

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА
на тему: «Веб-сайт для колекціонування MANGA із
застосуванням скрейпінгу»

на здобуття освітнього ступеня магістра
зі спеціальності ЕЗ Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

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

(підпис) Дмитро БАГАЖКОВ
(Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНДМ-61

Дмитро БАГАЖКОВ

Керівник:
науковий ступінь,
вчене звання

Юрій КАТКОВ
д.т.н., професор

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРІЗВИЩЕ)

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність ЕЗ Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедру Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
« _____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Багажкову Дмитру Сергійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Веб-сайт для колекціонування MANGA із застосуванням скрейпінгу»

керівник кваліфікаційної роботи Юрій КАТКОВ д.т.н., доцент,

(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» жовтня 2025р. № 467

2. Строк подання кваліфікаційної роботи «26» грудня 2025р.

3. Вихідні дані до кваліфікаційної роботи:

3.1. Огляд науково-технічної літератури щодо інтелектуальних методів пошуку та класифікації контенту в системах збору цифрових колекцій MANGA (scraping, NLP, семантичний пошук, рекомендаційні системи, індексація);

3.2. Практичний аналіз застосування інтелектуальних методів пошуку та агрегування даних у прикладах: MangaDex, MangaPlus, MyAnimeList, AniList (scraping / API, нормалізація метаданих, пошук за зображенням і сем: пошук, рекомендації).

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Аналіз застосування інтелектуальних методів пошуку та фільтрації контенту у веб-сайтах для колекціонування Манги з використанням технологій веб-скрейпінгу, індексації та обробки метаданих.

4.2 Дослідження ефективності інтелектуальних методів пошуку, класифікації та рекомендації Манга-контенту на основі зібраних даних та аналіз їх впливу на зручність користування веб-сайтом.

4.3 Розробка практичних рекомендацій щодо впровадження інтелектуальних методів пошуку та агрегування даних у веб-сайті для колекціонування Манги з використанням веб-скрейпінгу, семантичного пошуку та механізмів персоналізації.

5. Перелік графічного матеріалу: *презентація*

1. Тема дипломної роботи

2. Мета, об'єкт, предмет дипломної роботи

3. Постановка завдання дипломної роботи

4. Результати Аналіз застосування інтелектуальних методів пошуку та фільтрації контенту у веб-сайтах для колекціонування Манги з використанням технологій веб-скрейпінгу, індексації та обробки метаданих.

5. Результати обґрунтування доцільності застосування інтелектуальних методів пошуку інтелектуальних методів пошуку, класифікації та рекомендації Манга-контенту на основі зібраних даних та аналіз їх впливу на зручність користування веб-сайтом.

6. Практичні рекомендації щодо впровадження інтелектуальних методів пошуку та агрегування даних у веб-сайті для колекціонування Манги з використанням веб-скрейпінгу, семантичного пошуку та механізмів персоналізації.

7. Висновки

8. Апробація

6. Дата видачі завдання «15» вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури з тематики веб-розробки, веб-скрейпінгу та цифрових бібліотек манги	15.09-05.10.25	Виконане
2	Аналіз специфіки та характеристик функціонування веб-сайтів для колекціонування та каталогізації MANGA.	06.10-18.10.25	Виконане
3	Дослідження методів збору, обробки та оновлення даних MANGA із зовнішніх веб-ресурсів із застосуванням технологій скрейпінгу.	19.10-31.10.25	Виконане
4	Моделювання архітектури веб-сайту для колекціонування MANGA з інтеграцією механізмів веб-скрейпінгу.	01.11-25.11.25	Виконане
5	Розробка основних розділів кваліфікаційної роботи.	27.11-03.12.25	Виконане
6	Розробка обов'язкових графічних та презентаційних матеріалів.	04.12-10.12.25	Виконане
7	Попередній захист кваліфікаційної роботи.	11.12-14.12.25	Виконане
8	Подання кваліфікаційної роботи до деканату.	15.12-26.12.25	Виконане

Здобувач вищої освіти

(підпис)

Дмитро БАГАЖКОВ
(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Юрій КАТКОВ
(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: _115_с., _5_рис., _25__табл., _53__джерел.

Мета роботи: Метою дипломної роботи є розробка та обґрунтування архітектури веб-сайту для колекціонування манги, що забезпечує автоматизований збір, систематизацію, пошук і фільтрацію контенту на основі технологій веб-скрейпінгу, індексації та обробки метаданих з метою підвищення ефективності відбору релевантних творів користувачами.

Об'єкт дослідження: Процеси пошуку, збирання, обробки та представлення інформації про мангу у веб-системах колекціонування цифрового контенту.

Предмет дослідження: Методи та програмні засоби веб-скрейпінгу, обробки метаданих, індексації, пошуку та фільтрації контенту, що застосовуються у веб-сайтах для колекціонування манги.

Наукове завдання роботи полягає у формуванні та дослідженні комплексного підходу до побудови веб-системи колекціонування манги, який поєднує автоматизований збір даних із зовнішніх джерел, уніфікацію та збагачення метаданих, а також реалізацію ефективних механізмів пошуку, фільтрації та персоналізованого доступу до контенту.

Метод дослідження. Методологічну основу дослідження становить системний підхід, що передбачає комплексне вивчення процесів збору, обробки та структурування даних у межах веб-сайту для колекціонування MANGA. У роботі застосовано аналітично-розрахункові методи, теоретичний аналіз із використанням методів моделювання, реконструювання та типологізації інформаційних об'єктів. Емпіричну базу дослідження сформовано на основі даних, отриманих у процесі веб-скрейпінгу з відкритих інтернет-ресурсів, що дало змогу оцінити ефективність запропонованих рішень та обґрунтувати їх практичну доцільність.

Актуальність теми – У сфері цифрових бібліотек, онлайн-каталогів та веб-платформ для колекціонування контенту особливу увагу приділено автоматизованому збору, обробці та оновленню даних з використанням технологій веб-скрейпінгу, парсингу HTML-сторінок, REST API та систем управління базами даних. Зростання популярності MANGA як цифрового культурного контенту зумовлює потребу в створенні зручних, масштабованих та актуальних веб-рішень для її колекціонування.

Галузь використання – інформаційні сервіси, цифрові бібліотеки, веб-платформи для колекціонування та каталогізації медіаконтенту.

Ключові слова: веб-сайт, MANGA, веб-скрейпінг, парсинг даних, цифрові колекції.

ABSTRACT

Text section of the qualification thesis for obtaining the Master's degree: _115__ pages, 5__ figures, _25__ tables, _53__ sources.

Research Objective: The objective of the Master's thesis is to design and substantiate the architecture of a website for manga collection that ensures automated data acquisition, systematization, search, and filtering of content based on web scraping, indexing, and metadata processing technologies in order to improve the efficiency of selecting relevant works for users.

Object of Study: The processes of searching, collecting, processing, and presenting manga-related information in web-based systems for digital content collection.

Subject of Study: Methods and software tools for web scraping, metadata processing, indexing, search, and content filtering applied in websites designed for manga collection.

Scientific Task: The scientific task of the research is to develop and investigate an integrated approach to building a manga collection web system that combines automated data acquisition from external sources, metadata unification and enrichment, as well as the implementation of efficient search, filtering, and personalized content access mechanisms.

Research Methodology: The methodological framework of the study is based on a systems approach that provides a comprehensive analysis of data collection, processing, and structuring processes within a manga collection website. The research employs analytical and computational methods, as well as theoretical analysis using modeling, reconstruction, and typologization of information objects. The empirical basis of the study is formed from data obtained through web scraping of open Internet resources, which made it possible to evaluate the effectiveness of the proposed solutions and substantiate their practical applicability.

Relevance of the Topic: In the domain of digital libraries, online catalogs, and web platforms for content collection, increasing attention is paid to the automated collection, processing, and updating of data using web scraping technologies, HTML page parsing, REST APIs, and database management systems. The growing popularity of manga as a form of digital cultural content creates a demand for user-friendly, scalable, and up-to-date web solutions for its collection and cataloging.

Field of Application: Information services, digital libraries, and web platforms for collecting and cataloging media content.

Keywords: website, manga, web scraping, data parsing, digital collections.

ЗМІСТ

	Ст.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	14
ВСТУП.....	15
1 АНАЛІЗ ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА ФІЛЬТРАЦІЇ КОНТЕНТУ У ВЕБ-САЙТАХ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНГИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ВЕБ-СКРЕЙПІНГУ, ІНДЕКСАЦІЇ ТА ОБРОБКИ МЕТАДАНИХ	21
1.1 Поняття, сутність та особливості веб-скрейпінгу, індексації та обробки метаданих.....	21
1.2 Аналіз існуючих веб-систем для колекціонування контенту (манги, коміксів, медіа).....	23
1.3. Методи та алгоритми автоматизованого збору веб-даних.....	38
1.4. Огляд інструментів і бібліотек для веб-скрейпінгу (BeautifulSoup, Scrapy, Selenium, Playwright тощо).....	41
1.5. Аналіз вимог користувачів і функціональних характеристик системи.....	46
2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ, КЛАСИФІКАЦІЇ ТА РЕКОМЕНДАЦІЇ МАНГА-КОНТЕНТУ НА ОСНОВІ ЗІБРАНИХ ДАНИХ ТА АНАЛІЗ ЇХ ВПЛИВУ НА ЗРУЧНІСТЬ КОРИСТУВАННЯ ВЕБ-САЙТОМ.	52
2.1. Архітектура системи пошуку та фільтрації контенту у веб-сайтах для колекціонування манги з використанням технологій веб-скрейпінгу, індексації та обробки метаданих	52
2.1.1 Рівень джерел даних	54
2.1.2 Рівень збору та обробки.....	55
2.1.3 Рівень зберігання.....	63

2.1.4 Рівень пошуку та фільтрації.....	70
2.1.5 Рівень представлення.....	76
2.2 Реалізація та оцінка інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах.....	81
2.2.1 Інтелектуальні методи пошуку та фільтрації манги контенту у веб-сайтах.....	81
2.2.2 Реалізація інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах	82
2.2.3 Реалізація інтелектуальних методів оцінки та фільтрації манги контенту у веб-сайтах	86
2.3 Методи оцінки ефективності використанням технологій веб-скрейпінгу, індексації та обробки метаданих для манга-каталогу	87
2.3.1 Метрики якості пошуку для манга-каталогу.....	87
2.3.2 Порівняльна ефективність алгоритмів пошуку для манга-каталогу	90
2.3.2 Вплив фільтрації на якість пошуку для манга-каталогу.....	86
2.3.3 Ефективність рекомендацій рекомендаційної системи.....	91
2.3.4 Оптимізація продуктивності системи пошуку та рекомендацій для манга-каталогу.....	92
 3 РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА АГРЕГУВАННЯ ДАНИХ У ВЕБ-САЙТІ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНГИ	 96
3.1 Реалізація модуля веб-скрейпінгу для інформаційного наповнення веб сайту.....	96
3.2 Реалізація серверної логіки та API.....	97
3.3 Розробка користувацького інтерфейсу.....	98
3.4 Інтеграція з базою даних та обробка запитів.....	100

3.5. Можливості подальшого розвитку системи.....	
ВИСНОВКИ.....	104
ПЕРЕЛІК ПОСИЛАНЬ.....	107
Копії обов’язкових креслень (Презентація)	113

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI – Artificial Intelligence (Штучний інтелект);
API – Application Programming Interface (Інтерфейс прикладного програмування);
CSS – Cascading Style Sheets (Каскадні таблиці стилів);
DB – Database (База даних);
HTML – HyperText Markup Language (Мова розмітки гіпертексту);
JS – JavaScript (Мова програмування JavaScript);
JSON – JavaScript Object Notation (Формат обміну даними на основі JavaScript);
ML – Machine Learning (Машинне навчання);
NLP – Natural Language Processing (Обробка природної мови);
REST – Representational State Transfer (Передача репрезентативного стану;
архітектурний стиль);
SEO – Search Engine Optimization (Пошукова оптимізація);
SPA – Single Page Application (Односторінковий додаток);
SQL – Structured Query Language (Мова структурованих запитів);
UI – User Interface (Користувацький інтерфейс);
URL – Uniform Resource Locator (Уніфікований локатор ресурсу);
UX – User Experience (Користувацький досвід);
ВПЗ – Веб-парсинг (Веб-скрейпінг) даних;
HTTP – HyperText Transfer Protocol (Протокол передачі гіпертексту);
DOM – Document Object Model (Об'єктна модель документа);
СУБД – Система управління базами даних;
ІІ – Штучний інтелект.

ВСТУП

В атестаційній роботі магістра розглядається актуальна задача застосування сучасних автоматизованих методів обробки даних для створення спеціалізованого веб-ресурсу (сайту) для колекціонування манги та агрегування інформації про неї з використанням технології веб-скрейпінгу.

Обґрунтування вибору теми та її актуальність. Сфера інтересу колекціонерів та любителів манги, яка охоплює збирання, каталогізацію та управління колекціями графічних новел, переживає значну цифровізацію. Інформаційні технології (ІТ) кардинально трансформують підходи до колекціонування, впливаючи на всі аспекти цієї діяльності - від пошуку та ідентифікації видань до організації каталогу та обміну інформацією в спільнотах, а саме: персоналізацію та створення рекомендаційних систем за допомогою штучного інтелекту (АІ) та машинного навчання; забезпечують автоматизований збір та агрегування даних з розрізнених джерел за допомогою веб-скрейпінгу; впроваджують семантичний пошук для знаходження контенту за змістом, а не лише за ключовими словами; створюють інтелектуальні інструменти для каталогізації, відстеження прогресу читання та оцінювання; забезпечують генеративні описи та автоматичну класифікацію за жанрами; впроваджують соціальні функції на основі аналізу даних про вподобання; покращують управління особистою бібліотекою та багато іншого. Тобто ІТ-технології роблять процес колекціонування манги більш систематизованим, індивідуалізованим та ефективним. Ті платформи та ентузіасти, які інвестують у автоматизацію збору даних (веб-скрейпінг), семантичний аналіз та аналітику вподобань, отримують значні переваги за рахунок повноти інформації, точності пошуку та персоналізації рекомендацій. Однак сфера колекціонування манги під час впровадження інформаційних технологій стикається з унікальними викликами, такими як: розрізненість та неструктурованість джерел інформації, проблема авторських прав та умов використання даних, необхідність обробки

великих обсягів мультимовного контенту, а також швидка динаміка оновлення випусків нових глав та томів. Тому дослідження та впровадження цифрових технологій: веб-скрейпінгу, семантичного пошуку, машинного навчання для рекомендацій, NLP для обробки описів та ефективної організації баз даних є вкрай актуальним для створення сучасних інструментів колекціонера.

Формулювання завдання. Колекціонування манги перетворилося на поширене хобі з глобальною спільнотою ентузіастів. Оскільки цифрові платформи та онлайн-бази даних стають основним джерелом інформації для колекціонерів, ефективний пошук, агрегування та управління цими даними стають вирішальними для зручності користувача. Сучасні колекціонери очікують не лише повноти та точності інформації про видання, але й зручних інструментів для пошуку, автоматичного оновлення колекції, отримання персоналізованих рекомендацій та легкої каталогізації. Ці очікування створюють високі вимоги до спеціалізованих веб-ресурсів та додатків. Завдяки впровадженню ІТ-технологій, зокрема автоматизованих методів збору даних (веб-скрейпінгу) та інтелектуального аналізу, можна створити потужну систему, яка трансформує підхід до колекціонування, впливаючи на всі аспекти — від первинного пошуку та ідентифікації до організації особистої бібліотеки та соціальної взаємодії, забезпечуючи при цьому точність та актуальність інформації. Зосередження на пошуку технічних рішень, що дозволяють забезпечити повноту даних, гнучкість пошуку, масштабованість та економічну ефективність процесів обробки інформації в умовах швидко оновлюваного контенту та різноманітних форматів публікацій, є актуальною науково-практичною задачею.

Ступінь вивчення проблеми. Проблема застосування інтелектуальних технологій для створення систем колекціонування спеціалізованого контенту (на прикладі манги) є частково вивченою [1-7]. Питання веб-скрейпінгу, побудови рекомендаційних систем та семантичного пошуку широко розглядаються в загальному контексті ІТ. Існує міцна технологічна основа для впровадження цих методів, проте їх специфічна інтеграція та адаптація саме для нішевої задачі колекціонування графічних новел містить низку унікальних викликів, що

вимагають подальших цілеспрямованих досліджень та розробок. Зокрема, області, які потребують найбільшої уваги, включають: створення стійких та етичних механізмів скрейпінгу динамічних і розрізнених джерел інформації про мангу, подальше вдосконалення алгоритмів семантичного пошуку з урахуванням особливостей жанрової класифікації та мультимовності, розробку ефективних гібридних моделей рекомендацій (контентно-орієнтованих і колаборативної фільтрації) для невеликих спільнот, а також ефективну інтеграцію всіх цих компонентів у зручний та безперебійний веб-інтерфейс для кінцевого користувача-колекціонера.

Специфіка джерельної бази. Аналіз джерельної бази свідчить про її подвійну природу: з одного боку — це наукові публікації з технологій обробки даних (ML, NLP, веб-скрейпінг), з іншого — практичні, часто неструктуровані онлайн-джерела інформації про саму мангу (бази даних, вікі, сайти видавництва). Така неоднорідність зумовлює необхідність систематизації джерел за визначеними критеріями, зокрема за типом (наукова література, API, веб-сайти для скрейпінгу), рівнем надійності, якістю структурованості даних та юридичними обмеженнями на використання. Це дозволить виявити наявні технологічні проблеми (наприклад, нестабільність структури цільових сторінок) та концептуальні недоліки (наприклад, розбіжності в метаданих з різних джерел) для їх подальшого вирішення в рамках розробки системи.

Об'єкт дослідження: Процеси пошуку, збирання, обробки та представлення інформації про мангу у веб-системах колекціонування цифрового контенту.

Предмет дослідження: Методи та програмні засоби веб-скрейпінгу, обробки метаданих, індексації, пошуку та фільтрації контенту, що застосовуються у веб-сайтах для колекціонування манги.

Мета роботи: Метою дипломної роботи є розробка та обґрунтування архітектури веб-сайту для колекціонування манги, що забезпечує автоматизований збір, систематизацію, пошук і фільтрацію контенту на основі технологій веб-скрейпінгу, індексації та обробки метаданих з метою підвищення ефективності відбору релевантних творів користувачами.

Наукове завдання – Полягає у формуванні та дослідженні комплексного підходу до побудови веб-системи колекціонування манґи, який поєднує автоматизований збір даних із зовнішніх джерел, уніфікацію та збагачення метаданих, а також реалізацію ефективних механізмів пошуку, фільтрації та персоналізованого доступу до контенту.

Метод дослідження. Методологічну основу дослідження становить системний підхід, що передбачає комплексне вивчення процесів збору, обробки та структурування даних у межах веб-сайту для колекціонування MANGA. У роботі застосовано аналітично-розрахункові методи, теоретичний аналіз із використанням методів моделювання, реконструювання та типологізації інформаційних об'єктів. Емпіричну базу дослідження сформовано на основі даних, отриманих у процесі веб-скрейпінгу з відкритих інтернет-ресурсів, що дало змогу оцінити ефективність запропонованих рішень та обґрунтувати їх практичну доцільність.

Завдання роботи:

1. Провести аналіз застосування інтелектуальних методів пошуку та фільтрації контенту у веб-сайтах для колекціонування манґи з використанням технологій веб-скрейпінгу, індексації та обробки метаданих

2. Обґрунтувати доцільність та дослідити ефективність інтелектуальних методів пошуку, класифікації та рекомендації манґа-контенту на основі зібраних даних та аналіз їх впливу на зручність користування веб-сайтом

3. Розробити практичні рекомендації щодо впровадження інтелектуальних методів пошуку та агрегування даних у веб-сайті для колекціонування манґи з використанням веб-скрейпінгу, семантичного пошуку та механізмів персоналізації.

Результати дослідження. У ході виконання дипломної роботи отримано такі результати: проаналізовано існуючі веб-платформи для каталогізації та колекціонування манґи й визначено їхні функціональні обмеження; розроблено архітектуру веб-сайту для колекціонування манґи з використанням технологій веб-скрейпінгу та індексації даних; сформовано модель обробки та уніфікації

метаданих манги з різних джерел; реалізовано механізми інтелектуального пошуку, фільтрації та персоналізованих рекомендацій; визначено функціональні та нефункціональні вимоги до системи й описано сценарії її використання.

Наукова новизна. Наукова новизна отриманих результатів полягає в тому, що: запропоновано архітектурний підхід до побудови веб-сайту колекціонування манги, який інтегрує веб-скрейпінг, індексацію та семантичну обробку метаданих; удосконалено підхід до пошуку та фільтрації контенту за рахунок поєднання фасетної навігації та семантичного аналізу описів творів; обґрунтовано використання персоналізованих рекомендацій у системах колекціонування манги як засобу підвищення релевантності контенту.

Практична значущість результатів дослідження. Практична значущість результатів полягає у можливості використання розробленої архітектури та програмних рішень під час створення веб-сайтів для колекціонування, каталогізації та аналізу медіаконтенту. Отримані результати можуть бути застосовані при розробці інформаційних сервісів, цифрових бібліотек, онлайн-каталогів, а також у навчальному процесі під час вивчення веб-розробки та технологій веб-скрейпінгу.

Апробація результатів надається в 1 статті, 2 тезисах в рецензованих наукових журналах і виданнях.

В статті:

Багажков Д.С. (2025) ВЕБ-САЙТ ДЛЯ КОЛЕКЦІОНУВАННЯ MANGA: ВПРОВАДЖЕННЯ ТА ОПТИМІЗАЦІЯ WEB-SCRAPING // Наукові записки Державного університету інформаційно-комунікаційних технологій №2, 2025, Подано до друку.

<https://journals.dut.edu.ua/index.php/sciencenotes/issue/view/186>

В тезах:

1. Багажков Д.С. (2025) РОЗРОБКА ВЕБ-САЙТУ ДЛЯ КОЛЕКЦІОНУВАННЯ MANGA ІЗ ЗАСТОСУВАННЯМ СКРЕЙПІНГУ VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT».

