що запропонована в досліджуваному web-застосунку. Аналіз існуючих рішень та їхніх переваг і недоліків.

3. Розробка програмного забезпечення на базі веб-технологій та інструментів, що відповідають вимогам проекту. Використання сучасних фреймворків та бібліотек для забезпечення швидкої та ефективної розробки.

4. Тестування та оцінка розробленого web-застосунку залежно від функціональності, продуктивності та користувацької зручності. Використання автоматизованих тестів, проведення тестування залежності веб-застосунку від різних веб-браузерів та платформ.

Розширений обсяг дозволить детально описати кожен етап розробки, методи дослідження та тестування, а також провести аналіз отриманих результатів.

Додаткові розділи будуть присвячені дизайну і взаємодії з користувачем, аналізу сучасних трендів у сфері музичних веб-платформ та можливостям подальшого розширення функціоналу.

Методи дослідження: спостереження, порівняння, рахунок, вимірювання, експеримент, узагальнення, абстрагування, формалізація, аналіз і синтез, індукція і дедукція, аналогія.

Робота підтверджує наступні програмні результати навчання, що відповідають освітній програмі 121 «Інженерія програмного забезпечення»:

– ПР01 – аналіз, цілеспрямований пошук і вибір необхідних для вирішення професійних завдань інформаційно-довідникових ресурсів і знань з урахуванням сучасних досягнень науки і техніки;

- ПР03 – знання основних процесів, фаз та ітерацій життєвого циклу програмного забезпечення;
- ПР06 – уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- ПР08 – вміння розробляти людино-машинний інтерфейс;
- ПР11 – вибір вихідних даних для проектування, керуючись формальними методами опису вимог та моделювання;
- ПР18 – знання та вміння застосовувати інформаційні технології обробки, зберігання та передачі даних;
- ПР23 – вміння документувати та презентувати результати розробки програмного забезпечення.

Структура кваліфікаційної роботи бакалавра: вступ, три розділи, висновки, список використаних джерел, що містить 40 посилань та додатку. Робота містить 42 рисунки, 1 таблицю, обсяг – 50 сторінок.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз предметної області

Аналіз предметної області для веб-сайту розповсюдження та зберігання цифрової звукової інформації включає розгляд наступних аспектів:

1. Дослідження музичної індустрії та її тенденцій, включаючи зміни у способах розповсюдження, споживання та зберігання музики. Аналіз ринку музичних сервісів та конкурентів для визначення потенційних можливостей та викликів.

2. Вивчення сучасних технологій зберігання та передачі цифрової звукової інформації, таких як кодеки аудіо, формати файлів, компресія та розподілені системи зберігання даних. Оцінка їх ефективності, якості та пристосованості до потреб веб-сайту.

3. Вивчення структури аудіофайлів, їх форматів та метаданих, таких як назва треку, виконавець, альбом, жанр тощо. Аналіз методів отримання та обробки метаданих для забезпечення ефективного пошуку, сортування та каталогізації музичного контенту.

4. Розгляд авторських прав та ліцензування музичного контенту, включаючи вимоги щодо дозволів на розповсюдження та зберігання музики відповідно до законодавства. Аналіз можливих обмежень та вимог щодо захисту авторських прав на веб-сайті.

5. Огляд сучасних веб-технологій та фреймворків для розробки веб-сайтів, включаючи мови програмування, такі як HTML, CSS, JavaScript і інші. Аналіз можливостей і обмежень цих технологій у контексті розробки веб-сайту для розповсюдження та зберігання цифрової звукової інформації.

6. Вивчення потреб та очікувань користувачів від веб-сайту з цифровою звуковою інформацією. Аналіз методів навігації, пошуку, відтворення та взаємодії з музичним контентом для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача.

7. Розгляд заходів для захисту музичного контенту та особистих даних користувачів. Аналіз методів аутентифікації, авторизації, шифрування та інших заходів безпеки для забезпечення конфіденційності та цілісності даних.

8. Вивчення можливостей інтеграції веб-сайту з іншими сервісами та платформами, такими як соціальні мережі, музичні API, платіжні системи тощо. Аналіз вимог та можливостей інтеграції для поліпшення функціональності та зручності використання веб-сайту.

9. Огляд технічних аспектів розробки веб-сайту для розповсюдження та зберігання цифрової звукової інформації, включаючи вибір платформи, архітектури, бази даних, хостингу та інфраструктури. Аналіз можливостей розширення та масштабованості системи.

RHP (Hypertext Preprocessor) - це скриптова мова програмування загального призначення, яка використовується для розробки веб-додатків. Основною сферою застосування RHP є створення динамічних веб-сторінок, взаємодія з базами даних та реалізація функціоналу на стороні сервера.

Переваги RHP:

1. Простота вивчення – RHP має простий синтаксис, що дозволяє швидко навчитися його використовувати. Велика кількість ресурсів і документації допомагають розробникам освоїти мову швидко і ефективно.

2. Розширена функціональність – RHP має велику кількість вбудованих функцій і бібліотек, які дозволяють легко виконувати завдання, пов'язані з роботою з базами даних, обробкою форм, роботою з файлами, створенням зображень та багато іншого.

3. Велика спільнота – PHP має велику активну спільноту розробників, що означає, що завжди можна знайти підтримку, допомогу та відповіді на свої запитання.

4. Кросплатформна сумісність – PHP може працювати на різних операційних системах, таких як Windows, Linux, macOS, що дозволяє розробляти веб-додатки на різних середовищах.

Недоліки PHP:

1. Проблеми з безпекою – незнання правильних практик безпеки може призвести до вразливостей в програмах на PHP. Недостатня увага до безпеки може відкрити доступ до атак, таких як впровадження зловмисного коду або втрату конфіденційної інформації.

2. Швидкодія – PHP має деякі обмеження в швидкодії порівняно з деякими іншими мовами програмування. У деяких випадках це може вплинути на продуктивність веб-додатків з великим обсягом обробки даних або великою кількістю одночасних запитів.

3. Перевантаження кодом – PHP дозволяє розробникам використовувати різні підходи і стилі програмування, що може призвести до поганих практик і перевантаження кодом. Неправильне використання функцій і структур може ускладнити розуміння та підтримку коду.

Незважаючи на недоліки, PHP є популярною мовою програмування для розробки веб-додатків, завдяки своїй простоті та широкому спектру функціоналу.

1.2 Аналіз існуючих рішень

При аналізі існуючих рішень для кваліфікаційної роботи на тему "Розробка web-застосунку для розповсюдження та зберігання цифрової звукової інформації", можна врахувати наступні критерії:

1. Функціональність;

2. Інтерфейс користувача;
3. Масштабованість;
4. Безпека;
5. Швидкодія;
6. Підтримка;
7. Вартість;
8. Інтеграція.

1.2.1 Apple Music

Apple Music – це популярний сервіс потокового прослуховування музики, який розроблений компанією Apple Inc. Цей сервіс надає користувачам можливість отримати доступ до широкої бібліотеки музичних треків різних жанрів, виконавців і альбомів [2].

За допомогою Apple Music користувачі можуть шукати свої улюблені пісні, створювати персоналізовані плейлисти та отримувати рекомендації щодо нових виконавців і альбомів, враховуючи їх музичні уподобання. Сервіс також пропонує ексклюзивний контент, включаючи живі виступи, інтерв'ю з музикантами та радіо станції з різними жанрами музики.

Apple Music підтримує різні платформи, включаючи комп'ютери, смартфони та планшети, і забезпечує зручний інтерфейс користувача для прослуховування музики в будь-який час і в будь-якому місці. Крім того, він пропонує можливість зберігати музику офлайн, щоб користувачі могли слухати улюблені треки без доступу до Інтернету.

Оцінка сайту Apple Music за наведеними критеріями:

1. Apple Music надає широкий спектр функціональних можливостей, включаючи пошук та потокове прослуховування музики, створення та

управління плейлистами, рекомендації пісень, відтворення в офлайн-режимі та синхронізацію музики між пристроями.

2. Інтерфейс є естетично привабливим та добре організованим. Навігація є зручною, елементи інтерфейсу логічно розміщені.

3. Успішно масштабується, забезпечуючи стабільну роботу та доступність музичного контенту незалежно від кількості користувачів.

4. Використовує високі стандарти безпеки, включаючи шифрування даних та захист особистої інформації користувачів.

5. Має швидкий час завантаження сторінок та відгуків, забезпечуючи швидке переключення між треками та функціями.

6. Надає широкі можливості підтримки користувачів, включаючи допомогу через електронну пошту, онлайн-довідку та форуми.

7. Оцінка вартості підписки або доступу до повного функціоналу сайту Apple Music та порівняння її з іншими подібними музичними сервісами.

8. Пропонує підписку за певну вартість на місяць або рік, що може бути вище, ніж у деяких конкурентів. Однак, ціна виправдовується широким вибором музики та функціональністю.

Недоліки сервісу Apple Music:

1. Apple Music є найбільш підходящим для користувачів пристроїв Apple, таких як iPhone, iPad і Mac. Він може бути менш зручним для користувачів, які володіють пристроями на інших платформах, таких як Android.

2. Деякі користувачі вказують на складність і неінтуїтивність інтерфейсу Apple Music. Навігація та пошук можуть бути не такими зручними, порівняно з іншими музичними платформами.

3. Іноді Apple Music може мати обмеження щодо доступу до певних пісень або альбомів через авторські права. Деякі музичні твори можуть бути недоступні в деяких регіонах або обмежені для певних категорій користувачів.

4. Для використання Apple Music, особливо на комп'ютері, можуть вимагатися певні системні вимоги, такі як наявність оновленого операційного середовища або встановлення спеціального програмного забезпечення.

5. Порівняно з іншими музичними сервісами, Apple Music може вважатися дорожчим. Ціни на підписку можуть бути вищими, особливо для сімейних або студентських планів.

6. Інтеграція Apple Music зі сторонніми пристроями і платформами може бути обмеженою. Деякі функції та можливості Apple Music можуть бути недоступними, коли ви використовуєте сервіс на пристрої, який не належить до Apple.

Ці недоліки можуть варіюватися залежно від індивідуальних потреб та вимог користувачів.

1.2.2 YouTube Music

YouTube Music є музичною платформою, розробленою компанією YouTube, яка надає користувачам можливість потокового прослуховування музики через Інтернет [1]. Цей сервіс дозволяє знайти, відтворювати та відкривати нову музику, аудіозаписи, музичні відео та живі виступи.

За допомогою YouTube Music користувачі можуть шукати свої улюблені пісні, альбоми, виконавців або жанри, переглядати рекомендації від системи, створювати власні плейлисти та налаштовувати їх за своїми уподобаннями. Сервіс також надає можливість слухати музику офлайн, завантажуючи її на пристрої для прослуховування без підключення до Інтернету.

Проаналізуємо його за наведеними критеріями:

1. Сайт YouTube Music має широкий набір функцій, включаючи можливість пошуку, створення плейлистів, рекомендації на основі вподобань користувача, доступ до музичних відео та оригінального контенту.

2. Інтерфейс сайту є інтуїтивно зрозумілим і зручним для використання. Користувачі можуть легко переходити між розділами, відтворювати музику, виконувати пошук і налаштування.

3. YouTube Music є широко використовуваним сервісом, який має значну кількість користувачів і забезпечує стабільну роботу навіть при великому навантаженні.

4. Вживає заходів для захисту від несанкціонованого доступу та зберігання особистих даних користувачів. Він також має систему звітування і контролю над вмістом, що дозволяє користувачам безпечно використовувати платформу.

5. Забезпечує швидке завантаження музичних треків і відео, а також безперебійне потокове прослуховування навіть при поганому інтернет-з'єднанні.

6. YouTube Music надає підтримку користувачам шляхом надання довідкової інформації, часто задаваних питань та можливості зв'язатися з командою підтримки.

Недоліки сервісу YouTube Music:

1. В безкоштовній версії YouTube Music обмежено можливість прослуховування музики офлайн. Щоб мати повний доступ до функції офлайн-прослуховування, користувачам потрібно придбати платний підписку на YouTube Music Premium.

2. У безкоштовній версії YouTube Music можуть з'являтися рекламні відеоролики. Це може вплинути на користувацький досвід та заважати безперервному прослуховуванню музики.

3. YouTube Music в основному спрямований на музичні відеокліпи, що може бути незручним для користувачів, які бажають слухати музику без відео або просто потребують аудіофайли.

4. Деякі користувачі вказують на неефективність системи рекомендацій YouTube Music, яка може не завжди точно враховувати особисті вподобання користувачів та пропонувати нецікаву музику.

5. У деяких країнах або регіонах можуть бути обмеження щодо доступу до певних музичних творів або альбомів через авторські права або інші обмеження.

6. Підписка на YouTube Music Premium має вартість, яка може вважатися дорожчою порівняно з іншими музичними сервісами, особливо для сімейних або студентських планів.

Ці недоліки можуть бути індивідуальними та залежати від потреб та вимог користувачів.

1.3 Постанова задачі

Мета полягає у розробці web-застосунку для розповсюдження та зберігання цифрової звукової інформації. Застосунок повинен надати користувачам зручний і доступний інструмент для пошуку, відтворення та управління музичним контентом.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Розробити систему реєстрації та авторизації користувачів, що дозволяє створювати та керувати особистими обліковими записами.

2. Розробити інтерфейс для взаємодії з користувачами, забезпечуючи зручний та ефективний доступ до функцій пошуку, прослуховування та управління музичними композиціями.

3. Реалізувати функцію пошуку музичних треків за різними критеріями, такими як назва виконавця або назва пісні, забезпечуючи швидкий та точний пошук.

4. Розробити механізм формування та управління плейлистами, дозволяючи користувачам створювати власні колекції музики та редагувати їх склад.

5. Забезпечити можливість онлайн прослуховування музики без необхідності завантаження треків на пристрій користувача.

6. Розробити базу даних для зберігання та управління цифровою звуковою інформацією, забезпечивши безпеку та ефективну організацію даних.

7. Перевірити функціональність та ефективність розробленого веб-застосунку шляхом проведення відповідних тестів.

1.4 Висновки до першого розділу

У цьому розділі були проведені аналіз предметної області та існуючих рішень, що стосуються досліджуваної проблематики.

Був проведений аналіз предметної області, що дозволило отримати загальне уявлення про характеристики, особливості та основні проблеми, пов'язані з даною областю. Було виявлено ключові аспекти, які необхідно враховувати при вирішенні задачі, а також виявлені основні напрямки досліджень і розвитку в даній області.

Був проведений аналіз існуючих рішень, зокрема розглянуто два популярних сервіси: Apple Music і YouTube Music. Були проаналізовані їх основні характеристики, переваги та недоліки, функціональні можливості та специфіка використання. Цей аналіз дав змогу виявити прогалини і недоліки існуючих рішень, які можуть бути враховані при розробці нового рішення.

