

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ ПІДБОРУ МУЗИЧНИХ ТВОРІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
MUSIC SELECTION TOOLS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-133М
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ВІЛЬХОВСЬКИЙ А.В.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ВІЛЬХОВСЬКОГО Андрія Валентиновича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів підбору музичних творів. Research and Software Implementation of Music Selection Tools

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 07 ” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “9” вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Андрій ВІЛЬХОВСЬКИЙ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Михайло КОЦУР
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
86 с., 3 табл., 33 рис., 3 дод., 19 джерел.

PYCHARM, PYTHON, МОВА ПРОГРАМУВАННЯ, МУЗИЧНИЙ
ТВІР, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СЕРЕДОВИЩЕ РОЗРОБКИ.

Об'єкт дослідження – процес розробки програмного забезпечення для роботи з музичними творами.

Предмет дослідження – програмні засоби для підбору музичних творів.

Мета роботи – дослідження та розробка програмного забезпечення для підбору музичних творів.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки PyCharm.

Результати. Розроблено програмне забезпечення для підбору музичних творів за допомогою мови програмування Python та середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного забезпечення, що забезпечує підтримку процесу підбору музичних творів.

Висновки. Виконано проєктування програмного забезпечення для підбору музичних творів. Розроблено програмне забезпечення для підбору музичних творів. Здійснено тестування розробленого програмного забезпечення для підбору музичних творів.

Галузь використання – музичні програвачі.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 86 pages, 3 tables, 33 figures, 3 appendixes, 19 sources.

PYCHARM, PYTHON, PROGRAMMING LANGUAGE, MUSIC, SOFTWARE, DEVELOPMENT ENVIRONMENT.

The object of research is the process of developing software for working with musical works.

The subject of the research is software for selecting musical works.

The purpose of this work is to research and develop the software for the selection of musical works.

Materials, methods and technical tools: Python programming language, PyCharm development environment.

Results. The software tools for selection of musical works using the Python programming language and the PyCharm development environment have been developed.

The practical value of the work lies in the development of software for selection of musical works.

Conclusions. The software for selection of musical works has been designed. The software for selection of musical works has been developed. The developed software for selection of musical works has been tested.

Scope of use – music players.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для підбору музичних творів	11
1.2 Аналіз програмного забезпечення для підбору музичних творів	21
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування	33
2.2 Вибір середовища розробки для створення програмного забезпечення для підбору музичних творів	35
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для підбору музичних творів	37
2.4 Висновки за розділом 2	39
3 Опис програми	40
3.1 Структура програмного забезпечення для підбору музичних творів	40
3.2 Функціонування програмного забезпечення для підбору музичних творів	42
3.3 Розробка бази даних програмного забезпечення для підбору музичних творів	44
3.4 Проєктування інтерфейсу програмного забезпечення для підбору музичних творів	46
3.5 Висновки за розділом 3	50
4 Експлуатація, тестування та експериментальне дослідження програми	52
4.1 Призначення й умови застосування програми	52
4.2 Характеристики програми для підбору музичних творів	53
4.3 Інструкція по експлуатації програми	54
4.3.1 Звернення до програми	54
4.3.2 Вхідні й вихідні дані	54

4.3.3 Повідомлення	54
4.4 Виконання програмного забезпечення для підбору музичних творів	54
4.5 Тестування програмного забезпечення для підбору музичних творів....	59
4.6 Висновки за розділом 4	60
Висновки	61
Перелік джерел посилань	64
Додаток А Технічне завдання	66
Додаток Б Фрагмент тексту програми	70
Додаток В Слайди презентації	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – Application Programming Interface;

MVC – Model-View-Controller;

БД – база даних;

МТ – музичний твір;

ПЗ – програмне забезпечення;

ШІ – штучний інтелект.

ВСТУП

Актуальність програмного забезпечення для підбору музичних творів продовжує зростати завдяки розвитку штучного інтелекту, змінам у поведінці користувачів та широкому впровадженню потокових сервісів [1]-[7].

Сучасні музичні стримінгові платформи надають користувачам доступ до мільйонів музичних творів (МТ), що суттєво ускладнює вибір і пошук потрібних композицій, що відповідають індивідуальним уподобанням користувачів. Програмне забезпечення для підбору музики, яке використовує алгоритми машинного навчання та аналіз даних, допомагає користувачам знаходити відповідний контент на основі їхніх слухацьких звичок [1]-[7].

Використання штучного інтелекту (ШІ) дозволяє враховувати не лише жанри та стилі, а й емоційний контекст, ритм, темп і навіть поведінку користувача. Такі програми є корисними не лише для слухачів, а й для музичних кураторів, діджеїв, і навіть композиторів. Програмні засоби для підбору МТ забезпечують автоматичне створення плейлистів, що дозволяє економити час і швидше знаходити відповідні композиції для певних подій чи проєктів [1]-[7].

Враховуючи зростання популярності музики з різних регіонів, програмне забезпечення для підбору МТ здатне рекомендувати локальні та міжнародні композиції, адаптуючи контент до різноманітних культурних смаків [1]-[7].

Проте деякі програми для підбору музичних творів можуть не завжди точно відповідати музичним вподобанням користувачів, оскільки персоналізовані рекомендації часто базуються лише на обмеженому наборі параметрів, як-от жанр або ритм, без урахування контексту або настрою. Такі рекомендації можуть не завжди відповідати емоційним чи культурним потребам користувачів. Іншим недоліком такого програмного забезпечення є те, що часто пропонуються популярні або трендові композиції, що обмежує доступ до менш відомих чи незалежних виконавців. Деякі програми для

підбору музики збирають великі обсяги даних про вподобання користувачів, включаючи їхню музичну історію, час прослуховування та навіть інформацію про місцезнаходження, що може викликати занепокоєння щодо конфіденційності даних, оскільки існує ризик їх неналежного використання або витоку. Деякі користувачі програм для підбору музичних творів можуть мати проблеми з навігацією або взаємодією з інтерфейсом програм, оскільки не всі програми мають інтуїтивно зрозумілі інтерфейси або дозволяють легко налаштувати власні плейлисти та підборки МТ [1]-[7].

Тому актуальною є розробка програмного забезпечення для підбору музичних творів. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та розробки програмного забезпечення для підбору музичних творів.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підбору музичних творів;
- розробити модель взаємодії користувача з програмним забезпеченням для підбору музичних творів;
- здійснити проєктування програмного забезпечення для підбору музичних творів;
- створити програмне забезпечення для підбору музичних творів;
- виконати тестування розробленого програмного забезпечення для підбору музичних творів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для підбору музичних творів

Технологія рекомендацій музичних творів являє собою систему, яка допомагає пропонувати користувачам пісні, альбоми або виконавців, спираючись на їхні вподобання та історію прослуховування [1]-[7]. Технологія рекомендацій музичних творів використовує алгоритми для аналізу дій користувача, зокрема того, що він слухав, читав чи пропускав, щоб краще розуміти його музичні смаки. Завдяки цим даним програмна система може рекомендувати нові треки, які, ймовірно, зацікавлять користувача [1]-[7].

Програмне забезпечення для підбору музичних творів використовує пісні, що прослухав користувач, та додані до бібліотеки композиції для формування персоналізованих плейлистів та рекомендацій [1]-[7].

Іноді це можуть бути виконавці, а іноді вже знайомі користувачам альбоми [1]-[7].

Основна мета програмних систем для підбору музичних творів полягає у покращенні процесу відкриття нової музики, роблячи його зручним і цікавим [1]-[7].

Програмне забезпечення для підбору музичних творів працює, аналізуючи різні типи даних, щоб підібрати вам улюблені пісні або виконавців. Таке програмне забезпечення використовує різні технології для створення рекомендацій, об'єднуючи дані про поведінку користувачів, музичні патерни та навіть складний аналіз треків, щоб зробити рекомендації максимально індивідуальними. Основною метою є забезпечення плавного та персоналізованого музичного досвіду [1]-[7].

Проаналізуємо компоненти програмного забезпечення для підбору музичних творів, зокрема, спільне фільтрування, контентне фільтрування та розширені моделі оцінки [1]-[7].

Спільне фільтрування є одним з найпоширеніших підходів до надання музичних рекомендацій, який полягає в порівнянні уподобань конкретного

користувача з уподобаннями іншими користувачами. Програмна система аналізує історію прослуховування різних людей, підбираючи користувачів зі схожими смаками, та пропонує вам пісні, які сподобалися іншим, але ви їх ще не чули. Наприклад, якщо декілька людей з подібними музичними уподобаннями полюбили певного виконавця, відповідному користувачу теж його порекомендують. Однак цей підхід може давати передбачувані або повторювані рекомендації, якщо смаки користувача трохи відрізняються від основної аудиторії [1]-[7].

Фільтрація на основі контенту фокусується на характеристиках самих пісень, аналізуючи схожість між треками у вашій колекції та новими піснями. Програмні системи використовують такі дані для рекомендацій: жанр (визначення музичних стилів для підбору подібних пісень), темп (підтримка ритму або швидкості треків), інструменти (аналіз інструментів, що використовуються в композиціях), структура і гармонія (розуміння композиційної будови та гармонійних зв'язків), тексти (вивчення тем, настроїв та емоцій, переданих у піснях), вокал (оцінка вокалу за тембром, висотою та іншими характеристиками) [1]-[7].

Цей підхід дозволяє створювати рекомендації, адаптовані до уподобань користувача, хоча іноді важко запропонувати щось кардинально нове, що виходить за рамки історії прослуховування [1]-[7].

Сучасні моделі рекомендацій піднімають підбір музики на новий рівень, застосовуючи розумні алгоритми та методи машинного навчання для глибшого аналізу композицій і покращення рекомендацій [1]-[7].

Програмне забезпечення для підбору музичних творів досліджує музичну структуру, розглядаючи не тільки жанр або звук, але й деталі, такі як музичний розвиток [1]-[7].

Завдяки цьому підходу, програмна система для підбору музичних творів може пропонувати музику, яка відповідає уподобанням користувача. Для покращення рекомендацій програмне забезпечення для підбору музичних творів можуть використовувати додатково такі дані: аналіз звукових частот,

щоб зрозуміти варіації звучання у треках, оцінка ритмічних складностей та ритму пісні, вивчення частотних трендів і гармонійних переходів у композиції, оцінка тембру для аналізу якості звуку, його яскравості або тепла, аналіз змін висоти тону і динаміки звуку, оцінка емоційної насиченості та енергії музичних творів [1]-[7].

Цей великий обсяг інформації дозволяє створювати максимально персоналізовані рекомендації, налаштовані під індивідуальні смаки кожного слухача, відкриваючи для нього пісні, які важко було б знайти за допомогою стандартних підходів [1]-[7].

Такі механізми можна впроваджувати як у масштабних проєктах, так і в невеликих програмних засобах, завдяки ентузіазму спеціалістів у сфері музичних рекомендацій. Створивши систему персоналізованих рекомендацій, можна пропонувати музику, адаптовану до смаків користувачів [1]-[7].

Програмне забезпечення для подання рекомендацій у різних сферах часто являє собою інноваційну платформу на основі штучного інтелекту та машинного навчання, яка надає персоналізовані рекомендації для розв'язання практичних завдань, включаючи електронну комерцію, для покращення взаємодії з клієнтами через рекомендації продуктів [1]-[7].

Проаналізуємо процес створення програмного забезпечення для оцінювання персоналізованого музичного досвіду. Першим етапом розробки індивідуальної програмної системи музичних рекомендацій є чітке визначення бізнес-цілей і потреб. Це передбачає розуміння цільової аудиторії, типів рекомендацій, які ви хочете надавати (наприклад, пісні, виконавці чи альбоми), та ключових показників ефективності, які дозволять оцінити успішність проєкту [1]-[7].

На цьому етапі необхідно співпрацювати з ключовими учасниками проєкту для детального обговорення всіх вимог і узгодження очікуваних результатів [1]-[7].

Для ефективного управління проєктом варто використовувати відповідні програмні інструменти, які допоможуть стежити за виконанням

завдань та дотриманням вимог. Це також забезпечить наявність чіткого документа для контролю всіх етапів розробки [1]-[7].

Дані є ключовим компонентом будь-якої системи рекомендацій. Для побудови механізму музичних рекомендацій потрібно збирати інформацію про взаємодію користувачів (наприклад, історія прослуховувань, вподобання, активність), а також музичні метадані (жанр, виконавець, альбом) і технічні характеристики треків (темп, тональність, гучність) [1]-[7].

Після збору дані потрібно очистити, стандартизувати та перевірити на цілісність і придатність до використання [1]-[7]. Технологічний стек: варто використовувати відповідні програмні засоби для обробки даних у реальному часі, зокрема для обробки, також можна застосувати потокові фреймворки розробки програмного забезпечення та обробки даних [1]-[7].

На наступному етапі виконується вибір та навчання моделей. Підбір відповідного алгоритму машинного навчання є ключем до успіху у розробці рекомендаційної програмної системи. Залежно від типу даних і підходу, можна використовувати спільну фільтрацію, фільтрацію за контентом або гібридний метод. Модель повинна бути навчена на історичних даних, щоб точно прогнозувати вподобання користувачів [1]-[7].

Для створення та навчання моделей можна використовувати бібліотеки для машинного навчання, також можна їх інтегрувати для синтезу складніших моделей [1]-[7]. Варто оцінити продуктивність моделей за допомогою таких метрик, як точність, повнота та F1-оцінка, а також застосовувати перехресну перевірку, щоб переконатися, що модель добре працює на нових даних [1]-[7].

Після налаштування моделі необхідно інтегрувати її в існуючу інфраструктуру. Це включає створення прикладного програмного інтерфейсу (Application Programming Interface, API), що дозволить програмному інструменту для надання рекомендацій взаємодіяти з іншими застосунками або платформами [1]-[7].

Ключовим етапом є тестування, зокрема, модульне, інтеграційне та A/B-тестування, щоб гарантувати стабільну роботу системи у різних умовах.

Можна використовувати RESTful API з такими фреймворками Java, як Spring Boot. Для тестування підходять JUnit і Mockito для модульного тестування, а Apache JMeter для навантажувального тестування [1]-[7].

Варто переконатися, що розроблювана програмна система масштабована й здатна обробляти великі навантаження. Для оптимізації часу відповіді можна використовувати відповідні програмні засоби, а також розподілені баз даних для підвищення масштабованості [1]-[7].

Останній етап полягає у введенні розробленої програмної системи в експлуатацію та налагодження постійного моніторингу її роботи. Це дозволяє вчасно виявляти проблеми, контролювати ключові показники ефективності та покращувати продуктивність [1]-[7].

Моніторинг дає змогу виявляти зміни у продуктивності та оперативно повідомляти команду. Регулярно оновлюйте моделі, щоб забезпечити їхню актуальність і точність [1]-[7].

Механізми музичних рекомендацій ідеально підходять для стримінгових платформ [1]-[7].

Як уже зазначалося, персоналізовані системи музичних рекомендацій відіграють важливу роль у взаємодії з користувачами, що є критичним для успіху музичних компаній [1]-[7].

Можливість надавати індивідуальні рекомендації не лише підвищує задоволення користувачів, а й слугує важливим бізнес-показником. Впровадження музичних рекомендаційних систем може суттєво змінити правила гри для стримінгових сервісів, забезпечуючи значні переваги з точки зору залученості користувачів, їх утримання та загального зростання платформи [1]-[7].

Персоналізовані музичні рекомендації значно покращують користувацький досвід, надаючи релевантний контент, який відповідає вподобанням слухачів [1]-[7].

Такі програмні сервіси відзначають суттєве зростання залученості завдяки персоналізованим спискам, що підкреслює важливість таких систем для повної взаємодії користувачів з платформою [1]-[7].

Крім цього, такі інструменти стимулюють користувачів довше залишатися на платформі, досліджуючи нову музику та відкриваючи нових артистів [1]-[7].

Індивідуальний підхід до планування послідовності музичних творів відіграє ключову роль у збереженні користувачів. Коли люди відчувають, що платформа розуміє їхні музичні смаки та постійно пропонує музику, яка їм подобається, вони з більшою ймовірністю залишаються лояльними [1]-[7].

Дослідження свідчать, що платформи з надійними рекомендаційними системами можуть зменшити рівень відтоку на десятки відсотків. Це утримання користувачів є ключовим фактором для стабільного зростання бази користувачів [1]-[7].

Ефективна рекомендаційна система допомагає користувачам відкривати нові треки, які вони могли б пропустити. Це вигідно не тільки користувачам, але й виконавцям, оскільки сприяє зростанню їхньої популярності [1]-[7], допомагаючи користувачам щотижня знайомитися з новими артистами та піснями [1]-[7].

Персоналізовані рекомендації можуть безпосередньо впливати на ефективність онлайн платформ. Рекомендуючи пісні, альбоми або навіть товари, що відповідають уподобанням користувачів, такі системи сприяють збільшенню конверсії. Крім того, платформи, що пропонують контент чи бронювання, можуть використовувати ексклюзивні рекомендації для більш ефективного просування своїх послуг [1]-[7].