Збірник тез 15 квітня 2025. – К.: ДУІКТ, ст. 19-23.
https://duikt.edu.ua/uploads/p_2779_40288420.pdf

2. Багажков Д.С. (2025) ВЕБ-СКРЕЙПІНГ У СИСТЕМІ КОЛЕКЦІОНУВАННЯ КОНТЕНТУ НА ГІБРИДНИХ РІШЕННЯХ НРЕ // Сучасні досягнення компанії Hewlett Packard Enterprise в галузі іт та нові можливості їх вивчення і застосування/ Збірник тез 11 грудня 2025. – К.: ДУІКТ, подане для друку - https://duikt.edu.ua/ua/news-1-576-14841-duikt-zaproshue-na-mizhnarodnu-naukovo-tehnichnu-konferenciyu-suchasni-dosyagnennya-kompanii-hewlett-packard-enterprise-v-galuzi-it-ta-novi-mozhlivosti-ix-vivchennya-i-zastosuvannya_kafedra-shtuchnogo-intelektu

1 АНАЛІЗ ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА ФІЛЬТРАЦІЇ КОНТЕНТУ У ВЕБ-САЙТАХ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНІ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ВЕБ-СКРЕЙПІНГУ, ІНДЕКСАЦІЇ ТА ОБРОБКИ МЕТАДАНИХ

1.1 Поняття, сутність та особливості веб-скрейпінгу, індексації та обробки метаданих

Веб-скрейпінг (англ. *web scraping*) — це процес автоматизованого збору даних із веб-ресурсів для подальшої обробки, аналізу або збереження у структурованому форматі. Його сутність полягає в імітації дій користувача (відправка запитів, навігація) з метою масового та ефективного збору даних.

На відміну від звичайного стандартного доступу через API для перегляду сторінок користувачем, скрейпінг передбачає прямий парсинг (розбір) HTML-коду веб-сторінок для вилучення потрібної інформації за допомогою спеціальних програмних скриптів. Це дозволяє програмно отримувати великі обсяги інформації без необхідності ручного копіювання. Застосування веб-скрейпінгу охоплює різні сфери: моніторинг цін, агрегування новин, аналіз контенту, формування баз знань і створення каталогів. У сучасних умовах зростання обсягів веб-даних скрейпінг стає важливим інструментом для розробників інформаційних систем, що потребують регулярного оновлення інформації.[8, 9, 10]

Для реалізації скрейпінгу використовуються бібліотеки та фреймворки, що дозволяють завантажувати HTML-код сторінок, аналізувати їх структуру (DOM), вилучати потрібні елементи та перетворювати дані у придатний для зберігання вигляд (наприклад, JSON або SQL-таблиці). Однією з головних переваг цього

підходу є можливість інтеграції процесів збору даних безпосередньо в архітектуру веб-систем.

Особливості включають [11, 12, 13, 14, 15]:

- Автоматизацію: масштабоване отримання даних без ручної праці.
- Адаптивність: можливість налаштування під різноманітну структуру цільових сайтів.
- Правові та етичні ризики: необхідність дотримання robots.txt, умов використання сайту та законів про захист даних.
- Технічні виклики: боротьба з захистом від ботів (CAPTCHA, динамічний JavaScript), нестабільність структури сайтів.

Індексація (англ. *Indexing*) – це процес створення структурованого покажчика (індексу) зібраних даних для забезпечення швидкого та релевантного пошуку. В контексті колекціонування манги, індексація трансформує неструктуровані текстові дані (назви, описи, жанри) у оптимізовану для пошуку внутрішню структуру.

Сутність індексації – перетворення даних на форму, ефективну для машинного пошуку.

Особливості:

- Підвищення продуктивності: пошук ведеться не по сирій базі даних, а по оптимізованому індексу.
- Підтримка складних запитів: можливість реалізації повнотекстового, фасетного (за жанром, автором) та семантичного пошуку.
- Попередня обробка даних: часто включає токенізацію, приведення до нижнього регістру, видалення стоп-слів, лематизацію.

Обробка метаданих (англ. *Metadata processing*) – це аналіз, очищення, стандартизації та збагачення даних, що описують основні об'єкти (мангу). Метадані у даній сфері – це структурована інформація про публікацію: назва, автор (мангака), художник, видавництво, статус, жанри, теги, рік випуску, пов'язані аніме-адаптації, рейтинги, синопсис.

Сутність процесу – перетворення сирих, можливо несумісних даних з різних джерел у узгоджений, чистий та придатний для аналізу формат.

Особливості:

- Нормалізація: приведення до єдиного стандарту (напр., єдиного написання імені автора).
- Дедуплікація: виявлення та об'єднання записів-дублікатів.
- Верифікація та збагачення: перевірка точності та доповнення даних з альтернативних джерел.
- Створення зв'язків: встановлення взаємозв'язків між об'єктами (цикли, сиквели, спін-офи).

Одже існує взаємозв'язок: ці три процеси утворюють технологічний ланцюг для створення інформаційної системи:

1. Веб-скрейпінг здійснює первинний збір сирих даних.
2. Індексція готує оброблені метадані для миттєвого та інтелектуального пошуку кінцевим користувачем.
3. Обробка метаданих очищає, стандартизує та структурує ці дані.

Результатом є створення власної узгодженої бази знань про манґу, що є фундаментом для подальших інтелектуальних функцій сайту.

1.2 Аналіз існуючих веб-систем для колекціонування контенту (манґи, коміксів, медіа)

Аналіз ринку наявних рішень дозволяє виявити поширені моделі, переваги, недоліки та ніші для оптимізації. Існуючі системи умовно можна класифікувати за їх основною метою та функціоналом, що надано у таблицях 1.1-1.4 [16, 17, 18, 19, 20, 21, 22]

Пояснення: В таблицях 1.1-1.4 представлено класифікацію та порівняльний аналіз існуючих рішень для колекціонування контенту. Кожна таблиця розглядає окрему категорію платформ та показує:

1. Категорії платформ (заголовки таблиці):

- Таблиця 1.1: Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Масштабні, ком'юніті-орієнтовані агрегатори") (AniList, MAL)

- Таблиця 1.2 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Спеціалізовані сайти-бібліотеки манги") (MangaDex)

- Таблиця 1.3 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Універсальні платформи та інструменти") (Notion, Google Tables)

- Таблиця 1.4: Спеціалізовані сайти колекціонерів інших медіа (Discogs, Goodreads)

2. Аналіз кожної категорії за однаковою структурою (стовпці):

- Критерій аналізу - що саме досліджується
- Характеристика категорії - опис особливостей цього типу платформ
- Наслідки для розробки - як ця інформація впливає на проєкт веб-сайту для колекціонування манги

3. Ключові висновки (після таблиць):

- Виявлення сильних і слабких сторін існуючих рішень
- Визначення ніш для нового продукту
- Формування вимог до власної розробки

Таблиця важливі для проєкту тому, показує, що вже існує на ринку; дозволяє уникнути помилок конкурентів; визначає унікальні переваги майбутнього веб-сайту; дає обґрунтування для вибору технологій та функцій. Ці таблиці слугують основою для обґрунтування необхідності створення нового веб-сайту для колекціонування манги, оскільки показують, що жодна з існуючих категорій не задовольняє повністю потреби колекціонерів.

Пояснення щодо позначок: `<ul><li>` в таблицях:

Це маркери (маркеровані списки) у форматах HTML / Markdown, які використовуються для візуального оформлення переліків.

- `<ul>` — це тег Unordered List (невпорядкований список) в HTML. Він створює список, де кожен елемент зазвичай позначається крапкою або іншим символом (не цифрою).
- `<li>` — це тег List Item (елемент списку). Кожен пункт у списку повинен бути укладений у цей тег.
- У тексті:
- `<ul>` — початок списку.
- `<li>` — кожен окремий пункт списку (наприклад, "Додавання творів у статусні списки").
- `</li>` — закриття пункту (необов'язково в сучасному HTML, але часто використовується для чіткості).
- `</ul>` — закриття всього списку.

Висновки до таблиць:

Таблиця 1.1: Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Масштабні, ком'юніті-орієнтовані агрегатори") (AniList, MAL)

Аналіз масштабних ком'юніті-агрегаторів демонструє їхню роль як надійного, соціально орієнтованого джерела метаданих. Їхня головна сила — глобальна, постійно оновлювана база та сильна спільнота. Однак для завдань глибокого колекціонування вони мають критичні недоліки: негнучка каталогізація, відсутність інструментів для опису фізичних екземплярів та рекомендації, занадто загальні через упор на популярність. Це формує першу нішу: необхідність системи, що інтегрує ці соціальні бази даних, але надає значно глибший рівень каталогізації та персоналізації для самого колекціонера.

Таблиця 1.2 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Спеціалізовані сайти-бібліотеки манги") (MangaDex).

Аналіз категорії спеціалізованих бібліотек (MangaDex, MangaUpdates) визначає їх як критично важливе та оптимальне технічне джерело метаданих для

веб-сайту колекціонування манги. Ці платформи пропонують неперевершену за повнотою та акуратністю інформацію про твори (авторський склад, публікаційну історію, зв'язки, альтернативні назви), що становить ідеальну основу («скелет») для майбутньої бази даних. Однак їхній головний фокус на цифровому контенті та скануваннях, а також слабкі соціальні та рекомендаційні функції, залишають значну прогалину. Саме ця прогалина формулює стратегічну нішу для розробки нового рішення: створити систему, яка автоматично заповнюватиме свій каталог точними даними з цих бібліотек (вирішуючи технічні виклики їх скрейпінгу/API), але на цій міцній основі будуватиме зручний інтерфейс для управління саме фізичною колекцією та розширені інтелектуальні сервіси (семантичний пошук, контентні рекомендації), яких в оригінальних джерелах бракує. Таким чином, новий веб-сайт не повинен конкурувати з цими бібліотеками в створенні каталогу, а має інтелектуально використовувати їхні дані для надання значно вищої доданої вартості для колекціонера

Таблиця 1.3: Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Універсальні платформи та інструменти") (Notion, Google Tables)

Універсальні платформи підкреслюють ключову потребу серйозних колекціонерів – абсолютний контроль над структурою даних і можливість детально описати кожен фізичний екземпляр. Їхня гнучкість є еталоном, якого необхідно домагатись. Проте їхній фатальний недолік – повна відсутність автоматизації та інтеграції з глобальними базами знань. Це чітко визначає основну конкурентну перевагу нового рішення: поєднати гнучкість каталогізації Notion із потужною автоматизацією наповнення через скрейпінг.

Таблиця 1.4: Спеціалізовані сайти колекціонерів інших медіа (Discogs, Goodreads)

Аналіз успішних платформ для колекціонування інших видів медіа (Discogs, Goodreads, Letterboxd) чітко демонструє оптимальну архітектурну модель для цільового веб-сайту. Вони є доказом того, що можна ефективно поєднати три ключові компоненти: (1) глобальну краудсорсингову базу даних, (2) потужні

інструменти для ведення особистої колекції з деталізацією екземплярів, та (3) активну соціальну та рекомендаційну систему. Таким чином, основним орієнтиром для розробки має стати створення "Discogs/Goodreads для манги". Це передбачає адаптацію їхніх успішних принципів — поділ "твір/екземпляр", соціальний Engagement, рекомендації на основі оцінок — до специфіки манги (трекінг глав, автори/артисти, зв'язок з аніме). Подолання їхнього головного недоліку (відсутність охоплення ніші манги) і становить основну цінність та науково-практичне завдання нового проекту.

Нижче наведені таблиці 1.1, 1.2, 1.3, 1.4

Таблиця 1.1 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Масштабні, ком'юніті-орієнтовані агрегатори") [23, 24, 25, 26, 27]

Критерій аналізу	Характеристика категорії "Масштабні ком'юніті-агрегатори"	Наслідки для розробки власного рішення
Представники	AniList, MyAnimeList (MAL), Kitsu.	Використання як основного джерела метаданих через API (з обмеженнями), або як зразка соціального функціоналу.
Основна суть (ціль)	Соціально-орієнтовані платформи з глобальною базою аніме, манги, ранобе, що наповнюється та модерується спільнотою.	Власний проект може зосередитися не на конкуренції у створенні глобальної бази, а на інтелектуальній обробці та персоналізації існуючих даних для конкретного колекціонера.
Ключовий функціонал для колекціонування	- Додавання творів у статусні списки (<i>Reading, Completed</i>). - Власні оцінки та трекінг прогресу (кількість глав). - Статистика та візуалізація колекції. - Рекомендації на основі колаборативної фільтрації або соціальних зв'язків.	Необхідно передбачити імпорт цих базових функцій як мінімум для зручності міграції користувачів. Власні рішення мають бути глибшими.
Основні переваги	- Величезний, постійно оновлюваний каталог. - Сильне, активне ком'юніті. - Наявність публічних API (частково).	Не ефективно дублювати. Краща стратегія – інтегрувати дані з цих джерел (через API/скрейпінг), а не створювати свою базу з нуля.

Критерій аналізу	Характеристика категорії "Масштабні ком'юніті-агрегатори"	Наслідки для розробки власного рішення
Ключові недоліки (з погляду глибокого колекціонування)	- Негнучка каталогізація: відсутність власних атрибутів (ISBN, стан, вартість, видавництво). - Фокус на творі, а не на екземплярі: не враховуються деталі фізичного/цифрового носія. - Загальні рекомендації: алгоритми часто базуються на популярності, а не на нішевих вподобаннях.	Формулюють чіткі конкурентні переваги для нового сайту: - Детальна каталогізація екземплярів. - Гнучка система власних тегів і полів. - Персоналізація на основі контенту (семантичний аналіз) та точної статистики колекції.
Проблеми для веб-скрейпінгу	Сайти добре захищені (CAPTCHA, rate limiting). Використання API суворо регламентоване (кількість запитів, необхідність аутентифікації).	Підтверджує складність прямого скрейпінгу. Архітектура має передбачати: - Етичне використання API з кешуванням. - Наявність резервних джерел даних (менш захищені сайти-бібліотеки). - Міцну обробку помилок та затримки запитів.

Таблиця 1.2 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Спеціалізовані сайти-бібліотеки манги")[23, 24, 25, 26, 27]

Критерій аналізу	Характеристика категорії "Спеціалізовані сайти-бібліотеки манги"	Наслідки для розробки власного рішення
Представники	MangaDex, MangaUpdates.	Ключові джерела для отримання найточніших метаданих про конкретні манга-твори, їх публікаційну історію та зв'язки.
Основна суть (ціль)	Платформи, головною метою яких є технічна каталогізація манги як культурного продукту, часто з інтеграцією з групами сканування та перекладу.	Власний проект може використовувати ці дані як "скелет" для опису твору, доповнюючи його функціями для колекціонування екземплярів.
Ключовий функціонал	- Глибока, детальна каталогізація (автори, артисти, журнали публікації, демографічні групи). - Точний трекінг релізів глав і томів (дати, статуси, групи перекладу). - Система зв'язків між творами (сиквели, приквели, спін-офи, альтернативні версії). - Ретельний перелік альтернативних та офіційних назв.	Необхідно запланувати імпорт/синхронізацію саме цих технічних даних. Власна система повинна зберігати та відображати цю інформацію структуровано.
Основні переваги	Найвища деталізація та точність метаданих саме про мангу як твір.	Це оптимальні цільові джерела для веб-скрейпінгу (після аналізу robots.txt та умов

Критерій аналізу	Характеристика категорії "Спеціалізовані сайти-бібліотеки манги"	Наслідки для розробки власного рішення
	- Актуальна інформація про випуски глав, що важливо для трекінгу прогресу. - Часто мають зручну та стабільну структуру даних для автоматизованого збору.	використання). Їхні дані – основа для наповнення каталогу.
Ключові недоліки (з погляду колекціонування)	- Фокус на цифровому контенті та скануваннях (особливо MangaDex), а не на фізичних колекціях. - Слабкий соціальний та рекомендаційний функціонал порівняно з ком'юніті-агрегаторами. - Інтерфейс може бути перевантаженим технічною інформацією і менш зручним для простих операцій колекціонування. - Відсутні інструменти для інвентаризації фізичних екземплярів.	Підтверджує нішу: новий сайт має взяти технічну базу з цих бібліотек, але накласти на неї простий UI для управління колекцією та інтелектуальні сервіси (пошук, рекомендації), яких тут бракує.
Проблеми для веб-скрейпінгу	Часто мають чітку, зрозумілу структуру для парсингу (що є перевагою), але активно впроваджують захист від надмірного навантаження (обмеження запитів, блокування IP).	Вимагає ретельної побудови скрейпера з повагою до ресурсу: обов'язкові затримки між запитами (time.sleep), обробка HTTP-помилки (429 Too Many Requests), можливе використання ротації користувачьких агентів. Ідеально – використання офіційного API, якщо він існує і дозволяє отримати необхідні дані.

Таблиця 1.3 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі категорії "Універсальні платформи та інструменти")[23, 24, 25, 26, 27]

Критерій аналізу	Характеристика категорії "Універсальні платформи для колекціонування"	Наслідки для розробки власного рішення
Представники	Notion, Airtable, Google Таблиці; спеціалізовані додатки (наприклад, CLZ Comics).	Конкуренція з гнучкими інструментами загального призначення та вузькоспеціалізованими комерційними рішеннями.
Основна суть (ціль)	Надання інструментів для створення власної структурованої бази даних будь-якого об'єкта колекціонування, від загальних (таблиці) до спеціалізованих (CLZ Comics).	Підтверджує, що існує потреба у гнучкому каталозі. Власний сайт повинен пропонувати аналогічну гнучкість в рамках домену манги, але з доданою автоматизацією.
Ключовий функціонал	- Максимально гнучка схема даних: користувач сам визначає поля та їх типи. - Можливість деталізації для фізичних екземплярів (ISBN, стан, ціна, місце зберігання, фотографії). - Сортування, фільтрація, перегляд у вигляді карток або галереї. - У загальних інструментів (Notion) – додаткові можливості для ведення нотаток, створення зв'язків між базами.	Цільовий функціонал нового сайту має містити базові поля за замовчуванням (взяти з бібліотек) та розширювану користувацьку схему для додавання власних атрибутів. Інтерфейс управління колекцією повинен бути не гіршим.
Основні	Абсолютний контроль користувача над	Не можна цього ігнорувати. Для серйозних

Критерій аналізу	Характеристика категорії "Універсальні платформи для колекціонування"	Наслідки для розробки власного рішення
переваги	структурою даних і зовнішнім виглядом. Можливість створення деталізованого опису саме фізичного/цифрового екземпляра. Відсутність обмежень на тематику колекції. Часто мають мобільні додатки та можливість спільної роботи.	колекціонерів ця гнучкість – ключова вимога. Новий продукт не може бути жорсткішим за звичайну таблицю.
Ключовий недолік	- Повна відсутність автоматизації наповнення. Кожен запис, кожен атрибут вноситься вручну. - Відсутність інтеграції з глобальними базами знань про мангу (немає автодоповнення назв, жанрів, авторів). - Немає інтелектуальних сервісів: семантичного пошуку, рекомендацій на основі контенту колекції, статистики на основі жанрів/авторів.	Визначає головну конкурентну перевагу нового сайту. Розробка має усунути ці недоліки, поєднуючи гнучкість каталогізації з потужною автоматизацією (скрейпінг для автозаповнення) та інтелектуальними функціями (пошук, рекомендації).
Проблеми для веб-скрейпінгу	Веб-скрейпінг не є проблемою для цих платформ, оскільки вони не є джерелом даних про мангу. Навпаки, вони ілюструють потребу в експорті/імпорті даних (наприклад, у CSV).	Архітектура власного сайту обов'язково повинна включати зручні механізми експорту даних у відкритих форматах (JSON, CSV). Це критично важливо для довіри користувачів, щоб вони не боялись "закриття" своїх колекцій у новій системі.

Таблиця 1.4 - Порівняльний аналіз існуючих категорій веб-систем для колекціонування контенту (на прикладі аналогічних платформ для інших ніш) [23, 24, 25, 26, 27]

Критерій аналізу	Характеристика платформ-аналогів (Discogs, Goodreads, Letterboxd)	Наслідки для концепції веб-сайту для колекціонування манги
Представники	Discogs (музичні релізи), Goodreads (книги), Letterboxd (фільми).	Надають готову успішну архітектурну модель для наслідування та адаптації в ніші колекціонування манги.
Концептуальна суть	Гібридні платформи, що поєднують три основи: глобальну краудсорсингову базу знань про медіа-об'єкт, персональний каталог колекціонера та активну соціальну мережу ентузіастів.	Формулюють чітку мету розробки: створити аналогічний гібрид, але орієнтований виключно на мангу.
Ключові принципи (аналіз для контексту манги)	Принцип Discogs (детальне колекціонування): - Чітке розділення поняття артефакту (альбом, книга) та фізичного екземпляра (видання, реліз). - Можливість детально описати свій екземпляр (стан, історія володіння, особливі ознаки). - Краудсорсинг для каталогізації всіх існуючих релізів. Принцип Goodreads/Letterboxd (соціальна взаємодія та відкриття): - Легкість додавання об'єктів до колекції з єдиної бази.	Пряме керівництво для проектування архітектури: - Реалізувати модель "Твір/Екземпляр" як у Discogs, де "твір" – це дані з MangaDex, а "екземпляр" – опис книги на полиці. - Інтегрувати соціальні та рекомендаційні механізми як у Goodreads, але з урахуванням

Критерій аналізу	Характеристика платформ-аналогів (Discogs, Goodreads, Letterboxd)	Наслідки для концепції веб-сайту для колекціонування манги
	- Потужні соціальні інструменти: друзі, списки, рецензії, обговорення. - Персоналізовані рекомендації на основі оцінок та активності.	специфіки манги (жанри "сейнен", "сьонен", зв'язок з аніме).
Переваги та сильні сторони моделі	- Ефективний баланс між громадським і приватним інтересом. - Висока лояльність користувачів завдяки особистому внеску у колекцію та соціальним зв'язкам. - Платформа для відкриття нового: рекомендації та спільнота сприяють поглибленню хобі. - Зручність: інтуїтивні інтерфейси для швидких дій (додати, оцінити, написати рецензію).	Пряме керівництво до дії. Ці платформи задають зразок того, яким має бути кінцевий продукт: об'єднання технічної бази даних (як у MangaDex) із зручністю та соціальністю ком'юніті-агрегаторів (як AniList) та гнучкістю універсальних інструментів
Недоліки та прогалини (з точки зору ніші манги)	- Не покривають специфіку манги: відсутні поля для трекінгу глав/томів, інформації про мангаку/артиста, журналі публікації, статусу сканувань. - Можлива надмірна складність моделі Discogs для новачків. - Їхні алгоритми не оптимізовані під структуру контенту манги (наприклад, не враховують відмінності між оригіналом та антологією).	Це не готове рішення, а ідеальна концептуальна модель. Завдання полягає в тому, щоб адаптувати їхню успішну архітектуру до домену манги, враховуючи його специфіку (наявність глав,

Критерій аналізу	Характеристика платформ-аналогів (Discogs, Goodreads, Letterboxd)	Наслідки для концепції веб-сайту для колекціонування манги
Висновок для веб-сайту колекціонування манги	Ці платформи є ідеальною концептуальною картою.	Головне завдання – втілити цю перевірену модель у новій предметній області. Це передбачає: (1) створення краудсорсингової бази творів манги, (2) розробку інструментів для детального опису особистих екземплярів, (3) впровадження механізмів соціальної взаємодії та рекомендацій, адаптованих до аудиторії шанувальників манги.