Була поставлена задача, яка є основою для подальшого дослідження та розробки нового рішення. Враховуючи результати аналізу предметної області та існуючих рішень, були визначені основні вимоги та цілі, які повинні бути виконані в новому рішенні.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Розробка бази даних веб-сайту

Розробка бази даних є важливою складовою процесу створення веб-сайту. База даних веб-сайту забезпечує зберігання, управління та доступ до інформації, яка використовується на сайті. Вона дозволяє організувати структуроване зберігання даних і забезпечує ефективний доступ до них [3].

2.1.1 Побудова концептуальної моделі

Концептуальна модель бази даних (БД) - це відображення структури і взаємозв'язків даних в описовій формі. Вона описує загальну структуру БД, незалежно від конкретної технології або системи управління базами даних (СУБД), і зосереджується на концепціях і сутностях даних.

Концептуальна модель БД використовується для розуміння та представлення бізнес-домену, вимог користувачів та основних сутностей, які існують у системі. Вона не залежить від конкретних таблиць, полів або схеми БД, але фокусується на сутностях, атрибутах і взаємозв'язках між ними.

Концептуальна модель може бути представлена у вигляді схеми з поняттями, що відображають сутності, їх взаємозв'язки та атрибути. Вона може включати сутності, такі як люди, об'єкти, події або концепції, а також їх характеристики. Взаємозв'язки між сутностями відображаються за допомогою асоціацій, які показують, як сутності пов'язані між собою.

Розробимо концептуальну модель бази даних веб-сайту «Прослуховування музики».

Сутності та їх атрибути представлені у таблиці 2.1.

Таблиця сутностей та їх атрибутів

Сутність	Атрибути
1	2
Artists	Id; Name.
Genres	Id; Name.
Playlist	Id; Name; User id.
Playlist_music	Id; Playlist id; Music id.
Music	Id; Title; Artist id; Genre id; Path; Avatar.

Концептуальна модель даних представлена на рисунку 2.1.

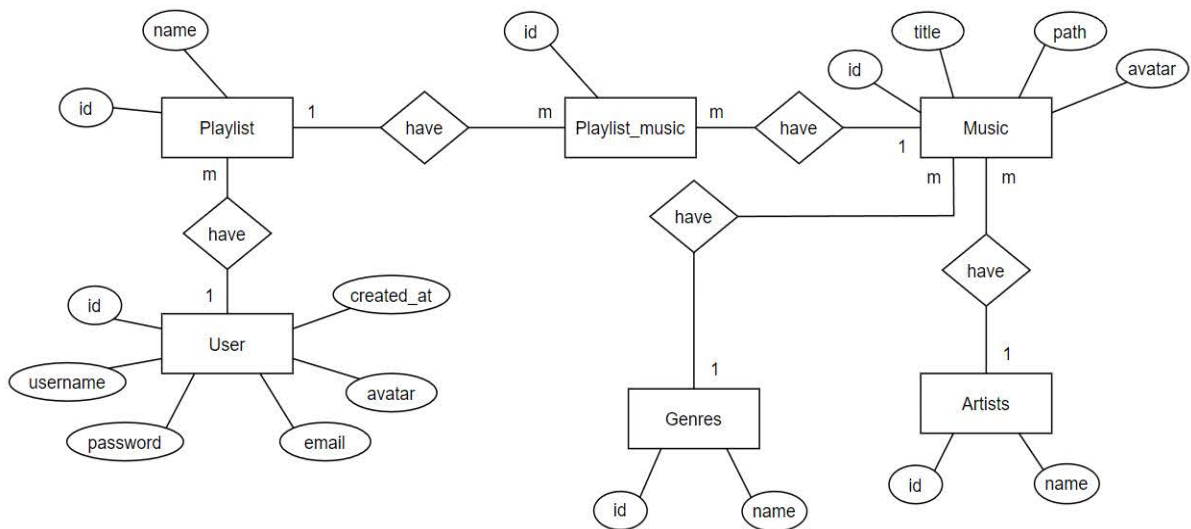


Рисунок 2.1 – Концептуальна модель даних веб-сайту

2.1.2 Побудова логічної моделі

Логічна модель бази даних (БД) є проміжним рівнем між концептуальною моделлю і фізичною реалізацією БД. Вона визначає спосіб організації даних у базі даних, включаючи таблиці, поля, ключі, зв'язки і обмеження цілісності.

Логічна модель БД заснована на концептуальній моделі, але вона враховує вже конкретну Систему Управління Базами Даних (СУБД), з якою буде працювати система. Це означає, що логічна модель використовує специфічні типи даних, синтаксис і можливості, які надає СУБД.

У логічній моделі БД сутності концептуальної моделі перетворюються на таблиці, атрибути стають полями, а взаємозв'язки між сутностями відображаються за допомогою зовнішніх ключів. Логічна модель також визначає обмеження цілісності, такі як унікальність значень або обов'язковість деяких полів [4, 5, 26].

На даному етапі представлена реляційна модель [7].

Логічна модель представлена на рисунку 2.2.

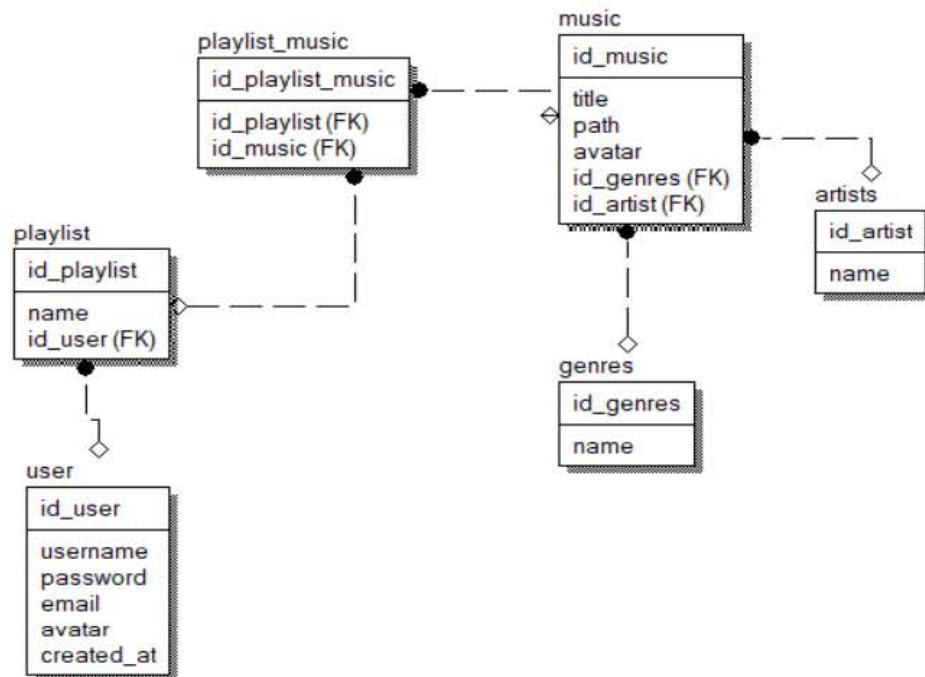


Рисунок 2.2 – Логічна модель веб-сайту

2.1.3 Розробка фізичної моделі бази даних

Фізична модель бази даних (БД) визначає спосіб фізичного зберігання і доступу до даних в БД на рівні конкретної Системи Управління Базами Даних (СУБД). Вона враховує характеристики конкретної СУБД, такі як типи даних, індексація, структура файлів, алгоритми доступу і оптимізації запитів.

Фізична модель БД заснована на логічній моделі, але деталізує спосіб фактичного зберігання даних. Вона визначає, як таблиці і поля логічної моделі перетворюються на фізичні файли, блоки або сегменти на диску, і як дані представлені у вигляді бінарних значень.

Фізична модель включає в себе такі елементи, як структура таблиць і файлів, методи індексування для прискорення пошуку даних, способи зберігання інформації у файловій системі або спеціальних структурах даних СУБД. Вона також враховує фактори швидкості, ефективності і безпеки доступу до даних.

Фізична модель БД визначає фізичну структуру таблиць, включаючи розташування полів і їх типи даних, індекси для підтримки швидкого пошуку, алгоритми сортування та оптимізації запитів для підвищення продуктивності. Крім того, вона визначає правила цілісності для забезпечення правильності даних на фізичному рівні.

Розробимо фізичну модель баз даних для сайту з прослуховування музики.

Фізичну модель розроблено у PhpMyAdmin, представлена на рисунку 2.3 представлена фізична модель БД [4].

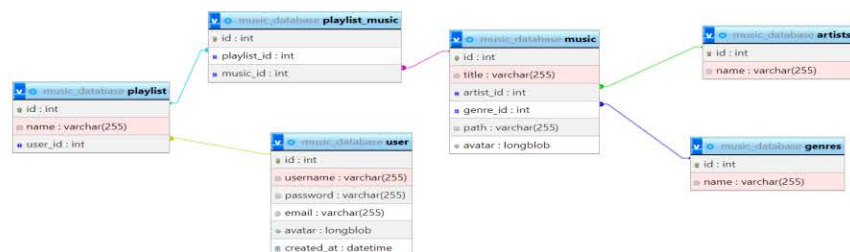


Рисунок 2.3 – Фізична модель БД веб-сайту

Відображення типу поля, типу даних та обмеження на ці поля кожної таблиці представлено на рисунках 2.4 -2.8.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id 🔑	int		Ні	Немає		AUTO_INCREMENT	Більше ▼
☐	2	name	varchar(255) <i>utf8mb3_general_ci</i>		Ні	Немає			Більше ▼

Рисунок 2.4 – Структура таблиці artists

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id 🔑	int		Ні	Немає		AUTO_INCREMENT	Більше ▼
☐	2	name	varchar(255) <i>utf8mb3_general_ci</i>		Ні	Немає			Більше ▼

Рисунок 2.5 – Структура таблиці genres

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id 🔑	int		Ні	Немає		AUTO_INCREMENT	Більше ▼
☐	2	title	varchar(255) <i>utf8mb3_general_ci</i>		Ні	Немає			Більше ▼
☐	3	artist_id 🔑	int		Ні	Немає			Більше ▼
☐	4	genre_id 🔑	int		Ні	Немає			Більше ▼
☐	5	path	varchar(255) <i>utf8mb3_general_ci</i>		Ні	Немає			Більше ▼
☐	6	avatar	longblob		Так	NULL			Більше ▼

Рисунок 2.6 – Структура таблиці music

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id 🔑	int		Ні	Немає		AUTO_INCREMENT	Більше ▼
☐	2	name	varchar(255) <i>utf8mb3_general_ci</i>		Ні	Немає			Більше ▼
☐	3	user_id 🔑	int		Ні	Немає			Більше ▼

Рисунок 2.7 – Структура таблиці playlist

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id 🔑	int		Ні	Немає		AUTO_INCREMENT	Більше ▼
☐	2	playlist_id 🔑	int		Ні	Немає			Більше ▼
☐	3	music_id 🔑	int		Ні	Немає			Більше ▼

Рисунок 2.8 – Структура таблиці playlist_music

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id			Hi	Немає		AUTO_INCREMENT	Більше
☐	2	username	varchar(255)	utf8mb3_general_ci	Hi	Немає			Більше
☐	3	password	varchar(255)	utf8mb3_general_ci	Hi	Немає			Більше
☐	4	email	varchar(255)	utf8mb3_general_ci	Hi	Немає			Більше
☐	5	avatar	longblob		Hi	Немає			Більше
☐	6	created_at	datetime		Hi	Немає			Більше

Рисунок 2.8 – Структура таблиці user

2.2 Розробка макету веб-сайту

Розробка макету веб-сайту є важливим етапом процесу створення веб-проекту.

Макет веб-сайту визначає його зовнішній вигляд, компоновання елементів, структуру та взаємодію між ними. Цей етап дозволяє визначити загальний дизайн та вигляд сторінок сайту перед їх фактичною розробкою [6].

Розроблений макет сторінок майбутнього веб-сайту представлено на рисунках 2.9 – 2.17.

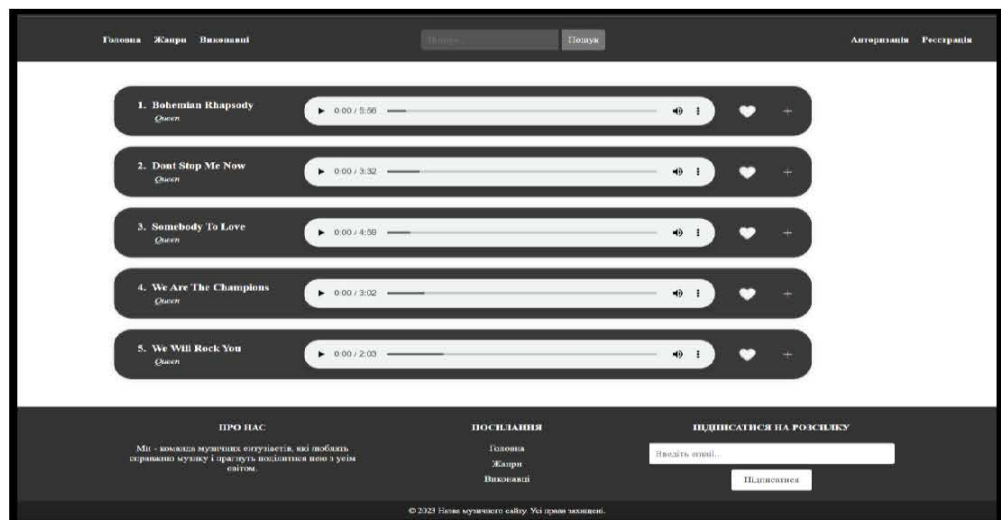


Рисунок 2.9 – Макет головної сторінки

The registration form is centered on a dark gray background. It features a title 'Регистрація користувача' and four input fields: 'Логін користувача:', 'Пароль:', 'Пошта:', and 'Аватар:'. The 'Аватар' field includes a file selection button 'Выберите файл' and a status indicator 'Файл не выбран'. A blue 'Зареєструватися' button is at the bottom right of the form. The top navigation bar includes links for 'Головна', 'Жанри', 'Виконавці', a search bar, and 'Авторизація' / 'Регистрація'. The footer contains three sections: 'ПРО НАС' with a mission statement, 'ПОСИЛАННЯ' with site navigation links, and 'ПІДПИСАТИСЯ НА РОЗСІЛКУ' with an email subscription form. A copyright notice is at the very bottom.

Рисунок 2.10 – Макет сторінки реєстрації

На рисунку 2.11 подано вигляд вікна авторизації. Для авторизації необхідно ввести ім'я користувача та пароль.

The authorization form is centered on a dark gray background. It features a title 'Авторизація користувача' and two input fields: 'Ім'я користувача:' and 'Пароль:'. A blue 'Увійти' button is at the bottom of the form. The top navigation bar is identical to the registration page. The footer sections are also identical, including the 'ПРО НАС', 'ПОСИЛАННЯ', and 'ПІДПИСАТИСЯ НА РОЗСІЛКУ' sections, along with the copyright notice.