Цифрові галузі, як-от соціальні мережі та електронна комерція, що активно впроваджують рекомендаційні системи, можуть суттєво збільшити свої доходи завдяки рекомендаціям контенту та пов'язаним із ним покупками [1]-[7].

Загальне враження від програмної платформи значно поліпшується, коли вона пропонує розумні й точні рекомендації. Користувачі можуть швидко знаходити музику, яка відповідає їх настрою чи потребам, без необхідності довгих пошуків. Такий рівень персоналізації забезпечує комфортну та приємну взаємодію, що стає значною перевагою для маркетологів на перенасиченому ринку [1]-[7].

Впровадження музичних рекомендаційних інструментів також дає цінну інформацію про вподобання та поведінку користувачів. Ці дані можна аналізувати, щоб виявляти тренди, популярні напрями та сегменти аудиторії, що допомагає приймати обґрунтовані рішення щодо контентної стратегії, маркетингу та управління користувацьким досвідом. Такі інсайти є ключовими для покращення пропозицій платформи та утримання лідерських позицій у галузі [1]-[7].

Музика відіграє ключову роль у формуванні та передачі емоцій користувачів, тому техніка емоційного оцінювання для надання рекомендацій є такою важливою. Коли музика гармоніює з емоційним станом слухача, це значно покращує його досвід прослуховування. Музика здатна глибоко впливати на емоційний та психічний стан людини, надаючи відчуття комфорту [1]-[7].

Рекомендаційні системи, що враховують емоції, допомагають відкривати нові пісні та виконавців, які викликають у слухачів емоційний відгук, покращуючи їх музичний досвід. Такі системи стимулюють залученість і лояльність користувачів, забезпечуючи постійно актуальні та цікаві музичні пропозиції. З бізнесової точки зору, це підвищує використання сервісів та збільшує потенційний дохід. Вміння таких систем адаптуватися до змін у музичних смаках робить їх потужним інструментом в умовах сучасної цифрової музичної індустрії [1]-[7].

Рекомендаційні програмні системи дозволяють покращити музичний досвід, інтегруючи емоційні стани у вибір музики. Це особливо важливо, оскільки музика має сильний емоційний вплив, а сучасні системи прийняття

рішень недостатньо враховують емоційний стан слухача. Це може сприяти створенню глибокого емоційного зв'язку через правильний вибір музики. Розвиток музичних рекомендаційних систем тісно пов'язаний з еволюцією цифрової музики. Спочатку ці системи базувалися на двох основних алгоритмах: колаборативній фільтрації та контентній фільтрації. Колаборативна фільтрація прогнозує вподобання користувачів, аналізуючи їхні попередні прослуховування та оцінки, що дозволяє ефективно використовувати колективний досвід. З іншого боку, контентна фільтрація оцінює характеристики самих пісень, такі як висота звуку, ритм і темп, для створення рекомендацій на основі особливостей музичних творів [1]-[7].

Зі зростанням складності вимог користувачів з'являються нові моделі, зокрема ті, що базуються на емоціях і поведінці користувача. Ці вдосконалені моделі пропонують рекомендації, враховуючи поточний досвід або дії користувача, що дозволяє досягти вищого рівня персоналізації. Гібридні підходи, які об'єднують кілька систем, показали кращі результати порівняно з окремими методами. Постійний розвиток рекомендаційних систем свідчить про зусилля щодо поліпшення досвіду користувачів у музичній індустрії, роблячи ці сервіси більш точними та адаптованими до будь-яких смаків і настроїв [1]-[7].

Музичні рекомендації, засновані на настрої, відкривають новий підхід до планування музики, комбінуючи обробку зображень і аналіз настроїв із персоналізованим підбором треків. За допомогою згорткових нейронних мереж визначається емоційний стан користувача, наприклад, гнів, радість, смуток або спокій, шляхом аналізу виразів обличчя на фото чи відео. Такий емоційний підхід є важливим, оскільки він лежить в основі музичної композиції. Унікальність програмної системи полягає в тому, що вона підбирає музику відповідно до спостережуваного настрою, підсилюючи її релевантність та емоційний вплив [1]-[7].

На наступному етапі музичні твори сортуються за емоційними характеристиками, а система контентної фільтрації пропонує рекомендації на

їхній основі. Це гарантує, що підібрана музика відповідає не лише вподобанням користувача, але й його поточному емоційному стану. Програмна система для підбору музичних творів має на меті покращити музичний досвід, пропонуючи композиції, які не тільки зацікавлюють, але й емоційно підходять. Інтегруючи емоційний інтелект у музичні системи, програмна система підбору музичних творів забезпечує глибший, персоналізований музичний досвід [1]-[7].

Згорткові нейронні мережі використовуються для надання рекомендацій щодо музичних творів, застосовуючи методи розпізнавання емоцій слухача на основі аналізу зображень або відео. Згорткові нейронні мережі відомі своєю здатністю обробляти візуальні дані, тому вони ефективно застосовуються для аналізу настроїв, що допомагає визначити емоційний стан користувача і відповідно підбирати музичні композиції [1]-[7].

Архітектура згорткових нейронних мереж складається з кількох ключових компонентів. Першим етапом є згорткові шари, які виділяють просторові особливості зображень. Ці шари обробляють зображення, розкриваючи деталі, які можуть бути корисними для класифікації емоцій. Для цього використовуються фільтри з ядрами, які змінюються у розмірі для захоплення різних рівнів деталізації. Максимальне об'єднання, яке виконується після кожного згорткового шару нейромоделі, зменшує просторові розміри даних, скорочуючи кількість обчислень та концентруючи важливі ознаки [1]-[7].

Щоб запобігти перенавчанню, застосовуються шари випадання, які вимикають випадкові нейрони під час навчання, змушуючи модель вивчати більш стійкі та узагальнені риси. Після виділення функцій дані перетворюються у одновірний масив і проходять через щільний шар нейронів, що допомагає моделі ухвалювати рішення щодо класифікації емоцій [1]-[7].

Остаточним етапом є вихідний шар, який використовує функцію активації softmax для розподілу ймовірностей між можливими емоційними класами. На основі результатів класифікації модель може визначити, чи

перебуває користувач у щасливому, сумному, роздратованому або спокійному стані. Така інформація дозволяє системі автоматично підбирати музичні рекомендації, що відповідають емоційному стану користувача, підвищуючи рівень персоналізації та задоволеності прослуховуванням [1]-[7].

Для надання рекомендацій стосовно музичних творів використовуються алгоритми, що класифікують пісні за різними ознаками та асоціюють їх із певними емоційними станами слухача. Це досягається шляхом навчання моделі, яка аналізує набір даних музичних треків і визначає їх характеристики, такі як темп, мелодійність, ритм і гармонія. Модель, як правило, використовує оптимізатор Адама та функцію втрат на основі категоріальної крос-ентропії, що підходить для багатокласових задач, таких як класифікація музичних творів за чотирма настроями: щасливий, сумний, роздратований або спокійний [1]-[7].

Для навчання моделі застосовується генератор даних, що дозволяє ефективно обробляти великі обсяги інформації, які можуть не поміститися в оперативну пам'ять. Процес навчання проводиться в кілька етапів, використовуючи метод ітераційного навчання, що охоплює весь набір тренувальних даних у кожній епосі. Перевірка результатів відбувається після кожної епохи, що дозволяє оцінювати ефективність роботи моделі та вносити корективи [1]-[7].

Згорткові нейронні мережі виступають основою архітектури системи. Вони використовують згорткові шари для виділення просторових особливостей музичних даних, шари максимального об'єднання для зменшення дискретизації, а також повністю з'єднані шари для забезпечення вищого рівня міркувань [1]-[7].

Окрім цього, для формування рекомендацій використовується фільтрація на основі вмісту. Вона дозволяє вибрати музичний трек, що найкраще відповідає вподобанням користувача та його поточному настрою. Для кластеризації пісень використовується метод К-середніх, а також аналіз

головних компонент для зменшення кількості вимірів і виявлення ключових особливостей музичних творів [1]-[7].

Визначено, що технологія рекомендацій музичних творів являє собою систему, яка допомагає пропонувати користувачам пісні, альбоми або виконавців, спираючись на їхні вподобання та історію прослуховування. Показано, що програмне забезпечення для надання рекомендацій музичних творів використовує алгоритми для аналізу дій користувача, зокрема того, що він слухав, читав чи пропускав, щоб краще розуміти його музичні смаки. Завдяки цим даним програмна система може рекомендувати нові треки, які, ймовірно, зацікавлять користувача. Програмне забезпечення для підбору музичних творів працює, аналізуючи різні типи даних, щоб підібрати вам улюблені пісні або виконавців. Таке програмне забезпечення використовує різні технології для створення рекомендацій, об'єднуючи дані про поведінку користувачів, музичні патерни та навіть складний аналіз треків, щоб зробити рекомендації максимально індивідуальними. Проаналізовано компоненти програмного забезпечення для підбору музичних творів, зокрема, спільне фільтрування, контентне фільтрування та розширені моделі оцінки.

1.2 Аналіз програмного забезпечення для підбору музичних творів

Проаналізуємо програмне забезпечення для підбору музичних творів [8]-[14].

Програмне забезпечення Pandora (рис. 1.1) являє собою музичний сервіс для потокового відтворення, що надає доступ до великої колекції альбомів, виконавців, плейлистів і подкастів. Одна з ключових особливостей Pandora – це система рекомендацій і кураторства, яка дозволяє створювати набори музичних творів, які автоматично налаштовуються під музику користувача. Чим більше він слухає музичних творів, тим точніше відповідні станції підлаштовуються під його вподобання [8], [10], [12].



Рисунок 1.1 – Програмне забезпечення Pandora [12]

Програмне забезпечення Pandora пропонує безкоштовний план і два варіанти платної підписки. Безкоштовна версія надає доступ до багатьох функцій, але переривається рекламними вставками як на настільній, так і на мобільній версіях. Однак, користувач не зможе створювати або ділитися плейлистами, а також завантажувати треки для прослуховування [8], [10], [12].

Десктопний інтерфейс Pandora виглядає менш привабливо в порівнянні з більш елегантними та організованими платформами, такими як Apple Music або Spotify, і справляє враження дещо застарілого [8], [10], [12].

Функція пошуку інтуїтивно зрозуміла, а профіль надає можливість відстежувати кількість оцінок, створених списків відтворення, підписників і користувачів, за якими ведеться спостереження. Через значок шестірні у верхньому правому куті можна відкрити меню налаштувань для підключення до різних сервісів, налаштування сповіщень, вибору якості звуку та інших функцій [8], [10], [12].

Серед можливостей Pandora можна також відзначити можливість ставити оцінки трекам, створювати на основі виконавців "станції" та отримати доступ до популярних пісень через платну підписку. Програмне забезпечення Pandora дозволяє користувачам створювати персоналізовані музичні списки відтворення. Завдяки великій музичній бібліотеці та зручному інтерфейсу легко відкривати нові композиції та налаштовувати списки відповідно до власних вподобань. Сервіс підходить для створення як спокійних, так і енергійних плейлистів [8], [10], [12].

Основними особливостями та характеристиками програмного забезпечення Pandora є такі [8], [10], [12]:

- персоналізовані станції: програмне забезпечення Pandora дозволяє користувачам створювати станції на основі улюблених виконавців, пісень або жанрів. Ці станції автоматично налаштовуються під вподобання користувача, пропонуючи нові композиції, схожі на ті, що він слухає [8], [10], [12];

- можливість оцінювання треків: користувачі можуть ставити позначки "подобається" або "не подобається" пісням, що допомагає алгоритму краще налаштовувати майбутні рекомендації [8], [10], [12];

- фільтрація на основі вподобань: Pandora використовує свої алгоритми для глибокого аналізу музичних характеристик, щоб пропонувати музику, що відповідає смаку користувача [8], [10], [12];

- можливість офлайн прослуховування: платні підписники можуть завантажувати пісні та слухати їх в режимі офлайн через додатки для iOS та Android [8], [10], [12];

- різні рівні підписки: Pandora пропонує безкоштовну версію з рекламою та дві платні підписки, які дають доступ до додаткових функцій, таких як, необмежена кількість пропусків пісень і створення власних плейлистів [8], [10], [12];

- кросплатформність: програмне забезпечення підтримується на різних платформах, включаючи мобільні пристрої (iOS, Android), настільні комп'ютери та інші розумні пристрої [8], [10], [12];

– інтеграція з іншими сервісами: програмне забезпечення Pandora можна підключити до різних систем, зокрема, Alexa і Apple HomePod, що дає можливість керувати відтворенням музики через голосові команди користувача [8], [10], [12];

– рекомендації: програмне забезпечення Pandora надає спеціальні музичні плейлисти, створені редакторами Pandora, та рекомендує нову музику на основі того, що подобається користувачам [8], [10], [12];

– якість звуку: користувачі можуть налаштовувати якість звуку відповідно до своїх потреб [8], [10], [12].

Можливість офлайн прослуховування доступна для всіх платних планів Pandora, але тільки через додатки для iOS та Android. Це може бути недоліком порівняно з іншими сервісами, які дозволяють завантажувати треки безпосередньо на комп'ютер і переносити їх на інші пристрої, якщо вони сумісні [8], [10], [12]. Ще одна проблема полягає в тому, що Pandora недоступна в багатьох інших країнах, перед використанням слід перевірити доступність Pandora у конкретному регіоні [8], [10], [12]. Незважаючи на популярність Pandora, сервіс іноді отримує критику за технічні збої. Деякі користувачі стикаються з проблемами пропуску, пауз або повторного відтворення пісень, особливо при слабкому Інтернет з'єднанні. Pandora працює над виправленням цих недоліків, зберігаючи популярність завдяки персоналізованим функціям та зручності користування [8], [10], [12].

Програмне забезпечення SoundCloud (рис. 1.2) є платформою для потокової передачі музики, яка дозволяє створювати персоналізовані плейлисти та відкривати нові треки з великої бібліотеки музики. Завдяки інтуїтивно зрозумілому інтерфейсу, користувачам легко шукати, впорядковувати та прослуховувати обрані композиції [8]-[11], [13].

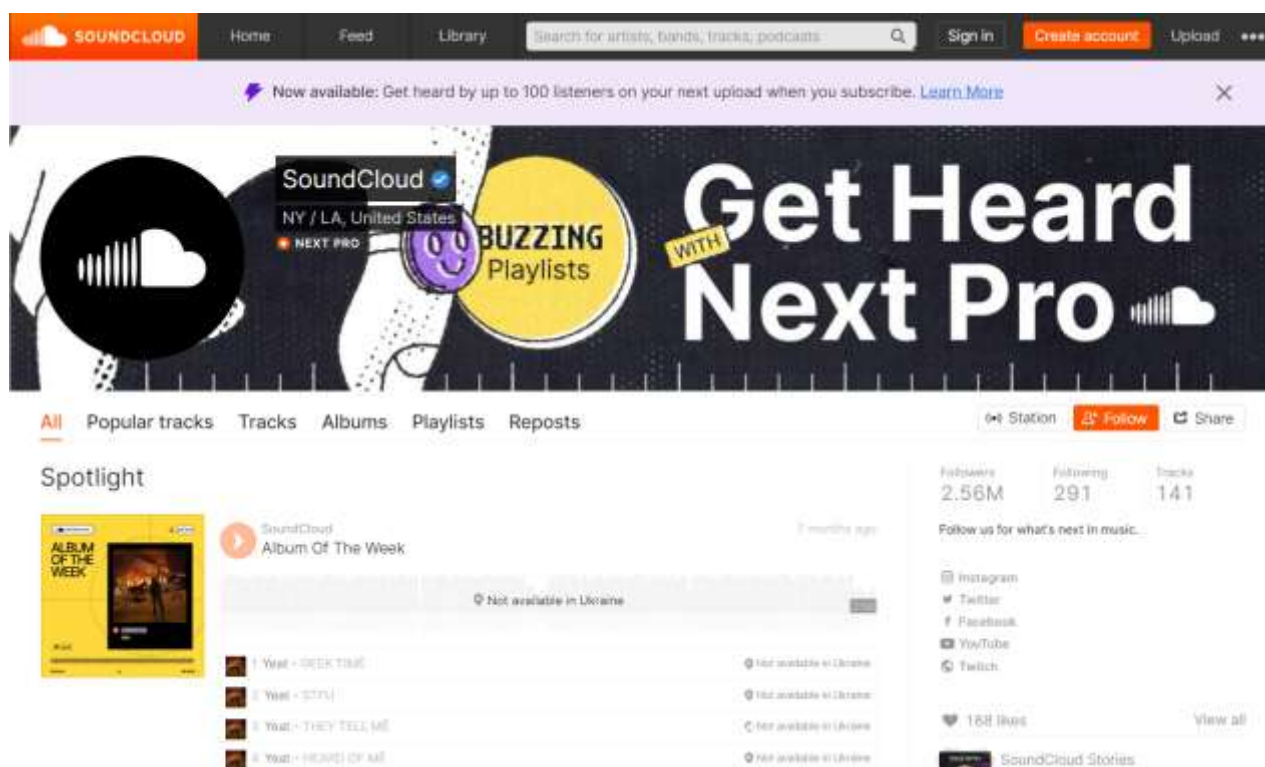


Рисунок 1.2 – Програмне забезпечення SoundCloud [13]

Користувачам програмного забезпечення SoundCloud доступне офлайн-прослуховування, що дає змогу насолоджуватися музикою без необхідності підключення до Інтернету. SoundCloud також підтримує можливість завантаження власних музичних треків, дозволяючи музикантам і творцям просувати свої роботи на платформі [8]-[11], [13].

