

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

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

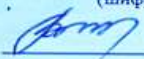
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))


МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія ранжування зображень»

Виконав: студент 2 курсу, групи 2КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)


Воронков О. І.
(прізвище та ініціали)


Керівник: доцент кафедри КН

Колодний В. В.
(прізвище та ініціали)

« 7 » 12 2023 р.

Опонент: к.т.н., доц каф. САІТ

Крижановський Є.М.
(прізвище та ініціали)

« 7 » 12 2023 р.

Допущено до захисту
Завідувач кафедри КН

д.т.н., проф. Яровий А.А.
« 8 » 12 2023 р.

Вінниця ВНТУ – 2023 рік

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
проф., д.т.н. Яровий А.А.

“ 29 ” 08 2023 року

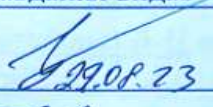
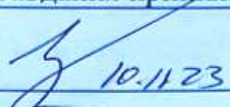


ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Воронкову Олександрю Ігоровичу

(прізвище, ім'я, по-батькова)

1. Тема роботи: Інформаційна технологія ранжування зображень.
Керівник роботи: Колодний Володимир Володимирович, к.т.н., доц. кафедри КН,
затверджені наказом вищого навчального закладу від «18» 09 2023 року № 247
2. Термін подання студентом роботи 13.11.2023 року
3. Вихідні дані до роботи: потужність множини правил – не менше 30од; потужність множини рекомендацій – не менше 30од; потужність БД – не менше 50од; потужність вихідного вектора – 3од; мова програмування – мультипарадигмова.
4. Зміст текстової частини (перелік питань, які потрібно розробити):
Вступ, Аналіз сучасних технологій для ранжування зображень; Розробка інформаційної технології ранжування зображень; Програмна реалізація інформаційної технології ранжування зображень; Економічна частина; Висновки; Перелік використаних джерел; Додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):
UML-діаграма діяльності інформаційної технології ранжування зображень; UML-діаграми активності модулів інформаційної технології ранжування зображень; Структурна схема взаємодії модулів; Загальний вигляд інтерфейсу користувача.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
1-3	Колодний В.В., к.т.н., доцент каф. КН	 29.08.23	 10.11.23
4	Ратушняк О. Г., к.т.н., доцент каф. ЕПВМ	 3.11.23	 10.11.23

7. Дата видачі завдання "29" 08 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз сучасних технологій для ранжування зображень	1.09.23 - 15.09.23		
2	Розробка інформаційної технології ранжування зображень	16.09.23 - 27.09.23		
3	Програмна реалізація інформаційної технології ранжування зображень	28.09.23 - 5.10.23		
4	Тестування інформаційної технології ранжування зображень	6.10.23 - 8.10.23		
5	Підготовка економічної частини	9.11.23 - 10.11.23		
6	Оформлення додатків	4.11.23 - 12.11.23		
7	Розробка інструкції користувача	12.11.23 - 13.11.23		
8	Оформлення пояснювальної записки, графічного матеріалу та презентації	12.11.23 - 13.11.23		

Студент  Воронков О. І.
(підпис) (прізвище та ініціали)

Керівник роботи  Колодний В.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.8

Воронков О. І. Інформаційна технологія ранжування зображень. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 107с.

На укр. мові. Бібліогр.: 20 назв; рис.: 17; табл. 12.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології ранжування зображень. Основною метою роботи було покращення процесу оцінювання альтернатив шляхом використання візуального представлення. У даній роботі розглядаються основні підходи та методи ранжування зображень, а також використовуються сучасні інструменти та алгоритми для автоматизації даного процесу. У ході роботи було проведено аналіз сучасних технологій ранжування груп альтернатив, що дозволило визначити найефективніші підходи. На основі цього аналізу були розроблені програмні засоби, спроможні забезпечити візуальне ранжування зображень. Була створена структурна організація основних модулів та сервісів, що входять до складу розробленої програми.

Для реалізації інформаційної технології у вигляді веб-сайту було використано мову програмування TypeScript, що є найпопулярнішою надбудовою над JavaScript, що є однією з найбільш популярних мов у сфері веб-розробки.

В розділі економічної частини проведено оцінювання комерційного потенціалу розробки інформаційної віртуального ігрового автомата, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, які склали 129271,48 грн, розраховано період окупності – 1,2 роки.

Ключові слова: JavaScript, TypeScript, ранжування, зображення, ТЗ, Prisma, MongoDB, порівняння, аналітика.

ABSTRACT

Voronkov O. I. Information technology of image ranking. Master's thesis on specialty 122 - computer science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2023. 107p.

In Ukrainian speech Bibliography: 20 titles; Fig.: 17; table 12.

This master's thesis is devoted to the development of information technology for image ranking. The main goal of the work was to improve the process of evaluating alternatives by using visual representation. The main approaches and methods of image ranking are considered in this work, and modern tools and algorithms are used to automate this process. In the course of the work, an analysis of modern technologies for ranking groups of alternatives was carried out, which made it possible to determine the most effective approaches. On the basis of this analysis, software tools capable of providing visual ranking of images were developed. A structural organization of the main modules and services included in the developed program was created.

To implement information technology in the form of a website, the TypeScript programming language was used, which is the most popular add-on to JavaScript, which is one of the most popular languages in the field of web development.

In the section of the economic part, an assessment of the commercial potential of the development of an informational virtual gaming machine was carried out, the costs of carrying out scientific work and implementing the results were predicted, which amounted to UAH 129,271.48, and the payback period was calculated - 1.2 years.

Keywords: JavaScript, TypeScript, ranking, images, T3, Prisma, MongoDB, comparison, analytics.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РАНЖУВАННЯ ЗОБРАЖЕНЬ.....	7
1.1 Задача ранжування зображень в сучасних умовах	7
1.2 Особливості функціонування сучасних програмних засобів ранжування зображень	10
1.3 Постановка задачі.....	18
1.4 Висновок до розділу 1.....	20
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ЗОБРАЖЕНЬ .	21
2.1 Уніфікована мова моделювання UML в процесі розробки інформаційної технології візуального ранжування зображень.....	21
2.2 Математична модель візуального ранжування зображень	24
2.3 Розробка структури інформаційної технології візуального ранжування зображень	26
2.4 Висновок до розділу 2.....	31
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ЗОБРАЖЕНЬ	32
3.1 Обґрунтування вибору мови програмування для розробки інформаційної технології ранжування зображень	32
3.2 Обґрунтування вибору середовища програмування для розробки інформаційної технології ранжування зображень	40
3.3 Розробка бази даних інформаційної технології ранжування зображень.....	44
3.4 Розробка алгоритму функціонування інформаційної технології ранжування зображень	49
3.5 Тестування програмного забезпечення ранжування зображень та аналіз його результатів.....	54
3.6 Висновок до розділу 3.....	60

	3
4 ЕКОНОМІЧНА ЧАСТИНА	61
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	61
4.2 Розрахунок витрат на проведення науково-дослідної роботи.....	66
4.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	73
4.4 Висновки до розділу 4	78
ВИСНОВКИ.....	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
ДОДАТКИ.....	82
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	83
Додаток Б (обов'язковий) Лістинг програми	84
Додаток В (обов'язковий) Ілюстративна частина.....	95
Додаток Г (довідниковий) Інструкція користувача.....	103

ВСТУП

Актуальність теми дослідження. У сучасному світі, де інформаційні технології швидко розвиваються, а обсяг інформації навколо нас зростає, стає все важливіше вміти швидко приймати рішення. Завдання прийняття рішень в умовах великої кількості неупорядкованої інформації можуть бути особливо складними. В таких ситуаціях важко зосередитися і врахувати важливі думки та правильно їх оцінити.

Особливо важкими є завдання прийняття рішень, де мета є нечіткою або частково визначеною. Очевидно, що в таких випадках залучення експертів для точної оцінки альтернатив за конкретними шкалами не має сенсу. Набагато ефективніше використовувати методику простих питань, які не вимагають значних інтелектуальних зусиль і часу, і звертатися безпосередньо до інтуїції людини, використовуючи систему швидкого інтуїтивного мислення.

Інформаційна технологія, реалізована у вигляді веб-додатку є доступним інструментом, який можна використовувати з будь-якого пристрою. Використовуючи його, людина може полегшити та прискорити процес порівняння набору зображень і отримання кінцевих результатів ранжування, покладаючись на свою інтуїцію.

Один із найефективніших методів включає візуальне порівняння і перегляд підсумкових результатів рейтингу у формі візуальних діаграм.

Зв'язок роботи з науковими програмами, планами, темами

Магістерська кваліфікаційна робота виконана відповідно до напряму наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є розширення функціональних можливостей програмного забезпечення ранжування зображень. Для досягнення цієї мети необхідно вирішити наступні завдання:

1. Провести аналіз сучасних технологій візуального ранжування картинок.
2. Вивчити переваги та недоліки програм-аналогів візуального ранжування картинок.
3. Розробити структуру інформаційної технології візуального ранжування зображень.
4. Розробити UML-діаграми функціонування інформаційної технології візуального ранжування зображень.
5. Здійснити програмну реалізацію інформаційної технології візуального ранжування зображень у вигляді веб-додатку.
6. Здійснити аналіз функціонування програмного забезпечення ранжування зображень.
7. Здійснити економічне обґрунтування доцільності розробки нової інформаційної технології

Об'єкт дослідження – процес ранжування зображень.

Предмет дослідження – інформаційна технологія ранжування зображень.

Методи дослідження. У процесі досліджень використовувались: метод синтетичної контрольної групи, методи алгоритмізації та програмної реалізації під час розробки програмного забезпечення, методи теорії баз даних для розробки структури бази даних.

Наукова новизна отриманих результатів. Удосконалено інформаційну технологію ранжування зображень що відрізняється від існуючих удосконаленою моделлю порівняння зображень, що забезпечило розширення функціональних можливостей програмного забезпечення ранжування зображень.

Практичне значення одержаних результатів полягає в тому, що:

- розроблено алгоритм роботи інформаційної технології ранжування зображень;
- здійснено програмну реалізацію інформаційної технології ранжування зображень.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології ранжування зображень.

Особистий внесок здобувача. Усі наукові результати, наведені у магістерській кваліфікаційній роботі, отримані автором самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [1] – описано особливості проведення візуального ранжування зображень.

Апробація результатів роботи. Результати роботи були апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» (м. Вінниця, Україна, 2023 р.) [1].

Публікації. За результатами магістерської кваліфікаційної роботи опубліковано тезу доповіді на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» [1]. Подано до реєстрації в Український національний офіс інтелектуальної власності та інновацій свідоцтво про реєстрацію авторського права на твір (комп'ютерна програма).

1 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РАНЖУВАННЯ ЗОБРАЖЕНЬ

1.1 Задача ранжування зображень в сучасних умовах

Питання вибору завжди буде актуальним для людини, незважаючи на її вік, стать, національність, освіту і т.д. Життєвий досвід, звичайно, вчить, і з часом приймати рішення стає простіше, але він зовсім не дає ніякої гарантії, що всі рішення без винятку будуть правильними і ефективними. Прийняття рішень – це вміння, для оволодіння яким необхідно докласти чимало зусиль, а також здобути досить велику кількість знань.

Піти ж тут можна різними шляхами: або пізнавати все методом спроб і помилок, витрачаючи на це багато часу і нервів, або пошукати можливість отримати необхідну інформацію в структурованій і компактній формі, тим самим заощадивши свої психічні і тимчасові ресурси. Вибирати стратегію своєї поведінки нам доводиться в більшості життєвих ситуацій, і вибираємо ми завжди, навіть тоді, коли нам здається, що ми цього не робимо. Але навичка прийняття рішень – це вміння свідомо обирати з безлічі варіантів той, який вплине на ситуацію найкращим чином. Говорячи простіше, є певна цільова функція, яка допомагає оцінити «корисність» тієї чи іншої ситуації. Вона може стосуватися не тільки людини, що робить вибір, але і його близьких, друзів, колег або взагалі всього людства. І вміння приймати рішення – це навичка вибору сценарію з максимальним значенням цільової функції. Важливо зауважити, що вибір не завжди правильний об'єктивно – він просто є найкращим на думку конкретної людини [5].

Є ситуації, в яких зробити вибір і прийняти рішення складно. Наприклад, цільова функція може видавати однакові значення, різні варіанти можуть бути рівноцінні, обидва сценарії можуть бути однаково цінні для людини. Інший варіант труднощів з прийняттям рішень виражається в тому, що цільова функція не

встановлена. Інакше кажучи, людина просто не знає, чого хоче. Цей випадок набагато складніше, і вимагає більш суворого визначення причин труднощів.

Виходячи зі сказаного, вміння приймати рішення являє собою цілий набір навичок:

- навичка бачення максимальної кількості можливих варіантів рішень;
- навичка розставлення пріоритетів (визначення цільової функції) для кожної конкретної ситуації;
- навичка вибору одного рішення серед множини інших.

Не буде зайвим відзначити, що всі ці функції за людину спочатку (поки вона знаходиться в дитячому віці) виконують дорослі. Але чим старше стає людина, тим більше вибору їй доводиться робити самостійно. І прийнято вважати, що після досягнення повноліття людина вже в змозі приймати будь-які рішення, які визначають її подальше життя.

Людина розвивається далі, і вже вчиться приймати рішення, які стосуються не тільки її, а й інших людей. Тобто вона вчиться приймати рішення за інших, наприклад, коригувати поведінку своїх дітей вдома або керувати співробітниками на роботі. Цей навик теж дуже важливий, але він і більш складний, тому що кількість варіантів вибору розширюється, а цільова функція включає в себе особисті інтереси і інтереси оточуючих [6].

Актуальність цієї теми обумовлена тим, що вміння приймати рішення – це в першу чергу здатність діяти, вирішувати ситуації, вирішувати проблеми і покращувати власне життя. І від того, наскільки людина володіє цією навичкою, можуть залежати як її власні життєві і професійні результати, так і результати інших людей.

Ранжування визначає ступінь різниці між об'єктами за певними ознаками. Цей процес використовується, коли не можливо або неефективно давати пряму оцінку. Ранжування надає лише інформацію про те, який об'єкт кращий, без вказівок на скільки разів він переважає інший [2].

Простий метод ранжування полягає у впорядкуванні альтернатив залежно від їх важливості, зменшуючи переваги. Метод безпосередньої оцінки включає присвоєння суб'єктом оцінювання певного значення з використанням шкали оцінок [3].

Метод парного порівняння визначає перевагу однієї альтернативи над іншою, утворюючи матрицю з альтернативами в рядках і стовпцях. Цей підхід передбачає проведення всіх можливих порівнянь між альтернативами [3]. Під час парного порівняння визначається ступінь переваги або непереваги однієї альтернативи над іншою. Структура матриці дозволяє числово виразити ці відносини, що полегшує аналіз. Метод забезпечує об'єктивне порівняння альтернатив, хоча може вимагати значної кількості порівнянь у складних ситуаціях. Використання цього методу вирішує завдання визначення переваги одного варіанту перед іншим і забезпечує системний підхід до процесу прийняття рішень.

У гештальт-ранжуванні використовується концепція "гештальт", що вказує на цілісну форму або структуру. Метод к-арного гештальт-ранжування дозволяє ефективно та інтуїтивно упорядковувати випадкові альтернативи. Теорія прийняття рішень використовує парні порівняння альтернатив та рівні, такі як "еквівалентність" і "строга перевага", використовуючи шкалу Сааті [3]. Гештальт-ранжування засноване на сприйнятті об'єкта як цілісної одиниці, де елементи взаємодіють, утворюючи нове значення. К-арне гештальт-ранжування використовується для швидкого визначення порядку альтернатив на основі їхнього сприйняття. Шкала Сааті, або аналогічні методи, застосовуються для кількісної оцінки важливості кожної альтернативи у порівнянні з іншими.

Ефект контексту вибору може вплинути на прийняття рішень, порушуючи незалежність. Для простоти і надійності порівнянь рекомендується порівнювати мінімальну кількість об'єктів одночасно. Таким чином, розглядаються трійки альтернатив у тернарних гештальт-ранжуваннях [4].

Тернарне ранжування забезпечує вищу роздільну здатність і є надійним методом висловлення уподобань. Практика довела ефективність методу трирівневого аналізу ранжувань у випадках з обмеженою кількістю альтернатив або при складнощах визначення окремих критеріїв. Використання гештальтоцінок альтернатив виявилось доцільним [5]. Тернарне ранжування враховує три рівні уподобань, що сприяє більш точному визначенню їхнього ступеня. Цей метод особливо ефективний у випадках, коли складно встановити конкретні критерії, але легко створити гештальтоцінки для альтернатив.

1.2 Особливості функціонування сучасних програмних засобів ранжування зображень

Часто люди в колективі мають проблеми комунікації. Трапляються ситуації, що наші думки, які ми намагаємося розповісти іншому співрозмовнику всіх деталях, залишаються незрозумілими. Справа в тому, що наші слова, різні люди розуміють по-різному. У кожного з нас свій особистий, особливий досвід, тому і ставлення до одного і той же може сильно відрізнятись. Навіть одну і ту ж презентацію на конференції слухачі сприймуть по-своєму: запам'ятають різні слайди, придумують різні питання. Виникає запитання, як домогтися бездоганного і однозначного розуміння серед таких різних людей одночасно? Малюнки в блокноті, замальовки на серветці, схеми в інтернеті, що пояснюють відео та інші засоби комунікації дозволяють фіксувати наші думки і доносити інформацію наочно. Проблема комунікації і пошуку істини лежить в основі багатьох звичних нам речей – фотографій, фільмів, сучасної інфографіки або класичного живопису [7].

Візуальні образи сприймаються нами більш однозначно, ніж їх словесний опис. Краще один раз побачити. Більш того, в одну одиницю часу ми можемо побачити набагато більше, ніж можемо почути. Візуальні образи в 60 000 разів

сприймаються швидше, ніж слова. Дослідження Массачусетського інституту МІТ від 2014 року показали, що людина може зрозуміти, що зображено на фотографії, подивившись на неї всього 13 мілісекунд.

Згодом стало зрозуміло, що візуалізація інформації може допомогти в рішенні задач і істотно скоротити час пошуку рішення. У XVII столітті, вчені почали візуалізувати метео-дані на картах і вести прості таблиці. Через століття, завдяки Вільяму Плейфер, з'явилися перші наукові способи візуалізації статистичної інформації та економічних даних. Плейфер аргументував, що діаграми передають дані ефективніше, ніж таблиці.

На сучасний момент існує значна кількість різноманітних програм, які можуть бути використані для більш ефективного проведення оцінки альтернативних варіантів. Ці програми реалізовані на різних платформах, таких як настільні комп'ютери, веб-сайти та мобільні додатки, що розширює їхню універсальність і доступність для різних користувачів. Серед найбільш відомих таких засобів можна виокремити ВМПП, ТеГР та VisPA. Ці інструменти забезпечують розширені можливості для проведення об'єктивних оцінок, що полегшує вибір оптимальних рішень в різних сферах діяльності.

Веб-сайт "Візуальне моделювання попарних порівнянь" (ВМПП) представляє собою інноваційний інструмент, спрямований на впорядкування різноманітних альтернатив шляхом проведення попарних візуальних порівнянь. Цей онлайн ресурс надає можливість ефективно ранжувати кілька варіантів, а його результати відображаються як кардинальні, так і нормалізовані показники. На рисунку 1.1 наведено скріншоти роботи програми, які ілюструють інтерфейс та функціонал даного інструменту для зручності користувача. Використання ВМПП дозволяє здійснювати об'єктивні оцінки та систематизувати вибір оптимальних альтернатив у візуально привабливий та зрозумілий спосіб.

Альтернативи
червоний | синій | зелений | чорний

Попарне порівняння

Альтернатива 1: **синій** (64)
Альтернатива 2: **червоний** (36)

Далі

Альтернативи
синій | зелений | червоний | чорний

Результат попарного порівняння

Найкраща альтернатива: синій

Ранжування:

синій	17	0.461
зелений	6	0.249
червоний	19	0.152
чорний	24	0.138

Почати спочатку

Рисунок 1.1 – Приклад роботи програми ВМПП

Програма ТеГР (Тернарні гештальт-ранжування) є вдосконаленим інструментом для візуального моделювання та систематизації тернарних гештальт-ранжувань. За допомогою цієї програми користувач може легко ранжувати три альтернативи, використовуючи візуальні засоби та налаштовуючи шкалу оцінок від 5 до 500. Результати ранжування відображаються у формі як кардинальних значень, так і нормалізованих показників, надаючи повноту та об'єктивність в оцінці. Окрім того, програма дозволяє зберігати результати ранжування у вигляді зображень, що сприяє зручній подальшій обробці та обміну інформацією. Скріншоти функціоналу ТеГР можна побачити на рисунку 1.2, які відображають інтуїтивно зрозуміле використання програми та її розширені можливості.



Рисунок 1.2 – Приклад роботи програми ТеГР

Мобільний додаток VisPA (Візуальні попарні порівняння), спеціально розроблений для операційної системи Android, відкриває перед користувачем можливість візуального порівняння різних альтернатив за допомогою методу попарного порівняння. Цей додаток забезпечує зручний та інтуїтивний інтерфейс для впорядкування та оцінки альтернатив. Результати порівнянь відображаються у формі графіків, де значення кардинально центровані, надаючи зрозумілу та структуровану інформацію.

Цей мобільний додаток дозволяє користувачам взаємодіяти з різними альтернативами та отримувати комплексні результати, які полегшують процес прийняття рішень. Скріншоти роботи застосунку можна розглянути на рисунку 1.3, де чітко представлені можливості та функціонал додатку, дозволяючи зрозуміло використовувати його для визначення найбільш придатних варіантів.

