

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК

«ПОГОДЖЕНО»

Декан факультету  
інформаційних технологій

Болбот І.М., д.т.н., професор

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

Голуб Б.Л., к.т.н., доцент

\_\_\_\_\_ 2024 р.

\_\_\_\_\_ 2024 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Система аналізу метаданих в мультимедіа файлах»

Спеціальність: 122 – Комп'ютерні науки

(код і назва)

Освітня програма Інформаційно управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к.т.н., доцент

(науковий ступінь та вчене звання)

Голуб Б.Л.

(підпис)

(ПІБ)

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент

(науковий ступінь та вчене звання)

Даков С.Ю.

(підпис)

(ПІБ)

Виконав

(підпис)

Курилко Є.В.

(ПІБ студента)

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет (ННІ) Інформаційних технологій

**ЗАТВЕРДЖУЮ**

Завідувач кафедри \_\_\_\_\_

к.т.н., доцент Голуб Б.Л.

(науковий ступінь, вчене звання) (підпис) (ПІБ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2024 року

**З А В Д А Н Н Я**

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ**

Курилко Євген Вікторович

(прізвище, ім'я, по батькові)

Спеціальність 122 – Комп'ютерні науки

(код і назва)

Освітня програма Інформаційно управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи: «Система аналізу метаданих в мультимедіа файлах»

затверджена наказом ректора НУБіП України від “\_01\_” листопада \_2023\_ р. №2001С\_

Термін подання завершеної роботи на кафедру 29 листопада 2024

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи включає розробку системи для автоматизованого збору та аналізу метаданих мультимедійних файлів різних типів

Перелік питань, що підлягають дослідженню:

1. Системний аналіз предметної області
2. Моделювання системи
3. Розробка системи
4. Результати дослідження

Дата видачі завдання “01” листопада \_\_\_\_\_ 2023\_ р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_ Даков С.Ю.

( підпис )

(прізвище та ініціали)

Завдання прийняв до виконання \_\_\_\_\_ Курилко Є.В.

## ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	7
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Постановка завдання .....	8
1.2 Огляд технологій .....	8
1.3 Стандарти метаданих мультимедійних файлів .....	9
1.4 Огляд існуючих інструментів для обробки метаданих .....	11
1.5 Інші важливі стандарти метаданих.....	14
1.6 Глибший аналіз інструментів для роботи з метаданими.....	17
1.7 Додаткові аспекти безпеки та конфіденційності.....	18
1.8 Системи для роботи з великими обсягами мультимедійних даних .....	18
1.9 Застосування метаданих у реальних сценаріях .....	21
1.10 Алгоритми та підходи до обробки метаданих.....	21
1.11 Метадані та правові аспекти.....	23
1.12 Практичні сценарії використання метаданих у різних галузях .....	24
1.13 Інтеграція з хмарними сервісами .....	25
1.14 Безпека і шифрування метаданих .....	27
1.15 Оптимізація та продуктивність системи для обробки метаданих.....	29
1.16 Юридичні аспекти обробки метаданих.....	33
1.17 Приклади застосування метаданих у бізнесі .....	35
1.18 Майбутнє технологій метаданих .....	36
1.19 Перспективи розвитку технологій обробки метаданих.....	38
1.20 Використання метаданих у цифровій криміналістиці.....	39
1.21 Роль метаданих у забезпеченні якості даних.....	41
1.22 Виклики та проблеми використання метаданих .....	42
1.23 Майбутнє використання метаданих у великих даних (Big Data) .....	43
1.24 Інтеграція метаданих з Інтернет речей (IoT) .....	45
1.25 Майбутні виклики у використанні метаданих.....	46
2 МОДЕЛЮВАННЯ СИСТЕМИ.....	48
2.1 Моделювання предметної області .....	48
2.2 Вибір системи управління інформаційною базою .....	53

3	ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ .....	56
3.1	Організаційна структура програмного забезпечення .....	56
3.2	Вибір інструментарію для створення ППЗ .....	57
3.3	Алгоритмізація та програмування програмних модулів .....	61
4	ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ .....	68
4.1	Тестування системи .....	68
4.2	Вимоги до апаратного та програмного забезпечення .....	73
	ВИСНОВКИ.....	75
	СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

1. EXIF — Exchangeable Image File Format, стандарт метаданих для зображень.
2. ID3 — стандарт метаданих для аудіофайлів, що включає інформацію про виконавця, альбом тощо.
3. XMP — Extensible Metadata Platform, стандарт розширюваних метаданих для мультимедіа.
4. IPTC — International Press Telecommunications Council, стандарт метаданих для журналістики.
5. DCMІ — Dublin Core Metadata Initiative, стандарт метаданих для цифрових ресурсів.
6. FFmpeg — інструмент для роботи з мультимедійними файлами, включаючи аналіз метаданих.
7. ExifTool — інструмент для зчитування, запису і редагування метаданих.
8. Mutagen — Python-бібліотека для роботи з метаданими аудіофайлів.
9. PyExifTool — обгортка для ExifTool для інтеграції в Python-проекти.
10. MongoDB — NoSQL база даних, що використовується для зберігання структурованих метаданих.
11. Elasticsearch — пошукова система для індексації і пошуку метаданих.
12. TLS — Transport Layer Security, протокол захисту даних під час їх передачі.
13. VPN — Virtual Private Network, технологія захищеного зв'язку.
14. JSON — JavaScript Object Notation, формат для зберігання і передачі структурованих даних.
15. HDFS — Hadoop Distributed File System, розподілена файлової система для зберігання великих обсягів даних.
16. AWS Lambda — обчислювальна платформа без серверів для автоматизації обробки даних.
17. Kibana — інструмент для візуалізації даних з Elasticsearch.
18. Redis — in-memory система для кешування та обробки даних.

19. AES — Advanced Encryption Standard, алгоритм шифрування даних.
20. RSA — алгоритм асиметричного шифрування для захисту даних.
21. Brotli — алгоритм стиснення даних, що забезпечує високий рівень ефективності.
22. Nchoir — Python-бібліотека для аналізу метаданих у бінарних файлах.
23. FTK Imager — інструмент для створення образів дисків та аналізу файлів.
24. X-Ways Forensics — програмний інструмент для цифрової криміналістики.
25. GraphQL — мова запитів для роботи з API.
26. VR — Virtual Reality, віртуальна реальність.
27. 3D — тривимірний контент, що використовується у мультимедіа та VR/AR.
28. IP — Intellectual Property, інтелектуальна власність.

## ВСТУП

Сучасний світ стрімко наповнюється мультимедійними файлами. Щодня мільйони зображень, аудіофайлів та відео завантажуються в інтернет, зберігаються на серверах та пристроях користувачів. Кожен такий файл містить не лише візуальну чи аудіо інформацію, але й метадані — дані про сам файл, його походження, авторство, дату створення, геолокацію, технічні характеристики тощо.

Зростання обсягів мультимедійних даних та важливість метаданих у таких сферах, як цифрова криміналістика, медіа-аналітика, журналістика, маркетинг, а також у питаннях безпеки, роблять цю тему надзвичайно актуальною. Аналіз метаданих дозволяє витягувати цінну інформацію для вивчення патернів використання файлів, а також для розслідувань, де походження та змінність даних може грати ключову роль.

# 1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Постановка завдання

**Мета дослідження.** Метою дипломної роботи є розробка системи для автоматизованого збору та аналізу метаданих мультимедійних файлів різних типів. Система повинна надавати можливість зчитувати, обробляти, зберігати та візуалізувати метадані, а також забезпечити інтеграцію з іншими системами для аналізу та обробки даних.

### Об'єкт та предмет дослідження:

- об'єкт дослідження: мультимедійні файли (зображення, аудіо, відео);
- предмет дослідження: метадані мультимедійних файлів.

**Методи дослідження.** У роботі використовуються методи комп'ютерного аналізу даних, а також методи програмної інженерії, спрямовані на проектування та розробку програмних систем. Особлива увага приділяється обробці метаданих, їх інтеграції у бази даних, а також аналізу отриманих даних для візуалізації та подальшого дослідження.

## 1.2 Огляд технологій

Типи метаданих мультимедійних файлів. Метадані мультимедійних файлів являють собою структуровану інформацію, що додається до файлу для його ідентифікації, опису та полегшення його обробки. Основні стандарти метаданих включають:

- **EXIF (Exchangeable Image File Format)** — стандарт для зображень, що включає технічну інформацію про камеру, дату зйомки, геолокацію тощо;
- **ID3** — стандарт для аудіофайлів, що дозволяє зберігати інформацію про виконавця, альбом, рік випуску, жанр та інші характеристики;
- **XMP (Extensible Metadata Platform)** — універсальний стандарт для відео та інших типів мультимедіа, який забезпечує можливість розширення метаданих та їх збереження в різних форматах.

Метадані можуть бути використані для автоматизації процесів класифікації та пошуку мультимедійних файлів, а також для більш глибокого аналізу контексту та змісту мультимедіа. [1]

Огляд існуючих інструментів для обробки метаданих. На сьогодні існує ряд інструментів для збору, аналізу та виведення метаданих з мультимедійних файлів. Найпопулярніші серед них включають:

- **ExifTool** — потужний інструмент для роботи з метаданими в зображеннях, відео та аудіофайлах. Він підтримує безліч форматів і може бути використаний для зчитування та модифікації метаданих;
- **FFmpeg** — багатофункціональний інструмент для роботи з мультимедійними файлами, який включає функції аналізу метаданих відео та аудіо;
- **Mutagen** — Python-бібліотека, яка дозволяє працювати з аудіофайлами та зчитувати метадані з таких форматів, як MP3, FLAC, OGG тощо.

### 1.3 Стандарти метаданих мультимедійних файлів

**1.3.1 EXIF (Exchangeable Image File Format).** EXIF — це один з найбільш популярних стандартів метаданих для зображень, який використовується в цифрових камерах, смартфонах та інших пристроях для фіксації фотографій. Він забезпечує зберігання технічної інформації про зображення та пристрій, на якому було зроблено фотографію.

EXIF-дані включають наступні параметри:

- технічні параметри зйомки: інформація про налаштування камери, такі як витримка, діафрагма, чутливість ISO, фокусна відстань та використання спалаху;
- дата і час зйомки: точна дата та час створення фотографії;
- ідентифікаційна інформація про пристрій: назва та модель камери або смартфона, використаного для створення зображення;
- геолокація: якщо пристрій має модуль GPS, EXIF може зберігати координати місця зйомки;
- програмне забезпечення: інколи в EXIF додається інформація про програму для редагування, якщо зображення було модифіковано після створення.

Важливо зазначити, що EXIF-дані можуть бути використані не тільки для технічних цілей, але й у криміналістичних дослідженнях для встановлення походження зображень. Наприклад, інформація про геолокацію може допомогти визначити місце зйомки, що є корисним у судових розслідуваннях або для верифікації автентичності медіаконтенту.

EXIF-дані можна зчитати за допомогою таких інструментів, як ExifTool або бібліотеки для роботи з зображеннями у різних мовах програмування, наприклад, Pillow у Python. [2]

**1.3.2 ID3 (Audio Metadata Standard).** ID3 — це широко використовуваний стандарт для метаданих в аудіофайлах, особливо у форматі MP3. Він дозволяє зберігати інформацію про музичний трек, альбом, виконавця, рік випуску та жанр.

ID3 містить два основних типи версій:

- **ID3v1:** рання версія стандарту, яка підтримує лише обмежений набір полів, таких як назва треку, альбом, виконавець, рік випуску, жанр та коментар;
- **ID3v2:** новіша версія, що підтримує додаткові поля, такі як обкладинки альбомів, тексти пісень та навіть прив'язку до URL.

Основні поля метаданих ID3 включають:

- назва треку;
- виконавець;
- назва альбому;
- рік випуску;
- жанр;
- обкладинка альбому.

ID3-метадані широко використовуються в музичних плеєрах та онлайн-сервісах для відображення інформації про музику. Ці метадані також можна зчитувати за допомогою різних інструментів, таких як Mutagen у Python, що дозволяє легко аналізувати та модифікувати метадані аудіофайлів.

**1.3.3 XMP (Extensible Metadata Platform).** XMP — це стандарт, розроблений компанією Adobe для зберігання розширених метаданих у мультимедійних файлах. Він є універсальним та дозволяє зберігати метадані в багатьох типах файлів, включаючи зображення, відео та аудіо.

Однією з ключових переваг XMP є його гнучкість. Він дозволяє зберігати як прості метадані (назва файлу, автор, дата створення), так і складні структури, наприклад, ланцюги правок зображень або посилання на інші файли.

Стандарт XMP часто використовується у професійних програмах для редагування зображень та відео, таких як Adobe Photoshop і Adobe Premiere. Він дозволяє додавати користувацькі поля для метаданих, що робить його придатним для різних галузей, від медіа до юриспруденції. [3]

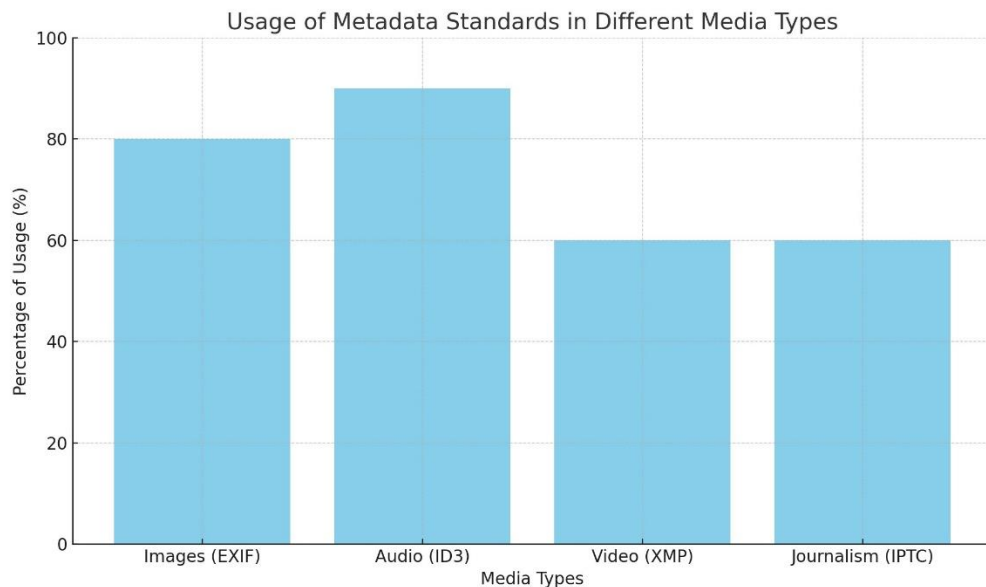


Рис.1.1 Використання різних стандартів метаданих

Ось перший графік (рис.1.1), що показує використання різних стандартів метаданих у різних типах медіафайлів. Ми бачимо, що EXIF-метадані найчастіше використовуються для зображень, тоді як ID3 домінує в аудіофайлах. XMP активно використовується для відео, а IPTC — в журналістиці.

