

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))


Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційна технологія для надання рекомендацій щодо
підбору музичного концерту»**

Виконав: студент 2-го курсу, групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)



_____ Стойко М.І.
(прізвище та ініціали)

Керівник: д.т.н., професор каф. КН
_____ Яровий А.А.
(прізвище та ініціали)

« 15 » _____ 12 _____ 2022 р.

Опонент: к.т.н., доцент каф. АПТ
_____ Богач І.В.
(прізвище та ініціали)

« 15 » _____ 12 _____ 2022 р.


Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)

« 16 » _____ 12 _____ 2022 р.

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та
 автоматизації
 Кафедра комп'ютерних наук
 Рівень вищої освіти II-й (магістерський)
 Галузь знань – 12 «Інформаційні технології»
 Спеціальність – 122 «Комп'ютерні науки»
 Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

Д.т.н., проф. Яровий А.А.

14^ч 09 2022 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Стойко Максим Ігорович

(прізвище, ім'я, по батькові)

- Тема роботи Інформаційна технологія для надання рекомендацій щодо підбору музичного концерту
 Керівник роботи д.т.н., професор кафедри КН Яровий А.А.
 Затверджені наказом вищого навчального закладу від 14^ч 09 2022 року № 203
- Строк подання студентом роботи 18 листопада 2022 року
- Вихідні дані до роботи:
Мова об'єктно-орієнтованого програмування, браузерне середовище роботи додатку, мінімальна кількість критеріїв класифікації концертів у фільтрі – 5 од., мова текстів, що класифікуються – англійська.
- Зміст текстової частини:
Вступ, аналіз сучасного стану розвитку систем надання рекомендацій щодо підбору музичного концерту, обґрунтування вибору інструментів технічної реалізації, проектування інформаційної технології, програмна реалізація інформаційної технології, аналіз результатів тестування, економічна частина, висновки, список використаних джерел, додатки.
- Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): узагальнена структурна схема інформаційної технології надання рекомендації щодо підбору музичного концерту, детальна структурно-функціональна схема роботи інформаційної технології, UML-діаграма класів інформаційної технології, схема узагальненого алгоритму роботи інформаційної технології, робочі вікна інформаційної технології надання рекомендацій щодо підбору музичного концерту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконав прийняв
1-3	Яровий А.А., д.т.н., проф. каф. КН	14.09.2022	14.10.2022
4	Нікіфорова Л. О., к. е. н., доц. каф. ЕПВМ	19.09.2022	17.12.2022

7. Дата видачі завдання _____ 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного стану розвитку систем для надання рекомендацій щодо підбору музичного концерту	14.09.2022	01.10.2022	Аналітичний огляд літературних джерел задачі досліджень, розділ 1
2	Обґрунтування методу розв'язання задачі, обґрунтування вибору інструментів технічної реалізації	02.10.2022	16.10.2022	Моделі, розділ 2
3	Проектування інформаційної технології. Програмна реалізація інформаційної технології. Аналіз результатів тестування	12.10.2022	02.11.2022	розділ 2, 3
4	Підготовка економічної частини	08.11.2022	24.11.2022	розділ 4
5	Апробація та/або впровадження результатів дослідження	23.11.2022	01.12.2022	тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.2022	14.12.2022	Пояснювальна записка, графічний матеріал, презентація

Студент

Керівник роботи

(підпис)

(підпис)

Стойко М.І.

Яровий А.А.

АНОТАЦІЯ

УДК 004.8

Стойко М.І. Інформаційна технологія для надання рекомендацій щодо підбору музичного концерту. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2022. 89 с.

На укр. мові. Бібліогр.: 40 назв; рис.: 17; табл. 7.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для аналізу надання рекомендацій щодо підбору музичного концерту. Розглянуто та проаналізовано існуючі програмні рішення і їх функціональні можливості, а саме системи підбору, та обрано гібридний режим функціонування з доданням нових критеріїв для більш точної рекомендації. Проаналізовано різні підходи до вирішення задачі надання рекомендацій щодо підбору музичного концерту та обрано підхід на основі теоремі Баєса через високу точність достовірності рекомендації. Спроектовано програму надання рекомендацій щодо підбору музичного концерту, написану мовою програмування JavaScript з використанням HTML та CSS.

Графічна частина складається з п'яти плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 968520 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 0,93 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: інформаційна технологія, надання рекомендацій, веб-клієнт, підбір музичного концерту.

ABSTRACT

Stoyko M.I. Information technology for providing recommendations on the selection of a musical concert. Master's thesis on specialty 122 - Computer science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2022.89p.

In Ukrainian speech Bibliography: 40 titles; Fig.: 17; table 7.

This master's thesis is devoted to the development of software for the analysis of providing recommendations for the selection of a musical concert. Existing software solutions and their functionality, namely selection systems, were reviewed and analyzed, and a hybrid mode of operation was chosen with the addition of new criteria for a more accurate recommendation. Various approaches to solving the problem of providing recommendations for selecting a musical concert were analyzed and an approach based on Bayes' theorem was chosen due to the high accuracy of the recommendation's reliability. A program for providing recommendations for the selection of a music concert was designed, written in the JavaScript programming language using HTML and CSS.

The graphic part consists of five posters.

In the economic section, the amount of costs for the development and production of a new technical solution is calculated, which is 968,520 hryvnias, the estimated amount of costs for each of the cost items is predicted, the net profit is calculated, the payback period for the manufacturer is 0.93 years, and the economic effect for the consumer when using this developments.

Keywords: information technology, provision of recommendations, web client, selection of a musical concert.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ.....	9
1.1 Огляд проблем в області надання рекомендацій щодо підбору музичного концерту	9
1.2 Аналіз систем-аналогів.....	11
1.3 Постановка задачі та формулювання вимог до інформаційної технології.....	15
1.4 Висновок до розділу 1	16
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ..	17
2.1 Аналіз методів надання рекомендацій	17
2.2 Обґрунтування вибору підходу до реалізації інформаційної технології	23
2.3. Розробка загальної структурної схеми функціонування інформаційної технології	25
2.4 Моделювання інформаційної технології надання рекомендацій щодо підбору концертів із використанням мови UML	28
2.5 Розробка узагальненого алгоритму роботи інформаційної технології надання рекомендацій щодо підбору музичного концерту	30
2.6 Висновок до розділу 2	32
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ.....	34
3.1 Обґрунтування вибору мови програмування для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту	34

3.2 Обґрунтування вибору середовища програмування для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту	36
3.3 Обґрунтування вибору системи керування базами даних для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту	39
3.4 Програмна реалізація компонентів інформаційної технології надання рекомендацій щодо підбору музичного концерту	42
3.5 Тестування та аналіз результатів роботи програми надання рекомендацій щодо підбору музичного концерту	46
3.6 Висновок до розділу 3	52
4 ЕКОНОМІЧНА ЧАСТИНА	54
4.1 Комерційний та технологічний аудит науково-технічної розробки.....	54
4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	57
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	63
4.4 Висновок до розділу 4	68
ВИСНОВКИ.....	69
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
Додаток А Протокол перевірки МКР на наявність текстових запозичень.....	74
Додаток Б Лістинг програми.....	75
Додаток В ІЛЮСТРАТИВНА ЧАСТИНА.....	85

ВСТУП

Актуальність теми дослідження. Сьогодні відбувається стрімкий розвиток культури по всьому світу. Вже протягом останніх 5 років у сфері світового музичного ринку відбувається стрімкий стабільний розвиток, який керується неймовірною музикою від чудових артистів у співпраці з талановитими та захопленими людьми в звукозаписуючих компаніях та організацій концертної діяльності по всьому світу [1]. Театри, концерти, фестивалі стають все більш популярним відпочинком не тільки серед молоді, але і людей любого віку. Існує багато різноманітної музики та нішевого продукту, який стає масовим [2]. Кожного дня в медіапросторі з'являються нові цікаві артисти, створюючи щось нове – звичайні рамки музичних жанрів стають все більш розмиті, а конкуренція серед виконавців і кількість концертів тільки зростає. Проте, з ростом кількості цікавих музикантів, зростає і складність пошуку та вибору концерту, необхідного під потреби відвідувача. Актуальним на сьогодні є питання підбору музичних концертів, які цікаві споживачу. Існує ряд квиткових операторів, які вбудовують певні критерії та фільтри до існуючих в базі концертів, щоб допомогти відвідувачу знайти те, що йому цікаво. Проте навіть цього дуже мало, оскільки користувачу треба витратити багато часу на пошук серед підібраної бази, враховуючи насичений ринок, конкуренцію та особливості певних концертів. Саме тому проблема створення інформаційної технології для надання рекомендацій щодо підбору музичного концерту є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету спеціальності 122 «Комп'ютерні науки» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології

та підвищення якості надання рекомендацій щодо підбору музичних концертів.

Задачі дослідження:

1. Аналіз сучасного стану розвитку систем для надання рекомендацій щодо підбору музичного концерту.
2. Аналіз методів надання рекомендацій.
3. Проектування інформаційної технології для надання рекомендацій щодо підбору музичного концерту.
4. Розробка алгоритму програмної реалізації інформаційної технології для надання рекомендацій щодо підбору музичного концерту.
5. Програмна реалізація інформаційної технології для надання рекомендацій щодо підбору музичного концерту.
6. Тестування інформаційної технології для надання рекомендацій щодо підбору музичного концерту та аналіз результатів тестування.

Об'єктом дослідження є процес надання рекомендацій щодо підбору музичного концерту.

Предметом дослідження є програмні засоби надання рекомендацій щодо підбору музичного концерту.

Методи дослідження - методи надання рекомендацій, методи теорії експертних системи, методи масштабування та інтеграції програмного забезпечення, методи та підходи до розробки веб-орієнтованих програмних додатків, методи об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів:

Розроблемо web-орієнтовану інформаційну технологію для надання рекомендацій щодо підбору музичного концерту, що відрізняється від існуючих застосувань баєсового підходу при організації механізму логічного виведення, що забезпечило розширення функціональних можливостей та підвищення якості надання рекомендацій щодо підбору музичного концерту.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритм програмної реалізації web-орієнтованої інформаційної технології для надання рекомендацій щодо підбору музичного концерту.

2. Здійснено програмну реалізацію інформаційної технології для надання рекомендацій щодо підбору музичного концерту з розширеними функціональними можливостями.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація. Основні результати роботи апробовано на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи», ВНТУ Вінниця, 2022.

Публікації. За результатами роботи опубліковано одні тези доповіді [3].

1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ СИСТЕМ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ

1.1 Огляд проблем в області надання рекомендацій щодо підбору музичного концерту

Світовий музичний ринок складається з багатьох галузей, але залишилася одна галузь, яка протягом усієї історії популярної музики залишалася незмінно стабільною і прибутковою — живі виступи. Саме вони є основним джерелом доходів для музикантів в усьому світі. На рівні більшості артистів — тури, фестивалі та корпоративи стали головним джерелом прибутку й існування. За даними дослідження музичного ринку, 70% артистів мають живі виступи як основне джерело прибутку [4].

Гастролі становлять величезну частину життя артиста та левову частку доходів галузі. У той же час, це єдина частина музичної кар'єри, яка на 99% залишається «фізичною» в індустрії, яка в іншому випадку є цифровою індустрією. Хоча деякі артисти можуть легко охопити мільйони шанувальників за допомогою прослуховувань на музичних сервісах, організувати міжнародний тур для такої ж аудиторії – надзвичайно складний процес. Незважаючи на те, що до 2022 року очікується, що дохід від прослуховувань на музичних сервісах зросте до 23 мільярдів доларів США, прогнозується, що до того ж року світова вартість індустрії живої музики досягне колосальних 31 мільярда доларів. Глобальні доходи від живої музики продовжують зростати (значна частина цього зростання пов'язана зі світовим вибухом фестивалів електронної музики та музичних турів нових виконавців, які здобувають популярність серед молоді завдяки сучасній соціальній мережі TikTok). У той же час, якщо подивитися на добре розвинені музичні економіки, нещодавнє дослідження показало, що 52% населення у світі відвідують живі шоу принаймні раз на рік [5].

Задача кожного квиткового оператора, окрім надання сервісу купівлі квитків на концерти, також полегшити процес підбору бажаної події шляхом застосування певних критеріїв та фільтрів при пошуку відвідувачем. Кожен такий оператор має свою інформацію, якою може оперувати користувач.

Відповідно, на сайтах квиткових операторів існують наступні критерії для вибору музичного концерту:

- назва виконавця
- жанр виконання;
- локація проведення;
- місто проведення;
- дата проведення.

Вищенаведений список є неповним, саме тому і відкриває таку проблему як недостатність або некоректність необхідної інформації, щоб відвідувач міг обрати необхідний для себе концерт.

На даний момент ринок квиткових операторів характеризується невисокою лояльністю серед кінцевих споживачів. Це пояснюється тим, що послуги операторів не несуть особливих цінностей і вигод для кінцевого споживача, що є причиною низького рівня прихильності та довіри до них.

Саме тому необхідно знаходити компроміс у вигляді розробки єдиної інформаційної технології з повним необхідним набором побажань, можливостей та інтересів потенційного відвідувача для коректності та достатньої точності наданої рекомендації щодо запропонованого музичного концерту.

Вирішити цю проблему може багатокритеріальний аналіз отриманої інформації з використанням покращеного підходу фільтрації, де до існуючих критеріїв ми додаємо ще декілька для більш точної рекомендації. Такий підхід не покриває повністю потреби користувачів, але забезпечує більшу достовірність підбору.

1.2 Аналіз систем-аналогів

Проблему надання рекомендацій щодо підбору музичного концерту вирішують декілька основних квіткових операторів, в яких дозволено проводити пошук за певними критеріями та фільтрами. Розглянемо деякі з існуючих квіткових операторів, а саме їх системи підбору музичних концертів.

Першою є система концертного оператора Concert.ua [6], який є найбільшим на українському ринку. Ця система має найбільший перелік критеріїв, а саме – жанр, локацію, місто та дату проведення, завдяки чому серед існуючих операторів має найбільш точний підбір, проте враховуючи базу існуючих концертів, навіть після задання критеріїв, користувачу треба витратити не мало часу для того, щоб дослідити отримані концерти та вибрати для себе необхідний.

Вигляд головного вікна сайту концертного оператора Concert.ua зображено на рисунку 1.1.

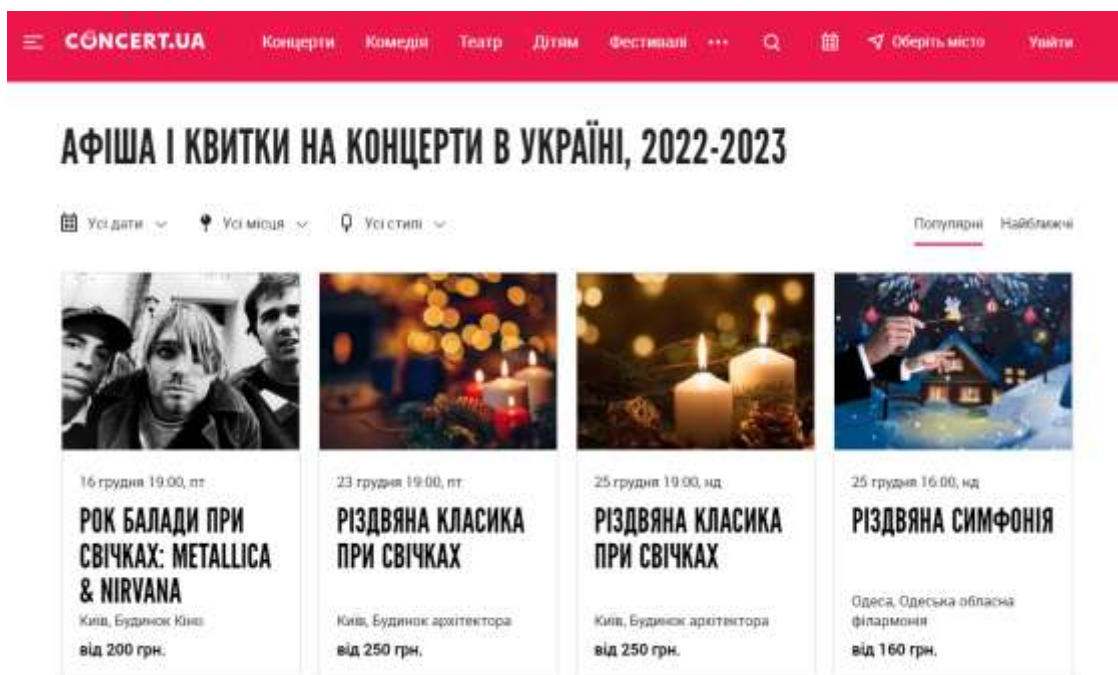


Рисунок 1.1 – Вигляд головного вікна сайту концертного оператора Concert.ua

Наступною є система оператора Karabas.com [7], яка першою вийшла на український ринок і є головним конкурентом першої системи. Головним її недоліком є те, що система пошуку не враховує жанр бажаного виконавця та локацію проведення, через що ми отримуємо ще більше існуючих концертів, ніж в першому варіанті, що робить підбір зовсім не точним.