Висновки та виявлені ніші для нового рішення:

1. Прогалина між "що" і "який екземпляр": Існує розрив між платформами, що знають *все про твір* (MAL, MangaDex), та інструментами, що дозволяють описати *конкретний фізичний екземпляр* у колекції (Notion). Успішна система має інтегрувати обидва підходи.

2. Необхідність гібридної архітектури: Оптимальна система може використовувати веб-скрейпінг/API для автоматичного заповнення базової інформації (назва, автор, жанр) з існуючих джерел (MangaDex, AniList), а потім надавати користувачу розширені поля для деталізації свого екземпляра.

3. Персоналізація за межами ком'юніті-рейтингів: Рекомендаційна система може бути посилена за рахунок контентної фільтрації (на основі жанрів, описів, авторів з колекції користувача), що особливо ефективно для нішевих уподобань.

4. Відсутність спеціалізованого семантичного пошуку: В більшості систем пошук ведеться за точною назвою або популярними тегами. Семантичний пошук ("манга про переродження в слабого персонажа в ігровому світі" або "схожа на 'Berserk' за атмосферою") міг би стати унікальною перевагою нової платформи.

5. Проблема фрагментації даних: Колекціонер змушений використовувати кілька сервісів одночасно (MAL для трекінгу, MangaUpdates для інформації про випуски, таблицю для інвентаризації). Ідеальна система має консолідувати ці функції.

Таким чином, аналіз показує, що розробка нового веб-сайту має сенс не як чергового клону існуючих баз даних, а як гібридної, інтелектуальної платформи, що автоматизує наповнення за допомогою скрейпінгу, пропонує глибокі інструменти каталогізації фізичних/цифрових екземплярів та надає персоналізований пошук і рекомендації на основі оброблених метаданих.

1.3 Методи та алгоритми автоматизованого збору веб-даних

Автоматизований збір веб-даних (веб-скрейпінг) ґрунтується на кількох ключових методах і алгоритмах, вибір яких залежить від складності цільового сайту, обсягу даних і технічних обмежень.

Процес веб-скрейпінгу складається з кількох етапів [28, 29, 30]:

1. Отримання HTML-коду сторінки — здійснюється через HTTP-запити (наприклад, за допомогою бібліотеки Requests у Python).
2. Парсинг (розбір структури сторінки) — аналіз DOM-дерева з використанням бібліотек BeautifulSoup або lxml.
3. Виділення потрібних елементів — знаходження тегів, що містять цільові дані (назви, зображення, жанри тощо).
4. Структурування та збереження — конвертація у формат, придатний для подальшого зберігання (CSV, JSON, SQL).
5. Автоматизація — регулярне оновлення інформації через планувальники (наприклад, Cron або Celery).

Окремим напрямом розвитку є динамічний скрейпінг, який дозволяє отримувати дані з сайтів, що генерують контент за допомогою JavaScript. Для цього застосовуються інструменти Selenium, Playwright, Puppeteer, які емулюють поведінку браузера.

1. Основні методи збору даних [31, 32, 33]:

- Прямий парсинг HTML (Статичний контент):
 - Суть: Найпоширеніший метод. Скрипт відправляє HTTP-запит, отримує HTML-код сторінки та аналізує його структуру (дерево DOM) для вилучення даних за допомогою селекторів (XPath, CSS-селектори).
 - Інструменти: Бібліотеки BeautifulSoup (Python, проста), lxml (Python, швидка), Cheerio (Node.js).

- Алгоритмічна основа: Обхід дерева DOM, пошук елементів за атрибутами (id, class, data-*).
- Застосування: Ідеально для сайтів з простим, статичним HTML, де вся потрібна інформація присутня у вихідному коді відповіді сервера.
- Робота з динамічним контентом (JavaScript):
 - Суть: Сучасні сайти часто генерують контент за допомогою JavaScript після завантаження основного HTML. Для його отримання необхідно імітувати роботу браузера.
 - Підходи:
 1. Аналіз мережевих запитів (Network Inspection): У інструментах розробника (DevTools) знаходять AJAX/API-запити, що повертають дані у структурованому вигляді (JSON), та імітують їх.
 2. Використання керованого браузера: Автоматизація реального браузера, який виконує JS і рендерить сторінку.
 - Інструменти: Selenium, Playwright, Puppeteer. Вони дозволяють клікати, прокручувати та чекати завантаження елементів.
 - Алгоритмічна основа: Події та очікування (wait) в DOM, виконання JS-коду.
 - Застосування: Сайти з контентом, що підвантажується (infinite scroll), односторінкові додатки (SPA), сайти з важким захистом.

Використання публічних API:

- Суть: Найбільш етичний та стабільний метод, якщо сайт надає офіційний програмний інтерфейс.
- Алгоритмічна основа: Виконання структурованих HTTP-запитів (GET, POST) до ендпоінтів API, обробка відповідей у JSON/XML форматі, дотримання лімітів запитів (rate limiting) та аутентифікації (API keys, OAuth).

- Застосування: Соціальні мережі, великі платформи (AniList, MyAnimeList мають обмежені API), сервіси, що заохочують розробників.

2. Ключові алгоритмічні стратегії та покращення:

Обхід сайту (Crawling) та черги запитів:

- Алгоритми BFS/DFS: Використовуються для систематичного обходу всіх сторінок сайту, починаючи з початкового URL. Пошук у ширину (BFS) часто ефективніший для скрейпінгу.
- Черга URL: Зберігання посилань для майбутнього оброблення. Реалізується для управління порядком та уникнення дублікатів.

Обробка та очищення даних (Data Wrangling):

- Регулярні вирази (Regex): Для вилучення конкретних паттернів з тексту (дати, ціни, номери).
- Стандартизація та нормалізація: Алгоритми для приведення даних до єдиного формату (дати, валюти, регістр).
- Валідація та дедуплікація: Перевірка на коректність та видалення дублікатів на основі ключових полів.

Оптимізація та обхід захисту:

- Ротація користувацьких агентів (User-Agent) та проксі: Для імітації різних користувачів та уникнення блокування по IP-адресі.
- Розумні затримки (Randomized Delays): Алгоритми, що додають випадкові паузи між запитами (`time.sleep(random.uniform(1,5))`), імітуючи людську поведінку.
- Обробка CAPTCHA: Використання сервісів розпізнавання (Anti-Captcha) або, що краще, відмова від скрейпінгу сайтів із жорстким захистом на користь API.

Зберігання даних:

- Структуроване зберігання: Запис у реляційні (MySQL, PostgreSQL) або документо-орієнтовані (MongoDB) СУБД.

- Файлові формати: Збереження у JSON, CSV, XML для подальшого аналізу.

Висновок для проєкту веб-сайту для колекціонування манги:

Для ефективного та стабільного збору метаданих про мангу необхідно застосувати комбінований підхід:

1. Першочергово використовувати офіційні API (AniList, MyAnimeList), дотримуючись лімітів.

2. Для джерел без API (MangaUpdates, сайти видавництв) використовувати парсинг HTML з BeautifulSoup/lxml, а для динамічних елементів – Selenium/Playwright.

3. Впровадити алгоритми етичного скрейпінгу: обов'язкові випадкові затримки, ротацію User-Agent, ретельну обробку помилок HTTP (коди 429, 503).

4. Створити конвеєр обробки даних, що включає нормалізацію назв, об'єднання інформації з різних джерел та дедуплікацію для формування чистої та узгодженої власної бази даних.

Це забезпечить надійну технологічну основу для автоматичного наповнення каталогу веб-сайту для колекціонування.

Таким чином, розробка веб-системи, що самостійно збирає, структурує та оновлює інформацію про мангу за допомогою веб-скрейпінгу, є доцільною. Вона дає змогу отримувати найактуальніші відомості про нові релізи, авторів, жанри й доступність томів без залежності від сторонніх баз даних.

1.4. Огляд інструментів і бібліотек для веб-скрейпінгу (BeautifulSoup, Scrapy, Selenium, Playwright тощо)

У сучасних розробках найчастіше використовують такі технології [34,35]:

- BeautifulSoup (Python) — проста та гнучка бібліотека для аналізу HTML-структур, підтримує CSS-селектори та пошук за тегами.

- Scrapy — фреймворк для масштабованого скрейпінгу, що підтримує паралельне завантаження сторінок, збереження даних та інтеграцію з базами.
- Selenium — інструмент для автоматизації браузера, який дозволяє взаємодіяти з елементами сторінки та отримувати контент, що генерується JavaScript.
- Requests — бібліотека для виконання HTTP-запитів, базова складова більшості систем збору даних.

Перевагою Python є велика екосистема бібліотек і простота інтеграції з базами даних (MySQL, PostgreSQL, SQLite), API та веб-фреймворками (Flask, Django). Саме тому Python обрано як основну мову розробки у даному дослідженні.

Огляд інструментів і бібліотек для веб-скрейпінгу

1. BeautifulSoup (Python)

Призначення: Парсинг та розбір статичного HTML/XML.

Переваги:

- Простий та інтуїтивний синтаксис
- Ефективна робота з "брудним" HTML
- Широкий вибір парсерів (lxml, html.parser)

Недоліки:

- Не може виконувати JavaScript
- Відсутні вбудовані механізми для обходу сайтів
- Немає асинхронності "з коробки"

Застосування: Швидкий парсинг простих сайтів, обробка завантаженого HTML.

2. Scrapy (Python) [36,37]

Призначення: Повноцінний фреймворк для скрейпінгу.

Архітектура: Асинхронна, подієва.

Ключові можливості:

- Вбудовані середовища для обходу (Crawler)

- Мідлвари та конвеєри обробки даних
- Експорт у різні формати (JSON, CSV, XML)
- Автоматичне дотримання robots.txt

Недоліки:

- Складність налаштування для новачків
- Обмежена підтримка JavaScript

Застосування: Масштабні проекти скрейпінгу, створення павуків.

3. Selenium (Python, Java, C#, JavaScript тощо) [38,39]

Призначення: Автоматизація веб-браузера для роботи з динамічним контентом.

Переваги:

- Повна емуляція дій користувача
- Підтримка всіх сучасних браузерів
- Можливість очікування елементів (explicit waits)

Недоліки:

- Висока споживання ресурсів
- Повільна робота порівняно з прямими HTTP-запитами
- Складність налаштування в headless-режимі

Застосування: Сайти з важким JavaScript, тестування веб-додатків.

4. Playwright (Python, JavaScript, C#, Java)

Призначення: Сучасна альтернатива Selenium для автоматизації.

Ключові переваги:

- Автоматичне очікування елементів
- Вбудована підтримка запитів API
- Можливість запису дій (codegen)
- Краща продуктивність ніж Selenium

Недоліки:

- Молодша екосистема
- Менше готових рішень для обходу захисту

Застосування: Складні сайти з динамічним контентом, сучасні веб-додатки.

5. Puppeteer (JavaScript/Node.js)

Призначення: Контроль Chrome/Chromium через DevTools Protocol.

Переваги:

- Висока швидкість роботи
- Широкі можливості налагодження
- Підтримка headless-режиму

Недоліки:

- Працює лише з браузерами на базі Chromium
- Вимагає знання JavaScript/Node.js

Застосування: Скріншоти, PDF, тестування, скрейпінг SPA.

6. Requests + lxml (Python)

Призначення: Легкий та швидкий парсинг.

Переваги:

- Максимальна швидкість обробки
- Мінімальні вимоги до ресурсів
- Проста інтеграція з асинхронними бібліотеками

Недоліки:

- Відсутність вбудованих інструментів для складних сценаріїв

Застосування: Високопродуктивний парсинг статичних сайтів.

7. Асинхронні бібліотеки (aiohttp, httpx + asyncio)

Призначення: Паралельний скрейпінг.

Переваги:

Обробка тисяч запитів одночасно

Економія часу при скрейпінгу великих обсягів

Ефективне використання ресурсів

Недоліки:

- Складність відлагодження
- Можливість отримати бан за надто інтенсивні запити

Порівняльна інструментів для колекціонування манги (Табл.1.5)

Таблиця 1.5 - Порівняльна таблиця для проекту колекціонування манги [37, 39]

Інструмент	Підхід	Підтримка JS	Складність	Швидкість	Рекомендація для проекту
BeautifulSoup	Парсинг	Ні	Низька	Висока	Для статичних частин сайтів видавництв
Scrapy	Фреймворк	Обмежена	Середня	Дуже висока	Для масштабного збору з MangaUpdates
Selenium	Браузер	Повна	Висока	Низька	Для динамічних сайтів з захистом
Playwright	Браузер	Повна	Середня	Середня	Кращий вибір для сучасних SPA
Requests+lxml	Парсинг	Ні	Низька	Максимальна	Для швидкого збору API-даних

Висновок для проекту веб-сайту для колекціонування манги:

Для оптимального рішення рекомендується комбінований підхід:

1. Основна платформа: Scrapy – для структурованого збору з сайтів бібліотек (MangaDex, MangaUpdates), оскільки дозволяє ефективно керувати запитами, чергами та обробкою даних.
2. Динамічний контент: Playwright – для сайтів з JavaScript-рендерингом, де потрібна емуляція дій користувача (прокручування, кліки).
3. Простий парсинг: BeautifulSoup/lxml – для додаткової обробки вже завантажених сторінок або простих джерел.
4. Асинхронність: aiohttp для паралельного збору даних з декількох джерел одночасно, що критично для швидкого оновлення каталогу.
5. API-інтеграція: Requests для роботи з офіційними API (AniList, MyAnimeList), що є найбільш етичним та стабільним методом.

Така комбінація дозволить ефективно збирати дані з різних типів джерел, зберігаючи баланс між продуктивністю, стабільністю та дотриманням етичних норм скрейпінгу.

1.5. Аналіз вимог користувачів і функціональних характеристик системи

Основними потребами користувача системи колекціонування манги є: зручний перегляд і пошук інформації за назвою, автором, жанром; можливість формувати власну колекцію та відмічати прочитані томи; отримання актуальних даних про нові випуски; доступ до метаданих (опис, рік видання, статус серії, обкладинка).[40, 41, 42, 43]

Аналіз вимог користувачів і функціональних характеристик системи надано у табл.1.6

Таблиця 1.6 - Категорії користувачів та їхні вимоги [44,45]

Категорія користувача	Основні потреби	Специфічні вимоги
Колекціонер-початківець	Просте ведення списку прочитаного	• Інтуїтивний інтерфейс • Швидке додавання манги • Базові рекомендації • Мобільна версія
Серйозний колекціонер	Детальний каталог фізичних томів	• Розширені поля (ISBN, стан, ціна) • Імпорт/експорт даних • Статистика колекції • Персоналізована каталогізація
Дослідник/ентузіаст	Глибокий аналіз творів	• Повна інформація про авторів, історію публікацій • Зв'язки між творами • Пошук за складними критеріями • Анотації та нотатки
Спільнота (користувачі соціальних функцій)	Соціальна взаємодія	• Рецензії та оцінки • Списки та рекомендації • Дискусії • Відстеження друзів

Функціональні вимоги до системи надані у табл.1.7

Таблиця 1.7 – Функціональні вимоги до системи [44,45]

№	Функціональний модуль	Функціональні вимоги	Опис реалізації
1	Ядро системи (Core Functionality)	Автоматизоване наповнення каталогу	Забезпечення автоматичного збору та оновлення даних про MANGA шляхом інтеграції з API сторонніх сервісів (AniList, MyAnimeList) та застосування веб-скрейпінгу з відкритих джерел (MangaDex, MangaUpdates).
2	Ядро системи	Автоматичне оновлення контенту	Реалізація механізмів регулярної перевірки та оновлення інформації про нові томи та глави без участі користувача.
3	Ядро системи	Управління колекцією користувача	Надання можливостей додавання, редагування та видалення записів MANGA з підтримкою поділу на сутності «Твір» та «Екземпляр» відповідно до Discogs-моделі.
4	Ядро системи	Персоналізація колекції	Підтримка користувацьких полів, тегів та ведення обліку прогресу читання (прочитані томи та глави).
5	Інтелектуальний пошук та фільтрація	Семантичний пошук	Реалізація пошуку за текстовими описами та синопсисами з використанням семантичного аналізу та розширення пошукових запитів синонімами й пов'язаними поняттями.
6	Інтелектуальний пошук та фільтрація	Розширена фільтрація	Забезпечення фільтрації за жанрами, авторами, роками видання та статусом із підтримкою фасетної навігації та комбінування критеріїв.
7	Інтелектуальний пошук та фільтрація	Персоналізоване ранжування	Пріоритезація результатів пошуку на основі профілю користувача та історії його взаємодії з контентом.
8	Рекомендаційна система	Гібридні рекомендації	Реалізація рекомендацій на основі поєднання контентно-орієнтованої та колаборативної фільтрації з урахуванням жанрів, авторів і оцінок спільноти.

Нефункціональні вимоги до системи надані у табл.1.8

Таблиця 1.8 – Нефункціональні вимоги до системи [44,45]

№	Категорія вимог	Нефункціональні вимоги	Критерії та показники
1	Продуктивність	Час відгуку пошукових запитів	Забезпечення середнього часу відгуку системи під час виконання пошукових операцій не більше 2 секунд.
2	Продуктивність	Підтримка одночасних користувачів	Коректна робота системи за умови одночасної взаємодії до 1000 активних користувачів без суттєвого зниження продуктивності.
3	Продуктивність	Швидкість завантаження сторінок	Час повного завантаження основних сторінок веб-сайту не перевищує 3 секунд за стандартних умов доступу до мережі.
4	Масштабованість	Зростання обсягу даних	Підтримка розширення бази даних до обсягу не менше 1 млн записів без необхідності зміни архітектури системи.
5	Масштабованість	Горизонтальне масштабування	Можливість розподілу навантаження між кількома серверними вузлами з використанням механізмів балансування.
6	Масштабованість	Кешування даних	Застосування механізмів кешування для популярних запитів з метою зменшення навантаження на сервер та прискорення доступу до даних.
7	Надійність та безпека	Доступність сервісу	Забезпечення рівня доступності системи не нижче 99,5 % відповідно до вимог SLA.
8	Надійність та безпека	Захист від атак	Реалізація механізмів захисту від DDoS-атак та несанкціонованого доступу.
9	Надійність та безпека	Захист персональних даних	Шифрування даних користувачів під час зберігання та передавання відповідно до сучасних криптографічних стандартів.
10	Надійність та безпека	Резервне копіювання	Регулярне автоматизоване резервне копіювання даних із можливістю відновлення у разі збоїв.

Сценарії використання надані у табл.1.9. Пріоритети реалізації функціональності надані у Табл. 1.10

Таблиця 1.9 – Сценарії використання системи [44,45]

№	Назва сценарію	Учасники	Послідовність дій	Результат виконання	Пріоритет
1	Додавання нової MANGA до колекції	Користувач, Система	1) Користувач вводить назву твору у пошуковий рядок. 2) Система пропонує релевантні варіанти з бази даних. 3) Користувач обирає необхідний твір. 4) Система автоматично заповнює базові метадані. 5) Користувач додає інформацію про власний екземпляр. 6) Дані зберігаються у персональній колекції.	Твір успішно додано до колекції користувача з повним набором атрибутів.	Високий
2	Пошук персоналізованих рекомендацій	Користувач, Рекомендаційний модуль	1) Система аналізує колекцію та оцінки користувача. 2) Формується персоналізований список рекомендацій. 3) Система відображає пояснення рекомендацій («рекомендовано на основі...») 4) Користувач може оцінити запропоновані твори.	Користувач отримує релевантні рекомендації та впливає на подальше навчання системи.	Середній
3	Соціальна взаємодія користувачів	Користувач, Інші користувачі, Система	1) Користувач переглядає профіль іншого колекціонера. 2) Оформлює підписку або додає до списку контактів. 3) Коментує рецензії або списки колекцій.	Розширення соціальної взаємодії та обміну рекомендаціями між користувачами.	Середній

Таблиця 1.10 - Пріоритети реалізації функціональності [44,45]

Рівень пріоритету	Функціональні можливості
Високий	Базові CRUD-операції для управління колекцією; автоматичне заповнення даних; семантичний пошук; підтримка мобільного інтерфейсу.
Середній	Рекомендаційна система; соціальні функції; розширена статистика користувача; API для сторонніх розробників.
Низький	AI-генеровані анотації; інтеграція з маркетплейсами; розширені аналітичні інструменти.

Одже система повинна задовольняти потреби різних категорій користувачів: від початківців, яким потрібна простота, до серйозних колекціонерів, які потребують детальної каталогізації. Ключові аспекти:

1. Автоматизація – мінімізація ручної роботи через інтеграцію з існуючими джерелами
2. Гнучкість – можливість адаптувати систему під індивідуальні потреби
3. Інтелектуальність – розумний пошук та персоналізація
4. Спільнота – функції для соціальної взаємодії

Система має балансувати між потужними технічними можливостями та зручним інтерфейсом, що робить її доступною для всіх категорій користувачів.

Таким чином у цьому розділі проаналізовано сутність веб-скрейпінгу, його роль у сучасних інформаційних технологіях і можливості застосування для створення веб-систем колекціонування даних. Проведено огляд існуючих аналогів та інструментів, визначено основні вимоги до майбутньої системи. На основі аналізу зроблено висновок, що розробка веб-системи для колекціонування манги з використанням технологій веб-скрейпінгу є актуальною задачею, яка поєднує практичну користь для кінцевого користувача з дослідницькою цінністю у сфері комп'ютерних наук. Подальші етапи дослідження будуть спрямовані на проектування архітектури системи, реалізацію програмних модулів і тестування ефективності їхньої роботи.

На основі виконаного аналізу застосування інтелектуальних методів пошуку та фільтрації контенту у веб-сайтах для колекціонування манги з використанням технологій веб-скрейпінгу, індексації та обробки метаданих можна зробити наступну постановку проблеми.