Особливістю SoundCloud є те, що його можна використовувати безкоштовно, хоча безкоштовна версія містить рекламні вставки. Платна підписка дозволяє отримати додаткові переваги, такі як офлайн-прослуховування та відсутність реклами. Користувачі також можуть слідкувати за своїми улюбленими виконавцями, отримуючи оновлення про нові релізи через персоналізовану стрічку в додатку [8]-[11], [13].

Програмне забезпечення SoundCloud має такі особливості [8]-[11], [13]:

– велика бібліотека музичних творів: Програмне забезпечення SoundCloud надає доступ до мільйонів треків, серед яких можна знайти як відомих виконавців, так і незалежних музикантів [8]-[11], [13];

– персоналізовані плейлисти: користувачі програмного забезпечення SoundCloud можуть створювати власні списки відтворення та налаштовувати їх під свій смак [8]-[11], [13];

– офлайн прослуховування: у платній версії програмного забезпечення SoundCloud можна завантажувати треки для прослуховування без підключення до Інтернету [8]-[11], [13];

– завантаження власних треків: музиканти та творці контенту можуть завантажувати свою музику та поширювати її серед користувачів [8]-[11], [13];

– інтуїтивно зрозумілий інтерфейс: простота навігації та використання для створення плейлистів і пошуку треків [8]-[11], [13];

– безкоштовний доступ з рекламою: SoundCloud пропонує безкоштовний варіант користування з періодичними рекламними вставками. Для користувачів доступний платний варіант, що надає додаткові функції, такі як офлайн-доступ та відсутність реклами [8]-[11], [13];

– персоналізована стрічка: користувачі можуть слідкувати за своїми улюбленими виконавцями та отримувати оновлення про нові релізи через стрічку у додатку [8]-[11], [13];

– спільнота творців: платформа активно використовується музикантами для просування своїх робіт, що робить її важливим майданчиком для нових талантів [8]-[11], [13];

– кросплатформеність: SoundCloud підтримує різні операційні системи, що дозволяє користуватися ним на різних пристроях, включаючи смартфони та персональні комп'ютери [8]-[11], [13].

Серед недоліків програмного забезпечення SoundCloud можна відзначити те, що безкоштовна версія містить велику кількість реклами, яка може заважати прослуховуванню. Якість потокової музики в безкоштовній версії обмежена, порівняно з іншими сервісами, що пропонують вищу якість звуку навіть у безкоштовному варіанті. Через те, що будь-хто може завантажувати свою музику, користувачі іноді стикаються з неякісним контентом або неоригінальними треками. Іншим недоліком є обмежена

інтеграція з іншими музичними сервісами: порівняно з конкурентами, SoundCloud має меншу інтеграцію з іншими програмними платформами та сервісами [8]-[11], [13].

Програмне забезпечення Musicolet (рис. 1.3) зручний та ефективний музичний плеєр для мобільних пристроїв, який дозволяє користувачам легко організовувати свої музичні файли в персоналізовані списки відтворення. Завдяки простому та інтуїтивно зрозумілому інтерфейсу плеєр надає можливість швидкого доступу до всіх треків без додаткових ресурсомістких функцій або постійного кешування бібліотеки, забезпечуючи плавне і персоналізоване прослуховування [11], [14].

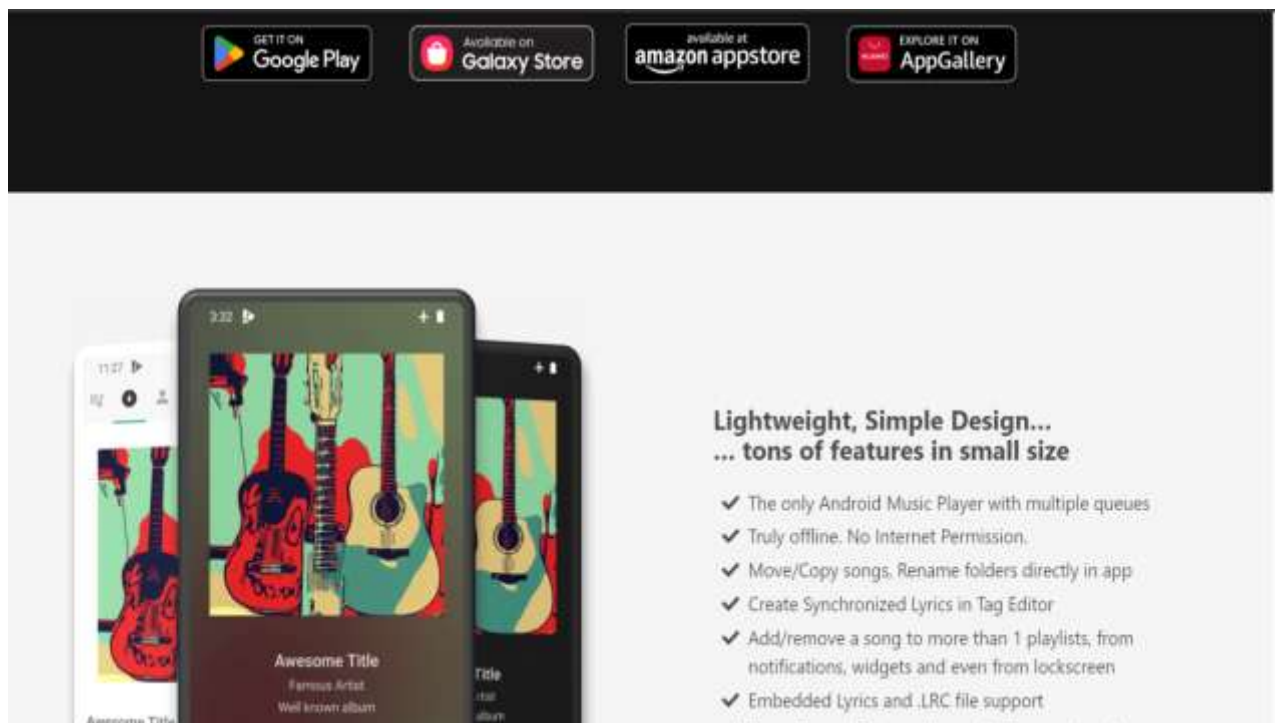


Рисунок 1.3 – Програмне забезпечення Musicolet [14]

Програмне забезпечення Musicolet пропонує широкий спектр налаштованих параметрів, що дозволяє користувачам адаптувати свої музичні вподобання до індивідуальних потреб. Інструменти, такі як детальне налаштування еквайзера та можливість створювати синхронізовані тексти пісень, надають можливість користувачам максимально оптимізувати якість звучання [11], [14].

Завдяки одноразовій оплаті за професійну версію Musicolet, користувачі можуть прослуховувати музичні твори без реклами та будь-яких обмежень, що робить цей інструмент ефективним вибором для звичайних слухачів [11], [14].

Програмне забезпечення Musicolet має такі особливості [11], [14]:

- офлайн програвання: програмне забезпечення Musicolet працює повністю офлайн, без необхідності підключення до Інтернету, дозволяючи відтворювати локальні музичні файли без затримок або буферизації [11], [14];

- багаточерговість: програмне забезпечення Musicolet підтримує створення кількох черг відтворення, що дозволяє користувачам перемикатися між різними списками відтворення в реальному часі [11], [14];

- еквалайзер: вбудований еквалайзер дозволяє точно налаштувати звук відповідно до особистих уподобань [11], [14];

- синхронізація текстів пісень: програмне забезпечення Musicolet дозволяє створювати та синхронізувати тексти пісень для відображення під час відтворення [11], [14];

- легкий інтерфейс: інтуїтивно зрозумілий та мінімалістичний інтерфейс спрощує навігацію та управління музичною бібліотекою [11], [14];

- підтримка віджетів: програмне забезпечення Musicolet підтримує кілька віджетів, що дозволяють легко керувати музикою з головного екрану програми [11], [14];

- миттєве сортування: програмне забезпечення Musicolet дозволяє сортувати та організовувати музичні файли за різними критеріями (виконавець, альбом, жанр) [11], [14];

- жести для керування: управління відтворенням у програмному забезпеченні Musicolet може здійснюватися за допомогою жестів, що спрощує взаємодію з додатком під час використання [11], [14];

- безкоштовна та платна версія: програма пропонує безкоштовний доступ до всіх основних функцій, а платна версія відкриває додаткові можливості без будь-яких обмежень [11], [14].

Серед недоліків програмного забезпечення Musicolet можна відзначити відсутність стримінгових функцій, оскільки Musicolet працює виключно офлайн і не підтримує потокове відтворення музики через Інтернет, що може бути недоліком для тих, хто хоче слухати онлайн радіо або стримінгові сервіси. Іншим недоліком є те, що програма Musicolet не підтримує всі можливі формати аудіофайлів, зокрема рідкісні або менш поширені формати можуть не працювати. Musicolet не інтегрується з музичними платформами, такими як Spotify, YouTube Music або інші популярні сервіси, що може обмежувати функціональність для користувачів, які використовують кілька додатків для музики. Крім того, Musicolet не підтримує синхронізацію з хмарними сервісами, що ускладнює керування музичними файлами на різних пристроях [9], [14].

Порівняльну характеристику програмного забезпечення для підбору музичних творів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для підбору музичних творів

Критерій порівняння	Pandora [12]	SoundCloud [13]	Musicolet [14]
Інтеграція з іншими сервісами	+—	—	+—
Ефективні рекомендаційні системи	+	+—	—
Можливість оцінювання треків	+	—	—
Фільтрація на основі вподобань	+	+—	—
Кросплатформність	+—	+	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для підбору музичних творів. Проте деякі програми для підбору музичних творів можуть не завжди точно

відповідати музичним вподобанням користувачів, оскільки персоналізовані рекомендації часто базуються лише на обмеженому наборі параметрів, як-от жанр або ритм, без урахування контексту або настрою. Такі рекомендації можуть не завжди відповідати емоційним чи культурним потребам користувачів. Іншим недоліком такого програмного забезпечення є те, що часто пропонуються популярні або трендові композиції, що обмежує доступ до менш відомих чи незалежних виконавців. Деякі програми для підбору музики збирають великі обсяги даних про вподобання користувачів, включаючи їхню музичну історію, час прослуховування та навіть інформацію про місцезнаходження, що може викликати занепокоєння щодо конфіденційності даних, оскільки існує ризик їх неналежного використання або витоку. Деякі користувачі програм для підбору музичних творів можуть мати проблеми з навігацією або взаємодією з інтерфейсом програм, оскільки не всі програми мають інтуїтивно зрозумілі інтерфейси. Тому актуальною є розробка програмного забезпечення для підбору музичних творів.

При розробці програмного забезпечення для підбору музичних творів необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення списку музичних творів, доступних для прослуховування;
- підтримка можливості відображення списку тегів, що являють собою узагальнення інформації про набір відповідних пісень або представляють відповідний музичний жанр;
- можливість пошуку музичних творів;
- можливість надання рекомендацій користувачу щодо музичних творів на основі його попередніх уподобань;
- підтримка можливості додавання, видалення та редагування інформації про музичні твори (функція адміністратора);
- підтримка можливості керування тегами музичних творів.

1.3 Висновки за розділом 1

Визначено, що технологія рекомендацій музичних творів являє собою систему, яка допомагає пропонувати користувачам пісні, альбоми або виконавців, спираючись на їхні вподобання та історію прослуховування. Показано, що програмне забезпечення для надання рекомендацій музичних творів використовує алгоритми для аналізу дій користувача, зокрема того, що він слухав, читав чи пропускав, щоб краще розуміти його музичні смаки. Завдяки цим даним програмна система може рекомендувати нові треки, які, ймовірно, зацікавлять користувача. Програмне забезпечення для підбору музичних творів працює, аналізуючи різні типи даних, щоб підібрати вам улюблені пісні або виконавців. Таке програмне забезпечення використовує різні технології для створення рекомендацій, об'єднуючи дані про поведінку користувачів, музичні патерни та навіть складний аналіз треків, щоб зробити рекомендації максимально індивідуальними. Проаналізовано компоненти програмного забезпечення для підбору музичних творів, зокрема, спільне фільтрування, контентне фільтрування та розширені моделі оцінки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підбору музичних творів. Проте деякі програми для підбору музичних творів можуть не завжди точно відповідати музичним вподобанням користувачів, оскільки персоналізовані рекомендації часто базуються лише на обмеженому наборі параметрів, як-от жанр або ритм, без урахування контексту або настрою. Такі рекомендації можуть не завжди відповідати емоційним чи культурним потребам користувачів. Іншим недоліком такого програмного забезпечення є те, що часто пропонуються популярні або трендові композиції, що обмежує доступ до менш відомих чи незалежних виконавців. Деякі програми для підбору музики збирають великі обсяги даних про вподобання користувачів, включаючи їхню музичну історію, час прослуховування та навіть інформацію про місцезнаходження, що може викликати занепокоєння щодо

конфіденційності даних, оскільки існує ризик їх неналежного використання або витоку. Деякі користувачі програм для підбору музичних творів можуть мати проблеми з навігацією або взаємодією з інтерфейсом програм, оскільки не всі програми мають інтуїтивно зрозумілі інтерфейси. Тому актуальною є розробка програмного забезпечення для підбору музичних творів.

Сформульовано функціональні вимоги до програмного забезпечення для підбору музичних творів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для підбору музичних творів було використано мову програмування Python [15], [16].

Python – це мова програмування високого рівня, що розроблена для спрощення написання та розуміння коду. Вона має вбудовану стандартну бібліотеку з численними корисними функціями, а також підтримує безліч плагінів і зовнішніх бібліотек, які додають додаткові можливості та алгоритми. Python підтримує кілька стилів програмування, зокрема об'єктно-орієнтоване, структуроване, функціональне та імперативне програмування. Важливі особливості цієї мови включають динамічну типізацію, автоматичне керування пам'яттю, обробку винятків та підтримку багатозадачності. Код і підручники можна організовувати в модулі, які, своєю чергою, можуть бути об'єднані в пакети [15], [16].

Python пропонує розробникам різні підходи для оцінки й створення програмного забезпечення, включаючи об'єктно-орієнтоване, процесно-орієнтоване або багатозадачне програмування. Завдяки можливостям абстракції, обробці винятків та маніпулюванню даними в класах, мова програмування Python дає змогу використовувати широкий спектр методів та технік програмування [15], [16].

Серед переваг Python слід відзначити високу читабельність коду, швидкість розробки, можливість роботи на різних платформах і легку інтеграцію з іншими мовами завдяки наявним інтерфейсам. Крім того, численні зовнішні бібліотеки та середовища для Python розширюють його функціонал, полегшуючи створення спеціалізованих програм [15], [16].

Python також надає можливість розробляти програми для різних операційних систем, що робить його зручним для кросплатформених рішень. Однак однією з недоліків Python є те, що його висока абстракція зменшує

ефективність при виконанні ресурсомістких обчислень порівняно з іншими мовами програмування [15], [16].

Мова програмування Python має одну з найбільших спільнот розробників, що забезпечує велику кількість навчальних матеріалів, документації та підтримки. Це важливо при вирішенні проблем або пошуку найкращих практик для реалізації методів та алгоритмів підбору музичних творів [15], [16].

Важливою характеристикою мови Python є те, що вона надає прості засоби для інтеграції з музичними інтерфейсами прикладних програм (Application Programming Interface, API) таких сервісів, як Spotify, Deezer або Apple Music, через бібліотеки на зразок Requests або HTTPX [15], [16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для підбору музичних творів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для підбору музичних творів

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	Java
Простий та зрозумілий синтаксис	+	+-	-
Засоби розв'язання завдань штучного інтелекту	+	+-	+-
Потужні фреймворки для розробки вебзастосунків	+	+-	+
Зручність для створення ПЗ для підбору музичних творів	+	-	-
Засоби для інтеграції з музичними API	+	+-	+

Отже, для реалізації програмного забезпечення для підбору музичних творів обрано мову програмування Python, яка має простий та зрозумілий синтаксис, є зручною для розробки вебзастосунків, зокрема, для роботи з музичними творами, має потужні фреймворки та надає прості засоби для інтеграції з музичними API таких сервісів, як Deezer та ін.