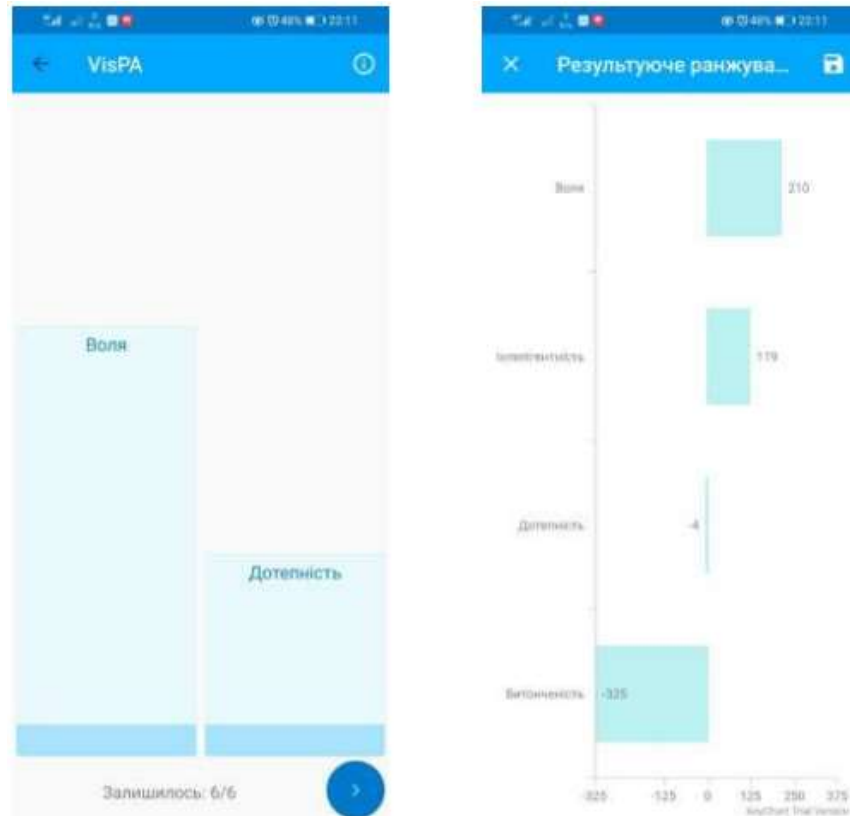


Рисунок 1.3 – Приклад роботи програми VisPA

Інтерактивна система визначення переваг (ІСВП) є інноваційним інструментом, спрямованим на систематизацію множини альтернатив через використання тернарних ранжувань. Цей підхід відрізняється від звичайних попарних порівнянь тим, що дозволяє враховувати додаткову інформацію від особи, що приймає рішення (ОПР). ІСВП створена з метою структурування великої кількості альтернатив, допомагаючи ОПР у визначенні їхнього відносного

значення. Використання тернарних ранжувань дозволяє більш глибоко враховувати індивідуальні вподобання та особисті критерії прийняття рішень. Ця інтерактивна система надає зручний інтерфейс для взаємодії з користувачем та забезпечує ефективний аналіз та управління великим обсягом інформації.

При виконанні процесу ранжування важливо розміщувати кожен з трьох представлених альтернатив на віртуальних "полицях" переваг. За такою системою, чим вище розташована альтернатива на полиці, тим вища якість відповідного варіанту. У випадку, коли дві або всі три альтернативи володіють однаковою якістю, їх слід спільно розміщувати на одній полиці.

У випадку, коли одна з альтернатив переважає за якістю інші, її важливо розмістити вище на "полиці" для чіткої індикації її переваги. Якщо різниця в якості між альтернативами значна, рекомендується розташовувати більш високоякісну альтернативу мінімум на дві "полиці" вище.

Цей підхід допомагає візуально відобразити важливість та відмінності між альтернативами, забезпечуючи чітке розуміння їхнього порівняльного рівня якості.

Процес розміщення альтернатив на "полицях" переваг визначається індивідуальними вподобаннями особи, що приймає рішення (ОПР). Після завершення ранжування, важливо натискати кнопки "Завершити ранжування" та "Продовжити" для збереження та переходу до наступних етапів. Цей підхід дозволяє забезпечити гнучкість інструменту, враховуючи індивідуальні схильності та уподобання кожного користувача.

Після завершення всіх ранжувань ІСВП виводить результативне ранжування для всієї множини альтернатив, яке відображає загальні вподобання та переваги ОПР. Скріншоти роботи програми, які подані на рисунку 1.4, ілюструють простоту та ефективність інтерфейсу, а також якісний результат після завершення процесу ранжування.

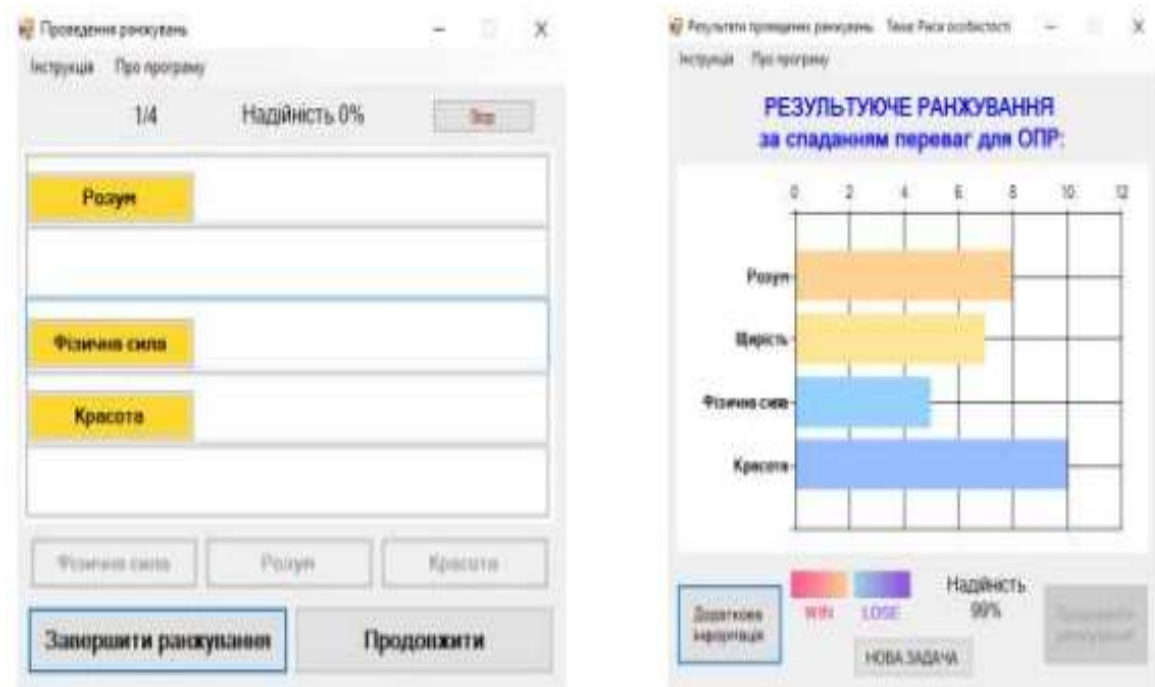


Рисунок 1.4 – Приклади ранжування та результатів програмою ІСВП

Користувачам, які виявляють зацікавленість у вивченні деталей та отриманні розрахункової інформації, надається зручна можливість ознайомитися з відповідними аспектами, спрямованими на системний аналіз. За допомогою скріншотів роботи програми, які зображено на рисунку 1.5, відкривається доступ до обширного спектру розрахунків та аналізу, які лежать в основі утворення кінцевого ранжування.

Цей доступ дозволяє системним аналітикам глибше поглибитися в суть критеріїв, вагових коефіцієнтів та інших параметрів, які визначають окремі результати. Скріншоти не лише відображають роботу програми, але й стають візуальним ключем до розуміння та аналізу процесів, що відбуваються в системі. Такий підхід сприяє отриманню докладної інформації та вирішенню складних завдань, забезпечуючи аналітикам всі необхідні засоби для розкриття глибинних аспектів роботи програми та формування фундаментальних висновків.

Зазначена можливість ретельного огляду розрахунків та аналізу, які становлять основу результуючого ранжування, підкреслює нашу відданість

наданню користувачам всебічної та деталізованої інформації для максимального розуміння та ефективного використання даного програмного продукту.

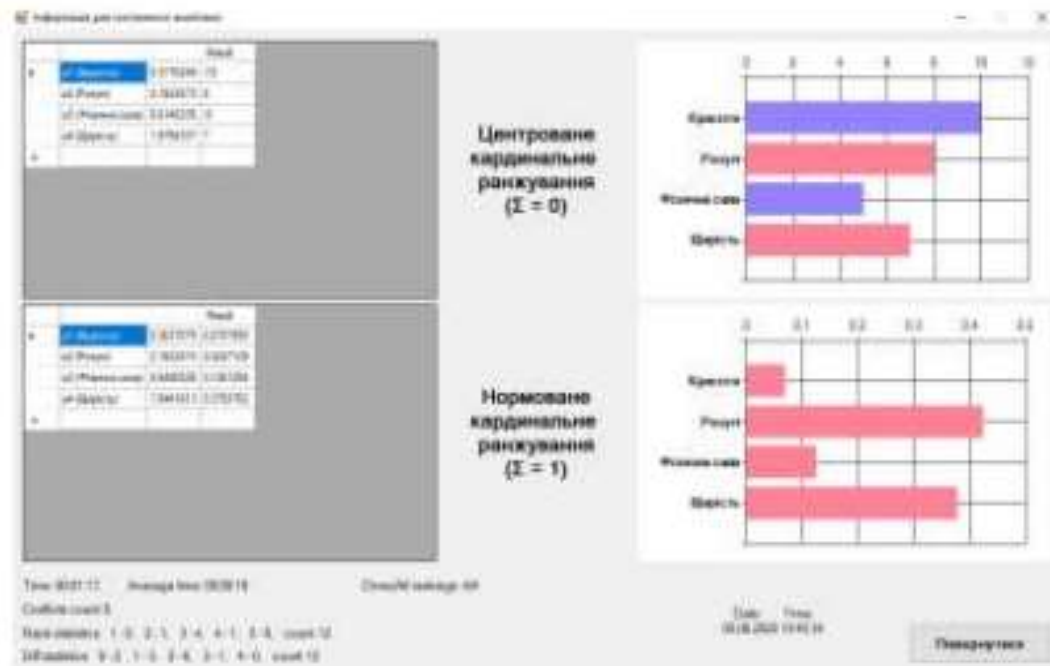


Рисунок 1.5 – Приклад детальної інформації ранжування ІСВП.

Абсолютно, розглядаючи подані аналогії, можна визначити, що мобільний додаток стає найбільш зручним і сучасним інструментом для реалізації програм такого типу. Однією з його ключових переваг є мобільність — завдяки цьому додатку ви маєте постійний доступ до інструментів оцінки та ранжування альтернатив, незалежно від вашого місця перебування. Це робить процес взаємодії з програмою максимально зручним і пристосованим до сучасного темпу життя.

Мобільні додатки забезпечують гнучкість і доступність, що робить їх ідеальними інструментами для використання у будь-який момент часу. Крім того, їхні інтуїтивні інтерфейси та можливості взаємодії роблять їх досить простими у використанні, що є додатковим плюсом для користувачів.

1.3 Постановка задачі

Постановка завдання для розробки інформаційної технології для ранжування зображень ґрунтується на попередніх обговореннях та доступних даних. Основна мета цього проекту полягає в поліпшенні процесу ранжування зображень шляхом забезпечення швидкості, ефективності та автоматизації його етапів.

Ця інформаційна технологія має вдосконалити процес прийняття рішень, пов'язаних із зображеннями, шляхом використання новаторських методів та алгоритмів. Очікується, що вона буде здатна автоматизувати процес оцінювання та ранжування зображень, зменшуючи часові та людські витрати.

Основні вимоги, що пред'являються до нашої розробленої технології, визначаються низкою ключових критеріїв, які націлені на забезпечення високої ефективності та універсальності. Перш за все, ми ставимо перед собою завдання забезпечити швидкість обробки, щоб користувачі могли отримувати результати в найкоротший термін.

Висока точність ранжування є ще однією важливою вимогою, яку ми враховуємо при розробці. Наша технологія спрямована на те, щоб надавати користувачам надійні та точні результати, що відображають суттєві аспекти аналізу зображень.

Зручний інтерфейс користувача є не менш важливим аспектом, оскільки ми прагнемо до того, щоб користувачі мали легкий та приємний досвід взаємодії з програмним продуктом. Інтуїтивно зрозумілі елементи управління та навігації роблять користування нашою технологією максимально зручним.

Можливість автоматичного аналізу зображень з використанням передових технологій є важливим фактором, оскільки це дозволяє розширити сфери застосування нашого інструменту. Проект орієнтований на створення інноваційного інструменту, який може бути успішно використаний у різних галузях, де важлива точна та об'єктивна оцінка зображень, від медичної діагностики

до визначення якості виробів та багатьох інших областей. Такий підхід дозволяє нам впроваджувати передові технології для досягнення максимальної користі для користувачів у різноманітних сценаріях використання. Одним з головних завдань є створення зручного та ефективного інструменту для організаторів, який дозволить їм створювати, налаштовувати та аналізувати процес ранжування. Цей інструмент має забезпечувати ефективний та чіткий аналіз результатів ранжування, а також надавати доступ до інформації для всіх зацікавлених сторін. Додатковою задачею є розробка модуля для аналізу результатів ранжування з метою отримання відповідного рейтингу. Цей модуль повинен проводити перевірки з використанням різних нечітких алгоритмів, щоб забезпечити узагальнену інформацію з аналізу.

Під час розробки інформаційної технології важливо також враховувати можливість її інтеграції з іншими платформами та сервісами. Це дозволить забезпечити зручну взаємодію з існуючими системами та ефективний обмін даними між різними компонентами технології.

Інтеграція з іншими платформами може включати в себе можливість використання стандартних протоколів обміну даними, таких як API (інтерфейс програмування застосунків), що сприяє взаємодії технології з іншими програмами та сервісами. Також, важливо врахувати сумісність із різними операційними системами та забезпечити зручність використання технології для користувачів на різних пристроях.

Це дозволить створити гнучкий та універсальний інструмент, який може інтегруватися в різноманітні екосистеми та максимально задовольняти потреби користувачів у спільній обробці та обміні зображеннями.

Загальна мета постановки задачі полягає у створенні інформаційної технології, що задовольняє потреби користувачів, які цікавляться ранжуванням зображень з метою отримання чіткого рейтингу на основі відповідей респондентів. Для досягнення цієї мети потрібно вирішити наступні завдання:

1. Розробити покращену математичну модель ранжування зображень.

2. Розробити вдосконалений метод ранжування зображень.
3. Розробити алгоритм функціонування модулів інформаційної технології.
4. Розробити загальну структуру інформаційної технології.
5. Обґрунтувати вибір інструментів для технічної реалізації інформаційної технології ранжування зображень.
6. Реалізувати інформаційну технологію організації заходів з використанням обраної мови програмування.
7. Провести тестування інформаційної технології та проаналізувати отримані результати.

1.4 Висновок до розділу 1

Проведено детальний аналіз тематичної області ранжування, прослідковуючи інформацію про відомі методи та алгоритми для оцінювання альтернатив. Відзначено, що проведений огляд включав розгляд основних концепцій, лежать в основі ранжування, та вивчення популярних технік у виборі найкращих альтернатив.

В рамках аналізу були розглянуті інноваційні програмні аналоги, що реалізують візуальне ранжування альтернатив. Це дозволило отримати загальне уявлення про те, які підходи та інструменти активно використовуються в цій області. Вивчений матеріал надав не лише теоретичні концепції, а й практичні приклади програм, які реалізують ці концепції.

Проведений аналіз виявив ключові принципи, що допомагають в оцінці та ранжуванні альтернатив, а також виокремив популярні програмні рішення, які активно використовуються в даній сфері. Цей огляд дозволяє зорієнтуватися у сучасних можливостях та тенденціях у галузі ранжування альтернатив.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ЗОБРАЖЕНЬ

2.1 Уніфікована мова моделювання UML в процесі розробки інформаційної технології візуального ранжування зображень

З середини 1960-х і до недавнього часу структурні методології аналізу, проектування та розробки інформаційних систем характеризуються штучним (часто неоптимальним) поділом системи на підсистеми, а також слабким зв'язком процесів і даних, присутніх в системі.. Навпаки, об'єктно-орієнтовані технології, зосереджені на тісному взаємозв'язку між процесами та даними в системах, дозволяють програмним системам бути більш надійними, легшими у реалізації та більш стійкими до змін. Більше того, ця філософія моделювання найбільше відповідає загальним уявленням про поведінку систем реального світу [6].

Незважаючи на явну перевагу технологій об'єктно-орієнтованого аналізу та проектування над структурними технологіями, їх поширення було незначним, оскільки жоден із методів не привів до створення єдиної і комплексної моделі об'єктно-орієнтованої системи. Кожен метод добре охоплював один або кілька аспектів реальної системи, залишаючи багато інших, не менш важливих аспектів в тіні. Більше того, відсутність єдиного стандарту суттєво заважала широкому застосуванню об'єктно-орієнтованих методів у розробці програмного забезпечення [6].

UML (англ. Unified Modeling Language) — це уніфікована мова моделювання, яка використовується в парадигмі об'єктно-орієнтованого програмування. Це невід'ємна частина єдиного уніфікованого процесу розробки програмного забезпечення. UML – це мова широкого профілю, відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, яка називається моделлю UML. UML створено для визначення, візуалізації,

проектування та документування переважно програмних систем. UML не є мовою програмування, але шляхом створення моделей UML як інтерпретованого коду можна генерувати код [6].

Перша версія (1.0) UML була випущена 13 січня 1997 року від імені Object Management Group (OMG), організації, відповідальної за прийняття стандартів об'єктно-орієнтованих технологій і баз даних.

UML можна використовувати на всіх етапах життєвого циклу Business Intelligence та розробки додатків. Різноманітність типів діаграм, які підтримує UML, і найбагатший набір можливостей для представлення деяких аспектів системи роблять UML універсальним інструментом для опису як програмного забезпечення, так і бізнес-систем. Діаграми дають змогу представити систему (як бізнес, так і програмне забезпечення) таким чином, що її можна легко перевести в програмний код [6].

Основною причиною використання UML є спілкування між розробниками. Крім того, UML був створений спеціально для оптимізації процесу розробки програмного забезпечення, що дозволяє в кілька разів підвищити ефективність їх впровадження і значно підвищити якість кінцевого продукту. UML зарекомендував себе в багатьох успішних програмних проектах. Засоби автоматичного генерування коду перетворюють моделі UML у вихідний код об'єктно-орієнтованих мов програмування, ще більше прискорюючи процес розробки. Майже всі інструменти CASE (програмне забезпечення для аналізу проектування та автоматизації) мають підтримку UML. Моделі, розроблені в UML, дозволяють значно спростити процес кодування і спрямувати зусилля програмістів безпосередньо на реалізацію системи. Діаграми підвищують підтримку проекту та полегшують створення документації [6].

UML необхідний:

- менеджерам проектів, які керують розподілом завдань і контролем за проектом

- проектувальникам інформаційних систем, які займаються розробкою технічних завдань для програмістів;
- бізнес-аналітикам, які аналізують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії;
- програмістам, які займаються реалізацією модулів інформаційної системи.

При модифікації системи об'єктний підхід легко інтегрує в систему нові потужності та ліквідовує застаріле без істотних змін його рентабельності. Використання побудованої моделі в модифікаціях системи дозволяє усунути небажані наслідки змін, оскільки вони не порушують структуру системи, а лише змінюють поведінку об'єктів [6].

Створення діаграми прецедентів (Use Case Diagram, діаграма використання) є невідемним процесом проектування інформаційних технологій. В єдиній мові моделювання (UML) діаграма використання може підсумовувати деталі користувачів вашої системи (також відомих як актори) та їх взаємодії з системою. Для створення даної діаграми, буде використовуватися набір спеціалізованих символів та роз'ємів. Діаграма ефективного використання допомагає обговорити та представити:

- Сценарії, в яких система або програма взаємодіє з людьми, організаціями чи зовнішніми системами;
- Цілі, яких система чи додаток допомагає досягти цим суб'єктам (відомим як актори);
- Обсяг системи.

Основними елементами діаграми прецедентів є актори, система та цілі:

- Актори. Користувачі, які взаємодіють із системою. Актором може бути особа, організація або зовнішня система, яка взаємодіє з вашим додатком або системою. Вони повинні бути зовнішніми об'єктами, які виробляють або споживають дані [12];

- Система. Конкретна послідовність дій та взаємодії між акторами та системою. Система також може називатися сценарієм;
- Цілі. Кінцевий результат більшості випадків використання. Успішна діаграма повинна описувати діяльність та варіанти, які використовуються для досягнення мети.

Діаграма прецедентів не є деталізованою – наприклад вона не моделює порядок виконання кроків. Натомість, діаграма правильного використання зображує високооглядний огляд взаємозв'язку між варіантами використання, суб'єктами та системами. Експерти рекомендують використовувати діаграми варіантів використання для доповнення більш описового текстового випадку використання.

UML – це інструментарій моделювання, який можна використовувати для побудови своїх діаграм. Випадки використання зображені з позначеною овальною формою. Фігурки представляють акторів процесу, а участь актора в системі моделюється лінією між актором та варіантом використання. Щоб зобразити межу системи, намалюйте рамку навколо самого варіанта використання.

Діаграми варіантів використання UML ідеально підходять для:

- Представлення цілей взаємодії системи та користувача;
- Визначення та організація функціональних вимог у системі;
- Визначення контексту та вимог до системи;
- Моделювання основного потоку подій у випадку використання.

2.2 Математична модель візуального ранжування зображень

В основі інформаційної технології лежить метод безпосередньої оцінки. Він полягає в тому, що діапазон зміни будь-якої кількісної змінної поділяється на інтервали, кожному з яких надається певна вага, наприклад від 1 до 5. Шкала може містити як позитивні так і негативні оцінки, наприклад -5 до +5. Кожний об'єкт

включається в певний інтервал в залежності від його значення. Число інтервалів може бути різним у різних експертів.

В деяких випадках зручніше спочатку вибрати найбільш переважаючий фактор, а потім ранжування.

Сумарні оцінки можуть піддаватись нормалізації. Це дозволяє встановити зв'язок між оцінками які експерти надали об'єктам. Для цього кожна оцінка ділиться на суму оцінок по всім об'єктам.

При проведенні колективної експертизи потрібно отримати усереднену оцінку для кожного фактора. Для цього розраховується середнє арифметичне (M):

$$M = \frac{\sum V}{N}, \quad (2.1)$$

де V – індивідуальна думка кожного експерта;

\sum - знак суми;

n – кількість експертів.

Задача полягає в тому щоб обрати таке рішення з множини A можливих варіантів, що максимально задовольняє переваги експертів.

Систему переваг експертів можна подати у вигляді:

$$P = \langle A, Q, K, F^Q, N \rangle, \quad (2.2)$$

де Q – множина критеріїв оцінювання якості рішень;

K – множина шкал;

F^Q – відображення множини альтернатив A в множину Q .

N – множина експертів.

Множина A – це сукупність альтернатив, що відповідні можливим способам досягнення мети і не порушують певних обмежень.

Множина критеріїв Q відображає аспекти мети роботи системи які є суттєвими для експертів.

Множина шкал K відповідає кожному критерію множини шкали Q , в якому вимірюються значення факторів.

Відображення F^Q відповідає кожній альтернативі множини A в просторі критеріїв.

Система переваг P – це елемент який певною мірою зумовлює слабку структурованість задач. Апроксимація системи переваг експертів в більшості випадків дозволяє за певних припущень спростити умови для побудови вирішального правила R або описати механізм вибору альтернативи з множини A .

2.3 Розробка структури інформаційної технології візуального ранжування зображень

Структура інформаційної технології відображає призначення та взаємодію головних структурних модулів програми між собою та з користувачем.