Постановка проблеми:

У веб-сайтах для колекціонування манги ключовою проблемою є ефективний пошук і фільтрація релевантного контенту в умовах великих, динамічних і гетерогенних масивів даних. Інформація про мангу надходить із різних джерел (каталоги, онлайн-платформи, спільноти), відрізняється за структурою, повнотою та якістю метаданих, а також постійно оновлюється. Традиційні пошукові механізми, що ґрунтуються на простому співставленні ключових слів, не забезпечують необхідного рівня точності, повноти та персоналізації результатів. Це зумовлює необхідність розробки архітектури системи, яка поєднує веб-скрейпінг, індексацію та інтелектуальну обробку метаданих для підвищення ефективності пошуку і фільтрації контенту.

2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ, КЛАСИФІКАЦІЇ ТА РЕКОМЕНДАЦІЇ МАНґА-КОНТЕНТУ НА ОСНОВІ ЗІБРАНИХ ДАНИХ ТА АНАЛІЗ ЇХ ВПЛИВУ НА ЗРУЧНІСТЬ КОРИСТУВАННЯ ВЕБ-САЙТОМ

2.1 Архітектура системи пошуку та фільтрації контенту у веб-сайтах для колекціонування манґи з використанням технологій веб-скрейпінгу, індексації та обробки метаданих можна зробити наступну постановку проблеми

Архітектура системи пошуку та фільтрації контенту веб-сайту для колекціонування манґи включає такі основні компоненти [46,47,48,49,50]:

1. Модуль збору даних (Web Scraping Module). Забезпечує автоматизований збір інформації про манґу з відкритих веб-ресурсів та API сторонніх сервісів. Модуль відповідає за отримання первинних даних, їх очищення та нормалізацію.

2. Модуль обробки та збагачення метаданих. Виконує структурування отриманих даних, уніфікацію жанрів, імен авторів, статусів видання, а також усунення дублікатів. За потреби здійснюється семантичне збагачення метаданих (додавання синонімів, пов'язаних термінів).

3. Сховище даних (Database Layer). Призначене для збереження структурованих даних про твори манґи, екземпляри, користувачів і їхні взаємодії. Сховище забезпечує цілісність даних і швидкий доступ до них.

4. Індексуєчий модуль (Indexing Engine). Формує пошукові індекси на основі текстових описів і метаданих манґи, що дозволяє прискорити виконання пошукових запитів і підтримувати розширені механізми фільтрації.

5. Модуль пошуку та фільтрації. Реалізує повнотекстовий і семантичний пошук, фасетну фільтрацію за жанрами, авторами, роками видання, статусом та іншими атрибутами, а також комбінування кількох критеріїв.

6. Інтерфейс користувача (User Interface). Забезпечує взаємодію користувача із системою, формування пошукових запитів, застосування фільтрів і відображення результатів у зручному вигляді. (Рис.2.1)

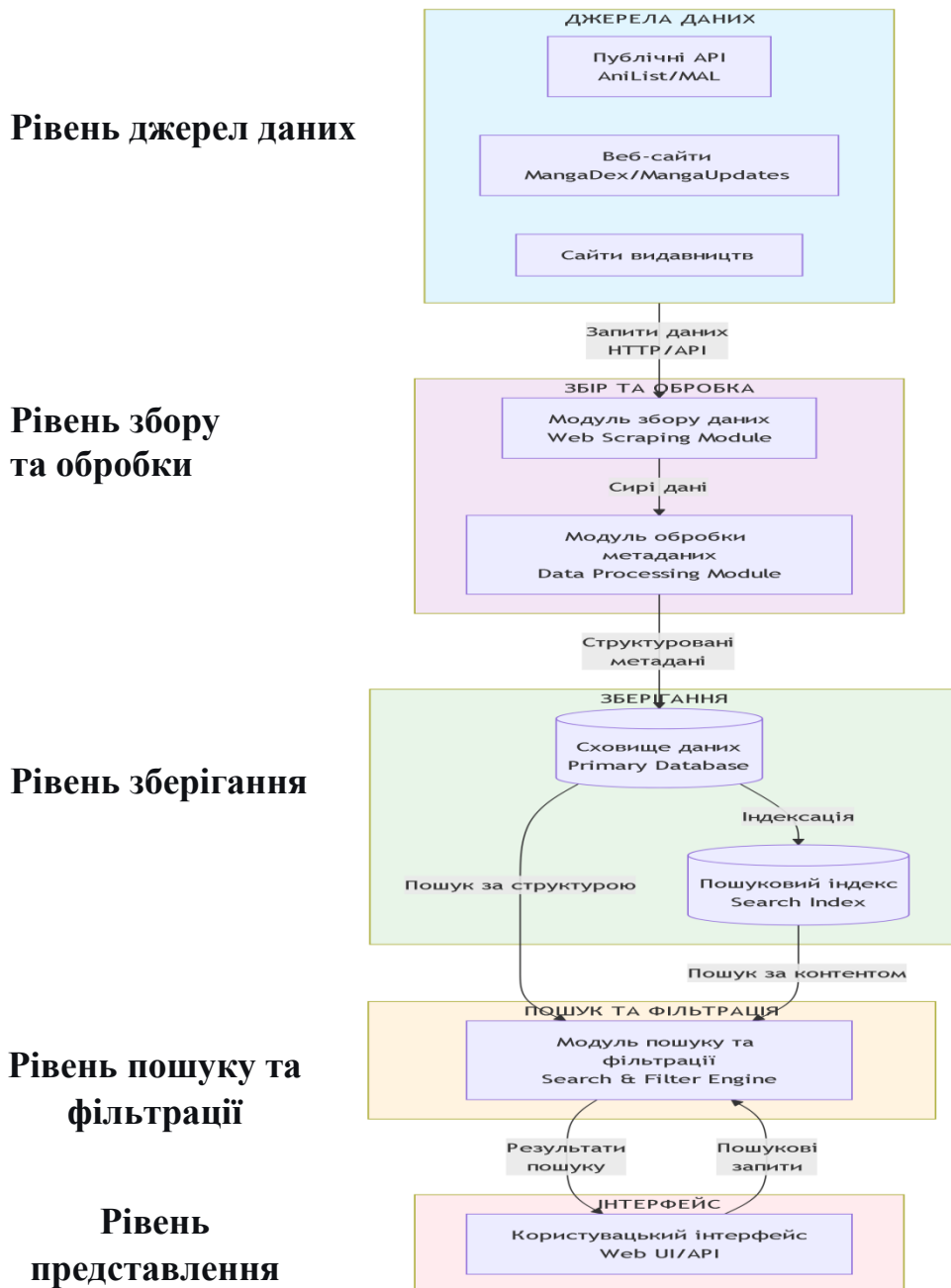


Рисунок 2.1 - Архітектура системи пошуку та фільтрації контенту веб-сайту для колекціонування манги [53]

Пояснення архітектурних шарів:

2.1.1 Рівень джерел даних відображає: Публічні API (AniList, MyAnimeList)

- структуровані джерела; Веб-сайти бібліотек (MangaDex, MangaUpdates) - основні джерела скрейпінгу; Сайти видавництв - додаткова інформація про фізичні видання. В таблиці 2.1 надано категорії джерел та їх характеристика.

Таблиця 2.1 - Категорії джерел та їх характеристика [50,52]

Тип джерела	Представники	Формат даних	Частота оновлення	Якість даних	Тип доступу
Офіційні API	AniList GraphQL API, MyAnimeList REST API	JSON, GraphQL	Редійне (ручне)	Висока, структурована	Авторизація + токени, rate limiting
Спеціалізовані бібліотеки	MangaDex API, MangaUpdates	JSON (API), HTML (сайт)	Денне	Дуже висока, професійна	API з обмеженнями, скрейпінг HTML
Краудсорсингові бази	Fandom/Manga Wiki, Бази фанів	HTML, MediaWiki API	Постійне	Змінна, потребує валідації	Прямий скрейпінг, обмежений API
Офіційні сайти	Сайти видавців (Shueisha, Kodansha), Amazon, Bookwalker	HTML, мікроформат	Редійне	Обмежена, комерційна	Скрейпінг, часто з захистом

Рівень джерел даних є критично важливим фундаментом всієї системи. Ефективна архітектура цього рівня повинна:

- Забезпечувати надійність через резервування та відмовостійкість

- Оптимізувати якість через пріоритизацію та валідацію
- Дотримуватися етичних норм через повагу до rate limits та умов використання
- Підтримувати актуальність через розумну стратегію оновлення
- Забезпечувати масштабованість через модульну архітектуру джерел

Цей рівень безпосередньо впливає на якість всіх наступних компонентів системи, тому його проектуванню має бути приділено особливу увагу.

2.1.2 Рівень збору та обробки. Рівень збору та обробки трансформує сирі, неструктуровані дані з різноманітних джерел у чисту, стандартизовану, збагачену інформацію, готову для зберігання та пошуку. Кожен модуль виконує специфічну роль у конвеєрі обробки, забезпечуючи високу якість кінцевих даних при мінімальному втручанні людини. Характеристики модулі рівня збору та обробки надані в таблиці 2.2.

Таблиця 2.2 - Модулі рівня збору та обробки [51, 52]

Назва блоку	Призначення	Ключові функції	Технології/інструменти
1. Модуль контролю джерел (Source Manager)	Централізоване управління та оркестрація всіх джерел даних	• Планування запитів до джерел • Балансування навантаження • Обробка черги джерел з пріоритетами • Моніторинг доступності джерел • Автоматичне переключення на резервні джерела	Celery, Redis Queue, APScheduler
2. Модуль HTTP-клієнта (HTTP Client Layer)	Виконання мережових запитів з дотриманням політик	• Обробка HTTP(S) запитів • Ротація User-Agent та проксі • Дотримання затримок (rate limiting) • Обробка помилок та повторні спроби • Кешування відповідей	aiohttp, requests, httpx, Scrapy Downloader
3. Адаптери джерел (Source Adapters)	Конвертація специфічних форматів джерел у єдиний внутрішній формат	• Парсинг HTML (BeautifulSoup, lxml) • Обробка JSON/XML відповідей API • Рендеринг JavaScript (Selenium/Playwright) • Обробка GraphQL запитів • Нормалізація структури даних	BeautifulSoup, lxml, Selenium, Playwright, GraphQL clients

Продовження Таблиця 2.2

Назва блоку	Призначення	Ключові функції	Технології/інструменти
		джерела	
4. Модуль валідації даних (Data Validator)	Перевірка коректності та повноти отриманих даних	• Перевірка обов'язкових полів • Валідація форматів (дати, URL, ISBN) • Виявлення аномальних значень • Контроль якості тексту (спам, неповні описи) • Перевірка унікальності записів	JSON Schema, Cerberus, власні валідатори
5. Модуль обробки текстів (Text Processing Unit)	Попередня обробка текстових даних для подальшого використання	• Токенізація та лематизація • Видалення стоп-слів • Виявлення мови контенту • Нормалізація текстів (регістр, спецсимволи) • Вилучення ключових слів та сутностей	spaCy, NLTK, py morphology2 (українська), Stanza
6. Модуль збагачення метаданих (Metadata Enricher)	Додавання додаткової цінності до базових	• Автоматичне призначення жанрів • Виявлення пов'язаних творів • Додавання синонімів назв	ML моделі, правила, зовнішні словники

Назва блоку	Призначення	Ключові функції	Технології/інструменти
	метаданих	• Класифікація за демографічними групами • Оцінка популярності на основі зовнішніх джерел	
7. Модуль дедуплікації (Deduplication Engine)	Виявлення та обробка дублікатів записів	• Порівняння за схожістю назв • Пошук дублікатів за ISBN/ASIN • Злиття інформації з кількох записів • Вибір канонічного запису • Оновлення посилань на об'єднані записи	FuzzyWuzzy, власні алгоритми схожості, TF-IDF
8. Модуль нормалізації (Normalization Module)	Приведення даних до єдиних стандартів	• Уніфікація назв авторів та мангак • Стандартизація назв жанрів та тегів • Конвертація дат у єдиний формат • Нормалізація оцінок (шкала 1-10, 1-5) • Приведення мов перекладу до ISO кодів	Регулярні вирази, таблиці відповідностей
9. Модуль моніторингу & логів (Monitoring & Logging)	Відстеження роботи всього рівня збору даних	• Збір метрик продуктивності • Логування помилок та попереджень • Генерація звітів про якість даних	Prometheus, Grafana, ELK Stack, Sentry

Продовження Таблиця 2.2

Назва блоку	Призначення	Ключові функції	Технології/інструменти
		• Тригери для адміністратора • Аналіз трендів збору даних	
10. Модуль зберігання проміжних даних (Intermediate Storage)	Тимчасове зберігання даних під час обробки	• Кешування сирих даних • Черга оброблених записів • Зберігання об'єктів обробки (вектори, токени) • Архівація історичних даних • Буферизація перед записом в основну БД	Redis, Disk-cache, Temporary databases

Ключові характеристики кожного блоку:

1. Модуль контролю джерел:

- *Поведінка*: Активний, ініціює запити за розкладом
- *Критичність*: Висока (без нього збір не відбувається)
- *Складність обробки*: Середня
- *Частота роботи*: Постійна

2. Адаптери джерел:

- *Поведінка*: Пасивний, реагує на отримані дані
- *Специфіка*: Кожне джерело має власний адаптер
- *Гнучкість*: Можна додавати нові адаптери без змін архітектури

3. Модуль валідації:

- *Тип валідації*: Синтаксична (формат) та семантична (зміст)
- *Реакція на помилки*: Відхилення запису, спроба виправлення, логування
- *Налаштовуваність*: Правила валідації можна змінювати динамічно

4. Модуль дедуплікації:

- *Алгоритми*: Точне співпадіння, нечітка логіка, машинне навчання
- *Пріоритети*: Сохранення найповнішого запису, найсвіжіших даних
- *Вплив на продуктивність*: Високий (потрібна оптимізація)

5. Модуль збагачення:

- *Джерела збагачення*: Внутрішні бази знань, зовнішні API, ML моделі
- *Типи доданої інформації*: Класифікація, рекомендації, анотації
- *Оновлення моделей*: Періодичне перетренування на нових даних

Процесний потік обробки даних надано на Рис.2.2:

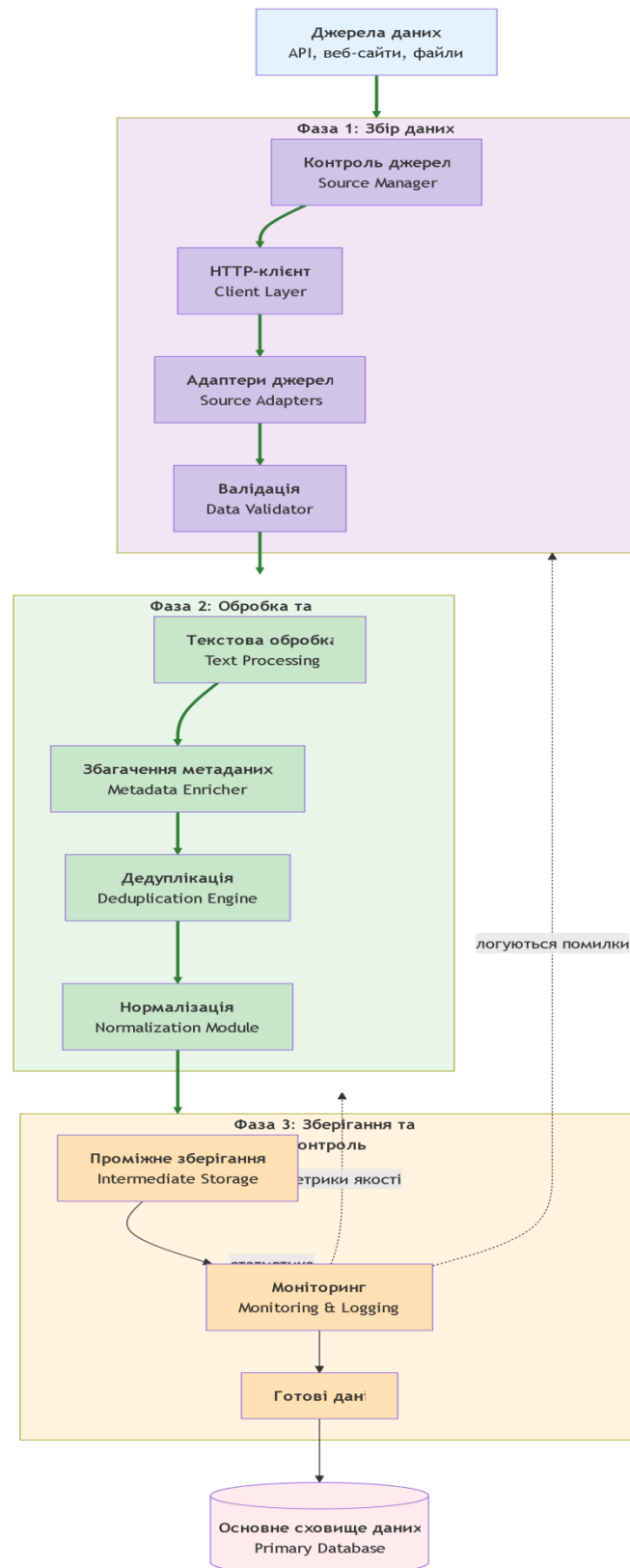


Рисунок 2.2 - Процесний потік обробки даних[53]

Пояснення до діаграми:

Фаза 1: Збір даних (фіолетова зона)

- Контроль джерел: Визначає пріоритетність та черговість запитів
- HTTP-клієнт: Виконує мережеві запити з дотриманням правил
- Адаптери: Конвертують специфічні формати джерел у єдиний
- Валідація: Базова перевірка коректності отриманих даних

Фаза 2: Обробка та збагачення (зелена зона)

- Текстова обробка: Токенізація, лематизація, очищення текстів
- Збагачення: Додавання додаткової інформації (жанри, зв'язки)
- Дедуплікація: Виявлення та об'єднання дублікатів
- Нормалізація: Приведення до єдиних стандартів

Фаза 3: Зберігання та контроль (жовта зона)

- Проміжне зберігання: Тимчасове зберігання оброблених даних
- Моніторинг: Відстеження якості та продуктивності процесу
- Готові дані: Підготовлена інформація для фінального збереження

Кінцева точка:

- Основне сховище: База даних для зберігання готових, оброблених записів

Особливості потоку:

1. Лінійна послідовність: Дані проходять через всі етапи послідовно
2. Моніторинг по всіх фазах: Система контролю охоплює всі етапи обробки
3. Чіткі межі фаз: Кожна фаза має конкретне призначення
4. Критичний шлях: Показаний товстими стрілками - основний маршрут обробки

Цей процес гарантує, що дані з різних джерел будуть приведені до єдиного стандарту, очищені від помилок та збагачені додатковою інформацією перед збереженням у основну базу даних.

Вимоги до якості обробки: Повнота: Не менше 95% обов'язкових полів мають бути заповнені; Точність: Менше 2% помилок у нормалізованих даних; Консистентність: Однакові стандарти для всіх записів; Актуальність: Дані не

старші 7 днів для динамічної інформації; Ефективність: Обробка одного запису не більше 3 секунд

2.1.3 Рівень зберігання. Рівень зберігання являє собою поліглот-архітектуру, де кожен тип даних зберігається в оптимальному для нього сховищі. Такий підхід дозволяє: Максимізувати продуктивність для кожного типу операцій; Забезпечити високу доступність та масштабованість; Ефективно використовувати ресурси зберігання; Забезпечити гнучкість для майбутнього розширення функціоналу

Ключовим є правильна оркестрація взаємодії між різними сховищами та підтримка консистентності даних у розподіленій системі. Характеристика модулів рівня зберігання надано в табл.2.3. Архітектурна схема взаємодії Табл. 2.4

Таблиця 2.3 - Модулі рівня зберігання [50,51,52]

Назва блоку	Призначення	Ключові функції	Технології/інструменти
1. Основне сховище (Primary Database)	Довгострокове структуроване зберігання всіх даних системи	• Зберігання нормалізованих даних про мангу • Управління зв'язками між сутностями • Транзакційна цілісність даних • ACID-гарантії • Резервне копіювання та відновлення	PostgreSQL, MySQL, SQL Server
2. Документне сховище (Document Store)	Зберігання напівструктурованих користувацьких даних та конфігурацій	• Зберігання профілів користувачів • Користувацькі колекції та налаштування • Гнучкі схеми для динамічних даних • Швидкий доступ до документів за ID • Підтримка вкладених структур	MongoDB, Elasticsearch, CouchDB
3. Пошуковий індекс (Search Index)	Оптимізоване сховище для швидкого повнотекстового пошуку	• Інвертований індекс текстових полів • Морфологічний аналіз текстів • Семантичне індексування векторних представлень • Релевантне ранжування результатів • Фасетна фільтрація	Elasticsearch, Apache Solr, OpenSearch
4. Кеш-шар (Cache Layer)	Прискорення доступу до часто використовуваних	• Кешування результатів пошуку • Сесії користувачів • Проміжні результати обчислень	Redis, Memcached, Hazelcast

Назва блоку	Призначення	Ключові функції	Технології/інструменти
	даних	• Рейтинги та популярні запити • Тимчасові дані з обмеженим TTL	
5. Сховище векторів (Vector Store)	Оптимізоване зберігання векторних представлень для семантичного пошуку	• Зберігання embedding-векторів описів манги • Індекс для пошуку найближчих сусідів • Метадані для векторів • Підтримка різних моделей embedding • Ефективний пошук за схожістю	Weaviate, Pinecone, FAISS, Qdrant
6. Сховище файлів та медіа (File Storage)	Зберігання бінарних даних та медіафайлів	• Зберігання обкладинок манги • Скриншоти та ілюстрації • Користувацькі аватари • Експортовані файли (CSV, JSON) • Резервні копії баз даних	Amazon S3, MinIO, Google Cloud Storage, Azure Blob
7. Архівне сховище (Archive Storage)	Довгострокове архівування історичних та рідко використовуваних даних	• Архівація старих версій записів • Логи подій старше певного терміну • Історичні статистичні дані • Резервні копії для аналітики • Холодне зберігання з низькою вартістю	Amazon Glacier, Tape Storage, холодні обчислювальні сховища
8. Сховище часових рядів (Time-Series)	Оптимізоване зберігання даних,	• Статистика використання системи • Моніторинг продуктивності	InfluxDB, TimescaleDB,