2.2 Вибір середовища розробки для створення програмного забезпечення для підбору музичних творів

В якості середовища розробки для створення програмного забезпечення для підбору музичних творів було обрано середовище PyCharm [17], [18].

Середовище розробки PyCharm є популярним серед розробників та програмістів, які працюють з Python, завдяки своїй ефективності та багатофункціональності. Великий набір можливостей робить його потужним інструментом, водночас не ускладнюючи роботу користувача. Це особливо корисно для початківців, які лише починають вивчати Python [17], [18].

Середовище розробки PyCharm забезпечує автоматичне завершення коду, рефакторинг, інтелектуальні підказки та відладку, що значно прискорює написання й тестування коду [17], [18].

Середовище розробки PyCharm забезпечує повну підтримку роботи з бібліотеками для машинного навчання та обробки даних, такими як NumPy, Pandas, scikit-learn та TensorFlow, що особливо важливо для створення алгоритмів рекомендацій у програмі підбору музичних творів. Зручний відладчик та підтримка наукових інструментів допомагають легко аналізувати та візуалізувати дані, що може бути корисним при налаштуванні алгоритмів для персоналізованих рекомендацій [17], [18].

Програма для підбору музичних творів вимагає роботу з базами даних для зберігання інформації про користувачів, плейлисти та музичні композиції.

PyCharm має вбудовану підтримку для роботи з різними типами баз даних, включаючи реляційні та NoSQL бази [17], [18].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для підбору музичних творів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для підбору музичних творів

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Sublime Text	Visual Studio
Спеціалізація на мові Python	+	+	–
Вбудована підтримка фреймворків для веброзробки	+	–	+–
Підтримка роботи з бібліотеками для машинного навчання	+	+–	+–
Зручний відладчик	+	+–	+
Зручність роботи з базами даних	+	+–	+

Таким чином, для створення програмного забезпечення для підбору музичних творів обрано середовище розробки PyCharm, що спеціалізується на мові Python, надаючи інструменти та функції, які оптимізують процес розробки саме для цієї мови програмування. PyCharm має вбудовану підтримку фреймворків для веброзробки, таких як Django та Flask, що дозволяє легко працювати над вебзастосунками для підбору музичних творів. Крім того, PyCharm має вбудовану підтримку для роботи з різними типами баз

даних, включаючи реляційні та NoSQL бази. PyCharm доступний для Windows, macOS та Linux, що дозволяє використовувати його на будь-якій операційній системі без втрати функціональності.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для підбору музичних творів

З метою можливості розробки програмного забезпечення для підбору музичних творів створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для підбору музичних творів враховуємо те, що користувачі програми мають можливості відображення та відтворення списку музичних творів, відображення списку тегів, що являють собою узагальнення інформації про набір відповідних пісень, а також перегляду списку музичних творів за обраним тегом, можливість пошуку музичних творів, отримання рекомендацій щодо музичних творів на основі попередніх уподобань користувача.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для підбору музичних творів можна описати таким чином:

- користувач програмного забезпечення для підбору музичних творів авторизується;
- після авторизації користувача сервер програмного забезпечення для підбору музичних творів у відповідь повертає основну сторінку вебсайту для роботи з музичними творами;
- користувач бачить відповідну сторінку програмного забезпечення для підбору музичних творів та шукає потрібну пісню;
- користувач прослуховує пісню, додає її до списку улюблених;
- користувач отримає перелік пісень, рекомендованих до прослуховування на основі його попередніх уподобань;
- користувач завершує роботу з програмним забезпеченням для підбору музичних творів.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для підбору музичних творів подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

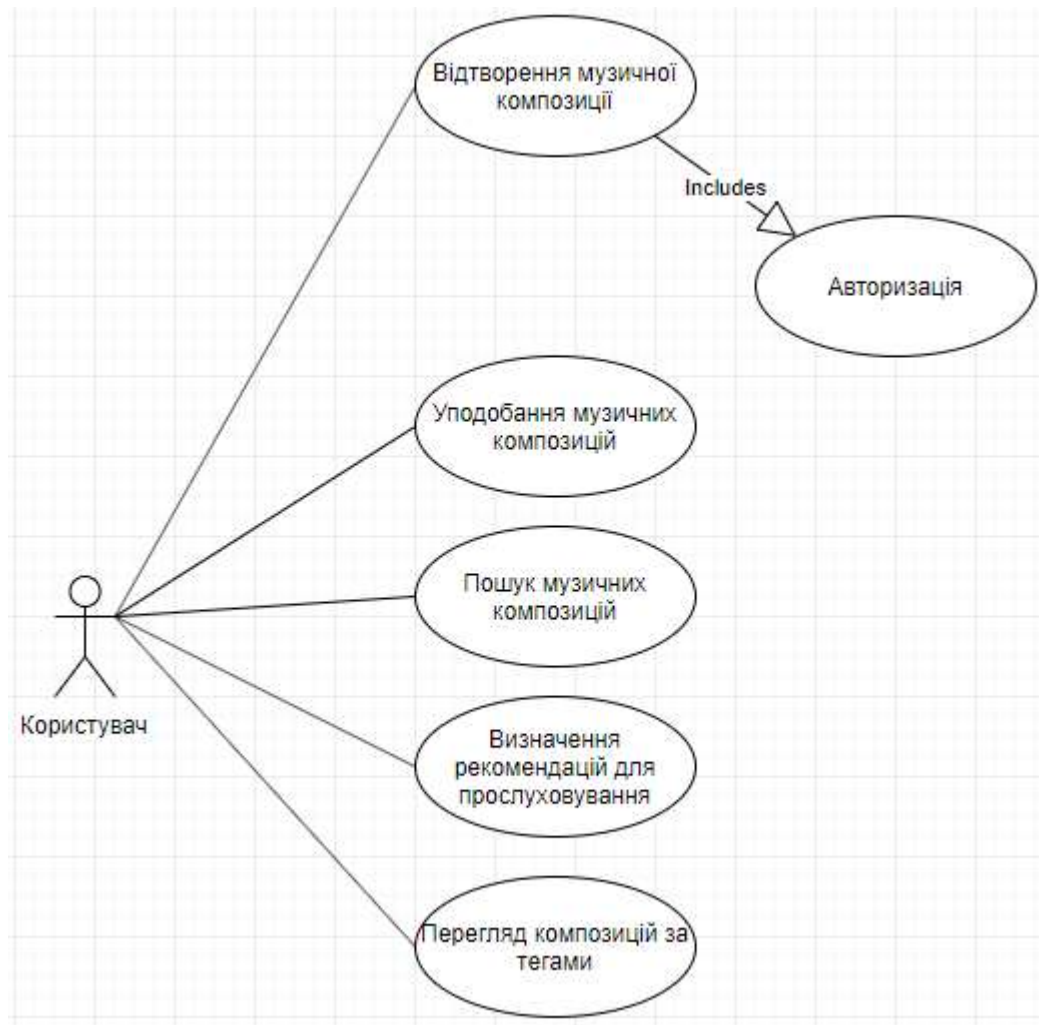


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для підбору музичних творів

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для підбору музичних творів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для підбору музичних творів.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для підбору музичних творів обрано мову програмування Python, яка має простий та зрозумілий синтаксис, є зручною для розробки вебзастосунків, зокрема, для роботи з музичними творами, має потужні фреймворки та надає прості засоби для інтеграції з музичними API таких сервісів, як Deezer та ін.

Для створення програмного забезпечення для підбору музичних творів обрано середовище розробки PyCharm, що спеціалізується на мові Python, надаючи інструменти та функції, які оптимізують процес розробки саме для цієї мови програмування. PyCharm має вбудовану підтримку фреймворків для веброзробки, таких як Django та Flask, що дозволяє легко працювати над вебзастосунками для підбору музичних творів. Крім того, PyCharm має вбудовану підтримку для роботи з різними типами баз даних, включаючи реляційні та NoSQL бази. PyCharm доступний для Windows, macOS та Linux, що дозволяє використовувати його на будь-якій операційній системі без втрати функціональності.

Створено модель взаємодії користувача з програмним забезпеченням для підбору музичних творів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для підбору музичних творів.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для підбору музичних творів

Структуру програмного забезпечення для підбору музичних творів наведено на рис. 3.1.

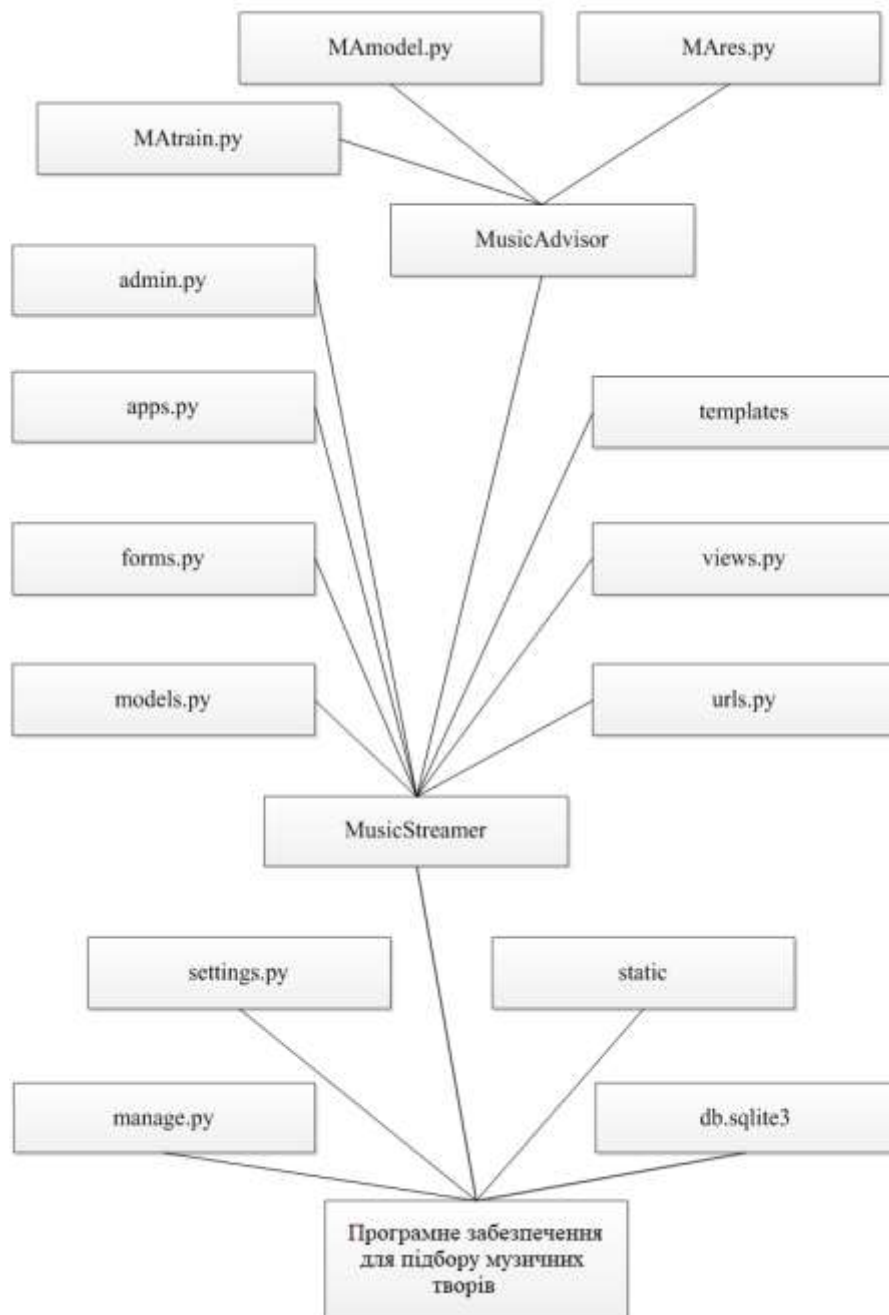


Рисунок 3.1 – Структура програмного забезпечення для підбору музичних творів

Програмне забезпечення для підбору музичних творів побудоване на основі багаторівневої архітектури, яка використовує вебфреймворк для керування функціональністю і забезпечення взаємодії між користувачем і системою. Основними складовими системи є модулі для керування базою даних, обробки запитів, управління музичними рекомендаціями та відображення інформації на вебсторінках.

Скрипти керування проектом використовуються для управління всією системою, включаючи запуск серверу, міграції бази даних та інші адміністративні функції.

У проекті використовується база даних, яка зберігає інформацію про користувачів, музичні твори, а також дані, необхідні для аналізу і формування рекомендацій. Дані організовані у вигляді таблиць, і їх взаємодія з вебзастосунком відбувається через об'єктно-реляційне відображення, що спрощує маніпуляцію даними.

Для відображення вебсторінок використовуються статичні файли, такі як каскадні таблиці стилів та зображення. Вони забезпечують створення сучасного та зручного інтерфейсу для користувачів.

Модуль налаштувань містить конфігурації, необхідні для коректної роботи всієї системи. Сюди входять шляхи до бази даних, параметри безпеки, налаштування статичних файлів та інші загальні конфігурації.

Основна логіка роботи застосунку організована у спеціальних модулях, що відповідають за окремі аспекти функціонування системи:

Адміністративний інтерфейс забезпечує доступ адміністраторів до даних системи для управління контентом і користувачами.

Форми використовуються для збору інформації від користувачів, наприклад, для реєстрації, входу в систему або налаштувань персоналізованих рекомендацій.

Моделі описують структуру даних і взаємодію з базою даних. Кожна модель відповідає за певний тип об'єктів, наприклад, музичні композиції, користувачі або параметри рекомендацій.

Вебсторінки створюються на основі шаблонів, що забезпечують відображення даних у зручній і зрозумілій формі для користувача.

Основною частиною програмного забезпечення є система рекомендацій, яка працює на основі прикладного програмного інтерфейсу Deezer API, який надає рекомендації на основі вподобань користувача та інших факторів [19].

Ця структура забезпечує ефективне функціонування програмного забезпечення для підбору музичних творів, надаючи користувачам інтуїтивний інтерфейс і точні рекомендації на основі їхніх вподобань та взаємодії з системою.

3.2 Функціонування програмного забезпечення для підбору музичних творів

Функціонування програмного забезпечення для підбору музичних творів базується на двох основних сценаріях використання: адміністрування системи та робота звичайного користувача. Залежно від ролі користувача, система автоматично адаптує доступні функції та можливості.

Для звичайного користувача інтерфейс починається з відображення доступного списку музичних творів. В залежності від його вподобань, користувач може виконувати кілька дій. Пошук музики: користувач може здійснювати пошук музичних композицій за назвою, жанром, виконавцем або іншими параметрами. Фільтрація за тегами: музичні твори можуть бути згруповані за певними тегами, що дозволяє користувачу швидко знайти музику відповідного стилю або настрою. Перегляд рекомендацій: користувач може отримати список рекомендованих треків, який формується на основі його попередніх прослуховувань або вподобань. Алгоритм аналізує музичні вподобання і пропонує схожі композиції. Вибір музичної композиції: після вибору конкретної композиції користувач може запустити її відтворення або додати до власних уподобань.

Функціонування програмного забезпечення для підбору музичних творів подамо за допомогою схеми, зображеної на рис. 3.2.