В веб-додатку для візуальних ранжувань є наступні модулі: модуль створення множини альтернатив, модуль процесу ранжування множини альтернатив, модуль перегляду та відображення результатів ранжувань, ранжування для подальшої можливості їх використання. Точкою входу в програму є домашня сторінка, з якої починається навігація програмою по інших її складових.

Розроблена структура інформаційної технології візуального ранжування альтернатив зображено на рисунку 2.1.



Рисунок 2.1 – Структура інформаційної технології візуального ранжування альтернатив

Ця схема відображає роботу застосунку на найвищому рівні абстракції, тому варто також звернути увагу на функціонування веб застосунків з точки зору їх життєвого циклу.

Життєвий цикл застосунку контролюється браузером і залежить від потреб користувача, доступних ресурсів, тощо. Як тільки користувач потрапляє на сторінку, браузер починає процес завантаження сценаріїв JavaScript і подальше їх опрацювання. Далі, після опрацювання всіх сценаріїв, починається процес рендеру, під час якого і створюється користувацький інтерфейс. Браузер рендерить сторінку розумно, без повторювань певних процедур і з подальшою оптимізацією роботи додатку. Таким чином додаток споживає набагато менше ресурсів і працює швидше. Після всіх цих процесів, починається етап завантаження даних із серверу. Після отримання даних по HTTP протоколу, додаток може працювати з цими даними за певними написаними сценаріями.

Всі вище описані модулі та компоненти програми які відповідають за перемикання візуальних компонентів наслідуються від батьківського класу `React.Component`, що реалізує властивості життєвого циклу React застосунку.

Для більш детального опису компонентів програми та зв'язків між ними використаємо діаграму класів, для застосунку візуального ранжування картинок вона зображена на рисунку 2.2.

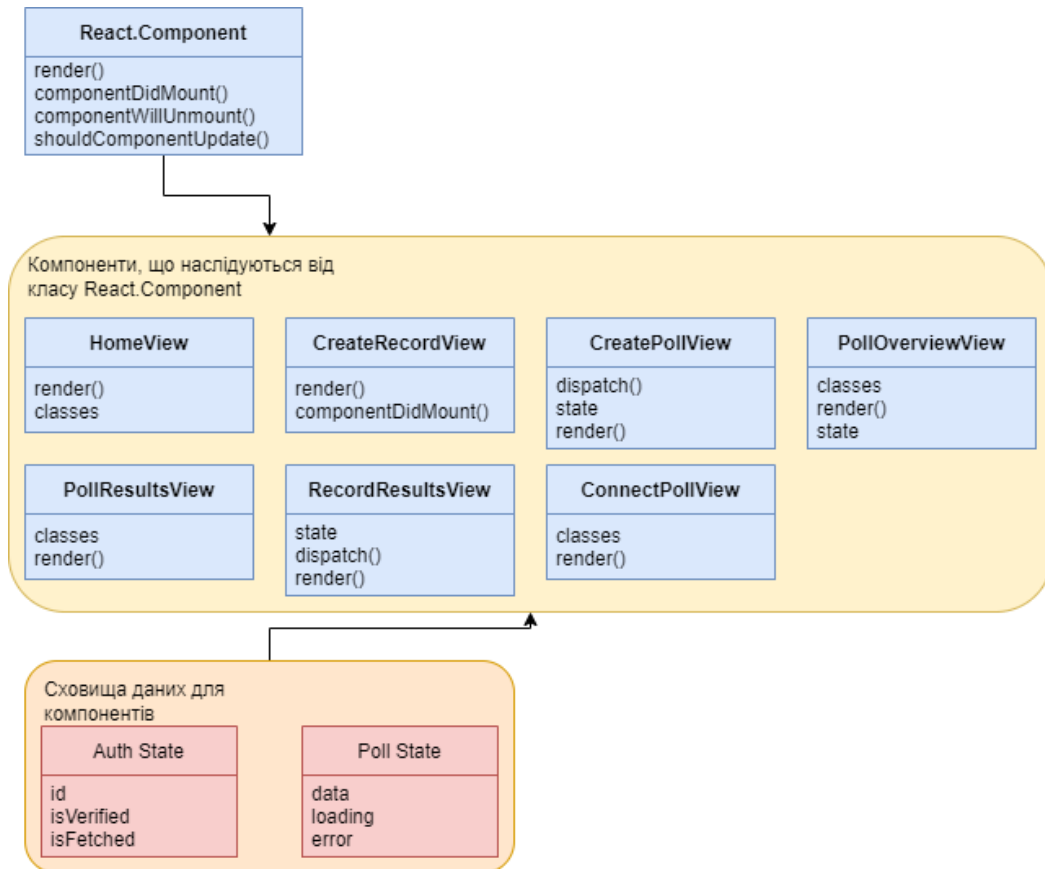


Рисунок 2.2 – Діаграма класів інформаційної технології візуального ранжування картинок

Діаграма прецедентів, що зображена на рисунку 2.3, показує взаємодію користувача із застосунком, його можливі дії та реакції системи на них.

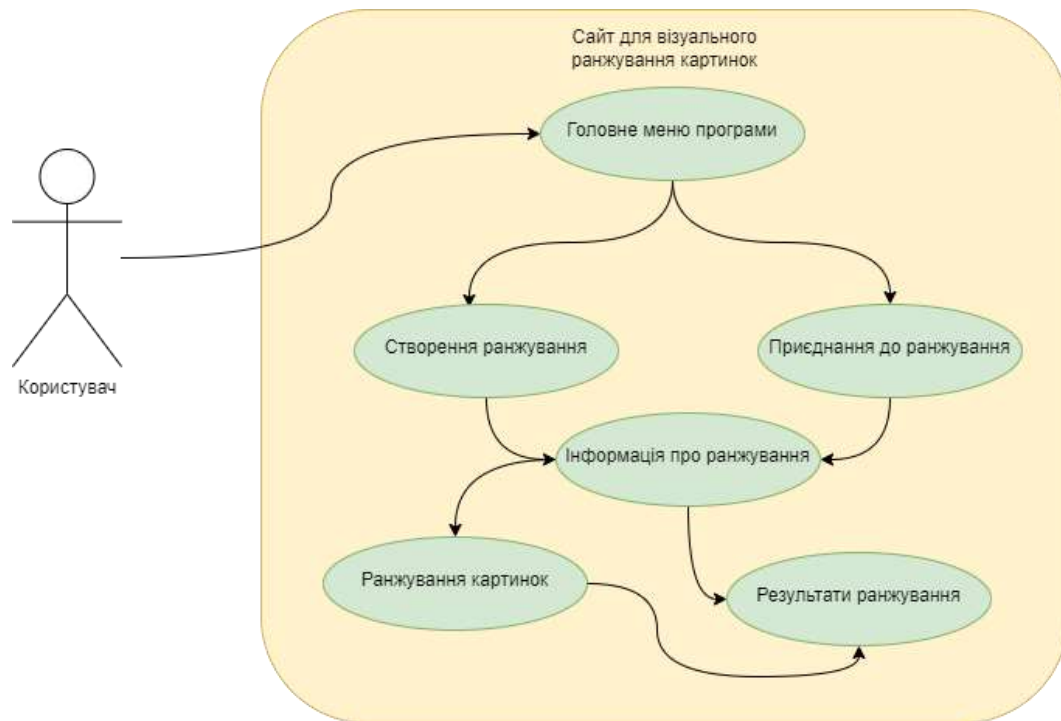


Рисунок 2.3 – Діаграма прецедентів інформаційної технології візуального ранжування картинок

Збереження даних у базу даних MongoDB є ключовим аспектом нашого веб-сайту, надаючи надзвичайну ефективність та гнучкість обробки інформації. MongoDB, як документно-орієнтована система управління базами даних, надає нам можливість зберігати дані в форматі, що відображає структуру документів. Це значно спрощує процес організації та взаємодії з інформацією.

У форматі документів MongoDB, дані структуруються у вигляді JSON-подібних об'єктів, що сприяє легкому читанню та розумінню структури. Це дозволяє нам ефективно використовувати дані в наших додатках, виконуючи операції читання, запису та оновлення з високою продуктивністю.

Безпека та надійність MongoDB грають також важливу роль у забезпеченні стійкості системи. Механізми автентифікації та авторизації, а також можливості шифрування, що вбудовані у цю базу даних, дозволяють нам зберігати та обробляти дані в безпечний спосіб.

Цей підхід до зберігання даних в базі даних MongoDB покликаний не лише забезпечити оптимальну продуктивність та легкість взаємодії, але й гарантує стабільність та захищеність інформації, що є критичним для впевненості у високому рівні обробки та зберігання даних нашого веб-сайту.

Діаграма діяльності даного модуля наведена на рисунку 2.4.

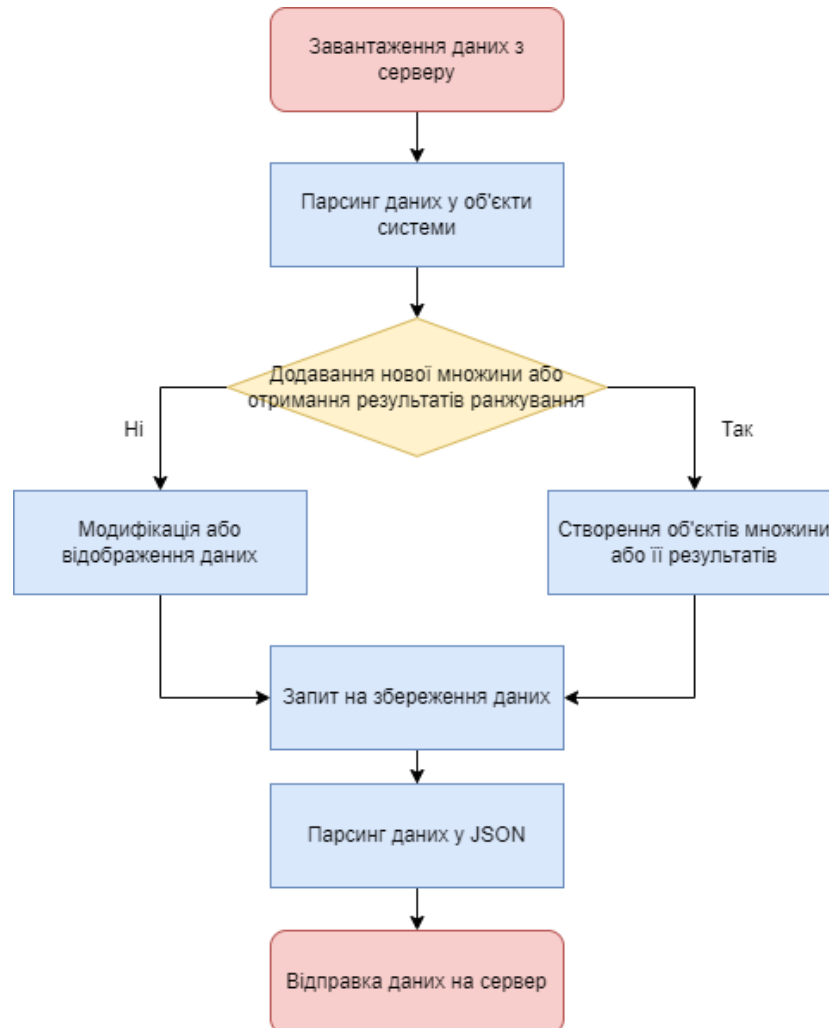


Рисунок 2.4 – Діаграма діяльності роботи модуля зберігання даних

Процес збереження даних на кожному етапі створення або редагування є важливою функцією нашого веб-сайту, спрямованою на забезпечення повноцінності та надійності інформації. Цей механізм дозволяє користувачам

уникнути втрати даних та повторного введення інформації, забезпечуючи безперервність їх робочого процесу.

Під час створення нових записів або внесення змін у існуючі дані, система автоматично здійснює процес збереження. Це дозволяє нам впевнено гарантувати, що навіть при непередбачуваних обставинах чи випадках втрати з'єднання, дані будуть надійно збережені.

Такий підхід до автоматичного збереження даних на кожному етапі взаємодії з системою сприяє ефективності та зручності для користувача. Крім того, це є важливим елементом забезпечення цілісності інформації, дозволяючи користувачам спокійно працювати з додатком, знаючи, що їх дані будуть надійно та автоматично збережені на будь-якому етапі їх введення чи редагування.

2.4 Висновок до розділу 2

У даному розділі вивчалися особливості та основні можливості уніфікованої мови моделювання UML. Були розглянуті основні поняття та особливості роботи веб-застосунків. Здійснено розробку схеми функціонування, діаграми класів, діаграми прецедентів та діаграми діяльності роботи веб-застосунку. Проведений аналіз дозволив визначити структуру та взаємодію складових системи, а також надав загальне уявлення про архітектурні аспекти UML. Розглянуті діаграми відображають ключові взаємодії та функції веб-застосунку, що сприяє кращому розумінню їх роботи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РАНЖУВАННЯ ЗОБРАЖЕНЬ

3.1 Обґрунтування вибору мови програмування для розробки інформаційної технології ранжування зображень

Перед початком реалізації інформаційної технології слід вибрати для використання правильну мову веб-програмування. Варто проаналізувати чому в конкретному випадку слід вибирати певні мови замість інших. З моменту виникнення Інтернету веб-програмування залишалось популярним і затребуваним навиком. У суспільстві, яке керується Інтернетом, програмування для Інтернету продовжує свою тенденцію до зростання.

Вкрай важливим є вибір правильної мови веб-програмування, що найкраще підійде під певну задачу. Існує кілька факторів, що визначають, яка мова є найкращою. Ось як вибрати правильну мову веб-програмування для використання. Вибір «правильної» мови може ввести в оману. Це означає, що існують неправильні мови. Хоча деякі мови просто не сприяють реалізації ваших задач у програмуванні, існує багато різних методів досягнення бажаного результату. Особливо це стосується веб-програмування. Існують переважно веб-орієнтовані мови, такі як JavaScript, а також мови загального призначення, як-от Java. Так як прийнято рішення реалізувати інформаційну технологію у вигляді веб-додатку зосередимося на питаннях, які посприяють у визначенні потрібної мови веб-програмування.

Для створення сервера для візуального ранжування картинок необхідно обрати мову програмування, яка володіє відповідними бібліотеками, середовищем розробки і відповідає всім вимогам для додатку. З різних мов серверної розробки, які підходять для створення сервера невеликих розмірів з мінімальним функціоналом отримання і віддачі даних, можна вибрати C#, Java і JavaScript, PHP.

Мова програмування PHP існує вже більше двох десятиліть і зарекомендувала себе як потужне і надійне рішення, здобувши армію прихильників і шанувальників. Однак ми повинні визнати, що ця мова кодування поступово втрачає свою популярність на користь новіших і оптимізованих аналогів. Назва PHP означає Hypertext Preprocessor і позначає мову сценаріїв на стороні сервера, що означає, що програми, написані на ньому, працюють на веб-серверах і не залежать від веб-браузера. Проте з роками сфера його використання змінилася, і сьогодні мова кодування PHP входить до числа найкращих і найпопулярніших інструментів програмування для веб-розробки завдяки своїм багатьом перевагам. Це вважається дуже ефективною технологією, яка пропонує зручний процес розробки з багатьма додатковими інструментами, які допомагають йому. Насправді, згідно з індексом популярності мови програмування (PYPL), PHP є п'ята за популярністю мова кодування в світі.

Мова PHP в основному використовується для веб-розробки, і вона справді хороша у цій області. Хоча спочатку вона використовувалась для створення динамічних веб-сторінок, розробники вважають за краще використовувати цю мову сценаріїв для створення серверної частини веб-додатків. Однак для початку PHP є мовою загального призначення, тому, якщо потрібно, вона може мати інші реалізації. Наприклад, можна створювати настільні програми за допомогою PHP. Більше того, починаючи з версії 5, PHP підтримує об'єктно-орієнтоване програмування, пропонуючи цілий новий набір можливостей [17].

Універсальність цієї мови сценаріїв є результатом її чудової здатності поєднуватися з іншими мовами програмування. Наприклад, розробники можуть писати розширення до PHP, використовуючи мову C, що дозволяє додати ще більше функцій. Крім того, PHP має велику кількість доступних бібліотек і фреймворків, які ще більше розширюють його можливості. Найпопулярнішими прикладами таких фреймворків є Laravel, Symfony, Phalcon, Zend Framework і Yii.

Два з найвідоміших прикладів програмного забезпечення, написаного на PHP – це Facebook і Wordpress. Wordpress є найпопулярнішою системою керування вмістом в Інтернеті: серед усіх веб-сайтів, які використовують такі системи, близько 48% з них використовують Wordpress. Завдяки великій кількості плагінів, як вбудованих, так і сторонніх, Wordpress підходить практично для будь-якої ролі. Ви можете використовувати його для створення блогу, фотогалереї, інтернет-магазину, порталу новин та багатьох інших типів веб-сайтів.

Facebook є ще одним очевидним прикладом того, що може робити PHP. Ця соціальна мережа вже давно перетворилася на складне середовище, яке має різноманітний спектр функцій, включаючи обмін миттєвими повідомленнями, рекламу, ведення блогів, презентацію новин, відтворення відео та інші, але все ще базується на коді PHP, тісно пов'язаному з іншими веб-технологіями [18].

Популярність мови PHP є логічним результатом її численних переваг, які роблять її потужним та ефективним інструментом розробки. Розглянемо переваги PHP:

1. Великий вибір доступних спеціалістів. Популярність PHP породила численну спільноту розробників, частина з яких може бути потенційними кандидатами на роботу. Велика кількість наявних спеціалістів призводить до високої конкурентоспроможності та меншої затребуваної заробітної плати, що вигідно для зниження витрат на розробку. Крім того, цю мову досить легко вивчити та впровадити, тому навіть молодші розробники зазвичай можуть ефективно реалізувати базову функціональність програми.

2. Велика документація. Доступно багато навчальних посібників, посібників та інших довідкових матеріалів, які полегшують розробку та можуть стати допомогою та джерелом натхнення у важких ситуаціях. Ці матеріали також безцінні для початківців програмістів, що полегшують процес навчання з поступово зростаючими труднощами. І, як було описано вище, вивчення PHP є порівняно

легким, хоча й не таким легким, як вивчення Python, наприклад, що ще більше додає його популярності.

3. Покращена швидкість завантаження. Використання PHP прискорює завантаження сторінок веб-сайтів у порівнянні з багатьма іншими технологіями веб-розробки. Наприклад, наразі PHP приблизно втричі швидший за Python для більшості сценаріїв використання. У свою чергу, менший час завантаження є важливим фактором рейтингу SEO, який сприяє подальшому просуванню веб-сайту, забезпечуючи конкурентні переваги. Вища швидкість додатків забезпечує задоволення клієнтів і, у поєднанні з іншими перевагами, допомагає створити та зберегти клієнтську базу.

4. Широкий вибір баз даних. PHP дозволяє підключатися практично до будь-якого типу бази даних. Найпоширенішим вибором є MySQL, головним чином тому, що він безкоштовний, ефективний і популярний серед розробників. Інші надійні варіанти систем керування базами даних, сумісні з PHP – це mSQL, MS-SQL, SQLite, PostgreSQL тощо. Крім того, PHP можна однаково добре використовувати з Elasticsearch, Redis, MongoDB та іншими нереляційними базами даних. Таким чином, розробники не обмежуються використанням конкретної бази даних і можуть вибрати найбільш оптимальну для майбутнього додатка, враховуючи всі важливі фактори.

5. Недороге програмне забезпечення з відкритим кодом. PHP — це безкоштовна технологія, яка дає значну економію бюджету на розробку. Крім того, більшість інструментів розробки, які зазвичай використовуються в поєднанні з PHP, є програмним забезпеченням з відкритим кодом і можуть використовуватися безкоштовно; тим самим вони додатково знижують вартість проекту. Крім того, існують численні фреймворки, такі як Laravel і CodeIgniter, а також різні CMS, такі як Wordpress і Drupal, наприклад, які розширюють функціональність PHP і роблять процес розробки простішим і ефективнішим.

6. Дешевші послуги хостингу. Найпоширенішим сценарієм роботи веб-сайту PHP є стек LAMP. Це означає, що веб-сайт працює на веб-сервері Apache HTTP, розгорнутому в системі Linux, і використовує MySQL як базу даних. Всі ці компоненти безкоштовні, а стек добре перевірений, що передбачає скорочення часу та коштів на розробку.

7. Відмінна комбінованість з HTML. PHP пропонує вбудоване програмування HTML, що є причиною неймовірної синергії між цими двома технологіями. У більшості випадків PHP-скрипт не втручається в HTML-код веб-сторінки, а завершує його, залишаючись у межах, визначених тегами `<?php ?>` [18].

8. Хороша гнучкість. Завдяки гнучкості PHP може ефективно поєднуватися з багатьма іншими мовами програмування, щоб програмний продукт міг використовувати найбільш ефективну технологію для кожної окремої функції. Крім того, PHP є міжплатформною мовою, що означає, що розробники можуть використовувати будь-яку основну операційну систему – Windows, Linux, macOS – для виконання кодування. Така гнучкість значно полегшує процес розробки, роблячи його швидшим і менш дорогим [19].

9. Сумісність з хмарними сервісами. Нині багато сучасних продуктів, як правило, використовують рішення хмарних обчислень, як-от Amazon Web Services, для різних цілей. Програми, написані на PHP, підтримуються різними хмарними сервісами, такими як AWS Lambda, наприклад. Таким чином, PHP-додаток можна розгорнути на хмарному сервері і досягти відмінної масштабованості та інших корисних ефектів. Більш того, область хмарних обчислень не монополізована іншими мовами кодування, тому PHP зайняв своє місце в таких реалізаціях.

Хоча PHP, безсумнівно, корисний у сфері веб-розробки, він також має кілька недоліків, які не дозволяють йому домінувати в цій області. Розглянемо ці недоліки та дізнаємося, як вони можуть бути шкідливими для майбутнього програмного забезпечення та його бізнес-реалізації. Три основних недоліки PHP:

1. Зниження популярності. Хоча PHP є потужним інструментом, який підтримується великою спільнотою та великою довідковою документацією, існують простіші мови програмування для веб-програм. З цієї причини початківці розробники вважають за краще вивчати JavaScript як свою першу мову і рідко замислюються про додавання PHP до своїх навичок. Зараз PHP домінує в сегменті веб-розробки, але, швидше за все, це зміниться в майбутньому. Згодом кількість фахівців скоротиться, а початківців розробників, які пропонують базові навички за низькою ціною, буде бракувати, тому вартість продуктів, створених на PHP, ймовірно, зросте.

2. Відсутність спеціалізованих бібліотек для сучасних потреб. Наприклад, машинне навчання зараз є гарячою тенденцією, і вона точно збереже свою популярність і в найближчому майбутньому. Незважаючи на те, що PHP має свій набір бібліотек, він не може конкурувати з JavaScript у розробці веб-додатків за допомогою машинного навчання. Наразі PHP не може запропонувати настільки ж швидкі й ефективні альтернативи. Таким чином, якщо вашій програмі потрібна функціональність ML або може знадобитися це в майбутньому, коли ваш бізнес буде розширюватися, PHP не найкращий вибір.