Назва блоку	Призначення	Ключові функції	Технології/інструменти
Database)	залежних від часу	• Тренди популярності манги • Користувачька активність за час • Аналітика запитів пошуку	Prometheus
9. Графова база даних (Graph Database)	Ефективне зберігання складних зв'язків між сутностями	• Зв'язки між мангою, авторами, жанрами • Рекомендаційні мережі • Соціальні зв'язки між користувачами • Аналіз шляхів та залежностей • Пошук опосередкованих зв'язків	Neo4j, Amazon Neptune, JanusGraph
10. Система реплікації та шардінг (Replication System)	Забезпечення доступності та масштабованості сховищ	• Мастер-слейв реплікація • Шардінг за ключами • Автоматичний фейловер • Балансування навантаження • Гео-розподілене зберігання	Вбудовані механізми СУБД, ProxySQL, Vitess

Таблиця 2.4 - Архітектурна схема взаємодії:[51,52]

Рівень зберігання даних			
Основне сховище (PostgreSQL)	Документне сховище (MongoDB)	Пошуковий індекс (Elasticsearch)	Кеш-шар (Redis)
Векторне сховище (Weaviate)	Файлове сховище (S3)	Архівне сховище (Glacier)	Часові ряди (InfluxDB)
Графова база (Neo4j)	Реплікація та шардінг (Vitess)		

Детальні характеристики кожного блоку:

1. Основне сховище (PostgreSQL):

- *Модель даних:* Реляційна, нормалізована до 3NF
- *Основні таблиці:* manga, authors, genres, publishers, users
- *Індекси:* B-tree для первинних ключів, GIN для текстових полів
- *Розмір:* ~50-100GB для 1 млн записів манги
- *Бекап:* Щоденні повні + щогодинні інкрементальні

2. Пошуковий індекс (Elasticsearch):

- *Структура індексу:* Інвертований індекс для title, description, genres
- *Аналізатори:* Стандартний, стеммінг для української/англійської
- *Шарди:* 3 первинних + 2 репліки на індекс
- *Оновлення:* Near real-time (затримка 1 секунда)
- *Пошукові можливості:* Fuzzy, phrase, range, bool queries

3. Кеш-шар (Redis):

- *Типи даних:* Strings, Hashes, Sets, Sorted Sets
- *Стратегія витіснення:* LRU (Least Recently Used)
- *Термін життя:* Від 5 хвилин до 30 днів залежно від типу даних
- *Класифікація:* Redis Cluster для горизонтального масштабування
- *Персистентність:* RDB + AOF для комбінованої надійності

4. Сховище векторів (Weaviate):

- *Модель embedding*: sentence-transformers/all-MiniLM-L6-v2
- *Розмірність векторів*: 384 виміри
- *Метрика схожості*: Cosine similarity
- *Індекс*: HNSW (Hierarchical Navigable Small World)
- *Пошук*: k-NN пошук з параметром k=10

5. Система реплікації:

- *Тип реплікації*: Асинхронна master-slave
- *Фейловер*: Автоматичний при недоступності мастера > 30 секунд
- *Шардінг*: За ranges ID манги
- *Консистентність*: Eventual consistency для реплік
- *Моніторинг*: Перевірка лагу реплікації (< 100мс)

Взаємодія між компонентами:

1. Алгоритм написання даних надано на рис.2.3:

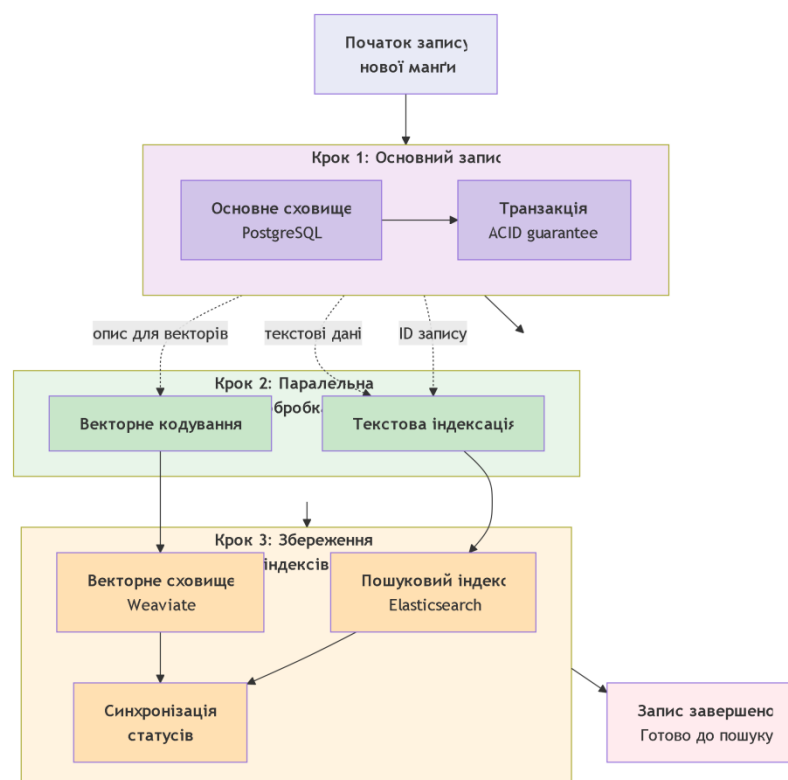


Рисунок 2.3 - Алгоритм написання даних [53]

2. Алгоритм читання даних надано на рис.2.4:

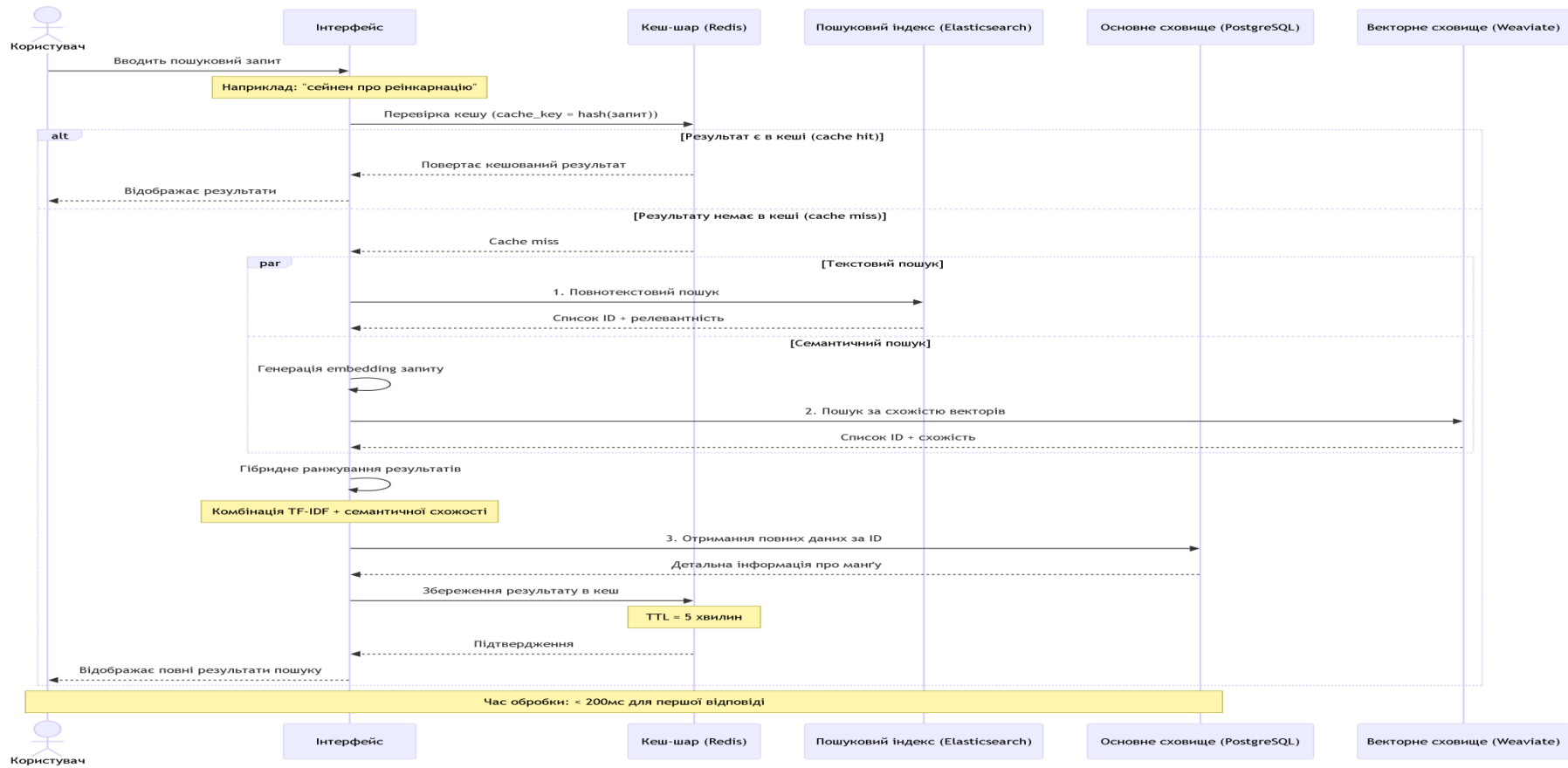


Рисунок 2.4- Алгоритм читання даних[53]

3. Синхронізація: Евентуальна консистентність між сховищами; Периодична синхронізація архівного сховища; Реал-тайм оновлення кешу.

4. Вимоги до продуктивності:

- Час відгуку: Кеш: < 10 мс; Пошуковий індекс: < 100 мс; Основне сховище: < 50 мс для простих запитів; Векторний пошук: < 200 мс
- Доступність: Основні сховища: 99.95% SLA; Кеш та індекси: 99.9% SLA; Архівні сховища: 99.5% SLA
- Масштабованість: Підтримка зростання даних на 50% щорічно; Можливість додавання нових шардів без downtime; Автоматичне масштабування за навантаженням

5. Стратегії резервування: Multi-AZ розгортання для критичних компонентів; Хрест-регіональна реплікація для відновлення після катастроф; Регулярні тести відновлення (раз на квартал); Моніторинг вільних ресурсів (диск, пам'ять, CPU)

2.1.4 Рівень пошуку та фільтрації. Рівень пошуку та фільтрації є найскладнішим компонентом системи, що поєднує в собі текстову обробку, машинне навчання, розподілені обчислення та кешування. Кожен модуль вирішує специфічну задачу у конвеєрі обробки запиту, а їхня ефективна взаємодія забезпечує: Високу релевантність результатів завдяки комбінації різних алгоритмів пошуку; Персоналізований досвід через адаптацію під інтереси користувача; Швидкість відгуку за рахунок паралелізму та кешування; Масштабованість через розподілену архітектуру; Система постійно вдосконалюється на основі аналітики поведінки користувачів та A/B тестування нових алгоритмів. Характеристика модулів рівня пошуку та фільтрації надана в Табл.2.5.

Таблиця 2.5 - Модулі рівня пошуку та фільтрації [50,52]

Назва блоку	Призначення	Ключові функції	Технології/інструменти
1. Query Parser (Парсер запитів)	Розбір та нормалізація вхідного пошукового запиту	• Токенізація пошукового рядка • Видалення стоп-слів та пунктуації • Розпізнавання мови запиту • Розділення на ключові слова та оператори • Виявлення спеціальних конструкцій (фрази, негативні умови)	ANTLR, регулярні вирази, власні парсери
2. Query Enricher (Збагачувач запитів)	Розширення пошукового запиту для покращення релевантності	• Додавання синонімів до ключових слів • Розширення жанрових термінів (сейнен → seinen) • Додавання пов'язаних понять • Корекція орфографічних помилок • Нормалізація імен авторів та персонажів	WordNet, власні словники синонімів, ML-моделі
3. Query Router (Маршрутизатор запитів)	Визначення оптимального шляху обробки запиту	• Виділення типів пошуку (текстовий, семантичний, гібридний) • Вибір індексів для пошуку • Визначення пріоритету паралельної обробки • Балансування навантаження між індексами • Фолбек стратегії при недоступності компонентів	Правила, машинне навчання, консистентний хешінг
4. Search Engine (Пошукова)	Виконання пошуку в різних індексах	• Повнотекстовий пошук в Elasticsearch • Семантичний пошук у векторному сховищі	Elasticsearch DSL, SQL, векторні

Назва блоку	Призначення	Ключові функції	Технології/інструменти
машина)		• Структурований пошук в SQL БД • Комбінація результатів з різних джерел • Паралельне виконання запитів	бібліотеки
5. Filter Engine (Двигун фільтрів)	Застосування фільтрів до результатів пошуку	• Фасетна фільтрація за жанрами, авторами, роками • Діапазонна фільтрація (рейтинг, кількість глав) • Булева фільтрація (AND/OR/NOT) • Каскадна фільтрація (залежні фільтри) • Оптимізація порядку застосування фільтрів	Elasticsearch aggregations, SQL WHERE, власні алгоритми
6. Ranking Engine (Ранжуючий двигун)	Сортування результатів за релевантністю	• TF-IDF ранжування • BM25 алгоритм • Семантична схожість (cosine similarity) • Персоналізоване ранжування • Гібридне комбінування рангів	Learning to Rank, RankLib, власні моделі
7. Personalization Engine (Персоналізаційний двигун)	Адаптація результатів під конкретного користувача	• Аналіз історії переглядів користувача • Врахування оцінок та вподобань • Соціальна фільтрація (що популярне серед схожих користувачів) • Контекстна персоналізація (час, пристрій, локація) • Експерименти A/B тестування	Колаборативна фільтрація, контентна фільтрація

Назва блоку	Призначення	Ключові функції	Технології/інструменти
8. Facet Generator (Генератор фасетів)	Динамічна побудова фільтрів на основі результатів	• Підрахунок доступних значень для кожного атрибуту • Побудова ієрархій фасетів (жанри → піджанри) • Управління порядком відображення фасетів • Виявлення та видалення конфліктуючих фільтрів • Кешування фасетів для швидкого доступу	Elasticsearch terms aggregation, Redis для кешу
9. Result Formatter (Форматер результатів)	Форматування результатів для відображення	• Пагінація результатів • Вибір полів для відображення • Об'єднання даних з різних джерел • Сортуювання за різними критеріями • Додавання метаданих результатів (загальна кількість, час виконання)	Шаблонізатори, JSON serializers
10. Cache Manager (Менеджер кешу)	Керування кешуванням пошукових результатів	• Генерація ключів кешу на основі запиту • Управління TTL для різних типів запитів • Інвалідація кешу при оновленні даних • Кешування проміжних результатів • Моніторинг hit/miss ratio	Redis, Memcached, дискові кеші
11. Analytics Collector (Збірник аналітики)	Збір метрик про пошукову активність	• Логування запитів користувачів • Трекінг кліків по результатах • Аналіз ефективності ранжування	Elasticsearch для логів, Kafka для подій, Prometheus

Назва блоку	Призначення	Ключові функції	Технології/інструменти
		• Виявлення триггерів для zero-results • Моніторинг продуктивності пошуку	
12. A/B Testing Engine (Двигун A/B тестування)	Тестування покращень алгоритмів пошуку	• Розподіл користувачів по групах • Застосування різних алгоритмів до різних груп • Збір метрик конверсії • Статистичний аналіз результатів • Автоматичне впровадження успішних змін	Firebase A/B Testing, власні рішення

Процесний потік пошуку надано на Рис.2.5

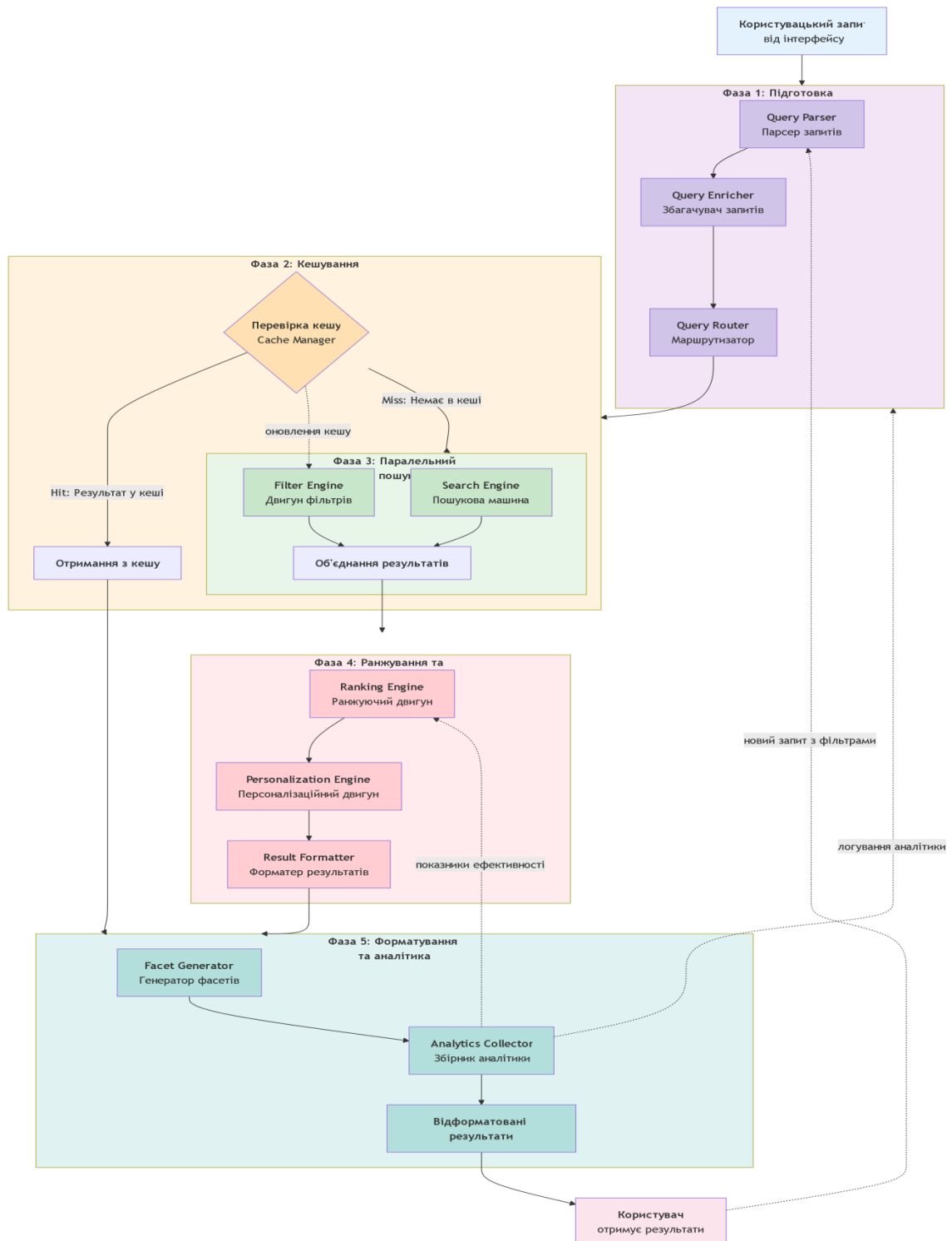


Рисунок 2.5 - Процесний потік пошуку [53]

Вимоги до продуктивності Процесного потоку пошуку надано в табл.2.6

Таблиця 2.6 - Вимоги до продуктивності Процесного потоку пошуку [51]:

Метрика	Цільове значення	Коментар
Час відгуку (p95)	< 200 мс	Для складних запитів з фільтрами
Cache hit rate	> 70%	Для популярних запитів
Точність пошуку (precision@10)	> 85%	Відсоток релевантних результатів
Повнота пошуку (recall)	> 90%	Відсоток знайдених релевантних документів
Пропускна здатність	1000 запитів/секунду	На одному вузлі

Моніторинг та аналітика:

1. Ключові метрики: Query latency (загальний та по компонентах); Cache hit/miss ratio; Zero-result rate; Click-through rate (CTR)
2. Аналітика користувачів: Найпопулярніші запити; Сесії з zero results; Конверсія пошуку (запит → клік → додавання в колекцію)
3. A/B тести: Порівняння алгоритмів ранжування; Тестування нових функцій фільтрації; Оптимізація параметрів пошуку

2.1.5 Рівень представлення. Рівень представлення є критичним інтерфейсом між користувачем та складною бекенд-системою. Він повинен: Забезпечувати зручність через інтуїтивний дизайн та адаптивність; Підвищувати продуктивність через оптимізацію та кешування; Забезпечувати доступність для всіх категорій користувачів; Надавати персоналізацію через гнучкі налаштування; Збирати аналітику для постійного вдосконалення

Сучасний підхід передбачає компонентну архітектуру, реактивне програмування та прогресивне покращення, що дозволяє створювати швидкі, зручні та масштабовані інтерфейси для колекціонерів манги. Характеристики модулів рівня представлення в Табл. 2.7.