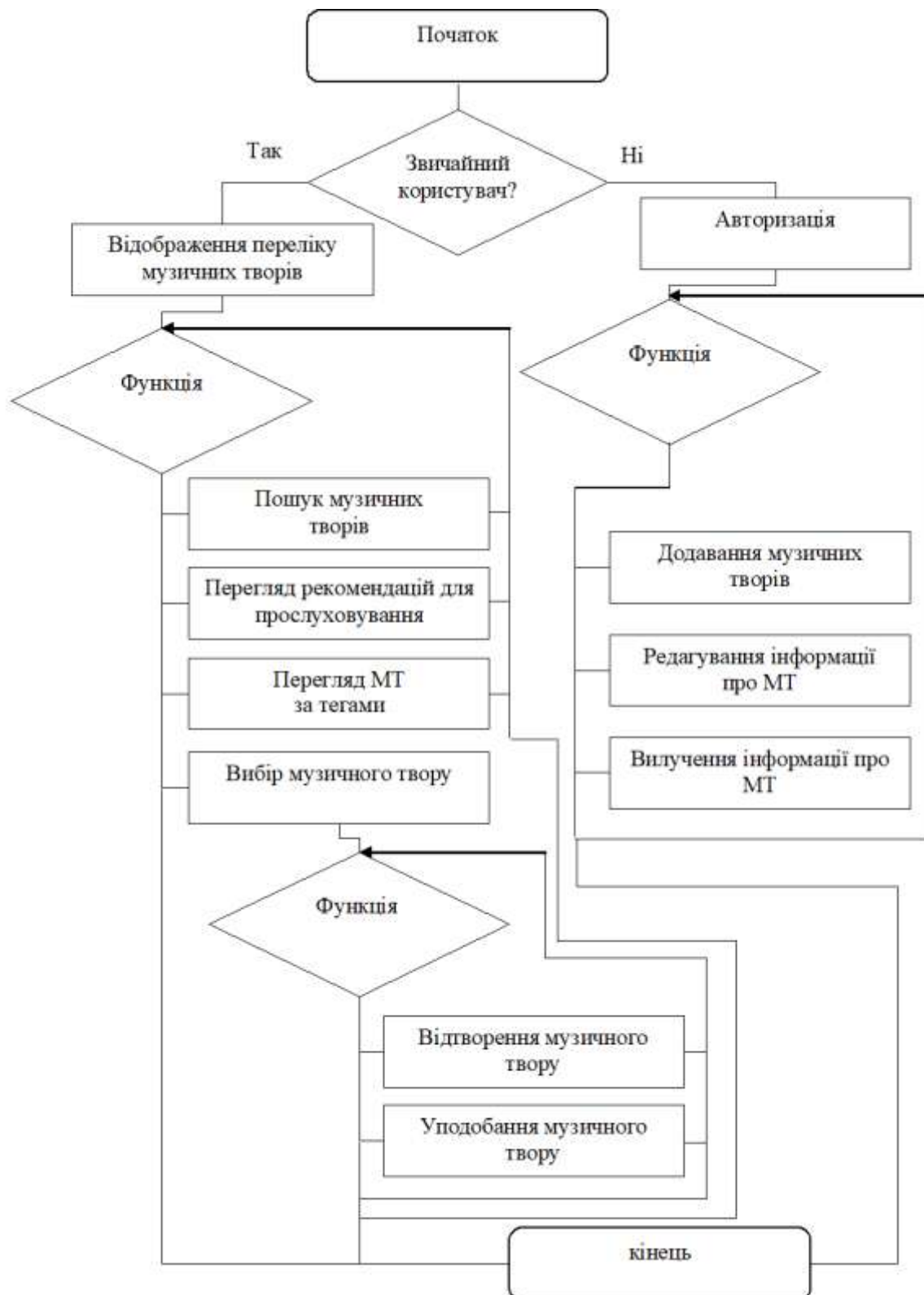


Рисунок 3.2 – Схема функціонування програмного забезпечення для підбору музичних творів

Деякі функції, як-от збереження улюблених треків або перегляд персоналізованих рекомендацій, вимагають авторизації через музичний сервіс. Якщо користувач не авторизований під час поточного сеансу, система запропонує йому увійти, щоб мати змогу користуватися цими функціями.

Програмне забезпечення використовує прикладний програмний інтерфейсу Deezer API, який надає рекомендації на основі вподобань користувача та інших факторів [19]. Цей інтерфейс дозволяє аналізувати історію прослуховувань та вподобання користувача, зокрема інформацію про жанри, темпи та тривалість композицій, щоб пропонувати треки, які ймовірно будуть цікавими для користувача [19].

Таким чином, програмне забезпечення для підбору музичних творів створює персоналізований музичний досвід, адаптуючи свої функції до потреб користувачів та надаючи адміністраторам повний контроль над вмістом.

Коли адміністратор входить у систему, йому доступні інструменти для керування музичною бібліотекою. Адміністратор спочатку проходить процес авторизації, після чого може виконувати такі дії. Додавання нових музичних творів включає завантаження інформації про композиції та її інтеграцію в базу даних. Редагування існуючих записів: адміністратор може змінювати інформацію про наявні музичні треки, оновлювати метадані, додавати або змінювати теги. Видалення композицій: адміністратор має можливість вилучити ті треки, які більше не актуальні або не відповідають критеріям. Після виконання адміністративних дій, адміністратор може продовжити роботу або завершити сеанс.

3.3 Розробка бази даних програмного забезпечення для підбору музичних творів

База даних програмного забезпечення для підбору музичних творів призначена для зберігання даних про музичні твори, теги, користувачів та ін.

Схему бази даних програмного забезпечення для підбору музичних творів наведено на рис. 3.3.

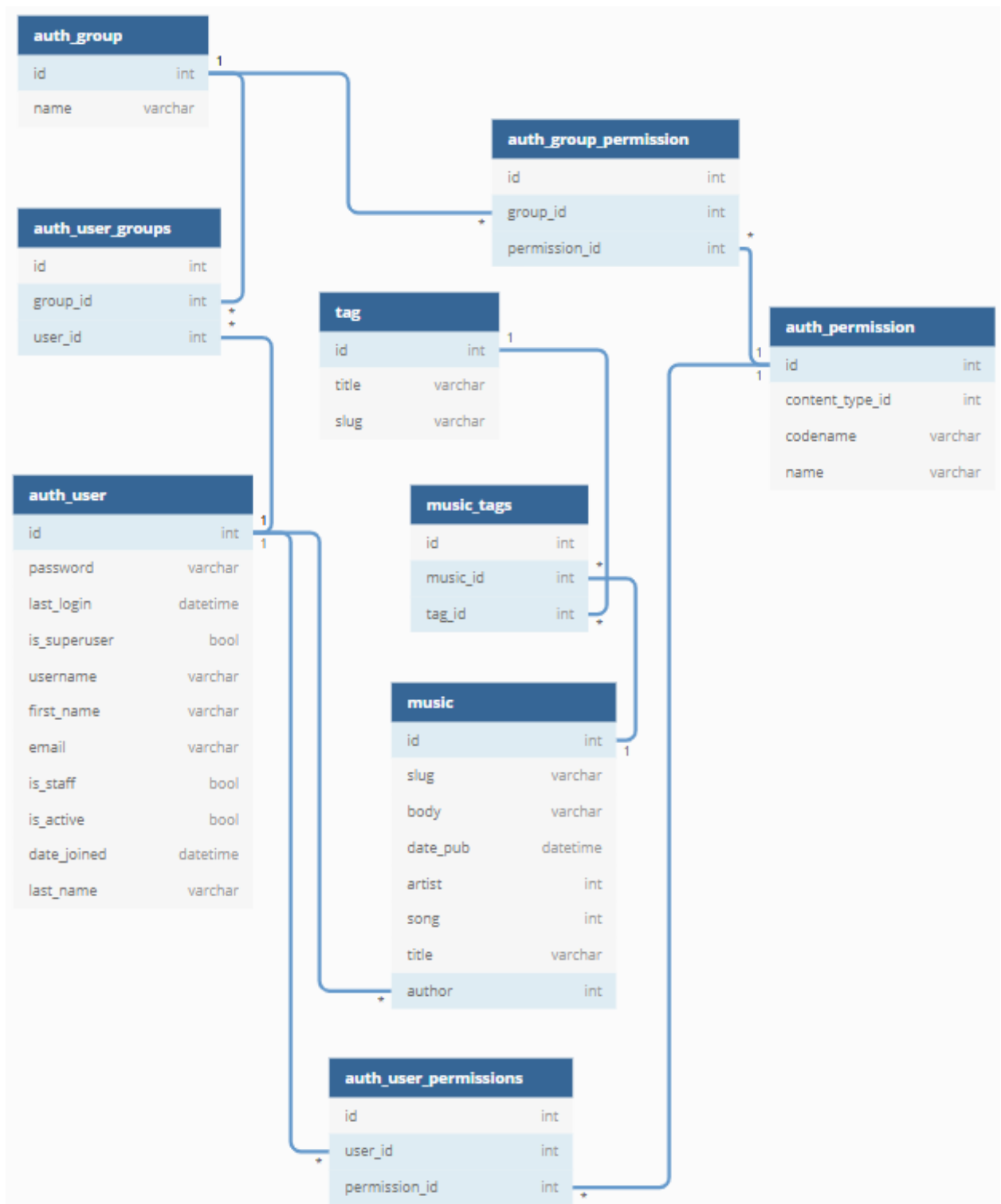


Рисунок 3.3 – Схема бази даних програмного забезпечення для підбору музичних творів

Схема бази даних програмного забезпечення для підбору музичних творів відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: музичний твір (music), користувач (auth_user), тег (tag) та інші.

3.4 Проєктування інтерфейсу програмного забезпечення для підбору музичних творів

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підбору музичних творів виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для підбору музичних творів враховані функціональні вимоги до програми:

- підтримка можливості відображення списку музичних творів, доступних для прослуховування;
- підтримка можливості відображення списку тегів, що являють собою узагальнення інформації про набір відповідних пісень або представляють відповідний музичний жанр;
- можливість пошуку музичних творів;
- можливість надання рекомендацій користувачу щодо музичних творів на основі його попередніх уподобань;
- підтримка можливості додавання, видалення та редагування інформації про музичні твори (функція адміністратора);
- підтримка можливості керування тегами музичних творів.

Інтерфейс головної форми програмного забезпечення для підбору музичних творів наведено на рис. 3.4.

Інтерфейс сторінки Теги наведено на рис. 3.5.

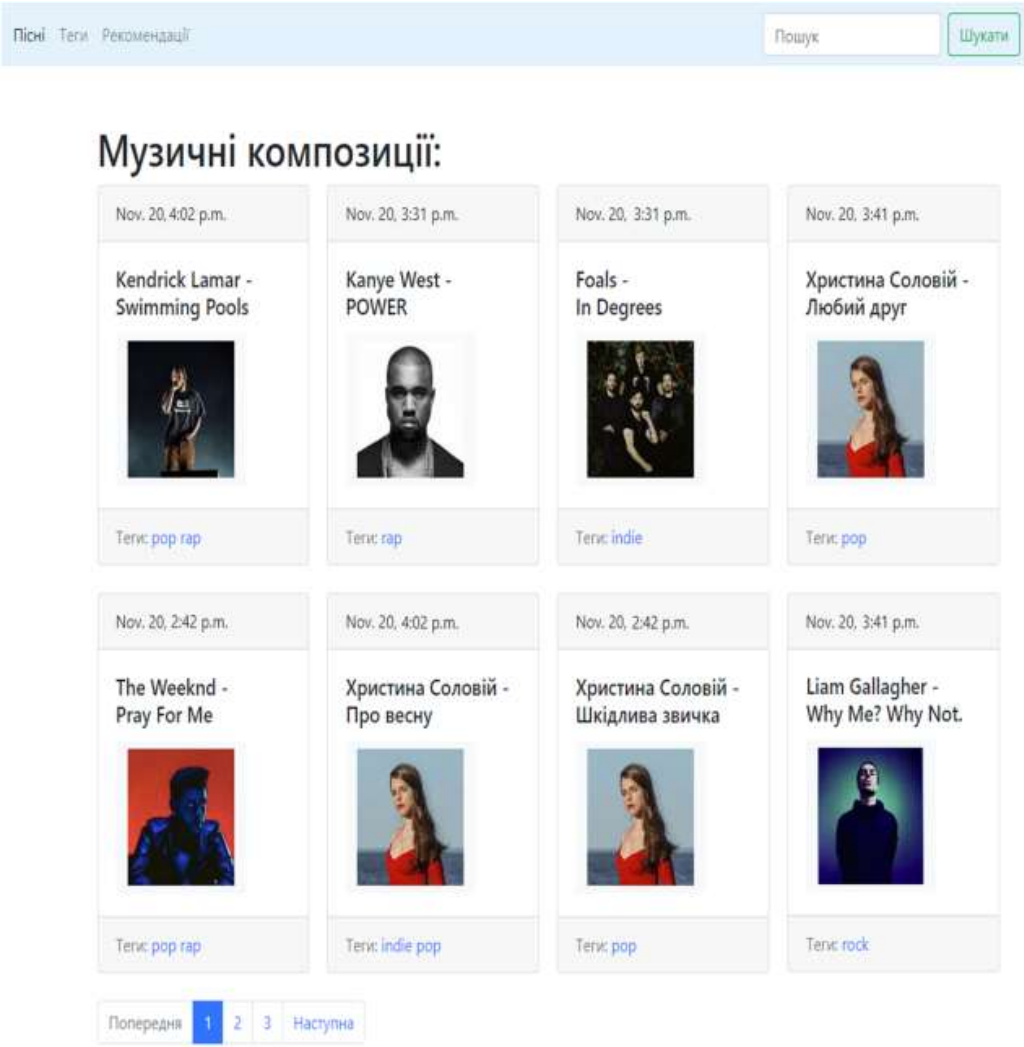


Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для підбору музичних творів

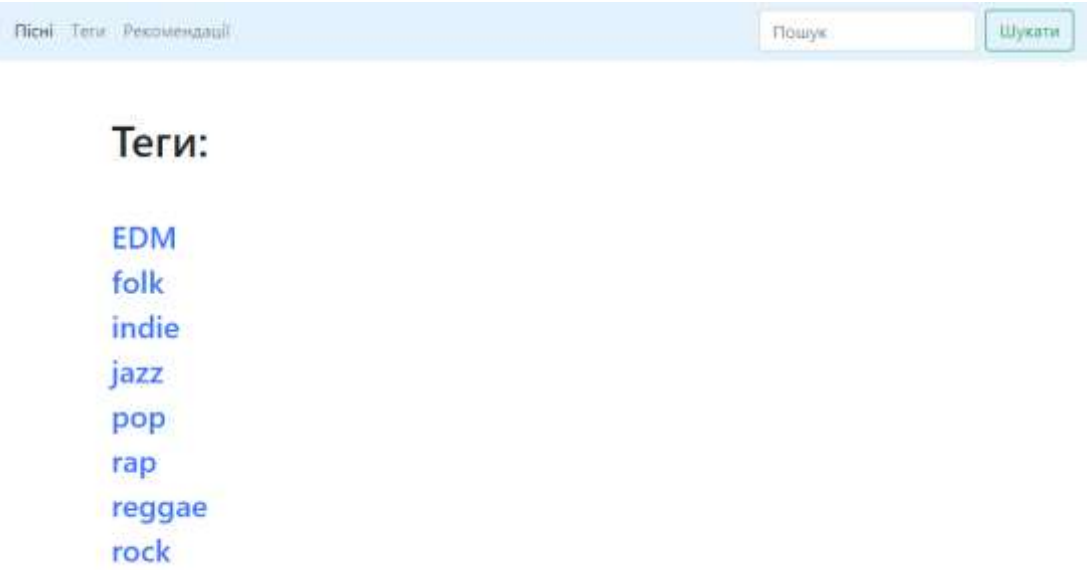


Рисунок 3.5 – Сторінка Теги

Інтерфейс сторінки прослуховування за допомогою вбудованого програвача наведено на рис. 3.6.

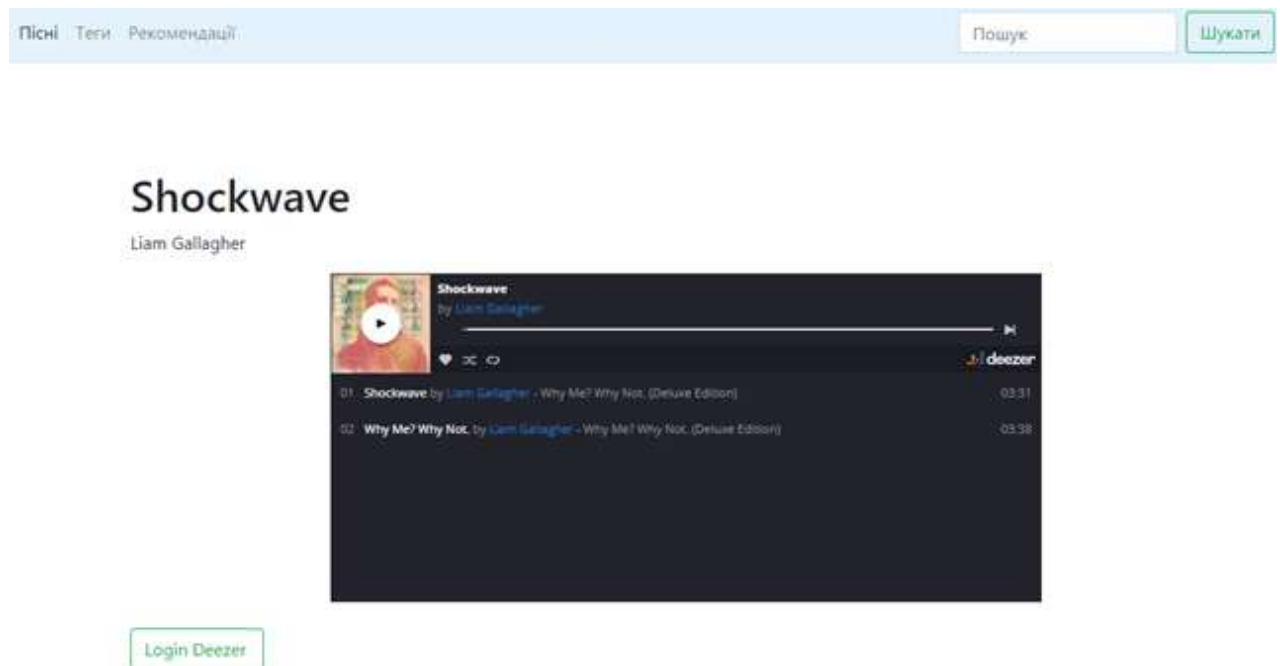


Рисунок 3.6 – Інтерфейс сторінки прослуховування за допомогою вбудованого програвача

Інтерфейс сторінки рекомендованих музичних творів наведено на рис. 3.7.