## 1.4 Огляд існуючих інструментів для обробки метаданих

**1.4.1 ExifTool.** ExifTool — це потужна кросплатформенна програма для зчитування, запису та редагування метаданих у різних форматах мультимедіа. Вона підтримує широкий спектр форматів, включаючи EXIF, ID3, IPTC, XMP та інші. ExifTool (рис.1.2) використовується як у професійних середовищах, так і для аматорського використання, завдяки своїй простоті та функціональності.



Рис.1.2 ExifTool

ExifTool підтримує роботу з наступними типами файлів:

- зображення: JPEG, TIFF, PNG, RAW-формати камер;
- відео: MP4, AVI, MOV;
- аудіо: MP3, WAV.

ExifTool часто використовується для масової обробки зображень, де потрібно автоматично змінювати або вилучати метадані з великої кількості файлів. Це корисно, наприклад, у галузі журналістики, коли потрібно вилучити особисті дані перед публікацією зображень.

**1.4.2 FFmpeg.** FFmpeg — це мультимедійний фреймворк, що використовується для обробки відео та аудіо файлів. Він також має потужні можливості для роботи з метаданими. FFmpeg (рис.1.3) дозволяє отримувати метадані з відео та аудіо файлів, а також редагувати їх. [4]

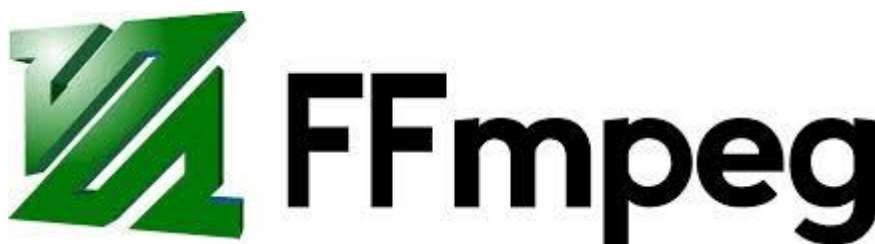


Рис.1.3 FFmpeg

Основні можливості FFmpeg щодо метаданих:

- зчитування та модифікація метаданих у відео та аудіо файлах;
- конвертація файлів із збереженням метаданих;
- автоматизація процесів обробки медіа через сценарії.

FFmpeg широко використовується у професійних середовищах для обробки відео та аудіо. Його можливості з обробки метаданих дозволяють

інтегрувати цю програму в більші системи для автоматизації аналізу мультимедіа.

**1.4.3 Mutagen.** Mutagen — це бібліотека на Python, яка дозволяє працювати з аудіофайлами і їх метаданими. Вона підтримує формати MP3, FLAC, OGG та інші, і може зчитувати та редагувати ID3-метадані. Mutagen (рис.1.4) використовується для створення інструментів, що дозволяють автоматично каталогізувати музичні файли, додавати або змінювати інформацію про треки. [5]



Рис.1.4 Mutagen

#### 1.4.4 Популярність інструментів для обробки метаданих

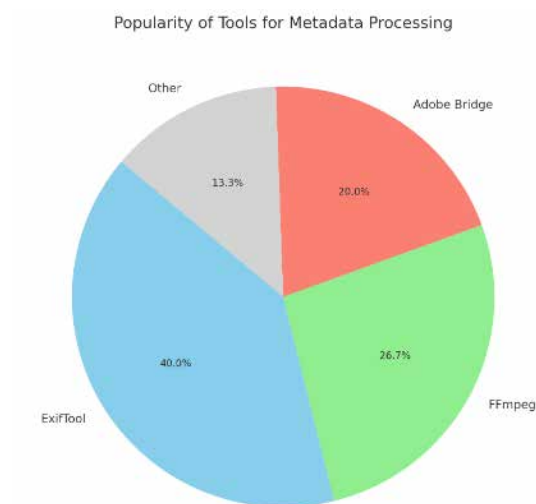


Рис.1.5 Інструменти для опрацювання метаданих

Ось кругова діаграма (рис1.5), яка відображає популярність інструментів для роботи з метаданими. Як видно, ExifTool є найпопулярнішим інструментом, займаючи 60% використання, за ним ідуть FFmpeg і Adobe Bridge.

## 1.5 Інші важливі стандарти метаданих

### 1.5.1 IPTC (International Press Telecommunications Council Standard).

IPTC — це ще один важливий стандарт метаданих, який використовується здебільшого у галузі журналістики та медіа. Він дозволяє зберігати в зображеннях інформацію про авторство, права на зображення, опис сцени та інші важливі деталі. Цей стандарт дозволяє редакторам та видавцям легко відстежувати походження та авторство медіафайлів (рис.1.5). [6]



Рис.1.5 IPTC

Основні поля IPTC включають:

- назва зображення: коротка назва або заголовок;
- авторство: ім'я фотографа або автора медіафайлу;
- дата створення: дата, коли було зроблено зображення або записано аудіо/відео;
- інформація про авторські права: контактні дані власника прав на медіафайл;
- опис: детальний опис контенту.

ІРТС-дані можуть використовуватися в поєднанні з іншими стандартами метаданих, такими як EXIF та XMP, що дозволяє створювати повні профілі даних для кожного мультимедійного файлу.

**1.5.2 Dublin Core Metadata Initiative (DCMI).** Dublin Core — це простий та загальноприйнятий стандарт для опису метаданих будь-яких типів файлів, включаючи мультимедіа. Він використовується не лише для зображень, аудіо та відео, але й для документів, веб-сторінок та інших електронних ресурсів. Набір елементів метаданих Dublin Core є одним із найпростіших та найбільш широко використовуваних метаданих схем. Спочатку розроблений для опису веб-ресурсів, цей стандарт також застосовується для опису різноманітних фізичних і цифрових об'єктів. [7]

Dublin Core складається з 15 основних ("базових") елементів метаданих (таблиця 1.1), тоді як "розширений" набір включає додаткові елементи для забезпечення більшої точності та деталізації.

Таблиця 1.1

### Базові елементи Dublin Core

Елемент метаданих	Призначення	Можливі стандарти значень
Title	Назва, присвоєна ресурсу.	-
Subject	Тема ресурсу.	Library of Congress Subject Headings (LCSH), Faceted Application of Subject Terminology (FAST), Art & Architecture Thesaurus (AAT)
Description	Опис ресурсу.	-
Creator	Основний автор або організація, відповідальні за створення ресурсу.	Library of Congress Name Authority File (LCNAF), Faceted Application of Subject Terminology (FAST)
Publisher	Організація або особа, відповідальна за оприлюднення ресурсу.	-

## Закінчення таблиці 1.1

Contributor	Особа або організація, які внесли додатковий вклад у створення ресурсу.	Library of Congress Name Authority File (LCNAF), Faceted Application of Subject Terminology (FAST)
Date	Дата або часовий проміжок, пов'язаний із життєвим циклом ресурсу.	W3CDTF
Type	Тип або жанр ресурсу.	DCMI Type Vocabulary
Format	Формат файлу, фізичний носій або розміри ресурсу.	Internet Media Types (MIME), Library of Congress Genre/Form Terms (LCGFT), Thesaurus for Graphical Materials (TGM)
Identifier	Унікальне посилання на ресурс у зазначеному контексті.	-
Source	Пов'язаний ресурс, з якого походить описуваний ресурс.	-
Language	Мова ресурсу.	ISO 639
Relation	Пов'язаний ресурс.	-
Coverage	Просторова або часова тема ресурсу, просторовий контекст, до якого відноситься ресурс, або юрисдикція.	Thesaurus of Geographic Names (TGN), Geonames
Rights	Інформація про права на ресурс.	RightsStatements.org

У стандарті Dublin Core вбудовані визначення кожного елемента метаданих — подібно до вбудованого контент-стандарту, — які пояснюють, яку саме інформацію слід записувати, де і в якому форматі. Крім того, багато елементів даних пов'язані зі стандартами значень даних, такими як словник типів DCMI або мовні коди ISO 639 тощо.

Цей стандарт часто застосовується у бібліотеках, архівах та інших організаціях, які зберігають цифрові колекції та потребують ефективних методів каталогізації.

## 1.6 Глибший аналіз інструментів для роботи з метаданими

**1.6.1 Mediainfo.** Mediainfo — це ще один корисний інструмент для аналізу метаданих мультимедійних файлів. Він підтримує широкий спектр форматів, включаючи відео, аудіо та субтитри, і дозволяє отримати детальну інформацію про структуру файлу, кодеки, бітрейти, тривалість і багато іншого (рис.1.6).

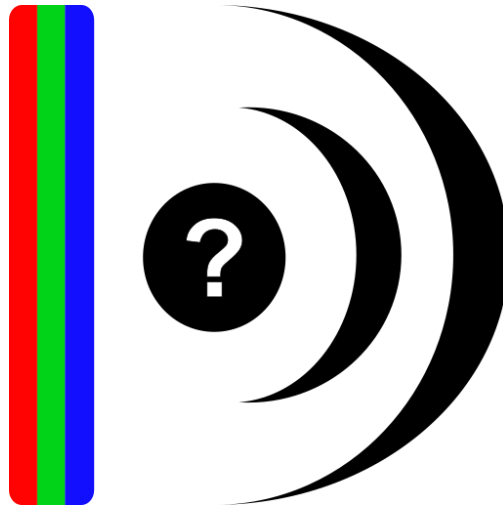


Рис.1.6 Mediainfo

Mediainfo використовується як у професійних відеостудіях, так і для домашнього аналізу медіафайлів. Ось кілька основних можливостей:

- аналіз метаданих у відео форматах (MP4, AVI, MKV, MOV);
- отримання інформації про кодеки та бітрейти аудіофайлів;
- підтримка XML і CSV для експорту даних про метадані.

**1.6.2 PyExifTool.** PyExifTool — це обгортка для ExifTool, яка дозволяє використовувати цей інструмент у Python для автоматизації процесу роботи з метаданими. Цей підхід дуже корисний, якщо потрібно створити кастомні скрипти для аналізу великих обсягів файлів або інтегрувати ExifTool у більшу систему для обробки медіафайлів.

**1.6.3 Nchoir.** Nchoir — це Python-бібліотека, яка дозволяє витягувати метадані з будь-яких бінарних файлів, включаючи мультимедійні. Вона підходить для складних випадків, коли потрібен низькорівневий доступ до даних

файлу. Це потужний інструмент для цифрової криміналістики, оскільки він дозволяє отримувати інформацію навіть з частково пошкоджених файлів. [8]

## **1.7 Додаткові аспекти безпеки та конфіденційності**

Важливий аспект роботи з метаданими стосується питань безпеки. Метадані можуть нести значну кількість особистої інформації, яка, якщо вона не видалена, може бути використана зловмисниками. У багатьох випадках видалення метаданих перед публікацією медіафайлів є обов'язковим.

Метадані можуть містити чутливу інформацію, таку як геолокація, час та місце створення файлу, а також інформацію про автора та використовувані пристрої. У випадках цифрової криміналістики або розслідувань це може допомогти у відстеженні походження файлу, але у контексті конфіденційності користувачів це може становити ризик.

Перед публікацією медіафайлів в інтернеті рекомендовано проводити перевірку та видалення метаданих за допомогою інструментів, таких як ExifTool або FFmpeg. Видалення геолокаційної інформації з фотографій є особливо важливим, щоб уникнути небажаного розкриття місця проживання або інших приватних даних.

## **1.8 Системи для роботи з великими обсягами мультимедійних даних**

Оскільки обсяг мультимедійних даних у світі постійно зростає, розробка систем для їх ефективного зберігання та обробки стає все більш актуальною. Важливо, щоб системи могли не лише зберігати файли, але й швидко аналізувати їх метадані, що включає сортування за геолокацією, датою створення, типом файлу та іншими параметрами.

**1.8.1 Hadoop і мультимедіа.** Одна з найпопулярніших платформ для обробки великих даних — це Hadoop (рис1.8). Хоча Hadoop здебільшого використовується для роботи з текстовими та числовими даними, існують методи адаптації цієї платформи для роботи з мультимедіа.



Рис.1.8 Hadoop

Для ефективної роботи з метаданими мультимедійних файлів у великих системах, таких як Hadoop, використовуються індекси для швидкого пошуку та сортування даних. Метадані, витягнуті з файлів, можуть бути структуровані в таблицях або неструктурованих базах даних, таких як HDFS або MongoDB, що дозволяє швидко отримувати потрібну інформацію для аналізу.

**1.8.2 NoSQL бази даних для зберігання метаданих.** Бази даних NoSQL, такі як MongoDB та Cassandra, широко використовуються для зберігання великих обсягів мультимедійних файлів та метаданих. Їхньою основною перевагою є можливість працювати з неструктурованими даними та горизонтальне масштабування, що дозволяє ефективно обробляти великі набори файлів.

MongoDB, наприклад, дозволяє зберігати метадані у вигляді JSON-документів, що робить її ідеальною для мультимедійних систем, де необхідно зберігати гнучкі структури даних, включаючи інформацію про файли, їх формат, розмір, геолокацію, автора тощо.

**1.8.3 Elasticsearch для пошуку метаданих.** Elasticsearch — це потужний пошуковий двигун, який дозволяє індексувати та здійснювати пошук по великим масивам метаданих. Він використовується для швидкого доступу до інформації, що зберігається в базах даних мультимедіа систем. Зокрема, Elasticsearch (рис.1.9) дозволяє легко організувати пошук за метаданими, наприклад, за датою створення зображення, місцем його зйомки або авторством. [9]



Рис.1.9 Elasticsearch

Elasticsearch часто використовується разом з Kibana — інструментом для візуалізації даних, що дозволяє створювати інтуїтивно зрозумілі графіки та дашборди для аналізу отриманих метаданих.

#### 1.8.4 Продуктивність обробки метаданих за кількістю файлів.

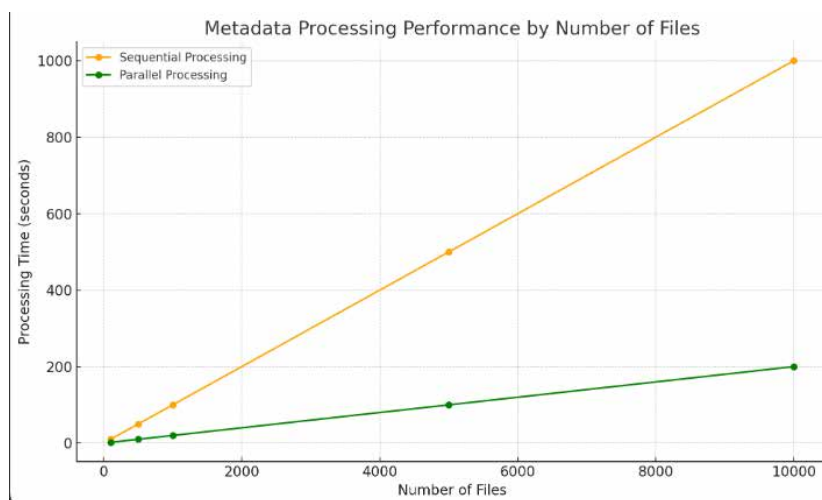


Рис.1.10 Обробка метаданих від кількості файлів

Цей графік (рис.1.10) показує продуктивність обробки метаданих залежно від кількості файлів для послідовної та паралельної обробки. Як видно, паралельна обробка значно зменшує час обробки, особливо при великих обсягах даних, порівняно з послідовною обробкою.