Вигляд головного вікна сайту концертного оператора Karabas.com зображено на рисунку 1.2.

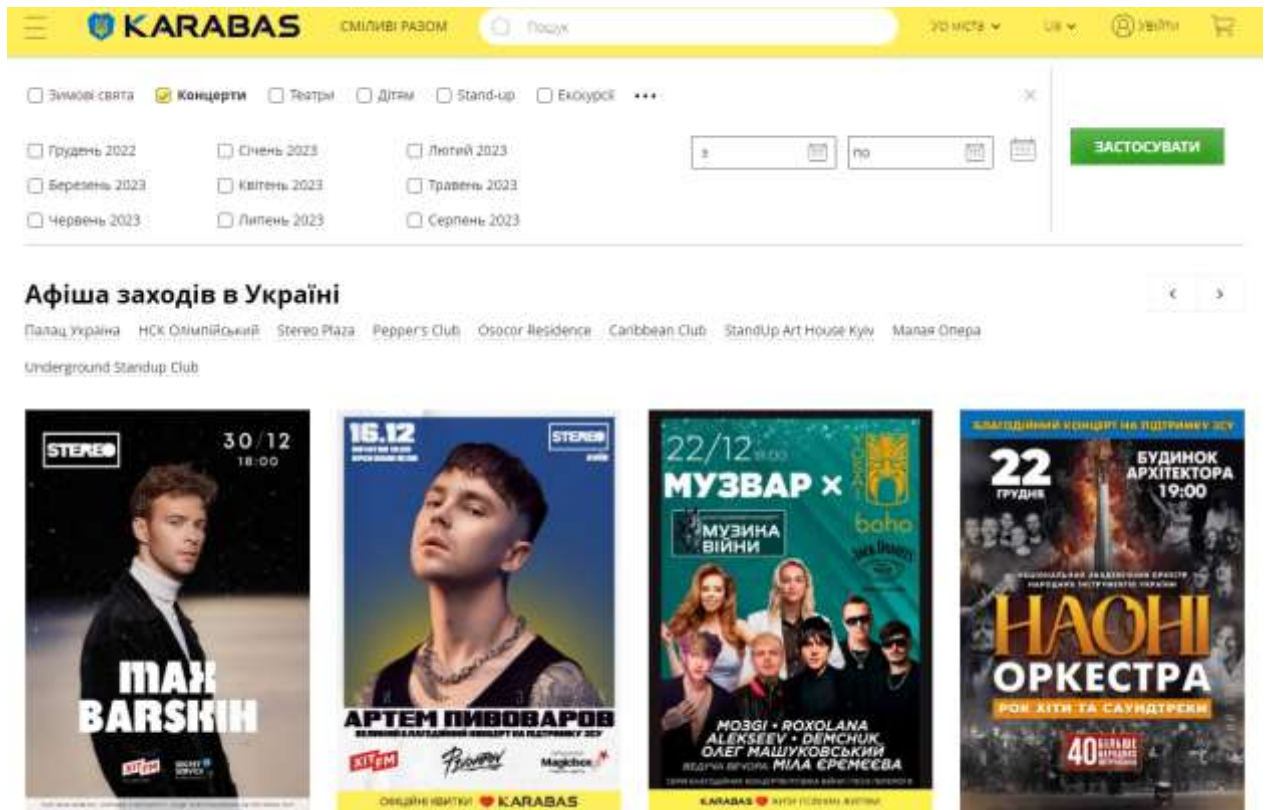


Рисунок 1.2 – Вигляд головного вікна сайту концертного оператора Karabas.com

Розглянемо також систему квиткового оператора Kasa.in.ua [8]. Цей квитковий оператор, який особливо поширений на західній частині України, є самим молодим на ринку, але має найбільший досвід серед стадіонних концертів. Система даного оператора має майже повний перелік існуючих критеріїв для вибору музичного концерту, окрім локації проведення. Таким

чином, при порівнянні, вона є значно кращою за Karabas.com, але гірше лідера ринку – Concert.ua.

Вигляд головного вікна сайту концертного оператора Kasa.in.ua зображено на рисунку 1.3.

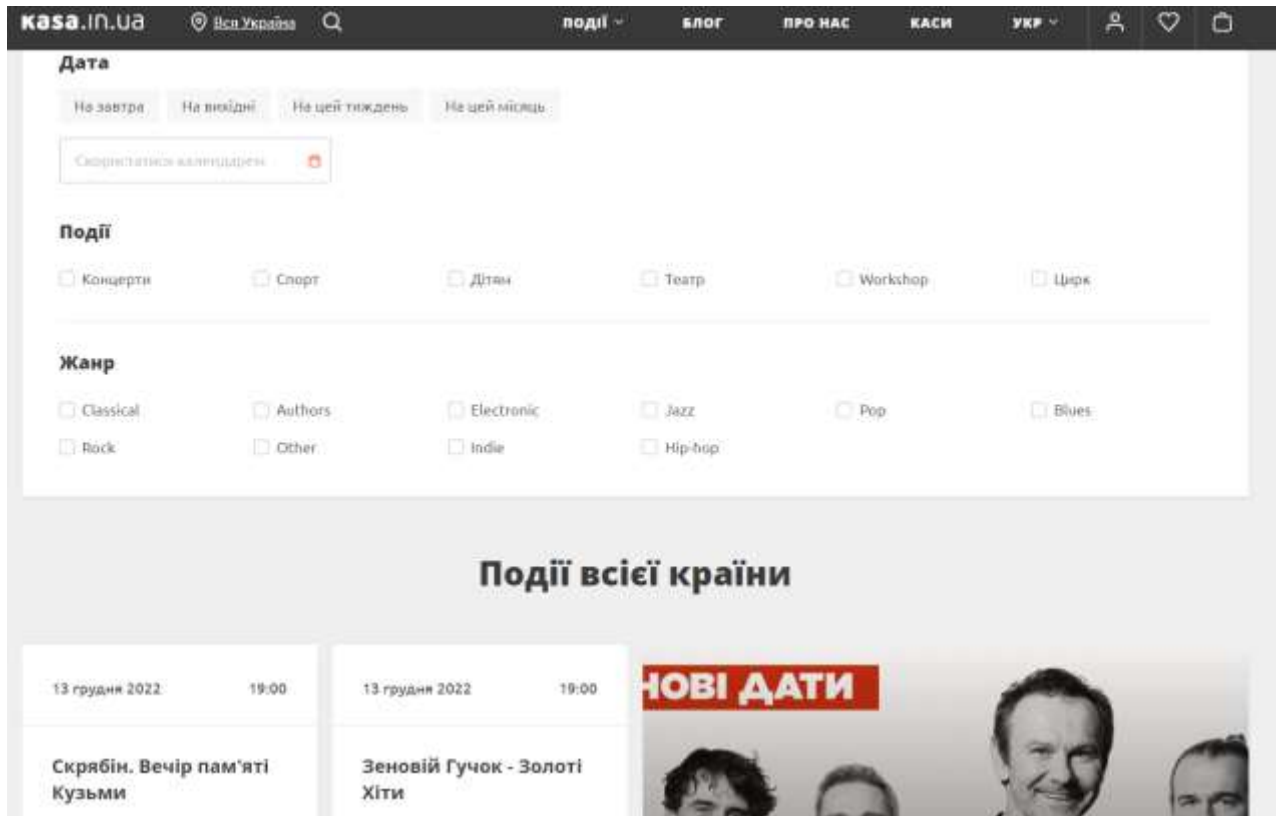


Рисунок 1.3 – Вигляд головного вікна сайту концертного оператора Kasa.in.ua

Усі вищенаведені системи пошуку стикаються з проблемою недостатньої точності підбору. Тому стоїть проблема створення інформаційної технології із веб-інтерфейсом та механізмом децентралізованого доступу до отриманої інформації клієнта з розширеною кількістю параметрів для надання більш точної рекомендації щодо підбору музичного концерту.

Порівняльна характеристика наведених аналогів приведена у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика існуючих систем пошуку музичних концертів квиткових операторів

Характеристика	Concert.ua	Karabas.com	Kasa.in.ua
Наявність пошуку за жанром виконавця	+	-	+
Наявність пошуку за містом проведення	+	+	+
Наявність пошуку за локацією проведення	+	-	-
Наявність пошуку за датою проведення	+	+	+
Візуалізація даних	+	+	+
Наявність пошуку за мовою виконання	-	-	-
Наявність пошуку за бюджетом відвідувача	-	-	-
Універсальність рішення	-	+	-

Можна побачити, що існуючі рішення є вузькоспеціалізованими та не покривають більшість проблем надання точної рекомендації щодо підбору музичних концертів. Тому створення компромісної системи може вирішити вищенаведені проблеми існуючих програмних рішень.

Таким чином, у даному розділі визначено актуальність та неоднозначність проблеми надання рекомендацій щодо підбору музичних концертів, визначено основні проблеми існуючих рішень. Зокрема, головною проблемою даної сфери є відсутність системи, яка би загально збільшила ефективність надання рекомендацій щодо підбору концертів через недостатність існуючих критеріїв. З цього можна сформулювати основну задачу даної роботи – розробку компромісного рішення, що збирає повну необхідну інформацію про користувача, яка дозволить універсально проводити процес

надання рекомендацій щодо підбору музичних концертів. Також розроблене рішення має вирішити питання інтеграції системи в існуючі програмні продукти для швидшого впровадження і використання програмних продуктів, що збільшить коло зацікавлених споживачів та можливу фінансову вигоду від розповсюдження.

1.3 Постановка задачі і формулювання вимог до інформаційної технології

Визначимо задачі і вимоги до інформаційної технології для надання рекомендацій щодо підбору музичного концерту. Задача: розширення функціоналу надання рекомендацій щодо підбору музичного концерту шляхом створення системи підбору музичних концертів з додаванням нових критеріїв для більш точного результату.

Інформаційна технологія має реалізовувати такі функції:

- наявність всіх необхідних критеріїв для пошуку;
- збір детальної інформації про користувача;
- аналіз та збереження детальної інформації про користувачів;
- доступ до фрагменту бази даних концертів;
- візуалізація даних.

Також необхідно провести тестування розробленої технології на наявність помилок, безпечність та коректність роботи. Розроблене програмне забезпечення має функціонувати у всіх браузерах. Усі наявні елементи повинні чітко працювати без будь-яких помилок.

Початком розробки інформаційної технології для надання рекомендацій щодо підбору музичного концерту є аналіз систем-аналогів та методів надання рекомендацій, а також існуючих критеріїв пошуку та впровадження нових. Далі – розробка загальної структурної схеми системи, моделювання інформаційної технології з використанням мови UML та розробка узагальненого алгоритму роботи інформаційної технології. Наступним етапом

є вибір методів надання рекомендацій та безпосередня розробка інформаційної технології. Заключним етапом є тестування та аналізх результатів розробленої інформаційної технології.

1.4 Висновок до розділу 1

У даному розділі розглянуто проблему надання рекомендацій щодо підбору музичного концерту, а саме недостатність або некоректність необхідної інформації від користувача, на основі якої би відбувався процес надання рекомендації системою підбору. Проаналізовано існуючі системи підбору концерту, а саме web-додатки таких концертних операторів як Concert.ua, Karabas.com та Kasa.in.ua. Показано недоліки даних систем підбору, а саме недостатня точність підбору та обмежений функціонал, так як для більш якісної рекомендації необхідна більша кількість критеріїв. Запропоновано компромісне рішення, яке буде мати розширений функціонал для надання рекомендацій та буде використовувати покращений метод фільтрації інформації для більш детального, точного підбору. Визначено задачі та вимоги до інформаційної технології для надання рекомендацій щодо підбору музичного концерту, а саме наявність необхідних критеріїв пошуку, збір, аналіз та збереження детальної інформації про користувача, доступ до фрагменту бази даних концертів та візуалізація даних, що дозволяє перейти до етапу аналізу існуючих методів надання рекомендацій та розробки інформаційної технології для надання рекомендацій щодо підбору музичного концерту.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ

2.1 Аналіз методів надання рекомендацій

Рекомендаційні системи представляють собою підклас систем фільтрації інформації, це програми та сервіси, що намагаються передбачити які об'єкти (фільми, музика, новини, товари, веб-сайти) будуть цікаві користувачеві, спираючись на деяку інформацію (профіль користувача чи опис об'єкту) [9]. Ці програми вдосконалили способи взаємодії між сервісом та відвідувачем, тому що замість того, щоб надавати статичну інформацію, користувач отримує інтерактивні можливості.

Існують три основні підходи до побудови рекомендаційних систем [10]:

- Підхід на основі вмісту (content-based filtering) передбачає, що про користувачів і про рекомендовані об'єкти відомо досить багато інформації, по якій і будується набір рекомендацій.
- Колаборативна фільтрація (collaborative filtering) – фільтрація, де набір рекомендацій будується виключно на підставі взаємодії користувачів з об'єктами.
- Гібридний підхід (hybrid filtering) використовує композиції алгоритмів, заснованих на ознакових описах і результатах колаборативної фільтрації.

Методи фільтрації на основі вмісту засновані на описі елемента та профілі переваг користувача [11].

Фільтрування на основі вмісту (content-based filtering) – це тип системи рекомендацій, яка намагається вгадати, що може сподобатися користувачеві на основі його активності. Фільтрація на основі вмісту дає рекомендації за допомогою ключових слів і атрибутів, призначених об'єктам у базі даних, і зіставляючи їх із профілем користувача. Профіль користувача створюється на основі даних, отриманих від дій користувача, таких як покупки, оцінки,

завантаження, пошук товарів на веб-сайті та/або розміщення в кошику та натискання посилань на продукти [12].

Ця модель легко масштабується завдяки невеликій кількості даних. Більше того, оскільки, на відміну від інших моделей, цю модель не потрібно порівнювати з даними інших користувачів, вона може запропонувати специфічні результати для поточного користувача. Однак ця модель вимагає достатнього обсягу знань предметної області від людей, які приписують продуктам функції. Отже, його точність значною мірою залежить від точності знання галузі. Крім того, фільтрація на основі вмісту значною мірою залежить від раніше відомих інтересів користувачів. Тому він обмежений тим, що не може розширити відомі інтереси користувачів [13].

Ці методи найкраще підходять для ситуацій, коли є відомі дані про об'єкт (ім'я, місце розташування, опис тощо.), але не про користувача. Контент-орієнтовані рекомендації розглядають рекомендації як специфічну для користувача проблему класифікації створюють класифікатор для симпатій і антипатій користувача на основі характеристик продукту. Схема методу фільтрації інформації на основі вмісту побажань користувача показана на рисунку 2.1.