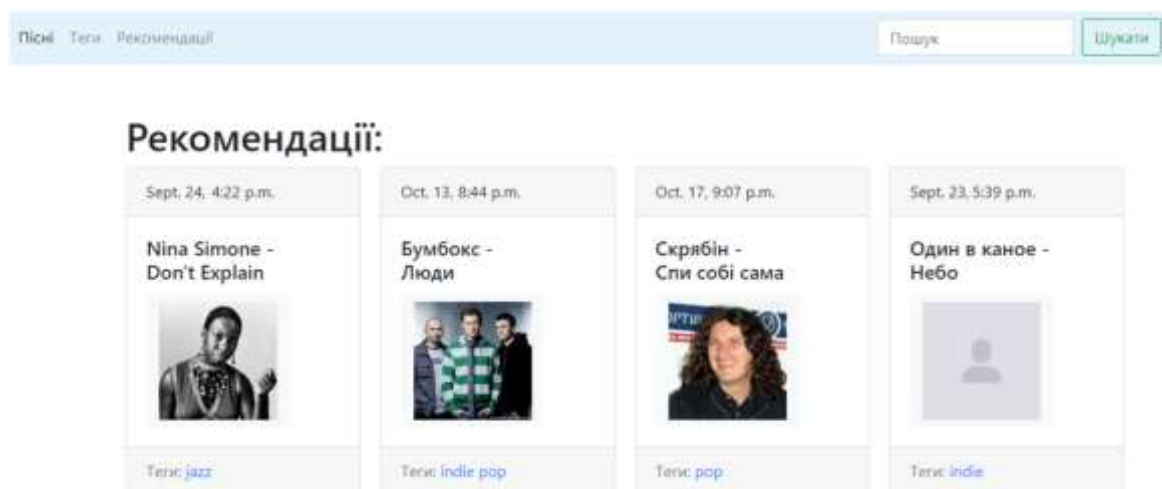


Рисунок 3.7 – Сторінка рекомендованих музичних творів

Інтерфейс сторінки редагування даних про музичний твір наведено на рис. 3.8.

Пісні Теги Рекомендації

Пошук Шукати

Назва
Sunrise

Короткий ідентифікатор
sunrise

Номер пісні у базі Deezer
3130150

Теги

- ☐ EDM
- ☐ folk
- ☐ indie
- ☒ jazz
- ☐ pop
- ☐ rap
- ☐ reggae
- ☐ rock

Оновити

Адміністратор
Створити ▾

Рисунок 3.8 – Сторінка редагування даних про музичний твір

Інтерфейс сторінки створення тегу наведено на рис. 3.9.

Пісні Теги Рекомендації

Пошук Шукати

Назва

Текстовий ідентифікатор

Створити тег

Адміністратор

Створити ▾

Рисунок 3.9 – Сторінка створення нового тегу

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для підбору музичних творів.

Запропоновано структуру програмного забезпечення для підбору музичних творів. Визначено, що програмне забезпечення для підбору музичних творів побудоване на основі багаторівневої архітектури, яка використовує вебфреймворк для керування функціональністю і забезпечення взаємодії між користувачем і системою. Основними складовими системи є модулі для керування базою даних, обробки запитів, управління музичними рекомендаціями та відображення інформації на вебсторінках.

Розроблено базу даних програмного забезпечення для підбору музичних творів, яка відображає концептуальні зв'язки між обраними

елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: музичний твір (music), користувач (auth_user), тег (tag) та інші.

Описано функціонування програмного забезпечення для підбору музичних творів.

Описано особливості реалізації програмного забезпечення для підбору музичних творів. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підбору музичних творів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для підбору музичних творів. Програма написана на мові Python, яка має простий та зрозумілий синтаксис, є зручною для розробки вебзастосунків, має потужні фреймворки та надає прості засоби для інтеграції з музичними API таких сервісів, як Deezer та ін. В якості середовища розробки для створення програмного забезпечення для підбору музичних творів обрано PyCharm, що спеціалізується на мові Python, надаючи інструменти та функції, які оптимізують процес розробки саме для цієї мови програмування.

Програмне забезпечення для підбору музичних творів забезпечує виконання таких функцій:

- підтримка можливості відображення списку музичних творів, доступних для прослуховування;
- підтримка можливості відображення списку тегів, що являють собою узагальнення інформації про набір відповідних пісень або представляють відповідний музичний жанр;
- можливість пошуку музичних творів;
- можливість надання рекомендацій користувачу щодо музичних творів на основі його попередніх уподобань;
- підтримка можливості додавання, видалення та редагування інформації про музичні твори (функція адміністратора);
- підтримка можливості керування тегами музичних творів.

Для експлуатації програмного забезпечення для підбору музичних творів необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять мінімум 4 Гб;
- не менше 600 Мб вільного простору на накопичувачі;
- процесор із тактовою частотою не менше 3 ГГц;

- наявність вебсерверу;
- встановлена операційна система на стороні клієнта.

Функціональні характеристики розробленого програмного забезпечення для підбору музичних творів наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для підбору музичних творів наведено у додатку Б.

4.2 Характеристики програми для підбору музичних творів

Програмне забезпечення для підбору музичних творів побудоване на основі багаторівневої архітектури, яка використовує вебфреймворк для керування функціональністю і забезпечення взаємодії між користувачем і системою. Основними складовими системи є модулі для керування базою даних, обробки запитів, управління музичними рекомендаціями та відображення інформації на вебсторінках.

Програма має простий і зрозумілий інтерфейс для користувачів, що дозволяє легко знаходити, слухати і підбирати улюблені музичні композиції. Програма підтримує функції пошуку музичних творів за різними критеріями та тегами. Програмне забезпечення для підбору музичних творів надає релевантні рекомендації, пропонуючи нові пісні або виконавців на основі поведінки користувача.

Програма має інтеграцію з музичною платформою Deezer, що дозволяє користувачам отримувати доступ до великої музичної бібліотеки.

Програмне забезпечення для підбору музичних творів підтримує роботу на різних пристроях, зокрема, на комп'ютерах, смартфонах і планшетах.

Вбудований музичний плеєр програмного забезпечення для підбору музичних творів підтримує відтворення музичних композицій з можливістю контролю гучності, переходу між треками та іншими функціями.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для підбору музичних творів на серверній частини необхідно запустити відповідний ру-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для підбору музичних творів є інформація про користувачів, обрані користувачами теги, музичні твори, параметри для пошуку музичних творів та ін.

Вихідними даними програмного забезпечення для підбору музичних творів є пропонувані користувачу музичні твори, аудіо потоки, що відповідають обраному музичному твору та програватимуться користувачу, список музичних творів, що відповідають критерію пошуку та ін.

4.3.3 Повідомлення

Користувач програмного забезпечення для підбору музичних творів може отримати такі повідомлення: повідомлення про пропонувані користувачу музичні твори та ін.

4.4 Виконання програмного забезпечення для підбору музичних творів

Після запуску програмного забезпечення для підбору музичних творів відображається головна сторінка (рис. 4.1), де користувач бачить список музичних творів, доступних для прослуховування. Також користувач має можливість знайти потрібний музичний твір за допомогою відповідного пошукового інструменту.

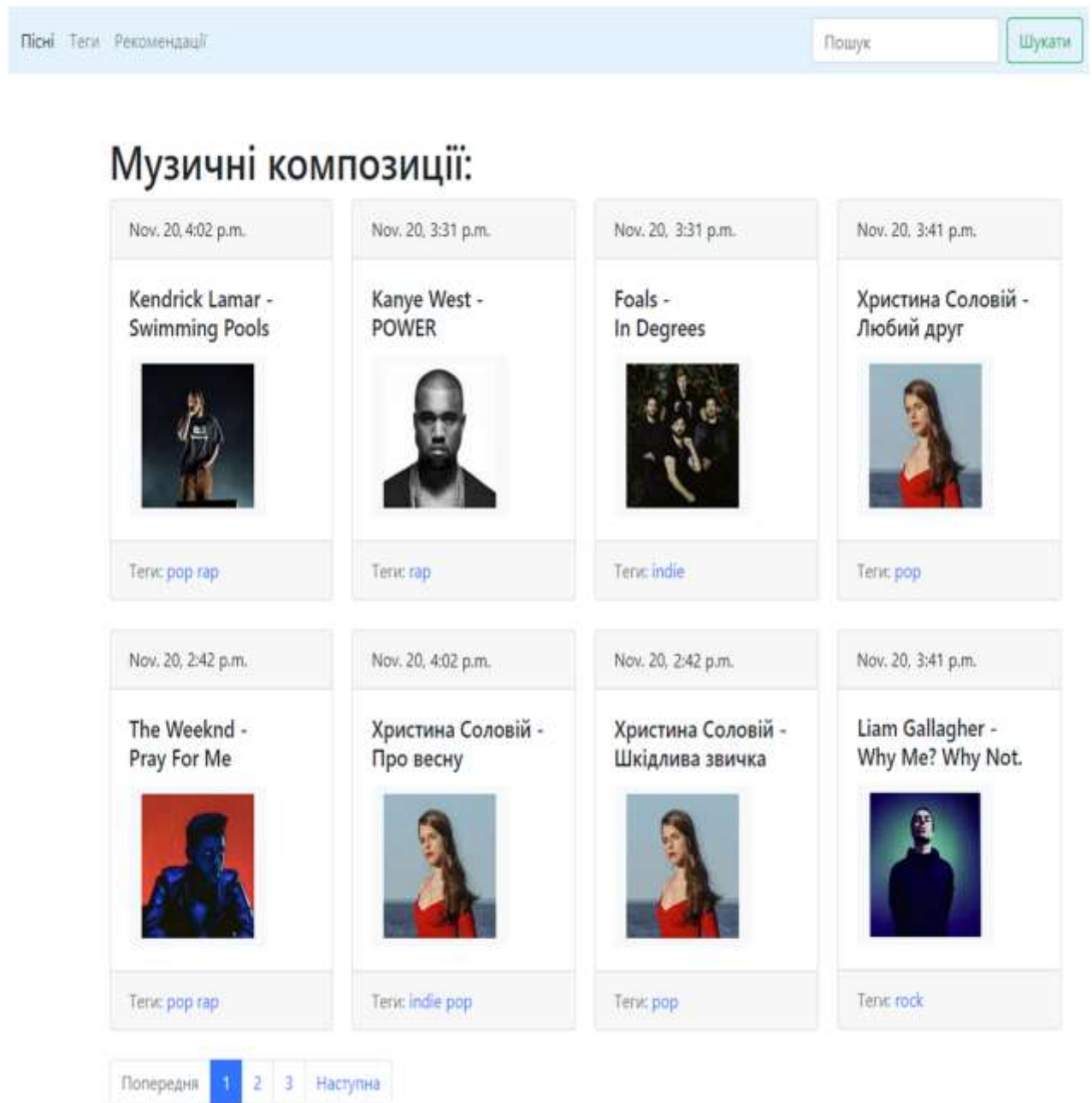


Рисунок 4.1 – Інтерфейс головної сторінки програмного забезпечення для підбору музичних творів

За допомогою головного меню, розташованого вгорі сторінки, користувач може перейти до сторінки Теги (рис. 4.2), де відображено список тегів, що являють собою узагальнення інформації про набір відповідних пісень або представляють відповідний музичний жанр.

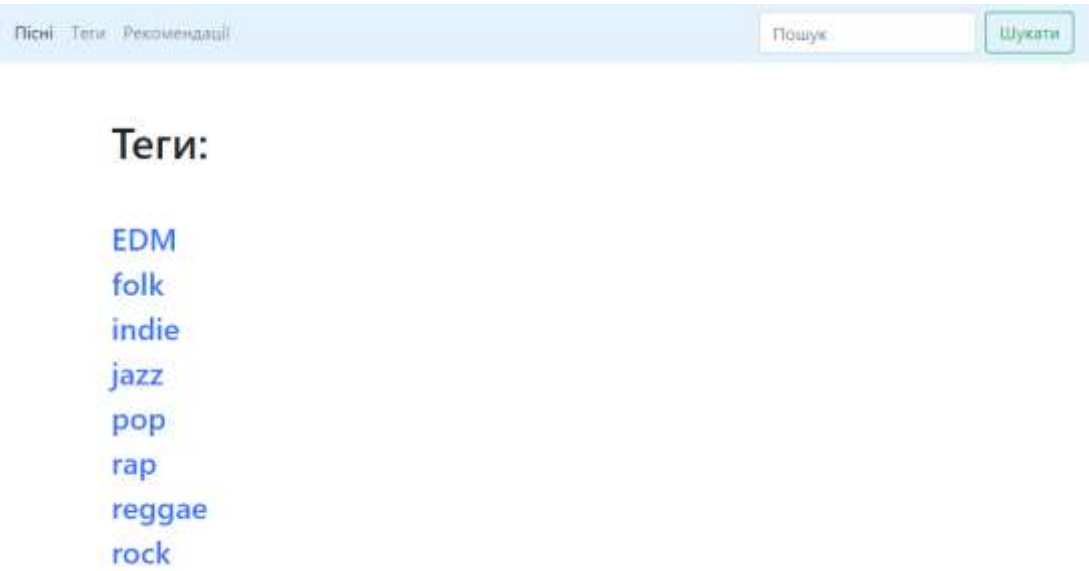


Рисунок 4.2 – Сторінка Теги

При виборі тегу користувачу відображається перелік музичних творів, що відповідають обраному тегу (рис. 4.3).

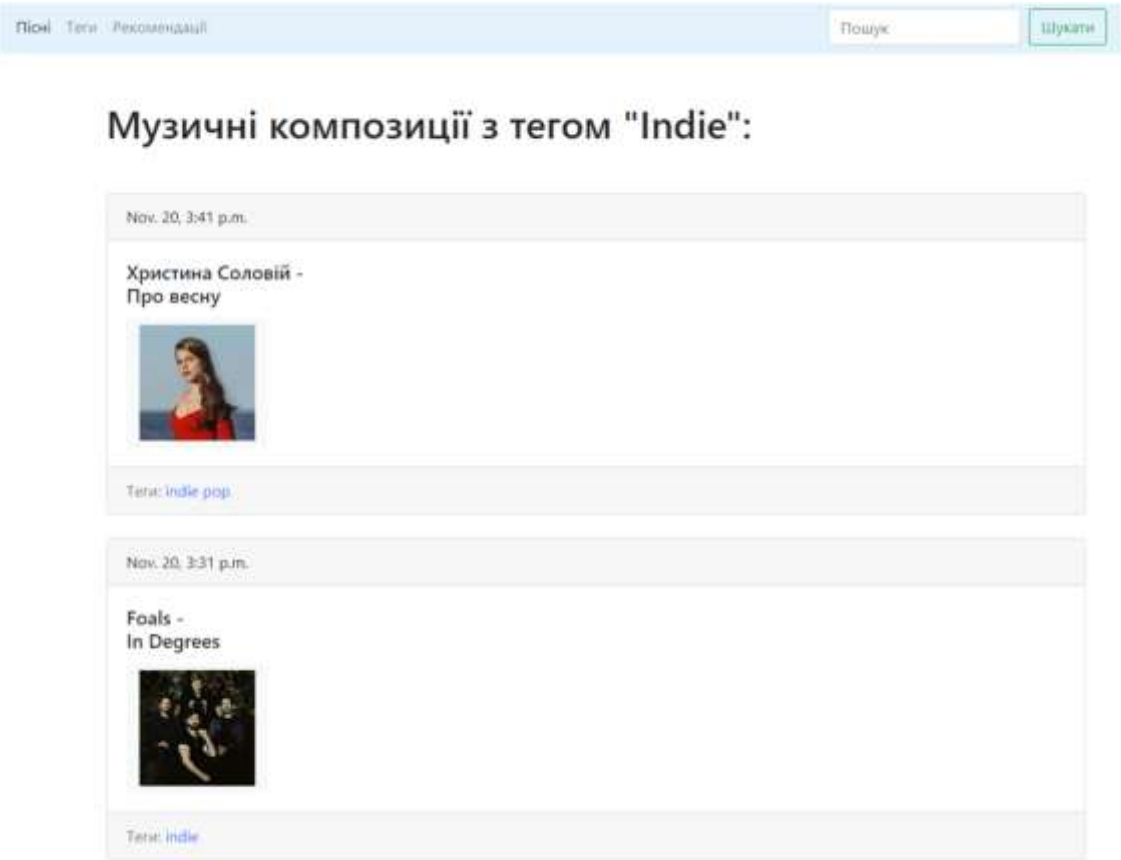


Рисунок 4.3 – Відбір музичних творів за тегом

Користувач, обираючи відповідний музичний твір, має можливість його прослуховування за допомогою вбудованого програвача (рис. 4.4).