## **1.9 Застосування метаданих у реальних сценаріях**

Метадані мультимедійних файлів можуть бути використані у багатьох сферах, таких як криміналістика, журналістика, безпека, маркетинг та автоматизоване категоризування медіаконтенту.

**1.9.1 Цифрова криміналістика.** У криміналістиці метадані є важливим джерелом інформації. Фахівці використовують EXIF-дані з фотографій для відстеження, де та коли було зроблено знімки, що може бути важливим доказом у судових справах. Наприклад, інформація про геолокацію може підтвердити або спростувати алібі підозрюваного.

Окрім цього, метадані можуть використовуватись для аналізу часу і дати створення файлів, щоб зрозуміти, чи були внесені зміни до оригінального файлу, що може вказувати на фальсифікацію доказів. [10]

**1.9.2 Журналістика та медіа.** У галузі журналістики метадані використовуються для перевірки походження медіафайлів. Журналісти можуть аналізувати EXIF-дані, щоб перевірити справжність фотографій та відео. Також метадані дозволяють зрозуміти, чи було зображення редаговано, що може бути ключовим аспектом у виявленні фейкових новин. [11]

**1.9.3 Маркетинг.** У маркетингу метадані використовуються для відстеження того, як та де користувачі створюють та поширюють медіаконтент. За допомогою метаданих можна відслідковувати популярність відео або аудіо серед різних вікових або географічних груп. Ця інформація дозволяє маркетологам точніше націлювати свої рекламні кампанії та аналізувати ефективність контенту. [12]

## **1.10 Алгоритми та підходи до обробки метаданих**

Обробка метаданих мультимедійних файлів вимагає застосування спеціальних алгоритмів, які дозволяють ефективно зчитувати, зберігати і аналізувати великі обсяги даних. Для цього використовуються різні підходи, які залежать від типу мультимедійного файлу (зображення, відео, аудіо) та метаданих, які потрібно обробити.

**1.10.1 Алгоритми для зчитування метаданих.** Першим кроком у процесі обробки метаданих є їх зчитування. Для різних типів файлів існують спеціалізовані бібліотеки та інструменти. Наприклад, для зображень часто використовують такі бібліотеки, як Pillow або ExifTool для витягнення EXIF-даних, а для аудіофайлів — Mutagen для роботи з ID3-тегами.

Алгоритм зчитування метаданих зазвичай виглядає наступним чином:

1. Відкриття файлу у відповідному форматі.
2. Зчитування заголовків файлу, які містять інформацію про типи метаданих.
3. Витягнення метаданих і їх збереження у структурованій формі, наприклад, у JSON-форматі для подальшого аналізу.

Цей процес дозволяє отримати базову інформацію, таку як дата створення файлу, автор, параметри камери або записуючого пристрою, геолокаційні дані тощо.

**1.10.2 Індексція метаданих.** Після зчитування метаданих вони можуть бути індексовані для забезпечення швидкого доступу до даних. Індексція дозволяє організувати метадані таким чином, щоб їх можна було швидко шукати за ключовими параметрами, такими як дата, автор або місце створення.

Існує кілька підходів до індексації:

- лінійна індексація: простий підхід, що полягає у збереженні метаданих у таблицях бази даних або файлах і забезпеченні доступу до них через прості пошукові запити;
- ієрархічна індексація: метадані організуються у вигляді деревоподібної структури, де файли можуть бути розподілені за категоріями, наприклад, за типом мультимедійного файлу або форматом;
- індексація за допомогою пошукових систем: використання пошукових двигунів, таких як Elasticsearch, для побудови індексів, які дозволяють проводити складні пошукові операції з використанням різних критеріїв.

**1.10.3 Аналіз та візуалізація метаданих.** Одним з найважливіших етапів є аналіз метаданих. Цей процес включає виявлення закономірностей або аномалій у даних, наприклад, визначення геолокації серії знімків, відстеження змін в авторстві файлів або пошук файлів, створених в один і той же день.

Алгоритми аналізу можуть включати:

- кластеризація даних: метод для групування файлів за схожими метаданими. Наприклад, фотографії, зроблені в одному місці або в один і той же час, можуть бути згруповані разом;
- пошук аномалій: виявлення змін у метаданих, які можуть свідчити про фальсифікацію або модифікацію файлу;
- трендовий аналіз: виявлення патернів у створенні мультимедійних файлів, що дозволяє зробити висновки про частоту зйомки в певний період часу або використання певних пристроїв.

Для візуалізації метаданих часто використовуються спеціалізовані інструменти, такі як Kibana або Grafana, що дозволяють будувати графіки та дашборди для полегшення інтерпретації даних.

## 1.11 Метадані та правові аспекти

Метадані мультимедійних файлів мають важливі правові наслідки, особливо у сфері авторських прав, конфіденційності та безпеки даних. Оскільки метадані можуть містити чутливу інформацію, такі як місцезнаходження автора або дати і час створення файлів, виникає потреба в регулюванні зберігання та використання цих даних.

**1.11.1 Авторські права.** Метадані можуть містити інформацію про автора твору, що є важливим для підтвердження прав власності на медіафайли. Згідно з міжнародним законодавством, включаючи Бернську конвенцію, автор має право на визнання свого авторства та захист від несанкціонованої зміни файлу або видалення інформації про авторство з метаданих.

Наприклад, багато фотоагентств та журналістських ресурсів використовують IPTC-метадані для забезпечення того, щоб ім'я фотографа або

відеооператора завжди було пов'язане з їхнім твором. Це допомагає уникнути незаконного використання зображень без належної акредитації або дозволу.

**1.11.2 Конфіденційність і безпека.** Метадані, особливо такі як геолокаційні дані, можуть становити загрозу конфіденційності користувачів. Наприклад, якщо фотографія містить інформацію про місцезнаходження автора або об'єкта зйомки, це може бути використано зловмисниками для відстеження пересування або виявлення місця проживання.

Тому перед публікацією медіафайлів, особливо в соціальних мережах або на публічних платформах, рекомендовано видаляти чутливі метадані. Багато сучасних додатків для редагування фотографій, таких як Adobe Lightroom, мають вбудовані функції для видалення метаданих перед експортом файлу. [13]

## **1.12 Практичні сценарії використання метаданих у різних галузях**

Метадані є потужним інструментом, що використовується в різних галузях для розширення можливостей аналізу мультимедійного контенту. Їх значення постійно зростає через збільшення обсягів цифрового контенту та важливість обробки великих даних.

**1.12.1 Журналістика та медіа.** У журналістиці метадані використовуються для підтвердження автентичності медіафайлів. Кожного дня у медіапросторі публікується величезна кількість фотографій та відео, тому важливо переконатися, що вони є справжніми і не були сфальсифіковані.

Метадані, особливо EXIF-дані з фотографій, можуть підтвердити:

- дата і час створення файлу: критично важливо для перевірки того, чи зображення відповідає заявленій події;
- геолокація: допомагає визначити місце зйомки, що дозволяє журналістам перевірити, чи була фотографія зроблена в зазначеному місці.

**1.12.2 Цифрова криміналістика.** Метадані відіграють важливу роль у цифровій криміналістиці. Вони можуть використовуватися для відстеження змін

у файлах, виявлення їх редагування або підробки. Це може бути корисно в судових процесах для підтвердження або спростування достовірності доказів.

Прикладом є використання метаданих для:

- виявлення, чи було зображення змінено після його створення, що може свідчити про фальсифікацію;
- аналіз геолокаційних даних для встановлення маршруту руху підозрюваного або дій свідків. [14]

**1.12.3 Маркетинг та аналіз контенту.** У маркетинговій сфері метадані використовуються для аналізу поведінки користувачів та трендів у створенні контенту. Маркетологи можуть використовувати метадані для вивчення таких аспектів, як:

- популярність контенту серед певних вікових або географічних груп;
- частота створення контенту: наприклад, як часто користувачі завантажують зображення або відео з певних регіонів.

Метадані також використовуються для автоматичного категоризування файлів, що дозволяє швидко знаходити потрібний контент у великих медіа-бібліотеках. [15]

## 1.13 Інтеграція з хмарними сервісами

Інтеграція з хмарними сервісами відкриває нові можливості для обробки великих обсягів мультимедійних даних та метаданих. Хмарні рішення дозволяють забезпечити масштабованість, доступність даних та високу продуктивність при роботі з великими архівами мультимедійних файлів.

**1.13.1 Хмарні сховища для мультимедійних даних.** Основні хмарні сервіси, що використовуються для зберігання мультимедійних даних:

- **Amazon S3 (Simple Storage Service):** найпоширеніше хмарне сховище, яке підтримує масове зберігання файлів та інтеграцію через API для доступу до метаданих (рис.1.11);



Рис.1.11 Amazon web services

- **Google Cloud Storage:** ще одне популярне рішення для зберігання даних у хмарі. Воно пропонує високошвидкісний доступ до файлів та підтримку розширених можливостей з обробки даних через Google Cloud Functions (рис1.12);

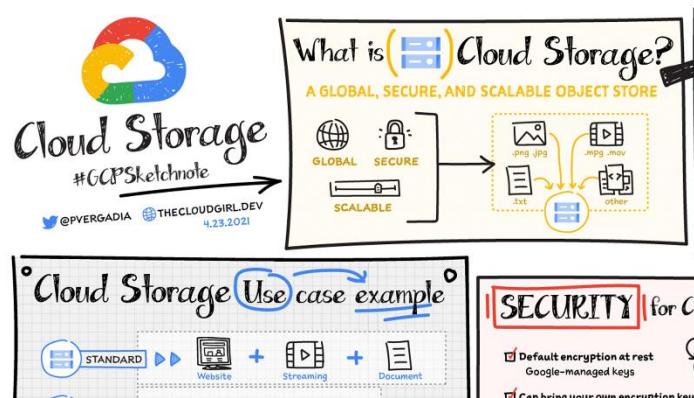


Рис.1.12 Cloud Storage

- **Microsoft Azure Blob Storage:** пропонує масштабоване рішення для зберігання будь-якого типу даних, включаючи мультимедіа, з доступом до аналізу через API.

**1.13.2 Використання API для обробки метаданих у хмарі.** Інтеграція з хмарними сховищами зазвичай здійснюється через API, що дозволяє отримувати доступ до файлів, завантажувати, зчитувати та обробляти метадані. Наприклад:

- **Amazon S3 API:** надає можливість отримати інформацію про файл і його метадані без необхідності завантажувати сам файл. Це особливо зручно для великих медіаархівів.
- **Google Cloud Storage API:** дозволяє здійснювати операції з файлами, включаючи аналіз їх метаданих, безпосередньо у хмарі.

### **1.13.3 Автоматизований аналіз метаданих у хмарних системах.**

Автоматизація процесу аналізу метаданих у хмарних системах можлива через використання хмарних функцій (наприклад, AWS Lambda або Google Cloud Functions). Це дозволяє зменшити затрати на обчислення, оскільки хмарні сервіси можуть виконувати обробку тільки за необхідності.

Прикладом такого процесу може бути автоматичний запуск функції аналізу метаданих кожного разу, коли новий файл завантажується у хмарне сховище. Ця функція може:

1. Зчитати метадані з нового файлу.
2. Зберегти метадані у базу даних для подальшого аналізу.
3. Автоматично повідомити користувача або іншу систему про новий файл і його метадані. [16]

**1.13.4 Масштабованість і продуктивність у хмарі.** Хмарні сервіси дозволяють автоматично масштабувати ресурси у залежності від кількості даних. Це особливо важливо при роботі з великими мультимедійними архівами, коли необхідно обробити тисячі файлів. За допомогою технологій таких як Auto-scaling в Amazon Web Services або Load Balancing в Google Cloud, система може автоматично збільшувати обчислювальні потужності під час пікових навантажень.

## **1.14 Безпека і шифрування метаданих**

З огляду на важливість метаданих для різних галузей, захист їх від несанкціонованого доступу або модифікації стає критично важливим завданням. Метадані можуть містити конфіденційну інформацію, таку як місцезнаходження, дата створення, ідентифікаційні дані автора, тому забезпечення безпеки під час їх обробки є обов'язковою вимогою для сучасних систем.

**1.14.1 Шифрування метаданих.** Шифрування — один із найбільш ефективних способів забезпечення безпеки метаданих. При шифруванні дані перетворюються в недоступний для розуміння формат, який може бути

відновлений лише за допомогою спеціального ключа. Існує кілька типів шифрування, що можуть бути застосовані до метаданих:

- симетричне шифрування: для шифрування і дешифрування використовується один і той же ключ (наприклад, AES);
- асиметричне шифрування: використовуються різні ключі для шифрування (публічний ключ) і дешифрування (приватний ключ) (наприклад, RSA).

**1.14.2 Захист метаданих при зберіганні та передачі.** При передачі файлів з метаданими через мережу важливо захистити дані від перехоплення. Для цього використовуються протоколи захисту даних:

- **TLS (Transport Layer Security):** забезпечує шифрування даних під час передачі через Інтернет;
- **VPN (Virtual Private Network):** дозволяє передавати дані через зашифрований канал зв'язку, що забезпечує конфіденційність інформації.

**1.14.3 Політики доступу до метаданих.** Одним з важливих аспектів забезпечення безпеки метаданих є управління доступом. Системи управління правами доступу дозволяють визначати, хто і які дії може виконувати з метаданими:

- контроль доступу на основі ролей (RBAC): користувачі отримують доступ до метаданих відповідно до їхньої ролі в системі (наприклад, адмін, користувач, гість);
- контроль доступу на основі політик (PBAC): доступ до даних регулюється відповідно до заданих політик, наприклад, на основі локації або часу.

**1.14.4 Приклади шифрування та захисту метаданих.** У багатьох сучасних програмах, таких як Adobe Lightroom або Google Photos, використовуються технології захисту метаданих під час зберігання та обробки файлів. Вони дозволяють користувачам контролювати, які метадані будуть включені у файл, а також використовують шифрування для захисту файлів при зберіганні в хмарі. [17]

## 1.15 Оптимізація та продуктивність системи для обробки метаданих

Розробка системи для обробки великих обсягів метаданих вимагає використання спеціальних технік оптимізації для підвищення її продуктивності. Це особливо важливо, коли система повинна працювати з тисячами або навіть мільйонами мультимедійних файлів.

**1.15.1 Кешування метаданих.** Одним з основних підходів для підвищення швидкості роботи системи є кешування метаданих. Це дозволяє зберігати вже зчитані метадані у швидкодоступному форматі, що зменшує потребу у повторному зчитуванні інформації з файлів. Кешування може бути реалізовано за допомогою:

- In-memory cache: зберігання даних у пам'яті (наприклад, Redis або Memcached) для швидкого доступу;
- дискове кешування: збереження результатів обробки на локальних дисках, щоб уникнути повторних запитів до файлів. [18]

**1.15.2 Паралельна обробка файлів.** Ще одним підходом до оптимізації є використання паралельної обробки файлів. Цей метод дозволяє одночасно обробляти кілька файлів, використовуючи можливості багатоядерних процесорів. У середовищах з великим навантаженням паралелізм може суттєво підвищити продуктивність системи.