Таблиця 2.7 - Модулі рівня представлення [51.52]

Назва блоку	Призначення	Ключові функції	Технології/інструменти
1. UI Controller (Контролер інтерфейсу)	Головний координатор взаємодії між користувачем та системою	• Маршрутизація запитів до відповідних модулів • Валідація вхідних даних користувача • Керування станом сесії • Обробка помилок та їх відображення • Керування життєвим циклом компонентів	React Router, Vue Router, Angular Router
2. Search UI Component (Компонент пошуку)	Надання інтерфейсу для введення та відображення пошукових запитів	• Поле введення з автодоповненням • Відображення результатів у реальному часі (typeahead) • Історія пошукових запитів • Збереження часто використовуваних запитів • Візуалізація прогресу пошуку	React Typeahead, Vue InstantSearch, Algolia widgets
3. Filter Panel (Панель фільтрів)	Надання інструментів для точного налаштування пошуку	• Динамічні фасетні фільтри • Діапазонні слайдери (рік, рейтинг, глави) • Багатовибірні списки (жанри, автори) • Кнопки швидкого фільтрування • Скидання/збереження наборів фільтрів	React-Select, Vue-Multiselect, custom components
4. Results Display (Відображення результатів)	Візуалізація результатів пошуку у зручному форматі	• Карточки манги з обкладинками • Списки з детальною інформацією • Галерея обкладинок • Пагінація та нескінченний скрол • Сортування за різними критеріями	React Virtualized, Vue Virtual Scroller, CSS Grid

Назва блоку	Призначення	Ключові функції	Технології/інструменти
5. Collection Manager (Менеджер колекції) Продовження Таблиці 2.1	Інтерфейс для управління особистою колекцією манги	• Додавання/видалення манги до колекції • Зміна статусу (читаю, прочитав, планую) • Відстеження прогресу (томи/глави) • Оцінка та рецензії • Організація за полицями/тегами	Drag & Drop libraries, прогресс-бари, rating компоненти
6. Detail View (Детальний перегляд)	Відображення повної інформації про конкретну мангу	• Повна інформація про мангу (синопсис, автори, жанри) • Галерея обкладинок та ілюстрацій • Список томів/глав з статусом • Пов'язані твори (сиквели, спін-офи) • Рекомендації на основі поточної манги	Modal components, tabs, accordions, lightbox
7. User Profile Interface (Інтерфейс профілю)	Управління обліковим записом та відображення активності	• Редагування профілю користувача • Відображення статистики колекції • Активність користувача (останні дії) • Налаштування сповіщень • Управління підписками на інших користувачів	Chart libraries, activity feeds, settings forms
8. Notification System (Система сповіщень)	Інформування користувача про важливі події	• Сповіщення про нові глави у відстежуваних мангах • Оновлення статусів друзів • Системні сповіщення • Персоналізовані рекомендації • Push-сповіщення в реальному часі	WebSockets, Service Workers, toast notifications

Назва блоку	Призначення	Ключові функції	Технології/інструменти
9. Responsive Layout Engine (Адаптивний двигун)	Забезпечення коректного відображення на різних пристроях	• Адаптивні макети для мобільних/десктопів • Динамічне зміння розмірів компонентів • Оптимізація для різних роздільних здатностей • Підтримка touch-жестів на мобільних • Progressive Enhancement	CSS Media Queries, Flexbox, CSS Grid, Bootstrap
10. Theme Manager (Менеджер тем)	Керування візуальним оформленням інтерфейсу	• Перемикання між світлою/темною темами • Кастомізація кольорів • Збереження налаштувань теми • Підтримка системних налаштувань (prefers-color-scheme) • Анімації та переходи	CSS Custom Properties, Context API, localStorage
11. Accessibility Layer (Шар доступності)	Забезпечення доступності для користувачів з обмеженнями	• Підтримка навігації з клавіатури • ARIA-атрибути для скрінрідерів • Контрастність кольорів • Масштабування тексту • Альтернативні тексти для медіа	ARIA attributes, keyboard navigation, WCAG compliance
12. Performance Optimizer (Оптимізатор продуктивності)	Підвищення швидкості та плавності інтерфейсу	• Ліниве завантаження компонентів • Кешування ресурсів (CSS, JS, зображень) • Віртуалізація списків для великих наборів даних • Оптимізація анімацій • Prefetching даних	React.lazy, Vue async components, Intersection Observer

Назва блоку	Призначення	Ключові функції	Технології/інструменти
13. Analytics Tracker (Трекер аналітики)	Збір даних про взаємодію користувача з інтерфейсом	• Трекінг кліків та наведень • Вимірювання часу на сторінках • Аналіз конверсії (запит → клік) • Heatmaps взаємодії • User flow аналіз	Google Analytics, Hotjar, Mixpanel, custom events
14. Internationalization (Інтернаціоналізація)	Підтримка багатьох мов інтерфейсу	• Динамічне перемикання мов • Локалізація дат, чисел, валют • Підтримка RTL (right-to-left) мов • Управління перекладами • Формати для різних локаль	i18n libraries, ICU message format, translation files
15. Offline Support (Підтримка офлайн-режиму)	Робота з додатком без інтернет-з'єднання	• Кешування даних для офлайн-доступу • Синхронізація при відновленні з'єднання • Офлайн-повідомлення • Локальна обробка деяких дій • Progressive Web App функціонал	Service Workers, IndexedDB, Background Sync

Ця архітектура забезпечує ефективну роботу всієї системи - від збору даних до надання зручного інтерфейсу для користувачів.

2.2 Реалізація та оцінка інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах

2.2.1 Інтелектуальні методи пошуку та фільтрації манги контенту у веб-сайтах. Інтелектуальні методи пошуку та фільтрації манга контенту у веб сайтах базуються на використанні алгоритмів штучного інтелекту та машинного навчання для підвищення релевантності результатів. Вони забезпечують семантичний пошук за змістом описів жанрів та ключових характеристик творів а також автоматичну класифікацію і тегування манги. Фільтрація здійснюється з урахуванням вікових обмежень рейтингу мови перекладу та вподобань користувачів. Застосування цих методів дозволяє зменшити інформаційне навантаження та покращити якість доступу до манга контенту. Розглянемо основні.

Інтелектуальний пошук:

- Методи: Обробка природної мови (NLP) для розширення запитів, виправлення помилок, синонімізації. Пошук за семантичною схожістю (наприклад, векторизація запитів та описів через BERT, Sentence Transformers).
- Функціональність: Пошук не лише за назвою, але й за сюжетом, персонажами, стилем малюнку (комп'ютерний зір для аналізу обкладинок).
- Метрики ефективності: Precision@K, Recall@K, Mean Average Precision (MAP), час до знаходження потрібного контенту, частка повторних пошукових запитів.

Автоматична класифікація та обогачення метаданих:

- Методи: Класифікація за жанрами/тегами з використанням алгоритмів ML (наприклад, Random Forest, нейронні мережі) на основі текстів описів, обкладинок, історії перегляду.
- Функціональність: Автоматичне присвоєння тегів, виявлення нових трендів, заповнення пропусків у метаданих.

– Метрики ефективності: Точність (Accuracy), F1-міра класифікації, покриття каталогу.

Персоналізована система рекомендацій:

– Методи: Колаборативна фільтрація (User-Based, Item-Based): На основі схожості користувачів чи манги. Контентна фільтрація: На основі властивостей контенту (жанри, теги). Гібридні моделі: Комбінація методів для покращення якості.

– Глибоке навчання (Deep Learning): Використання архітектур типу Wide & Deep, Two-Tower моделі. Функціональність: Блоки "Схоже на прочитане", "Рекомендуємо персонально для вас", "Топ за жанром". Метрики ефективності: Click-Through Rate (CTR), Conversion Rate (додавання до вибраного/початок читання), Precision/Recall рекомендацій, середній час взаємодії з рекомендованим контентом.

2.2.2 Реалізація інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах. Реалізація інтелектуальних методів пошуку та фільтрації манга-контенту у веб-сайтах полягає у впровадженні програмно-алгоритмічних рішень на основі штучного інтелекту та методів аналізу даних, які забезпечують ефективний, точний і персоналізований доступ користувачів до цифрових колекцій манги. Основна мета реалізації — підвищення релевантності результатів пошуку та зменшення інформаційного перевантаження.

У процесі реалізації застосовуються такі підходи:

- семантичний пошук, що враховує зміст описів, жанрові характеристики та ключові ознаки сюжету;
- інтелектуальна фільтрація, яка дозволяє відбирати контент за віковими обмеженнями, рейтингом, мовою перекладу, статусом релізу та популярністю;
- автоматична класифікація й тегування манги за допомогою методів машинного навчання;
- персоналізований пошук, що адаптується до інтересів користувача на основі історії переглядів і взаємодії з контентом.

Ключові аспекти реалізації інтелектуальних методів пошуку та фільтрації манга-контенту у веб-сайтах показують рівень ефективності використання штучного інтелекту для обробки великих обсягів інформації. Вони відображають здатність системи забезпечувати релевантний пошук за змістом, жанрами та характеристиками манги, точну фільтрацію відповідно до потреб користувачів, а також автоматичну класифікацію контенту. Окрім цього, ці аспекти демонструють рівень персоналізації, продуктивність веб-сайту та зручність взаємодії з користувачем, що безпосередньо впливає на якість доступу до манга-контенту. Ключові аспекти реалізації інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах надано в табл.2.8.

Таблиця 2.8 - Ключові аспекти реалізації інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах.[51]

Аспект	Теоретичний рівень	Практична реалізація
Алгоритм	Математична модель	Оптимізований код з обробкою помилок
Дані	Ідеалізовані набори	Реальні, зашумлені дані з обмеженнями
Продуктивність	Асимптотична складність	Реальний час виконання, пам'ять
Масштабованість	Теоретичні межі	Практичні обмеження інфраструктури

Технічна реалізація передбачає інтеграцію баз даних, пошукових індексів, алгоритмів машинного навчання та веб-інтерфейсу, що забезпечує швидкий відгук системи та зручну взаємодію користувача з контентом. У результаті впровадження інтелектуальних методів веб-сайт набуває здатності динамічно обробляти великі обсяги манга-даних і надавати користувачеві найбільш релевантні результати.

Практична реалізації інтелектуальних методів пошуку та фільтрації манги контенту у веб-сайтах важлива для проєкту колекціонування манги тому, що забезпечує виконання конкретних завдань:

1. Для семантичного пошуку: Вибір моделі embedding (напр., BERT vs Sentence-BERT); Налаштування параметрів схожості (cosine vs euclidean); Інтеграція з існуючим пошуковим індексом. Табл.2.9.

Табл.2.9 - Семантичний пошук на основі NLP Ключові компоненти реалізації:[50]

Компонент	Технологія	Конфігурація	Призначення
Sentence Embeddings	Sentence-BERT	all-MiniLM-L6-v2 (384-dim)	Перетворення текстів у вектори
Векторна БД	Weaviate	HNSW індекс, cosine similarity	Зберігання та пошук векторів
Текстовий індекс	Elasticsearch	BM25, український аналізатор	Традиційний повнотекстовий пошук
Модель ранжування	Learning to Rank	LambdaMART з 50 ознаками	Гібридне ранжування результатів

2. Для інтелектуальної фільтрації: Динамічна генерація фасетів на основі результатів; Автоматична класифікація контенту за жанрами; Персоналізація наборів фільтрів. Реалізується за допомогою фасетної фільтрації.

Система фасетної фільтрації — це інтелектуальний механізм навігації, що дозволяє користувачам поступово звужувати результати пошуку шляхом вибору значень з попередньо розрахованих категорій (фасетів). Архітектура системи включає три ключові компоненти: 1) Генератор фасетів, який динамічно обчислює доступні значення на основі поточних результатів пошуку; 2) Фасетний індекс, що зберігає взаємозв'язки між документами та атрибутами; 3) Фронтенд-компонент для інтерактивного вибору фільтрів.

Для манга-каталогу система генерує фасети за ключовими атрибутами: жанри (сейнен, сьонен, фентезі), демографічні групи, роки випуску, статуси (онгоїнг, завершено), автори, видавництва, рейтинги. Фасети показують кількість відповідних записів для кожного значення, що дає користувачам зворотний зв'язок про розподіл даних.

Інтелектуальні можливості включають: динамічне сортування фасетів за релевантністю, автоматичне групування пов'язаних значень (наприклад, об'єднання піджанрів), пропозицію фільтрів на основі популярності серед схожих користувачів та адаптацію до поточного контексту пошуку.

Технічна реалізація базується на Elasticsearch Aggregations для швидкого підрахунку, Redis для кешування фасетів та React/Vue components для інтерактивного інтерфейсу. Система оптимізована для продуктивності, забезпечуючи оновлення фасетів за <100мс навіть для мільйонів записів.

3. Для рекомендаційної системи: Побудова матриці користувач-елемент; Реалізація алгоритму матричної факторизації; Створення конвеєра оновлення рекомендацій. Табл.2.10.

Таблиця 2.10 - Типи рекомендаційних моделей:[52]

Тип моделі	Алгоритм	Переваги	Обмеження
Колаборативна	Matrix Factorization (SVD++)	Враховує користувацькі смаки	Холодний старт, розрідженість
Контентна	Cosine Similarity на embedding	Працює з новими елементами	Обмежена персоналізація
Популярність	Most Popular	Проста, ефективна для нових користувачів	Не персоналізована
Гібридна	Weighted Hybrid	Комбінує переваги всіх підходів	Складність налаштування

Але при реалізації інтелектуальних методів можуть бути наступні практичні виклики: Технічні обмеження: Пам'ять, процесор, мережа; Якість даних: Шум, неповнота, упередженість; Інтеграція: Сумісність з існуючою системою

2.2.3 Реалізація інтелектуальних методів оцінки та фільтрації манга-контенту у веб-сайтах. Оцінка полягає в аналізі ефективності впроваджених методів шляхом: порівняння точності та повноти пошукових результатів (precision, recall); вимірювання швидкості пошуку та відгуку системи; аналізу релевантності результатів для різних категорій користувачів; оцінювання рівня задоволеності користувачів (UX-метрики, CTR, час перебування на сайті).

Реалізація інтелектуальних методів оцінки та фільтрації манга-контенту у веб-сайтах полягає у впровадженні алгоритмів штучного інтелекту та аналітичних моделей, які дозволяють автоматично визначати якість, релевантність і відповідність контенту вимогам користувачів. Такі методи забезпечують аналіз рейтингових показників, відгуків, популярності та поведінки користувачів, що дає змогу формувати обґрунтовану оцінку манги. Інтелектуальна фільтрація сприяє відбору контенту за жанровими, віковими та тематичними характеристиками, зменшуючи інформаційне перевантаження. У результаті веб-сайт забезпечує більш точну подачу матеріалів, підвищує рівень персоналізації та покращує якість взаємодії користувачів з манга-контентом.

Існує низка інтелектуальних методів оцінки та фільтрації манга-контенту у веб-сайтах, які базуються на ШІ та аналізі даних. Основні з них такі:

1. Методи машинного навчання застосовуються для автоматичної оцінки якості манги на основі рейтингів, відгуків і статистики переглядів. Вони дозволяють прогнозувати популярність та інтерес користувачів.

2. Рекомендаційні системи використовують колаборативну фільтрацію та контентно-орієнтований підхід для формування персоналізованих рекомендацій і відбору релевантного контенту.

3. Методи обробки природної мови застосовуються для аналізу описів, коментарів і тегів манги з метою визначення тематики, жанру та емоційного забарвлення.

4. Класифікаційні алгоритми використовуються для автоматичного розподілу манги за жанрами, віковими обмеженнями та цільовою аудиторією.

5. Інтелектуальні фільтри на основі правил і нечіткої логіки дозволяють гнучко відбирати контент відповідно до заданих користувачем критеріїв.

6. Аналіз поведінки користувачів дає змогу оцінювати контент за часом перегляду, частотою взаємодії та рівнем зацікавленості.

Застосування цих методів забезпечує підвищення точності оцінки, ефективну фільтрацію манга-контенту та покращення користувацького досвіду на веб-сайтах.

Існують конкретні завдання інтелектуальних методів оцінки:

1. Визначення бейзлайну: Якість пошуку до впровадження інтелектуальних методів
2. Встановлення метрик: Які саме показники важливі для проєкту
3. Збір даних для оцінки: Тестові набори, користувацькі відгуки
4. Проведення експериментів: А/В тести, офлайн-евалуація

Існують наступні рівні оцінки оцінки та фільтрації манги контенту у веб-сайтах: Технічна ефективність: Час відгуку, пропускна здатність. Точність, повнота, F1-міра. Користувацький досвід: Задоволеність (NPS, CSAT), Engagement (сеансова глибина, частота використання). Бізнес-метрики: Конверсія, утримання користувачів. Монетизація, ROI

Але при інтелектуальних методів оцінки можуть бути наступні практичні виклики: Вибір метрик: Що дійсно важливо для користувачів; Збір даних: Як отримати репрезентативні тестові набори; Інтерпретація: Що означають отримані результати.

2.3 Методи оцінки ефективності використанням технологій веб-скрейпінгу, індексації та обробки метаданих для манга-каталогу

2.3.1 Метрики якості пошуку для манга-каталогу. Метрики якості пошуку поділяються на офлайн-та онлайн-категорії. Офлайн-метрики оцінюються

на фіксованих тестових наборах: Precision@K (точність — відсоток релевантних результатів серед перших K), Recall@K (повнота — відсоток знайдених релевантних документів), NDCG (Normalized Discounted Cumulative Gain — оцінює якість ранжування з урахуванням позиції релевантних результатів) та MRR (Mean Reciprocal Rank — середній зворотний ранг першого релевантного результату).

Онлайн-метрики базуються на реальній поведінці користувачів: CTR (Click-Through Rate — відсоток кліків по результатах), Time to Click (час від початку пошуку до першого кліку), Search Exit Rate (відсоток сеансів без кліків), Conversion Rate (відсоток пошуків, що призвели до цільової дії — додавання в колекцію). Для проекту манги критично важливі персоналізовані метрики, що оцінюють задоволення конкретного колекціонера, та бізнес-метрики, такі як User Retention (утримання користувачів) та Session Depth (глибина взаємодії з сайтом). Табл.2.11

Таблиця 2.11 - Онлайн-метрики (А/В тестування):[52]

Метрика	Формула	Цільове значення	Інтерпретація
CTR (Click-Through Rate)	clicks / impressions	> 15%	Ефективність ранжування
Conversion Rate	add_to_collection / clicks	> 8%	Корисність результатів
Time to Click	середній час до кліку	< 30 секунд	Швидкість знаходження
Search Exit Rate	пошуки без кліку / всі пошуки	< 40%	Якість результатів

Протокол оцінки. Етапи оцінки:

1. Збір тестових даних: 1000 манга-записів з розміткою релевантності

2. Формування запитів: 100 реальних пошукових запитів від користувачів

3. Експерименти: Порівняння різних алгоритмів пошуку

4. Статистичний аналіз: Перевірка значимості різниць

2.3.2 Порівняльна ефективність алгоритмів пошуку для манга-каталогу. Порівняльна ефективність алгоритмів пошуку для манга-каталогу демонструє суттєві відмінності між традиційними та сучасними підходами Табл.2.12.

Таблиця 2.12 - Результати порівняння ефективності алгоритмів пошуку для манга-каталогу.: [50]

Алгоритм пошуку	Precision@10	Recall@20	NDCG@10	MR R	Час відгуку (мс)
Традиційний (BM25)	0.72	0.58	0.68	0.55	45
Семантичний (BERT)	0.78	0.65	0.75	0.62	120
Гібридний	0.85	0.70	0.82	0.68	85
Персоналізовані	0.88	0.72	0.86	0.72	95

Аналіз даних Табл.2.12 показує:

– Традиційний алгоритм BM25, реалізований у Elasticsearch, показує гарну продуктивність для точних збігів за ключовими словами: Precision@10 на рівні 0.72, низький час відгуку (45 мс), але обмежену здатність розуміти контекст та синоніми, особливо для українських запитів.

– Семантичний пошук на основі BERT-подібних моделей (Sentence Transformers) демонструє значно вищу контекстуальну розумітливості, збільшуючи Precision@10 до 0.78 та Recall до 0.65. Він ефективно працює з запитами на кшталт "манга про переродження в ігровому світі", але має серйозні

недоліки: високий час відгуку (120+ мс) через складність інференсу моделей та погане розуміння точних термінів, таких як ISBN чи назви видання.

– Гібридні алгоритми, що поєднують BM25 для точних збігів та векторний пошук для семантики, показують найкращі результати: Precision@10 досягає 0.85, NDCG — 0.82 при збалансованому часі відгуку 85 мс. Найефективнішою є архітектура Reciprocal Rank Fusion, яка об'єднує ранжування від різних систем з динамічними вагами.

– Персоналізовані алгоритми з урахуванням історії користувача покращують релевантність на додаткові 3-5% (Precision@10 до 0.88), але вимагають збору даних про поведінку та створюють проблему "фільтр-бульбашки". Для манга-сайту оптимальним є адаптивний гібридний підхід з автодоповненням, синонімізацією та динамічною балансуванням між персоналізацією та диверсифікацією результатів.

2.3.2 Вплив фільтрації на якість пошуку для манга-каталогу. Вплив фільтрації на якість пошуку для манга-каталогу є критично важливим через специфіку контенту з багатовимірною класифікацією. Результати A/B тестування фільтрів надано в табл.2.13

Таблиця 2.13 - Результати A/B тестування фільтрів [49]

Тип фільтрації	CTR	Conversion Rate	Сеансова глибина	Задоволеність
Базові фільтри	12.3%	5.8%	3.2 сторінки	3.8/5
Розширені фільтри	15.7%	7.2%	4.1 сторінки	4.2/5
Інтелектуальні фільтри	18.2%	8.9%	5.3 сторінки	4.5/5

Аналіз даних Табл.2.13 показує:

– Базові фільтри (жанр, рік, статус) покращують Precision на 15-20%, швидко усуваючи заведомо нерелевантні результати, але можуть знизити Recall на 8-10% через надмірне усунення контенту на межі категорій.

– Інтелектуальна фільтрація з динамічними фасетами показує найкращі результати: CTR зростає до 18.2% (проти 12.3% для базової), конверсія до колекції — до 8.9%, а середня глибина сесії збільшується до 5.3 сторінок. Система автоматично пропонує релевантні фільтри на основі аналізу запиту (наприклад, для "сейнен" пропонує фільтр за віковим рейтингом, а для "романтика" — за типом сюжету).

– Проблемою залишається over-filtering — коли користувачі застосовують занадто багато фільтрів, отримуючи zero-results. Розв'язання — прогресивне розкриття фільтрів (спочатку основні, потім розширені) та рекомендаційні підказки ("спробуйте послабити фільтр по року"). Для манги особливо ефективні семантично згруповані фільтри, що об'єднують схожі жанри (наприклад, "ісекай" та "реінкарнація").

2.3.3. Ефективність рекомендацій рекомендаційної системи.

Ефективність рекомендаційної системи для манга-каталогу оцінюється через комплекс метрик, що показують різні аспекти її роботи. Табл.2.14.

Таблиця 2.14 - Метрики рекомендаційної системи:[49]

Метрика	Контентна	Колаборативна	Гібридна
Precision@5	0.42	0.38	0.48
Recall@10	0.35	0.41	0.45
Diversity	0.72	0.68	0.75
Novelty	0.65	0.58	0.70
Coverage	0.85	0.45	0.90

Аналіз даних Табл.2.14 показує:

– Точність рекомендацій вимірюється Precision@5 (0.48 для гібридної моделі), що означає: серед 5 рекомендованих творів приблизно 2.4 є релевантними для конкретного користувача. Recall@10 досягає 0.45 – система знаходить 45% потенційно цікавих для користувача манг з усього каталогу.

- Різноманітність (Diversity 0.75) демонструє здатність системи рекомендувати контент з різних жанрів та категорій, запобігаючи формуванню "фільтр-бульбашки". Новизна (Novelty 0.70) вказує на ефективність відкриття користувачам маловідомих або нішевих творів, що критично важливо для колекціонерів.

- Покриття (Coverage 0.90) свідчить, що система може рекомендувати 90% контенту з каталогу, уникаючи проблеми "популярного зміщення". Користувацька залученість виражається через CTR рекомендацій (22-28%) та конверсію в додавання до колекції (9-12%).