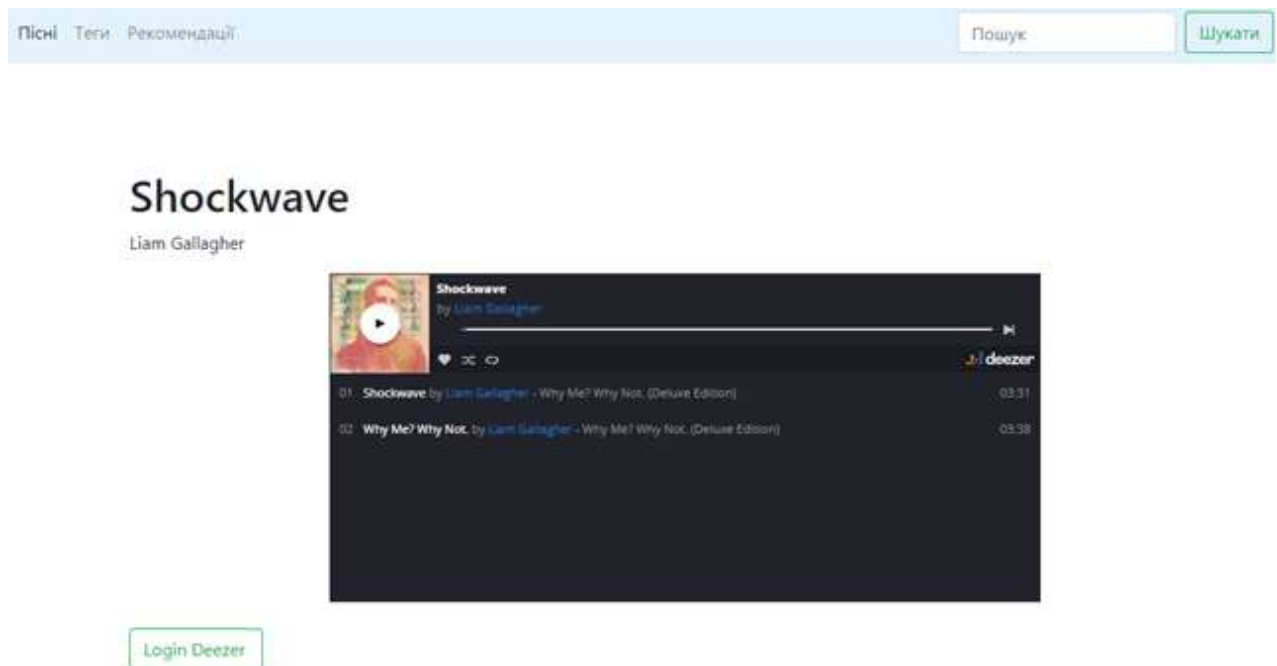


Рисунок 4.4 – Прослуховування музичного твору за допомогою вбудованого програвача

При виборі пункту меню Рекомендації користувачу виводиться перелік рекомендованих музичних творів (рис. 4.5), який генерується за допомогою API сервісу Deezer.

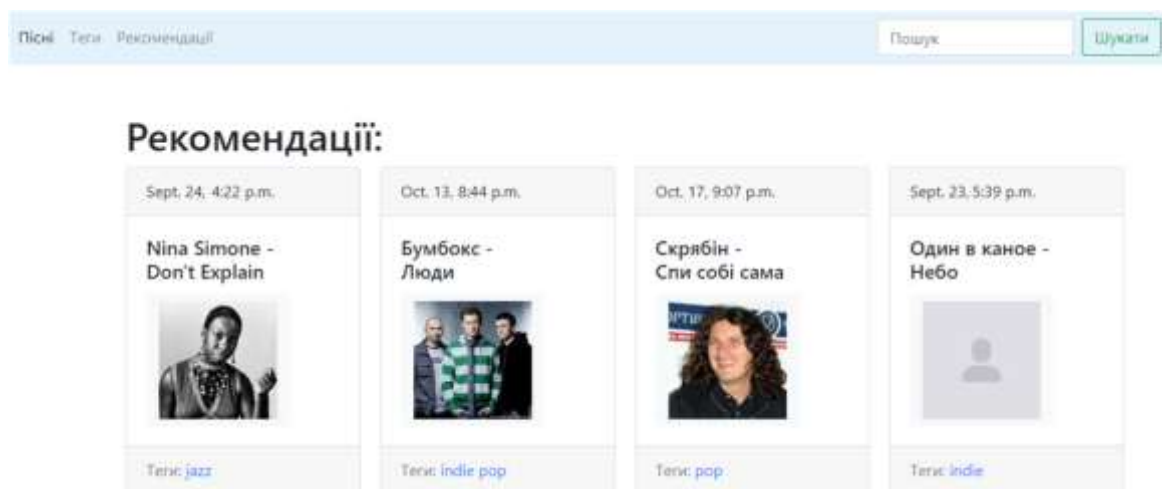


Рисунок 4.5 – Перелік рекомендованих музичних творів

Користувач з правами адміністратора має можливість додавання, видалення та редагування інформації про музичні твори. Приклад редагування даних про музичний твір наведено на рис. 4.6.

The screenshot shows a web interface for editing music data. At the top, there is a navigation bar with links: "Пісні", "Теги", and "Рекомендації". To the right of these links is a search bar with the placeholder text "Пошук" and a green button labeled "Шукати". Below the navigation bar, there are three input fields: "Назва" (Name) containing "Sunrise", "Короткий ідентифікатор" (Short identifier) containing "sunrise", and "Номер пісні у базі Deezer" (Deezer song number) containing "3130150". Below these fields is a section for tags, labeled "Теги", with a list of checkboxes: EDM, folk, indie, jazz (checked), pop, rap, reggae, and rock. At the bottom left of this section is a blue button labeled "Оновити". At the bottom right, there is a dropdown menu with "Адміністратор" selected and a "Створити" button with a downward arrow.

Рисунок 4.6 – Редагування даних про музичний твір

Адміністратор також має можливості керування тегами музичних творів. Приклад сторінки створення тегу наведено на рис. 4.7.

Рисунок 4.7 – Сторінка створення нового тегу

4.5 Тестування програмного забезпечення для підбору музичних творів

Виконано тестування програмного забезпечення для підбору музичних творів.

Виявлено, що програма для підбору музичних творів функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу підбору музичних творів.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу підбору музичних творів.

4.6 Висновки за розділом 4

Описано програмне забезпечення для підбору музичних творів. Виконано тестування розробленого програмного забезпечення для підбору музичних творів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу підбору музичних творів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для роботи з музичними творами.

Визначено, що технологія рекомендацій музичних творів являє собою систему, яка допомагає пропонувати користувачам пісні, альбоми або виконавців, спираючись на їхні вподобання та історію прослуховування. Показано, що програмне забезпечення для надання рекомендацій музичних творів використовує алгоритми для аналізу дій користувача, зокрема того, що він слухав, читав чи пропускав, щоб краще розуміти його музичні смаки. Завдяки цим даним програмна система може рекомендувати нові треки, які, ймовірно, зацікавлять користувача. Програмне забезпечення для підбору музичних творів працює, аналізуючи різні типи даних, щоб підібрати вам улюблені пісні або виконавців. Таке програмне забезпечення використовує різні технології для створення рекомендацій, об'єднуючи дані про поведінку користувачів, музичні патерни та навіть складний аналіз треків, щоб зробити рекомендації максимально індивідуальними. Проаналізовано компоненти програмного забезпечення для підбору музичних творів, зокрема, спільне фільтрування, контентне фільтрування та розширені моделі оцінки.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для підбору музичних творів. Проте деякі програми для підбору музичних творів можуть не завжди точно відповідати музичним вподобанням користувачів, оскільки персоналізовані рекомендації часто базуються лише на обмеженому наборі параметрів, як-от жанр або ритм, без урахування контексту або настрою. Такі рекомендації можуть не завжди відповідати емоційним чи культурним потребам користувачів. Іншим недоліком такого програмного забезпечення є те, що часто пропонуються популярні або трендові композиції, що обмежує доступ до менш відомих чи незалежних виконавців. Деякі програми для підбору

музики збирають великі обсяги даних про вподобання користувачів, включаючи їхню музичну історію, час прослуховування та навіть інформацію про місцезнаходження, що може викликати занепокоєння щодо конфіденційності даних, оскільки існує ризик їх неналежного використання або витоку. Деякі користувачі програм для підбору музичних творів можуть мати проблеми з навігацією або взаємодією з інтерфейсом програм, оскільки не всі програми мають інтуїтивно зрозумілі інтерфейси. Тому актуальною є розробка програмного забезпечення для підбору музичних творів.

Сформульовано функціональні вимоги до програмного забезпечення для підбору музичних творів.

Для реалізації програмного забезпечення для підбору музичних творів обрано мову програмування Python, яка має простий та зрозумілий синтаксис, є зручною для розробки вебзастосунків, зокрема, для роботи з музичними творами, має потужні фреймворки та надає прості засоби для інтеграції з музичними API таких сервісів, як Deezer та ін.

Для створення програмного забезпечення для підбору музичних творів обрано середовище розробки PyCharm, що спеціалізується на мові Python, надаючи інструменти та функції, які оптимізують процес розробки саме для цієї мови програмування. PyCharm має вбудовану підтримку фреймворків для веброзробки, таких як Django та Flask, що дозволяє легко працювати над вебзастосунками для підбору музичних творів. Крім того, PyCharm має вбудовану підтримку для роботи з різними типами баз даних, включаючи реляційні та NoSQL бази. PyCharm доступний для Windows, macOS та Linux, що дозволяє використовувати його на будь-якій операційній системі без втрати функціональності.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для підбору музичних творів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для підбору музичних творів.

Практична цінність результатів роботи полягає у тому, що розроблено програмне забезпечення для підбору музичних творів.

Запропоновано структуру програмного забезпечення для підбору музичних творів. Визначено, що програмне забезпечення для підбору музичних творів побудоване на основі багаторівневої архітектури, яка використовує вебфреймворк для керування функціональністю і забезпечення взаємодії між користувачем і системою. Основними складовими системи є модулі для керування базою даних, обробки запитів, управління музичними рекомендаціями та відображення інформації на вебсторінках.

Розроблено базу даних програмного забезпечення для підбору музичних творів, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних обрано такі: музичний твір (music), користувач (auth_user), тег (tag) та інші.

Описано функціонування програмного забезпечення для підбору музичних творів.

Описано особливості реалізації програмного забезпечення для підбору музичних творів. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для підбору музичних творів.

Виконано тестування програмного забезпечення для підбору музичних творів.

Виявлено, що програма для підбору музичних творів функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу підбору музичних творів.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу підбору музичних творів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Music Recommendation System: How Do Streaming Platforms Leverage AI in 2024 [Electronic resource]. – Access mode: <https://stratoflow.com/music-recommendation-system-guide/>.
2. Mood Based Music Recommendation System [Electronic resource]. – Access mode: <https://medium.com/@UTMSBA24/mood-based-music-recommendation-system-5afb8bb90082>.
3. Emotion-Based Music Recommendation System using a Deep Reinforcement Learning Approach [Electronic resource]. – Access mode: <https://medium.com/analytics-vidhya/emotion-based-music-recommendation-system-using-a-deep-reinforcement-learning-approach-6d23a24d3044>.
4. Building a Song Recommendation System with Spotify [Electronic resource]. – Access mode: <https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7>.
5. Music recommendation system: all you need to know [Electronic resource]. – Access mode: <https://www.eliftech.com/insights/all-you-need-to-know-about-a-music-recommendation-system-with-a-step-by-step-guide-to-creating-it/>.
6. Recommender Systems — A Complete Guide to Machine Learning Models [Electronic resource]. – Access mode: <https://towardsdatascience.com/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748>.
7. Content-driven music recommendation: Evolution, state of the art, and challenges [Electronic resource]. – Access mode: <https://www.sciencedirect.com/science/article/pii/S1574013724000029>.
8. What is the Best Music Recommendation App [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/best-music-recommendation-application/>.

9. Music Discovery Platforms You Need to Know About [Electronic resource]. – Access mode: <https://www.cambridgeaudio.com/usa/en/blog/10-music-discovery-platforms-you-need-know-about>.

10. Best of 8 apps for creating music playlists on mobile [Electronic resource]. – Access mode: <https://plan2winsoftware.com/post/the-best-apps-for-creating-music-playlists-on-android-iphone.html>.

11. The best free music players 2024: organize and enjoy your digital music collection [Electronic resource]. – Access mode: <https://www.techradar.com/news/the-best-free-music-player>.

12. Pandora [Electronic resource]. – Access mode: <https://www.pandora.com/>.

13. SoundCloud [Electronic resource]. – Access mode: <https://soundcloud.com/soundcloud>.

14. Musicolet [Electronic resource]. – Access mode: <https://krosbits.in/musicolet/>.

15. Python Forum [Electronic resource]. – Access mode: <https://python-forum.io/>.

16. Python Reference [Electronic resource]. – Access mode: https://www.w3schools.com/python/python_reference.asp.

17. Getting Started with Sourcery for JetBrains - IntelliJ, PyCharm and more [Electronic resource]. – Access mode: <https://docs.sourcery.ai/Coding-Assistant/Guides/Getting-Started/JetBrains/>.

18. PyCharm for Education. Learn Python in your IDE [Electronic resource]. – Access mode: <https://www.jetbrains.com/pycharm-edu/>.

19. Deezer for developers [Electronic resource]. – Access mode: <https://developers.deezer.com/login?redirect=/api>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для підтримки процесу підбору музичних творів.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів підбору музичних творів», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

А.2 Призначення розробки

Програмний продукт призначений для підбору музичних творів.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для підбору музичних творів забезпечує виконання таких функцій:

- підтримка можливості відображення списку музичних творів, доступних для прослуховування;
- підтримка можливості відображення списку тегів, що являють собою узагальнення інформації про набір відповідних пісень або представляють відповідний музичний жанр;
- можливість пошуку музичних творів;
- можливість надання рекомендацій користувачу щодо музичних творів на основі його попередніх уподобань;

- підтримка можливості додавання, видалення та редагування інформації про музичні твори (функція адміністратора);
- підтримка можливості керування тегами музичних творів.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для підтримки процесу підбору музичних творів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для підбору музичних творів повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для підбору музичних творів необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для підбору музичних творів необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять мінімум 4 Гб;
- не менше 600 Мб вільного простору на накопичувачі;
- процесор із тактовою частотою не менше 3 ГГц;
- наявність вебсерверу;

– встановлена операційна система на стороні клієнта.

A.3.6 Вимоги до маркування і пакування

Програма для підтримки процесу підбору музичних творів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для підбору музичних творів».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для підбору музичних творів є інформація про користувачів, обрані користувачами теги, музичні твори, параметри для пошуку музичних творів та ін.

Вихідними даними програмного забезпечення для підбору музичних творів є пропоновані користувачу музичні твори, аудіо потоки, що відповідають обраному музичному твору та програватимуться користувачу, список музичних творів, що відповідають критерію пошуку та ін.