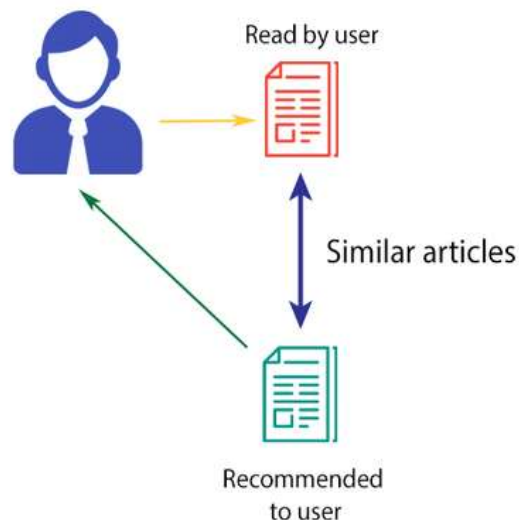


Рисунок 2.1 – Схема методу фільтрації інформації на основі вмісту побажань користувача

de Read by users – прочитані користувачем, Similar articles – подібні статті, Recommended to user – рекомендовані користувачу.

Метод спільної фільтрації для систем рекомендацій (collaborative filtering) — це метод, який базується виключно на минулих взаємодіях, які були записані між користувачами та елементами, щоб створити нові рекомендації. Спільна фільтрація намагається знайти те, чого хотіли б подібні користувачі, і надати рекомендації, щоб класифікувати користувачів у кластери подібних типів і рекомендувати кожному користувачеві відповідно до переваг його кластера. Основна ідея полягає в тому, що завдяки попереднім взаємодіям між користувачем і елементом під час обробки системою стає достатньо виявити схожих користувачів або подібні елементи, щоб зробити прогнози на основі цих оцінених фактів і ідей. Такі підходи, засновані на пам'яті, безпосередньо працюють із значеннями записаних взаємодій або даними і, по суті, базуються на пошуку найближчих сусідів, тобто пошуку найближчих користувачів від користувача, який вас цікавить, і пропонують найпопулярніші елементи серед цих сусідів [14].

Спільна фільтрація – це трьохступеневий процес, що починається зі збору користувацької інформації, де потім будується матриця для розрахунку асоціацій і, нарешті, дається дуже вірогідна рекомендація. Її основне припущення полягає в наступному: ті, хто однаково оцінював будь-які предмети в минулому, схильні давати схожі оцінки інших предметів і в майбутньому.

Наприклад, за допомогою колаборативної фільтрації музичний додаток здатен прогнозувати, яка музика сподобається користувачу, маючи неповний список його переваг (симпатій і антипатій). Прогнози складаються індивідуально для кожного користувача, хоча використовувана інформація зібрана від багатьох учасників [15].

Розглянемо переваги та недоліки колаборативної фільтрації [16].

Переваги:

- Немає необхідності в знаннях домену. Нам не потрібні знання предметної області, тому що вбудовування вивчаються автоматично.
- Інтуїція. Модель може допомогти користувачам відкрити для себе нові інтереси. В окремому випадку система може не знати, що користувач зацікавлений у певному предметі, але модель все одно може рекомендувати його, оскільки схожі користувачі зацікавлені в цьому предметі.
- Чудова відправна точка. Певною мірою системі потрібна лише матриця зворотного зв'язку для навчання моделі. Зокрема, системі не потрібні контекстні функції. На практиці це можна використовувати як один із кількох генераторів-кандидатів.

Недоліки:

- Не можна обробляти свіжу інформацію. Прогноз моделі для даної пари є скалярним добутком відповідних вкладень. Отже, якщо елемент не відображається під час навчання, система не може створити для нього вбудовування та не може запитати модель із цим елементом. Цю проблему часто називають проблемою холодного запуску.
- Важко включити сторонні функції для запиту/елемента. Сторонні функції – це будь-які функції за межами запиту чи ідентифікатора елемента. Наявність додаткових функцій покращує якість моделі.

Порівняємо ці два методи фільтрації. Список пунктів, які відрізняють підхід на підставі ознакових описів з колаборативною фільтрацією [17]:

- Підхід, заснований на вмісті, вимагає достатньої кількості інформації про функції елементів, а не використання взаємодії та відгуків користувачів. Спільна фільтрація, з іншого боку, не потребує нічого іншого, окрім історичних уподобань користувача щодо набору елементів для рекомендації, і оскільки вона базується на історичних даних, основним припущенням є те, що користувачі, які вподобали щось предмет в минулому, будуть мати тенденцію вподобання до цього предмету в майбутньому.

- Знання предметної області у випадку спільної фільтрації не є необхідним, оскільки вбудовування вивчаються автоматично, але у випадку підходу на основі вмісту, оскільки представлення функцій елементів певною мірою розроблено вручну, ця техніка вимагає багатьох предметних знань, якими потрібно оперувати.
- Алгоритм колаборативної фільтрації використовує лише поведінку користувача для рекомендації елементів, тоді як для фільтрації на основі вмісту ми повинні знати вміст як користувача, так і елемента.

Схема методу колаборативної фільтрації інформації на основі побажань користувачів показана на рисунку 2.2.

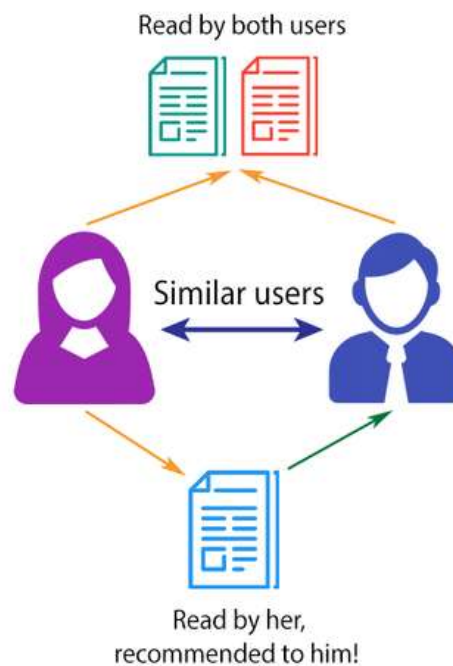


Рисунок 2.2 – Схема методу колаборативної фільтрації інформації на основі побажань користувачів

де Read by both users – прочитано двома користувачами, Similar users – схожі користувачі, Read by her, recommended to him! – прочитано нею, рекомендовано йому.

Гібридний підхід – це суміш методів колаборативної фільтрації та методів фільтрації на основі вмісту під час надання пропозицій, де також враховується контекст предметної області, на основі якої виконується підбір.

Відношення користувача до елемента та відношення користувача до користувача також відіграють важливу роль під час рекомендації. Ця структура дає рекомендації відповідно до знань користувача, надає унікальні рекомендації та вирішує проблему, якщо конкретний користувач ігнорує відповідні дані. Дані профілю користувача збираються з веб-сайту, контекст предмету також враховує вподобання користувача і інформацію про речі, які йому подобаються [18].

Якщо порівнювати цю систему з іншими підходами, ця система має вищу точність пропозицій. Основною причиною є відсутність інформації про доменні залежності фільтрації та інтерес людей до системи на основі вмісту. Коли ці два підходи працюють разом, можна отримати більше знань, що призведе до кращих результатів. Цей підхід досліджує нові шляхи до основного вмісту та методів спільної фільтрації з даними про поведінку користувачів. Дана фільтрація реалізувала обидві системи та подолала більшість слабких місць алгоритмів кожної системи та покращила продуктивність системи. Також в даному підході використовуються методи класифікації та кластеризації для отримання кращих рекомендацій, що підвищує якість і точність підбору [18].

Схема методу гібридної фільтрації інформації на основі побажань користувача та характеристики предметів показана на рисунку 2.3.

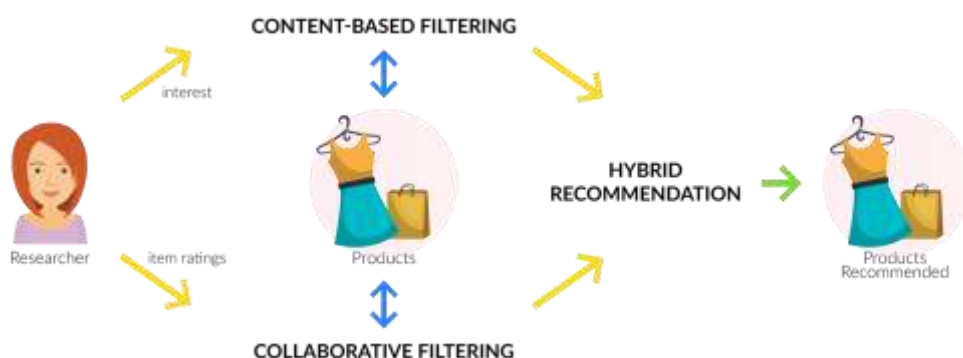


Рисунок 2.3 – Схема методу гібридної фільтрації інформації на основі побажань користувачів та характеристики предметів

де Researcher – користувач, interest – інтерес, item ratings – оцінка продуктів, content-based filtering – фільтрація на основі вмісту, collaborative filtering – колаборативна фільтрація, products – продукти, hybrid recommendation – гібридна фільтрація, products recommended – рекомендовані продукти.

Серед існуючих підходів обрано такий підхід, як фільтрація на основі вмісту, так як він заснований на описі елемента та профілю побажань користувача, якою ми і будемо оперувати.

Існує безліч методів, які виконуються за допомогою фільтрації на основі вмісту, серед яких для виконання інформаційної технології будуть використані методи на основі теорії експертних систем, які будуть розглянені в наступному розділі.

2.2 Обґрунтування вибору підходу до реалізації інформаційної технології

Експертна система (ЕС) – це програмний засіб, що використовує експертні знання для забезпечення високоефективного рішення неформалізованих задач у вузькій предметній області. ЕС являють собою реальний практичний додаток штучного інтелекту, що підкреслено ще одним визначенням: Експертна система - це інформаційна система, заснована на знаннях визначеної комплексної предметної області, що виконує роль експерта-консультанта для кінцевих користувачів [19]. Основу ЕС складає база знань (БЗ) про предметну область, яка накопичується в процесі побудови та експлуатації ЕС. Накопичення і організація знань – найважливіша властивість усіх ЕС.

Одним із факторів, що відрізняє ЕС від традиційних програм є явність та доступність знань. З цього випливають їх наступні властивості:

– використання високоякісного досвіду для вирішення проблем, який представляє рівень знань найбільш кваліфікованих експертів в даній області;

– наявність прогностичних можливостей, при яких ЕС надає відповіді не тільки для конкретної ситуації, а й показує, як змінюються ці відповіді в нових ситуаціях, з можливістю детального пояснення яким чином нова ситуація привела до змін;

– можливість використання ЕС для навчання і тренування керівних працівників, забезпечуючи нових службовців великим багажем досвіду і стратегій, за якими можна вивчати рекомендовану політику і методи.

Одним із перспективних методів буде використання підходу до реалізації інформаційної технології, заснованого на теорії експертних систем, тобто використовуючи формулу Байєса для виконання обрахунку точності підбору музичного концерту до побажань користувача.

Теорема (формула) Байєса – одна з основних теорем елементарної теорії ймовірностей, яка дозволяє визначити ймовірність будь-якої події за умови, що сталася інша статистично взаємозалежна з нею подія. Іншими словами, за формулою Байєса можна більш точно перерахувати ймовірність, врахувавши як раніше відому інформацію, так і дані нових спостережень [20].

Формування формули Байєса можна представити таким чином:

Нехай подія A настає лише за умови появи однієї з несумісних випадкових подій (гіпотез) H_1, H_2, \dots, H_n , що утворюють повну групу випадкових подій. Припустимо, що подія A вже відбулась, і необхідно визначити ймовірність того, що подія A відбулась саме завдяки реалізації гіпотези H_i . На це питання дає відповідь теорема гіпотез, або формула Байєса. Теорему гіпотез можна вважати наслідком теореми множення ймовірностей та формули повної ймовірності.

Ймовірність $P(H_i|A)$, яка визначається як ймовірність гіпотези H_i за умови, що подія A відбулась, називається апостеріорною ймовірністю (на відміну від апріорної ймовірності $P(H_i)$, відомої ще до початку випробувань). Апостеріорна ймовірність обчислюється за формулою Байєса [21]:

$$P(H_i|A) = \frac{P(H_i) \cdot P(A|H_i)}{P(A)}, \quad (2.1)$$

де ймовірність події A визначається за формулою повної ймовірності:

$$P(A) = \sum_{i=1}^N P(H_i) \cdot P(A|H_i) . \quad (2.2)$$

Сучасний метод статистичного аналізу, застосування якого зараз поширюється в економічних дослідженнях, спирається на саме поняття апостеріорної ймовірності та використання формули Байєса. Відомо, що одним із завдань теорії ймовірностей є розробка інструментарію статистичного аналізу, який допомагає приймати рішення в умовах невизначеності. Байєсівський аналіз відрізняється від класичної статистики припущенням, що параметри розподілів не є постійними, а випадковими змінними. Ймовірність Байєса можна легко зрозуміти, якщо розглядати її як ступінь впевненості у відповідній події в протилежність до класичного підходу, заснованого на об'єктивних свідченнях. Оскільки підхід Байєса заснований на суб'єктивній інтерпретації ймовірності, то він може бути корисний при виборі рішення і розробки мереж довіри.

Система надання рекомендацій щодо підбору музичного концерту являє собою експертну систему, яка використовує байєсівську систему логічного висновку, тобто заснована на теоремі гіпотез (формулі Байєса).

2.3 Розробка загальної структурної схеми функціонування інформаційної технології

Розробимо загальну структурну схему інформаційної технології надання рекомендацій щодо підбору музичних концертів, а також детальну структурно-функціональну схему роботи інформаційної технології.

Інформаційна технологія складається з трьох основних компонентів:

- програмний інтерфейс додатку;
- модуль надання рекомендацій;
- база даних.

Опишемо кожен із компонентів. Програмний інтерфейс включає в себе обробку даних та виведення інформації, отриманою інформаційною технологією.

Інформаційна технологія надання рекомендацій виконує функцію аналізу отриманих побажань та можливостей користувача та функцію підбору найбільш підходящого концерту згідно заданих критеріїв.

База даних виконує функцію збереження даних про існуючі концерти та їх характеристики, такі як:

- назва гурту;
- жанр;
- місто проведення;
- локація проведення;
- ціна квитка на концерт;
- мова виконання.

Узагальнена структурна схема інформаційної технології показана на рисунку 2.4.

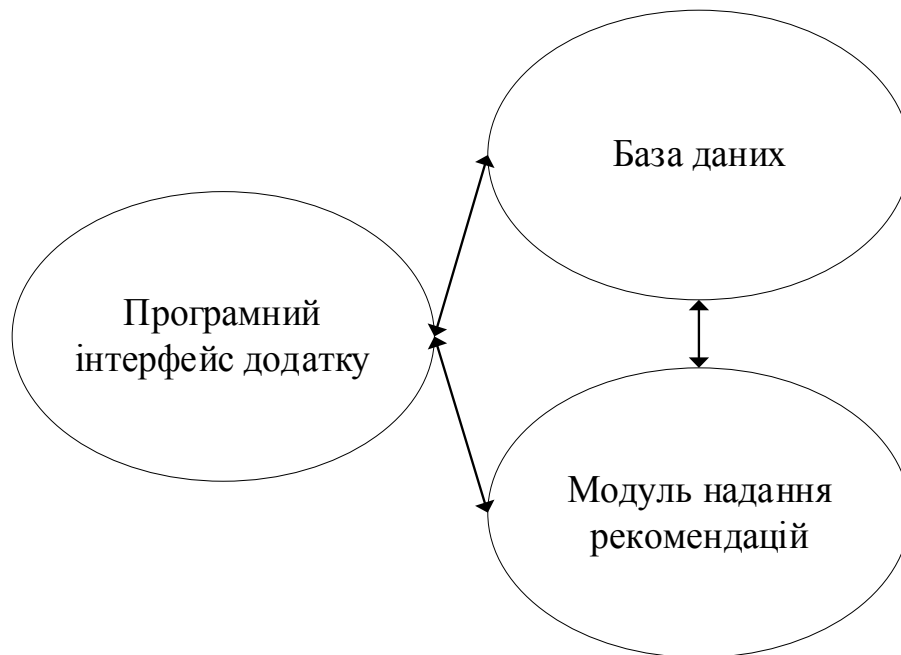


Рисунок 2.4 – Узагальнена структурна схема інформаційної технології надання рекомендації щодо підбору музичного концерту

Детальна структурно-функціональна схема роботи інформаційної технології показана на рисунку 2.5



Рисунок 2.5 – Детальна структурно-функціональна схема роботи інформаційної технології

База існуючих концертів являє собою всі наявні концерти та інформацію про них, на основі яких разом з побажаннями користувача буде виконуватися надання рекомендації для підбору музичного концерту.