Прикладом є використання бібліотеки `multiprocessing` у Python для запуску кількох процесів, які зчитують і аналізують метадані з файлів одночасно.

Паралельна обробка передбачає використання двох або більше процесорів або центральних процесорних одиниць (ЦП) одночасно для обробки різних компонентів однієї задачі. Завдяки поділу завдань на окремі частини, які виконуються на кількох процесорах, системи можуть значно скоротити час виконання програм. Сучасні комп'ютери часто використовують багатоядерні процесори та системи з кількома ЦП, що дозволяє ефективно виконувати паралельну обробку. Багатоядерні процесори – це інтегральні мікросхеми (ІС), які містять два або більше ЦП. Вони забезпечують підвищену швидкість

обробки, знижене енергоспоживання та ефективніше виконання кількох завдань одночасно. У більшості сучасних комп'ютерів можна знайти процесори з двома-чотирма ядрами, тоді як у високопродуктивних системах їх кількість може досягати 12. Паралельна обробка часто використовується для виконання складних операцій і розрахунків.

На базовому рівні різниця між паралельною та послідовною обробкою полягає у способі використання регістрів. Наприклад:

- послідовна обробка: зсувні регістри працюють покроково, обробляючи кожен біт окремо;
- паралельна обробка: регістри з паралельним завантаженням обробляють усі біти слова одночасно.

На складнішому рівні паралельна обробка керується набором функціональних блоків, які можуть одночасно виконувати однакові або різні завдання.

Розвиток паралельних обчислень:

- масово-паралельні процесори (MPP): У 1997 році суперкомп'ютер ASCI Red досягнув трильйона операцій з плаваючою точкою за секунду, що зробило MPP основою високопродуктивних обчислень;
- кластери: З кінця 1980-х років кластери стали заміною MPP для багатьох задач. Кластери – це паралельні комп'ютери, що об'єднують кілька комерційних комп'ютерів у мережу. Вони стали основою сучасної наукової обчислювальної техніки та центрів обробки даних.

Сьогодні паралельна обробка дозволяє звичайним настільним і портативним комп'ютерам вирішувати задачі, які раніше потребували потужних суперкомп'ютерів і експертної підтримки. До середини 1990-х років побутові комп'ютери виконували лише послідовну обробку даних. Сучасні операційні системи можуть ефективно координувати роботу кількох процесорів, роблячи паралельну обробку економічно вигіднішою порівняно з послідовною.

З ростом кількості сенсорів Інтернету речей (IoT) і необхідності в реальному часі обробляти дані, паралельні обчислення стають критично важливими. Завдяки доступності процесорів і графічних процесорів (GPU) через

хмарні сервіси паралельна обробка є основним елементом у розгортанні мікросервісів та інших високопродуктивних обчислювальних технологій.

Паралельна обробка передбачає розподіл завдання між двома чи більше мікропроцесорами. Суть методу проста: інженер або програміст за допомогою спеціалізованого програмного забезпечення розбиває складне завдання на окремі компоненти. Потім кожен процесор отримує свою частину роботи. Після виконання всіх частин, програмне забезпечення збирає результати для розв'язання початкової задачі. При використанні паралельної обробки велике завдання ділиться на кілька менших, які краще відповідають доступним процесорам за їх кількістю, розміром і типом. Після розподілу завдання кожен процесор починає виконувати свою частину. Процесори не взаємодіють між собою напряму, а використовують програмне забезпечення для обміну даними та синхронізації. Коли всі частини завдання завершені, програма збирає сегменти в єдиний результат. Це справедливо як для ситуацій, коли кількість завдань і процесорів однакова, і всі закінчують одночасно, так і коли завдання завершуються послідовно. [19]

Види паралельної обробки (рис 1.13).

1. Тонкозерниста паралельність (fine-grained parallelism) - завдання активно взаємодіють один з одним кілька разів на секунду, щоб отримувати результати в реальному часі або майже в реальному часі.

2. Грубозерниста паралельність (coarse-grained parallelism) - взаємодія між завданнями відбувається рідше, що уповільнює обробку.

Переваги паралельної обробки

Система паралельної обробки може виконувати дані одночасно, що дозволяє швидше завершувати завдання. Наприклад:

- процесор може отримувати наступну інструкцію з пам'яті, поки його арифметико-логічний блок (ALU) виконує поточну операцію.
- основна мета паралельної обробки – підвищити продуктивність комп'ютера та збільшити пропускну здатність (обсяг виконаної роботи за певний час).

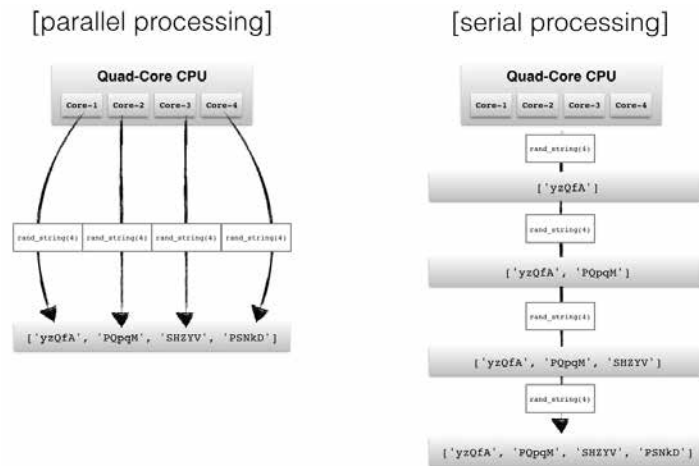


Рис.1.13 Різниця між обробками файлів

**1.15.3 Стиснення даних.** Оскільки метадані можуть займати значний обсяг, особливо коли система працює з великими архівами файлів, важливо застосовувати техніки стиснення даних. Для зменшення обсягу зберігання метаданих можуть бути використані такі алгоритми стиснення, як:

- **Gzip:** популярний алгоритм стиснення, що дозволяє суттєво зменшити розмір файлів метаданих;
- **Brotli:** новіший та ефективніший метод стиснення, що забезпечує краще співвідношення між швидкістю і обсягом даних. [20]

**1.15.4 Масштабованість системи.** Масштабованість системи забезпечує можливість її ефективної роботи навіть при зростанні кількості даних. Масштабовані системи можуть автоматично збільшувати обчислювальні потужності або зберігати дані у розподілених сховищах. Для досягнення масштабованості можна використовувати:

- розподілені бази даних (наприклад, MongoDB, Cassandra), які дозволяють зберігати метадані у розподілених кластерах, забезпечуючи високий рівень доступності;
- розподілені системи черг (наприклад, Apache Kafka), які дозволяють розподіляти обробку запитів по кількох серверах для підвищення продуктивності.

## 1.16 Юридичні аспекти обробки метаданих

Юридичні питання, пов'язані з обробкою метаданих, набувають все більшої важливості, оскільки ці дані можуть містити особисту інформацію або інформацію, що підлягає захисту згідно з міжнародними законами про конфіденційність та авторські права.

**1.16.1 Законодавчі акти щодо обробки персональних даних.** Метадані мультимедійних файлів можуть містити особисту інформацію, наприклад, геолокацію, ідентифікаційні дані автора, дати та місця створення файлів. Відповідно до законодавства, яке регулює обробку персональних даних, таких як General Data Protection Regulation (GDPR) у ЄС та California Consumer Privacy Act (CCPA) у США, компанії зобов'язані дотримуватися таких вимог:

- згода на обробку: компанії повинні отримувати згоду від користувачів на обробку їхніх персональних даних, включаючи метадані;
- право на забуття: користувачі мають право вимагати видалення своїх даних, включаючи метадані, із систем компаній. [21]

**1.16.2 Захист авторських прав.** Метадані, особливо IPTC-дані, часто містять інформацію про автора мультимедійних файлів, а також про права на їх використання. Згідно з міжнародними конвенціями, такими як Бернська конвенція, видалення або зміна метаданих, що містять інформацію про авторство, без дозволу автора, вважається порушенням авторських прав.

Багато фотоагентств та платформ для обміну контентом, таких як Getty Images та Shutterstock, зобов'язують зберігати метадані, що містять дані про авторство, при публікації та розповсюдженні медіаконтенту. [22]

**1.16.3 Конфіденційність і безпека.** Метадані можуть містити чутливу інформацію, таку як місцезнаходження або час створення файлу. Відповідно до законів про конфіденційність, такі дані повинні бути захищені від несанкціонованого доступу. Наприклад, згідно з GDPR, компанії повинні забезпечувати належний рівень захисту даних і не допускати їх витоку.

У випадках витоку персональних даних, компанії можуть бути оштрафовані на значні суми, як це було у випадку з Facebook, де витік персональних даних торкнувся мільйонів користувачів. [23]

**1.16.4 Відповідальність за фальсифікацію метаданих.** Фальсифікація метаданих може призвести до серйозних юридичних наслідків, особливо у сфері цифрової криміналістики. Наприклад, якщо метадані були змінені для того, щоб приховати або змінити інформацію про авторство чи місце зйомки, це може вважатися фальсифікацією доказів у суді.

Міжнародні угоди, такі як Договори ВОІВ (Всесвітня організація інтелектуальної власності), зобов'язують захищати інформацію, що міститься в метаданих, і не допускати її фальсифікації. [24]

#### **1.16.5 Відсоток конфіденційних метаданих на різних платформах.**

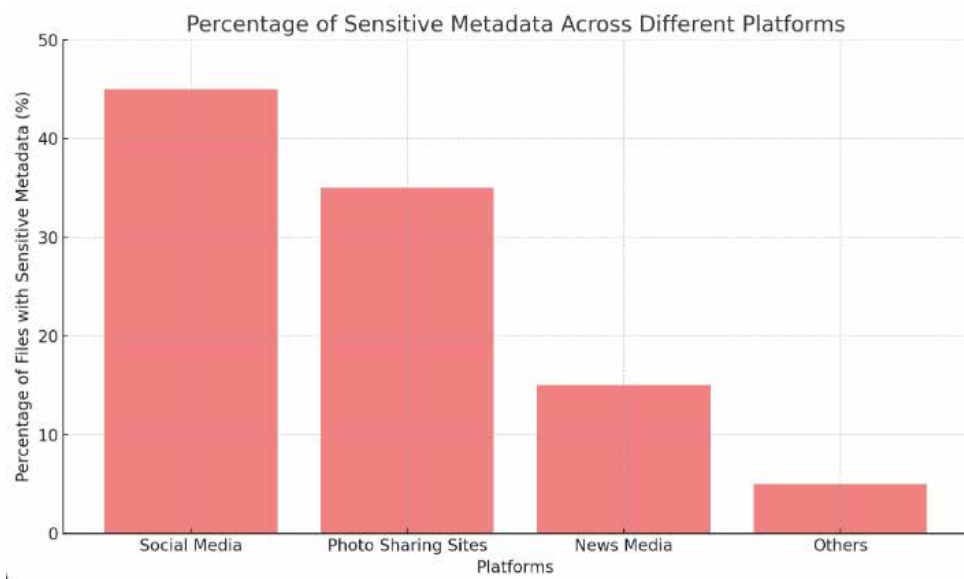


Рис.1.14 Частка конфіденційних метаданих

Цей графік (рис1.14) показує частку конфіденційних метаданих на різних платформах. Найбільше конфіденційних даних містять файли на соціальних медіа (45%) та фото-сайтах (35%), що підтверджує важливість контролю метаданих на цих платформах.

## 1.17 Приклади застосування метаданих у бізнесі

Метадані є важливим елементом у багатьох галузях бізнесу, оскільки вони дозволяють ефективно організовувати, аналізувати та керувати мультимедійним контентом. Різні компанії та організації використовують метадані для автоматизації бізнес-процесів, монетизації контенту та аналізу поведінки користувачів.

**1.17.1 Медіа та розваги.** У галузі медіа метадані дозволяють компаніям, таким як Netflix, Spotify та YouTube, рекомендувати контент своїм користувачам на основі їхніх уподобань. Відео та музичні стримінгові сервіси використовують метадані для наступних цілей:

- каталогізація контенту: автоматичне сортування відео та музики за жанром, виконавцем, датою випуску тощо;
- рекомендаційні системи: метадані про перегляди, оцінки та взаємодію користувачів допомагають побудувати моделі рекомендацій на основі поведінки користувачів;
- монетизація: за допомогою метаданих компанії можуть відслідковувати популярність певного контенту та налаштовувати рекламні кампанії для відповідних цільових груп. [25]

**1.17.2 Електронна комерція.** У сфері електронної комерції метадані застосовуються для підвищення ефективності пошукових систем на сайтах, таких як Amazon чи eBay. Продавці можуть додавати метадані до своїх товарів (назви, категорії, ціни, описи, технічні характеристики), що дозволяє користувачам швидко знаходити потрібні продукти через пошукові запити.

Метадані у електронній комерції також використовуються для:

- аналізу продажів: аналіз метаданих про покупки та вподобання користувачів допомагає компаніям прогнозувати попит на товари та коригувати ціни;
- персоналізація: за допомогою метаданих можна створювати індивідуальні пропозиції для користувачів на основі їхньої попередньої поведінки (переглядів, покупок, доданих у кошик товарів). [26]

**1.17.3 Медіаагенції та рекламні платформи.** Медіаагенції використовують метадані для ефективного управління великими обсягами візуального контенту. Наприклад, Getty Images та Shutterstock використовують IPTC-метадані для зберігання інформації про авторів зображень, дати їх створення та права на використання. Це дозволяє автоматично сортувати зображення за категоріями та створювати бази даних, доступні для пошуку та аналізу.

**1.17.4 Маркетинг і аналітика.** Маркетингові компанії застосовують метадані для відстеження популярності контенту в соціальних мережах та аналізу поведінки користувачів. Дані про час і місце створення контенту, кількість переглядів і взаємодій дозволяють маркетологам краще розуміти, які типи контенту залучають аудиторію.

Прикладом цього є кампанії, що використовують метадані для:

- ретаргетингу: націлювання на користувачів, які вже взаємодіяли з певним контентом або продуктами;
- аналізу трендів: виявлення популярних тем та контенту, що дозволяє швидко реагувати на нові тенденції та ефективно керувати контентом.

## **1.18 Майбутнє технологій метаданих**

Технології, пов'язані з метаданими, постійно розвиваються, відкриваючи нові можливості для аналізу даних, забезпечення конфіденційності та безпеки, а також застосування у нових сферах, таких як штучний інтелект (ШІ) та віртуальна реальність (VR).

**1.18.1 Метадані в штучному інтелекті та машинному навчанні.** Метадані відіграють важливу роль у тренуванні моделей штучного інтелекту. Наприклад, у сфері комп'ютерного зору метадані зображень використовуються для класифікації та виявлення об'єктів. ШІ-системи можуть навчатися на великих масивах зображень та відео з використанням метаданих для полегшення процесу навчання.

Машинне навчання використовує метадані для:

- підвищення точності моделей: метадані допомагають алгоритмам ШІ краще розуміти контекст зображення або відео;
- автоматизації класифікації: системи можуть автоматично сортувати та класифікувати медіафайли на основі їхніх метаданих.