ДОДАТОК Б
Фрагмент тексту програми

```

class Teg(Mdls.Mdl):
    nazv = Mdls.CharFld(maxdovg=50)
    slg = Mdls.SlgFld(maxdovg=50, unique=True)

    def gtabsturl(self):
        return reverse('Teg_detail_url', kwargs={'slg':
self.slg})

    def gtuptdturl(self):
        return reverse('Teg_uptdt_url', kwargs={'slg': self.slg})

    def gtdelurl(self):
        return reverse('Teg_delete_url', kwargs={'slg':
self.slg})

    def __str__(self):
        return '{}'.format(self.nazv)

    class Meta:
        ordering = ['nazv']

def mskslst(reqst):
    srchqr = reqst.GET.get('search', '')

    if srchqr:
        msks = Msk.tbjects.filter(Q(nazvkntns=srchqr) |
Q(bodykntns=srchqr))
    else:
        msks = Msk.tbjects.all()
    pgntnr = Pgntnr(msks, 8)

    pgnmbr = reqst.GET.get('strnk', 1)
    strnk = pgntnr.get_strnk(pgnmbr)

    ispgntd = strnk.has_other_strnks()

    if strnk.has_previous():
        prvurl = '?strnk={}'.format(strnk.previous_pgnmbr())
    else:
        prvurl = ''

```

```

if strnk.has_next():
    nxturl = '?strnk={}'.format(strnk.next_pgnmbr())
else:
    nxturl = ''

context = {
    'strnk_tbject': strnk,
    'ispgntd': ispgntd,
    'prvurl': prvurl,
    'nxturl': nxturl
}
return render(reqst, 'maa/index.html', context=context)

class MskDetail(TbjectDetailMixing, View):
    Mdl = Msk
    tmplt = 'maa/msk_datail.html'

class MskCreate(LoginRqrdMixin, TbjectCreateMixing, View):
    form_Mdl = MskForm
    tmplt = 'maa/msk_create_form.html'
    rsexception = True

class MskUptdt(LoginRqrdMixin, TbjectUptdtMixing, View):
    Mdl = Msk
    Mdl_form = MskForm
    tmplt = 'maa/msk_uptdt_form.html'
    rsexception = True

class MskDelete(LoginRqrdMixin, TbjectDeleteMixing, View):
    Mdl = Msk
    tmplt = 'maa/msk_delete_form.html'
    rdrkturl = 'mskslst_url'
    rsexception = True

def Tegslst(reqst):
    Tegs = Teg.tbjects.order_by('slg')

```

```

        return render(reqst, 'maa/Tegslst.html', context={'Tegs':
Tegs}))

```

```

class TegForm(forms.MdlForm):

```

```

    class Meta:

```

```

        Mdl = Teg

```

```

        flds = ['nazv', 'slg']

```

```

        widgets = {

```

```

            'nazv': forms.TextInput(attrs={'class': 'form-
control'}),

```

```

            'slg': forms.TextInput(attrs={'class': 'form-
control'})

```

```

        }

```

```

    def clean_slg(self):

```

```

        new_slg = self.cleaned_data['slg'].lower()

```

```

        if new_slg == 'create':

```

```

            raise ValidationError('Slg may not be "Create"')

```

```

        if Teg.tbjects.filter(slg__iexact=new_slg).count():

```

```

            raise ValidationError('Slg must be unique.')

```

```

        return new_slg

```

```

class MskForm(forms.MdlForm):

```

```

    class Meta:

```

```

        Mdl = Msk

```

```

        flds = ['nazv', 'slg', 'pish', 'Tegs']

```

```

        labels = {

```

```

            'Tegs': 'Tegs'

```

```

        }

```

```

        widgets = {

```

```

            'nazv': forms.TextInput(attrs={'class': 'form-
control'}),

```

```

            'slg': forms.TextInput(attrs={'class': 'form-
control'}),

```

```

            'pish': forms.MntberInput(attrs={'class': 'form-
control'}),

```

```

            'Tegs': forms.CheckboxSelectMultiple(attrs={'class':
'form-check-input'})

```

```

        }

```

```

def clean_slg(self):
    new_slg = self.cleaned_data['slg'].lower()

    if new_slg == 'create':
        raise ValidationError('Не можна створити')
    return new_slg

def stvrtslg(s):
    new_slg = slgify(s, allow_unicode=True)
    return new_slg + '-' + str(int(time()))

def stvrtvykonavec(mnt):
    vykonavec = []
    with urllib.request.urlopen("https://api.deezer.com/track/" +
str(mnt)) as url:
        data = json.loads(url.read().decode())
        vykonavec.append(data['vykonavec']['id'])
        vykonavec.append(data['vykonavec']['name'])
    return vykonavec

class Msk(Mdls.Mdl):
    nazv = Mdls.CharFld(maxdovg=50, db_index=True)
    slg = Mdls.SlgFld(maxdovg=150, blank=True, unique=True)
    pism = Mdls.IntegerFld(db_index=True, default=145429544)
    vykonavec = Mdls.IntegerFld(blank=True, default=66)
    body = Mdls.CharFld(maxdovg=150, blank=True)
    Tegn = Mdls.ManyToManyFld('Teg', blank=True,
related_name='msks')
    date_pub = Mdls.DateTimeField(auto_now_add=True)

    def gtabcurl(self):
        return reverse('msk_detail_url', kwargs={'slg':
self.slg})

    def gtuptdturl(self):
        return reverse('msk_uptdt_url', kwargs={'slg': self.slg})

    def gtdelurl(self):
        return reverse('msk_delete_url', kwargs={'slg':
self.slg})

```

```

def save(self, *args, **kwargs):
    if not self.slg:
        self.slg = stvrtslg(self.nazv)

    vykonavec = stvrtvykonavec(self.pisn)
    self.vykonavec = vykonavec[0]
    self.body = vykonavec[1]
    super().save(*args, **kwargs)

def __str__(self):
    return self.nazv

class Meta:
    ordering = ['-date_pub']

class TegDetail(TbjeetDetailMixing, View):
    Mdl = Teg
    tplmt = 'maa/Teg_detail.html'

class TegCreate(LoginRqrdMixin, TbjeetCreateMixing, View):
    form_Mdl = TegForm
    tplmt = 'maa/Teg_create.html'
    rsexception = True

class TegUptdt(LoginRqrdMixin, TbjeetUptdtMixing, View):
    Mdl = Teg
    Mdl_form = TegForm
    tplmt = 'maa/Teg_uptdt_form.html'
    rsexception = True

class TegDelete(LoginRqrdMixin, TbjeetDeleteMixing, View):
    Mdl = Teg
    tplmt = 'maa/Teg_delete_form.html'
    rdrkturl = 'Teglst_url'
    rsexception = True

class TbjeetDetailMixing:
    Mdl = None
    tplmt = None

    def get(self, reqst, slg):

```

```

    tbj = get_tbject_or_404(self.Mdl, slg__iexact=slg)
    pisns = []
    if self.Mdl is Msk:
        Teg_msk = []
        msklst = []
        for Teg in tbj.Tegs.all():
            Teg_msk.append(Teg.msk.all())
        for msk in Teg_msk:
            msklst.append(list(msk))
        for l in msklst:
            l.remove(tbj)
            for p in l:
                pisns.append(p.pisn)

    return render(reqst, self.tmplt,
context={self.Mdl.__name__.lower(): tbj, 'admntbject': tbj,
'detail': True, 'pisns': pisns})

class TbjectCreateMixing:

    form_Mdl = None
    tmplt = None

    def get(self, reqst):
        form = self.form_Mdl()
        return render(reqst, self.tmplt, context={'form': form})

    def msk(self, reqst):
        bndfrm = self.form_Mdl(reqst.POST)
        if bndfrm.is_valid():
            nwtbj = bndfrm.save()
            return redirect(nwtbj)
        return render(reqst, self.tmplt, context={'form':
bndfrm})

class TbjectUptdtMixin:

    Mdl = None
    Mdl_form = None
    tmplt = None

```

```

def get(self, reqst, slg):
    tbj = self.Mdl.tbjects.get(slg__iexact=slg)
    bndfrm = self.Mdl_form(instance=tbj)
    return render(reqst, self.tmplt, context={'form': bndfrm,
self.Mdl.__name__.lower(): tbj})

def msk(self, reqst, slg):
    tbj = self.Mdl.tbjects.get(slg__iexact=slg)
    bndfrm = self.Mdl_form(reqst.POST, instance=tbj)

    if bndfrm.is_valid():
        updt_Teg = bndfrm.save()
        return redirect(updt_Teg)
    return render(reqst, self.tmplt, context={'form': bndfrm,
self.Mdl.__name__.lower(): tbj})

class TbjectDeleteMixin:
    Mdl = None
    tmplt = None
    rdrkturl = None

    def get(self, reqst, slg):
        tbj = self.Mdl.tbjects.get(slg__iexact=slg)
        return render(reqst, self.tmplt,
context={self.Mdl.__name__.lower(): tbj})

    def msk(self, reqst, slg):
        tbj = self.Mdl.tbjects.get(slg__iexact=slg)
        tbj.delete()
        return redirect(reverse(self.rdrkturl))

class MAMdl(torch.nn.Module):

    def forward(self, indata, size, scrinfo):
        tns = torch.Tensor(128, 12, 32)
        for (n, seq) in emnterate(indata):
            lkm = torch.Tensor(12, 32)

            for (i, lks) in emnterate(seq):
                lks = torch.LongTensor(lks).rsltize_(1, len(lks))
                lks = self.embed(torch.autograd.Variable(lks))

```

```

        lks = self.dh(lks, dim=1)
        lks = lks.rslthape(32)
        lkm[i] = lks
        tns[n] = lkm
        rl = torch.nn.utils.rnn.pack_padded_sequence(tns, size,
batch_first=True)
        rslt, l = self.network(rl, scrinfo)
        prsp, _ = torch.nn.utils.rnn.pad_packed_sequence(rslt,
batch_first=True)
        return prsp, l

```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
 Національний університет «Запорізька політехніка»
 Кафедра програмних засобів

Дослідження та програмна реалізація засобів підбору музичних творів

Керівник	к.т.н., доцент Михайло КОЦУР
Виконав	студ. групи КНТ-133м Андрій ВІЛЬХОВСЬКИЙ

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процес розробки програмного забезпечення для роботи з музичними творами.
- **Предмет дослідження** – програмні засоби для підбору музичних творів.
- **Метою роботи є** дослідження та розробка програмного забезпечення для підбору музичних творів.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі необхідно розв’язати такі задачі:

- виконати аналіз предметної області та програмних засобів для підбору музичних творів;
- розробити модель взаємодії користувача з програмним забезпеченням для підбору музичних творів;
- здійснити проектування програмного забезпечення для підбору музичних творів;
- створити програмне забезпечення для підбору музичних творів;
- виконати тестування розробленого програмного забезпечення для підбору музичних творів.

3

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для підбору музичних творів

Критерій порівняння	Pandora	SoundCloud	Musicolet
Інтеграція з іншими сервісами	+-	-	+-
Ефективні рекомендаційні системи	+	+-	-
Можливість оцінювання треків	+	-	-
Фільтрація на основі вподобань	+	+-	-
Кросплатформність	+-	+	-

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	C#	Java
Простий та зрозумілий синтаксис	+	+-	-
Засоби розв'язання завдань штучного інтелекту	+	+-	+-
Потужні фреймворки для розробки вебзастосунків	+	+-	+
Зручність для створення ПЗ для підбору музичних творів	+	-	-
Засоби для інтеграції з музичними API	+	+-	+

5

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	PyCharm	Sublime Text	Visual Studio
Спеціалізація на мові Python	+	+	-
Вбудована підтримка фреймворків для веброзробки	+	-	+-
Підтримка роботи з бібліотеками для машинного навчання	+	+-	+-
Зручний відладчик	+	+-	+
Зручність роботи з базами даних	+	+-	+

6

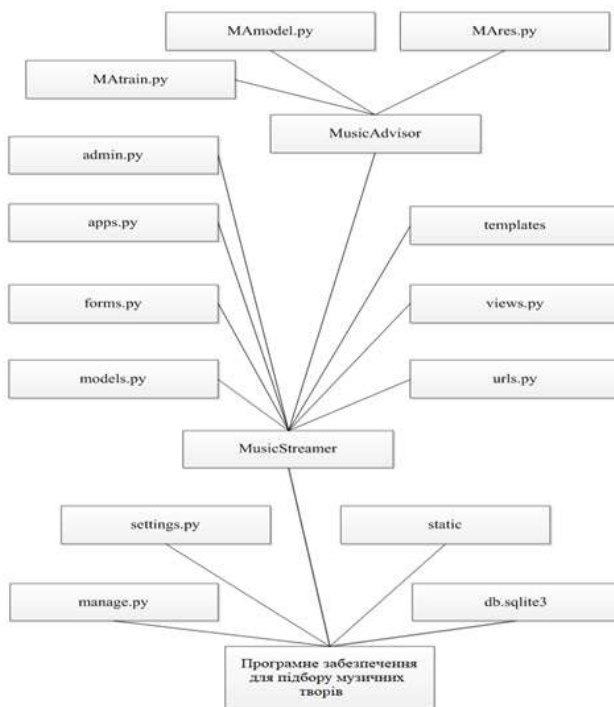
Рисунок В.6 – Слайд 6

Модель взаємодії користувача з ПЗ для підбору музичних творів



7

Рисунок В.7 – Слайд 7



8

Рисунок В.8 – Слайд 8

Структура програмного забезпечення для підбору музичних творів

Робота з програмою. Головна сторінка.

Відбір музичних творів за тегами

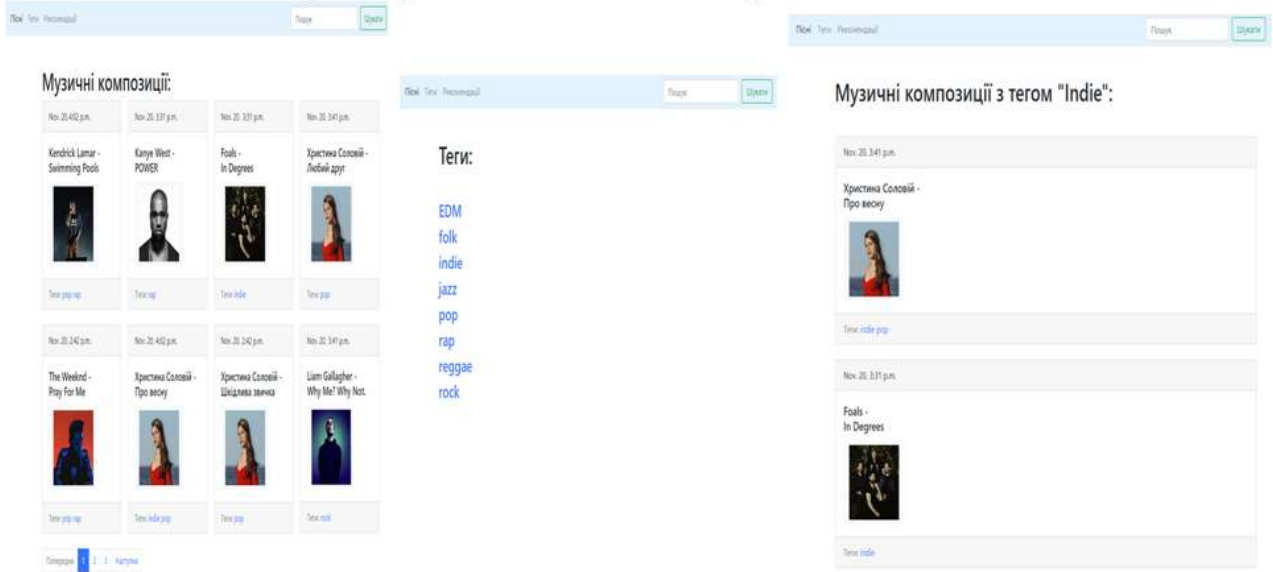


Рисунок В.11 – Слайд 11

Робота з програмою. Прослуховування

музики. Рекомендації музичних творів

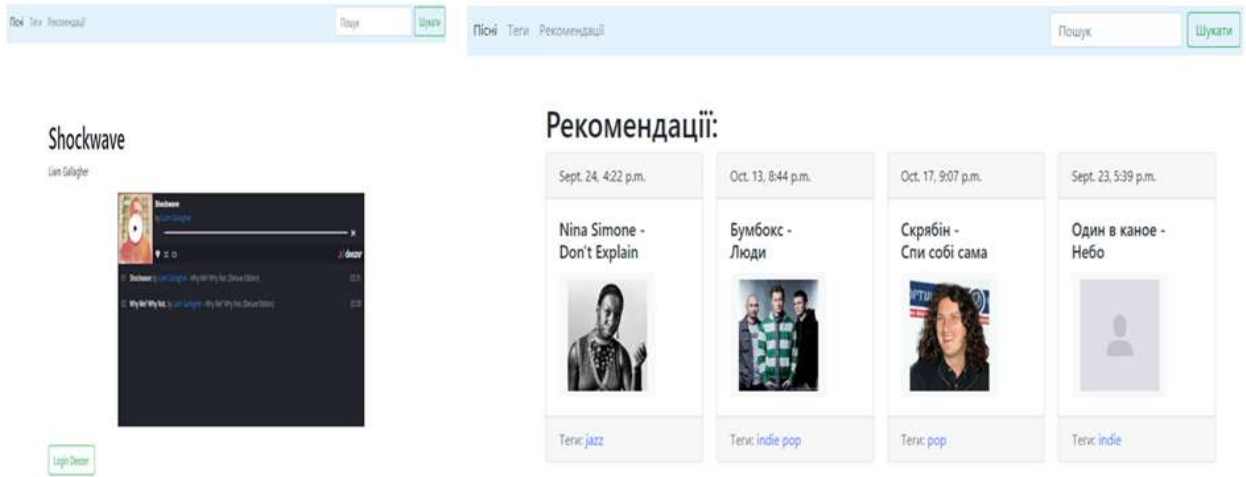


Рисунок В.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для роботи з музичними творами.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для підбору музичних творів, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для підбору музичних творів.

Практична цінність результатів роботи полягає у тому, що розроблено програмне забезпечення для підбору музичних творів.

13

Рисунок В.13 – Слайд 13