Інформаційна технологія надання рекомендацій щодо підбору музичного концерту виконується в декілька процесів, які розглянуті нище:

- Вибір способу підбору являє собою вибір серед двох методів – на основі концертів, які користувач відвідував раніше, або наявних побажань користувача;

- Питання до користувача можуть бути різними, в залежності від того, чи відвідував він концерти раніше. Якщо так, користувачу треба вибрати з бази концертів той концерт, на якому він був раніше. В іншому випадку користувачеві буде представлений лист запитань на тему його вподобань;
- Вибір заданих критеріїв виконується після того, як користувач відповів на питання і натиснув кнопку, тим самим підтверджуючи обрані варіанти відповідей;
- Процес обробки критеріїв відбувається одразу після того, як користувач натиснув кнопку підбору концерту, тим самим таким чином записується інформація про необхідні критерії;
- Процес виконання підбору концерту виконується в програмі за допомогою формули Байеса, яка обраховує точність подібності заданих критеріїв користувача до кожного наявного в базі даних концерту;
- Процес виведення результату виконується після обрахунку подібності критеріїв до кожного концерту, наявного в базі даних, підібравши лише ті, які мають найвищу оцінку подібності.

Виведення результату являє собою список концертів на web-додатку, де також описана повна інформація про кожен рекомендований концерт разом з його оцінкою подібності до критеріїв, заданих користувачем.

2.4 Моделювання інформаційної технології надання рекомендацій щодо підбору концертів із використанням мови UML

Модель – це абстракція, яка створюється з метою осмислення чого завгодно перед тим, як його створювати. Абстрагування – це вибіркоче вивчення деяких аспектів проблеми. Основна мета абстрагування полягає в тому, щоб ізолювати аспекти, важливі для деякої цілі, і відкинути всі інші [22].

UML (англ. Unified Modeling Language) — уніфікована мова моделювання. Є невід'ємною частиною уніфікованого процесу розробки

програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю [23].

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки додатків. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем. Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність реалізації програмних систем у кілька разів і помітно поліпшити якість кінцевого продукту [23].

На рисунку 2.6 зображено UML-діаграму класів інформаційної технології для надання рекомендацій щодо підбору музичного концерту.

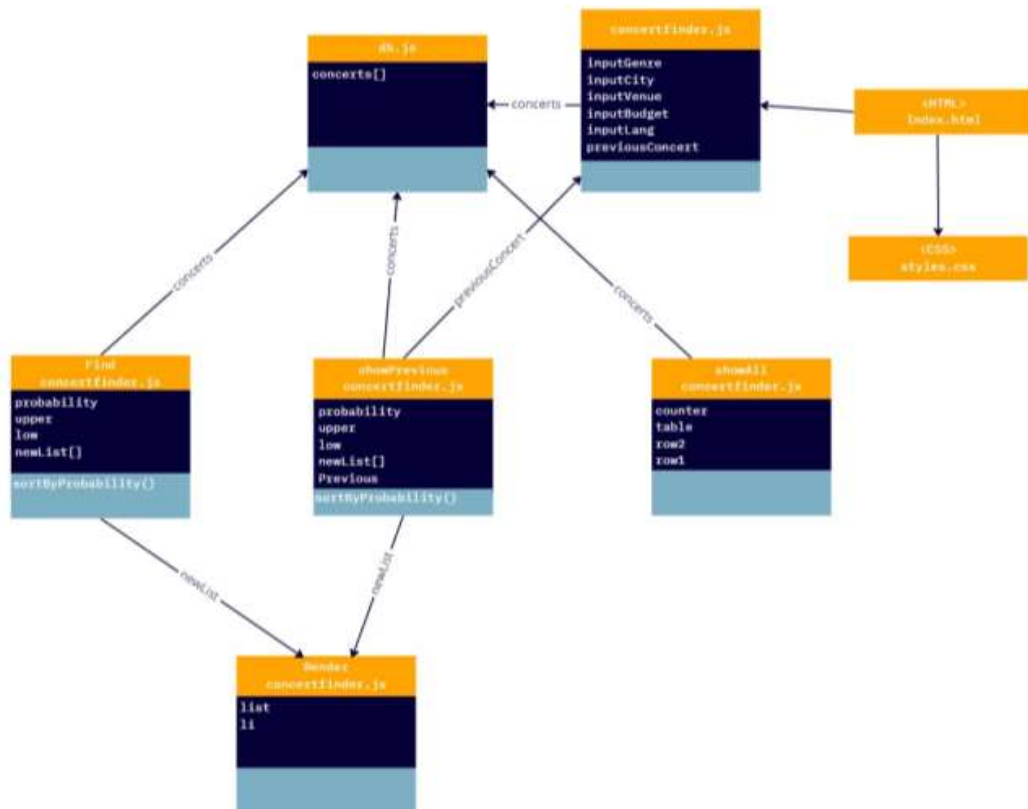


Рисунок 2.6 – UML-діаграма класів інформаційної технології для надання рекомендацій щодо підбору музичного концерту

Розглянемо детальніше опис кожного з класів:

Клас Find відповідає за виконання обрахунків та підбір концерту за заданими критеріями. Цей клас містить всі необхідні критерії, які враховуються для надання рекомендацій, та враховує ймовірність достовірності підбору щодо заданих даних, а також включає методи визначення коефіцієнтів важливості характеристик.

Клас Render виконує функцію виводу на екран концертів, які найбільше підходять, враховуючи задані критерії.

Клас Concerts виконує функцію бази даних, яка містить всі наявні концерти та інформацію про них.

Клас showAll відповідає за функцію виведення інформації про усі наявні в базі даних концерти.

Клас showPrevious відповідає за виконання обрахунків та підбір концерту на основі інформації про концерт, відвіданий користувачем раніше.

Змодельована в даному розділі UML-діаграма класів інформаційної технології для надання рекомендацій щодо підбору музичного концерту надає достатньо інформації про компоненти технології та принцип їх роботи, щоб перейти до етапу розробки узагальненого алгоритму роботи інформаційної технології.

2.5 Розробка узагальненого алгоритму роботи інформаційної технології надання рекомендацій щодо підбору музичного концерту

Для відображення циклу роботи програмного забезпечення використаємо текстовий опис та опис у вигляді схеми алгоритму роботи інформаційної технології надання рекомендацій щодо підбору музичного концерту.

Схема узагальненого алгоритму роботи інформаційної технології для надання рекомендації щодо підбору музичного концерту зображена на рисунку 2.7.



Рисунок 2.7 – Схема узагальненого алгоритму роботи інформаційної технології для надання рекомендацій щодо підбору музичного концерту

Кроки алгоритму:

1. Відкриття вкладки програми в меню
2. Вибір способу, на основі яких критеріїв буде виконуватись підбір музичного концерту

3. Введення можливостей та побажань користувача в систему.
4. Оброблення даних.
5. Виконання підбору концерту за теоремою Байєса та пошук найбільш підходящих концертів під побажання користувача.
6. Збереження даних у базі даних.
7. Виведення підібраних концертів під введені дані.
8. Завершення роботи.

2.6 Висновок до розділу 2

Таким чином, в даному розділі проаналізовано існуючі методи надання рекомендацій, такі як фільтрація на основі вмісту, колаборативна фільтрація та гібридна фільтрація. Проведено порівняння та визначено переваги та недоліки кожної з них, в результаті чого для розробки інформаційної технології обрано метод фільтрації на основі вмісту разом з використанням експертної системи як методу, що забезпечує надання рекомендацій щодо підбору музичного концерту. Виділено основні компоненти інформаційної технології надання рекомендацій щодо підбору музичного концерту, такі як база даних, програмний інтерфейс додатку та модуль надання рекомендацій. Розроблено детальну структурно-функціональну схему роботи інформаційної технології, яка виконує такі процеси як вибір способу підбору, формулювання питань до користувача, вибір критеріїв, процес обробки критеріїв, виконання підбору концерту та процес виведення результату. Проведено моделювання інформаційної технології для надання рекомендацій щодо підбору музичного концерту із використанням мови UML, яка містить такі класи, як Find, який відповідає за виконання обрахунків та підбір концерту за заданими критеріями, Render, який виконує функцію виводу на екран концертів, які найбільше підходять під критерії, Concerts, який виконує функцію бази даних, showAll, який виконує функцію виведення інформації про усі концерти, showPrevious, який відповідає за виконання обрахунків та підбору на основі

інформації про концерт, відвідуваний користувачем раніше. Розроблено узагальнений алгоритм роботи інформаційної технології, який містить такі кроки, як відкриття вкладки програми, вибіру способу підбору, введення побажань користувача, оброблення даних, виконання підбору концерту за теоремою Байеса, пошук найбільш підходящих концертів та виведення підібраних концертів під введені дані, що дозволяє перейти до програмної реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ

3.1 Обґрунтування вибору мови для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту

Визначимо інструменти, за допомогою яких буде створюватись програмна реалізація інформаційної технології для надання рекомендацій щодо підбору музичного концерту. Для реалізації програми необхідно використати одну зі сучасних об'єктно-орієнтованих мов програмування : C++, C#, Java, Delphi, Python, JavaScript. Крім цього, необхідно підібрати мову програмування для того, щоб створити веб-клієнт, програмний інтерфейс, універсальну функціональну систему підбору, а також систему керування базами даних для збереження даних.

Оберемо три мови програмування для аналізу та порівняння: C++, JavaScript, Python.

C++ – це мова програмування високого рівня з підтримкою одразу кількох парадигм програмування: узагальненої, процедурної та об'єктно-орієнтованої. Цю мову використовують як для системного програмування та розробки програмного забезпечення, так і для написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. З появою перших трансляторів мова C++ знайшла відразу дуже широке розповсюдження, на ній було створено величезну кількість програм і застосунків. Але у міру накопичення досвіду створення великих програмних систем спливали недоліки, які спонукали до пошуку альтернативних рішень [24].

Python – ще одна мова програмування, що швидко розвивається. Ця мова має доволі зрозумілий і простий синтаксис, що значно спрощує процес

розробки. Має динамічну типізацію та автоматичний контроль пам'яті та механізм обробки виключень. Мова підтримується великою кількістю бібліотек, як, наприклад, Django, MongoDB і т.д. Найчастіше використовується для створення веб-додатків. Використання бібліотек і низький поріг входження зробили Python однією з найбільш популярних мов програмування [25].

JavaScript – об'єктно-орієнтована мова програмування, що розвивається найбільш динамічно. Ця мова частіше всього використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [26].

Гнучкість цієї мови дозволяє використовувати багато шаблонів програмування в залежності від конкретних умов. У JavaScript взагалі відсутня статична типізація, тобто в момент оголошення не вказується тип змінної, він присвоюється лише разом із присвоєнням або зміною значення, що значно полегшує роботу з несумісністю типів. Прийняття нових стандартів призвело до спрощення синтаксису JavaScript і додало класи, що робить JavaScript повноцінною об'єктно-орієнтованою мовою.

У таблиці нижче була наведена порівняльна характеристика описаних в розділі мов програмування [27].

Таблиця 3.1 – Порівняльна характеристика мов програмування

Критерій/мова	C#	JavaScript	Python
Динамічна типізація	+	+	+
Ручне управління пам'яттю	+	-	-
Статична типізація	+	-	-
Необхідність компіляції	+	-	-
Необхідність у роботі з браузером	-	+	-

Продовження табл. 3.1

Наявність пакетів для машинного навчання	-	+	+
Підтримка на всіх платформах	-	+	-
Виконання на клієнтській частині браузера	-	+	-
Пряме підключення скриптів	-	+	-
Використання на сервері	+	+	+

Виходячи з такої необхідності як розробка веб-додатку, для створення програмного забезпечення інформаційної технології обрано мову програмування JavaScript, тому що вона є доволі універсальною, що зменшує ресурси і час розробки.

3.2 Обґрунтування вибору середовища програмування для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту

Також необхідно обрати середовище для розробки програмного засобу серед таких, як NetBeans, JetBrains Webstorm, Visual Studio Code, IntelliJ IDEA, Eclipse та MS Visual Studio [28].

Для порівняння обрано такі середовища: NetBeans, JetBrains Webstorm, Visual Studio Code та MS Visual Studio.

NetBeans пропонує легку інтеграцію з програмним забезпеченням для управління версіями. NetBeans був створений з відкритим вихідним кодом, тому кожен може внести свій вклад в його розвиток. Він безкоштовний і працює на більшості операційних систем і платформ. Також він автоматично виділяє синтаксичні конструкції, наявність підказок та довідок для стандартних методів у мовах програмування [29].

JetBrains Webstorm – інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, яке розроблене на основі платформи IntelliJ IDEA. WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей. WebStorm постачається з перед-установленим плагінами JavaScript. WebStorm забезпечує авто доповнення, аналіз коду в процеси, навігацію по коду та інтеграцію з системами управління версіями. Важливою перевагою інтегрованого середовища розробки WebStorm також є робота з проектами. Перевагами є вбудований Debugger, яким з легкістю можна перевірити коректність написаного коду, а також можна встановити точки зупинки. Середовище також має простий уніфікований інтерфейс для роботи з Git і GitHub та має зручний, приємний для роботи інтерфейс. Webstorm пропонує велику кількість готових шаблонів веб-додатків, що спрощує первинне налаштування системи [30]. Єдиним недоліком WebStorm є платна основа розповсюдження.

Microsoft Visual Studio — це повнофункціональне інтегроване середовище розробки із підтримкою популярних мов програмування [31]. Функціональність цього середовища охоплює всі етапи розробки програмного забезпечення, надаючи сучасні інструменти для написання коду, проектування графічних інтерфейсів, створення, налагодження та тестування програм. Також його можливості можна доповнити підключенням необхідних розширень. Редактор коду середовища підтримує підсвічування синтаксису, вставлення фрагментів коду, відображення структури та пов'язані функції. Вбудований налагоджувач MS Visual Studio використовується для пошуку та виправлення помилок у вихідному коді, в тому числі на низькому апаратному рівні. Засоби діагностики дозволяють оцінити якість коду з точки зору продуктивності та використання пам'яті. Конструктор форм Visual Studio незамінний при розробці програм з графічним інтерфейсом, допомагаючи спроектувати зовнішній вигляд майбутньої програми та роботу кожного елемента інтерфейсу. Нарешті, Visual Studio надає набір інструментів для автоматизації тестування додатків з точки зору перевірки роботи інтерфейсів,

модульного та навантажувального тестування. Для командних проектів Visual Studio пропонує підтримку групової роботи, дозволяючи спільно редагувати та налагоджувати будь-яку частину коду в режимі реального часу, а також використовувати Team Foundation або Git як систему контролю версій [32].

Visual Studio Code (VS Code) — це спрощений, але потужний редактор вихідного коду, який запускається на комп'ютері й доступний для Windows, macOS і Linux. Він підтримує JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов (наприклад, C++, C#, Java, Python, PHP і Go) і середовищ виконання (наприклад, .NET і Unity). VS Code надає додаткові можливості завдяки розширенням. Розширення VS Code можуть вносити додаткові функції до наявного набору. [33]

Visual Studio Code має деякі унікальні функції. Вони наведені нижче [34]:

- Підтримка декількох мов програмування;
- Intelli-Sense. Ця функція виявляє, якщо який-небудь фрагмент коду залишився неповним. Також загальні синтаксиси змінних та оголошення змінних складаються автоматично;
- Підтримка крос-платформ;
- Розширення та підтримка. Visual Studio Code підтримує всі мови програмування, але якщо користувач хоче використовувати мову, яка не підтримується, він може завантажити розширення;
- Репозиторій. Visual Studio Code пов'язаний з Git або може бути з'єднаний з будь-яким іншим сховищем для витягування або збереження екземплярів;
- Web-підтримка;
- Ієрархічна структура;
- Поліпшення коду. Деякі фрагменти коду можна оголосити інакше, що може допомогти користувачеві в коді;
- Підтримка терміналу;
- Мультипроектність;

- Коментування.