3. Вади безпеки. Протягом багатьох років побоювання щодо безпеки продуктів на основі PHP все ще зберігаються з кількох причин. Однією з них є природа PHP з відкритим вихідним кодом, що означає, що можливі вразливості коду стають загальновідомими після їх виявлення. Таким чином, принаймні теоретично, протягом періоду між їх виявленням і виправленням у нових версіях мови програмування ці вразливості можуть бути використані будь-яким програмістом зі зловмисним умислом і відповідними навичками. Однак цей недолік можна в рівній мірі віднести до інших технологій з відкритим кодом загалом, оскільки він переважно стосується моделі з відкритим кодом, а не мови PHP зокрема [18].

C# – це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Вона володіє високою продуктивністю та ефективністю розробки програм завдяки інтеграції з платформою .NET. C# підтримує сучасні технології та включає в себе багато корисних функцій для створення серверних додатків.

Java – це об'єктно-орієнтована мова програмування, яка володіє великою популярністю і широким спектром застосувань. Вона використовується для створення різноманітних серверних додатків і має багато бібліотек та інструментів для цієї цілі. Переваги мови програмування C# включають:

- Високу продуктивність та ефективність розробки завдяки інтеграції з платформою .NET.
- Широкий спектр корисних бібліотек і фреймворків.
- Велику підтримку від Microsoft і активну спільноту розробників.

Переваги мови програмування Java включають:

- Велику популярність і широке застосування.
- Платформонезалежність, що дозволяє запускати додатки на різних операційних системах.
- Багато бібліотек і фреймворків для створення серверних додатків.

JavaScript – це динамічна об'єктно-орієнтована прототипна мова програмування, яка найчастіше використовується для створення скриптів веб-сторінок. Вона дозволяє взаємодіяти з користувачем на боці клієнта, керувати браузером, асинхронно обмінюватися даними з сервером та змінювати структуру та зовнішній вигляд веб-сторінки. JavaScript класифікується як прототипна мова з динамічною типізацією, і вона підтримує інші парадигми програмування, включаючи імперативну та частково функціональну.

При розробці веб-сайту для візуального ранжування зображень обґрунтованим вибором мови програмування є JavaScript. Окрім створення серверної частини, важливо реалізувати ефективну клієнтську сторону. На

сьогоднішній день JavaScript є єдиною мовою програмування, яку підтримують усі сучасні браузері, що робить її універсальним інструментом для фронтенд-розробки.

Обираючи одну мову програмування для всіх аспектів веб-сайту, ми спрощуємо процес розробки та забезпечуємо більшу єдність у кодові. Це дозволяє створити повноцінний веб-додаток, який легше підтримувати та розвивати.

Окрім вибору мови програмування, важливо розглянути також фреймворки, які значно спростять розробку та нададуть багато готових рішень для створення веб-сайту. Для реалізації сервера для візуального ранжування зображень обрано фреймворк Next.js. Цей фреймворк використовується як для серверної, так і для клієнтської частини додатку.

Вибір Next.js обґрунтовується його універсальністю та зручністю в розробці як серверної, так і клієнтської частини. Його можливості, такі як сервер-side rendering (SSR) та готові компоненти, дозволяють швидко втілити функціонал веб-додатку та оптимізувати його продуктивність. Такий підхід сприяє створенню зручного та ефективного середовища для візуального ранжування картинок на веб-сайті.

Next.js – це популярний фреймворк для розробки веб-додатків, який базується на JavaScript і React. Однією з ключових особливостей Next.js є можливість використовувати рендеринг як на стороні сервера, так і на стороні клієнта, що робить його ідеальним вибором для створення універсальних додатків.

Next.js надає ряд переваг, таких як:

- Універсальність: Завдяки можливості використовувати рендеринг на обох сторонах (SSR та CSR), Next.js дозволяє створювати додатки, які працюють ефективно і на сервері, і на клієнті.
- Зручний роутинг: Вбудований маршрутизатор дозволяє легко налаштовувати маршрутизацію в додатку. Автоматична оптимізація: Next.js надає

автоматичну оптимізацію завантаження сторінок, що сприяє швидкому завантаженню та високій продуктивності.

- Підтримка серверного рендерингу: Для серверного рендерингу Next.js використовує Node.js, що дозволяє створювати ефективні та SEO-дружні сторінки.
- Широкий спектр бібліотек: Спільнота розробників активно розширює функціональність Next.js за допомогою бібліотек.

Таким чином, вибір Next.js для розробки серверної і клієнтської частини додатку дозволить створити потужний та ефективний інструмент для візуального ранжування картинок, забезпечуючи високу продуктивність та зручність розробки.

3.2 Обґрунтування вибору середовища програмування для розробки інформаційної технології ранжування зображень

Для виконання розробки важливо вибрати оптимальне середовище розробки, яке відповідає потребам і особливостям проєкту. Розглянемо порівняння трьох популярних середовищ розробки: Visual Studio Code, WebStorm і Atom.

Visual Studio Code є інтегрованим середовищем розробки, розробленим компанією Microsoft (рис. 3.1). Воно відзначається такими основними характеристиками:

- Безкоштовний та з відкритим вихідним кодом;
- Розширюваність через багато додаткових плагінів та розширень;
- Підтримка багатьох мов програмування;
- Швидкий та легкий у використанні;
- Більше підходить для фронтенд-розробки, але не завжди підходить для повномасштабних серверних проєктів.

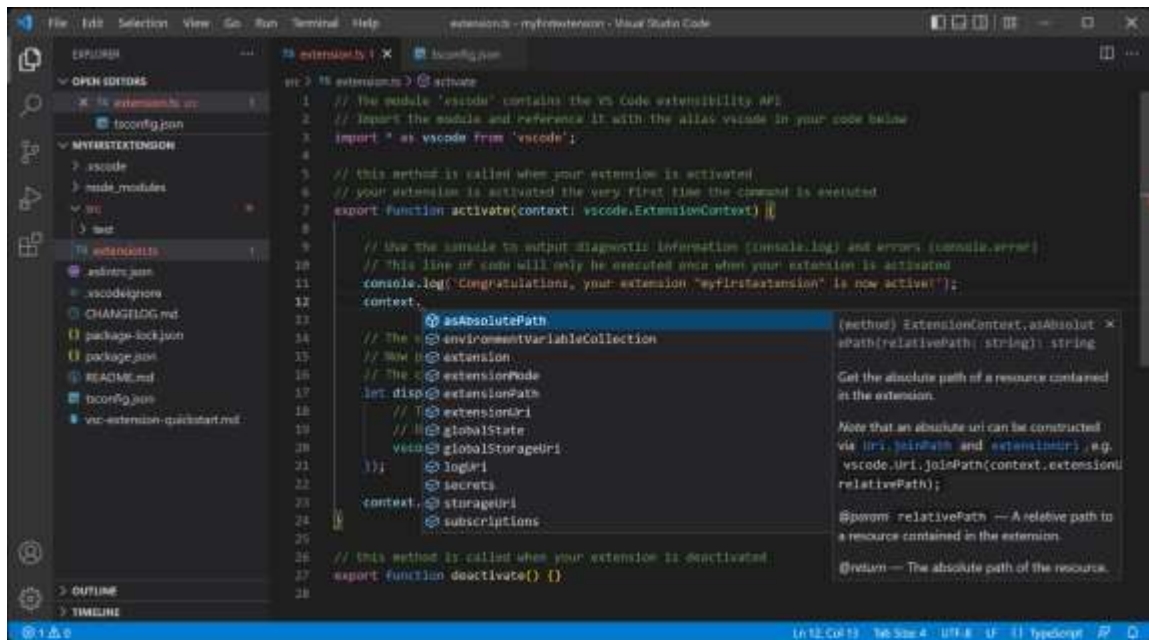


Рисунок 3.1 – Користувацький інтерфейс середовища Visual Studio Code

Atom є іншим інтегрованим середовищем розробки з відкритим вихідним кодом, яке відзначається такими характеристиками (рис. 3.2):

- Безкоштовний та з відкритим вихідним кодом;
- Розширюваність через велику кількість доступних плагінів;
- Зручний інтерфейс та дружньою спільнота розробників;
- Може бути повільним на великих проєктах;
- Може вимагати додаткового налаштування для підтримки деяких мов програмування.

Atom, вибране інтегроване середовище розробки (IDE) з відкритим вихідним кодом, представляє собою потужний інструмент, який дозволяє розробникам налаштовувати своє робоче середовище відповідно до власних потреб. Однією з ключових переваг Atom є його відкритий вихідний код, що означає, що розробники можуть вносити свої власні зміни та покращення до програми, щоб задовольнити специфічні вимоги проєкту.

Atom надає ряд вбудованих можливостей та плагінів, які полегшують роботу з різноманітними мовами програмування та технологіями. Відзначається високою ефективністю та швидкодією, Atom дозволяє розробникам швидко та зручно створювати, редагувати та відлагоджувати свій код.

Окрім цього, Atom має інтуїтивний інтерфейс, що робить його доступним для розробників будь-якого рівня. Зручний редактор коду, можливості взаємодії з системами контролю версій та інші вбудовані функції роблять його привабливим для широкого кола розробників.

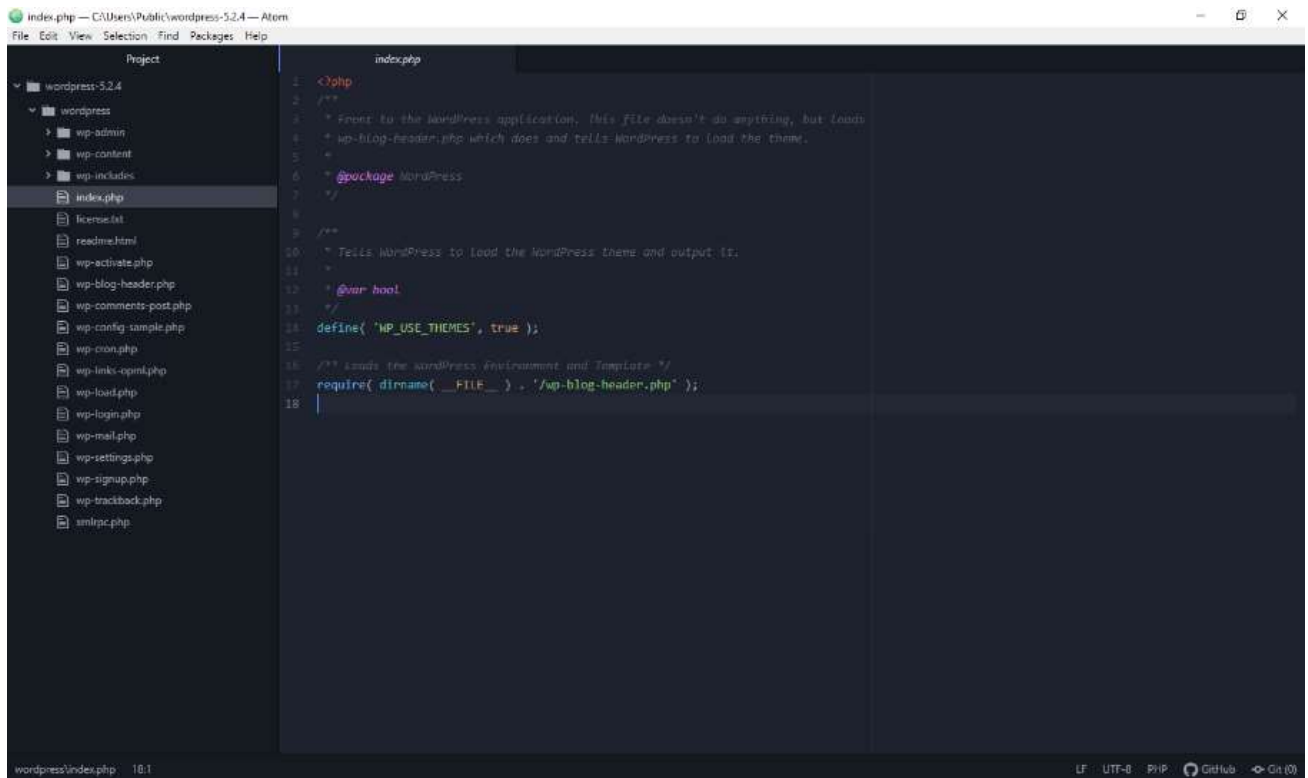


Рисунок 3.2 – Користувацький інтерфейс середовища Atom

WebStorm – це спеціалізоване середовище розробки, розроблене компанією JetBrains (рис. 3.3), і спрямоване переважно на розробку веб-додатків. Воно відзначається такими характеристиками:

- Спеціалізоване середовище для розробки веб-додатків, з основним акцентом на JavaScript, HTML та CSS;
- Вбудована підтримка популярних фреймворків, таких як React, Angular і Vue.js;
- Велика кількість корисних функцій, таких як автодоповнення, debugging та інші;
- Висока продуктивність та ефективність розробки;
- Платний продукт із пробною версією і знижкою для студентів;
- Може бути важким у використанні для розробки, не пов'язаної з веб-розробкою.

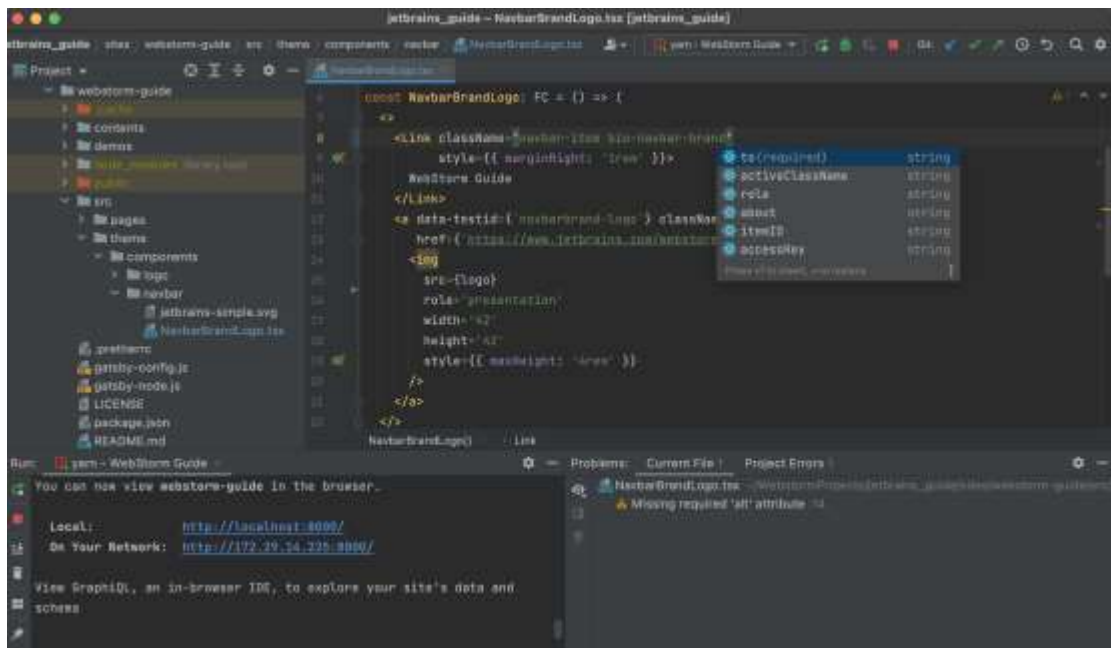


Рисунок 3.3 – Користувацький інтерфейс середовища Webstorm

WebStorm, з беззаперечно вражаючими можливостями, вибір ідеального середовища розробки для клієнтської розробки є повністю обґрунтованим. Його спеціалізовані інструменти, спрямовані на веб-розробку, роблять його вирішальним вибором для розробників, які цінують продуктивність та зручність у роботі.

Вбудована підтримка популярних фреймворків значно полегшує роботу з великим обсягом веб-коду. Це не лише забезпечує швидку інтеграцію та розширену функціональність, але й дозволяє розробникам ефективно працювати з будь-яким стеком технологій, який їм зручний. Можливість автоматичного виявлення та підтримки різних мов програмування та технологій дозволяє розробникам фокусуватися на креативності та розвитку, а не на вирішенні технічних проблем.

Крім того, WebStorm надає інтуїтивно зрозумілий інтерфейс та багатофункціональні інструменти для відлагодження, що важливо для ефективної розробки та виявлення помилок. Враховуючи всі ці переваги, можна визначити WebStorm як ідеальне середовище для клієнтської розробки, де велике значення має продуктивність та якість коду.

Також важливо відзначити, що WebStorm, хоча і є платним продуктом, але доступні пробні версії. Це надає можливість ефективно оцінити його переваги та визначити, наскільки воно відповідає потребам конкретного проекту перед прийняттям рішення щодо покупки ліцензії.

3.3 Розробка бази даних інформаційної технології ранжування зображень

Обрання MongoDB у ролі бази даних для розробки інформаційної технології ранжування зображень обумовлено кількома ключовими перевагами, які надає ця система управління базами даних.

Гнучкість сховища даних MongoDB є однією з важливих переваг, яка визначається можливістю зберігання різноманітних даних різної структури та типу. Це вкрай корисно для проєктів, де виникає потреба в роботі з різноманітними форматами даних або де структура інформації може змінюватися з часом. MongoDB

дозволяє легко адаптуватися до змін у вхідних даних, забезпечуючи при цьому стабільність та ефективність у роботі з інформацією.

Крім того, можливості MongoDB забезпечують ефективне використання для проєктів, де робиться акцент на обробці та аналізі зображень. Інтеграція даної бази даних дозволяє зберігати великі об'єми зображень та пов'язаної інформації з ефективністю та безпекою.

Загалом, обрання MongoDB підкреслює нашу увагу до гнучкості та пристосованості до різних умов та форматів даних у процесі розробки інформаційної технології для ранжування зображень.

Крім того, MongoDB має вбудовану підтримку горизонтального масштабування, що полегшує розширення бази даних за рахунок додавання нових серверів. Це робить його ефективним рішенням для проєктів з високою навантаженістю та потребою у зберіганні великої кількості даних.

MongoDB також славиться високою швидкістю завдяки використанню документ-орієнтованого підходу, що особливо важливо для операцій, пов'язаних із зображеннями та великими об'ємами даних.

Нарешті, його простий для розробки та інтеграції інтерфейс сприяє швидкому розвитку та адаптації системи до змін. У вигляді цілісного рішення, MongoDB створює ефективне та гнучке середовище для зберігання та обробки даних у процесі ранжування зображень:

- Гнучкість схеми. MongoDB використовує гнучку схему даних, що дозволяє зберігати дані різної структури в одній колекції. Це особливо корисно для проєктів, де структура даних може змінюватися з часом, як це має місце в роботі з ранжуванням зображень, де можуть додаватися нові поля та моделі;
- Горизонтальне масштабування. MongoDB дозволяє легко розширювати базу даних горизонтально, що робить його ідеальним вибором для

проектів з високою навантаженістю, де необхідно розподілене збереження даних;

- Розподілені запити. MongoDB підтримує розподілені запити та може обробляти їх ефективно завдяки своїй архітектурі.

Для створення моделей бази даних та взаємодії з MongoDB в додатку використовується Prisma. Prisma є сучасним ORM (Object-Relational Mapping) та інструментом для моделювання даних, який дозволяє розробникам визначати структуру бази даних та взаємодіяти з нею за допомогою коду на мові програмування.

Prisma впроваджує декларативний підхід до моделювання даних, де структура бази даних описується у вигляді коду. Це дозволяє відмовитися від складних SQL-запитів та зосередитися на розробці додатку, спрощуючи взаємодію з базою даних.

За допомогою Prisma розробники можуть ефективно взаємодіяти з MongoDB, використовуючи інтуїтивний інтерфейс та мову програмування, що полегшує створення та управління моделями даних. Такий підхід сприяє більш ефективній розробці та підтримці додатку, а також забезпечує зручну інтеграцію з MongoDB в рамках проекту.

Основні переваги використання Prisma включають:

- Генерація Prisma Client: Prisma автоматично генерує Prisma Client, який дозволяє взаємодіяти з базою даних за допомогою простого та зрозумілого API;
- Синхронізація з базою даних: Prisma автоматично синхронізує структуру бази даних з описом моделей, що полегшує процес розробки та зберігає консистентність даних;

- Підтримка різних баз даних: Prisma підтримує різні системи управління базами даних, що дозволяє легко змінювати базу даних без необхідності змінювати код додатку.

Використання Prisma в магістерській роботі спрощує розробку та гарантує надійну взаємодію з базою даних MongoDB. Цей інструмент створює зручне та потужне середовище для моделювання даних та їх ефективної взаємодії з інформаційною технологією ранжування зображень. При використанні Prisma розробники можуть легко описувати структуру бази даних та взаємодіяти з нею за допомогою коду, спрощуючи процес створення та управління моделями даних. Такий підхід дозволяє зосередитися на суттєвих аспектах розробки, сприяючи швидкому розвитку та надійній роботі інформаційної технології.

Було розроблено моделі БД на основі коду Prisma. Ці моделі відображають структуру та зв'язки між даними в інформаційній технології ранжування зображень:

- Модель "Account" – представляє дані про обліковий запис користувача. Вона включає інформацію про тип облікового запису, постачальника, токени доступу та інше;
- Модель "Session" – використовується для збереження даних про сесію користувача, включаючи ідентифікатор сесії та дату закінчення;
- Модель "User" – містить дані про користувачів, такі як ім'я, електронну пошту, пароль та інші важливі атрибути. Вона також містить зв'язки з іншими моделями, такими як "Account", "Session" та іншими;
- Модель "VerificationToken" – використовується для збереження токенів підтвердження, які використовуються для підтвердження адреси електронної пошти користувача;
- Модель "Variant" – містить інформацію про варіанти зображень для ранжування, включаючи зображення, назву та зв'язки з іншими моделями;

- Модель "Ranking" – представляє дані про ранжування зображень, включаючи назву, опис та інші важливі атрибути. Вона також містить зв'язки з іншими моделями, такими як "Variant" та "Report";
- Модель "Media" – містить інформацію про медіа-файли, такі як зображення, включаючи опис, URL та розміри;
- Модель "Report" – використовується для збереження звітів користувачів про ранжування зображень. Вона містить інформацію про ідентифікатор ранжування, анонімне ім'я користувача та звіти за раундами.

Ці моделі бази даних дозволяють ефективно зберігати та взаємодіяти з даними, необхідними для інформаційної технології ранжування зображень. Використання MongoDB та створення гнучких моделей даних допомагають забезпечити ефективну роботу та розширюваність системи (рис. 3.4).