Рисунок 2.11 – Макет сторінки авторизації

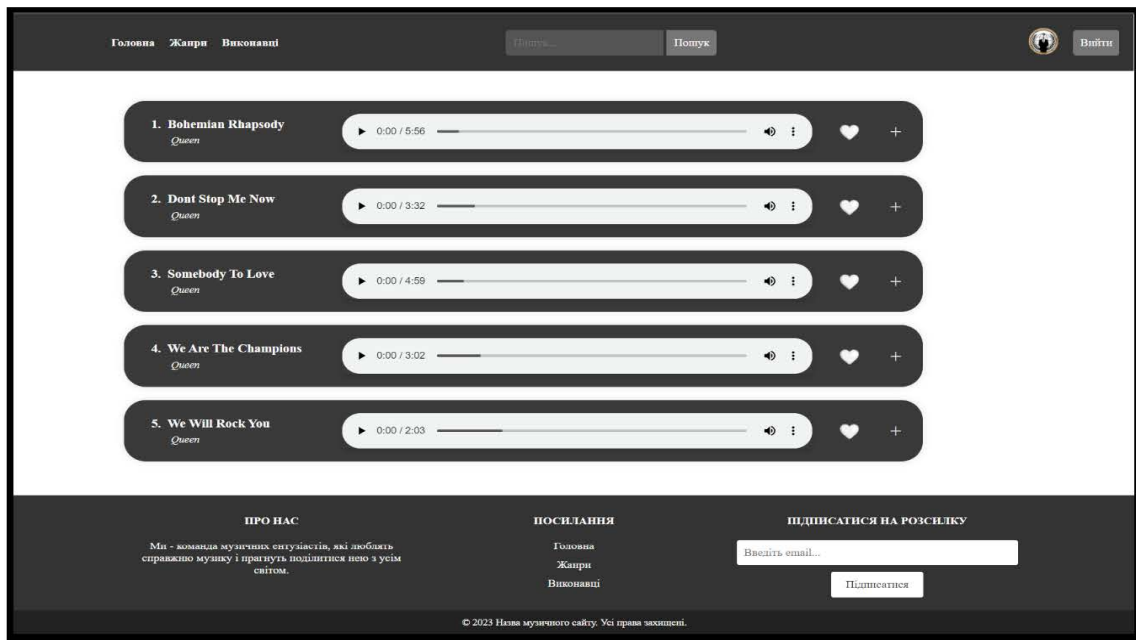


Рисунок 2.12 – Макет головної сторінки при авторизації користувача

На рисунку 2.13 подано макет сторінки профілю користувача. Як бачимо, макет розроблено з урахуванням останніх тенденцій в дизайні.

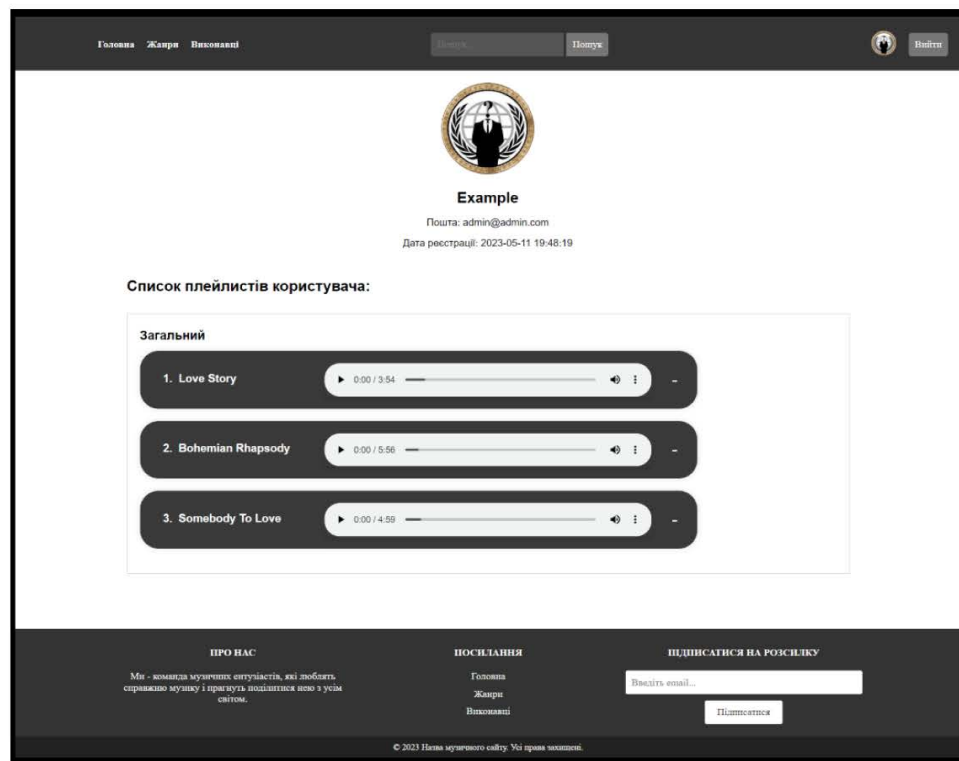


Рисунок 2.13 – Макет сторінки профілю користувача

Рисунок 2.14 – Макет сторінки додавання пісні до бази даних

На рисунку 2.14 та 2.15 подано макети сторінки додавання пісні до бази даних та сторінка жанрів.

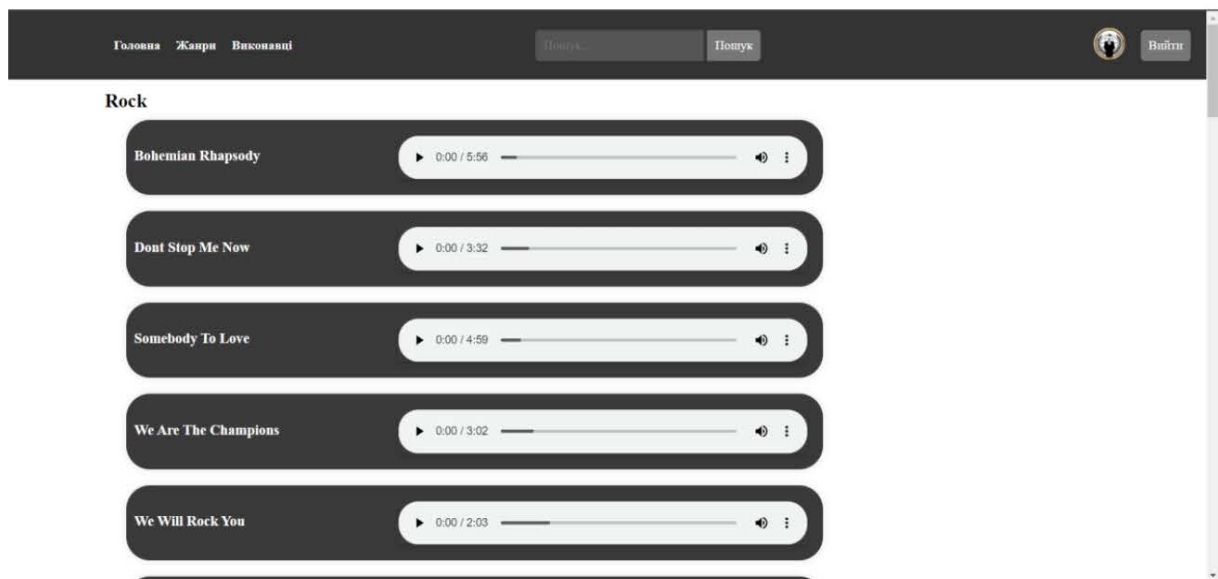


Рисунок 2.15 – Макет сторінки жанрів

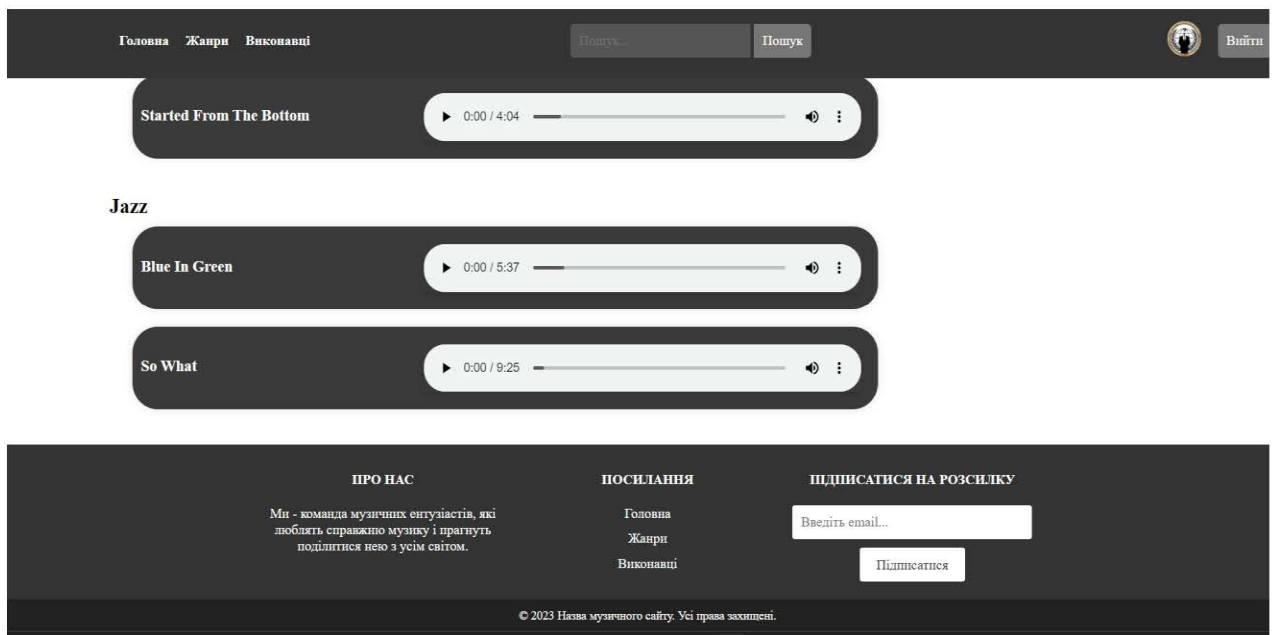


Рисунок 2.16 – Макет сторінки жанрів

На рисунку 2.17 подано вигляд макету форми додавання нового плейлиста. Як видно з рисунків, все витримано в одному стилі та дизайнерські обґрунтовано.

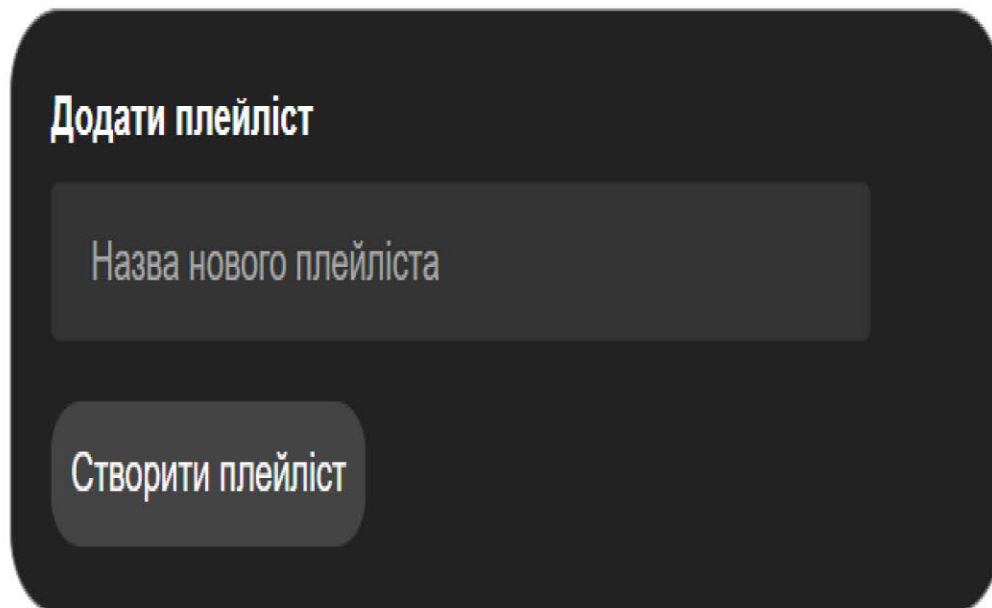


Рисунок 2.17 – Макет форми додавання нового плейліста

2.3 Висновки до другого розділу

У даному розділі було проведено проектування програмного забезпечення, зокрема розроблено базу даних веб-сайту та макет веб-сайту.

Була проведена робота над створенням бази даних для підтримки функціональності веб-сайту. Спочатку була побудована концептуальна модель, яка дозволила визначити основні сутності та взаємозв'язки між ними. На основі концептуальної моделі була розроблена логічна модель бази даних, яка визначила структуру таблиць та зв'язки між ними. Крім того, була розроблена фізична модель бази даних, в якій були визначені типи даних, індекси, обмеження та інші параметри, необхідні для її реалізації.

Було створено візуальне представлення веб-сайту у вигляді макету. Застосовано сучасні інструменти для розробки інтерфейсу користувача, включаючи графічні елементи, кольорову схему, компоненти та розташування елементів на сторінці. Макет був створений з урахуванням вимог функціональності та зручності використання веб-сайту.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ РОЗРОБКИ WEB-ДОДАТКУ ДЛЯ РОЗПОВСЮДЖЕННЯ ТА ЗБЕРІГАННЯ ЦИФРОВОЇ ЗВУКОВОЇ ІНФОРМАЦІЇ

3.1 Технічне завдання та етапи розробки

Для кваліфікаційної роботи бакалавра було поставлено наступні технологічні завдання:

1. Розробити систему реєстрації та входу в особистий акаунт, яка надасть користувачам можливість створення профілю та змінювати особисті дані. Детально розглянути процеси аутентифікації та авторизації для забезпечення безпеки і конфіденційності користувачів.

2. Розробити функціонал для пошуку музичних композицій.

3. Розробити можливість формування плейлистів, що дозволить користувачам створювати власні списки відтворення та додавати пісні до існуючих плейлистів. Забезпечити зручну навігацію по плейлистам та можливість редагування, видалення та переміщення пісень у списку.

4. Розробити функцію онлайн прослуховування музики, що дозволить користувачам стрімити пісні безпосередньо з сайту, не завантажуючи їх на свій пристрій. Забезпечити оптимальну якість звуку та мінімальні затримки при програванні.

5. Забезпечити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що сприятиме зручній навігації та використанню функціоналу web-застосунку.

6. Розглянути принципи дизайну і взаємодії, що сприятимуть покращенню користувацького досвіду.