**1.18.2 Метадані у віртуальній реальності (VR) та 3D-контенті.** З розвитком віртуальної реальності (VR) та 3D-контенту, метадані починають грати важливу роль у впорядкуванні таких файлів та забезпеченні їхньої інтеграції у VR-системи. Метадані 3D-об'єктів можуть включати інформацію про їхню текстуру, координати, орієнтацію у просторі та сценарії використання.

Це дозволяє створювати:

- інтерактивні середовища: де метадані визначають, як об'єкти взаємодіють між собою та з користувачем;
- автоматизовані VR-системи: що дозволяють відтворювати сценарії на основі метаданих контенту.

**1.18.3 Тенденції щодо конфіденційності та захисту даних.** З розвитком технологій та збільшенням обсягу персональних даних, включаючи метадані, які обробляються компаніями, питання конфіденційності та безпеки стають ще більш важливими. Очікується, що нові закони та регулювання, такі як GDPR та CCPA, продовжать розвиватися, щоб забезпечити користувачам більше контролю над своїми даними.

У майбутньому можна очікувати:

- більше стандартів для захисту метаданих: зокрема щодо зберігання та передачі персональних даних у метаданих;
- шифрування за замовчуванням: для забезпечення того, щоб метадані були доступні лише авторизованим користувачам.

**1.18.4 Розширення стандартів метаданих.** Очікується, що стандарти метаданих будуть продовжувати розширюватися, включаючи нові типи мультимедійного контенту. Наприклад, у сфері віртуальної реальності або 3D-моделювання вже існує потреба у нових стандартах для опису інтерактивних сценаріїв, динамічних об'єктів і користувацьких інтерфейсів.

Майбутні стандарти метаданих можуть включати:

- динамічні метадані: що змінюються в реальному часі у відповідь на взаємодію з користувачем;
- інтерактивні метадані: що дозволяють користувачам змінювати або додавати нову інформацію до медіафайлів під час їх використання.

## **1.19 Перспективи розвитку технологій обробки метаданих**

Технології обробки метаданих продовжують швидко розвиватися, відкриваючи нові можливості для бізнесу, дослідників та звичайних користувачів. Метадані вже стали важливою частиною багатьох систем для управління контентом, аналізу великих даних, машинного навчання, штучного інтелекту та цифрової криміналістики. Подальші перспективи їх розвитку можна розглядати у кількох ключових напрямках.

**1.19.1 Автоматизація процесів обробки метаданих.** Автоматизація обробки метаданих стала можливою завдяки впровадженню нових технологій, зокрема штучного інтелекту та машинного навчання. У майбутньому можна очікувати ще більших досягнень у цій галузі, зокрема:

- автоматизовані інструменти для обробки контенту: системи, які автоматично аналізуватимуть та класифікуватимуть файли на основі метаданих без втручання людини;
- інтеграція з системами машинного навчання: обробка та аналіз метаданих можуть бути повністю автоматизовані з використанням навчальних алгоритмів, що зможе швидше і точніше виконувати обробку великих обсягів мультимедійного контенту.

**1.19.2 Розвиток стандартів метаданих для нових типів контенту.** Традиційні стандарти метаданих, такі як EXIF, ID3, IPTC та XMP, добре підходять для зображень, аудіо та відео. Однак із зростанням популярності нових типів мультимедійного контенту, таких як 3D-моделі, віртуальна та доповнена реальність (VR/AR), з'являється потреба в нових стандартах:

- метадані для VR/AR: описують сценарії взаємодії віртуальних об'єктів із користувачами;
- метадані для 3D-моделей: містять інформацію про розміри, текстури, матеріали та розташування об'єктів у просторі.

**1.19.3 Підвищення безпеки метаданих.** У сучасному світі, де все більше обсягу інформації зберігається та передається через Інтернет, питання безпеки метаданих стають все важливішими. У майбутньому очікується розвиток нових стандартів та інструментів для захисту метаданих:

- шифрування метаданих за замовчуванням: забезпечення конфіденційності інформації навіть при доступі до файлів через загальнодоступні мережі;
- технології блокчейну: можуть бути використані для відстеження змін метаданих і гарантування їхньої автентичності.

**1.19.4 Інтеграція метаданих із штучним інтелектом.** Одним із ключових напрямків розвитку є інтеграція метаданих із технологіями штучного інтелекту (ШІ) та машинного навчання (ML). Метадані можуть надавати ШІ-системам додаткову інформацію для аналізу, що дозволяє підвищити точність алгоритмів та швидкість обробки даних.

Сфери застосування метаданих у ШІ включають:

- комп'ютерний зір: використання метаданих для ідентифікації об'єктів на зображеннях та відео;
- натуральна мова: аналіз текстових метаданих для поліпшення алгоритмів обробки природної мови (NLP).

## **1.20 Використання метаданих у цифровій криміналістиці**

Метадані відіграють важливу роль у цифровій криміналістиці, де їх використовують для відстеження та аналізу мультимедійних файлів у розслідуваннях. Цифрові криміналісти використовують метадані для збору доказів, які можуть допомогти у встановленні походження файлів, часу їх створення, а також можливих змін.

**1.20.1 Важливість метаданих у розслідуваннях.** У розслідуваннях часто виникає необхідність відстежити, де і коли був зроблений файл, а також чи зазнавав він змін. Метадані можуть містити інформацію про:

- дата та час створення: допомагають визначити, коли саме було зроблено знімок або запис відео;
- геолокація: визначає місце, де був зроблений знімок або запис;
- авторство: інформація про автора файлу може бути корисною для розслідувань, пов'язаних з інтелектуальною власністю.

**1.20.2 Використання метаданих у судових процесах.** Метадані можуть бути використані як докази в судових процесах. Однак для цього необхідно забезпечити їх автентичність і довести, що вони не зазнавали змін. Фальсифікація метаданих може призвести до серйозних правових наслідків, зокрема кримінальної відповідальності.

**1.20.3 Інструменти для аналізу метаданих у криміналістиці.** Існує багато інструментів, які використовуються для аналізу метаданих у криміналістичних розслідуваннях. Серед них:

- **ExifTool:** використовується для аналізу метаданих зображень та відео;
- **FTK Imager:** криміналістичний інструмент для створення образів дисків та аналізу файлів, включаючи їхні метадані;
- **X-Ways Forensics:** інструмент, що дозволяє аналізувати файли та метадані у великих обсягах даних.

**1.20.4 Виклики у використанні метаданих у криміналістиці.** Попри важливість метаданих, у цифровій криміналістиці існують певні виклики:

- фальсифікація метаданих: можлива зміна метаданих для приховування або підробки інформації;
- втрата даних: деякі інструменти та методи обробки файлів можуть видаляти або змінювати метадані, що ускладнює їх використання в розслідуваннях.

## **1.21 Роль метаданих у забезпеченні якості даних**

Метадані відіграють важливу роль у забезпеченні якості даних, оскільки вони надають додаткову інформацію про самі дані, їх походження та стан. Висока якість метаданих дозволяє забезпечити точність, актуальність та відповідність даних, що використовуються у системах управління та аналізу великих даних.

**1.21.1 Забезпечення точності метаданих.** Точність метаданих є важливим аспектом для багатьох галузей, включаючи наукові дослідження, журналістику та правові системи. Якщо метадані зображень або відео містять помилкову або неповну інформацію, це може призвести до неправильного аналізу або хибних висновків.

Методи забезпечення точності включають:

- автоматичне зчитування: системи повинні використовувати надійні інструменти для автоматичного зчитування метаданих без втручання користувача, щоб уникнути людських помилок;
- перевірка та валідація: кожен набір метаданих повинен бути перевірений на відповідність формату та стандартам, щоб уникнути помилок під час їх зчитування або обробки.

**1.21.2 Актуальність та оновлення метаданих.** Метадані повинні бути актуальними, щоб вони відповідали реальному стану файлів, з якими вони асоційовані. Наприклад, якщо зображення або відео зазнали змін після їх створення, відповідні метадані мають бути автоматично оновлені.

- автоматичне оновлення: системи можуть бути налаштовані на автоматичне оновлення метаданих у разі змін у мультимедійних файлах (наприклад, зміна формату файлу, розміру, дати модифікації);
- логування змін: для забезпечення прозорості система може вести лог змін у метаданих, що допомагає відстежувати, хто та коли вносив зміни.

**1.21.3 Захист цілісності метаданих.** Цілісність метаданих означає, що вони залишаються незмінними та автентичними протягом усього життєвого

циклу файлу. Захист цілісності метаданих особливо важливий у таких галузях, як цифрова криміналістика, медіа-архіви та наукові дослідження.

Методи захисту цілісності включають:

- цифрові підписи: забезпечують перевірку того, що метадані не були змінені після їх створення;
- технології блокчейну: можуть бути використані для захисту метаданих та відстеження їхніх змін у децентралізованих системах.

## **1.22 Виклики та проблеми використання метаданих**

Попри всі переваги, пов'язані з використанням метаданих, існують також певні виклики та проблеми, які можуть впливати на їх ефективність. Ці виклики включають технічні обмеження, правові та етичні питання, а також питання конфіденційності та безпеки.

**1.22.1 Технічні обмеження.** Одним з основних викликів використання метаданих є технічні обмеження, пов'язані з обробкою великих обсягів мультимедійних файлів та їхніх метаданих. Ці проблеми можуть включати:

- обмеження швидкості обробки: великі набори файлів можуть уповільнити роботу системи, якщо вона не була належним чином оптимізована;
- розбіжності у стандартах: різні мультимедійні формати підтримують різні стандарти метаданих, що може ускладнити їх інтеграцію в одну систему.

**1.22.2 Правові та етичні питання.** Метадані часто містять чутливу інформацію, таку як геолокація або інформація про автора файлу. Це породжує питання конфіденційності та юридичних наслідків їх використання:

- конфіденційність: у багатьох випадках метадані можуть розкривати приватну інформацію користувача, що порушує законодавство про захист даних (наприклад, GDPR або CCPA);
- етичне використання: неправильне використання метаданих може призвести до фальсифікації або маніпуляції з інформацією, що особливо важливо в контексті журналістики та наукових досліджень.

**1.22.3 Захист метаданих від фальсифікації.** Фальсифікація метаданих є серйозною проблемою, оскільки їх можна підробити з метою приховування або зміни важливих аспектів файлу, таких як дата створення або геолокація. Існують кілька методів захисту метаданих від фальсифікації:

- цифрові підписи: дозволяють верифікувати автентичність метаданих і файлів;
- інструменти контролю версій: дозволяють відстежувати всі зміни, внесені до метаданих, зберігаючи інформацію про те, хто і коли вніс зміни.

## **1.23 Майбутнє використання метаданих у великих даних (Big Data)**

Метадані грають центральну роль у великих даних (Big Data), оскільки вони забезпечують додаткову інформацію про дані та полегшують їхній аналіз. Із розвитком технологій, таких як штучний інтелект, машинне навчання та Інтернет речей (IoT), метадані стають важливим інструментом для збору, організації та аналізу величезних обсягів даних.

**1.23.1 Метадані у великих даних.** У контексті великих даних метадані дозволяють організувати і структурувати інформацію для швидкого доступу до неї. Вони можуть включати дані про джерела, обсяг, структуру та час створення кожного набору даних.

Метадані у Big Data можуть бути використані для:

- фільтрації та індексації: метадані дозволяють швидко знаходити та індексувати дані в масивних базах даних;
- верифікації та обробки: метадані надають інформацію про стан даних, допомагаючи верифікувати їхню точність та актуальність. [27]

**1.23.2 Інтеграція метаданих у системи штучного інтелекту.** Штучний інтелект та машинне навчання активно використовують метадані для навчання моделей і підвищення точності прогнозів. Метадані забезпечують додатковий контекст для даних, що дозволяє алгоритмам краще розуміти їхню структуру та значення.

Наприклад:

- навчання моделей: метадані можуть допомогти класифікувати великі обсяги мультимедійних файлів і використовувати їх для тренування моделей машинного навчання;
- контекстуалізація даних: метадані дозволяють штучному інтелекту краще зрозуміти контекст даних, наприклад, час і місце їх створення, що покращує точність прогнозів.

**1.23.3 Майбутні виклики і можливості.** Хоча метадані надають численні можливості для аналізу великих даних, вони також стикаються з викликами:

- обробка великих обсягів метаданих: для масштабних систем Big Data потрібні потужні обчислювальні ресурси для зберігання та обробки метаданих;
- конфіденційність і захист даних: оскільки метадані часто містять приватну інформацію, їхній захист стає важливим аспектом у системах Big Data.

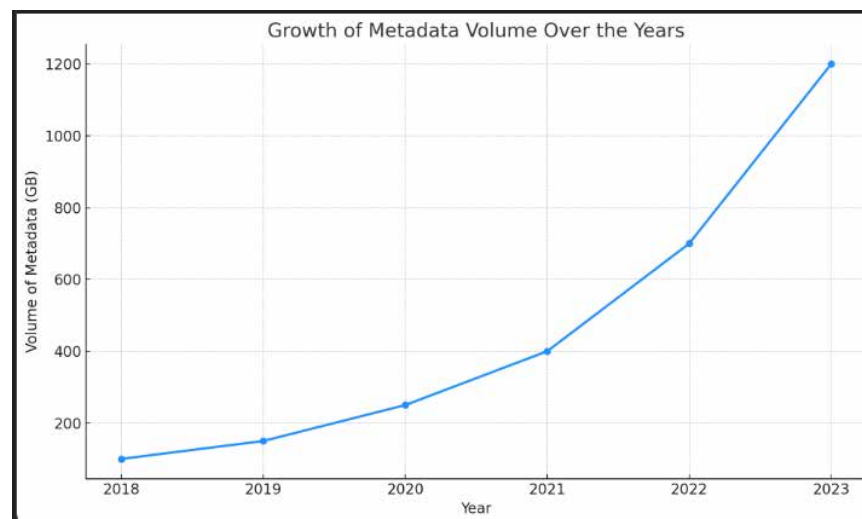


Рис.1.15 Обсяги метаданих по рокам

Цей графік (рис.1.15) ілюструє зростання обсягів метаданих за останні роки. Спостерігається значний ріст, особливо з 2021 року, що збігається зі збільшенням мультимедійного контенту та даних з Інтернету речей.

## 1.24 Інтеграція метаданих з Інтернет речей (IoT)

Інтернет речей (IoT) швидко розвивається, і кількість підключених пристроїв зростає щороку. Пристрої IoT генерують величезні обсяги даних, зокрема й метадані, що містять інформацію про стан, місце знаходження та функціонування цих пристроїв. Інтеграція метаданих з IoT надає нові можливості для аналізу та управління даними.

**1.24.1 Роль метаданих у системах IoT.** У системах IoT метадані дозволяють організувати та класифікувати великі обсяги даних, які збираються з різних датчиків і пристроїв. Метадані містять інформацію про:

- типи даних: опис типу інформації, що збирається (температура, вологість, швидкість, місцезнаходження тощо);
- тимчасові мітки: час збору даних, що дозволяє відстежувати зміну стану пристроїв або умов навколишнього середовища;
- джерела даних: ідентифікація пристрою або датчика, з якого надійшли дані.