Таким чином, середовищем розробки програми в межах даної магістерської роботи обрано Visual Studio Code.

3.3 Обґрунтування вибору системи керування базами даних для реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту

Розглянемо три системи управління базами даних: MongoDB, MySQL та PostgreSQL.

MongoDB — це система керування базами даних NoSQL з відкритим вихідним кодом; він не лише забезпечує легкість у зберіганні неструктурованих даних, але й керує ними. Щоразу, коли виникає питання про те, як керувати мільйонами рядків неструктурованих даних, рекомендується MongoDB [35].

MongoDB — це база даних NoSQL, яка використовується для зберігання даних у формі документів JSON, тому що в MongoDB можна легко зберігати і керувати величезною кількістю неструктурованих даних. База даних MongoDB належить до баз даних документів, що означає, що вона зберігає дані відповідно до документів JSON. Він не відповідає жодній схемі і дозволяє вставляти в неї будь-які типи даних. Набір цих документів, що містять дані, відомий як колекції. І ці колекції подібні до таблиць, які використовуються в реляційних базах даних для зберігання даних.

Існує багато причин, чому MongoDB настільки популярний у порівнянні з іншими реляційними та NoSQL базами даних, деякі з них [35]:

- розробники можуть визначити структуру відповідно до пар ключ-значення;
- як і реляційні бази даних, рядки та стовпці для даних не потрібні;
- ієрархічна структура MongoDB дозволяє зберігати дані у вигляді масивів, а також може зберігати складні дані;

- це дозволяє розробникам легко керувати базами даних, оскільки підтримує кілька мов програмування
- він також підтримує функції gridFS та реплікації.

MySQL – це одна з найпопулярніших реляційних баз даних. Цю СУБД часто включають до складу серверів. MySQL працює на всіх основних платформах, включаючи Red Hat, Fedora, Ubuntu, Debian, Solaris, Microsoft Windows та Apple MacOS. Реалізована підтримка кількох процесорів та багатопоточність. Через API з базами даних можна працювати за допомогою популярних мов програмування (C, C++, C#, PHP, Java, Ruby, Python та Perl). Бібліотека функцій SQL реалізується через високооптимізовану бібліотеку класів, завдяки чому досягається висока швидкість роботи і відсутні витрати пам'яті. У MySQL реалізовано підтримку Novell Cluster [36].

MySQL є системою клієнт-сервер, яка містить SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API). Також постачається сервер MySQL у вигляді багато поточної бібліотеки, яку можна підключити до призначеного для користувача додатком і отримати компактний, більш швидкий і легкий в управлінні продукт [37].

На жаль, MySQL не часто оновлюється та не підтримує гаряче резервне копіювання. MySQL часто критикують за недостатню надійність, а, крім того, вона має проблеми з використанням Unicode.

PostgreSQL - це система управління об'єктно-реляційними базами даних з відкритим кодом, ORDBMS, яка не належить або контролюється однією компанією чи особою. Оскільки програмне забезпечення postgresSQL є відкритим кодом, ним керують здебільшого за допомогою скоординованих онлайн-спроб активним світовим співтовариством розробників, ентузіастів та інших добровольців [38].

З PostgreSQL розробники можуть створювати додатки з високим рівнем безпеки. У адміністратора буде великий вибір інструментів роботи з базою даних, якщо він вибере цю СКБД [39].

Ця платформа дає можливість:

- визначати власні набори даних;
- розробляти свої шрифти;
- об'єднувати код, написаний різними мовами програмування.

PostgreSQL добре масштабується з точки зору:

- кількості даних;
- кількості одночасних користувачів в проєкті.

База даних Oracle (Oracle DB) - це система управління реляційними базами даних (RDBMS) від корпорації Oracle. Oracle DB є одним з найбільш надійних і широко використовуваних реляційних двигунів баз даних.

Система побудована на базі реляційних баз даних, в яких користувачі можуть безпосередньо отримувати доступ до об'єктів даних через структуровану мову запитів (SQL). Oracle - це повністю масштабована архітектура реляційних баз даних і часто використовується глобальними підприємствами, які керують та обробляють дані в широких і локальних мережах. База даних Oracle має власний мережевий компонент, який дозволяє здійснювати зв'язок по мережах. Ключовою особливістю Oracle є те, що його архітектура розділена між логічною та фізичною. Ця структура означає, що для широкомасштабних розподілених обчислень, також відомих як сіткові обчислення, розташування даних не має значення та прозорості для користувача, що забезпечує більш модульну фізичну структуру, яку можна додавати та змінювати, не впливаючи на активність бази даних, його дані або користувачів [40].

Після порівняння та аналізу систем управління базами даних, наведених вище, для розробки інформаційної технології обрано СУБД MongoDB, так як вона забезпечує простоту розробки, підтримки та використання при роботі з мовою JavaScript, а також не потребує додаткових інструментів для

приведення даних у необхідний формат, легко масштабується та має гнучку схему даних, що дозволяє спростити подальшу підтримку розробленого модуля.

3.4 Програмна реалізація компонентів інформаційної технології надання рекомендацій щодо підбору музичного концерту

Розглянемо особливості програмної реалізації інформаційної технології надання рекомендацій щодо підбору музичного концерту.

Головним компонентом інформаційної технології є програмний інтерфейс, який забезпечує взаємодію між веб-клієнтом, користувачем, технологією та базою даних, а також виконує аналіз отриманої інформації та функцію підбору концерту на основі цих даних.

Створене головне меню, з якого можна викликати інші вікна: вікно відображення всіх наявних в базі даних концертів та запитання, від якого залежить, яким способом буде надаватися рекомендація – на основі критеріїв чи на основі відвідуваного користувачем концерту раніше.

Фрагмент коду менеджера дій інформаційної технології, який реагує на відповідь користувача:

```
document.getElementById('question').addEventListener('submit', function(e){
  const checkButton = document.getElementById('check');
  if (checkButton.value == 'Yes') {
    document.getElementById("previous").style.visibility = "visible";
  } else
    document.getElementById("find").style.visibility = "visible";
  document.getElementById("check-button").setAttribute("disabled",true);

  e.preventDefault();
});
```


де `previous` – функція пошуку рекомендації на основі відвідуваного концерту, `find` – функція пошуку рекомендації на основі заданих критеріїв.

Для того, щоб відобразити всі наявні концерти в базі даних, необхідно натиснути на кнопку “Show all concerts”. Фрагмент коду, який виконує відображення концертів в базі даних:

```
function showAll(){
  let counter = 0;
  const table = document.querySelector('.all-concerts')
  for (const concert of concerts) {
    const row2 = document.createElement('td')
    row2.innerHTML = `
    <tr>
    Name of the artist: ${concert.name} <br>
    Genre: ${concert.genre} <br>
    City: ${concert.city} <br>
    Venue: ${concert.venue} <br>
    Price: ${concert.budget}$ <br>
    Language: ${concert.lang} <br>
    
    </tr> `
    table.append(row2)
    counter++;
    if (counter %5 == 0) {
      const row1 = document.createElement('tr')
      table.append(row1)}}}
```

де `genre`, `city`, `venue`, `budget`, `lang` – інформація, відома про кожен концерт.

Для того, щоб отримати рекомендацію на основі відвідуваного раніше концерту, необхідно у пункті «Choose a concert you visited before» вибрати необхідний концерт.

Надана рекомендація виконується на основі аналізу критеріїв за допомогою системи підбору, яка побудована на основі теореми Байєса. Фрагмент коду, який виконує систему підбору на основі теореми Байєса:

```
concerts.forEach((concert) => {
  let probability = 1;
```

```

let upper = 0.1;
let low = 0.1;
if (concert.genre == inputGenre.value) {
  low *= 0.5; } else upper *= 0.5;
if (concert.city == inputCity.value) {
  low *= 0.4; } else upper *= 0.4;
if (concert.venue == inputVenue.value) {
  low *= 0.7; } else upper *= 0.7;
if (concert.budget <= inputBudget.value) {
  low *= 0.2; } else upper *= 0.2;
if (concert.lang == inputLang.value) {
  low *= 0.6; } else upper *= 0.6;
probability = (probability * upper) / (upper + low);
concert.probability = probability;
})

```

де `inputGenre`, `inputCity`, `inputVenue`, `inputBudget`, `inputLang` – інформація про концерт, відвідуваний раніше, `probability` – показник вірогідності, що користувачеві підходить рекомендований концерт.

Фрагмент коду, який виконує відображення наданої рекомендації за відвідуваним раніше концертом:

```

let sortByProbability = (a, b) => a.probability <
b.probability ? 1 : -1;
concerts.sort(sortByProbability);
let newList = []
for (const concert of concerts) {
  if (concert.name != previousConcert.value)
  if (concert.probability > 0.5) newList.push(concert)}
function Render(arr) {
  const list = document.querySelector('.previous-list')
  for (const concert of arr) {
    const li = document.createElement('li')
    li.innerHTML = `
<div>Name of the artist: ${concert.name}</div>
<div>Genre: ${concert.genre}</div>
<div>City: ${concert.city}</div>
<div>Venue: ${concert.venue}</div>
<div>Price: ${concert.budget}$</div>
<div>Language: ${concert.lang}</div>
<div>Probability: ${concert.probability}</div>

```

```

    `
    list.append(li)}}
Render(newList)

```

де `previousConcert.value` – назва гурту, концерт якої користувач відвідував раніше.

Для того, щоб отримати рекомендацію на основі заданих критеріїв відвідуваного раніше концерту, необхідно у відповідному вікні вибрати необхідні критерії, які підходять користувачу і натиснути на кнопку «Find».

Фрагмент коду, який виконує відображення наданої рекомендації за критеріями схожий з тим, що відображає рекомендації на основі відвідуваного концерту, але має інший поріг вірогідності:

```

let sortByProbability = (a, b) => a.probability <
b.probability ? 1 : -1;
concerts.sort(sortByProbability);
let newList = []
for (const concert of concerts) {
  if (concert.probability > 0.6) newList.push(concert)}
function Render(arr) {
  const list = document.querySelector('.previous-list')
  for (const concert of arr) {
    const li = document.createElement('li')
    li.innerHTML = `
    <div>Name of the artist: ${concert.name}</div>
    <div>Genre: ${concert.genre}</div>
    <div>City: ${concert.city}</div>
    <div>Venue: ${concert.venue}</div>
    <div>Price: ${concert.budget}$</div>
    <div>Language: ${concert.lang}</div>
    <div>Probability: ${concert.probability}</div>
    `
    list.append(li)}}
Render(newList)

```

Таким чином, переходимо до тестування та аналізу результатів роботи програми надання рекомендацій щодо підбору музичного концерту на основі створених програмних компонентів програми.

3.5 Тестування та аналіз результатів роботи інформаційної технології надання рекомендацій щодо підбору музичного концерту

Визначимо критерії, за якими будемо проводити тестування розробленої програми. Так як метою дослідження є розширення функціональних можливостей в області надання рекомендації щодо підбору музичного концерту, визначимо критерії відповідності програми поставленій меті згідно поставленим задачам дослідження:

- веб-клієнт системи підбору музичного концерту доступний за відповідним посиланням;
- дані про концерти доступні на основній сторінці веб-клієнта;
- дані про введені критерії доступні на основній сторінці веб-клієнта;
- система підбору має більший функціонал за аналоги;
- рекомендації генеруються на основі заданих критеріїв та максимально точно відповідають їм;
- відображається детальна інформація про кожен концерт.

Проведемо тестування відповідно до визначених критеріїв.

Відкриємо веб-клієнт системи підбору музичного концерту. Отримаємо вікно з можливістю переглянути інформацію про всі наявні концерти та питання, залежно від відповіді на яке буде обрано різні способи надання рекомендації. Вигляд головного вікна показаний на рисунку 3.1.

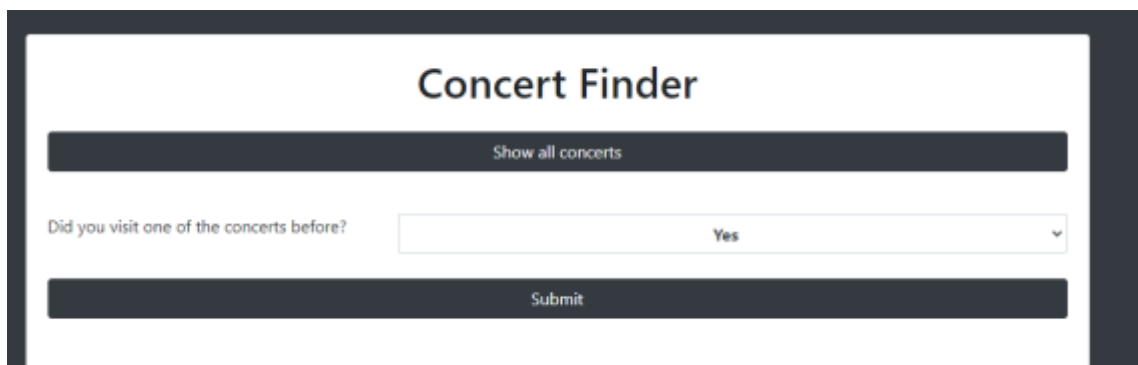


Рисунок 3.1 – Загальний вигляд інтерфейсного вікна «Головне вікно» програми

Переглянемо всі наявні концерти в базі даних та інформацію про них, натиснувши на кнопку «Show all concerts». Вигляд вікна відображення наявних концертів в базі даних показаний на рисунку 3.2.

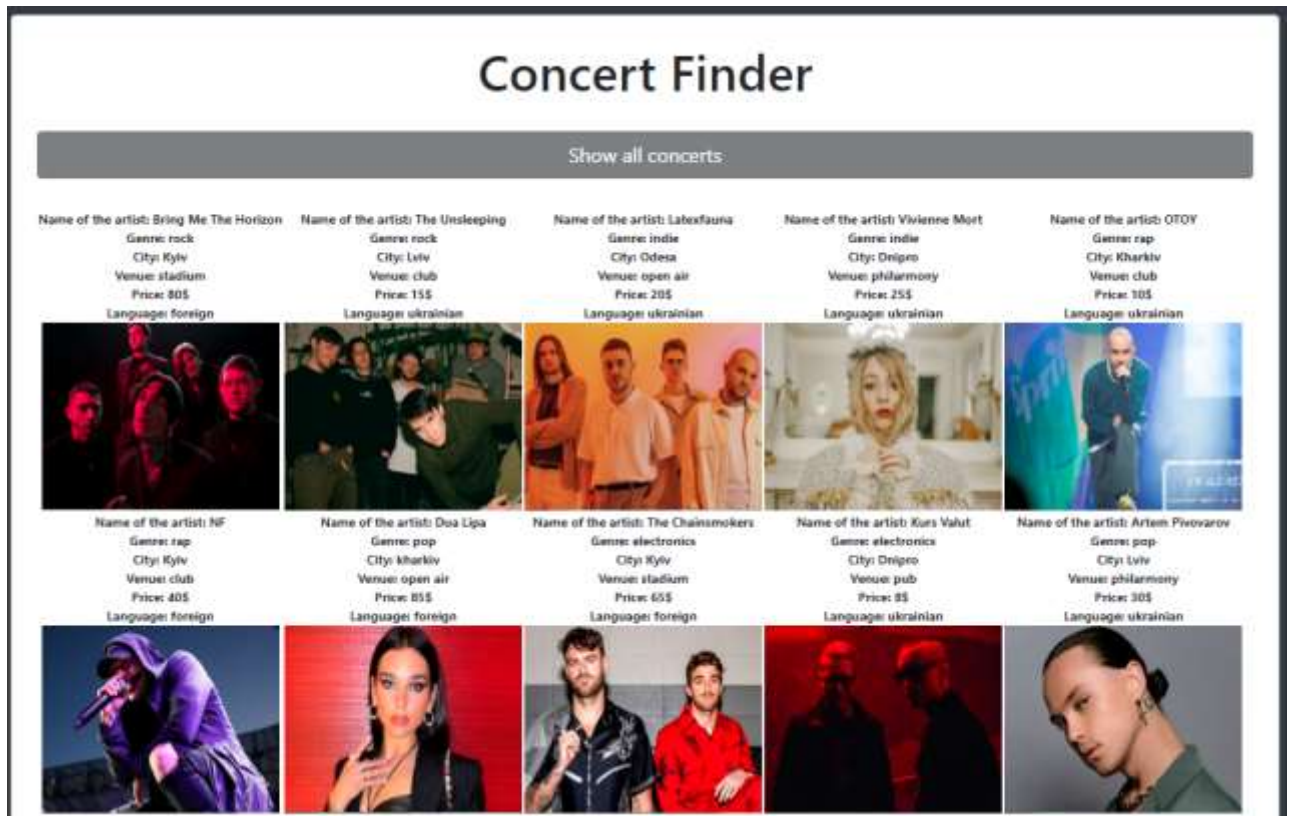


Рисунок 3.2 – Загальний вигляд інтерфейсного вікна «Відображення всіх концертів»


Для того, щоб отримати рекомендацію на основі відвідуваного раніше концерту, необхідно спочатку у пункті «Did you visit one of the concerts before?» вибрати відповідь «Yes», а після у пункті «Choose a concert you visited before» вибрати необхідний концерт.