Етапи розробки веб-сайту з прослуховування музики на PHP з маршрутизацією та наступними сторінками (головна, жанри, виконавці, авторизація, реєстрація, профіль):

1. Аналіз вимог. На цьому етапі проводиться детальний аналіз вимог користувачів до веб-сайту. Визначаються основні функції, які має виконувати сайт, такі як можливість пошуку та прослуховування музики, перегляду жанрів та виконавців, авторизація та реєстрація користувачів, редагування профілю тощо. Також визначаються технічні вимоги, такі як підтримка мови програмування PHP, бази даних MySQL та фреймворка для маршрутизації [10-12, 24-26].

2. Проектування бази даних.

На цьому етапі створюється структура бази даних для зберігання інформації про музичні жанри, виконавців, користувачів та інші необхідні дані. Визначаються таблиці, поля і зв'язки між ними. Наприклад, можна створити таблиці "жанри", "виконавці", "користувачі" з відповідними полями, які відображають характеристики цих об'єктів.

3. Вибір технологій.

На цьому етапі вибираються технології, які будуть використовуватися для розробки веб-сайту. У вашому випадку, це мова програмування PHP, база даних MySQL та фреймворк для маршрутизації, наприклад Laravel або Symfony. Вибір технологій залежить від вашого досвіду, потреб проекту та попередньої аналітики [10-14].

4. Розробка сторінок.

На цьому етапі розробляються окремі сторінки веб-сайту, такі як головна, жанри, виконавці, авторизація, реєстрація, профіль. Кожна сторінка має свій унікальний вигляд та функціонал. Для розробки використовуються HTML, CSS та JavaScript, які дозволяють створити інтерактивний та привабливий інтерфейс [15-21].

5. Реалізація маршрутизації.

На цьому етапі використовується фреймворк для маршрутизації, який дозволяє визначати правила маршрутизації для кожної сторінки. Наприклад, можна визначити, що URL `"/genres"` буде відповідати сторінці зі списком жанрів, а URL `"/profile"` - сторінці профілю користувача. Це дозволяє визначити, який контролер та дія буде викликано при доступі до певного URL та обробляти параметри запиту.

6. Робота з базою даних.

Розробляються функції для взаємодії з базою даних. Наприклад, можна реалізувати функції для отримання та збереження інформації про жанри, виконавців та користувачів. Для взаємодії з базою використовується мова запитів SQL.

7. Реалізація функціоналу авторизації та реєстрації.

Розробляються функції для реєстрації нових користувачів та авторизації існуючих. Валідація введених даних, збереження сесій та забезпечення безпеки від несанкціонованого доступу.

8. Розробка функціоналу профілю користувача.

Реалізується можливість користувачам редагувати свій профіль, змінювати пароль, додавати вибрані пісні до свого списку, переглядати історію прослуховування тощо.

9. Тестування та налагодження.

Після розробки кожного етапу проводиться тестування, щоб переконатися, що функціонал працює правильно та відповідає вимогам. виправляються помилки та вирішуються проблеми, що виникають під час тестування.

10. Розгортання.

Після успішного завершення тестування та налагодження сайт готовий до розгортання на веб-сервері. Завантаження файлів, налаштування веб-сервера та бази даних, перевірка працездатності на реальному середовищі.

3.2 Інтерфейс користувача

Результати розробки веб-застосунку для розповсюдження та зберігання цифрової звукової інформації полягають у створенні функціонального та зручного інструменту доступу до музичного контенту.

Під час виконання кваліфікаційна роботи створено наступні сторінки:

1. Головна сторінка;
2. Сторінка авторизації;
3. Сторінка реєстрації;
4. Сторінка профілю;
5. Сторінка додавання нової пісні до бази даних.

На головній сторінці користувачі зустрічаються зі спрощеним та привабливим дизайном, де вони можуть отримати доступ до основних функцій веб-застосунку, таких як пошук композицій, створення та перегляд плейлистів, а також прослуховування музики онлайн.

Сторінки авторизації та реєстрації призначено для реєстрації нових користувачів та входу в особистий кабінет. Користувачі можуть створити свій профіль, вказавши необхідну інформацію, яка буде використовуватись для подальшої ідентифікації та налагодження персоналізованого досвіду використання веб-застосунку. Ця сторінка також може містити інформацію про вже створені плейлисти та пісні, що додані до улюблених.

Сторінка додавання нової пісні до бази даних дозволяє користувачам завантажувати та додавати нову музичну композицію до колекції веб-застосунку. Користувачі можуть вказати інформацію про пісню, таку як назва, виконавець, жанр тощо, щоб зробити її доступною для пошуку та додавання до плейлистів.

Створені сторінки представлено на рисунках 3.1 – 3.6.

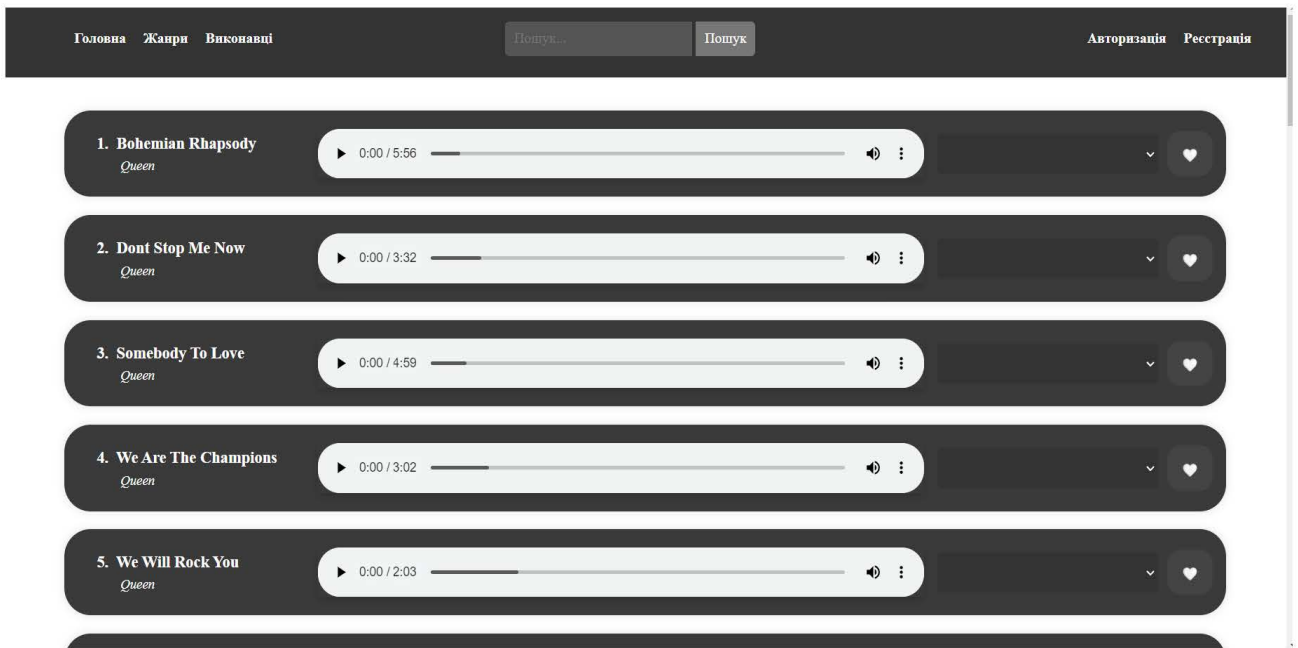


Рисунок 3.1 – Представлення розробленої головної сторінки

На рисунку 3.2 подано вигляд представлення розробленої сторінки реєстрації. Як бачимо, все реалізовано за вищеподаними макетами.

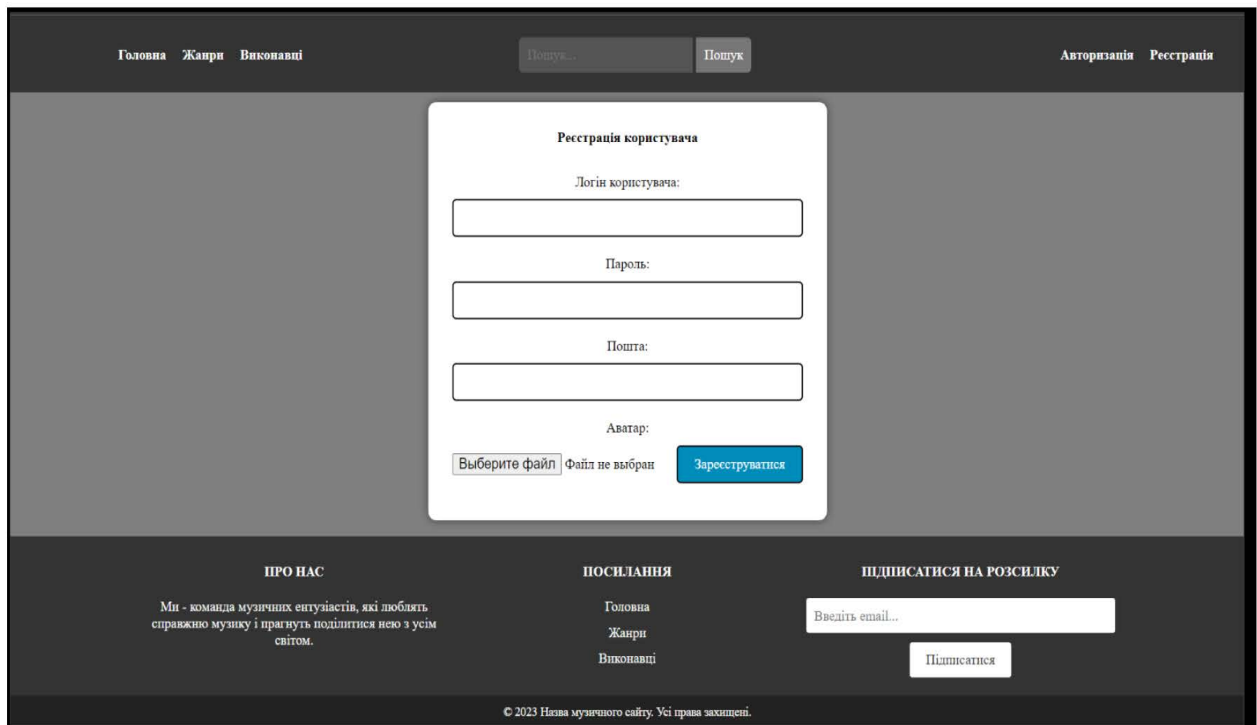


Рисунок 3.2 – Представлення розробленої сторінки реєстрації

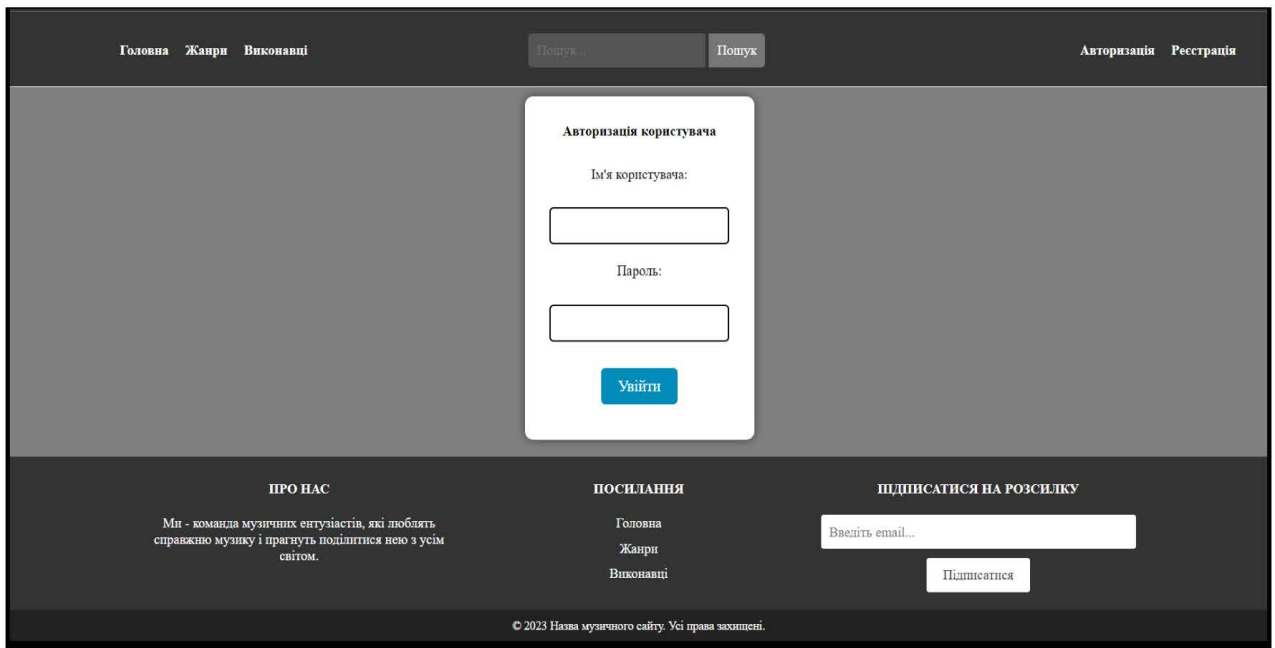


Рисунок 3.3 – Представлення розробленої сторінки авторизації

На рисунку 3.4 подано представлення розробленої головної сторінки при авторизації користувача згідно до розробленого дизайну.