**1.24.2 Метадані для підвищення ефективності систем IoT.** Метадані можуть використовуватися для підвищення ефективності систем IoT через:

- оптимізацію роботи пристроїв: на основі метаданих можна автоматично регулювати роботу пристроїв IoT, наприклад, змінювати режими роботи в залежності від часу доби або умов навколишнього середовища;
- моніторинг і діагностику: метадані можуть допомагати у відстеженні стану пристроїв IoT та їхньої працездатності. Це дозволяє виявляти несправності або прогнозувати можливі проблеми на основі аналізу метаданих.

**1.24.3 Приклади застосування метаданих у IoT.** Метадані в IoT вже використовуються у багатьох галузях, включаючи:

- розумні міста: метадані дозволяють контролювати та керувати міськими інфраструктурами, такими як транспорт, енергозбереження, контроль якості повітря та води;

- розумні будинки: системи управління домашніми пристроями використовують метадані для автоматизації дій, таких як регулювання температури, освітлення або сигналізації;
- Індустріальні IoT-системи: на виробничих підприємствах метадані використовуються для моніторингу обладнання, прогнозування технічного обслуговування та оптимізації виробничих процесів. [28]

## 1.25 Майбутні виклики у використанні метаданих

Попри величезні переваги, метадані стикаються з низкою викликів, які можуть ускладнювати їх використання в майбутньому. Ці виклики пов'язані з конфіденційністю, безпекою, стандартизацією та обсягами даних, які потрібно обробляти.

**1.25.1 Проблеми конфіденційності та захисту даних.** Зі зростанням обсягів даних, що збираються з мультимедійних файлів, IoT-пристроїв та інших джерел, питання конфіденційності стають ще більш актуальними. Метадані можуть містити чутливу інформацію, яка, у разі витоку, може використовуватися для ідентифікації користувачів, їх місця знаходження або дій.

Майбутні виклики включають:

- захист конфіденційних метаданих: необхідність розробки нових методів шифрування та збереження конфіденційності метаданих;
- відповідність законодавству: метадані підлягають регулюванню згідно з законами про конфіденційність, такими як GDPR і CCPA, що ускладнює їх обробку та зберігання.

**1.25.2 Стандартизація метаданих.** Однією з проблем, з якою стикаються багато компаній, є відсутність універсальних стандартів для метаданих. Кожен тип мультимедійних файлів або систем має свої унікальні формати метаданих, що ускладнює їх інтеграцію в одну систему.

- розбіжності у форматах: різні стандарти для зображень (EXIF), відео (XMP) та аудіо (ID3) ускладнюють об'єднання даних в єдину систему;

- необхідність універсальних стандартів: існує потреба у створенні єдиних універсальних стандартів для обробки метаданих, що можуть бути використані для всіх типів мультимедійного контенту.

**1.25.3 Обробка великих обсягів метаданих.** Зі зростанням кількості пристроїв та контенту зростають і обсяги метаданих, які потрібно зберігати та обробляти. Це створює виклики для систем Big Data та хмарних платформ, які повинні обробляти та зберігати величезні обсяги даних з високою продуктивністю.

Можливі рішення включають:

- масштабування хмарних систем: використання хмарних обчислювальних платформ, таких як Amazon Web Services (AWS) або Google Cloud, для автоматичного масштабування ресурсів під час обробки великих обсягів метаданих;

- оптимізація алгоритмів обробки: нові методи обробки та оптимізації дозволять швидше і ефективніше обробляти величезні обсяги даних.

**1.25.4 Автоматизація обробки метаданих.** Зі збільшенням обсягів даних та складністю сучасних систем автоматизація процесів обробки метаданих стає необхідною. Майбутні виклики включають:

- створення автономних систем: системи, що можуть автоматично зчитувати, аналізувати та зберігати метадані без втручання людини;

- машинне навчання для аналізу метаданих: використання штучного інтелекту для автоматичної класифікації та обробки метаданих.

## 2 МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1 Моделювання предметної області

Моделювання предметної області є ключовим етапом у процесі створення програмної системи. Він охоплює ідентифікацію об'єктів, встановлення їхніх взаємозв'язків, визначення властивостей і моделювання подій та процесів. Спершу визначаються основні сутності предметної області і зв'язки між ними, що описують їхню взаємодію. Далі деталізуються характеристики кожного об'єкта, такі як параметри чи стани, та моделюються події і процеси, які відбуваються в предметній області, наприклад, дії, зміни станів або послідовності подій. Для моделювання використовуються різноманітні методи і техніки, як-от діаграми класів, зв'язків та станів, що допомагають візуалізувати структуру та поведінку об'єктів. Результати моделювання стають базою для наступного етапу — проектування та розробки, допомагаючи чітко визначити основні вимоги до системи, а також слугуючи основою для аналізу, проектування та реалізації. Це дозволяє отримати глибоке розуміння предметної області, що сприяє успішному впровадженню системи. Додатковою перевагою моделювання є можливість розкрити складність предметної області, зрозуміти взаємозв'язки між об'єктами та процесами, а також виявити можливі проблеми ще до початку реалізації. Моделювання також об'єднує команду навколо спільного розуміння завдань, сприяє кращій комунікації, знижує ризики непорозумінь і забезпечує ефективну співпрацю. Коректно виконане моделювання дозволяє також проаналізувати вимоги до системи та визначити основні пріоритети для подальшої розробки. Таким чином, результати моделювання стають основою для формулювання функціональних та нефункціональних вимог до програмної системи.

Отже, моделювання предметної області є важливим кроком, що допомагає уявити майбутню систему, покращує комунікацію у команді розробників та забезпечує фундамент для подальшого проектування і розробки програмного продукту.

**2.1.1 Діаграма прецедентів.** Діаграма прецедентів (рис2.1) є візуальним інструментом для моделювання взаємодії користувачів (акторів) із системою. Вона відображає функціональні можливості системи через прецеденти, що є конкретними сценаріями використання системи користувачами. Основна мета діаграми прецедентів — визначити та описати вимоги до системи з точки зору користувачів. Вона дозволяє зрозуміти, які функції мають бути реалізовані, хто взаємодіє з системою та які саме сценарії підтримує система. Це допомагає встановити зв'язки між різними компонентами системи та визначити їх функціональні обов'язки. Діаграма прецедентів включає акторів, прецеденти та зв'язки між ними. Актори представляють користувачів або зовнішні системи, що взаємодіють із системою, прецеденти описують сценарії взаємодії, а зв'язки показують, які сценарії доступні кожному акторові та як актори впливають на кожен сценарій. Для створення діаграми прецедентів зазвичай застосовуються спеціалізовані програми або моделюючі інструменти, такі як UML-інструменти. Ця діаграма часто розробляється на етапі аналізу та проектування для визначення ключових функцій і вимог до системи. Таким чином, діаграма прецедентів є корисним інструментом для моделювання взаємодії користувачів із системою. Вона допомагає ідентифікувати та описати функціональні можливості системи, визначити вимоги користувачів і встановити зв'язки між компонентами системи, що в підсумку сприяє покращенню проектування та якості розробки програмного забезпечення.

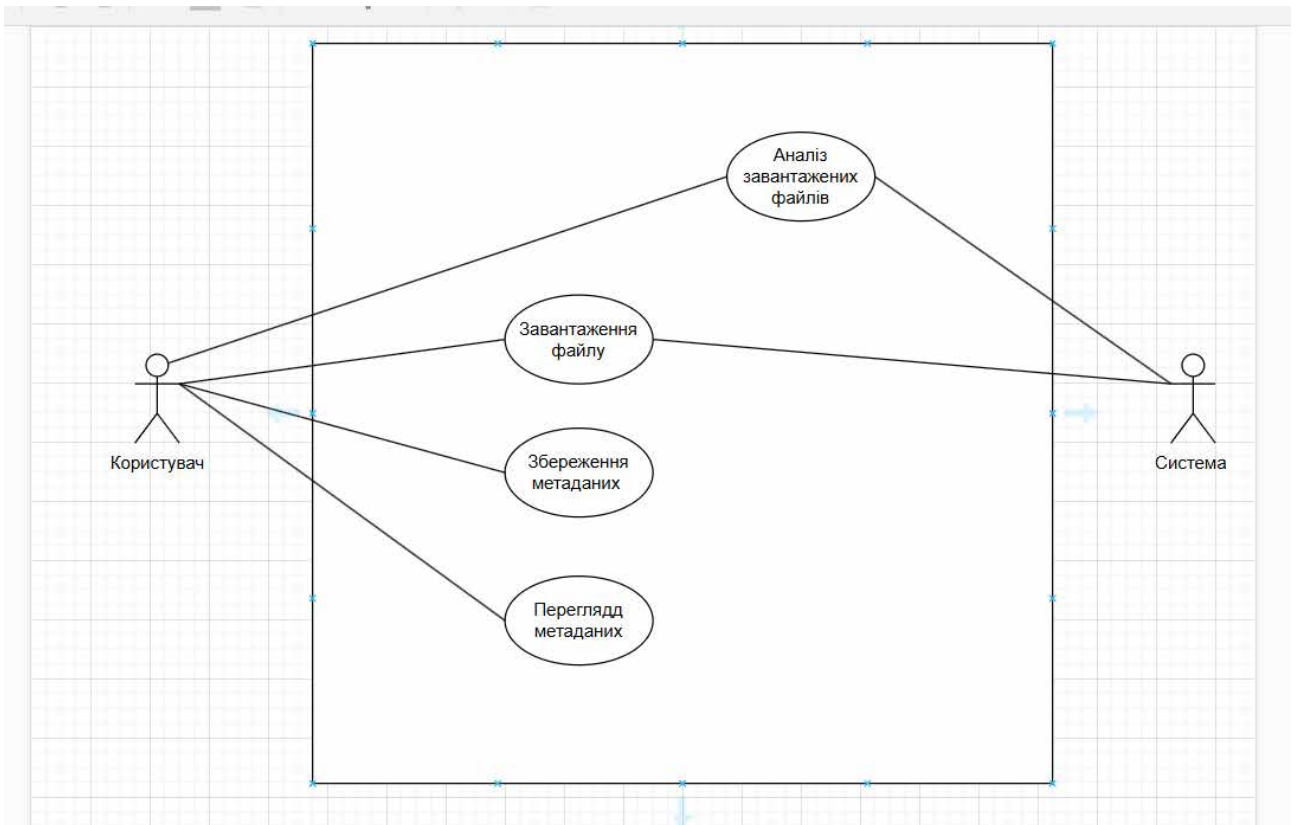


Рис.2.1 Діаграма прецедентів

**2.1.2 Діаграма послідовності.** Діаграми послідовностей застосовуються для моделювання логіки виконання сценаріїв, відображаючи обмін інформацією між об'єктами системи під час реалізації сценарію. Діаграми послідовностей (Sequence Diagrams) надають візуальне представлення процесу або взаємодії об'єктів під час виконання сценарію. На них зображено класи, необхідні для реалізації сценарію, та повідомлення, якими вони обмінюються (виклики, що відбуваються в межах варіанту використання). Діаграми послідовностей демонструють саме взаємодію об'єктів у сценарії, а не їхні структурні зв'язки. Такі діаграми часто застосовуються для відображення взаємодії між компонентами користувацького інтерфейсу чи програмного забезпечення.

Ця діаграма (рис.2.2) демонструє взаємодію між компонентами системи під час завантаження файлів, збереження метаданих і надання доступу до них для подальшого аналізу. Така структура забезпечує зрозумілий процес обробки та зберігання даних, а також підтримує чіткий зворотний зв'язок із користувачем.

Нижче наведено опис основних етапів, які відображені на діаграмі.

- User:
  - користувач взаємодіє із системою, надсилаючи запити на завантаження файлів та виконання аналізу;
  - запит на завантаження файлу передбачає передачу файлу у систему для збереження та обробки.
- Система:
  - система приймає запит від користувача на завантаження файлу;
  - передає запит у файлову систему для збереження файлу;
  - після збереження файлу система ініціює процес обробки метаданих, відправляючи запит на аналіз метаданих файлу.
- Файлова система:
  - здійснює збереження файлу, отриманого від системи;
  - після успішного збереження файлу відправляє сигнал системі про завершення процесу.
- Аналіз метаданих:
  - отримує файл для аналізу після його збереження;
  - здійснює обробку метаданих файлу, витягаючи ключову інформацію;
    - повертає оброблені метадані назад у систему для подальшого збереження.
- База даних:
  - після обробки метаданих база даних отримує запит на збереження цих метаданих;
    - зберігає метадані для подальшого доступу користувачем через систему.
- Зворотний зв'язок із користувачем:
  - після завершення всіх процесів система надає зворотний зв'язок користувачу, інформуючи його про успішне завантаження та обробку файлу.

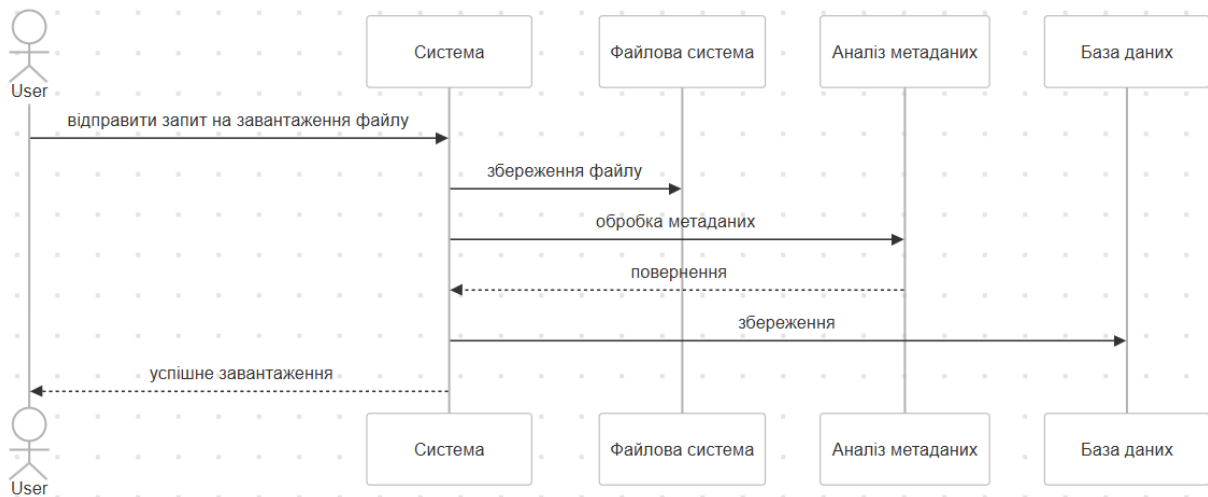


Рис.2.2 Діаграма послідовності

**2.1.3 Архітектура системи.** Архітектура системи — це структурний дизайн, який визначає основні компоненти програмного забезпечення, їхні взаємозв'язки та функції. Вона є планом, за яким будується система, і описує, як різні модулі, сервіси та бази даних взаємодіють між собою. Основна мета архітектури системи — забезпечити чітке розуміння того, як функціонує система, як вона обробляє дані та яким чином досягає своїх цілей. Це дозволяє оптимізувати її продуктивність, спрощує процес розробки та забезпечує зручність подальшого обслуговування та масштабування системи.