Вигляд вікна відображення наданої рекомендації на основі відвідуваного раніше концерту показаний на рисунку 3.3.

Did you visit one of the concerts before?

Choose a concert you visited before:

- Name of the artist: NF
 Genre: rap
 City: Kyiv
 Venue: club
 Price: 40\$
 Language: foreign
 Probability: 0.8793969849246231


- Name of the artist: Kurs Valut
 Genre: electronics
 City: Dnipro
 Venue: pub
 Price: 8\$
 Language: ukrainian
 Probability: 0.6268656716417911




Рисунок 3.3 – Загальний вигляд інтерфейсного вікна «Відображення рекомендації на основі відвідуваного раніше концерту»

Для того, щоб отримати рекомендацію на основі заданих критеріїв, необхідно спочатку у пункті «Did you visit one of the concerts before?» вибрати відповідь «No», а після – вибрати необхідні критерії, які підходять користувачу і натиснути на кнопку «Find».

Вигляд вікна відображення вводу необхідних критеріїв для надання рекомендації щодо підбору музичного концерту показаний на рис.3.4.

Choose a genre:

Choose a city:

Choose a venue:

Choose your budget(0-100\$):

Choose a language of the artist:

Рисунок 3.4 – Загальний вигляд інтерфейсного вікна «Відображення вводу необхідних критеріїв для надання рекомендації щодо підбору музичного концерту»

Кожен критерій має певний список наявних варіантів, приклад показаний на рисунку 3.5.

Choose a genre:

Choose a city:

Choose a venue:

Choose your budget:

Choose a language of the artist:

Рисунок 3.5 – Загальний вигляд інтерфейсного вікна «Вибір міста проведення концерту»

Після вибору необхідних критеріїв натискаємо на кнопку «Find», щоб отримати рекомендації. Вигляд вікна відображення наданої рекомендації на заданих критеріїв показаний на рисунку 3.6.

Choose a genre:

Choose a city:

Choose a venue:


Choose your budget(0-100\$):

Choose a language of the artist:

Find

Results

- Name of the artist: The Unsleeping
 Genre: rock
 City: Lviv
 Venue: club
 Price: 15\$
 Language: ukrainian
 Probability: 0.9554140127388535


- Name of the artist: Artem Pivovarov
 Genre: pop
 City: Lviv
 Venue: philarmony
 Price: 30\$
 Language: ukrainian
 Probability: 0.7241379310344828




Рисунок 3.6 – Загальний вигляд інтерфейсного вікна «Відображення рекомендації на основі заданих критеріїв»

Здійснимо тестування роботи розробленої інформаційної технології в контексті підвищення якості надання рекомендацій. Для цього порівняємо якість надання рекомендацій розробленої інформаційної технології з рекомендаціями експертів. Результати порівняння підвищення якості надання рекомендацій зображено в таблиці 3.2.

Таблиця 3.2 – Таблиця порівняння наданих рекомендацій

Критерій	Задані критерії №1	Задані критерії №2	Задані критерії №3	Задані критерії №4	Задані критерії №5	Задані критерії №6
Жанр	Rock	Rap	Electronics	Indie	Pop	Rock
Місто	Kyiv	Kharkiv	Kyiv	Kharkiv	Odesa	Kyiv
Локація	Stadium	Club	Open Air	Philharmony	Open Air	Open Air
Бюджет	95\$	5\$	90\$	50\$	35\$	40\$
Мова	ukrainian	ukrainian	foreign	ukrainian	ukrainian	ukrainian
Оцінка системи	0,955	0.704	0.966	0.904	0.879	0.823
Оцінка експерта	0.8	0.8	0.8	0.8	0.8	0.6

Приклад відображення наданої рекомендації разом з її точністю за заданими критеріями 1 зображена на рис.3.7.

The screenshot shows a web interface for a recommendation system. It features several input fields and a slider:

- Choose a genre:** A dropdown menu with "Rock" selected.
- Choose a city:** A dropdown menu with "Kyiv" selected.
- Choose a venue:** A dropdown menu with "Stadium" selected.
- Choose your budget (0-100\$):** A horizontal slider with a blue bar extending to approximately 80%.
- Choose a language of the artist:** A dropdown menu with "Ukrainian" selected.
- Find:** A button to execute the search.

Below the search form, the results are displayed under the heading "Results":

- Name of the artist:** Bring Me The Horizon
- Genre:** rock
- City:** Kyiv
- Venue:** stadium
- Price:** 80\$
- Language:** foreign
- Probability:** 0,9554140127388555

At the bottom of the results section, there is a small image of the band Bring Me The Horizon performing on stage.

Рисунок 3.7 – Загальний вигляд інтерфейсного вікна «Відображення рекомендації на основі заданих критеріїв»

Підрахуємо середню точність інформаційної технології та експерта.

Точність (система підбору):

$$A_{IT} = \frac{0,955 + 0,704 + 0,966 + 0,904 + 0,879 + 0,823}{6} = 0,871.$$

Точність (експерт):

$$A_E = \frac{0,8 + 0,8 + 0,8 + 0,8 + 0,8 + 0,6}{6} = 0,76.$$

Відсоток підвищення якості надання рекомендацій розробленою інформаційною технологією:

$$I = \frac{A_{IT}}{A_E} * 100\% = \frac{0,871}{0,76} * 100\% = 11,4\%$$

Таким чином, якість підбору підвищено на 11,4%.

Отже, в ході тестування створеної програми перевірено її роботу згідно встановлених критеріїв, що забезпечило підвищення точності підбору на 11,4%.

Серед усіх аналогів, найбільше критеріїв для більш точного підбору має квитковий оператор Concert.ua, який має чотири критерії, тим часом, як в розробленій системі підбору їх п'ять, тобто крім підвищення якості підбору відбулося розширення функціоналу інформаційної технології.

Деякий ілюстративний матеріал до програми (у т.ч. скріншоти) подано в додатку В.

3.6 Висновок до розділу 3

Таким чином, в даному розділі розглянуто та проаналізовано три мови програмування, а саме C++, JavaScript та Python. Виходячи з такої необхідності як розробка веб-додатку, для створення програмного забезпечення інформаційної технології обрано мову програмування JavaScript. Розглянуто для порівняння такі середовища програмування, як NetBeans, JetBrains,

WebStorm, Visual Studio Code та Microsoft Visual Studio Code. Таким чином, середовищем розробки інформаційної технології обрано Visual Studio Code. Розглянуто три системи керування базами даних, такі як MongoDB, MySQL та PostgreSQL. Для розробки інформаційної технології обрано MongoDB, так як вона забезпечує простоту розробки, не потребує додаткових інструментів для приведення даних та має гнучку схему даних, що дозволяє спростити подальшу підтримку розробленого модуля. Використовуючи обрані технології, розроблено програмну реалізацію інформаційної технології, що реалізує функції та виконує задачі, описані у попередніх розділах. Проведено тестування та аналіз результатів роботи інформаційної технології для надання рекомендацій щодо підбору концерту, що забезпечило виконання встановлених критеріїв та досягнення поставленої мети дослідження, а саме розширення функціоналу та підвищення якості підбору на 11,4%. Наступним етапом є тестування розробленого інформаційної технології на відповідність усім поставленим вимогам і на коректність виконання всіх поставлених задач.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційної технологія для надання рекомендацій щодо підбору музичного концерту. Особливістю програми є те, що дана інформаційна технологія аналізує концерти, відвідувані користувачем раніше, а також використовує теорему Баєса, що дозволяє удосконалити точність наданої рекомендації щодо підбору концерту.

Аналогом може бути Concert, ціна 20000 грн.; Karabas, ціна 16000 грн.; Kasa, ціна 14000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	42	41	43
Середньоарифметична сума балів	$(42+41+43) / 3 = 42$		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт надає розширені функціональні можливості в наданні рекомендацій щодо підбору музичних концертів. Актуальним на сьогодні є питання підбору музичних концертів, які цікаві споживачу, так як через стрімкий розвиток культури звичайні рамки музичних жанрів стають більш розмиті, а конкуренція серед виконавців тільки зростає. Особливість розробки полягає в тому, що інформаційна технологія аналізує концерти, відвідувані користувачем раніше, а також використовує теорему Баєса, що дозволяє удосконалити точність наданої рекомендації щодо підбору концерту.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 21 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.1.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	45000	2142,86	30	64285,714
Програміст	43500	2071,43	30	62142,857
Всього				126428,57

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 12,8 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 12,8 \% / 100 \% \quad (4.2)$$

$$Z_d = (126428,57 \cdot 12,8 \% / 100 \%) = 16182,86 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_3 = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (4.3)$$

$$H_3 = (126428,57 + 16182,86) \cdot 22 \% / 100 \% = 31374,51 \text{ (грн.)}$$

4.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_{в}} \cdot \frac{t_{вик}}{12} \text{ [Грн.]} \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 52000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,43 міс.

$$A_{обл} = \frac{52000}{2} \times \frac{1,43}{12} = 3095,238 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик}}{12} \quad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних

нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 920$ грн. (Windows 11 Pro, Visual Studio Code).

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	52000	2	1,43	3095,238
Офісне обладнання	25000	4	1,43	744,048
Приміщення	1300000	20	1,43	7738,095
Всього				11577,38

5.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,5$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$V_e = 0,9 \cdot 0,5 \cdot 8 \cdot 30 \cdot 6,2 = 669,6 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.7)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 126428,57 \cdot 105\% / 100\% = 132750 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 126428,57 \cdot 130\% / 100\% = 164357 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 126428,57 + 16182,86 + 31374,51 + 11577,38 + 920 + 669,60 + 132750 + 164357 = 484260,07 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 484260,07 / 0,5 = 968520 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.10)$$

де $\pm\Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_0 \pm \Delta\Pi_0$;

Π_0 – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 5000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 600 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 3500 шт., протягом другого року – на 3000 шт., протягом третього року на 2000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*600 + (5000 + 600) * 3500) * 0,8333 * 0,31 * (1 - 0,18) = 3707083,185 \text{ грн.}$$

$$\Delta\Pi_2 = (0*600 + (5000 + 600) * (3500+3000)) * 0,8333 * 0,31 * (1 - 0,18) = 7710733,025 \text{ грн.}$$

$$\Delta\Pi_3 = (0*600 + (5000 + 600) * (3500+3000+2000)) * 0,8333 * 0,31 * (1 - 0,18) = 10083266,263 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 21501082,47 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} \text{ПП} &= \\ & (3707083,185/(1+0,1)^1) + (7710733,025/(1+0,1)^2) + (10083266,263/(1+0,1)^3) = \\ & 3370075,62 + 6372506,632 + 7575707,185 = 17318289,44 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (4.12)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{инв} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 968520 = 1937040,27 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (4.13)$$

$$E_{abc} = 17318289,44 - 1937040,27 = 15381249,17 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формулу:

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_g = \sqrt[1]{1 + 15381249,17/1937040,27} - 1 = 1,075$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_B > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (4.16)$$

$$T_{ок} = 1 / 1,075 = 0,93 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,93 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновок до розділу 4

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 968520 гривень. Спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,93 роки.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології та програмного забезпечення надання рекомендації щодо підбору музичного концерту.

1. Проаналізовано сучасний стан розвитку систем для надання рекомендацій щодо підбору музичного концерту.
2. Проаналізовано методи надання рекомендацій.
3. Спроектовано інформаційну технологію для надання рекомендацій щодо підбору музичного концерту.
4. Розроблено алгоритм програмної реалізації інформаційної технології для надання рекомендацій щодо підбору музичного концерту.
5. Виконано програмну реалізацію інформаційної технології для надання рекомендацій щодо підбору музичного концерту.
6. Проведено тестування інформаційної технології для надання рекомендацій щодо підбору музичного концерту та аналіз результатів тестування.

Під час програмної реалізації інформаційної технології, виходячи з такої необхідності як розробка веб-додатку, обрано мову програмування JavaScript. У роботі виконано та описано схему алгоритму інформаційної технології. Під час тестування розробленого програмного продукту доведено, що відповідність усім поставленим вимогам і коректність виконання всіх поставлених задач була дотримана.

Мета дослідження досягнута за рахунок розробки інформаційної технології із застосуванням методу фільтрації на основі вмісту разом з використанням експертної системи як методу, що забезпечує надання рекомендацій щодо підбору музичного концерту, а саме – розширення функціональних можливостей інформаційною технологією та підвищення якості надання рекомендацій щодо підбору музичного концерту на 11,4%. За результатами роботи опубліковано одні тези доповіді.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Максимум звуку: Світовий музичний ринок зріс на 10% [Електронний ресурс] – режим доступу: <https://investory.news/maksimum-zvuku-svitovij-muzichnij-rinok-zris-na-10/>
2. Музичний ринок України: 7 ключових висновків з дослідження [Електронний ресурс] – режим доступу: [https://lb.ua/culture/2020/10/20/468513 muzichnij rinok ukraini 7 klyuchovih.html](https://lb.ua/culture/2020/10/20/468513_muzichnij_rinok_ukraini_7_klyucho_vih.html)
3. Стойко М.І., Ярова О.А., Яровий А.А. Інформаційна технологія для надання рекомендацій щодо підбору музичного концерту [Електронний ресурс] – режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/16886>
4. Тенденції ринку. Куди рухається музична індустрія [Електронний ресурс] – режим доступу: <https://suspilne.media/196716-tendencii-rinku-kudi-ruhaetsa-muzicna-industria-pisla-dvoh-rokiv-pandemii/>
5. The Mechanics of Touring: How the Live Music Industry Works [Електронний ресурс] – режим доступу: <https://soundcharts.com/blog/mechanics-of-touring>
6. Concert.ua [Електронний ресурс] – режим доступу: <https://concert.ua/>
7. Karabas [Електронний ресурс] – режим доступу: <https://karabas.com/ua/>
8. Kasa.in.ua [Електронний ресурс] – режим доступу: <https://kasa.in.ua/>
9. Recommender systems introduction. [Електронний ресурс] - режим доступу: <https://www.coursera.org/learn/recommender-systemsintroduction/home/week/1>
10. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы [Електронний ресурс]. — режим доступу: <https://www.ibm.com/developerworks/ru/library/os-recommender1/>
11. P. Lops, M. Gemmis, G. Semeraro Recommender Systems Handbook Springer-Verlag, 2011.