Рисунок 3.4 – Представлення розробленої головної сторінки при авторизації користувача

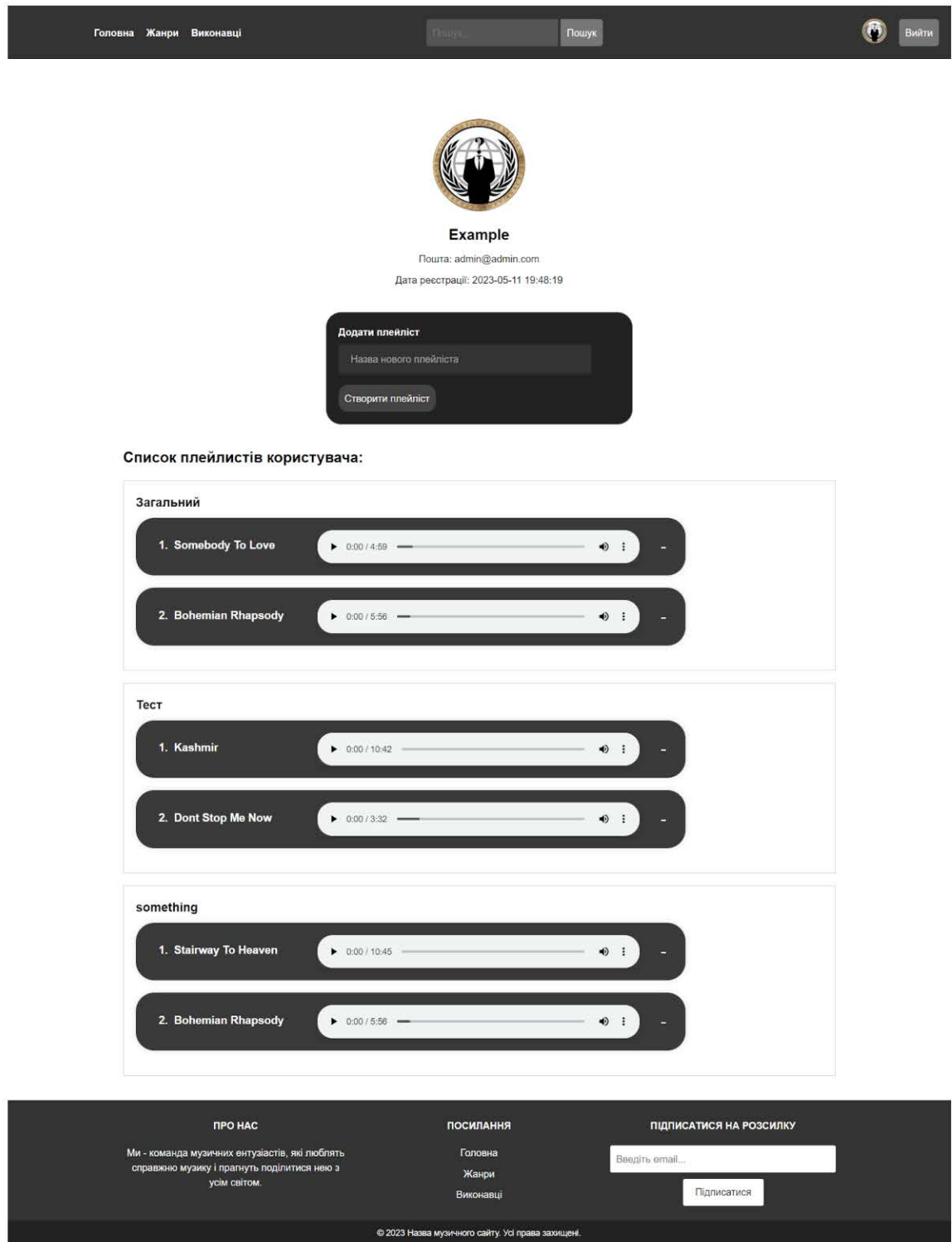


Рисунок 3.5 – Представлення розробленої сторінки профілю користувача

На рисунку 3.5 подано представлення розробленої сторінки профілю користувача. Як бачимо, сторінка реалізована згідно з вимогами користувача.

Рисунок 3.6 – Представлення розробленої сторінки додавання пісні до бази даних

На рисунку 3.7 подано представлення розробленої сторінки жанрів, що розроблена згідно з обраним дизайном та вимогами користувача.

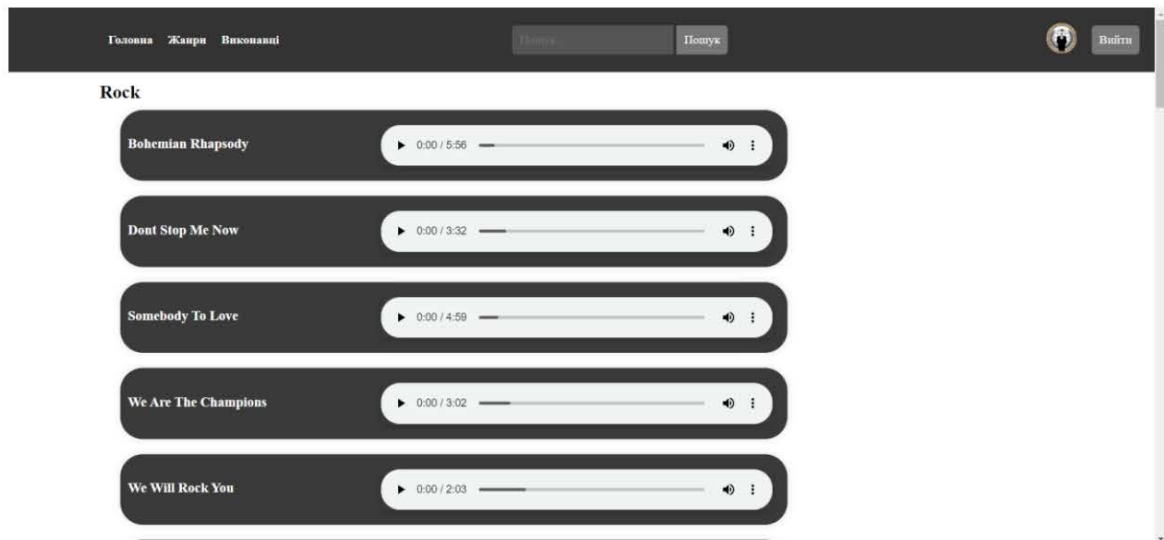


Рисунок 3.6 – Представлення розробленої сторінки жанрів

3.3 Інструкція користувача

Введення веб-сайту:

- Відкрийте ваш веб-браузер.
- Введіть адресу веб-сайту в адресному рядку браузера: `http://music-site/`, представлено на рисунку 3.7.

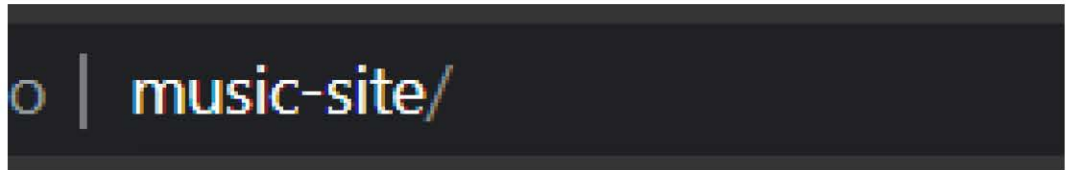


Рисунок 3.7 – Введення адреси сайту

Головна сторінка:

- Після завантаження веб-сайту ви побачите головну сторінку, представлено на рисунку 3.8.

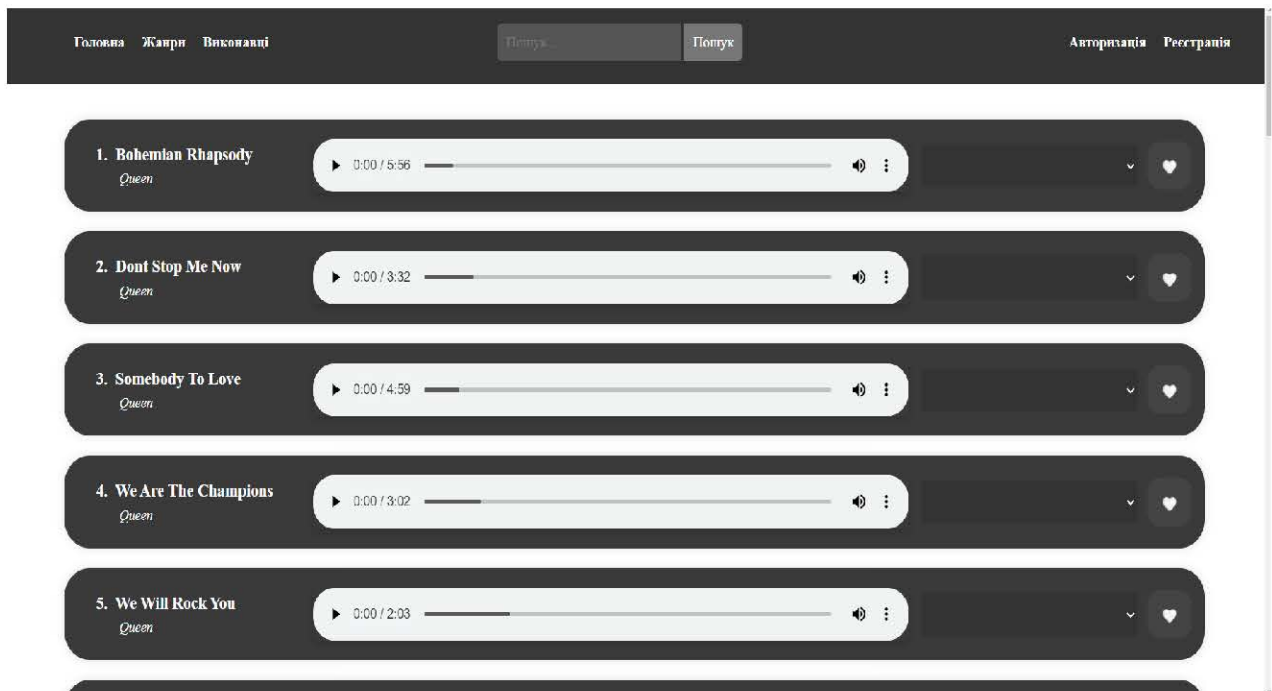


Рисунок 3.8 – Відображення головної сторінки

- На головній сторінці ви знайдете список популярних треків, виконавців та жанрів музики.
- Щоб перейти до інших сторінок, використовуйте навігаційне меню, представлено на рисунку 3.9.

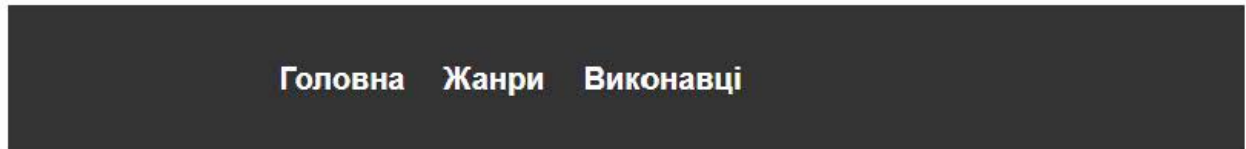


Рисунок 3.9 – Відображення навігаційного меню

Сторінка "Жанри":

- Клацніть на посилання "Жанри" у навігаційному меню, щоб перейти на сторінку жанрів, представлено на рисунку 3.9.
- На сторінці "Жанри" буде відображений список доступних жанрів музики, представлено на рисунку 3.10-3.11.

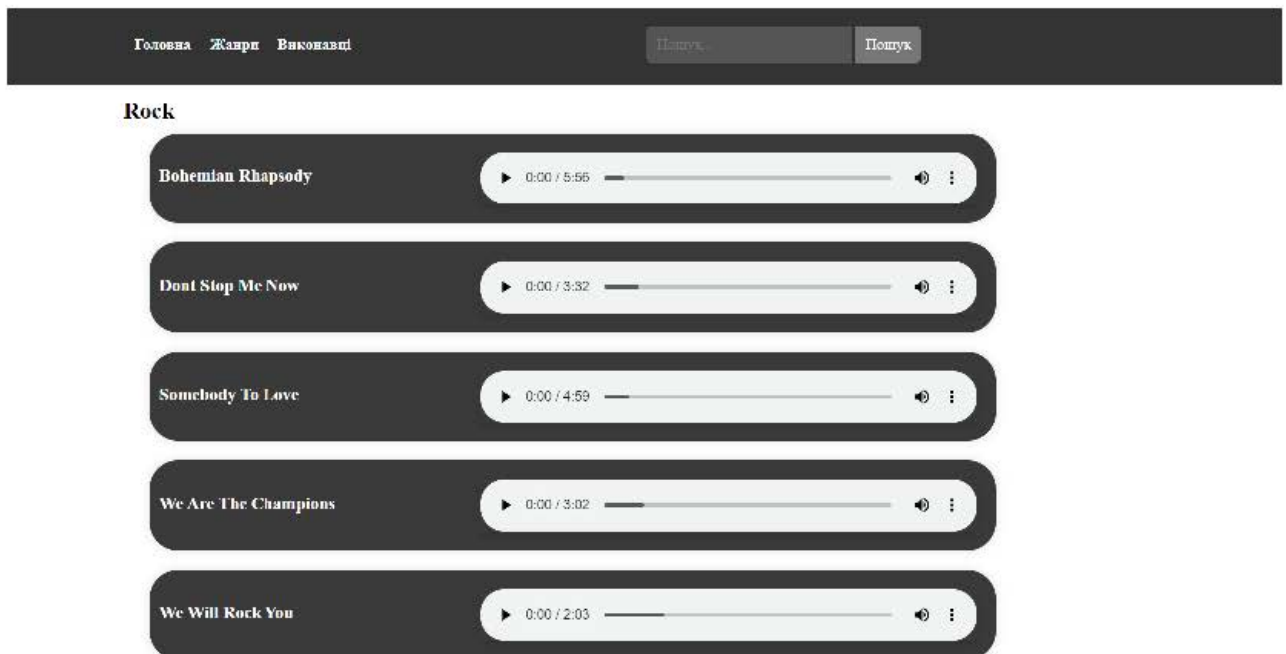


Рисунок 3.10 – Відображення сторінки жанрів

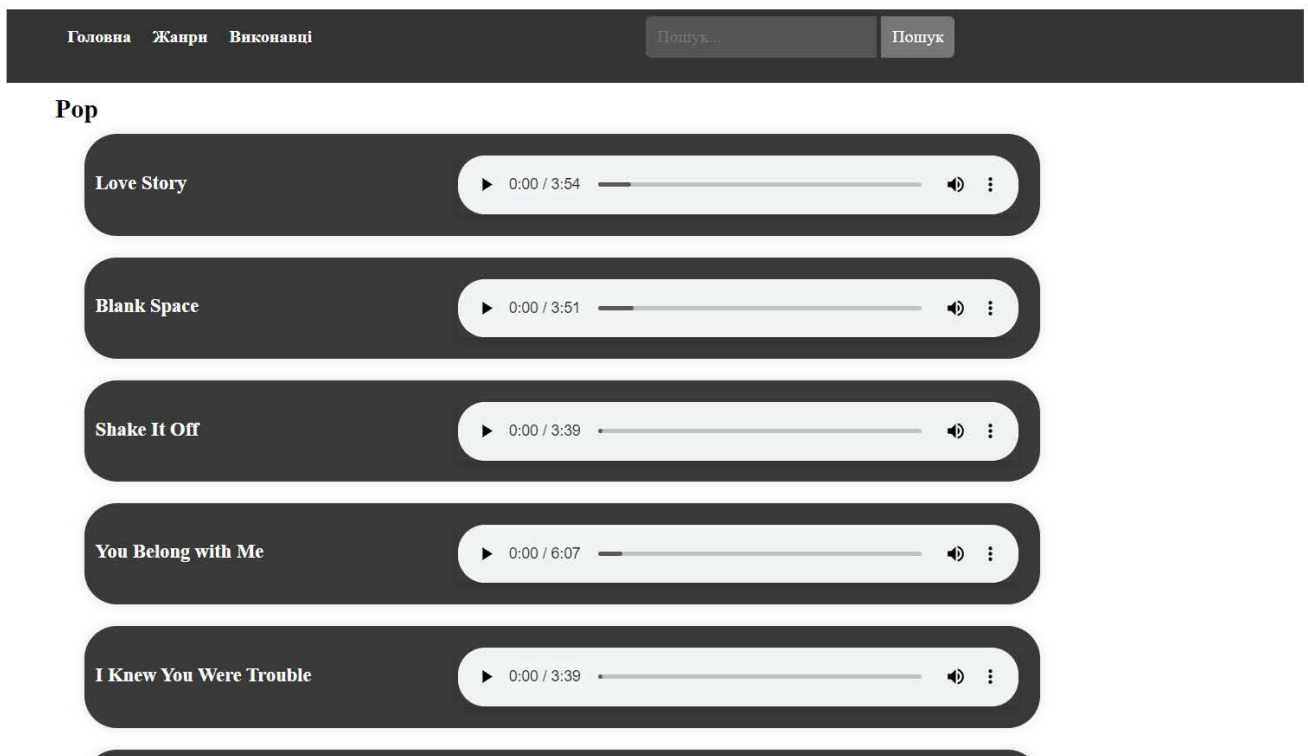


Рисунок 3.11 – Відображення сторінки жанрів

Сторінка "Виконавці":

– Клацніть на посилання "Виконавці" у навігаційному меню, щоб перейти на сторінку виконавців, представлено на рисунку 3.9.