Опис адаптованої архітектури (рис.2.3)

1. Робоча станція користувача:
  - модуль завантаження файлів: Користувач може завантажувати мультимедійні файли через веб-інтерфейс;
  - модуль перегляду метаданих: Дає можливість переглядати метадані завантажених файлів.
2. Сервер:
  - модуль обробки запитів: Серверний модуль (на Flask), який приймає запити на завантаження файлів, збереження метаданих, пошук і аналіз;

- о база даних (БД): MongoDB, в якій зберігаються всі метадані мультимедійних файлів, включаючи інформацію про сесію користувача для ізоляції даних.

### 3. Робоча станція аналітика:

- о модуль аналізу даних: забезпечує візуалізацію даних і графічний аналіз. Тут аналітик може переглядати статистику про файли, такі як розподіл типів файлів, створювати діаграми тощо.

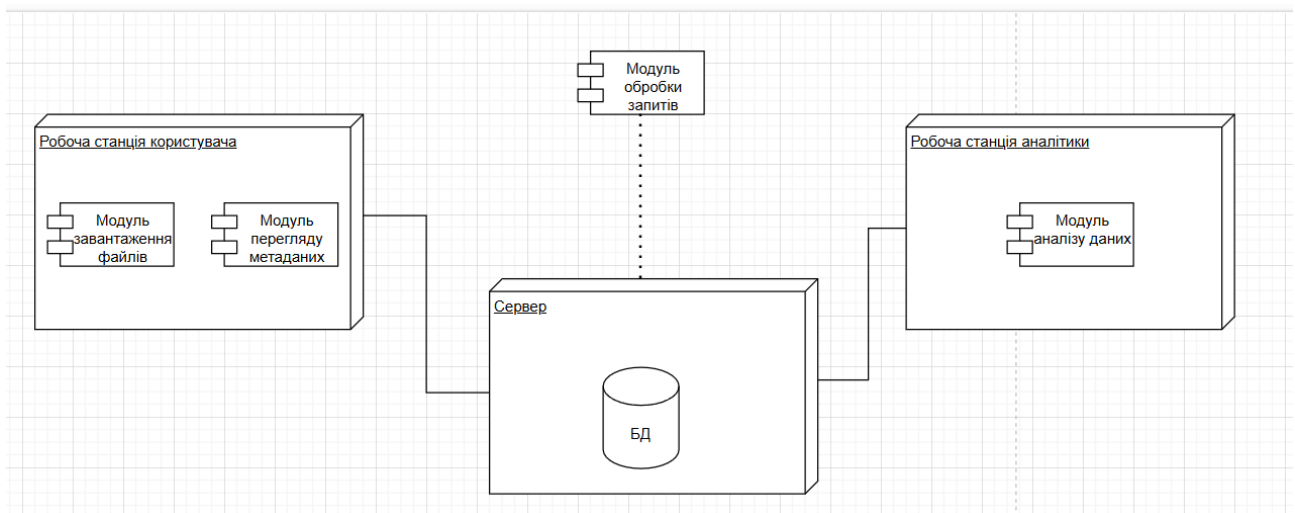


Рис.2.3 Архітектура системи

## 2.2 Вибір системи управління інформаційною базою

MongoDB, популярна відкрита NoSQL база даних, стала порятунком для компаній, які працюють із неструктурованими даними. Завдяки своїй гнучкій та масштабованій архітектурі MongoDB забезпечує ефективне управління великими наборами як структурованих, так і неструктурованих даних. На відміну від традиційних SQL-баз даних, які вимагають попередньо визначеної схеми, MongoDB є безсхемною. Це означає, що вона не прив'язана до жорстких структур і зберігає дані у форматі BSON (Binary JSON) — бінарному представленні документів, схожих на JSON. Такий підхід дозволяє MongoDB легко обробляти неструктуровані дані, забезпечуючи масштабованість і гнучкість.[]

Для нашої системи аналізу метаданих мультимедійних файлів було вирішено використовувати MongoDB (рис.2.4) як основну систему управління базами даних. Це обумовлено такими факторами:

- гнучкість схеми: MongoDB є документо-орієнтованою базою даних, що дозволяє зберігати документи у форматі BSON, подібному до JSON. Це забезпечує можливість динамічно змінювати структуру збережених даних, що є особливо зручним для зберігання метаданих, які можуть суттєво відрізнятися залежно від типу файлу (зображення, аудіо, відео);
- масштабованість: MongoDB добре підходить для роботи з великими обсягами даних і дозволяє легко масштабуватися горизонтально, додаючи нові сервери до системи. Це дозволяє зберігати значну кількість метаданих без втрати продуктивності, що важливо для систем, які можуть обробляти великий потік мультимедійних даних;
- швидкість і продуктивність: Завдяки зберіганню даних у документо-орієнтованому форматі MongoDB забезпечує високу швидкість запису та зчитування даних, що є критично важливим для нашої системи. Мультимедійні файли часто містять багато інформації, яку потрібно швидко обробляти;
- інтеграція з сучасними веб-фреймворками: MongoDB чудово інтегрується з популярними фреймворками, такими як Flask або Node.js, що спрощує розробку веб-інтерфейсу для користувачів.



Рис.2.4 MongoDB

Вибір MongoDB для зберігання та управління інформаційною базою метаданих виправданий її гнучкістю, швидкістю та масштабованістю, що дозволяє ефективно обробляти великі обсяги різномірних даних. Це забезпечує стабільну основу для роботи системи, адаптуючи її до сучасних вимог щодо швидкості та зручності роботи з даними. [29]

## 3 ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

### 3.1 Організаційна структура програмного забезпечення

Для розробки системи аналізу метаданих було обрано мову програмування Python. Це рішення обумовлено такими основними причинами:

- зручність та простота написання коду: Python має синтаксис, який легко зрозуміти та використовувати, що значно прискорює процес розробки. Завдяки цьому створення основних функцій системи, таких як обробка файлів та взаємодія з базою даних, стає значно простішим;
- багатий набір бібліотек для роботи з мультимедійними даними: Python має великий вибір бібліотек, таких як ExifTool, Mutagen для обробки метаданих мультимедійних файлів, Pillow для роботи з зображеннями, FFmpeg для обробки відеофайлів, що є необхідним для цього проєкту. Це дозволяє розробнику швидко інтегрувати та обробляти файли різних форматів;
- розвинена екосистема для веб-розробки: Python добре підходить для створення веб-додатків завдяки фреймворку Flask. Flask дозволяє швидко розробити API і веб-інтерфейс для системи, через який користувачі можуть завантажувати файли та переглядати метадані. Flask також легко інтегрується з базами даних, такими як MongoDB, що спрощує зберігання та доступ до даних;
- підтримка спільноти та наявність документації: Python має велике ком'юніті та велику кількість документації, що дозволяє легко знайти вирішення проблем, які можуть виникнути під час розробки. Це важливий фактор, оскільки забезпечує підтримку на кожному етапі проєкту.

Python був обраний як основна мова програмування через його простоту, багатий набір бібліотек для обробки даних, надійний веб-фреймворк Flask та підтримку спільноти. Це забезпечило ефективний процес розробки та зручність у створенні всіх функціональних можливостей, необхідних для системи аналізу метаданих мультимедійних файлів. [30]

## 3.2 Вибір інструментарію для створення ППЗ

**3.2.1 PyCharm.** PyCharm(рис.3.1) пропонує широкий набір функцій та інструментів, які полегшують процес розробки програмного забезпечення. Його інтуїтивний і зручний інтерфейс дозволяє зосередитися на написанні коду, зменшуючи потребу в тривалому налаштуванні середовища.

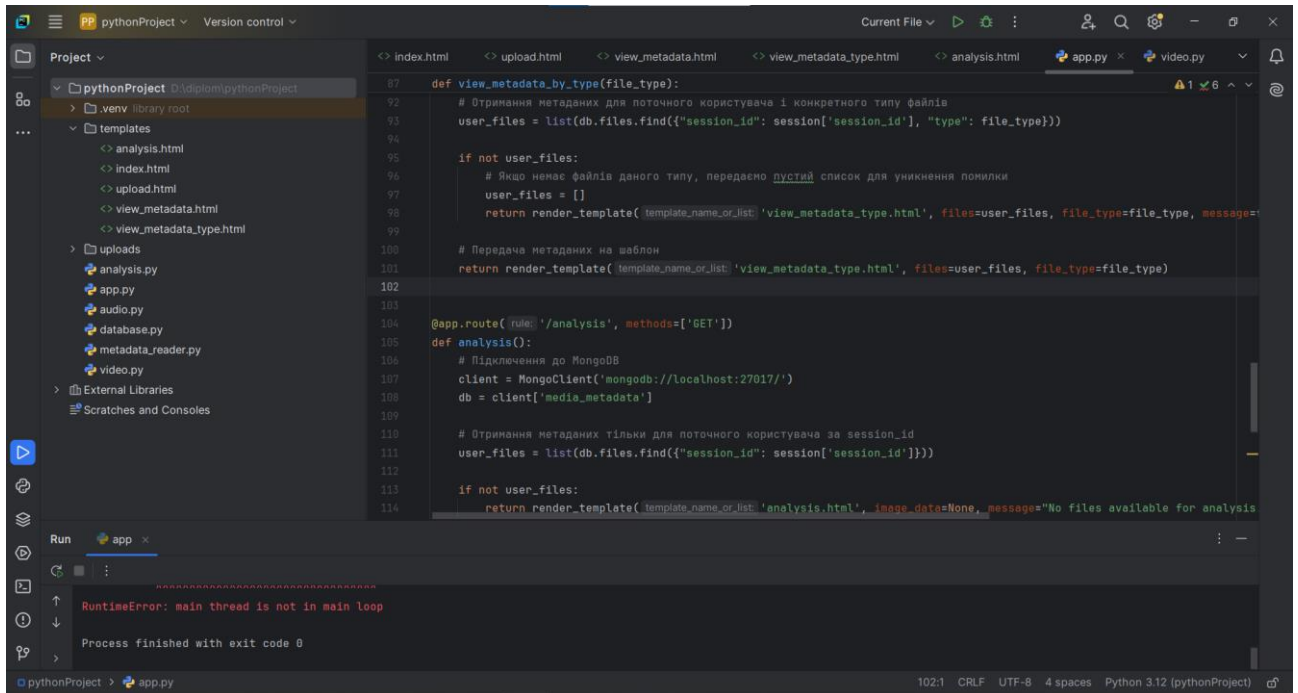


Рис.3.1 Головна сторінка проєкту

Основні переваги PyCharm для Python-розробки включають:

- синтаксичне підсвічування: кольорове виділення синтаксичних елементів для покращеної читабельності коду;
- автодоповнення: інтелектуальне доповнення коду, що пришвидшує написання і знижує ймовірність помилок;
- інтеграція з системами контролю версій: підтримка популярних систем, таких як Git, для зручного відстеження змін;
- відладка: інструменти відлагодження, включаючи точкові зупинки і перегляд значень змінних, що спрощує пошук і виправлення помилок;
- рефакторинг: зручні функції для зміни і покращення коду, такі як перейменування змінних і видалення зайвих частин коду;

- підтримка віртуальних середовищ: можливість ізолювати залежності проекту, забезпечуючи чистоту середовища розробки.

Основні функціональні можливості PyCharm:

1. Редактор коду: потужний редактор із синтаксичним підсвічуванням, автодоповненням, форматуванням за PEP 8 і підтримкою рефакторингу.
2. Управління проєктами: просте налаштування проєктів, підтримка віртуальних середовищ, зовнішніх бібліотек, контролю версій і залежностей.
3. Інтеграція з інструментами: сумісність із системами контролю версій (Git, SVN), базами даних (MySQL, PostgreSQL), веб-серверами (Django, Flask) та іншими.
4. Розширення та плагіни: налаштування функціоналу PyCharm за допомогою плагінів для розширення можливостей.
5. Кросплатформовість: доступність для Windows, macOS і Linux, що дозволяє використовувати PyCharm на будь-якій операційній системі для Python-розробки.

**3.2.2 Flask.** Flask — легкий веб-фреймворк для мови Python, який було обрано для створення серверної частини веб-додатку. Flask забезпечує простоту, дозволяючи швидко розробляти веб-додатки та API завдяки своїй мінімалістичній архітектурі (рис.3.2).



Рис.3.2 Flask

Основні особливості Flask:

- маршрутизація: Flask дозволяє легко організувати маршрути (URL-шляхи) для різних сторінок і функцій, таких як завантаження файлів, перегляд метаданих та аналіз даних;
- шаблонізатор Jinja2: Flask включає підтримку шаблонів Jinja2, що дозволяє створювати динамічний HTML із можливістю вставляти змінні, створювати цикли та умовні структури прямо в шаблонах;
- підтримка розширень: Flask має велику кількість розширень, таких як Flask-PyMongo для інтеграції з MongoDB, Flask-Login для керування сесіями користувачів, що значно полегшує розробку;
- асинхронна обробка: Flask добре працює з AJAX-запитами, що дозволяє створювати інтерактивні елементи веб-інтерфейсу, наприклад, для завантаження файлів та показу повідомлень.

**3.2.3 MongoDB.** MongoDB є NoSQL базою даних, яка ідеально підходить для зберігання метаданих мультимедійних файлів. Її гнучка схематичність дозволяє зберігати дані різного формату та структури:

- гнучка структура: оскільки метадані різних типів файлів можуть мати різну структуру, MongoDB дозволяє зберігати документи з унікальними полями, без жорсткої схеми;
- масштабованість: MongoDB дозволяє легко масштабувати систему в міру зростання кількості файлів, що завантажуються користувачами;
- інтеграція з Python: Бібліотека pymongo забезпечує просте підключення до MongoDB з Python, дозволяючи зберігати, оновлювати та витягувати дані.

**3.2.4 ExifTool, Mutagen і FFmpeg.** ExifTool є потужним інструментом для роботи з метаданими зображень. Він забезпечує доступ до широкого спектра форматів метаданих, включаючи інформацію про камеру, роздільну здатність, дату зйомки тощо:

- широка підтримка форматів: ExifTool підтримує більшість популярних форматів зображень, таких як JPEG, PNG, TIFF, що дозволяє забезпечити універсальність програми;
- масштабованість і швидкодія: ExifTool зчитує дані швидко і може обробляти кілька файлів одночасно, що важливо для роботи з великими обсягами даних.

Mutagen — бібліотека для роботи з метаданими аудіофайлів у Python. Вона дозволяє читати та змінювати метадані, такі як назва пісні, виконавець, альбом, рік випуску:

- підтримка основних аудіоформатів: Mutagen дозволяє працювати з популярними форматами аудіофайлів, включаючи MP3, FLAC, M4A;
- інтеграція з Python: Легко інтегрується з Python-кодом, що дозволяє безпосередньо зчитувати метадані з аудіофайлів у рамках веб-застосунку.

FFmpeg — це набір інструментів для роботи з мультимедійними файлами. У цьому проєкті FFmpeg використовується для отримання метаданих відеофайлів, зокрема, розмірів кадру, тривалості, кодека, частоти кадрів:

- широка підтримка форматів відео: FFmpeg підтримує більшість відеоформатів, включаючи MP4, AVI, MKV;
- потужність і гнучкість: FFmpeg здатний обробляти файли великих розмірів і різних форматів, забезпечуючи точність витягування метаданих.