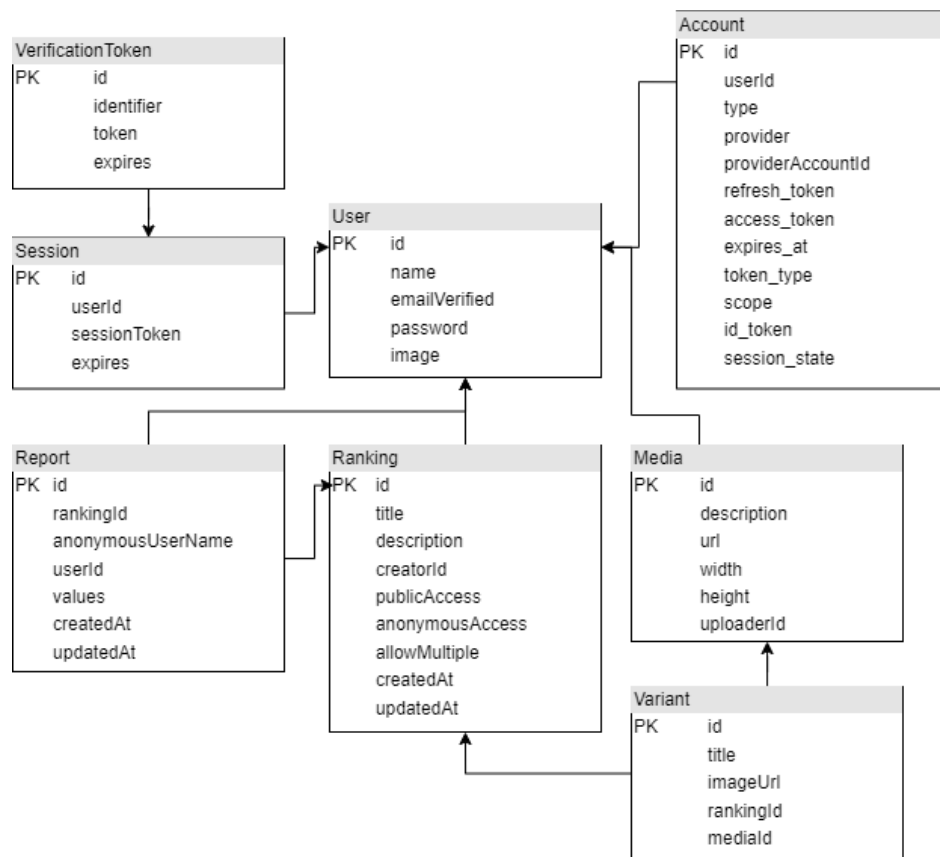


Рисунок 3.4 – Зображення взаємодії моделей БД

3.4 Розробка алгоритму функціонування інформаційної технології ранжування зображень

Розробка алгоритму функціонування в інформаційній технології є критичною для забезпечення її ефективності та точності. Для наочного представлення алгоритму функціонування системи ранжування зображень ми використовуємо два основні підходи: текстовий опис та схему алгоритму. Це дозволяє ясно та зрозуміло визначити кроки та процеси, які визначають результати ранжування.

Сам алгоритм може бути розділений на дві ключові частини: алгоритми створення ранжування та алгоритми його проходження. Алгоритми створення визначають методикку підготовки та оцінювання зображень перед ранжуванням. Вони можуть включати в себе обробку зображень, визначення критеріїв та вагових коефіцієнтів. Алгоритми проходження визначають порядок та спосіб оцінки зображень, враховуючи введені параметри.

Текстовий опис алгоритму дозволяє докладно розглянути кожен крок, надаючи змогу для глибшого розуміння логіки та прийнятих рішень. Схема алгоритму в свою чергу візуалізує структуру та взаємозв'язки між різними етапами, що полегшує сприйняття та аналіз алгоритмічного процесу.

Цей підхід дозволяє нам створити детальний та комплексний алгоритм ранжування зображень, який враховує усі необхідні аспекти для досягнення високої точності та ефективності в нашій інформаційній технології.

Для створення ранжування описані наступні кроки:

- Користувач заходить на головну сторінку веб-додатку;
- Автентифікація користувача за допомогою схеми логін-пароль або Google;
- Перенаправлення користувача на особистий кабінет;
- Перехід на сторінку створення ранжування;

- Введення даних по ранжуванню: опис картинок із посиланнями на картинку, або із завантажуванням файлу, назва, опис та налаштування ранжування;
- Валідація введених даних. Якщо введенні дані по ранжуванню не є валідними, користувач отримає повідомлення з інформацією про поля, які потрібно виправити, аби пройти валідацію;
- Відправка даних на сервер та збереження у базу даних;
- Перенаправлення користувача на сторінку із списком створених ранжувань.

Для проходження ранжування описані наступні кроки (рис. 3.5):

- Користувач заходить на головну сторінку веб-додатку;
- Обирає ранжування, яке в якому він хоче взяти участь та переходить на сторінку ранжування;
- Користувач проходить автоматичну перевірку доступу до ранжування:
 - Якщо ранжування публічне, користувач має змогу продовжити роботу, інакше – користувач отримає повідомлення про те, що ранжування закрито для проходження;



Рисунок 3.5 – Схема алгоритму створення ранжування

- Якщо ранжування закрите від анонімного проходження і користувач не авторизований в системі, користувачу буде запропонована авторизація, інакше – користувач отримає повідомлення про те, що ранжування закрите для анонімного проходження;
- Якщо ранжування закрите від багаторазового проходження і користувач не проходив його раніше, користувач буде мати змогу продовжити роботу, інакше – користувач отримає повідомлення про те, що ранжування закрите для багаторазового проходження.
- Після проходження перевірки, клієнт зможе розпочати роботу з ранжуванням (рис. 3.6), а саме вибір найкращого варіанту серед пар зображень;
- Відправка даних на сервер та збереження у базу даних;
- Після порівняння усіх зображень, користувач отримає аналіз на основі його проходження: загальний рейтинг зображень на основі вибору користувача, результати обрахунку за алгоритмами Сааті, К-шкали та матриці різниць, можливість перейти на колективний аналіз на основі всіх проходжень ранжування або перейти на головну сторінку:
 - Якщо користувач обрав повернутися на головну сторінку, перейти на крок 1;
 - Якщо користувач обрав перейти на колективний аналіз всіх проходжень, його буде перенаправлено на відповідну сторінку;

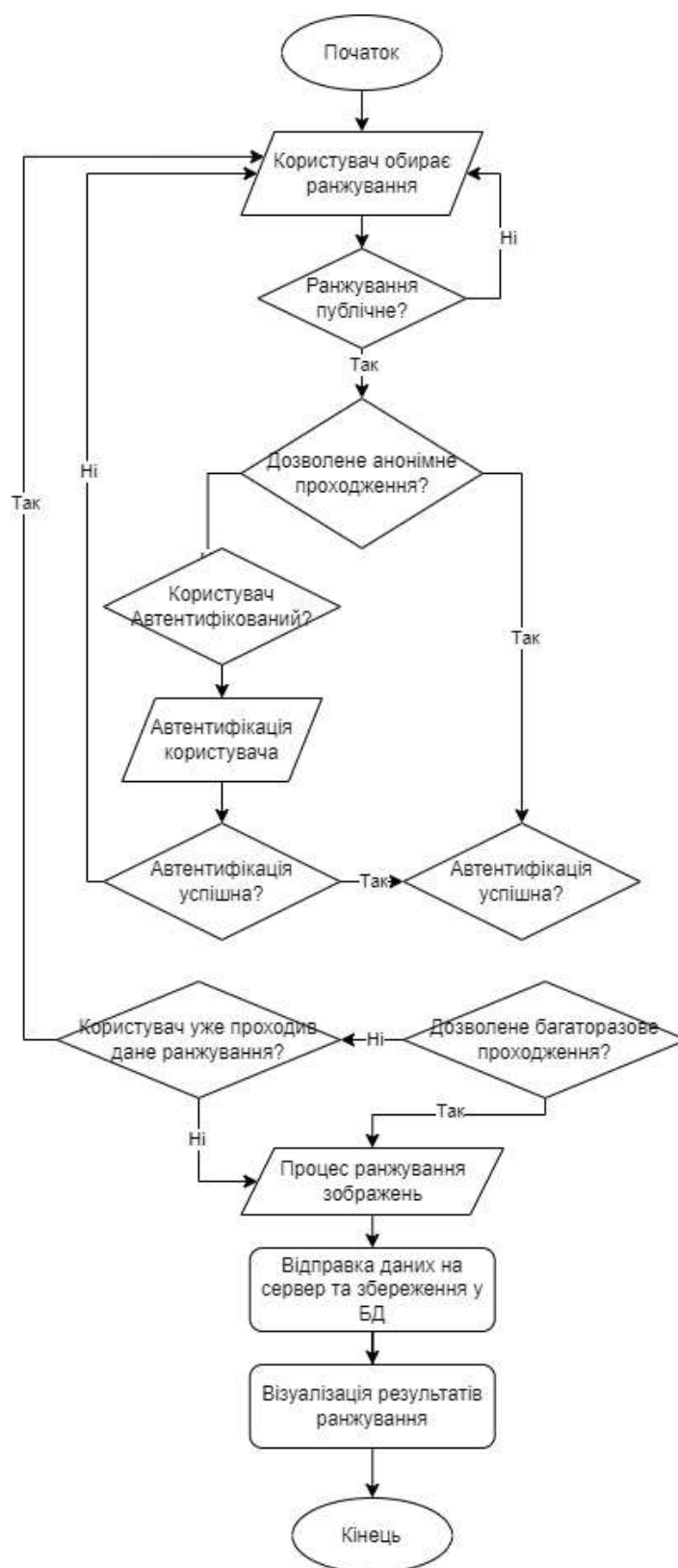


Рисунок 3.6 – Схема алгоритму проходження ранжування

3.5 Тестування програмного забезпечення ранжування зображень та аналіз його результатів

Тестування інформаційної технології ранжування зображень проводилося за двома сценаріями: створення ранжування та проходження існуючого. Такий підхід дозволяє систематично перевірити всі можливі варіанти поведінки користувача в додатку, а також ефективно оцінити його функціональність та коректність в різних ситуаціях.

Створення ранжування для користувача починається із головної сторінки веб-додатку (рис. 3.7).



Рисунок 3.7 – Головна сторінка веб-додатку для ранжування зображень

Далі, користувачу необхідно увійти до системи за допомогою одного із двох доступних методів: вхід або реєстрація через Google або ж використовуючи E-mail та пароль (рис. 3.8, 3.9).

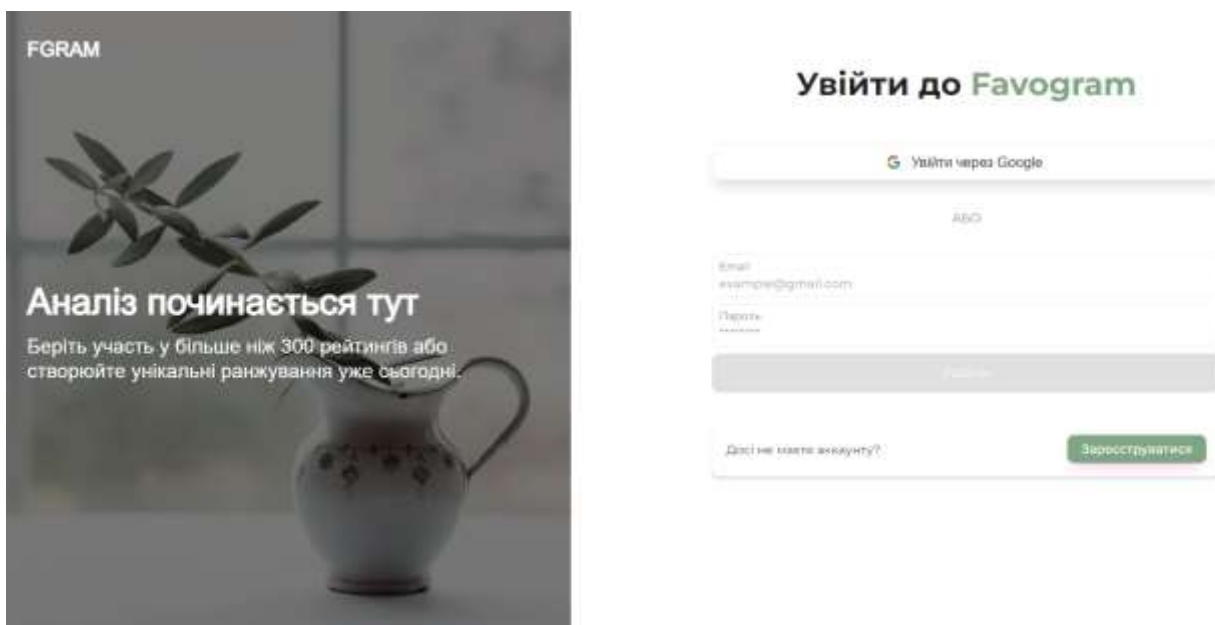


Рисунок 3.8 – Сторінка входу до системи

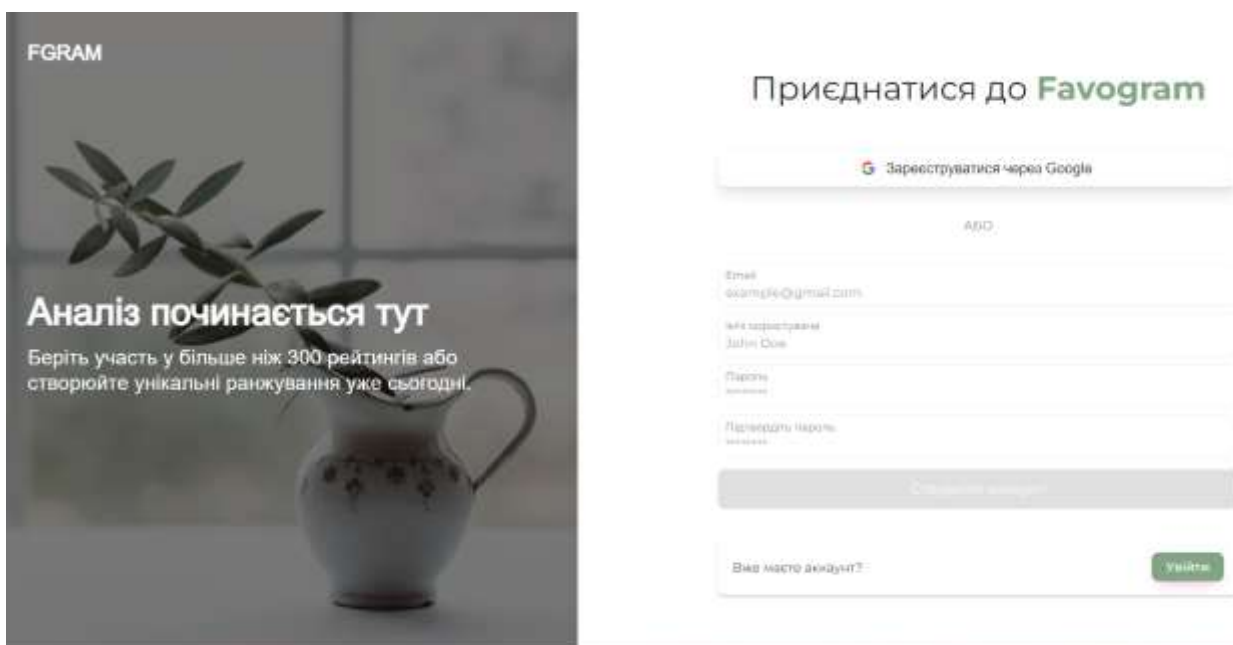


Рисунок 3.9 – Сторінка реєстрації в системі

Після авторизації, користувача буде направлено на сторінку особистого кабінету, де він за натиском на кнопку «Створити ранжування» перейде на сторінку створення. Для того, аби створити ранжування, користувачу необхідно ввести дані згідно з наявними правилами валідації та натиснути «Зберегти» (рис. 3.10).

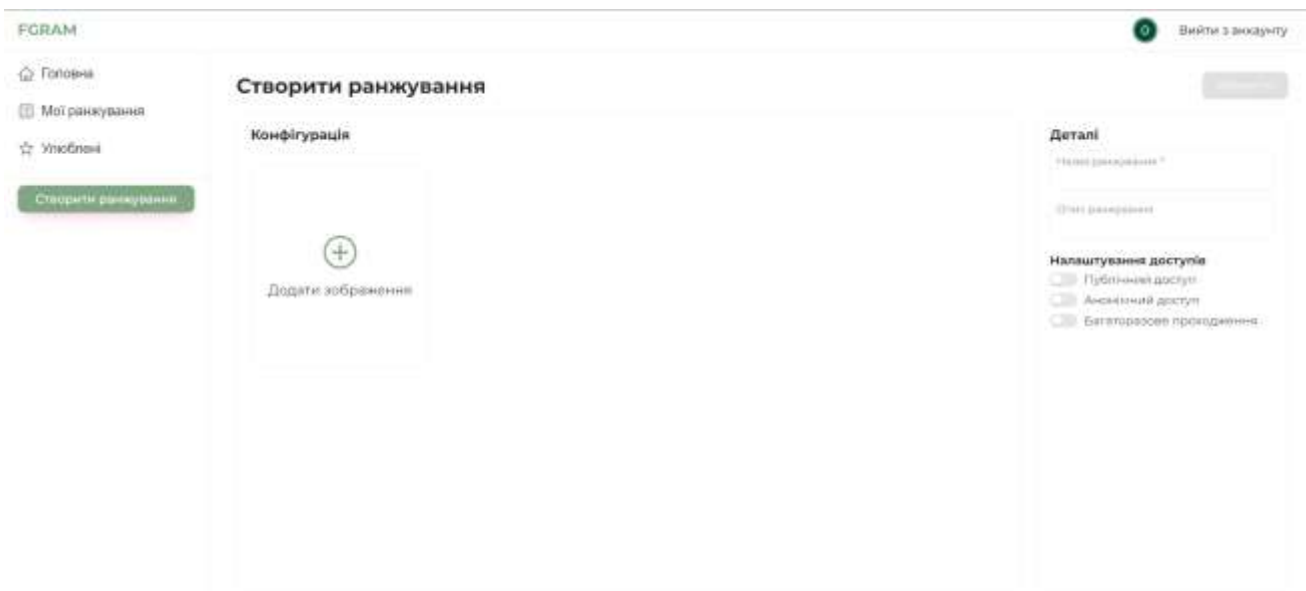


Рисунок 3.10 – Сторінка створення ранжування

Після збереження ранжування у базу даних, користувача буде направлено на сторінку «Мої ранжування», де відображається список створених ранжувань (рис. 3.11).

Назва	Опис	Анонімний доступ	Публічний доступ	Багатораз.	Кількість зображен...	Редагувати
Cool-Poll	—	Ні	Так	Ні	2	Редагувати
Private	Private	Ні	Так	Ні	2	Редагувати
Public Poll, Anonym, Multi	—	Так	Так	Так	2	Редагувати
Public Poll, No Anonym, No Multi	Public Poll	Ні	Так	Ні	2	Редагувати
Alchemist	—	Ні	Так	Ні	2	Редагувати
ACT Banking	—	Так	Так	Так	4	Редагувати
test2	test2	Так	Так	Так	2	Редагувати
main-test	main-test	Ні	Ні	Ні	5	Редагувати
updated333	test234436	Так	Так	Так	2	Редагувати
test	test	Ні	Ні	Ні	2	Редагувати

Рисунок 3.11 – Сторінка «Мої ранжування»

При переході на створене ранжування, користувач може оновити дані ранжування, але окрім зображень, так як ці дані залишаються незмінними і можуть бути видаленими лише із видаленням самого ранжування. Після збереження відредагованого ранжування, користувача буде перенаправлено на сторінку «Мої ранжування», яка зображена на рисунку 3.11.

Сценарій проходження ранжування розпочинається з головної сторінки, аналогічно стартовій точці при створенні ранжування. Головна сторінка додатку для ранжування зображена на рисунку 3.7. На цій сторінці користувач може обрати ранжування для проходження зі списку останніх ранжувань або скористатися можливістю пошуку.

При переході на сторінку ранжування, користувачу необхідно пройти автоматичну валідацію, яка визначає, чи є у користувача змога пройти ранжування на основі параметрів (рис. 3.12 - 3.14).

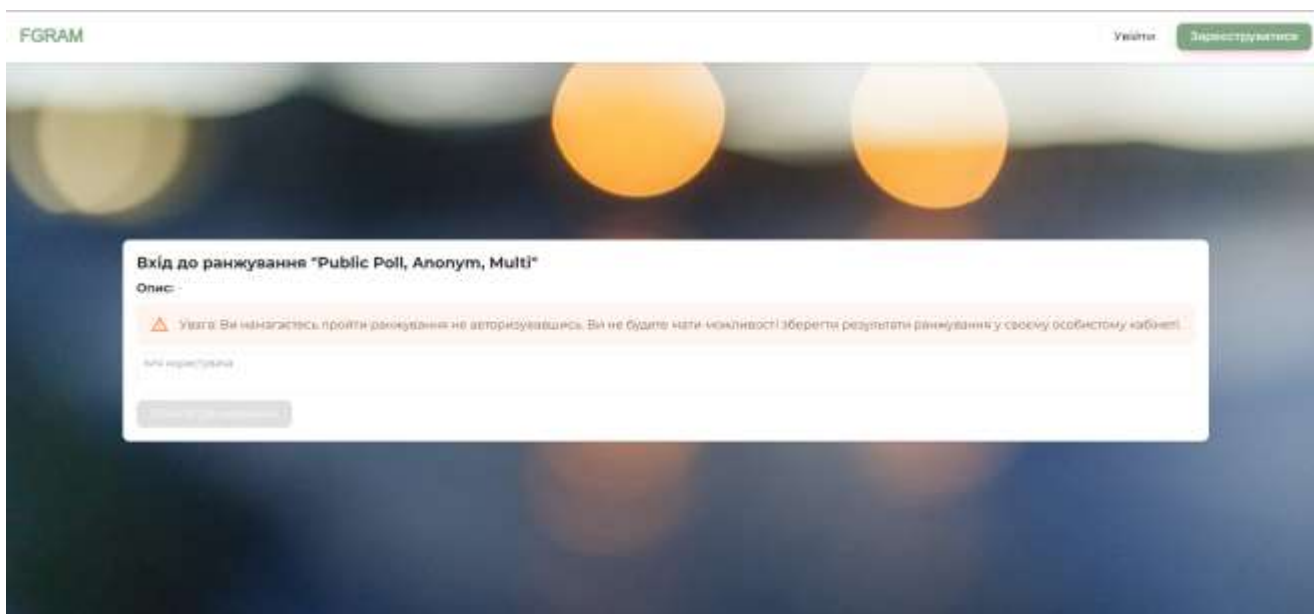


Рисунок 3.12 – Результат валідації анонімного користувача

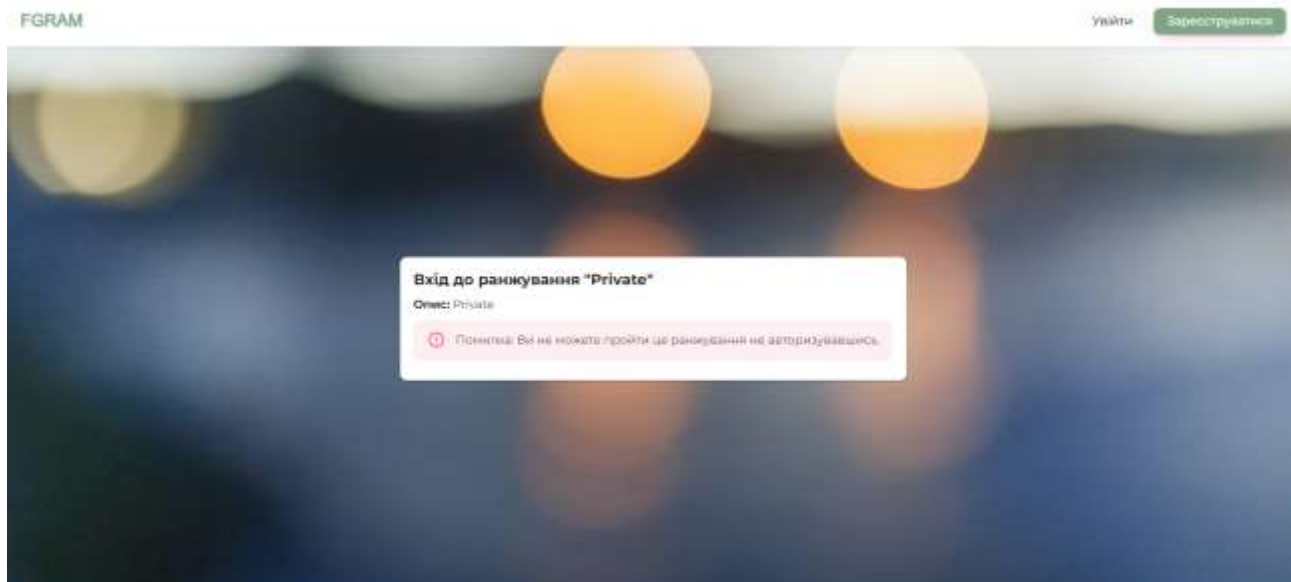


Рисунок 3.13 – Результат валідації анонімного користувача в закритому від анонімного проходження ранжуванні



Рисунок 3.14 – Результат валідації користувача при проходженні свого ранжування

Після валідації, користувач потрапляє на сторінку проходження ранжування, де він обирає зображення, яке йому більше до вподоби за допомогою слайдера. Бали зображенням вираховуються у відсотках (рис. 3.15).