– На сторінці "Виконавці" буде відображений список доступних виконавців музики, представлено на рисунку 3.12-3.13.

З рисунку 3.11 видно, що разом з назвою пісні, виконавцем відображається жанр, можливість змінити гучність та інше.

На рисунку 3.12 подано вигляд сторінки виконавців. Як бачимо, відображаються пісні від гурту Queen.

У такому разі не відображається жанр пісень, що відповідає загальній практиці подання плейлистів.

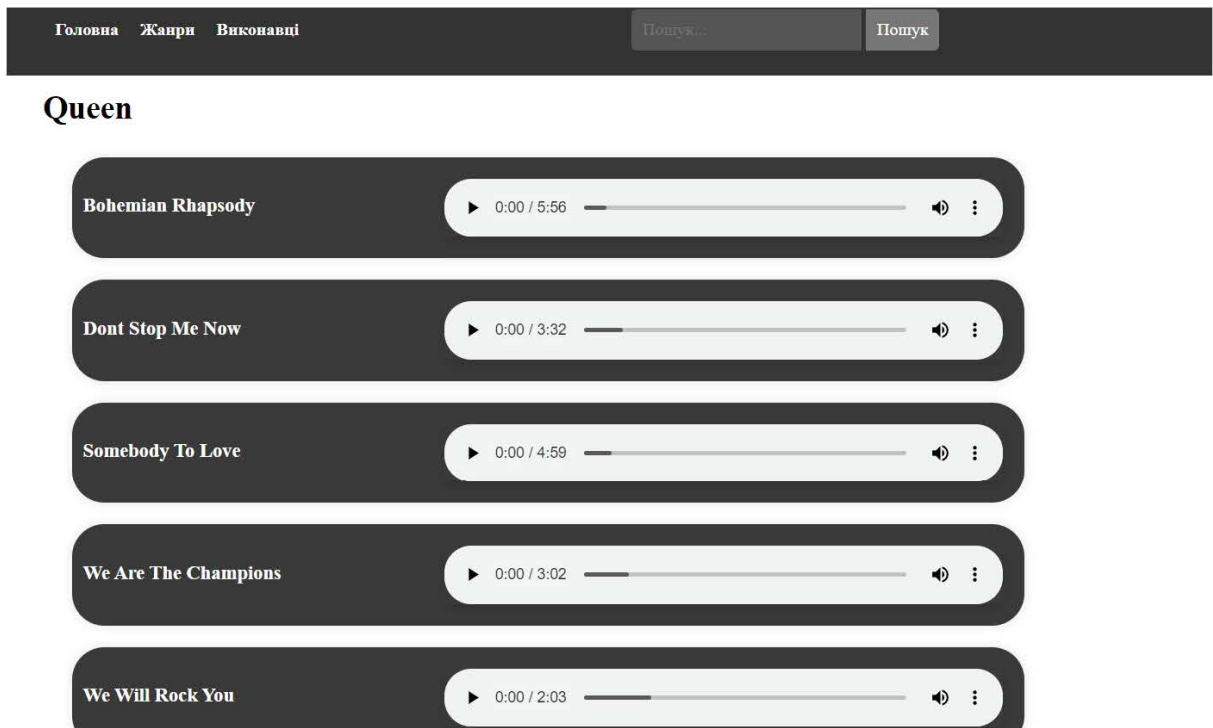


Рисунок 3.12 – Відображення сторінки виконавців

Led Zeppelin



Рисунок 3.13 – Відображення сторінки виконавців

На рисунку 3.13 подано пісні групи Led Zeppelin.

Розглянемо наступну реалізовану сторінку.

Сторінка "Авторизація".

– Якщо ви бажаєте увійти на сайт або створити новий обліковий запис, клацніть на посилання "Авторизація" у навігаційному меню, представлено на рисунку 3.14.

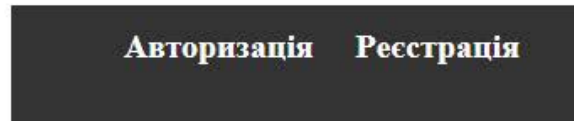


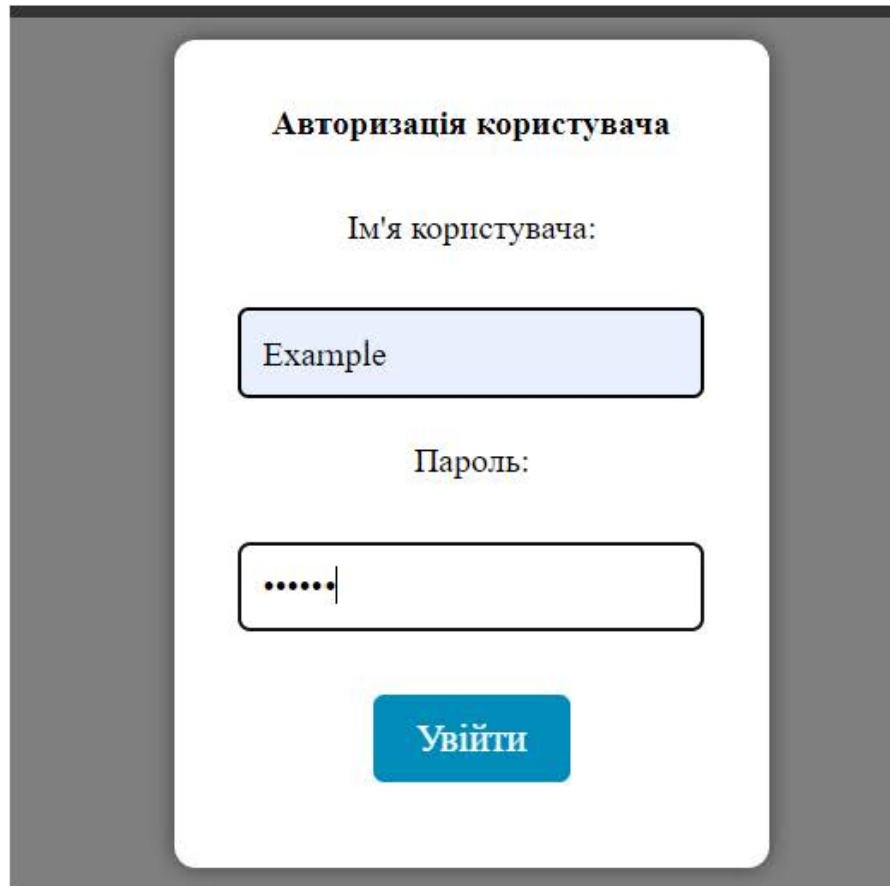
Рисунок 3.14 – Відображення навігаційного меню з кнопками для авторизації та реєстрації

– На сторінці "Авторизація" буде доступна форми для входу, представлено на рисунку 3.15.

Зображення форми авторизації на веб-сайті. Форма має темну шапку з пошуковими полями. Основна частина форми є білою карткою з заголовком "Авторизація користувача". Вона містить два текстові поля: "Ім'я користувача:" та "Пароль:". Під полями розташована синя кнопка "Увійти". Нижче форми знаходиться темна панель з посиланнями "Забудувати пароль" та "Головна".

Рисунок 3.15 – Відображення форми авторизації

– Введіть свої облікові дані або створіть новий обліковий запис згідно вказівок на сторінці, представлено на рисунку 3.16.



The image shows a login form titled "Авторизація користувача" (User Authorization). It contains two input fields: "Ім'я користувача:" (Username) with the text "Example" and "Пароль:" (Password) with masked characters ".....". Below the fields is a blue button labeled "Увійти" (Login).

Рисунок 3.16 – Відображення заповненої форми авторизації

Сторінка "Реєстрація".

– Якщо ви бажаєте увійти на сайт та створити новий обліковий запис, клацніть на посилання "Реєстрація" у навігаційному меню, представлено на рисунку 3.17.

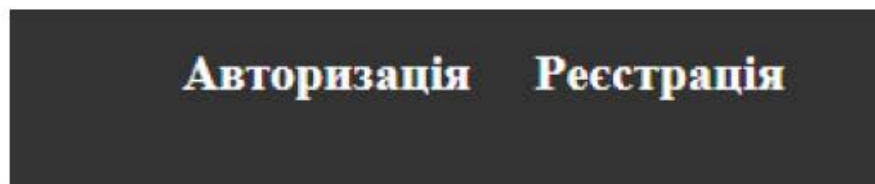
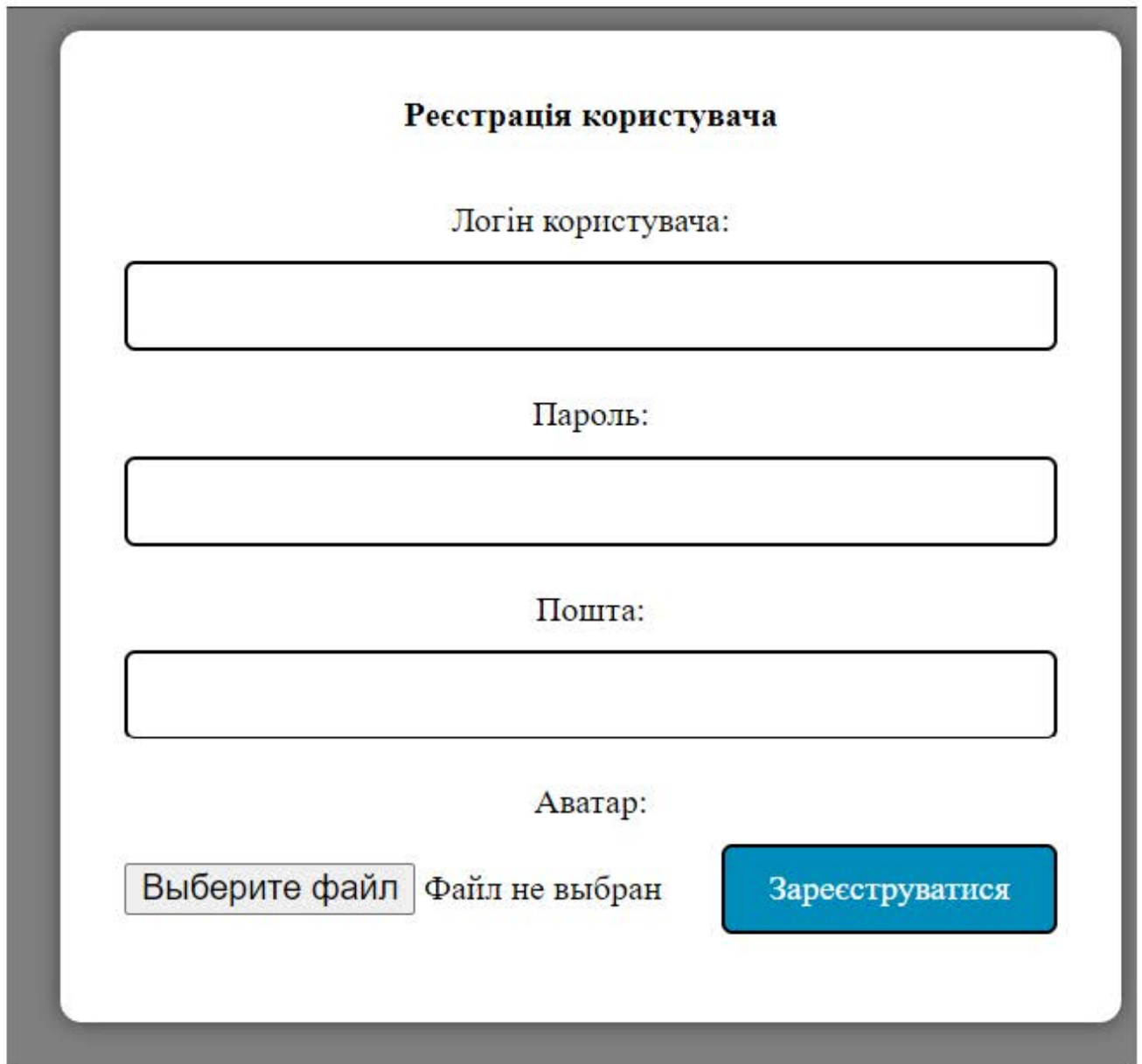


Рисунок 3.17 – Відображення навігаційного меню з кнопками для авторизації та реєстрації

– На сторінці "Реєстрація" буде доступна форма для реєстрації, представлено на рисунку 3.18.



The image shows a registration form titled "Реєстрація користувача" (User Registration). It contains four input fields: "Логін користувача:" (User Login), "Пароль:" (Password), "Пошта:" (Email), and "Аватар:" (Avatar). Below the "Аватар:" field, there is a button labeled "Выберите файл" (Select file) and the text "Файл не выбран" (File not selected). To the right of these is a blue button labeled "Зареєструватися" (Register).

Рисунок 3.18 – Відображення форми реєстрації

– Введіть свої облікові дані або створіть новий обліковий запис згідно вказівок на сторінці, представлено на рисунку 3.19.

The image shows a registration form titled "Реєстрація користувача" (User Registration). It contains four input fields: "Логін користувача:" (User Login) with the value "22120804", "Пароль:" (Password) with masked characters ".....", "Пошта:" (Email) with the value "example@mail.com", and "Аватар:" (Avatar). Below the avatar field, there is a button labeled "Выберите файл" (Select file) and the text "Файл не выбран" (File not selected). To the right of these is a blue button labeled "Зареєструватися" (Register).

Рисунок 3.19 – Відображення форми реєстрації з заповненими даними

Сторінка "Профіль":

– Після успішної авторизації ви зможете перейти на свою особисту сторінку, натиснувши на посилання "Профіль" у навігаційному меню, представлено на рисунку 3.20.

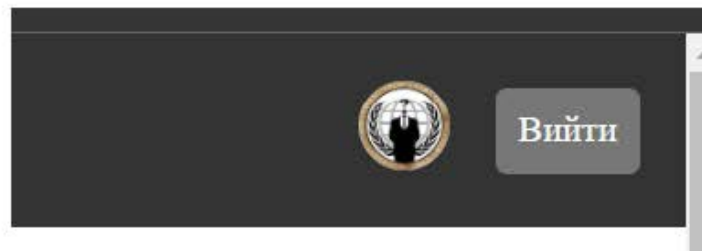


Рисунок 3.20 – Відображення навігаційного меню з посиланням на профіль та кнопкою виходу

– На сторінці "Профіль" буде відображена інформація про ваш обліковий запис та ваші налаштування, представлено на рисунку 3.21.