12. What Content-Based Filtering is and Why You Should Use It [Електронний ресурс] – режим доступу: <https://www.upwork.com/resources/what-is-content-based-filtering>
13. What is content-based filtering? [Електронний ресурс] – режим доступу: <https://www.educative.io/answers/what-is-content-based-filtering>
14. What Is Collaborative Filtering: A Simple Introduction [Електронний ресурс] – режим доступу: <https://builtin.com/data-science/collaborative-filtering-recommender-system>
15. Lytvyn V., Vysotska V., Shatskykh V. et al. Design of A Recommendation System Based on Collaborative Filtering and Machine Learning Considering Personal Needs of The User. Eastern European Journal of Enterprise Technologies. 2019. Vol. 4/2 (100). P. 6–28
16. Collaborative Filtering Advantages & Disadvantages [Електронний ресурс] – режим доступу: <https://developers.google.com/machine-learning/recommendation/collaborative/summary>
17. Collaborative Filtering Vs Content-Based Filtering for Recommender Systems [Електронний ресурс] – режим доступу: <https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/>
18. Collaborative Filtering In Recommender Systems: Learn All You Need To Know [Електронний ресурс] – режим доступу: <https://www.iteratorshq.com/blog/collaborative-filtering-in-recommender-systems/>
19. Експертні системи: особливості застосування. [Електронний ресурс] – режим доступу: <https://osvita.ua/vnz/reports/management/13574/>
20. Формула (теорема) Байеса и примеры решения задач [Електронний ресурс] – режим доступу: <https://intellect.icu/formula-bajesa-i-primery-reeniya-zadach-4527>

21. Теорема гіпотез (формула Байєса) [Електронний ресурс] – режим доступу: <http://ebooks.git-elt.hneu.edu.ua/tvms/p-2-5.html>
22. Загальна характеристика UML [Електронний ресурс] – режим доступу: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2>
23. Уніфікована мова моделювання UML [Електронний ресурс] – режим доступу: <http://www.znannya.org/?view=uml>
24. C++ [Електронний ресурс] – режим доступу: <https://sites.google.com/site/korkholeh2/cppfund.pdf>
25. Welcome to Python [Електронний ресурс] – Режим доступу: <https://www.python.org>
26. Д. Фленаган. «JavaScript. Полное руководство» 2018 с.358-362
27. К. Сімпсон «ES6 і не тільки» 2017 с.224-229
28. 10 лучших IDE [Електронний ресурс] – режим доступу: <https://timeweb.com/ru/community/articles/5-luchshih-ide-1>
29. NetBeans IDE [Електронний ресурс] – режим доступу: <https://biblprog.org.ua/ua/netbeans/>
30. WebStorm: The Smartest JavaScript IDE by JetBrains [Електронний ресурс] – режим доступу: <https://www.jetbrains.com/webstorm/>
31. Visual Studio Tutorial: Everything You Need to Know [Електронний ресурс] – режим доступу: <https://www.edureka.co/blog/visual-studio-tutorial/>
32. Обзор функционала Microsoft Visual Studio [Електронний ресурс] – режим доступу: <http://psihdocs.ru/uchebnoj-praktike-up-01-01-razrabotka-modulej-programmnogo-obe.html?page=2>.
33. Використання розширення Visual Studio Code [Електронний ресурс] – режим доступу: <https://learn.microsoft.com/uk-ua/power-apps/maker/portals/vs-code-extension>
34. Що таке Visual Studio Code? [Електронний ресурс] – режим доступу: <https://uk.education-wiki.com/3958588-what-is-visual-studio-code>
35. Найпопулярніші запитання для інтерв'ю MongoDB у 2021 році [Електронний ресурс] – режим доступу:

<https://ciksiti.com/uk/chapters/9117-top-mongodb-interview-questions-in-2021>

36. СУБД: які бувають, як вибрати [Електронний ресурс] – режим доступу: <https://highload.today/uk/subd-yaki-buvayut-yak-vibrati/>
37. База даних MySQL [Електронний ресурс] – режим доступу: <https://promoter.net.ua/articles/baza-danix-mysql.html>
38. Що таке PostgreSQL? [Електронний ресурс] – режим доступу: <https://uk.theastrologypage.com/postgresql>
39. PostgreSQL: як встановити та налаштувати на Ubuntu? [Електронний ресурс] – режим доступу: https://blog.iteducenter.ua/guides/postgresql_ubuntu/
40. Що таке база даних Oracle (Oracle DB)? [Електронний ресурс] – режим доступу: <https://uk.theastrologypage.com/oracle-database>

Додаток А

Протокол перевірки МКР на наявність текстових запозичень



Ім'я користувача:
Озеранський В.С. КН

ID перевірки:
1013337266

Дата перевірки:
20.12.2022 21:57:37 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
20.12.2022 21:59:39 EET

ID користувача:
62038

Назва документа: 122МКР-СтойкоМІ2022

Кількість сторінок: 48 Кількість слів: 8090 Кількість символів: 63481 Розмір файлу: 1.01 MB ID файлу: 1013097238

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

5.07%
Схожість

Найбільша схожість: 5.07% з джерелом з Бібліотеки (ID файлу: 1013061113)

Не знайдено джерел з Інтернету

5.07% Джерела з Бібліотеки

1

Сторінка 50

0.28% Цитат

Цитати

1

Сторінка 51

Не знайдено жодних посилань

16.3%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

10.6% Вилучення з Інтернету

87

Сторінка 52

12.6% Вилученого тексту з Бібліотеки

298

Сторінка 53

Модифікації

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

Замінені символи

3

Підозріле форматування

8

сторінок

Додаток Б

Лістинг програми

```

index.html
<!doctype html>
<html lang="en">
  <head>
    <title>Concert Finder!</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
    <link rel="stylesheet" type="text/css" href="styles.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta.2/css/bootstrap.min.css" integrity="sha384-
PshH8R72JQ3S0dhVi3uxftmaW6Vc51MKb0q5P2rRUppVrszue4W1povHYgTpBfshb"
crossorigin="anonymous">
  </head>
  <body class="bg-dark">
    <div class="container">
      <div class="row">
        <div class="col-md-11 mx-auto">
          <div class="card card-body text-center mt-5">
            <h1 class="heading display-5 pb-3">Concert Finder</h1>
            <form id="concerts">
              <div class="form-group">
                <input type="submit" class="btn btn-dark btn-block" value="Show
all concerts" id="all-concerts-button">
                <div id="show-all" class="pt-4">
                  <table class="all-concerts"></table>
                </div>
              </div>
            </form>
            <form id="question">
              <div class="input-group">
                <label for="check">Did you visit one of the concerts
before?</label>
                <select name="check" id="check" class="form-control"
placeholder="check">

```

```

        <option value="Yes">Yes</option>
        <option value="No">No</option>
    </select>
</div><br>
    <input type="submit" class="btn btn-dark btn-block" value="Submit"
id="check-button"><br>
</form>
<form id="previous">
    <div class="form-group">
        <div class="input-group">
            <label for="previous-concert">Choose a concert you visited
before:</label>
            <select name="previous-concert" id="previous-concert"
class="form-control" placeholder="previous-concert">
                <option value="Bring Me The Horizon">Bring Me The
Horizon</option>
                <option value="The Unsleping">The Unsleping</option>
                <option value="Latexfauna">Latexfauna</option>
                <option value="Vivienne Mort">Vivienne Mort</option>
                <option value="OTOY">OTOY</option>
                <option value="NF">NF</option>
                <option value="Dua Lipa">Dua Lipa</option>
                <option value="The Chainsmokers">The Chainsmokers</option>
                <option value="Kurs Valut">Kurs Valut</option>
                <option value="Artem Pivovarov">Artem Pivovarov</option>
            </select>
        </div>
        <br>
        <input type="submit" class="btn btn-dark btn-block" value="Show
recommendations based on previous concert" id="previous-concerts-button">
        <div id="show-previous" class="pt-4">
            <ul class="previous-list"></ul>
        </div>
    </div>
</form>
<br>
<form id="find">
    <div class="form-group">
        <div class="input-group">

```

```

        <label for="genre">Choose a genre:</label>
        <select name="genre" id="genre" class="form-control"
placeholder="Genre">
            <option value="rock">Rock</option>
            <option value="rap">Rap</option>
            <option value="indie">Indie</option>
            <option value="electronics">Electronics</option>
            <option value="pop">Pop</option>
        </select>
    </div>
</div>
<div class="form-group">
    <div class="input-group">
        <label for="city">Choose a city:</label>
        <select name="city" id="city" class="form-control"
placeholder="City">
            <option value="Kyiv">Kyiv</option>
            <option value="Lviv">Lviv</option>
            <option value="Dnipro">Dnipro</option>
            <option value="Kharkiv">Kharkiv</option>
            <option value="Odesa">Odesa</option>
        </select>
    </div>
</div>
<div class="form-group">
    <div class="input-group">
        <label for="venue">Choose a venue:</label>
        <select name="venue" id="venue" class="form-control"
placeholder="Venue">
            <option value="stadium">Stadium</option>
            <option value="club">Club</option>
            <option value="pub">Pub</option>
            <option value="open-air">Open Air</option>
            <option value="philarmony">Philarmony</option>
        </select>
    </div>
</div>
<div class="form-group">
    <div class="input-group">

```

```

        <label for="budget">Choose your budget(0-100$):</label>
        <input type="range" name="budget" class="form-control"
id="budget" placeholder="Budget">
        </div>
    </div>
    <div class="form-group">
        <div class="input-group">
            <label for="lang">Choose a language of the artist:</label>
            <select name="lang" id="lang" class="form-control"
placeholder="Language">
                <option value="ukrainian">Ukrainian</option>
                <option value="foreign">Foreign</option>
            </select>
        </div>
    </div>
    <div class="form-group">
        <input type="submit" value="Find" class="btn btn-dark btn-
block" id="find-button">
    </div>
</form>
<div id="results" class="pt-4">
    <h3>Results</h3>
    <div class="form-group">
        <ul class="concerts-list"></ul>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js"
integrity="sha384-
vFJXuSJphR0IrBnz7yo7oB41mKfc8JzQZiCq4NCceLEa04IHwicKwpJf9c9IpFgh"
crossorigin="anonymous"></script>

```

```

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
beta.2/js/bootstrap.min.js" integrity="sha384-
alpBpkh1PF0epccYVYDB4do5UnbKysX5WZXM3XxPqe5iKTfUKjNkCk9SaVuEZf1J"
crossorigin="anonymous"></script>
    <script src="concertfinder.js"></script>
  </body>
</html>

```

concertfinder.js

```

document.getElementById('question').addEventListener('submit', function(e){
  const checkButton = document.getElementById('check');
  if (checkButton.value == 'Yes') {
    console.log(checkButton.value);
    document.getElementById("previous").style.visibility = "visible";
  } else
  document.getElementById("find").style.visibility = "visible";
  document.getElementById("check-button").setAttribute("disabled",true);

  e.preventDefault();
});

document.getElementById('find').addEventListener('submit', function(e){
  document.getElementById('results').style.display = 'block';

  setTimeout(Find, 1000);
  document.getElementById("find-button").setAttribute("disabled",true);
  document.getElementById("results").style.visibility = "visible";

  e.preventDefault();
});

document.getElementById('concerts').addEventListener('submit', function(e){
  document.getElementById('show-all').style.display = 'block';

  setTimeout(showAll, 1000);
  document.getElementById("all-concerts-button").setAttribute("disabled",true);

  e.preventDefault();
});

```

```

});

document.getElementById('previous').addEventListener('submit', function(e){
    document.getElementById('show-previous').style.display = 'block';

    setTimeout(showPrevious, 1000);
    document.getElementById("previous-concerts-
button").setAttribute("disabled",true);

    e.preventDefault();
});

document.getElementById("find").style.visibility = "hidden";
document.getElementById("results").style.visibility = "hidden";
document.getElementById("previous").style.visibility = "hidden";

const inputGenre = document.getElementById('genre');
const inputCity = document.getElementById('city');
const inputVenue = document.getElementById('venue');
const inputBudget = document.getElementById('budget');
const inputLang = document.getElementById('lang');
const previousConcert = document.getElementById('previous-concert');

const concerts = [
    { name: 'Bring Me The Horizon', genre: 'rock', city: 'Kyiv', venue:
'stadium', budget: 80, lang: 'foreign', probability: 0, img: 'BMTH.jpg' },
    { name: 'The Unsleping', genre: 'rock', city: 'Lviv', venue: 'club', budget:
15, lang: 'ukrainian', probability: 0, img: 'TheUnsleping.jpg' },
    { name: 'Latexfauna', genre: 'indie', city: 'Odesa', venue: 'open air',
budget: 20, lang: 'ukrainian', probability: 0, img: 'Latexfauna.jpg' },
    { name: 'Vivienne Mort', genre: 'indie', city: 'Dnipro', venue: 'philarmony',
budget: 25, lang: 'ukrainian', probability: 0, img: 'VivienneMort.jpg' },
    { name: 'OTOY', genre: 'rap', city: 'Kharkiv', venue: 'club', budget: 10,
lang: 'ukrainian', probability: 0, img: 'OTOY.jpg' },
    { name: 'NF', genre: 'rap', city: 'Kyiv', venue: 'club', budget: 40, lang:
'foreign', probability: 0, img: 'NF.jpg' },
    { name: 'Dua Lipa', genre: 'pop', city: 'kharkiv', venue: 'open air', budget:
85, lang: 'foreign', probability: 0, img: 'DuaLipa.jpg' },

```

```

    { name: 'The Chainsmokers', genre: 'electronics', city: 'Kyiv', venue:
'stadium', budget: 65, lang: 'foreign', probability: 0, img:
'TheChainsmokers.jpg' },
    { name: 'Kurs Valut', genre: 'electronics', city: 'Dnipro', venue: 'pub',
budget: 8, lang: 'ukrainian', probability: 0, img: 'KursValut.jpg' },
    { name: 'Artem Pivovarov', genre: 'pop', city: 'Lviv', venue: 'philarmony',
budget: 30, lang: 'ukrainian', probability: 0, img: 'ArtemPivovarov.jpg' }
];

```

```

function Find(){
  console.log(inputGenre.value);
  console.log(inputCity.value);
  console.log(inputVenue.value);
  console.log(inputBudget.value);
  console.log(inputLang.value);
  concerts.forEach((concert) => {
    let probability = 1;
    let upper = 0.1;
    let low = 0.1;
    if (concert.genre == inputGenre.value) {
      low *= 0.5; } else upper *= 0.5;
    if (concert.city == inputCity.value) {
      low *= 0.4; } else upper *= 0.4;
    if (concert.venue == inputVenue.value) {
      low *= 0.7; } else upper *= 0.7;
    if (concert.budget <= inputBudget.value) {
      low *= 0.2; } else upper *= 0.2;
    if (concert.lang == inputLang.value) {
      low *= 0.6; } else upper *= 0.6;

    probability = (probability * upper) / (upper + low);
    console.log(concert.name + `:` + probability);
    concert.probability = probability;
  })
  let sortByProbability = (a, b) => a.probability < b.probability ? 1 : -1;
  concerts.sort(sortByProbability);
  console.log(concerts);
  let newList = []

```



```

for (const concert of concerts) {
  if (concert.probability > 0.6) newList.push(concert)
}

console.log (newList);

function Render(arr) {
  const list = document.querySelector('.concerts-list')

  for (const concert of arr) {
    const li = document.createElement('li')
    li.innerHTML = `
    <div>Name of the artist: ${concert.name}</div>
    <div>Genre: ${concert.genre}</div>
    <div>City: ${concert.city}</div>
    <div>Venue: ${concert.venue}</div>
    <div>Price: ${concert.budget}$</div>
    <div>Language: ${concert.lang}</div>
    <div>Probability: ${concert.probability}</div>
    `
    list.append(li)
  }
}

Render(newList)
}

function showAll(){
  let counter = 0;
  const table = document.querySelector('.all-concerts')
  for (const concert of concerts) {
    const row2 = document.createElement('td')
    row2.innerHTML = `
    <tr>
    Name of the artist: ${concert.name} <br>
    Genre: ${concert.genre} <br>
    City: ${concert.city} <br>
    Venue: ${concert.venue} <br>
    Price: ${concert.budget}$ <br>