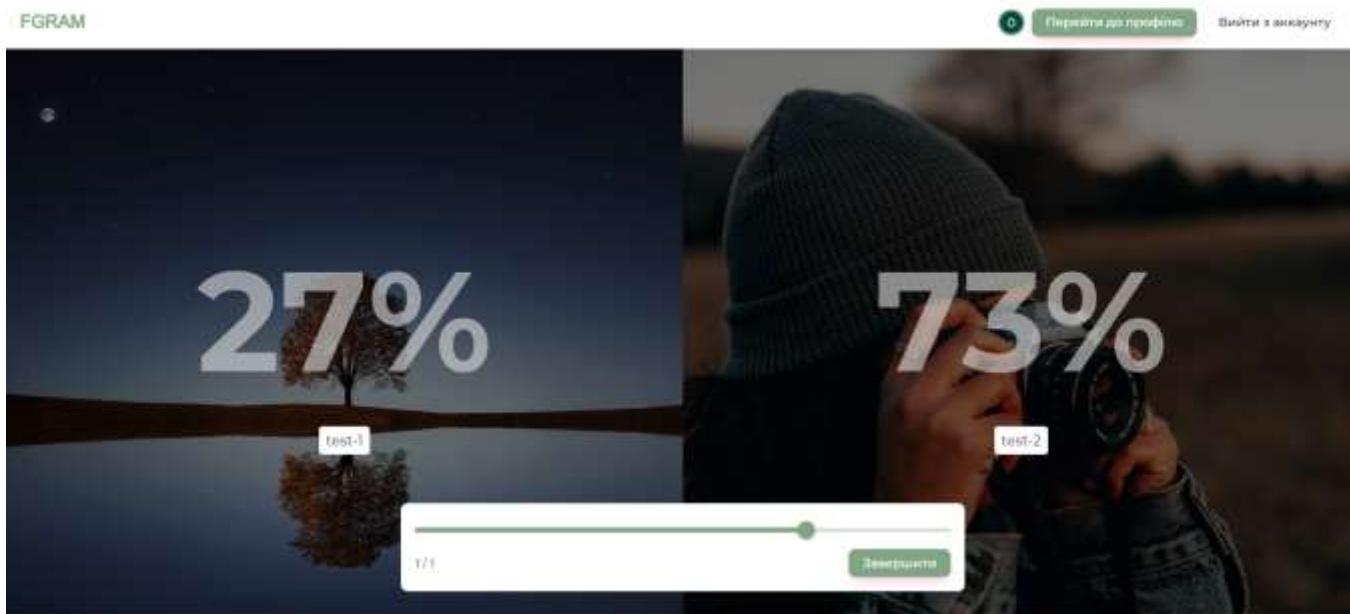


Рисунок 3.15 – Процес ранжування зображень

Коли користувач завершує проходження ранжування, його переводить на візуалізацію результатів ранжування (рис. 3.16).

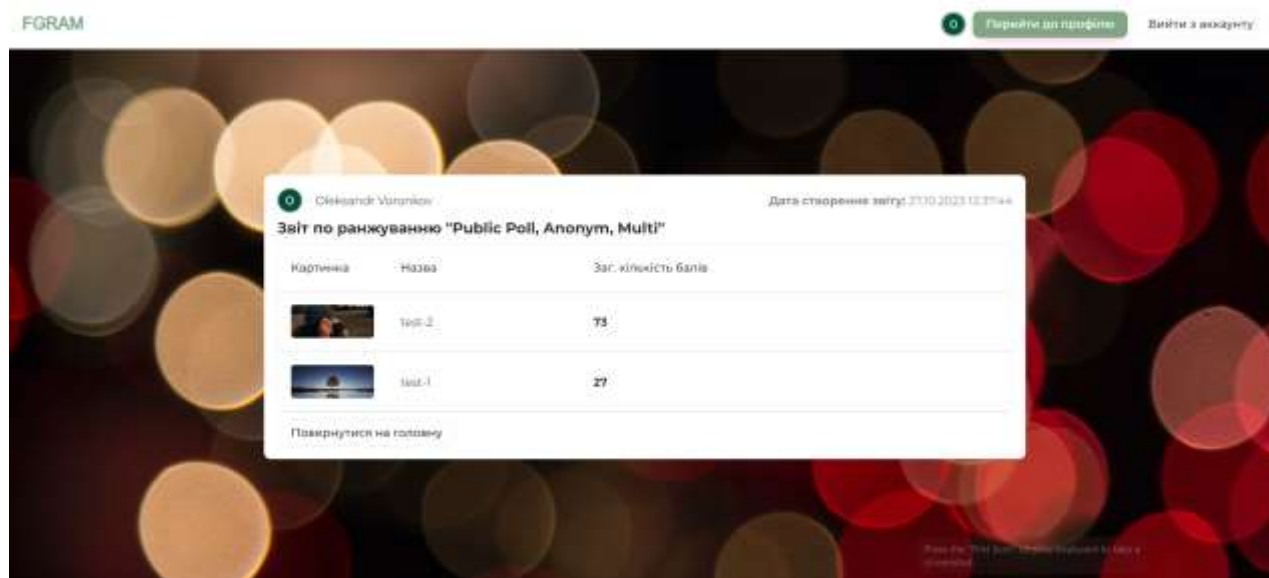


Рисунок 3.16 – Сторінка результату ранжування

Проведемо порівняння функціональних можливостей програм ранжування зображень (табл. 3.1)

Таблиця 3.1 – Результати тестування інформаційної технології ранжування зображень.

	Авторизація клієнта	Індивідуальне ранжування	Колективне ранжування	Ранжування зображень	Наявність шкал ранжування
Створена програма PP SwordS	+	+	+	+	+
TeGP	+	+	-	-+	+
VisPA.	-	+	-	-	+

З таблиці 3.1 видно, що новий програмний продукт володіє 2 новими функціональними можливостями: проведення колективного ранжування зображень та проведення ранжування зображень у вигляді графічних файлів.

3.6 Висновок до розділу 3

Проаналізовано та обрано мову програмування, аргументовано переваги та недоліки обраної мови програмування та її альтернатив. Було проаналізовано різні середовища розробки, аргументовано вибір певного середовища шляхом порівняння із альтернативами. Розроблено алгоритм роботи інформаційної технології.

Також, було проведене тестування інформаційної технології ранжування зображень, описано декілька сценаріїв поведінки користувача у системі. Розроблена система показала стабільну роботу та відсутність критичних помилок.

4 ЕКОНОМІЧНА ЧАСТИНА

Для ефективного впровадження науково-технічної розробки важливо, щоб вона відповідала сучасним вимогам науково-технічного прогресу та враховувала економічні аспекти. Оцінка економічної ефективності результатів науково-дослідної роботи є необхідною складовою цього процесу. Дослідження, представлене в магістерській роботі, яке присвячено розробці та вивченню "Інформаційна технологія ранжування зображень", відноситься до науково-технічних робіт, спрямованих на виведення на ринок. Цей вихід на ринок може бути визначений під час проведення самої роботи і розглядається як комерціалізація науково-технічної розробки. Цей напрямок вважається пріоритетним, оскільки результати можуть бути корисними для різних зацікавлених сторін, приносячи економічні вигоди. Проте для успішного втілення цього процесу важливо знайти зацікавленого інвестора, який був би зацікавлений у реалізації даного проекту, і переконати його в обґрунтованості вкладання інвестицій у цю розробку.

Для цього визначені наступні етапи виконання робіт:

1. Проведено комерційний аудит науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.
2. Розраховані витрати на реалізацію науково-технічної розробки.
3. Проведено розрахунок економічної ефективності науково-технічної розробки у разі її впровадження та комерціалізації потенційним інвестором, а також обґрунтовано економічну доцільність комерціалізації для інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою "Інформаційна технологія ранжування зображень" є розширення функціональних можливостей програмного забезпечення для захисту файлів.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [20].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
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 працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	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

Продовження таблиці 4.1

Практична здійсненість					
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 використовують ся у військово-промисловому секторі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
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

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів: Максименюк М. Р. – інженер, компанія Tango; Філіпенко В. В. – інженер, компанія Tango; Нікітенко Ю. О. – Менеджер, компанія Tango.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПБ, посада)		
	Максименюк М. Р.	Філіпенко В. В.	Нікітенко Ю. О.
	Бали:		
1. Технічна здійсненність концепції	3	2	3
2. Ринкові переваги (наявність аналогів)	4	3	4
3. Ринкові переваги (ціна продукту)	2	2	3
4. Ринкові переваги (технічні властивості)	4	4	4
5. Ринкові переваги (експлуатаційні витрати)	2	2	3
6. Ринкові перспективи (розмір ринку)	1	2	2
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	2	2
9. Практична здійсненність (наявність фінансів)	2	2	2
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	СБ ₁ =37	СБ ₂ =34	СБ ₃ =38
Середньоарифметична сума балів СБ _с	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{34 + 37 + 38}{3} = 36$		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [20].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Інформаційна технологія ранжування зображень" становить 36 балів, що, відповідно до таблиці 4.3 рівень комерційного потенціалу розробки вище середнього, що свідчить про комерційну важливість проведення даних досліджень.

Інформаційну технологію, яка розробляється в магістерській роботі буде реалізовано як веб-додаток. Користуватись може будь хто із доступом до мережі інтернет. Найбільше буде поширене серед студентів, для вирішення колективних проблем вибору якогось варіанту. Також, додаток може використовуватись для суспільних виборів якогось варіанту, наприклад, «Міс Київ 2023».

В якості аналога для розробки було обрано додаток Vispa. VisPA (Візуальні попарні порівняння) — мобільний додаток для операційної системи Android, що дозволяє візуально порівнювати альтернативи за допомогою методу попарного порівняння. Результати представлені у вигляді графіків із кардинально центрованими значеннями. Основними недоліками аналога є: неможливість проводити ранжування зображеннями, тільки текст; незручний користувацький

інтерфейс; необхідність встановлення додатку на девайс для його використання; неможливість поділитися результатами ранжування з іншими користувачами

У розробці дана проблема вирішується шляхом заміни текстового відображення даних на зображення. Також, система доступна за посиланням в інтернеті, не потребує завантажування, а всі результати ранжування можуть бути переглянуті іншими користувачами за певних налаштувань.

4.2 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему "Інформаційна технологія ранжування зображень", під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [20]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$Z_o = 16000 \cdot 5 / 21 = 3636$ грн.

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	16000	727,3	5	3636
Інженер-програміст	14000	636,4	35	22273
Тестувальник	12000	545,5	20	10909
Всього				36818

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.2)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (36818) \cdot 11 / 100\% = 4050 \text{ грн.}$$

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{доо}}) \cdot \frac{H_{zn}}{100\%}, \quad (4.3)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (36818 + 4050) \cdot 22 / 100\% = 8991 \text{ грн.}$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою "Інформаційна технологія ранжування зображень".

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\epsilon j}, \quad (4.4)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

$C_{\epsilon j}$ – вартість відходів j -го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір А4	190	1	190
Ручка	15	1	15
Диск оптичний CD	14	2	28
Flesh-пам'ять 64	300	1	300
Всього			533
З врахуванням коефіцієнта транспортування			586,3

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему "Інформаційна технологія ранжування зображень" відсутні.

До статті «Специфікування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання специфікування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення в роботі відсутні.

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення. Програмні засоби, які були використанні при написанні магістерської роботи є безкоштовними.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б.}}{T_e} \cdot \frac{t_{вик}}{12}, \quad (4.5)$$

де $Ц_{б.}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

T_e – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (45000 \cdot 1) / (2 \cdot 12) = 1875 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук	45000	2	1	1875,00
Принтер	4500	2	1	187,50
Приміщення лабораторії	230000	20	1	958,33
Всього				3020,83

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{ени}}{\eta_i}, \quad (4.6)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,5$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 245,0 \cdot 7,5 \cdot 0,5 / 0,8 = 287,11 \text{ грн.}$$

До статті «Службові відрядження» дослідної роботи на тему "Інформаційна технологія ранжування зображень" належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

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

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (36818) \cdot 20 / 100\% = 7363,64 \text{ грн.}$$

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

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» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (4.8)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати», прийmemo $H_{\text{ів}} = 50\%$.

$$I_{\text{в}} = (36818) \cdot 50 / 100\% = 18409,09 \text{ грн.}$$

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» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.9)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{\text{нзв}} = 100\%$.

$$B_{\text{нзв}} = (36818) \cdot 100 / 100\% = 36818,18 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему "Інформаційна технологія ранжування зображень". розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{одд}} + Z_n + M + K_v + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сн}} + I_v + B_{\text{нзв}}. \quad (4.10)$$

$$B_{\text{заг}} = 36818 + 40508991 + 586,3 + 3020,83 + 287,11 + 7363,64 + 18409,09 + 36818,18 \\ = 116344,33 \text{ грн.}$$

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

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (4.11)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,9$.

$$ZB = 116344,33 / 0,9 = 129271,48 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим 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.

Результати дослідження проведені за темою "Інформаційна технологія ранжування зображень" передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1 особа

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 2100,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на 500,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [20]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{g}{100}\right), \quad (4.12)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 40\%$;

ρ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\rho = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (1 \cdot 500 + 2100 \cdot 600) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 266574,75 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (1 \cdot 500 + 2100 \cdot (600 + 500)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 489063,79 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (1 \cdot 500 + 2100 \cdot (600 + 500 + 400)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 666723,35 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 18\%$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} \Pi\Pi &= 266574,75 / (1 + 0,18)^1 + 489063,79 / (1 + 0,18)^2 + 666723,35 / (1 + 0,18)^3 = \\ &= 949403,11 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (4.14)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 129271,48 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 129271,48 = 258542,96 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (4.15)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 949403,11 грн;

PV – теперішня вартість початкових інвестицій 179277,79 грн.

$$E_{абс} = ПП - PV = 949403,11 - 258542,96 = 690860,15 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = T_{ж} \sqrt{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.16)$$

де E_{abc} – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_e = T_{ж} \sqrt[3]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 690860,15 / 258542,96)^{1/3} - 1 = 0,85.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (4.17)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{min} = 0,1 + 0,25 = 0,35 < 0,85$ свідчить про те, що внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою "Інформаційна технологія ранжування зображень" доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_e}, \quad (4.18)$$

де E_g – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,85 = 1,2 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.4 Висновки до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Інформаційна технологія ранжування зображень" становить 36 балів, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

Також термін окупності становить 1,2 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою "Інформаційна технологія ранжування зображень".

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було вирішено завдання щодо розробки інформаційної технології ранжування зображень. Проаналізовано існуючі рішення та аналоги інформаційної технології у цьому контексті. Вибір методу вирішення було обґрунтовано. Програмно реалізовано інформаційну технологію ранжування зображень. Проведено тестування розробленого програмного забезпечення та проаналізовано його результати.

Дослідження було успішним завдяки розробці веб-застосунка за використанням запропонованого шаблону. Це дозволило вирішити проблеми гнучкості та монетизації, які існують у вже існуючих аналогів розробленої інформаційної технології, при цьому не втрачаючи в швидкості розробки і швидкодії програми.

У процесі програмної реалізації інформаційної технології ранжування зображень було обґрунтовано вибір мови програмування та інших технологій розробки. Створено та описано схему алгоритму роботи програми, подано тестові приклади та проаналізовано результати роботи як серверу, так і клієнту, демонструючи їх здатність до швидкої відповіді користувачу.

Розроблену інформаційну технологію можна покращувати шляхом додавання нових алгоритмів аналізу ранжування та покращення відгуку на взаємодію з користувачем. При цьому використано мову програмування TypeScript, фреймворк Next.JS, середовище розробки Webstorm, документно-орієнтована база даних MongoDB та клієнт для управління базою даних Mongo Atlas.

Під час аналізу результатів роботи розробленого програмного забезпечення доведено, що новий програмний продукт володіє 2 новими функціональними можливостями: проведення колективного ранжування зображень та проведення ранжування зображень у вигляді графічних файлів.

Мету роботи досягнуто. Поставлені задачі виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В.В. Колодний, О.І. Воронков. Інформаційна технологія ранжування зображень. Матеріали Всеукраїнської науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2023)» – [Електронний ресурс]. – <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/18945>
2. Канеман Д. Мислення швидко й повільно / Д. Канеман. – Київ: «Наш Формат», 2022. – 480 с.
3. Методи обробки експертної інформації URL: https://uk.wikipedia.org/wiki/Методи_обробки_експертної_інформації.
4. Застосування гештальт-ранжувань для виявлення переваг ОПР / В. В. Колодний, В. В. Зубко // «ІНТЕРНЕТ-ОСВІТА-НАУКА-2016»: Збірник матеріалів конференції. – Вінниця : ВНТУ, 2016. – С. 43-44.
5. Інформаційна технологія візуального моделювання та оброки тернарних гештальт-ранжувань / В. В. Колодний, Д. С. Кудрявцев // “ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП’ЮТЕРНА ІНЖЕНЕРІЯ”: Том 42 №2, 2018. – С. 26-34.
6. Метод некритеріального структурування множини альтернатив за допомогою аналізу тернарних тривірневих ран-жувань / В. В. Колодний, В. В. Зубко // «ІНТЕРНЕТ-ОСВІТА-НАУКА-2014»: Збірник матеріалів конференції. – Вінниця : ВНТУ, 2014. – С. 13-14.
7. Unified Modeling Language [Електронний ресурс]: Режим доступу: https://uk.wikipedia.org/wiki/Unified_Modeling_Language.
8. JavaScript [Електронний ресурс]: Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>.
9. Python [Електронний ресурс]: Режим доступу: <https://uk.wikipedia.org/wiki/Python>.
10. Ruby [Електронний ресурс]: Режим доступу: <https://uk.wikipedia.org/wiki/Ruby>.

11. Documentation for JS developers [Електронний ресурс]: Режим доступу: <https://developer.mozilla.org/en-US/>.
12. David Herman: Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript (Effective Software Development Series) / Herman D.: Addison-Wesley Professional, 2012 – 124 с.
13. Crockford D. JavaScript: The Good Parts: The Good Parts / Crockford D.: O'Reilly Media, 2008.– 140 с.
14. Solutions M. Advantages and Disadvantages of PHP Frameworks [Електронний ресурс] / Mindfire Solutions. – 2018. – Режим доступу до ресурсу: <https://medium.com/mindfiresolutions.usa/advantages-and-disadvantages-of-php-frameworks-c046d50754e5>.
15. TechTarget Contributor. What is Ruby? - Definition from WhatIs.com [Електронний ресурс] / TechTarget Contributor. – 2019. – Режим доступу до ресурсу: <https://whatis.techtarget.com/definition/Ruby>.
16. Nick Gorbikoff. What are the advantages and disadvantages of Ruby? [Електронний ресурс] / Nick Gorbikoff. – 2018. – Режим доступу до ресурсу: <https://www.quora.com/What-are-the-advantages-and-disadvantages-of-Ruby>.
17. What is JavaScript? JavaScript - Overview. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/javascript/javascript-overview.htm>.
18. Колодний В. В. Мобільний застосунок візуального порівняння альтернатив VisPA / В. В. Колодний, П. О. Зоря // “Інформаційні технології та комп’ютерна інженерія”. Вінниця: ВНТУ, 2019. - С.3
19. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.
20. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2016. – 113 с.

ДОДАТКИ

Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія ранжування зображень

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФШТА
(кафедра, факультет)

Показники звіту подібності Unischek

Оригінальність 94,95% Схожість 5,03%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unischek щодо роботи.

Автор роботи



Воронков О.І.