Рисунок 3.21 – Відображення сторінки профілю з інформацією про користувача

– Ви можете перевірити плейлісти, пароль або виконати інші дії, що доступні на сторінці профілю.

Навігація:

– Для переходу між сторінками використовуйте навігаційне меню, розташоване у верхній частині веб-сайту, представлено на рисунку 3.9.

– Клацайте на посилання, щоб перейти на відповідну сторінку.

Відтворення музики: на багатьох сторінках будуть присутні елементи управління музикою, які дозволять вам відтворювати аудіозаписи, представлено на рисунку 3.22.

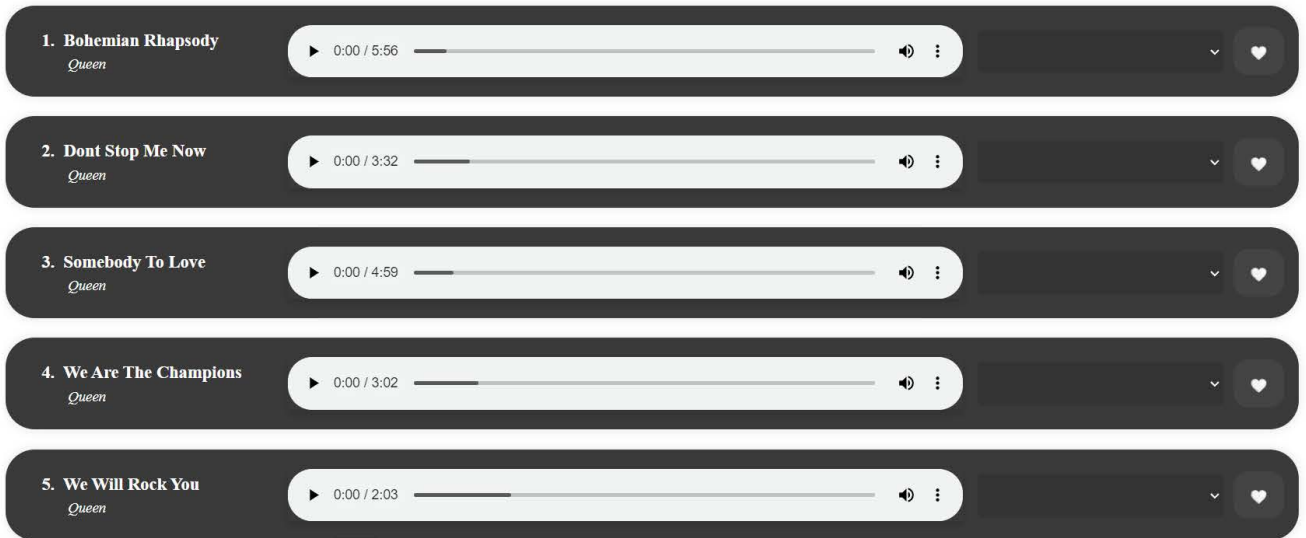


Рисунок 3.22 – Відображення елементів управління музикою

Клацайте на кнопку "Play" (або відповідний піктограму), щоб почати відтворення треку, представлено на рисунках 3.23-3.24.



Рисунок 3.23 – Відображення елементу управління «Пуск»



Рисунок 3.24 – Відображення відтворення треку

Виходьте з облікового запису: якщо ви бажаєте вийти зі свого облікового запису, скористайтеся посиланням "Вийти" або "Logout", яке зазвичай знаходиться у верхньому меню, представлено на рисунку 3.25.

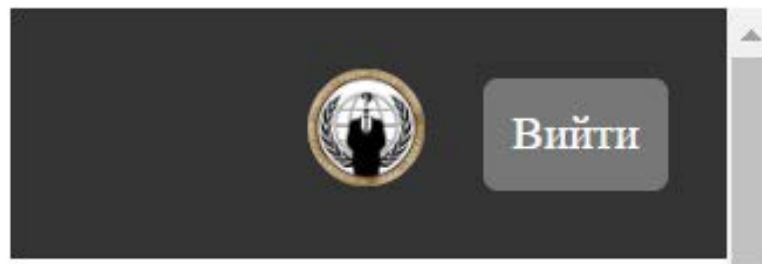


Рисунок 3.25 – Відображення кнопки «Вийти»

Після виходу з облікового запису ви будете перенаправлені на головну сторінку.

3.4 Висновки до третього розділу

У даному розділі були представлені результати розробки веб-сайту та представлення коду маршрутизації.

Було описано функціональність та основні особливості розробленого веб-сайту. Були реалізовані задумані функції, враховані вимоги, що були поставлені в постановці задачі. Веб-сайт демонструє зручний інтерфейс користувача, який дозволяє зручно виконувати необхідні дії та отримувати потрібну інформацію. Результат розробки веб-сайту є підтвердженням успішної реалізації поставлених завдань.

Також була представлена інформація про реалізацію маршрутизації веб-сайту. Були описані алгоритми та методи, використані для написання коду маршрутизації, який забезпечує перехід між сторінками та взаємодію з користувачем. Представлений код є чітким, структурованим та ефективним з точки зору роботи веб-сайту.

ВИСНОВКИ

Проведене дослідження та розробка веб-застосунку для розповсюдження та зберігання цифрової звукової інформації принесли значні результати і виявилися досить успішними.

Під час виконання кваліфікаційної роботи було досягнуто мету та виконані завдання кваліфікаційної роботи: проаналізовано предметну область; було проведено аналіз існуючих рішень в сфері збереження та відтворення музичного контенту; сформульовано постановку завдання; розроблена база даних для роботи веб-додатку; побудовано концептуальну модель бази даних; побудовано логічну модель бази даних; розроблено фізичну модель бази даних; розроблено макет веб-сайту; сформульоване технічне завдання та етапи розробки; розроблено інтерфейс користувача; створено інструкцію користувача; протестовано створений програмний продукт та надано можливі сценарії його використання.

Проведений аналіз предметної області дозволив отримати загальне уявлення про характеристики, особливості та основні проблеми, пов'язані з даною областю. Було виявлено ключові аспекти, які необхідно враховувати при вирішенні задачі, а також виявлені основні напрямки досліджень і розвитку в даній області.

Був проведений аналіз існуючих рішень, зокрема розглянуто два популярних сервіси: Apple Music і YouTube Music. Були проаналізовані їх основні характеристики, переваги та недоліки, функціональні можливості та специфіка використання. Цей аналіз дав змогу виявити прогалини і недоліки існуючих рішень, які можуть бути враховані при розробці нового рішення.

Була поставлена задача, яка є основою для подальшого дослідження та розробки нового рішення. Враховуючи результати аналізу предметної області та

існуючих рішень, були визначені основні вимоги та цілі, які повинні бути виконані в новому рішенні.

Була проведена робота над створенням бази даних для підтримки функціональності веб-сайту. Спочатку була побудована концептуальна модель, яка дозволила визначити основні сутності та взаємозв'язки між ними. На основі концептуальної моделі була розроблена логічна модель бази даних, яка визначила структуру таблиць та зв'язки між ними. Крім того, була розроблена фізична модель бази даних, в якій були визначені типи даних, індекси, обмеження та інші параметри, необхідні для її реалізації.

Було створено візуальне представлення веб-сайту у вигляді макету. Застосовано сучасні інструменти для розробки інтерфейсу користувача, включаючи графічні елементи, кольорову схему, компоненти та розташування елементів на сторінці. Макет був створений з урахуванням вимог функціональності та зручності використання веб-сайту.

Результатом розробки є веб-сайт з інтуїтивно зрозумілим інтерфейсом, який забезпечує зручну навігацію та взаємодію з музичним контентом.

Під час дослідження та розробки веб-застосунку були використані такі інструменти і технології:

- мова програмування JavaScript для розробки функціональності та взаємодії з користувачем;
- HTML та CSS для створення інтерфейсу та дизайну веб-застосунку;
- мова програмування PHP для реалізації серверної частини та взаємодії з базою даних;
- база даних MySQL для зберігання музичного контенту та метаданих.

Використання PHP, JavaScript, HTML та CSS дозволило реалізувати функціональність завантаження, зберігання, стрімінгу, пошуку та сортування музичного контенту. Оцінка результатів розробки підтверджує ефективність та

потенціал веб-застосунку для розповсюдження та зберігання цифрової звукової інформації.

Було описано функціональність та основні особливості розробленого веб-сайту. Були реалізовані задумані функції, враховані вимоги, що були поставлені в постановці задачі. Веб-сайт демонструє зручний інтерфейс користувача, який дозволяє зручно виконувати необхідні дії та отримувати потрібну інформацію. Результат розробки веб-сайту є підтвердженням успішної реалізації поставлених завдань.

Також була представлена інформація про реалізацію маршрутизації веб-сайту. Були описані алгоритми та методи, використані для написання коду маршрутизації, який забезпечує перехід між сторінками та взаємодію з користувачем. Представлений код є чітким, структурованим та ефективним з точки зору роботи веб-сайту.

Незважаючи на успішний результат, існують можливості для подальшого удосконалення. Розширення функціональності, оптимізація продуктивності та підтримка більш широкого спектру платформ і веб-браузерів є потенційними шляхами для поліпшення веб-застосунку у майбутньому. Загальною висновком є те, що результати дослідження та розробки підтверджують ефективність веб-застосунку та його потенціал для задоволення потреб користувачів у розповсюдженні та зберіганні цифрової звукової інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сайт YouTube Music. URL: <https://music.youtube.com>.
2. Сайт Apple Music. URL: <https://music.apple.com/ua/browse>.
3. MySQL. URL: <https://www.mysql.com/>.
4. OpenServer. URL: <https://ospanel.io/>.
5. PhpMyAdmin. URL: <https://www.phpmyadmin.net/>.
6. Figma. URL: <https://www.figma.com/>.
7. Сайт з інформацією про можливості ErwinDM. URL: https://ua.bmstu.wiki/Erwin_Data_Modeler.
8. Сайт з корисною інформацією про мову програмування php. URL: http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/43_PHP/index.html.
9. Сайт з корисною інформацією щодо мови програмування JavaScript. URL: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>.
10. Сайт Codecademy – Learn PHP. URL: <https://www.codecademy.com/learn/learn-php>.
11. Сайт Udemy – PHP for Beginners – Become a PHP Master – CMS Project. URL: <https://www.udemy.com/course/php-for-complete-beginners-includes-mysql-object-oriented/>.
12. Сайт PHP.net – PHP Manual. URL: <https://www.php.net/manual/en/index.php>.
13. Сайт Laracasts – PHP Practitioner. URL: <https://laracasts.com/series/php-for-beginners>.
14. Сайт Coursera – Web Applications for Everybody: PHP, SQL, and JavaScript. URL: <https://www.coursera.org/specializations/web-applications>.
15. Сайт Codecademy – JavaScript. URL: <https://www.codecademy.com/learn/introduction-to-javascript>.

16. Сайт FreeCodeCamp – JavaScript Algorithms and Data Structures Certification. URL: <https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/>.

17. Сайт Udemy – The Complete JavaScript Course 2021: From Zero to Expert! URL: <https://www.udemy.com/course/the-complete-javascript-course/>.

18. Сайт MDN Web Docs – JavaScript Guide. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.

19. Сайт Pluralsight – JavaScript Fundamentals. URL: <https://www.pluralsight.com/courses/javascript-fundamentals-2553>.

20. Сайт SoloLearn – JavaScript Tutorial. URL: <https://www.sololearn.com/Course/JavaScript/>.

21. Сайт W3Schools – JavaScript Tutorial. URL: <https://www.w3schools.com/js/>.

22. Сайт Codecademy. URL: <https://www.codecademy.com/learn/learn-sql>.

23. Сайт Udemy. URL: <https://www.udemy.com/topic/mysql/>.

24. Сайт SQLZoo. URL: <https://sqlzoo.net/>.

25. Сайт W3Schools. URL: <https://www.w3schools.com/sql/>.

26. Сайт MySQL Tutorial. URL: <https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/>.

27. Сайт FreeCodeCamp. URL: <https://www.freecodecamp.org/>.

28. Сайт Udemy. URL: <https://www.udemy.com/topic/frontend-web-development/>.

29. Сайт Coursera. URL: <https://www.coursera.org/courses?query=frontend%20development>.

30. Сайт MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer.

31. Канал Traversy Media. URL: <https://www.youtube.com/user/TechGuyWeb>.

32. Канал The Net Ninja. URL: <https://www.youtube.com/c/TheNetNinja>.

33. Канал Dev Ed. URL: <https://www.youtube.com/channel/UC1b90NQQcskPUGDIXsQEz5Q>.
34. Сайт Udemy. URL: <https://www.udemy.com>.
35. Сайт Laravel документація. URL: <https://laravel.com/docs>.
36. Сайт Symfony документація. URL: <https://symfony.com/doc>.
37. Сайт SitePoint. URL: <https://www.sitepoint.com>.
38. Курс "Learn Sublime Text" на Udemy. URL: <https://www.udemy.com/course/sublime-text-3-from-scratch/>.
39. "Sublime Text Power User" на Pluralsight. URL: <https://www.pluralsight.com/courses/sublime-text-3-power-user>.
40. Офіційна документація Sublime Text. URL: <https://www.sublimetext.com/docs>.