```

```

    Language: ${concert.lang} <br>

    
  </tr> `
  table.append(row2)
  counter++;
  if (counter %5 == 0) {
    const row1 = document.createElement('tr')
    table.append(row1)
  }
}
}

function showPrevious(){

  console.log(previousConcert.value)

  concerts.forEach((concert) => {
    if (concert.name == previousConcert.value) {
      console.log('Here it is!' + concert.name);
      Previous = concert;
    }
  })
  console.log(Previous);
  concerts.forEach((concert) => {
    let probability = 1;
    let upper = 0.1;
    let low = 0.1;
    if (concert.genre == Previous.genre) {
      low *= 0.5; } else upper *= 0.5;
    if (concert.city == Previous.city) {
      low *= 0.4; } else upper *= 0.4;
    if (concert.venue == Previous.venue) {
      low *= 0.7; } else upper *= 0.7;
    if (concert.budget <= Previous.budget) {
      low *= 0.2; } else upper *= 0.2;
    if (concert.lang == Previous.lang) {
      low *= 0.6; } else upper *= 0.6;
  })
}

```

```

    probability = (probability * upper) / (upper + low);
    console.log(concert.name + `:` + probability);
    concert.probability = probability;
  })

  let sortByProbability = (a, b) => a.probability < b.probability ? 1 : -1;
  concerts.sort(sortByProbability);
  console.log(concerts);
  let newList = []

  for (const concert of concerts) {
    if (concert.name !== previousConcert.value)
      if (concert.probability > 0.5) newList.push(concert)
  }

  console.log (newList);

  function Render(arr) {
    const list = document.querySelector('.previous-list')

    for (const concert of arr) {

      const li = document.createElement('li')
      li.innerHTML = `
        <div>Name of the artist: ${concert.name}</div>
        <div>Genre: ${concert.genre}</div>
        <div>City: ${concert.city}</div>
        <div>Venue: ${concert.venue}</div>
        <div>Price: ${concert.budget}$</div>
        <div>Language: ${concert.lang}</div>
        <div>Probability: ${concert.probability}</div>
        `
      list.append(li)
    }
  }
  Render(newList)
}

```

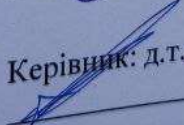
Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ НАДАННЯ
РЕКОМЕНДАЦІЙ ЩОДО ПІДБОРУ МУЗИЧНОГО КОНЦЕРТУ

Виконав: студент 1-го курсу,
групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Стойко М.І.
(прізвище та ініціали)

Керівник: д.т.н., професор каф. КН

Яровий А.А.
(прізвище та ініціали)

« 15 » 12 2022 р.

Вінниця ВНТУ - 2022 рік

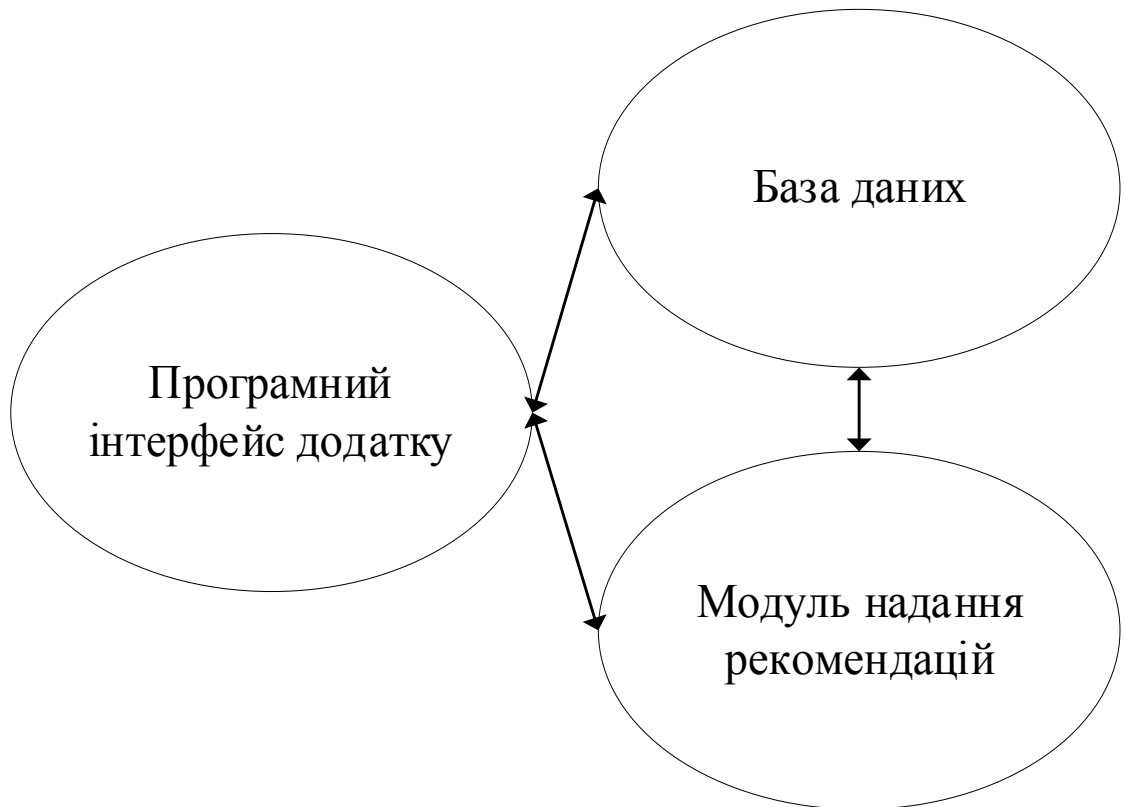


Рисунок В.1 – Узагальнена структурна схема інформаційної технології надання рекомендації щодо підбору музичного концерту

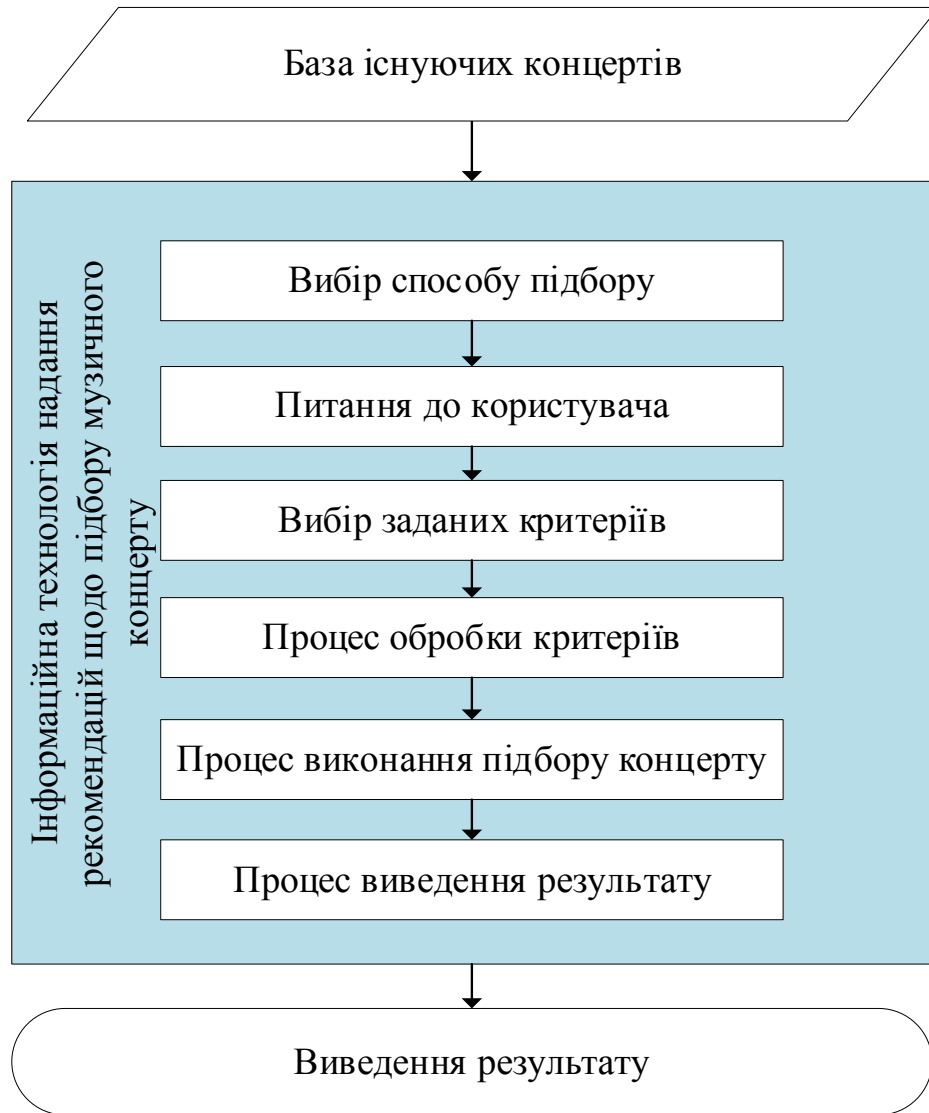


Рисунок В.2 – Детальна структурно-функціональна схема роботи інформаційної технології

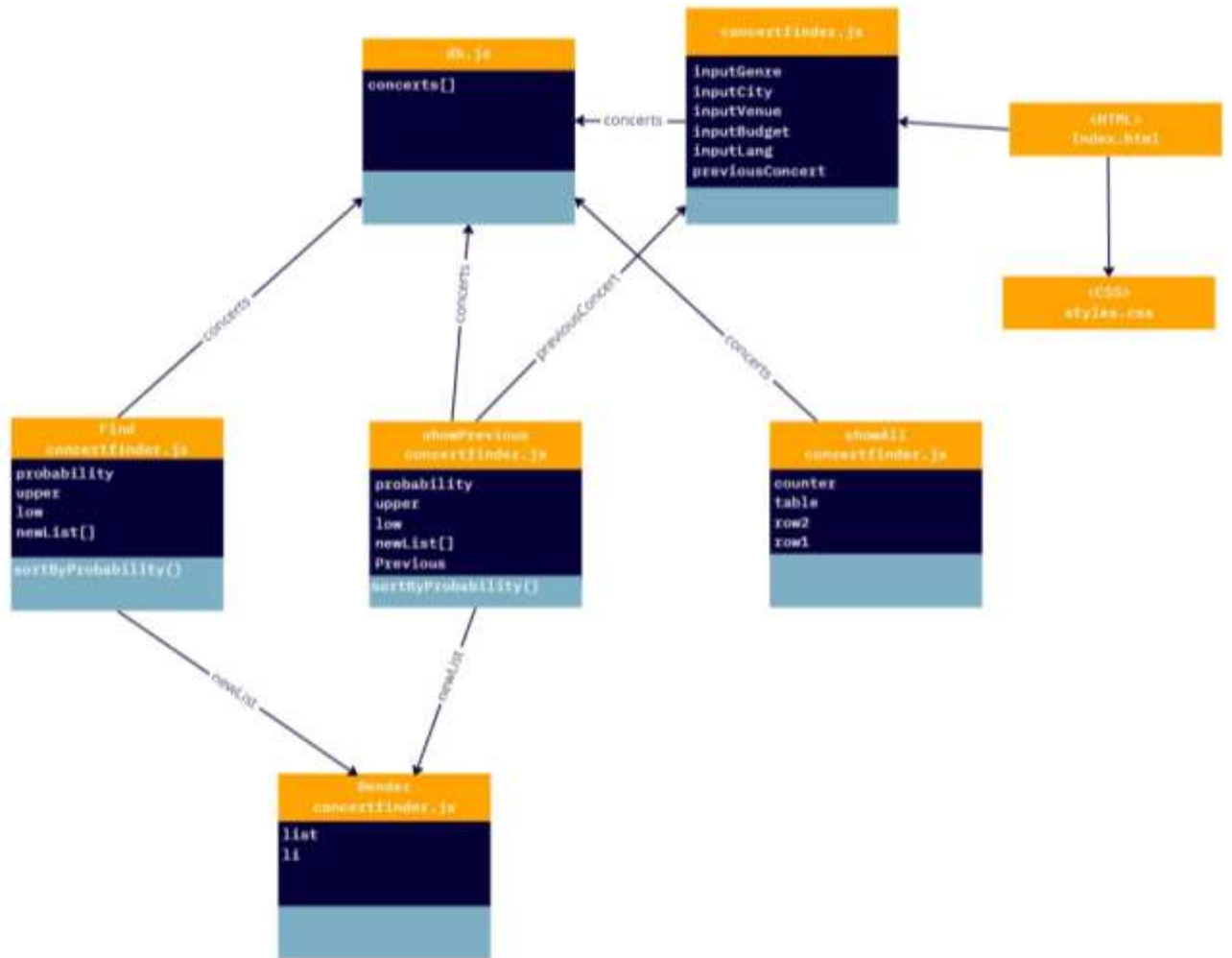


Рисунок В.3 – UML-діаграма класів інформаційної технології для надання рекомендацій щодо підбору музичного концерту

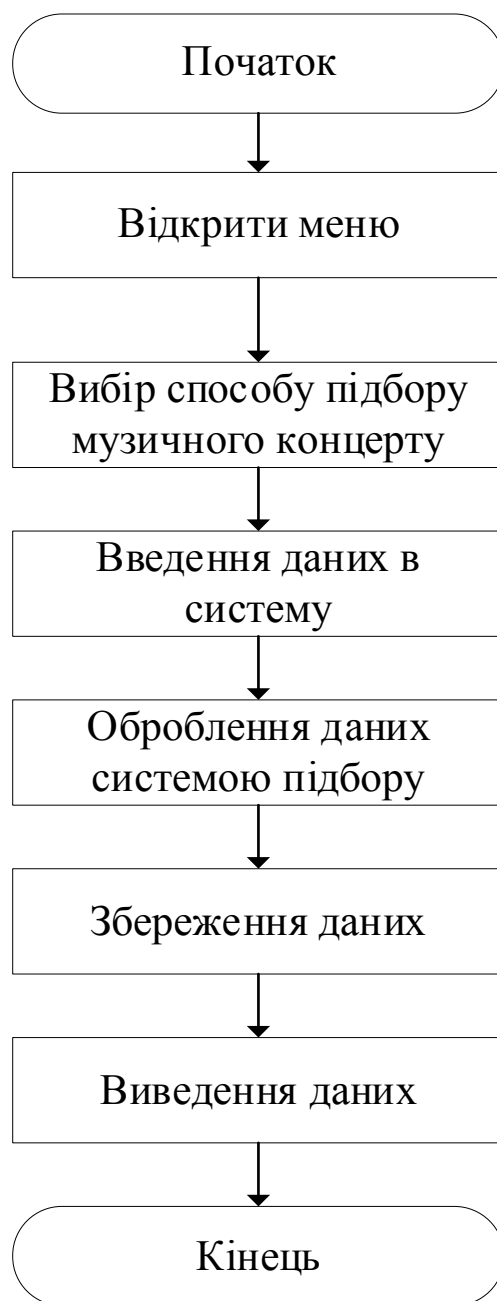




Рисунок В.4 – Схема узагальненого алгоритму роботи інформаційної технології для надання рекомендацій щодо підбору музичного концерту

Did you visit one of the concerts before?

Choose a concert you visited before:

- Name of the artist: NF
 Genre: rap
 City: Kyiv
 Venue: club
 Price: 40\$
 Language: foreign
 Probability: 0.8793969849246231


- Name of the artist: Kurs Valut
 Genre: electronics
 City: Dnipro
 Venue: pub
 Price: 8\$
 Language: ukrainian
 Probability: 0.6268656716417911



Choose a genre:

Choose a city:

Choose a venue:

Choose your budget(0-100\$):

Choose a language of the artist:

Results

- Name of the artist: Bring Me The Horizon
 Genre: rock
 City: Kyiv
 Venue: stadium
 Price: 80\$
 Language: foreign
 Probability: 0.9554140127388535




Рисунок В.5 – Робочі вікна інформаційної технології надання рекомендацій щодо підбору музичного концерту