Керівник роботи



Колодний В.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

import Link from "next/link";
import { AuthLoginForm } from "@components/auth-login-form";
import { AuthGoogleButton } from "@components/auth-google-button";
import { Box, Button, Typography } from "@mui/material";
import {
  AuthStyledContainer,
  AuthStyledDividerText,
  AuthStyledHeader,
  AuthStyledPaper,
} from "@app/(auth)/auth.styles";

export default async function LoginPage() {
  return (
    <AuthStyledContainer>
      <AuthStyledHeader variant="h1">
        У в і й т и д о <span>Favogram</span>
      </AuthStyledHeader>

      <Box width="100%" mt={6}>
        <AuthGoogleButton>У в і й т и ч е р е з Google</AuthGoogleButton>

        <AuthStyledDividerText variant="body2" color="secondary">
          А б о
        </AuthStyledDividerText>

        <AuthLoginForm />

        <AuthStyledPaper elevation={1} sx={{ mt: 4 }}>
          <Typography variant="body2">Д о с і н е м а є т е а к к а у н т у ?</Typography>

          <Link href="/register">
            <Button variant="contained" color="primary">
              З а р е є с т р у в а т и с я
            </Button>
          </Link>
        </AuthStyledPaper>
      </Box>
    </AuthStyledContainer>
  );
}

import { AuthGoogleButton } from "@components/auth-google-button";
import { AuthRegisterForm } from "@components/auth-register-form";
import Link from "next/link";
import {
  AuthStyledContainer,

```



```

    AuthStyledDividerText,
    AuthStyledHeader,
    AuthStyledPaper,
  } from "@app/(auth)/auth.styles";
import { Box, Button, Typography } from "@mui/material";

export default function RegisterPage() {
  return (
    <AuthStyledContainer>
      <AuthStyledHeader>
        Приєднатися до <span className="text-primary font-bold">Favogram</span>
      </AuthStyledHeader>

      <Box width="100%" mt={6}>
        <AuthGoogleButton>Зареєструватися через Google</AuthGoogleButton>

        <AuthStyledDividerText variant="body2" color="secondary">
          Або
        </AuthStyledDividerText>

        <AuthRegisterForm />

        <AuthStyledPaper elevation={1} sx={{ mt: 4 }}>
          <Typography variant="body2">Вже маєте акаунт?</Typography>

          <Link href="/login">
            <Button variant="contained" color="primary">
              Увійти
            </Button>
          </Link>
        </AuthStyledPaper>
      </Box>
    </AuthStyledContainer>
  );
}

"use client";

import { Paper, styled, Typography } from "@mui/material";
import { grey } from "@mui/material/colors";

export const AuthStyledContainer = styled("div")(() => ({
  display: "flex",
  flexDirection: "column",
  alignItems: "center",
  maxWidth: 576,
  margin: "0 auto",
})));

export const AuthStyledHeader = styled(Typography)(({ theme }) => ({

```

```

fontSize: 36,
lineHeight: "40px",

"& span": {
  color: theme.palette.primary.main,
},
}));

export const AuthStyledPaper = styled(Paper)(({ theme }) => ({
  display: "flex",
  alignItems: "center",
  justifyContent: "space-between",

  borderColor: grey[200],
  padding: theme.spacing(2),
}));

export const AuthStyledDividerText = styled(Typography)(({ theme }) => ({
  textAlign: "center",
  textTransform: "uppercase",
  margin: theme.spacing(4, 0),
}));

import Link from "next/link"
import { PropsWithChildren } from "react"
import LoginBackground from "@public/images/login-background.jpg"
import Image from "next/image"

export default function AuthLayout({ children }: PropsWithChildren) {
  return <div className="min-h-screen flex">
    <div className="relative flex flex-col justify-between py-8 px-6 w-5/12">
      <Link href="/" className="logo text-xl z-20">
        fgram
      </Link>

      <div className="z-20">
        <h2 className="text-white text-4xl font-bold mb-4">Аналіз починається тут</h2>

        <p className="text-white text-xl">Беріть участь у більше ніж 300 рейтингів або
        створюйте унікальні ранжування уже сьогодні.</p>
      </div>

      <div></div>

      <Image src={LoginBackground} alt="Home Background" layout="fill" objectFit="cover" objectPosition="center"
      className="z-0" />

      <div className="overlay" />
    </div>
  </div>
}

```

```

    <div className="flex-1 py-16 overflow-auto">
      {children}
    </div>
  </div>
}

import { DashboardEditorForm } from "@components/dashboard/dashboard-editor-form";
import { getSingleParams } from "@lib/utills";

interface DashboardEditorPageProps {
  params: {
    id?: [string];
  };
}

export default async function DashboardEditorPage({
  params,
}: DashboardEditorPageProps) {
  const { id } = getSingleParams(params);

  return <DashboardEditorForm rankingId={id} />;
}

export default function DashboardFavoritesPage() {
  return <</>;
}

import { Fragment } from "react";
import { Typography } from "@mui/material";
import { DashboardRankingsTable } from "@components/dashboard/dashboard-rankings-table";

export const metadata = {
  title: "Favogram - Мої р а н ж у в а н н я",
};

export default async function DashboardRankingsPage() {
  return (
    <Fragment>
      <Typography variant="h1">Мої р а н ж у в а н н я</Typography>

      <DashboardRankingsTable />
    </Fragment>
  );
}

import { redirect } from "next/navigation";

export default function DashboardPage() {
  redirect("/dashboard/rankings");
}

```

```

    return null;
  }

import Link from "next/link";
import { PropsWithChildren } from "react";
import { DashboardNavbar } from "@components/dashboard/dashboard-navbar";
import { getCurrentUser } from "@lib/session";
import { DashboardDrawer } from "@components/dashboard/dashboard-drawer";

export default async function DashboardLayout({ children }: PropsWithChildren) {
  const user = await getCurrentUser();

  return (
    <div className="min-h-screen flex flex-col">
      <DashboardNavbar user={user} />

      <div className="flex flex-1 items-stretch">
        <DashboardDrawer />

        <div className="flex flex-col flex-1 overflow-auto relative py-6 px-8">
          {children}
        </div>
      </div>
    </div>
  );
}

import { getAuthOptions } from '@lib/auth';
import NextAuth from 'next-auth'

const handler = NextAuth(getAuthOptions());

export { handler as GET, handler as POST };

import { registerUserSchema } from "@schemas/auth-schemas";
import { NextResponse } from "next/server";
import { BAD_REQUEST } from "http-status";
import { db } from "@lib/db";
import bcrypt from "bcrypt";

export async function POST(request: Request) {
  const requestData = await request.json();

  try {
    const validatedData = await registerUserSchema.parseAsync(requestData);

    const dbUser = await db.user.findFirst({
      where: {
        email: validatedData.email,
      },
    },

```

```

    });

    if (dbUser) {
      return NextResponse.json(
        {
          message: "USER_ALREADY_EXISTS",
        },
        { status: BAD_REQUEST }
      );
    }

    const { confirmPassword, ...userData } = validatedData;

    const hashedPassword = await bcrypt.hash(validatedData.password, 10);

    const user = await db.user.create({
      data: {
        ...userData,
        password: hashedPassword,
      },
    });

    return NextResponse.json({ user });
  } catch (error) {
    return NextResponse.json(
      {
        message: "VALIDATION_ERROR",
      },
      { status: BAD_REQUEST }
    );
  }
}

import { NextResponse } from "next/server";
import { BAD_REQUEST, NOT_FOUND, UNAUTHORIZED } from "http-status";
import { db } from "@lib/db";
import { createRankingSchema } from "@schemas/dashboard-schemas";
import { getCurrentUser } from "@lib/session";

export async function GET(request: Request, route: { params: { id: string } }) {
  const user = await getCurrentUser();

  if (!user) {
    return NextResponse.json(
      {
        message: "NO_USER_HEADER",
      },
      { status: UNAUTHORIZED }
    );
  }
}

```

```

const ranking = await db.ranking.findFirst({
  where: {
    id: route.params.id,
    creatorId: user.id,
  },
  include: {
    variants: true,
  },
});

if (!ranking) {
  return NextResponse.json(
    {
      message: "NO_RANKING_FOUND",
    },
    { status: NOT_FOUND }
  );
}

return NextResponse.json({ ranking });
}

export async function PUT(request: Request, route: { params: { id: string } }) {
  const user = await getCurrentUser();

  if (!user) {
    return NextResponse.json(
      {
        message: "NO_USER_HEADER",
      },
      { status: UNAUTHORIZED }
    );
  }

  const requestData = await request.json();

  try {
    const { variants, ...validatedData } =
      await createRankingSchema.parseAsync(requestData);

    const dbRanking = await db.ranking.findFirst({
      where: {
        id: route.params.id,
        creatorId: user.id,
      },
    });

    if (!dbRanking) {
      return NextResponse.json(

```

```

    {
      message: "NO_RANKING_FOUND",
    },
    { status: NOT_FOUND }
  );
}

const updatedRanking = await db.ranking.update({
  where: {
    id: dbRanking.id,
  },
  data: {
    title: validatedData.title,
    description: validatedData.description,
    publicAccess: validatedData.publicAccess,
    allowMultiple: validatedData.allowMultiple,
    anonymousAccess: validatedData.anonymousAccess,
  },
});

return NextResponse.json({
  ranking: updatedRanking,
});
} catch (error) {
  return NextResponse.json(
    {
      message: "VALIDATION_ERROR",
    },
    { status: BAD_REQUEST }
  );
}
}

export async function DELETE(
  request: Request,
  route: { params: { id: string } }
) {
  const user = await getCurrentUser();

  if (!user) {
    return NextResponse.json(
      {
        message: "NO_USER_HEADER",
      },
      { status: UNAUTHORIZED }
    );
  }

  try {
    await db.ranking.delete({

```

```

    where: {
      id: route.params.id,
    },
  });

  return NextResponse.json(true);
} catch (error) {
  return NextResponse.json(
    {
      message: "VALIDATION_ERROR",
    },
    { status: BAD_REQUEST }
  );
}
}

import { createRankingSchema } from "@/schemas/dashboard-schemas";
import { db } from "@/lib/db";
import { NextResponse } from "next/server";
import { BAD_REQUEST, UNAUTHORIZED } from "http-status";
import { getCurrentUser } from "@/lib/session";

export async function GET(request: Request) {
  const user = await getCurrentUser();

  if (!user) {
    return NextResponse.json(
      {
        message: "NO_USER_HEADER",
      },
      { status: UNAUTHORIZED }
    );
  }

  const rankings = await db.ranking.findMany({
    where: {
      creatorId: user.id,
    },
    include: {
      variants: true,
    },
    orderBy: [{ updatedAt: "desc" }],
  });

  return NextResponse.json({ rankings });
}

export async function POST(request: Request) {
  const user = await getCurrentUser();

```



```

if (!user) {
  return NextResponse.json(
    {
      message: "NO_USER_HEADER",
    },
    { status: UNAUTHORIZED }
  );
}

const requestData = await request.json();

try {
  const { variants, ...validatedData } =
    await createRankingSchema.parseAsync(requestData);

  const createdRanking = await db.ranking.create({
    data: {
      title: validatedData.title,
      description: validatedData.description,
      publicAccess: validatedData.publicAccess,
      allowMultiple: validatedData.allowMultiple,
      anonymousAccess: validatedData.anonymousAccess,
      creatorId: user.id,
    },
  });

  const createdVariants = await db.variant.createMany({
    data: variants.map((variant) => ({
      ...variant,
      rankingId: createdRanking.id,
    })),
  });

  return NextResponse.json({
    ranking: createdRanking,
    variants: createdVariants,
  });
} catch (error) {
  return NextResponse.json(
    {
      message: "VALIDATION_ERROR",
    },
    { status: BAD_REQUEST }
  );
}

import { getCurrentUser } from "@lib/session";
import { NextResponse } from "next/server";
import { BAD_REQUEST, UNAUTHORIZED } from "http-status";

```

```
import { db } from "@lib/db";
import { uploadFile } from "@lib/media-utils";

export async function POST(request: Request) {
  const user = await getCurrentUser();

  if (!user) {
    return NextResponse.json(
      {
        message: "NOT_AUTHORIZED",
      },
      { status: UNAUTHORIZED }
    );
  }

  const dto = await request.formData();

  const file = dto.get("file") as File;

  try {
    const uploadedFile = await uploadFile(file);

    const media = await db.media.create({
      data: {
        uploaderId: user.id,
        description: "",
        url: uploadedFile.secure_url,
        width: uploadedFile.width,
        height: uploadedFile.height,
      },
    });

    return NextResponse.json({ media });
  } catch (error) {
    console.log(error);

    return NextResponse.json(
      {
        message: "UPLOAD_ERROR",
      },
      { status: BAD_REQUEST }
    );
  }
}
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РАНЖУВАННЯ ЗОБРАЖЕНЬ

Виконав: студент 2 курсу, групи 2КН-22м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)



Воронков О.І.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН



Колодний В.В.

(прізвище та ініціали)

« 7 » 12 2023 р.



Рисунок В.1 - Структура інформаційної технології візуального ранжування альтернатив

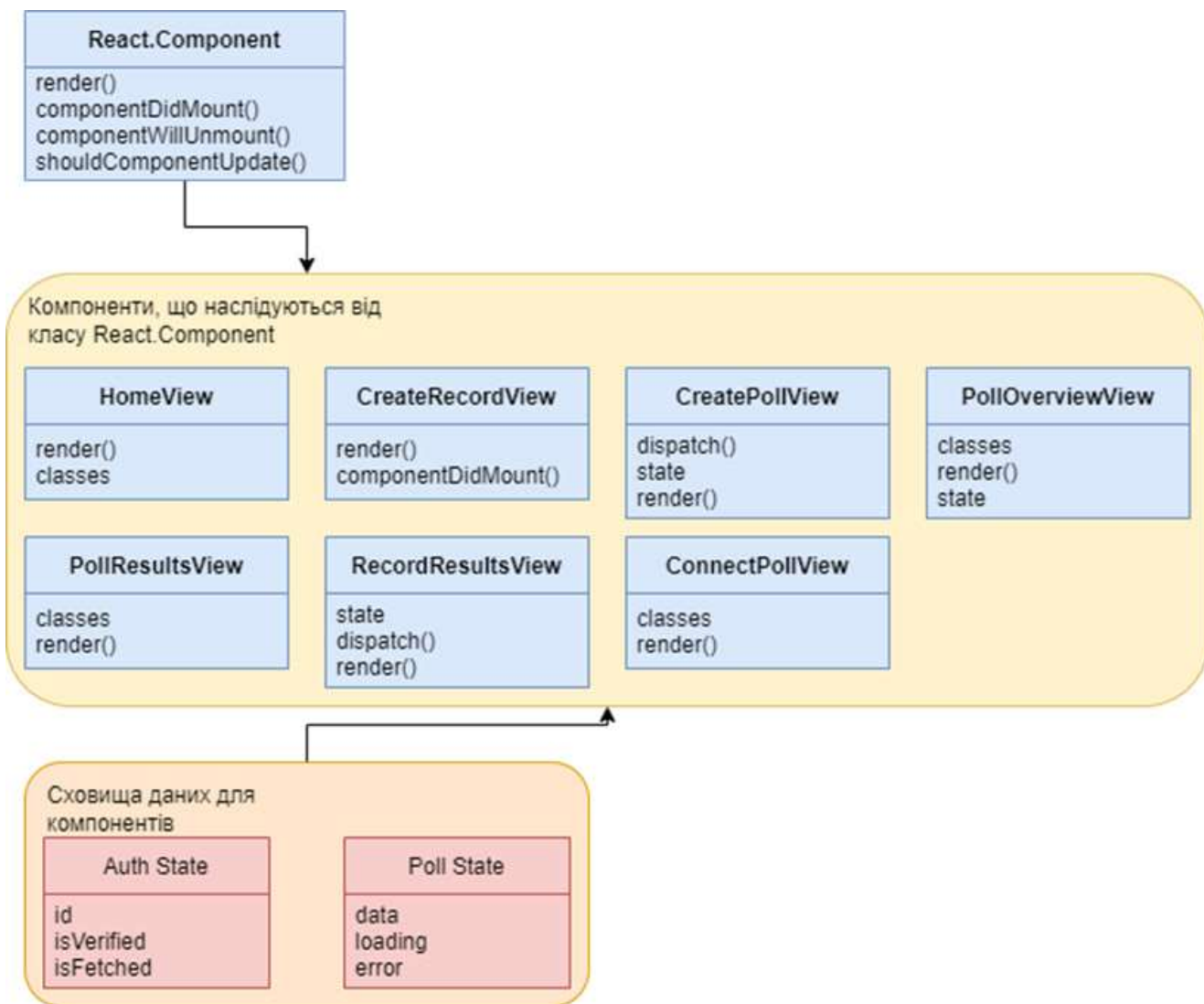


Рисунок В.2 - Діаграма класів інформаційної технології візуального ранжування картинок

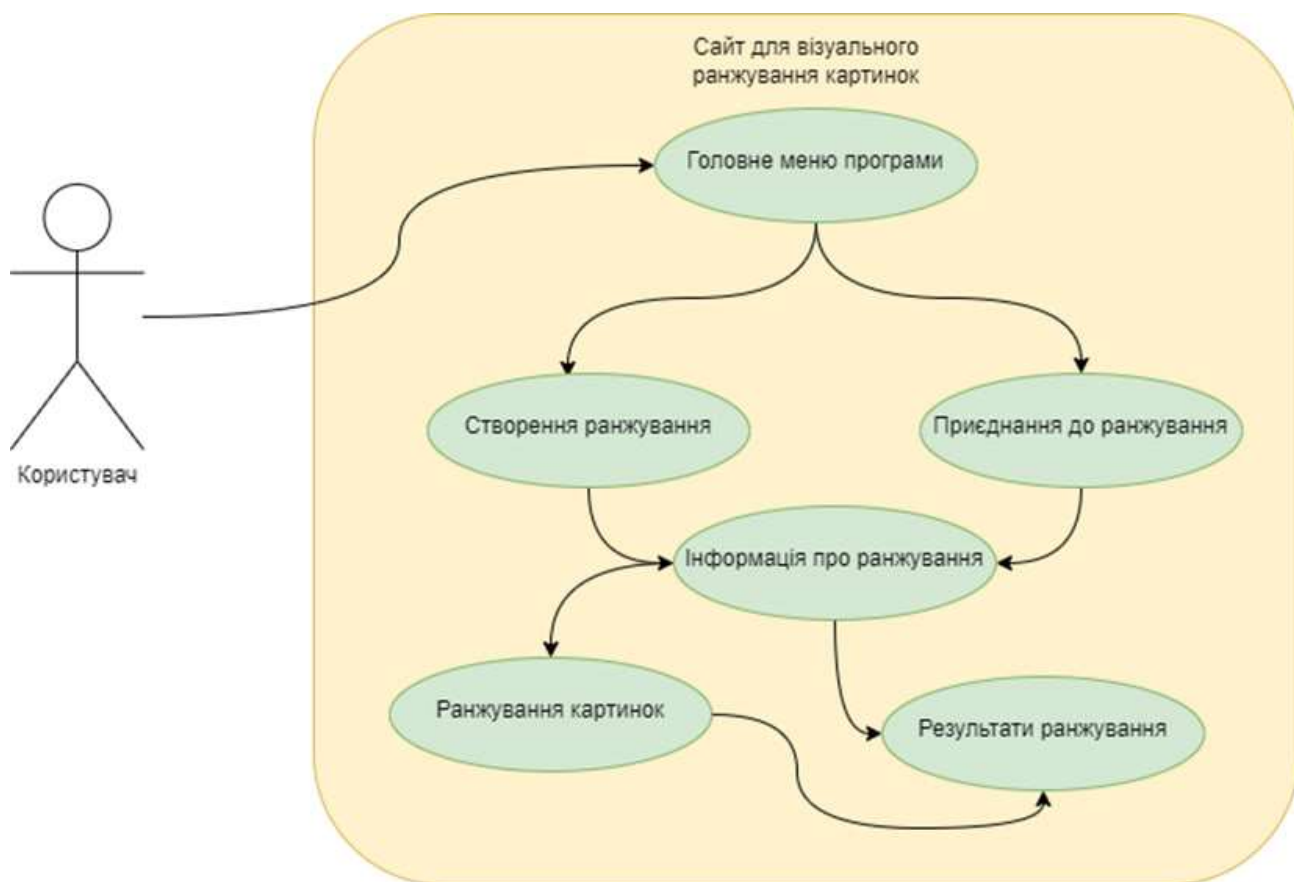


Рисунок В.3 - Діаграма прецедентів інформаційної технології візуального ранжування картинок