ДОДАТОК

Лістинг А1 – Файл index.php.

```
<?php

$conn = new mysqli("localhost", "root", "", "music_database");

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$url = isset($_GET['url']) ? $_GET['url'] : 'home';

switch ($url) {
    case 'home':
        include('includes/header.php');
        include('pages/home.php');
        include('includes/footer.php');

        break;
    case 'add-song':
        include('includes/header.php');
        include('pages/add_song.php');
        include('includes/footer.php');

        break;
    case 'login':
        include('includes/header.php');
        include('includes/log-forms.php');
        include('includes/footer.php');

        break;
```

```
case 'register':
    include('includes/header.php');
    include('includes/reg-forms.php');
    include('includes/footer.php');

    break;
case 'profile':
    include('includes/header.php');
    include('pages/profile.php');
    include('includes/footer.php');

    break;
case 'genres':
    include('includes/header.php');
    include('pages/genres.php');
    include('includes/footer.php');

    break;
case 'artists':
    include('includes/header.php');
    include('pages/artists.php');
    include('includes/footer.php');

    break;
default:
    include('includes/header.php');
    include('pages/home.php');
    include('includes/footer.php');
    break;
}
?>
```

Лістинг А2 – Файл home.php

```
<?php
// Підключення до бази даних
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_database";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Помилка підключення до бази даних: " . $conn->connect_error);
}

// Запит для отримання всіх пісень з бази даних
$music_query = "SELECT music.id, music.title, artists.name as artist_name, genres.name as
genre_name, music.path, music.avatar FROM music INNER JOIN artists ON music.artist_id =
artists.id INNER JOIN genres ON music.genre_id = genres.id";
$music_result = $conn->query($music_query);
$user_id = $_SESSION['user_id'];

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="static/css/home.css">
    <title>Document</title>
</head>
<body style="margin-top:8%;">
    <div class="home-content">
```

```

<div class="songs">
  <?php if ($music_result->num_rows > 0) {
    $num_music = 1;
    while ($row = $music_result->fetch_assoc()) {
      $music_id = $row["id"];
    ?>
    <div class="song">
      <div class="song-details">
        <div class="song-info">
          <p class="song-title"><b><?= $num_music ?>. &nbsp;<?= $row["title"] ?></b></p>
          <p class="artist-name"><i><?= $row["artist_name"] ?></i></p>
        </div>
        <div class="audio-player">
          <audio controls><source src="..<?= $row["path"] ?>" type="audio/mpeg"></audio>
        </div>
        <div class="modal-content">
          <form method="post" action="static/php/add_to_playlist.php">
            <input type="hidden" name="music_id" value="<?= $music_id ?>">
            <input type="hidden" name="user_id" value="<?= $user_id ?>">
            <select name="playlist_id">
              <?php
                $playlist_query = "SELECT * FROM playlist WHERE user_id=$user_id";
                $playlist_result = $conn->query($playlist_query);
                if ($playlist_result->num_rows > 0) {
                  while ($playlist_row = $playlist_result->fetch_assoc()) {
                    echo "<option value=\"" . $playlist_row["id"] . "\"> . $playlist_row["name"] .
"</option>";
                  }
                } else {
                  echo "<option value=' disabled>У вас немає створених плейлистів</option>";
                }
              ?>
            </select>

```

```

        <input type="submit" name="add_to_playlist_button" value="Додати до списку">
    </form>
</div>

</div>
</div>
<?php
    $num_music++;
}
} else {
?>
    <div class="no-songs">Немає наявних пісень</div>
<?php
}
?>
</div>
</div>

<script src="static/js/main.js"></script>
</body>
</html>

```

ЛІСТИНГ А3 – Файл profile.php

```

<?php
$mysqli = new mysqli("localhost", "root", "", "music_database");

$query = "SELECT * FROM user WHERE id = ?";
$stmt = $mysqli->prepare($query);
$stmt->bind_param("i", $_SESSION['user_id']);
$stmt->execute();
$result = $stmt->get_result();
$user = $result->fetch_assoc();
$stmt->close();

```

?>

```
<link rel="stylesheet" href="static/css/profile.css">
```

```
<body>
```

```
<div class="container">
```

```
<div class="profile">
```

```

```

```
<h2 class="profile-username"> <?= $user['username'] ?></h2>
```

```
<p class="profile-email">Пошта: <?= $user['email'] ?></p>
```

```
<p class="profile-date">Дата реєстрації: <?= $user['created_at'] ?></p>
```

```
</div>
```

```
<div class="modal-content-2">
```

```
<h3>Додати плейліст</h3>
```

```
<form class="create_playlist_form" method="post" id="create_playlist_form"
style="display:block">
```

```
<input type="hidden" name="user_id" value="<?= $_SESSION['user_id'] ?>">
```

```
<input type="text" name="playlist_name" placeholder="Назва нового плейліста" required>
```

```
<input type="submit" name="create_playlist" value="Створити плейліст">
```

```
</form>
```

```
</div>
```

```
<div class="playlists">
```

```
<h3>Список плейлістів користувача:</h3>
```

```
<?php
```

```
$user_id = $_SESSION['user_id'];
```

```
$playlists_query = "SELECT * FROM playlist WHERE user_id = ?";
```

```
$playlists_stmt = $mysqli->prepare($playlists_query);
```

```
$playlists_stmt->bind_param("i", $user_id);
```

```
$playlists_stmt->execute();
```

```
$playlists_result = $playlists_stmt->get_result();
```

```
while ($playlist = $playlists_result->fetch_assoc()) {
```

```
    $playlist_id = $playlist['id'];
```

```

$playlist_name = $playlist['name'];
?>
<div class="playlist">
  <h4><?= $playlist_name ?></h4>
  <?php
    $music_query = "SELECT music.* FROM music INNER JOIN playlist_music ON music.id =
playlist_music.music_id WHERE playlist_music.playlist_id = ?";
    $music_stmt = $mysqli->prepare($music_query);
    $music_stmt->bind_param("i", $playlist_id);
    $music_stmt->execute();
    $music_result = $music_stmt->get_result();
    $num_music = 1;
  ?>
  <div class="songs">
    <?php if ($music_result->num_rows > 0) {
      $num_music = 1;
      while ($row = $music_result->fetch_assoc()) {
        $music_id = $row["id"];
      ?>
      <div class="song">
        <div class="song-details">
          <div class="song-info">
            <p class="song-title"><b><?= $num_music ?>. &nbsp;<?= $row["title"] ?></b></p>
            <p class="artist-name"><i><?= $row["artist_name"] ?></i></p>
          </div>
          <div class="audio-player">
            <audio controls><source src=".<?= $row["path"] ?>" type="audio/mpeg"></audio>
          </div>
          <form method="post">
            <input type="hidden" name="music_id" value="<?= $row["id"] ?>">
            <input type="hidden" name="user_id" value="<?= $user_id ?>">
            <input type="hidden" name="playlist_id" value="<?= $playlist_id ?>">

```

```

        <button class="delete-button" type="submit" name="delete_music_id" value="<?=$music_id ?>" - </button>
    </form>
</div>
</div>
<?php $num_music++;
}
} else { ?>
    <div class="no-songs">Нема наявних пісень.</div>
    <?php } ?>
</div>
</div>
<?php } ?>
</div>
</div>
</body>

```

```

<?php
if (isset($_POST['delete_music_id'])) {
    $delete_music_id = $_POST['delete_music_id'];
    $delete_user_id = $_POST['user_id'];
    $delete_playlist_id = $_POST['playlist_id'];

    $delete_query = "DELETE FROM playlist_music WHERE music_id = ? AND playlist_id = ? AND
playlist_id IN (SELECT id FROM playlist WHERE user_id = ?)";
    $delete_stmt = $mysqli->prepare($delete_query);
    $delete_stmt->bind_param("iii", $delete_music_id, $delete_playlist_id, $delete_user_id);
    $delete_stmt->execute();
    $delete_stmt->close();
}
if (isset($_POST["create_playlist"])) {
    $playlist_name = $_POST["playlist_name"];
    $user_id = $_POST["user_id"];

```

```

// Валідація даних
if (!empty($playlist_name)) {
    $sexisting_query = "SELECT * FROM playlist WHERE name = ? AND user_id = ?";
    $sexisting_stmt = $mysqli->prepare($sexisting_query);
    $sexisting_stmt->bind_param("si", $playlist_name, $user_id);
    $sexisting_stmt->execute();
    $sexisting_result = $sexisting_stmt->get_result();
    if ($sexisting_result->num_rows > 0) {
        echo "<script>alert('Плейліст з такою назвою уже існує');</script>";
    } else {
        // Додавання плейлиста до бази даних
        $screate_query = "INSERT INTO playlist (name, user_id) VALUES (?, ?)";
        $screate_stmt = $mysqli->prepare($screate_query);
        $screate_stmt->bind_param("si", $playlist_name, $user_id);
        if ($screate_stmt->execute()) {
            echo "<script>alert('Плейліст успішно створено');</script>";
            header("Location: profile.php"); // Перенаправлення на профіль після створення плейлиста
            exit();
        } else {
            echo "<script>alert('Помилка при створенні плейлиста: " . $mysqli->error . "');</script>";
        }
    }
    $sexisting_stmt->close();
    $screate_stmt->close();
} else {
    echo "<script>alert('Назва плейлиста не може бути порожньою');</script>";
}
}

$mysqli->close();
?>

```

Лістинг А4 – Файл genres.php

```
<?php
$mysqli = new mysqli("localhost", "root", "", "music_database");

// Отримати список жанрів
$genres_query = "SELECT * FROM genres";
$genres_result = $mysqli->query($genres_query);
?>

<!DOCTYPE html>
<html>
<head>
    <title>Сторінка жанрів</title>
    <link rel="stylesheet" href="static/css/genres.css">
</head>
<body>
    <div class="container">
        <?php while ($genre = $genres_result->fetch_assoc()) { ?>
            <div class="genre">
                <h3><?= $genre['name'] ?></h3>
                <?php
                    // Отримати пісні для даного жанру
                    $genre_id = $genre['id'];
                    $songs_query = "SELECT music.* FROM music
                                INNER JOIN artists ON music.artist_id = artists.id
                                INNER JOIN genres ON music.genre_id = genres.id
                                WHERE genres.id = ?";
                    $songs_stmt = $mysqli->prepare($songs_query);
                    $songs_stmt->bind_param("i", $genre_id);
                    $songs_stmt->execute();
                    $songs_result = $songs_stmt->get_result();
                    ?>
                <div class="songs">
```

```

<?php if ($songs_result->num_rows > 0) {
    while ($song = $songs_result->fetch_assoc()) {
        ?>
        <div class="song">
            <div class="song-details">
                <div class="song-info">
                    <p class="song-title"><b><?= $song["title"] ?></b></p>
                    <p class="artist-name"><i><?= $song["artist_name"] ?></i></p>
                </div>
                <div class="audio-player">
                    <audio controls><source src="<?= $song["path"] ?>" type="audio/mpeg"></audio>
                </div>
            </div>
        </div>
        <?php
        }
        } else {
            echo "<div class='no-songs'>Нема наявних пісень для цього жанру.</div>";
        }
        ?>
    </div>
</div>
<?php } ?>
</div>
</body>
</html>

<?php
$mysqli->close();
?>

Лістинг А5 – Файл add_song.php

<?php
$servername = "localhost";

```

```

$username = "root";
$password = "";
$dbname = "music_database";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$artist_query = "SELECT * FROM artists";
$artist_result = $conn->query($artist_query);

$genre_query = "SELECT * FROM genres";
$genre_result = $conn->query($genre_query);

?>
<!DOCTYPE html>
<html>
<head>
    <title>Додати пісню</title>
    <link rel="stylesheet" href="static/css/add_song.css">
</head>
<body class="add-song-file">
    <h1 style="text-align: center;">Додати пісню</h1>
    <form class="add-song-form" method="post" action="static/php/add-song-script.php">
        <label>Назва пісні:</label><br>
        <input type="text" name="title"><br><br>
        <label>Виконавець:</label><br>
        <select name="artist_id">
            <?php
            if ($artist_result->num_rows > 0) {
                while ($row = $artist_result->fetch_assoc()) {
                    echo "<option value='" . $row["id"] . "'>" . $row["name"] . "</option>";

```

```

    }
}
?>
</select><br><br>
<label>Жанр:</label><br>
<select name="genre_id">
    <?php
    if ($genre_result->num_rows > 0) {
        while ($row = $genre_result->fetch_assoc()) {
            echo "<option value='" . $row["id"] . "'> " . $row["name"] . "</option>";
        }
    }
    ?>
</select><br><br>
<label>Пшлях до файлу:</label><br>
<input type="text" name="path"><br><br>
<input type="submit" value="Додати псню">
</form>
</body>
</html>

```

Лістинг А6 – Файл add-song-script.php

```

<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_database";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

if($_SERVER["REQUEST_METHOD"] == "POST") {
    $title = $_POST["title"];
    $artist_id = $_POST["artist_id"];
    $genre_id = $_POST["genre_id"];
    $path = addslashes($_POST["path"]);

    $song_query = "INSERT INTO `music` (title, artist_id, genre_id, path) VALUES ('$title',
    $artist_id, $genre_id, '$path')";

    if($conn->query($song_query) === TRUE) {

        header("Location: /");
        exit();
    } else {

        echo "Error: " . $song_query . "<br>" . $conn->error;
    }
}

$conn->close();
?>

```

ЛІСТИНГ А7 – Файл add_to_playlist.php

```

<?php
if (isset($_POST["add_to_playlist_button"])) {
    $music_id = $_POST["music_id"];
    $user_id = $_POST["user_id"];
    $playlist_id = $_POST["playlist_id"];

    $servername = "localhost";
    $username = "root";
    $password = "";

```

```

$dbname = "music_database";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$check_query = "SELECT * FROM playlist_music WHERE playlist_id = $playlist_id AND
music_id = $music_id";
$check_result = $conn->query($check_query);
if ($check_result->num_rows > 0) {
    echo ("<script>alert('Пісня вже додана до цього плейліста');</script>");
} else {
    $sql = "INSERT INTO playlist_music (playlist_id, music_id) VALUES ('$playlist_id',
'$music_id')";
    if ($conn->query($sql) === TRUE) {
        header("Location: /");
        exit();
    } else {
        echo "Помилка при додаванні пісні до плейліста: " . $conn->error;
        header("Location: /");
        exit();
    }
}
}

?>

```

Лістинг А8 – Файл login.php

```

<?php
session_start();

if (isset($_POST['login'])) {

```

```

$username = $_POST['username'];
$password = $_POST['password'];
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "music_database";
$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT * FROM user WHERE username='$username' AND password='$password'
LIMIT 1";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) == 1)
    $row = mysqli_fetch_assoc($result);
    $user_id = $row['id'];
    $username = $row['username'];
    $email = $row['email'];
    $avatar = $row['avatar'];
    $_SESSION['user_id'] = $user_id;
    $_SESSION['username'] = $username;
    $_SESSION['email'] = $email;
    $_SESSION['avatar'] = $avatar;
    header("Location: /");
    exit;
} else {
    echo "Неверное имя пользователя или пароль";
}

mysqli_close($conn);
}

```

Лістинг А8 – Файл register.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "music_database";
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if (isset($_POST['register'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $email = $_POST['email'];
    $avatar = $_FILES['avatar']['name'];
    $created_at = date("Y-m-d H:i:s");

    $target_dir = "uploads/";
    $target_file = $target_dir . basename($_FILES["avatar"]["name"]);
    $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
    $extensions_arr = array("jpg", "jpeg", "png", "gif");
    if (in_array($imageFileType, $extensions_arr)) {
        move_uploaded_file($_FILES['avatar']['tmp_name'], $target_dir.$avatar);
    }

    $sql = "INSERT INTO user (username, password, email, avatar, created_at)
        VALUES ('$username', '$password', '$email', '$avatar', '$created_at')";
    if ($conn->query($sql) === TRUE) {
        header("Location: /");
    }
}
```

```

    } else {
        echo "Ошибка: " . $sql . "<br>" . $conn->error;
    }

```

```

    $conn->close();
}
?>

```

Лістинг А9 – Файл exit.php

```

<?php
session_start();

if (isset($_POST['logout'])) {

    session_unset();
    session_destroy();

    header("Location: /home");
    exit();
}
?>

```