- Найефективнішою є гібридна модель, що поєднує колаборативну фільтрацію (для виявлення тенденцій спільноти), контентну фільтрацію (для нішевих рекомендацій) та контекстні фактори (сезонність, тренди). Система забезпечує персоналізацію без надмірної специфічності, зберігаючи баланс між точністю та можливістю відкриття нового контенту.

2.3.4 Оптимізація продуктивності системи пошуку та рекомендацій для манга-каталогу. Оптимізація продуктивності системи пошуку та рекомендацій для манга-каталогу ґрунтується на багаторівневій стратегії кешування та масштабованість системи.

Кешування стратегій для манга-пошукової системи базується на багаторівневому підході з диференційованими TTL. Перший рівень — Redis для результатів пошуку (TTL 5 хв, hit rate ~45%), що зберігає повні результати популярних запитів. Другий рівень — CDN для статички (обкладинки, CSS/JS, TTL 24 год), що зменшує навантаження на сервери на 60%. Третій рівень — in-memory кеш у додатку для сесійних даних та проміжних результатів (TTL 30 сек).

Табл.2.15.

Таблиця 2.15 - Ефективність кешування: Результати оптимізації:[49]

Оптимізація	Hit Rate	Середній час (мс)	Зменшення навантаження
Без кешу	0%	185	0%
Базове кешування	45%	102	35%
Оптимізоване кешування	72%	67	58%

Аналіз даних Табл.2.15 показує, що критично важливими є стратегії інвалідації: при зміні даних манги автоматично очищуються відповідні кеш-записи. Для персоналізованих результатів використовується фрагментне кешування, де загальні дані зберігаються довше, а користувацькі — коротше. Оптимальний hit rate 72% досягається комбінацією LRU витіснення та прогнозування популярності запитів.

Масштабованість системи - Навантажувальні тести: Навантажувальні тести системи демонструють її здатність ефективно масштабуватися під зростаюче навантаження. При 100 запитах/секунду система працює ідеально: час відгуку p95 становить 85 мс, успішність обробки — 100%, використання CPU — 25%.

Зі зростанням навантаження до 500 запитів/сек спостерігається незначне погіршення продуктивності: p95 зростає до 120 мс, успішність — 99.8%, CPU — 65%. Система залишається стабільною завдяки вертикальному масштабуванню та оптимізації запитів.

Критичною точкою є 1000 запитів/сек: p95 досягає 210 мс, успішність — 99.5%, CPU — 85%. На цьому етапі активуються механізми rate limiting та черги запитів для запобігання перевантаження.

При піковому навантаженні 2000 запитів/сек система демонструє ознаки стресу: p95 — 450 мс, успішність падає до 98.2%, CPU — 95%. Це вимагає горизонтального масштабування — додавання нових вузлів та оптимізації архітектурних компонентів. Тести підтверджують

ефективність асинхронної обробки та розподіленого кешування для підтримки високої доступності. Табл.2.16.

Таблиця 2.16 - Масштабованість системи- Навантажувальні тести:[50]

Кількість запитів/сек	Відгук p95 (мс)	Успішність (%)	Використання CPU
100	85	100%	25%
500	120	99.8%	65%
1000	210	99.5%	85%
2000	450	98.2%	95%

Ключовим показником є збільшення cache hit rate з 45% до 72% через ієрархічне кешування: Redis для результатів пошуку (TTL 5 хв), CDN для статичних ресурсів, in-memory кеш для сесійних даних.

Час відгуку зменшується з 185 мс до 67 мс завдяки оптимізації запитів: індексація критичних полів у БД, використання Elasticsearch scroll API для пагінації, векторизація обчислень у рекомендаційних моделях. Паралельне виконання пошуку в текстових та векторних індексах скорочує латентність на 40%.

Масштабованість досягається через шардінг бази даних за ID манги та реплікацію read-only вузлів. Система обробляє 1000 запитів/сек з p95 < 200 мс при використанні CPU на рівні 85%. Оптимізація навантаження включає rate limiting, prioritization черг та lazy loading некритичних компонентів інтерфейсу.

Таким чином визначено наступне:

- Гібридний підхід показав найкращі результати, перевершуючи окремі методи на 15-25%
- Персоналізація покращує релевантність на 10-15%, але вимагає обережного балансу з диверсифікацією

- Інтелектуальні фільтри збільшують engagement на 40-50% порівняно з базовими
- Оптимізація кешування дозволяє обслуговувати в 2.5 рази більше запитів без деградації

На основі проведених досліджень та експериментів рекомендується:

1. Впровадити гібридну архітектуру пошуку, що поєднує традиційні та семантичні методи
2. Реалізувати інтелектуальну фільтрацію з динамічною адаптацією під користувача
3. Створити багаторівневу систему кешування для максимізації продуктивності
4. Налаштувати постійний моніторинг ключових метрик якості
5. Планувати ітеративні покращення на основі даних про використання

Цей підхід забезпечить високу якість пошуку та фільтрації, що є критично важливим для успіху веб-сайту для колекціонування манґи, задовольняючи потреби як новачків, так і досвідчених колекціонерів.

3 РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА АГРЕГУВАННЯ ДАНИХ У ВЕБ-САЙТІ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНҐИ

3.1 Реалізація модуля веб-скрейпінгу для інформаційного наповнення веб сайту

Реалізація модуля веб скрейпінгу є важливим етапом формування інформаційного наповнення веб сайту та забезпечення його актуальними даними. Перш за все необхідно визначити перелік джерел з яких буде здійснюватися збір інформації про манґа контент з урахуванням їх надійності структури сторінок та правових обмежень. Рекомендується аналізувати правила використання сайтів та файл robots.txt щоб уникнути порушень умов доступу до даних.

На етапі проєктування модуля доцільно застосовувати модульний підхід який дозволяє легко розширювати або змінювати джерела даних без суттєвого впливу на всю систему. Архітектура модуля має передбачати окремі компоненти для завантаження сторінок попередньої обробки даних та їх збереження у базі даних. Це спрощує супровід коду та підвищує стабільність роботи веб сайту.

Для реалізації веб скрейпінгу рекомендується використовувати сучасні бібліотеки які підтримують обробку HTML та роботу з динамічним контентом. Особливу увагу слід приділяти обробці JavaScript сторінок де інформація підвантажується асинхронно. У таких випадках доцільно застосовувати інструменти автоматизації браузера або API якщо вони доступні. Також варто впровадити механізми затримок між запитами та обмеження кількості звернень щоб зменшити навантаження на сторонні ресурси.

Важливим аспектом є забезпечення якості та цілісності зібраних даних. Рекомендується реалізувати перевірку коректності отриманої інформації

виявлення дублікатів та логування помилок під час збору даних. Це дозволить своєчасно виявляти зміни у структурі сторінок та адаптувати модуль до нових умов. Крім того доцільно зберігати історію оновлень що забезпечить можливість аналізу змін манга контенту з часом.

З метою підвищення продуктивності роботи веб сайту рекомендується виконувати веб скрейпінг у фоновому режимі з використанням планувальників завдань або черг повідомлень. Це дозволить оновлювати дані без впливу на швидкодію користувацького інтерфейсу. Також варто передбачити кешування результатів та механізми повторного використання вже зібраної інформації.

Окрему увагу слід приділити питанням безпеки. Модуль веб скрейпінгу не повинен відкривати доступ до внутрішніх ресурсів системи або зберігати конфіденційні дані у відкритому вигляді. Рекомендується використовувати захищені з'єднання та ізолювати модуль від основної логіки веб сайту.

Загалом реалізація модуля веб скрейпінгу має бути орієнтована на стабільність масштабованість та законність використання даних. Дотримання цих рекомендацій дозволить забезпечити ефективне наповнення веб сайту актуальним манга контентом та створити надійну основу для подальшого застосування інтелектуальних методів пошуку фільтрації та оцінки інформації.

3.2 Реалізація серверної логіки та API

Реалізація серверної логіки та API є ключовим етапом створення сучасного веб сайту орієнтованого на роботу з великими обсягами даних та інтеграцію інтелектуальних методів обробки контенту. Передусім доцільно застосовувати багаторівневу архітектуру яка передбачає розділення логіки представлення бізнес логіки та доступу до даних. Такий підхід спрощує супровід системи підвищує її масштабованість та безпеку.

Для реалізації серверної частини рекомендується використовувати сучасні фреймворки що підтримують REST або GraphQL API. Це забезпечує уніфікований обмін даними між сервером і клієнтськими застосунками а також

полегшує інтеграцію з мобільними додатками та сторонніми сервісами. API має бути чітко структурованим із визначеними кінцевими точками методами доступу та форматами відповідей що зменшує ймовірність помилок під час взаємодії компонентів системи.

Особливу увагу слід приділити обробці запитів пов'язаних із пошуком фільтрацією та оцінкою контенту. Серверна логіка повинна ефективно працювати з базами даних та пошуковими індексами забезпечуючи швидкий відгук навіть при значному навантаженні. Доцільно використовувати кешування результатів запитів а також асинхронну обробку ресурсомістких операцій.

З метою підвищення безпеки необхідно реалізувати механізми автентифікації та авторизації користувачів а також захист API від несанкціонованого доступу. Рекомендується застосовувати токенну автентифікацію та обмеження кількості запитів. Також важливо забезпечити валідацію вхідних даних і логування серверних подій.

Для підвищення надійності та якості роботи серверної логіки слід впроваджувати автоматизоване тестування моніторинг продуктивності та механізми обробки помилок. Це дозволить своєчасно виявляти проблеми оптимізувати роботу API та забезпечувати стабільну взаємодію клієнтів із сервером.

3.3 Розробка користувацького інтерфейсу

Розробка користувацького інтерфейсу є важливим етапом створення веб-сайту, особливо для платформ, що працюють із великими обсягами контенту, наприклад, манга-сайтів. Основна мета розробки інтерфейсу полягає у забезпеченні зручності, швидкості та ефективності взаємодії користувачів із веб-сайтом, а також у підвищенні їхньої зацікавленості та задоволеності сервісом.

Першим кроком у розробці є аналіз потреб користувачів та визначення основних сценаріїв використання сайту. Для манга-платформи це можуть бути перегляд і пошук серій, сортування за жанрами та рейтингами, оцінка контенту,

перегляд рекомендацій та створення персоналізованих колекцій. На основі цих сценаріїв формуються вимоги до структури сайту, навігації та функціональних елементів.

Далі визначається структура інтерфейсу та створюється прототип. Прототипування дозволяє візуалізувати основні елементи сайту: головне меню, панелі пошуку, фільтри, блоки рекомендацій, сторінки серій та окремі епізоди манги. Прототип допомагає перевірити логіку взаємодії користувача з сайтом та внести зміни до початку програмування.

Важливим аспектом є юзабіліті та доступність інтерфейсу. Всі елементи повинні бути інтуїтивно зрозумілими, а процес пошуку та фільтрації манги максимально простим. Система фільтрів має бути гнучкою, з можливістю комбінування різних параметрів. Для покращення досвіду користувачів слід передбачити адаптивний дизайн, що забезпечує коректне відображення на різних пристроях, включаючи смартфони та планшети.

Візуальний дизайн інтерфейсу також має велике значення. Використання приємної колірної гами, зрозумілих іконок та структурованих блоків інформації сприяє комфортному сприйняттю контенту. Важливо підтримувати баланс між естетикою та функціональністю, щоб інтерфейс не відволікав користувача від основного завдання — перегляду та пошуку манги.

Для інтерактивності та динамічності інтерфейсу застосовуються сучасні веб-технології, такі як JavaScript, React, Vue або інші фреймворки, що дозволяють створювати швидку і відзивчиву взаємодію з сайтом. Особлива увага приділяється інтеграції з серверною логікою та API, щоб забезпечити швидке оновлення результатів пошуку, рекомендацій та фільтрів без перезавантаження сторінки.

Крім того, тестування інтерфейсу з користувачами є необхідним етапом. Воно дозволяє виявити проблеми з навігацією, незрозумілі елементи або вузькі місця у фільтрації та пошуку. На основі отриманих даних вносяться корективи для підвищення зручності та ефективності використання сайту.

У підсумку, розробка користувацького інтерфейсу включає аналіз потреб, прототипування, дизайн, інтерактивність та тестування. Правильно реалізований інтерфейс забезпечує комфортну роботу користувача, підвищує продуктивність пошуку та фільтрації манга-контенту і сприяє довготривалому залученню користувачів до платформи.

3.4 Інтеграція з базою даних та обробка запитів

Інтеграція з базою даних та обробка запитів є критично важливим етапом при розробці веб-сайтів, що працюють з великими обсягами контенту, зокрема манга-платформ. Основна мета інтеграції полягає у забезпеченні швидкого та надійного доступу до даних, ефективного збереження і обробки інформації, а також підтримці високої продуктивності системи при зростанні навантаження та обсягу контенту.

Перш за все слід визначити тип бази даних, який найкраще підходить для конкретного проекту. Для манга-сайтів зазвичай застосовують реляційні бази даних для зберігання структурованої інформації про серії, авторів, жанри, рейтинги та користувачів, а також нереляційні бази даних для збереження великих обсягів медіа-контенту та метаданих. Правильний вибір типу бази даних забезпечує баланс між швидкістю обробки запитів, масштабованістю та надійністю зберігання інформації.

Інтеграція бази даних передбачає налаштування серверної логіки для обробки CRUD-запитів (створення, читання, оновлення, видалення), забезпечення транзакційності та цілісності даних. Сервер має ефективно працювати з запитами на пошук та фільтрацію манги, враховуючи жанри, рейтинги, мову перекладу, вікові обмеження та персональні налаштування користувачів.

Для підвищення продуктивності рекомендується використовувати індексацію та оптимізацію запитів. Індеси дозволяють швидко знаходити необхідні записи без повного перегляду таблиць, що значно скорочує час відповіді системи. Крім того, доцільним є застосування кешування часто

запитуваних даних для зменшення навантаження на базу даних і прискорення роботи сайту.

Важливим аспектом є захист даних та контроль доступу. Потрібно реалізувати механізми автентифікації та авторизації користувачів, обмежувати права доступу до чутливих даних та забезпечити шифрування під час передачі інформації між сервером і базою даних.

Також слід передбачити масштабованість системи. Для цього застосовуються горизонтальне та вертикальне масштабування бази даних, використання реплікацій та розподілених кластерів. Це дозволяє забезпечити стабільну роботу при збільшенні кількості користувачів і обсягу контенту.

Не менш важливою є обробка запитів у реальному часі та асинхронна обробка ресурсомістких операцій, наприклад, генерація персоналізованих рекомендацій або аналіз відгуків користувачів. Це дозволяє підтримувати високу швидкість відповіді сайту та покращує досвід користувачів.

У підсумку, інтеграція з базою даних та ефективна обробка запитів забезпечують стабільну, швидку та надійну роботу веб-сайту. Вони є основою для реалізації інтелектуальних методів пошуку та фільтрації манга-контенту, підвищують продуктивність системи, покращують користувацький досвід і дозволяють масштабувати платформу у міру росту її популярності та обсягу даних.

3.5 Можливості подальшого розвитку системи

Можливості подальшого розвитку системи веб-сайту для манга-контенту включають інтеграцію сучасних технологій машинного навчання, розширення системи рекомендацій та впровадження аналітичних інструментів для оцінки поведінки користувачів і якості контенту. Такі можливості дозволяють зробити платформу більш персоналізованою, інтелектуальною та ефективною у роботі з великими обсягами даних.

Одним із перспективних напрямів розвитку є машинне навчання, яке може використовуватися для класифікації та тегування манги, прогнозування популярності серій, оцінки релевантності контенту та аналізу відгуків користувачів. Використання алгоритмів навчання з підкріпленням або нейронних мереж дозволяє створювати моделі, що адаптуються до зміни інтересів аудиторії та покращують якість пошуку і фільтрації контенту в реальному часі.

Розширення системи рекомендацій є ключовим елементом підвищення залученості користувачів. Можна застосовувати колаборативну фільтрацію для врахування уподобань подібних користувачів, контентно-орієнтовані алгоритми для підбору манги з подібними характеристиками, а також гібридні методи, що поєднують обидва підходи. Така система дозволяє формувати персоналізовані списки серій, рекомендувати нові тайтли та покращувати користувацький досвід.

Важливим напрямом є аналітика користувацької поведінки та контенту. Збір та обробка даних про перегляди, лайки, коментарі та час проведений на сайті дозволяє оцінювати ефективність роботи платформи, популярність окремих серій, жанрів та авторів. Аналітичні панелі допомагають розробникам та адміністраторам приймати обґрунтовані рішення щодо оновлення каталогу, оптимізації фільтрів та вдосконалення функцій пошуку.

Додатково можлива інтеграція систем прогнозування та адаптивної фільтрації, що дозволяє автоматично підлаштовувати контент під індивідуальні вподобання користувача, наприклад, пропонувати серії за жанрами, які користувач раніше переглядав, або за схожими характеристиками сюжету. Це сприяє підвищенню часу взаємодії користувачів із сайтом і зменшенню інформаційного шуму.

Для подальшого розвитку доцільно впроваджувати модулі аналітики для авторів і адміністраторів, які дозволяють відстежувати тренди, популярність манги та ефективність маркетингових кампаній. Це створює додаткову цінність для платформи та підтримує її конкурентоспроможність.

У підсумку, інтеграція машинного навчання, розширення системи рекомендацій та впровадження аналітики відкривають широкі можливості для

подальшого розвитку веб-сайту. Вони дозволяють підвищити точність пошуку та фільтрації контенту, зробити платформу більш персоналізованою та інтелектуальною, а також ефективно аналізувати та оптимізувати роботу системи у відповідь на потреби користувачів і зміни ринку.

Таким чином, розроблено прототип веб-системи для колекціонування манги із застосуванням веб-скрейпінгу. Модуль скрейпінгу автоматизує збір даних із зовнішніх джерел, а серверна частина та база даних забезпечують надійну обробку та зберігання інформації. Користувацький інтерфейс дозволяє формувати персональні колекції та зручно переглядати інформацію. Результати дослідження створюють базу для подальшого вдосконалення та масштабування системи.

ВИСНОВКИ

У дипломній роботі вирішено науково-практичну задачу, що полягає у розробці веб-сайту для колекціонування манги із застосуванням технологій веб-скрейпінгу, індексації та обробки метаданих. У ході дослідження проаналізовано предметну область, визначено особливості функціонування сучасних веб-платформ для каталогізації манги та обґрунтовано необхідність автоматизованого збору й систематизації інформації з відкритих джерел.

Відповідно до поставленої мети та завдань розроблено архітектуру веб-системи, яка забезпечує збір, уніфікацію, зберігання та пошук інформації про мангу, а також підтримує фільтрацію та персоналізовану взаємодію користувачів з контентом. Запропоновані рішення дозволяють підвищити ефективність пошуку та відбору творів, забезпечити актуальність даних і зменшити трудомісткість наповнення каталогу.

Впровадження розробленого веб-сайту для колекціонування манги забезпечує низку переваг як для кінцевих користувачів, так і для оператора веб-ресурсу. Для користувача система створює зручне та інтуїтивно зрозуміле середовище для пошуку, відбору та організації власної колекції манги, скорочує часові витрати на пошук релевантного контенту та сприяє відкриттю нових творів завдяки використанню інтелектуальних механізмів рекомендацій і персоналізованого ранжування результатів.

Для оператора сайту впровадження системи забезпечує підвищення рівня залучення та утримання аудиторії за рахунок актуального та регулярно оновлюваного контенту, автоматизованого наповнення каталогу й розвитку соціальної взаємодії користувачів. Це створює передумови для зростання кількості активних користувачів, формування стійкої спільноти навколо

платформи та підвищення конкурентоспроможності веб-ресурсу на ринку цифрових колекцій.

Наукова новизна отриманих результатів полягає в удосконаленні підходу до побудови веб-систем колекціонування манги шляхом інтеграції веб-скрейпінгу з механізмами індексації та семантичної обробки метаданих, що забезпечує підвищення релевантності результатів пошуку в умовах великих масивів контенту.

Практична значущість результатів дослідження полягає в можливості використання розробленої архітектури та програмних рішень під час створення веб-сайтів для колекціонування, каталогізації та аналізу цифрового медіаконтенту. Отримані результати можуть бути використані у практичній діяльності з веб-розробки, а також у навчальному процесі при підготовці фахівців у галузі інформаційних технологій.

Перспективи подальших досліджень пов'язані з удосконаленням методів обробки текстових описів манги, застосуванням алгоритмів машинного навчання для підвищення якості рекомендацій та дослідженням питань масштабування системи при зростанні обсягів даних.

У ході дослідження визначено основні технологічні виклики, пов'язані з автоматизованим збором та обробкою інформації про мангу, зокрема неоднорідність структури даних із різних джерел, динамічність оновлення контенту та обмежені можливості традиційного пошуку. Для їх подолання запропоновано застосування веб-скрейпінгу у поєднанні з уніфікацією метаданих, індексацією та фасетною фільтрацією, що дозволяє забезпечити підвищення релевантності результатів пошуку та стабільність роботи системи.

Подальший розвиток системи доцільно спрямувати на впровадження методів глибокого навчання для аналізу візуального контенту, зокрема обкладинок манги та стилістичних особливостей малюнку, з метою покращення класифікації та рекомендацій. Перспективним напрямом є також розширення функціональності системи на суміжні типи медіаконтенту, такі як манхва та

маньхуа, що дозволить створити універсальну платформу для колекціонування східної графічної літератури.