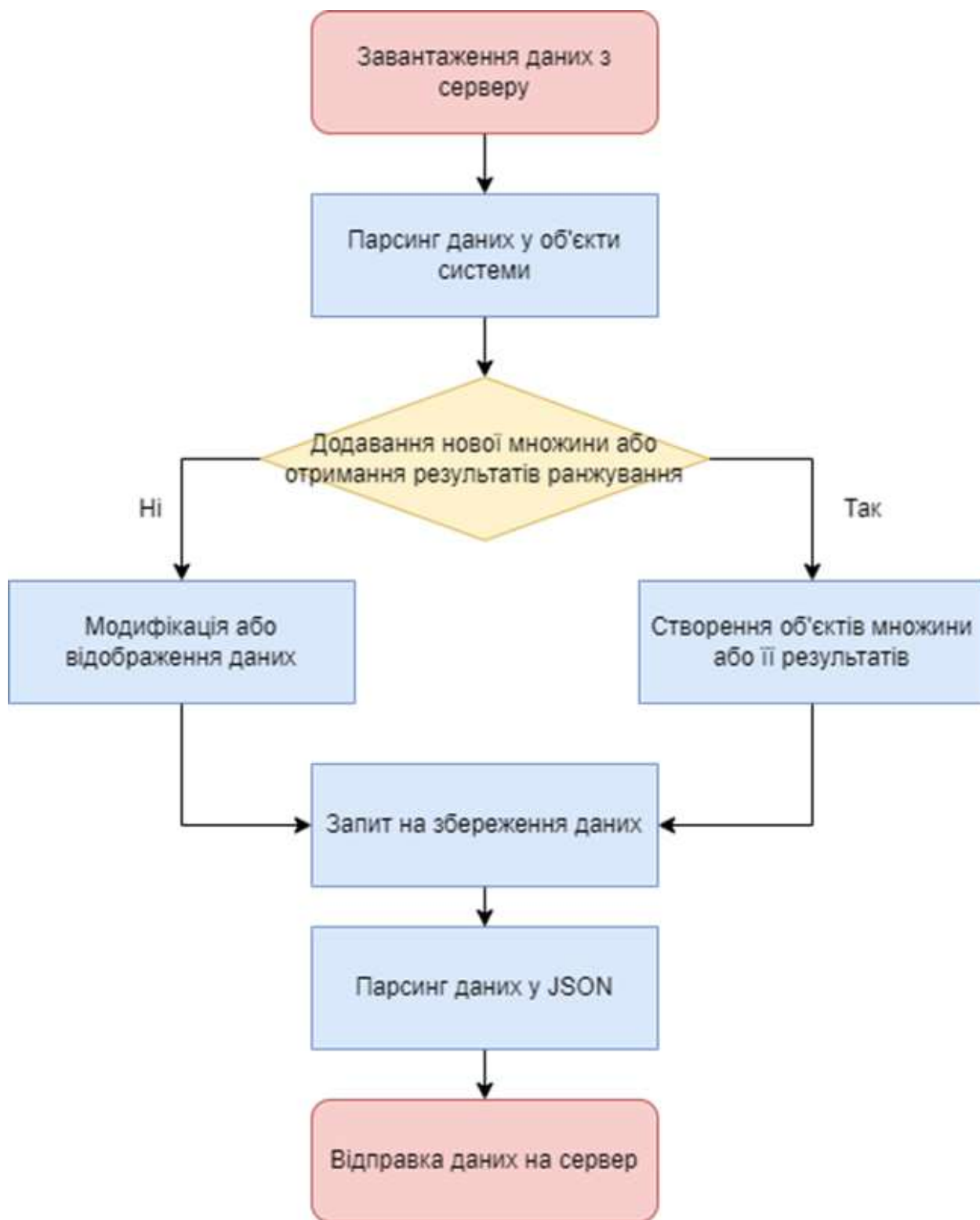


Рисунок В.4 - Діаграма діяльності роботи модуля зберігання даних

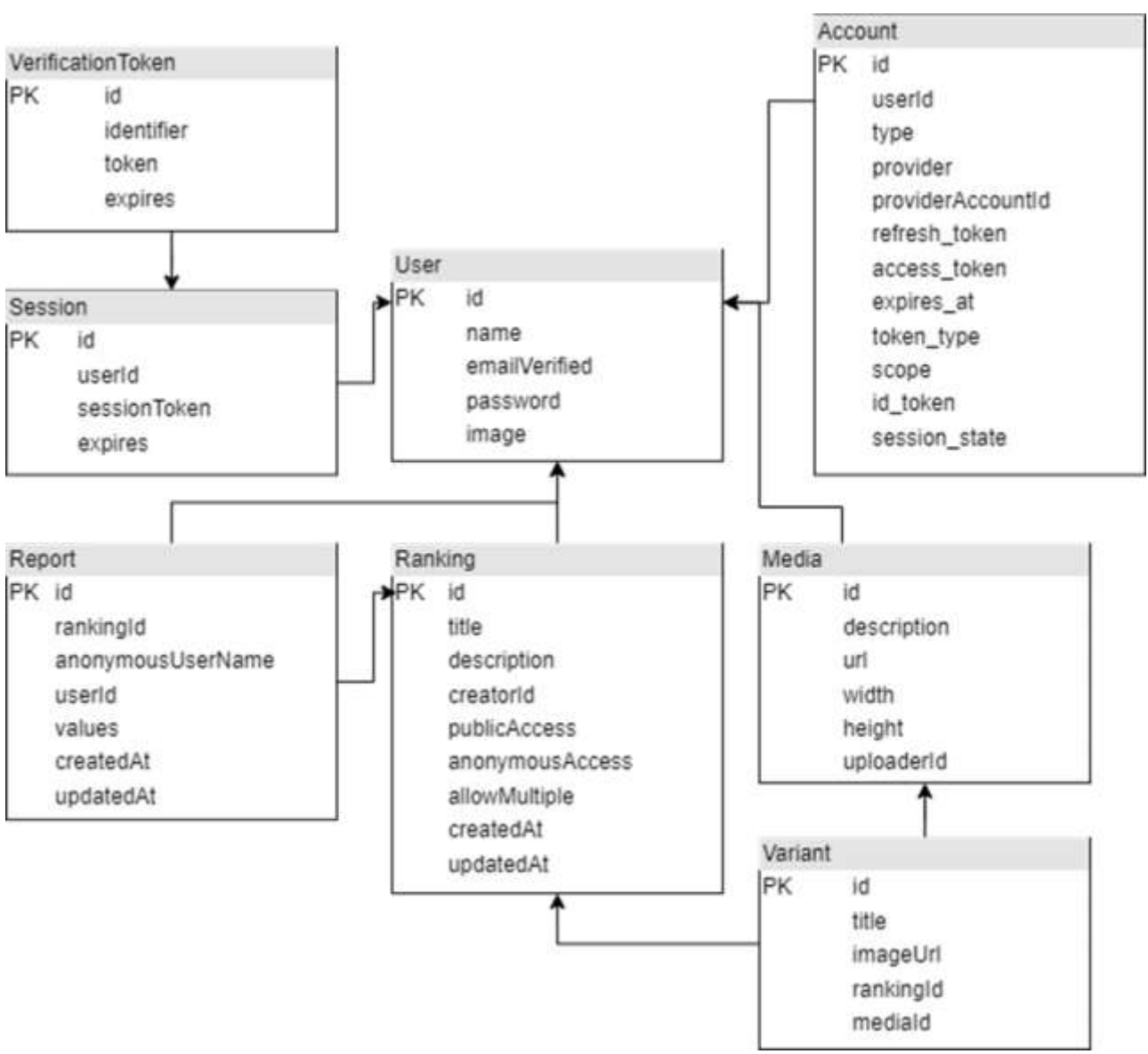


Рисунок В.5 - Зображення взаємодії моделей БД



Рисунок В.6 - Схема алгоритму створення ранжування

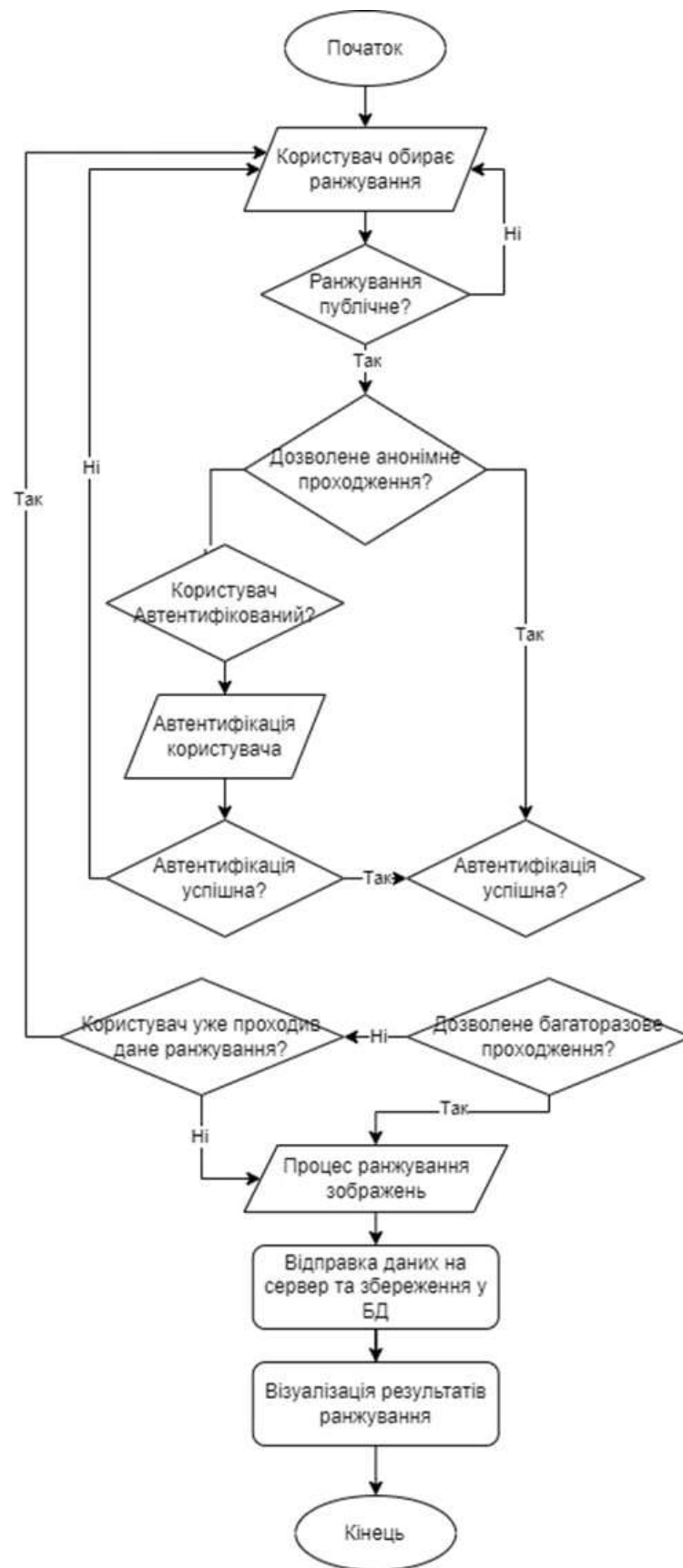


Рисунок В.7 - Схема алгоритму проходження ранжування

Додаток Г (довідниковий) Інструкція користувача

Створення ранжування для користувача починається із головної сторінки веб-додатку (рис. Г.1).



Рисунок Г.1 – Головна сторінка веб-додатку для ранжування зображень

Далі, користувачу необхідно увійти до системи за допомогою одного із двох доступних методів: вхід або реєстрація через Google або ж використовуючи E-mail та пароль (рис. Г.2, Г.3).

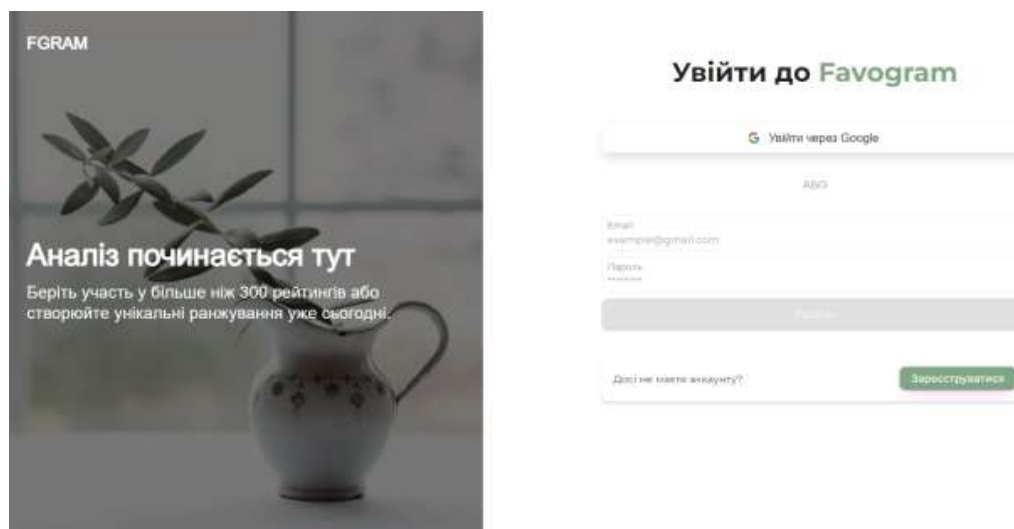


Рисунок Г.2 – Сторінка входу до системи

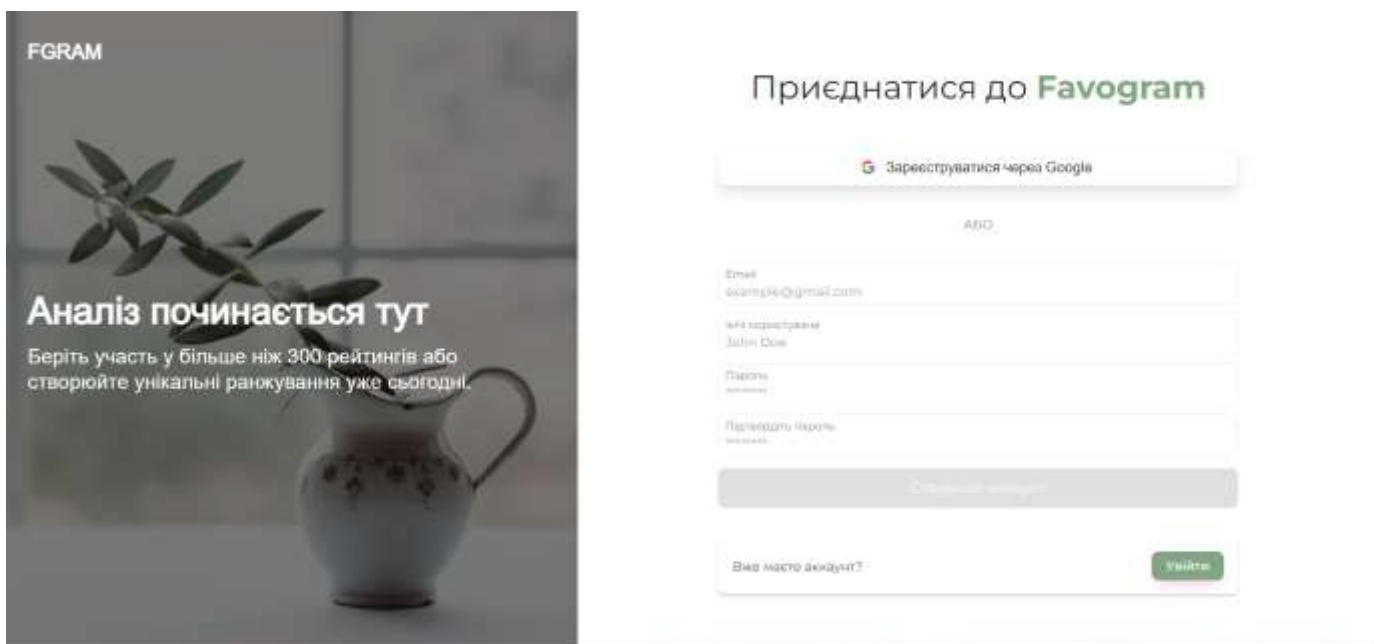


Рисунок Г.3 – Сторінка реєстрації в системі

Після авторизації, користувача буде направлено на сторінку особистого кабінету, де він за натиском на кнопку «Створити ранжування» перейде на сторінку створення. Для того, аби створити ранжування, користувачу необхідно ввести дані згідно з наявними правилами валідації та натиснути «Зберегти» (рис. Г.4).

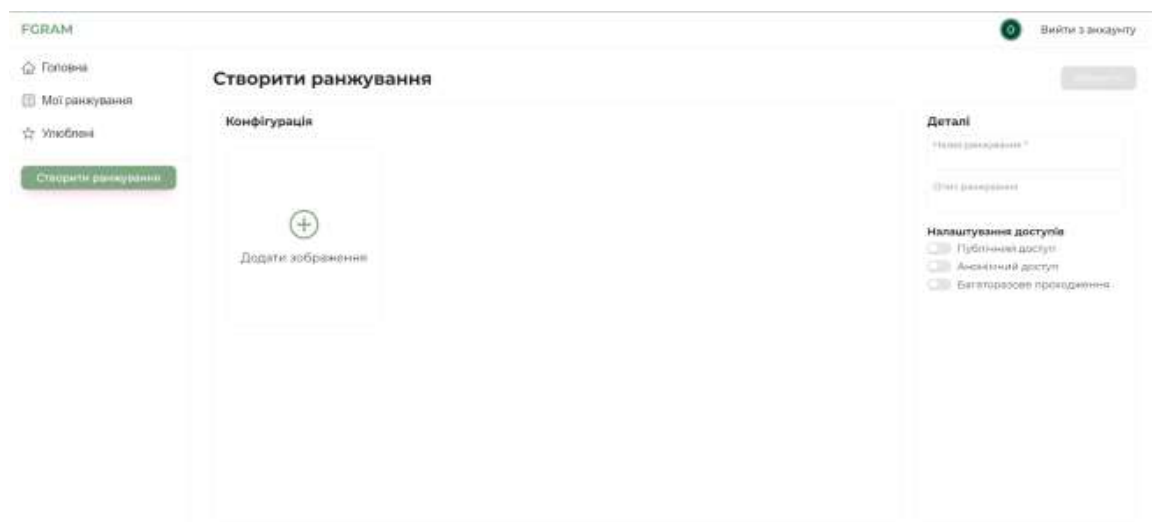


Рисунок Г.4 – Сторінка створення ранжування

Після збереження ранжування у базу даних, користувача буде направлено на сторінку «Мої ранжування», де відображається список створених ранжувань (рис. Г.5).

Назва	Опис	Анонімний доступ	Публічний доступ	Багатораз.	Кількість зображе...	Редагувати
Cool-Poll	—	Ні	Так	Ні	2	Редагувати
Private	Private	Ні	Так	Ні	2	Редагувати
Public-Poll, Anonym, Multi	—	Так	Так	Так	2	Редагувати
Public-Poll, No Anonym, No Multi	Public-Poll	Ні	Так	Ні	2	Редагувати
Alchemist	—	Ні	Так	Ні	2	Редагувати
ADT Ranking	—	Так	Так	Так	4	Редагувати
test12	test12	Так	Так	Так	2	Редагувати
main-test	main-test	Ні	Ні	Ні	5	Редагувати
updated333	test(234636)	Так	Так	Так	2	Редагувати
test	test	Ні	Ні	Ні	2	Редагувати

Рисунок Г.5 – Сторінка «Мої ранжування»

При переході на створене ранжування, користувач може оновити дані ранжування, але окрім зображень, так як ці дані залишаються незмінними і можуть бути видаленими лише із видаленням самого ранжування. Після збереження відредагованого ранжування, користувача буде перенаправлено на сторінку «Мої ранжування».

Сценарій проходження ранжування розпочинається з головної сторінки, аналогічно стартовій точці при створенні ранжування.

При переході на сторінку ранжування, користувачу необхідно пройти автоматичну валідацію, яка визначає, чи є у користувача змога пройти ранжування на основі параметрів (рис. Г.6 - Г.8).

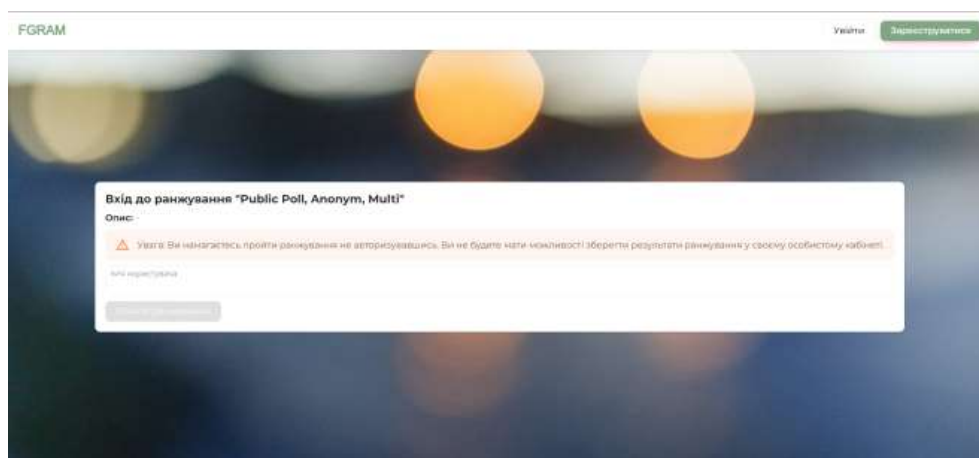


Рисунок Г.6 – Результат валідації анонімного користувача

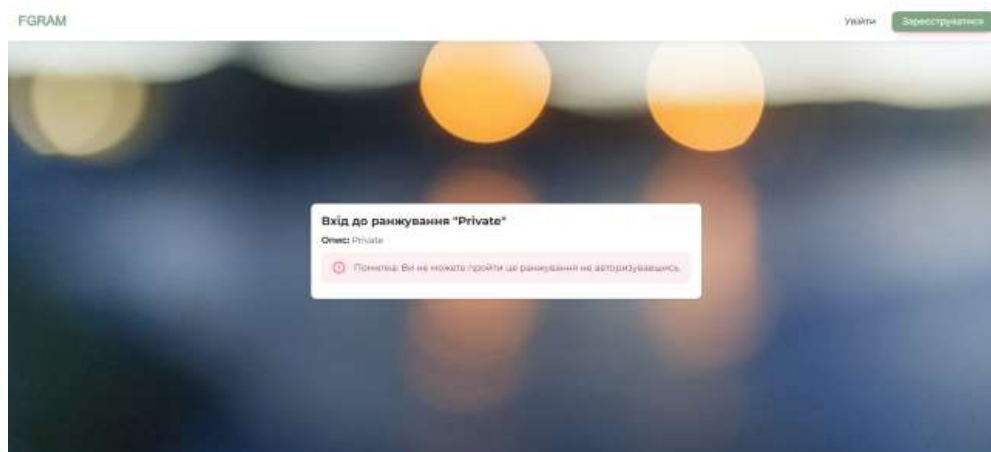


Рисунок Г.7 – Результат валідації анонімного користувача в закритому від анонімного проходження ранжуванні



Рисунок Г.8 – Результат валідації користувача при проходженні свого ранжування

Після валідації, користувач потрапляє на сторінку проходження ранжування, де він обирає зображення, яке йому більше до вподоби за допомогою слайдера. Бали зображенням вираховуються у відсотках (рис. Г.9).

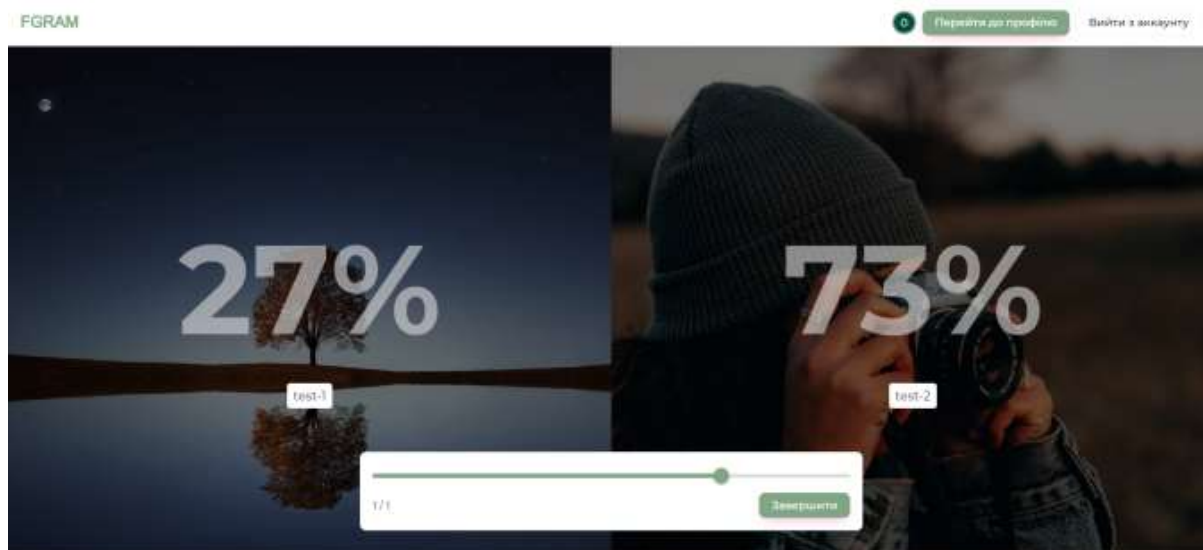


Рисунок Г.9 – Процес ранжування зображень

Коли користувач завершує проходження ранжування, його переводить на візуалізацію результатів ранжування (рис. Г.10).



Рисунок Г.10 – Сторінка результату ранжування