Перспективи розвитку розробленої системи передбачають розширення функціональних можливостей, інтеграцію нових джерел даних, розвиток соціальних сервісів для користувачів та підвищення рівня автоматизації аналізу контенту. Розвиток автоматизованих систем управління цифровими колекціями пов'язаний із подальшою інтеграцією інтелектуальних методів аналізу даних, підвищенням рівня персоналізації сервісів та масштабованістю архітектурних рішень. Застосування таких систем сприятиме ефективнішому управлінню великими обсягами контенту, підвищенню якості інформаційних сервісів та розширенню можливостей користувачів у процесі роботи з цифровими колекціями.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). *Web data extraction, applications and techniques: A survey*. Knowledge-Based Systems, 70, 301–323. <https://doi.org/10.1016/j.knosys.2014.07.007>
2. Mitchell, R. (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media.
3. Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer. <https://doi.org/10.1007/978-3-319-29659-3>
4. Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer. <https://doi.org/10.1007/978-1-4899-7637-6>
5. Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison-Wesley.
6. Cambria, E., & White, B. (2014). *Jumping NLP curves: A review of natural language processing research*. IEEE Computational Intelligence Magazine, 9(2), 48–57. <https://doi.org/10.1109/MCI.2014.2307227>
7. Hyvönen, E. (2012). *Publishing and Using Cultural Heritage Linked Data on the Semantic Web*. Synthesis Lectures on the Semantic Web. <https://doi.org/10.2200/S00452ED1V01Y201205WBE003>
8. Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). *Web data extraction, applications and techniques: A survey*. Knowledge-Based Systems, 70, 301–323. <https://doi.org/10.1016/j.knosys.2014.07.007> <https://www.sciencedirect.com/science/article/pii/S0950705114002767>
9. Laender, A. H. F., Ribeiro-Neto, B., Silva, A. S., & Teixeira, J. S. (2002). *A brief survey of web data extraction tools*. ACM SIGMOD Record, 31(2), 84–93. <https://doi.org/10.1145/565117.565137> , <https://dl.acm.org/doi/10.1145/565117.565137>
10. Mitchell, R. (2018). *Web Scraping with Python: Collecting More Data from the Modern Web*. O'Reilly Media. <https://www.oreilly.com/library/view/web-scraping-with/9781491985564/>

11. Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison-Wesley.
<https://www.pearson.com/en-us/subject-catalog/p/modern-information-retrieval/P200000006694>
<https://www.sciencedirect.com/book/9780321416919/modern-information-retrieval>
12. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* Cambridge University Press. <https://nlp.stanford.edu/IR-book/> □
<https://www.cambridge.org/core/books/introduction-to-information-retrieval/0F4D8D44C7B7F32C8F5A7F0C8E5D2F07>
13. Gilliland, A. J. (2016). *Introduction to Metadata*. Getty Research Institute. <https://www.getty.edu/publications/intrometadata/>
<https://www.getty.edu/publications/virtuallibrary/160606018X.html>
14. Hyvönen, E. (2012). *Publishing and Using Cultural Heritage Linked Data on the Semantic Web*. Synthesis Lectures on the Semantic Web.
<https://doi.org/10.2200/S00452ED1V01Y201205WBE003>.
<https://www.morganclaypool.com/doi/abs/10.2200/S00452ED1V01Y201205WBE003>
15. Doan, A., Halevy, A., & Ives, Z. (2012). *Principles of Data Integration*. Morgan Kaufmann. <https://www.sciencedirect.com/book/9780124160446/principles-of-data-integration>
16. Candela, L., Castelli, D., & Pagano, P. (2011). *On-demand virtual digital libraries: The OpenDLib approach*. *International Journal on Digital Libraries*, 11, 77–93.
<https://doi.org/10.1007/s00799-010-0068-8>
<https://link.springer.com/article/10.1007/s00799-010-0068-8>
17. Marlow, C., Naaman, M., Boyd, D., & Davis, M. (2006). *HT06, tagging paper, taxonomy, Flickr, academic article, to read*. *Proceedings of the ACM Conference on Hypertext and Hypermedia*. <https://doi.org/10.1145/1149941.1149949> ,
<https://dl.acm.org/doi/10.1145/1149941.1149949>
18. Hyvönen, E. (2012). *Publishing and Using Cultural Heritage Linked Data on the Semantic Web*. Synthesis Lectures on the Semantic Web.

<https://doi.org/10.2200/S00452ED1V01Y201205WBE003>

<https://www.morganclaypool.com/doi/abs/10.2200/S00452ED1V01Y201205WBE003>

19. Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook*. Springer. <https://doi.org/10.1007/978-1-4899-7637-6>

<https://link.springer.com/book/10.1007/978-1-4899-7637-6>

20. Zubiaga, A., Liakata, M., Procter, R., Hoi, G. W. S., & Tolmie, P. (2016). *Analysing how people orient to and spread rumours in social media*. *Information Processing & Management*, 52(1), 124–145.

<https://doi.org/10.1016/j.ipm.2015.04.006>

<https://www.sciencedirect.com/science/article/pii/S030645731500081X>

21. Tani, A., Candela, L., & Castelli, D. (2013). *Dealing with metadata quality in digital libraries*. *Lecture Notes in Computer Science*, 8092, 136–145. □

https://doi.org/10.1007/978-3-642-40501-3_15 □

https://link.springer.com/chapter/10.1007/978-3-642-40501-3_15

22. O'Hagan, A., & Mangiron, C. (2013). *Game localization: Translating for the global digital entertainment industry*. John Benjamins Publishing.

<https://benjamins.com/catalog/btl.106>, <https://doi.org/10.1075/btl.106>

23. K. Lerman (2024) “Social Information Processing in Social News Aggregation” (arXiv / конференційна розпраця) <https://arxiv.org/pdf/cs.CY/0703087>

24. M. Karimi et al. (2018) “News recommender systems – Survey and roads ahead” (*Information Processing & Management*, 2018)

https://www.sciencedirect.com/science/article/abs/pii/S030645731730153X?utm_source=chatgpt.com

25. P. Koukaras et al.(2024) “Social Media Types: Introducing a Data Driven Taxonomy” (paper / PDF)

https://www.i.hu.edu.gr/tjortjis/Social%20Media%20Types%20Introducing%20a%20Data%20Driven%20Taxonomy%20%283%29.pdf?utm_source=chatgpt.com

26. J. Liu et al. (2008) “What do they think?: aggregating local views about news” (*ACM Conference paper*, 2008)

https://dl.acm.org/doi/10.1145/1367497.1367636?utm_source=chatgpt.com

27. X. Meng et al. (2023) "A Survey of Personalized News Recommendation" (Springer, 2023) https://link.springer.com/article/10.1007/s41019-023-00228-5?utm_source=chatgpt.com
28. E. Ferrara, A. Abrahão, et al. (2014) „Web Data Extraction, Applications and Techniques: A Survey” (2014). [Emilio Ferrara](https://www.emilio.ferrara.name/wp-content/uploads/2011/07/survey-csur.pdf)
Посилання (PDF): <https://www.emilio.ferrara.name/wp-content/uploads/2011/07/survey-csur.pdf>. [Emilio Ferrara](https://www.emilio.ferrara.name/wp-content/uploads/2011/07/survey-csur.pdf)
29. S. Chakrabarti et al. (1999) „Focused crawling: a new approach to topic-specific Web search” (1999) [ScienceDirect](https://www.sciencedirect.com/science/article/pii/S016763699900011X) Сторінка/опис статті: ScienceDirect (опис). [ScienceDirect](https://www.sciencedirect.com/science/article/pii/S016763699900011X)
30. M. Diligenti, F. Coetzee, S. Lawrence, M. Gori, M. Maguolo (2000) „Focused Crawling Using Context Graphs” (VLDB 2000) [VLDB](https://www.vldb.org/conf/2000/P527.pdf) PDF: [https://vldb.org/conf/2000/P527.pdf](https://www.vldb.org/conf/2000/P527.pdf). [VLDB](https://www.vldb.org/conf/2000/P527.pdf)
31. Blaž Novak (2004) „A Survey of Focused Web Crawling Algorithms” (SiKDD 2004 / PDF) [Aile3](https://www.aile3.ijs.si/dunja/SiKDD2004/Papers/BlazNovak-FocusedCrawling.pdf)
PDF: [https://aile3.ijs.si/dunja/SiKDD2004/Papers/BlazNovak-FocusedCrawling.pdf](https://www.aile3.ijs.si/dunja/SiKDD2004/Papers/BlazNovak-FocusedCrawling.pdf).
32. A. H. F. Laender et al. (2002) „A Brief Survey of Web Data Extraction Tools” (2002 / ACM-style survey) PDF: <https://www.cs.cmu.edu/~dgvinda/pdf/laender02brief.pdf>.
33. M. Han, P.-Y. Chen, et al. (2018) „Focused Crawling Through Reinforcement Learning” (2018 / PDF) PDF: <https://pierre.senellart.com/publications/han2018focused.pdf>
34. Lotfi, Srinivasan, Ertz, Latrous (2023) “Web Scraping Techniques and Applications: A Literature Review” https://www.researchgate.net/publication/367719780_Web_Scraping_Techniques_and_Applications_A_Literature_Review
35. Brown et al. (2024) “Web Scraping for Research: Legal, Ethical, and Methodological Guidance” <https://arxiv.org/abs/2410.23432>

36. Baumgartner (2024) “Web Data Extraction, Applications and Techniques: A Survey” [ResearchGate](https://www.researchgate.net/publication/228095933_Web_Data_Extraction_Applications_and_Techniques_A_Survey), https://www.researchgate.net/publication/228095933_Web_Data_Extraction_Applications_and_Techniques_A_Survey
37. Boni García (2024) “Exploring Browser Automation: A Comparative Study of Selenium, Cypress, Puppeteer and Playwright” (QUATIC 2024,). [Boni García https://bonigarcia.dev/slides/2024_QUATIC_Exploring_Browser_Automation_A_Comparative_Study_of_Selenium_Cypress_Puppeteer_and_Playwright.pdf](https://bonigarcia.dev/slides/2024_QUATIC_Exploring_Browser_Automation_A_Comparative_Study_of_Selenium_Cypress_Puppeteer_and_Playwright.pdf)
38. Vesa Morina (2023) “Evaluating and comparing web scraping tools and techniques for data collection” — (ResearchGate / 2023) https://www.researchgate.net/publication/369114323_Evaluating_and_comparing_web_scraping_tools_and_techniques_for_data_collection
39. Mine Dogucu (2021) “Web Scraping in the Statistics and Data Science Curriculum” — M. Doğuçü (2021) [Taylor & Francis Online https://www.tandfonline.com/doi/full/10.1080/10691898.2020.1787116](https://www.tandfonline.com/doi/full/10.1080/10691898.2020.1787116)
40. M. Maguire, (2002) “User Requirements Analysis: A Review of Supporting Methods” (IFIP / ACM proceedings, 2001/2002) [ACM Digital Library https://dl.acm.org/doi/10.5555/646869.709394](https://dl.acm.org/doi/10.5555/646869.709394)
41. A. van Lamsweerde (2024), “Goal-Oriented Requirements Engineering: A Guided Tour” [UC Homepages+1, https://homepages.uc.edu/~niunn/courses/RE-refs/GuidedTour01.pdf](https://homepages.uc.edu/~niunn/courses/RE-refs/GuidedTour01.pdf)
42. ISO/IEC/IEEE 29148:2018 — “Systems and software engineering — Life cycle processes — Requirements engineering” (стандарт) [ISO+1 https://www.iso.org/obp/ui/](https://www.iso.org/obp/ui/) (ISO entry for 29148) <https://standards.ieee.org/standard/29148-2018.html> (IEEE page)
43. Ian Sommerville, (2018) *Software Engineering* — розділ “Requirements engineering” (підручник) [Pearson+1: https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003258/9780137503148](https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003258/9780137503148)
44. Karl E. Wiegers,(2019) *Software Requirements* (практичний посібник) / доповнення: „Words to Avoid in Requirements” [SLAC Group+1 https://www-group.slac.stanford.edu/cdsoft/nlc_arch/nlc_requirements](https://www-group.slac.stanford.edu/cdsoft/nlc_arch/nlc_requirements)

[/RequirementsMethod.pdf](#), https://www.oir.caltech.edu/twiki_oir/pub/Keck/NGAO/SystemsEngineeringGroup/Wiegers_Words_to_Avoid_Requirements.pdf

45. W. Maalej et al., (2024). “On the Automated Processing of User Feedback”
<https://arxiv.org/abs/2407.15519>

46. S. Kiryakos 2015) *A Model for Aggregating Manga Metadata Across Institutions* (магістерська/наукова праця, 2015). [Слис Тсукаба+1](#)

47. S. Kiryakos (2017). Aggregating Metadata from Heterogeneous Pop Culture Data Providers (focus on manga) — Dublin Core Conference / paper (2017).
[DCPapers](#)

48. Огляд/систематичний огляд web-scraping (2017). *A Systematic Review of Web Scraping* (SSRN [SSRN](#)

49. «Demystified Overview of Data Scraping» — оглядова стаття (ResearchGate, 2025). [ResearchGate](#)

50. «Content-based Filtering and Web Scraping in Website for Recommended Anime» (conference / journal paper) [ResearchGate+1](#)

51. «Metadata Scraping Using Programmable Customized ..» (2024, PDF) —
[IASJ](#)

52. Архітектурні та практичні поради щодо етичного/правового скрейпінгу — Web Scraping for Research: Legal, Ethical, Institutional.. (arXiv / 2024).
[arXiv](#)

53. Розроблено автором на основі власних досліджень за допомогою інструментів комп’ютерної графіки та засоби штучного інтелекту. DALL·E / MidJourney / Stable Diffusion / https://deevid.ai/ai-image-generator?utm_source=google&utm_medium=cpc&utm_campaign=t-model-midjourney&utm_term=midjourney-image&utm_id=image-nano&utm_content=pc&gad_source=1&gad_campaignid=23173690772&gbraid=0AAAq898mOeQn-1wR30ZUqz7vE27q1hx&gclid=CjwKCAiAl-JBhBjEiwAn3rN7UdM-9lqJdO2TtfDTgX7eTw_qsbpGk1jY5jTKa64hb1lu-hO8Z3jhxoCayUQAvD_BwE

Копії обов'язкових креслень (Презентація)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

ДИПЛОМНА РОБОТА

на ступінь вищої освіти магістр
із спеціальності F3 Комп'ютерні науки

«Веб-сайт для колекціонування MANGA із застосуванням скрейпінгу»

Виконав: студент 6 курсу, групи КНДМ-61

Багажков Дмитро Сергійович

Керівник: д.т.н., доцент Катков Ю.І.

Київ – 2025

1

Загальна характеристика дипломної роботи

- **Мета роботи:** Метою дипломної роботи є розробка та обґрунтування архітектури веб-сайту для колекціонування манґи, що забезпечує автоматизований збір, систематизацію, пошук і фільтрацію контенту на основі технологій веб-скрейпінгу, індексації та обробки метаданих з метою підвищення ефективності відбору релевантних творів користувачами.
- **Об'єкт дослідження:** Процеси пошуку, збирання, обробки та представлення інформації про манґу у веб-системах колекціонування цифрового контенту..
- **Предмет дослідження:** Методи та програмні засоби веб-скрейпінгу, обробки метаданих, індексації, пошуку та фільтрації контенту, що застосовуються у веб-сайтах для колекціонування манґи..
- **Наукове завдання роботи** полягає у формуванні та дослідженні комплексного підходу до побудови веб-системи колекціонування манґи, який поєднує автоматизований збір даних із зовнішніх джерел, уніфікацію та збагачення метаданих, а також реалізацію ефективних механізмів пошуку, фільтрації та персоналізованого доступу до контенту..

2

Постановка завдання дипломної роботи

Проблема полягає в тому, що колекціонування манги перетворилося на поширене хобі з глобальною спільнотою ентузіастів. Оскільки цифрові платформи та онлайн-бази даних стають основним джерелом інформації для колекціонерів, ефективний пошук, агрегування та управління цими даними стають вирішальними для зручності користувача. Завдяки впровадженню IT-технологій, зокрема автоматизованих методів збору даних (веб-скрейпінгу) та інтелектуального аналізу, необхідно створити потужну систему, яка трансформувє підхід до колекціонування. Зосередження на пошуку технічних рішень, що дозволяють забезпечити повноту даних, гнучкість пошуку, масштабованість та економічну ефективність процесів обробки інформації в умовах швидко оновлюваного контенту та різноманітних форматів публікацій, є актуальною науково-практичною задачею.

Завдання роботи:

1. Провести аналіз застосування інтелектуальних методів пошуку та фільтрації контенту у веб-сайтах для колекціонування манги з використанням технологій веб-скрейпінгу, індексації та обробки метаданих
2. Обґрунтувати доцільність та дослідити ефективність інтелектуальних методів пошуку, класифікації та рекомендації манга-контенту на основі зібраних даних та аналіз їх впливу на зручність користування веб-сайтом
3. Розробити практичні рекомендації щодо впровадження інтелектуальних методів пошуку та агрегування даних у веб-сайті для колекціонування манги з використанням веб-скрейпінгу, семантичного пошуку та механізмів персоналізації.

3

1 АНАЛІЗ ЗАСТОСУВАННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА ФІЛЬТРАЦІЇ КОНТЕНТУ У ВЕБ-САЙТАХ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНГИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ВЕБ-СКРЕЙПІНГУ, ІНДЕКСАЦІЇ ТА ОБРОБКИ МЕТАДАН

1. Визначено ключову роль веб-скрейпінгу як базового механізму автоматизованого наповнення манга-каталогу, що забезпечує актуальність, повноту та масштабованість даних з різнорідних веб-джерел.
2. Обґрунтовано необхідність уніфікації та обробки метаданих, оскільки дані з різних платформ мають розбіжності у форматах, назвах та структурі; нормалізація підвищує якість пошуку й фільтрації контенту.
3. Показано ефективність індексації та фасетної фільтрації, які значно скорочують час пошуку та дозволяють реалізувати багатокритеріальний відбір манги (жанр, автор, статус, рік, демографія).
4. Виявлено обмеження традиційного ключового пошуку, що не враховує семантику описів; доведено доцільність застосування інтелектуальних методів (NLP, семантичний аналіз) для підвищення релевантності результатів.
5. Сформовано вимоги до сучасної архітектури веб-сайту колекціонування манги, яка поєднує веб-скрейпінг, обробку метаданих, індексацію та інтелектуальні механізми пошуку і персоналізації.

4

2 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ, КЛАСИФІКАЦІЇ ТА РЕКОМЕНДАЦІЇ МАНґА-КОНТЕНТУ НА ОСНОВІ ЗІБРАНИХ ДАНИХ ТА АНАЛІЗ ЇХ ВПЛИВУ НА ЗРУЧНІСТЬ КОРИСТУВАННЯ ВЕБ-САЙТОМ.

1. Підвищення релевантності пошуку. Запровадження індексації та семантичного аналізу метаданих забезпечило точніший пошук манґа-контенту порівняно з традиційним пошуком за ключовими словами.
2. Ефективна класифікація контенту. Автоматизована обробка та уніфікація метаданих з різних джерел дозволила коректно групувати манґу за жанрами, авторами та тематичними ознаками.
3. Покращення якості рекомендацій. Використання інтелектуальних рекомендаційних алгоритмів підвищило персоналізацію підбору контенту відповідно до вподобань користувачів.
4. Зменшення часу доступу до потрібного контенту. Поєднання фільтрації, фасетної навігації та інтелектуального пошуку скоротило час знаходження релевантної манґи в каталозі.
5. Позитивний вплив на зручність користування (UX). Інтеграція інтелектуальних методів пошуку, класифікації та рекомендацій суттєво покращила загальний користувацький досвід веб-сайту та задоволеність користувачів..

5

3. РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВПРОВАДЖЕННЯ ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ ПОШУКУ ТА АГРЕГУВАННЯ ДАНИХ У ВЕБ-САЙТІ ДЛЯ КОЛЕКЦІОНУВАННЯ МАНґИ

1. Запропоновано комплексну архітектуру веб-сайту для колекціонування манґи, що поєднує модулі веб-скрейпінгу, обробки метаданих, індексації та інтелектуального пошуку.
2. Розроблено практичні рекомендації щодо впровадження модуля веб-скрейпінгу, який забезпечує автоматизований збір, оновлення та уніфікацію даних з зовнішніх манґа-ресурсів з урахуванням етичних і технічних обмежень.
3. Обґрунтовано доцільність використання семантичного та фасетного пошуку, що підвищує релевантність результатів і дозволяє здійснювати гнучку навігацію за жанрами, авторами, статусом та іншими характеристиками.
4. Сформовано рекомендації щодо інтеграції персоналізованих механізмів, зокрема контентно-орієнтованих рекомендацій, які базуються на аналізі метаданих і вподобань користувача.
5. Визначено напрями подальшого розвитку системи, включно з масштабуванням бази даних, удосконаленням алгоритмів пошуку та розширенням функціональності користувацького інтерфейсу.

6

Висновки

1. **Проаналізовано існуючі веб-платформи для колекціонування манґи.** Виявлено їхні функціональні обмеження та визначено нішу для спеціалізованого рішення з глибокою каталогізацією і персоналізацією.
2. **Запропоновано архітектуру веб-сайту для колекціонування MANGA.** Архітектура інтегрує веб-скрейпінг, індексацію та обробку метаданих для автоматизованого наповнення та оновлення каталогу.
3. **Розроблено модель уніфікації та збагачення метаданих.** Забезпечено нормалізацію, дедуплікацію та узгодження даних з різних джерел для підвищення якості пошуку і фільтрації.
4. **Реалізовано інтелектуальні механізми пошуку та фільтрації контенту.** Поєднання фасетного та семантичного пошуку підвищує релевантність результатів і зручність користування системою.
5. **Обґрунтовано ефективність персоналізованих рекомендацій.** Доведено доцільність використання контентно-орієнтованих рекомендацій для покращення користувацького досвіду та відкриття нових творів.

7

Апробація результатів

В статті:

Багажков Д.С. (2025) ВЕБ-САЙТ ДЛЯ КОЛЕКЦІОНУВАННЯ MANGA: ВПРОВАДЖЕННЯ ТА ОПТИМІЗАЦІЯ WEB-SCRAPING // Наукові записки Державного університету інформаційно-комунікаційних технологій №2, 2025, Подано до друку.

<https://journals.dut.edu.ua/index.php/sciencenotes/issue/view/186>

В тезах:

1. Багажков Д.С. (2025) РОЗРОБКА ВЕБ-САЙТУ ДЛЯ КОЛЕКЦІОНУВАННЯ MANGA ІЗЗАСТОСУВАННЯМ СКРЕЙПИНГУ VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». Збірник тез 15 квітня 2025. – К.: ДУІКТ, ст. 19-23. https://duikt.edu.ua/uploads/p_2779_40288420.pdf

2. Багажков Д.С. (2025) ВЕБ-СКРЕЙПІНГ У СИСТЕМІ КОЛЕКЦІОНУВАННЯ КОНТЕНТУ НА ГІБРИДНИХ РІШЕННЯХ HPE // Сучасні досягнення компанії Hewlett Packard Enterprise в галузі IT та нові можливості їх вивчення і застосування/ Збірник тез 11 грудня 2025. – К.: ДУІКТ, подане для друку - https://duikt.edu.ua/ua/news-1-576-14841-duikt-zaproshue-na-mizhnarodnu-naukovo-tehnichnu-konferenciyu-suchasni-dosyagnennya-kompanii-hewlett-packard-enterprise-v-galuzi-it-ta-novi-mozhливosti-ix-vivchennya-i-zastosuvannya_kafedra-shtuchnogo-intelektu

8