**3.2.5 Bootstrap.** Bootstrap (рис.3.3) є популярною CSS-бібліотекою, яка забезпечує стильний та адаптивний дизайн для веб-додатка:

- адаптивний дизайн: Використовуючи Bootstrap, можна швидко створювати інтерфейси, які добре відображаються на різних пристроях;
- готові компоненти: Кнопки, форми, карти та навігаційні панелі забезпечують зручність розробки та стандартизований вигляд;
- легка інтеграція з Flask: CSS і JavaScript файли Bootstrap можна легко підключити до шаблонів Flask для покращення вигляду;



Рис.3.3 Bootstrap

Обраний інструментарій для розробки системи є оптимальним поєднанням для забезпечення стабільності, гнучкості та продуктивності. Використання Flask і MongoDB забезпечило надійну серверну частину з високою продуктивністю, а інструменти для обробки метаданих зробили систему універсальною, що дозволяє працювати з різними типами мультимедійних файлів. Інструменти фронтенду, такі як jQuery і Bootstrap, забезпечили зручний та привабливий інтерфейс для користувачів.

### 3.3 Алгоритмізація та програмування програмних модулів

Давайте розглянемо методи які використовуються в коді:

**3.3.1 Модуль завантаження файлів та обробки запитів.** Для взаємодії користувача із сервером Flask використовується кілька методів, які відповідають за завантаження файлів, маршрутизацію запитів та організацію сесії користувача (рис3.4).

- Метод `upload_file()`
  - Опис: `upload_file()` є основною функцією для обробки HTTP POST-запитів, що відповідає за завантаження файлів користувачем.
  - Алгоритм:
    1. При отриманні файлу від користувача, він зберігається у визначеній директорії (`UPLOAD_FOLDER`).
    2. Функція `secure_filename()` використовується для уникнення можливих проблем із назвами файлів.
    3. Визначається тип файлу на основі його розширення: `jpg`, `png` (зображення), `mp3`, `wav` (аудіо), `mp4`, `avi` (відео).

4. Залежно від типу файлу викликається відповідна функція для зчитування метаданих (наприклад, `get_image_metadata()` для зображень).
  5. Зчитані метадані зберігаються у базі даних.
  6. Користувач отримує повідомлення про успіх або помилку, що робить інтерфейс більш інформативним.
- Інструменти: використовується функція `Flask request.files.get()` для отримання файлу та `redirect()` для перенаправлення після завантаження.

```

@app.route(rule='/upload', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        file = request.files.get('file')
        if file:
            try:
                filename = secure_filename(file.filename)
                file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
                file.save(file_path)

                # Виклик функції для зчитування метаданих і збереження у базу даних
                metadata = get_image_metadata(file_path)
                if metadata:
                    save_metadata_to_db(metadata, collection_name="images")
                    flash(message="File uploaded successfully!", category="success")
                else:
                    flash(message="Failed to retrieve metadata.", category="danger")

                # Визначаємо тип файлу і отримуємо метадані
                metadata = None
                if filename.lower().endswith(('jpg', 'jpeg', 'png')):
                    metadata = get_image_metadata(file_path)
                elif filename.lower().endswith(('mp3', 'wav')):
                    metadata = get_audio_metadata(file_path)
                elif filename.lower().endswith(('mp4', 'avi', 'mkv')):
                    metadata = get_video_metadata(file_path)

```

Рис.3.4 Модуль завантаження та обробки

- Метод `before_request()`
  - Опис: Метод генерує унікальний `session_id` для кожного користувача при першому відвідуванні сайту (рис.3.5).
  - Алгоритм:
    1. Перевіряє наявність `session_id` у поточній сесії.
    2. Якщо `session_id` відсутній, створює унікальний ідентифікатор за допомогою `uuid.uuid4()` та зберігає його в сесії.
    3. Забезпечує можливість відстежувати файли, завантажені кожним користувачем, у межах окремої сесії.
  - Інструменти: використовується модуль `uuid` для створення унікальних ідентифікаторів сесій.

```

# Додавання унікального session_id для кожного користувача
@app.before_request
def before_request():
    if 'session_id' not in session:
        session['session_id'] = str(uuid.uuid4()) # Генеруємо унікальний session_id для кожного користувача

```

Рис.3.5 Модуль генерації сесії

**3.3.2. Модуль зчитування метаданих файлів.** Кожен тип файлу (зображення, аудіо, відео) має унікальну структуру метаданих, тому для кожного типу використовуються окремі методи зчитування даних.

- Функція `get_image_metadata(file_path)`
  - Опис: Функція зчитує метадані зображень за допомогою ExifTool (рис.3.6).
  - Алгоритм:
    1. Виконує системну команду для запуску ExifTool з параметрами, що дозволяють отримати метадані у форматі JSON.
    2. Зчитує JSON-дані з метаданими, наприклад, дату створення, розмір та камеру.
    3. Повертає метадані як словник Python для зручної обробки.
  - Інструменти: використовується бібліотека `subprocess` для запуску команд у системі та `json` для роботи з JSON-форматом.

```
# Функція для зчитування метаданих із зображення за допомогою ExifTool
def get_image_metadata(file_path):
    try:
        # Виконати команду ExifTool і отримати результат у JSON-форматі
        result = subprocess.run(
            args=["D:\\exiftool-12.98_64\\exiftool.exe", "-j", file_path],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True
        )

        # Перевірка, чи команда виконалася успішно
        if result.returncode != 0:
            print(f"Error while reading metadata: {result.stderr}")
            return None
```

Рис.3.6 Зчитування метаданих з зображень

- Функція `get_audio_metadata(file_path)`
  - Опис: Зчитує метадані з аудіофайлів за допомогою бібліотеки Mutagen (рис3.7-3.8).
  - Алгоритм:
    1. Mutagen відкриває аудіофайл та зчитує інформацію про виконавця, альбом, трек, рік випуску тощо.
    2. Форматує ключові значення метаданих у зручний для збереження формат (словник).
    3. Повертає оброблені дані як словник.
  - Інструменти: використовується бібліотека `mutagen` для роботи з аудіофайлами у різних форматах, таких як MP3 та WAV.

```

from mutagen import File
from database import save_metadata_to_db

# Функція для зчитування метаданих з аудіо файлу
def get_audio_metadata(file_path): 2 usages
    try:
        audio = File(file_path)
        if audio is not None and audio.tags is not None:
            # Створимо словник для зберігання тільки зрозумілих метаданих
            readable_metadata = {}
            for key in audio.tags.keys():
                value = str(audio.tags[key])

```

Рис.3.7 Зчитування метаданих з аудіофайлів

```

        readable_metadata["FileName"] = file_path
        return readable_metadata
    else:
        print("No metadata found in this audio file.")
        return {}
except Exception as e:
    print(f"Error while reading audio metadata: {e}")
    return None

```

Рис.3.8 Зчитування метаданих з аудіофайлів

- Функція `get_video_metadata(file_path)`
  - Опис: Використовує FFmpeg для зчитування метаданих відеофайлів (рис3.9).
  - Алгоритм:
    1. Виконує команду FFmpeg для отримання інформації про відео, зокрема, кодеки, розмір, тривалість і частоту кадрів.
    2. Форматує метадані у вигляді словника, який можна легко обробити та зберегти.
    3. Повертає результати у форматі словника.
  - Інструменти: використовується FFmpeg через бібліотеку `subprocess`, що дозволяє обробляти інформацію про відео.

```

import subprocess
import json
from database import save_metadata_to_db

def get_video_metadata(file_path): 2 usages
    try:
        # Вказуємо абсолютний шлях до ffprobe
        ffmpeg_path = r"D:\\ffmpeg\\bin\\ffprobe.exe"

        # Використовуємо ffprobe для зчитування метаданих
        result = subprocess.run(
            args=[ffmpeg_path, "-v", "quiet", "-print_format", "json", "-show_format", "-show_streams", file_path],
            stdout=subprocess.PIPE,
            stderr=subprocess.PIPE,
            text=True
        )

        # Перетворюємо вивід JSON у словник Python
        metadata = json.loads(result.stdout)

```

Рис.3.9 Зчитування метаданих з відеофайлів

**3.3.3. Модуль збереження метаданих у базу даних.** Усі зчитані метадані зберігаються в MongoDB для забезпечення доступу до них у майбутньому.

- Функція `save_metadata_to_db(metadata, collection_name)`
  - Опис: Функція зберігає метадані у визначену колекцію бази даних MongoDB (рис.3.10).
  - Алгоритм:
    1. Підключається до бази даних MongoDB та вибирає колекцію за назвою (наприклад, "files").
    2. Включає ідентифікатор сесії користувача (`session_id`) у метадані для подальшої ідентифікації.
    3. Вставляє метадані у вибрану колекцію MongoDB.
  - Інструменти: `rumongo` для інтеграції Python з MongoDB.

```

from pymongo import MongoClient

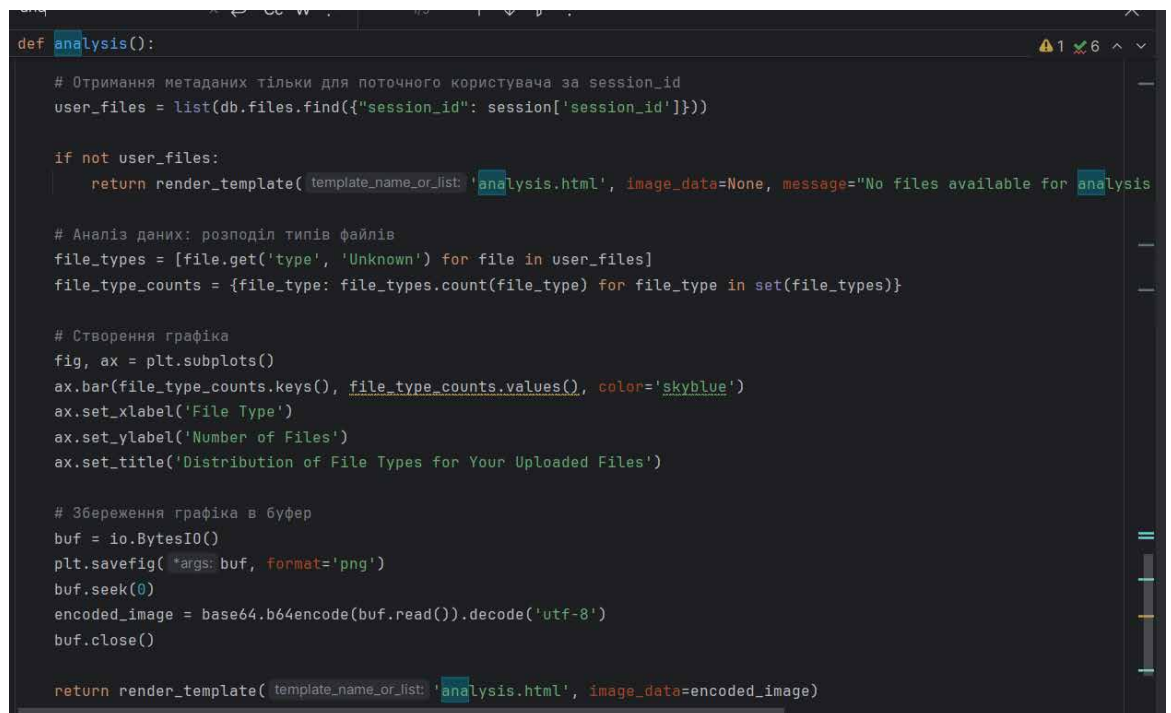
def save_metadata_to_db(metadata, collection_name): 6 usages
    client = MongoClient('mongodb://localhost:27017/')
    db = client['media_metadata']
    collection = db[collection_name]
    collection.insert_one(metadata)

```

Рис.3.10 Модуль збереження в БД

**3.3.4. Модуль аналізу даних.** Даний модуль містить методи для виконання аналізу метаданих, таких як розподіл типів файлів.

- Метод `analysis()`
  - Опис: Метод, що виконує аналіз даних у системі та відображає результати у вигляді графіків(3.11).
  - Алгоритм:
    1. Отримує всі файли поточного користувача з бази даних.
    2. Обчислює кількість файлів кожного типу (зображення, аудіо, відео).
    3. Використовує бібліотеку `matplotlib` для побудови графіка.
    4. Графік кодується у формат `Base64` для відображення на сторінці `HTML`.
  - Інструменти: `matplotlib` для побудови графіків та `base64` для кодування зображення.



```
def analysis():
    # Отримання метаданих тільки для поточного користувача за session_id
    user_files = list(db.files.find({"session_id": session['session_id']}))

    if not user_files:
        return render_template(template_name_or_list='analysis.html', image_data=None, message="No files available for analysis")

    # Аналіз даних: розподіл типів файлів
    file_types = [file.get('type', 'Unknown') for file in user_files]
    file_type_counts = {file_type: file_types.count(file_type) for file_type in set(file_types)}

    # Створення графіка
    fig, ax = plt.subplots()
    ax.bar(file_type_counts.keys(), file_type_counts.values(), color='skyblue')
    ax.set_xlabel('File Type')
    ax.set_ylabel('Number of Files')
    ax.set_title('Distribution of File Types for Your Uploaded Files')

    # Збереження графіка в буфер
    buf = io.BytesIO()
    plt.savefig(*args=buf, format='png')
    buf.seek(0)
    encoded_image = base64.b64encode(buf.read()).decode('utf-8')
    buf.close()

    return render_template(template_name_or_list='analysis.html', image_data=encoded_image)
```

Рис.3.11 Модуль для аналізу метаданих

**3.3.5. Модуль відображення інтерфейсу.** Для рендерингу сторінок використовується `Jinja2` та `Flask`.

- Метод `render_template()`
  - Опис: Функція, яка використовується для відображення `HTML`-шаблонів у `Flask`.
  - Алгоритм:
    1. Вибирає `HTML`-шаблон для рендерингу.
    2. Передає дані, такі як метадані файлів чи аналіз, у шаблон.
    3. Відображає згенеровану `HTML`-сторінку для користувача.

Модульна структура та організація методів у системі дозволяють ефективно виконувати операції з обробки, зберігання та аналізу метаданих мультимедійних файлів. Така система легко масштабується та є гнучкою, що дозволяє її розширення і додавання нових функцій для інших типів файлів або форматів.

## 4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ

### 4.1 Тестування системи

Тестування програмного забезпечення є важливим етапом у процесі розробки, оскільки воно дозволяє перевірити коректність роботи системи, її продуктивність та відповідність поставленим вимогам.

Основна мета тестування нашої системи полягала в забезпеченні її стабільної роботи під час обробки мультимедійних файлів, коректного зчитування та збереження метаданих, а також в оцінці зручності її використання для кінцевого користувача. Процес тестування охоплював перевірку функціональності завантаження файлів різних форматів, таких як зображення, аудіо та відео, їхньої обробки і збереження відповідної інформації у базі даних MongoDB. Також перевірялися можливості системи для візуалізації аналітичних даних, що базуються на метаданих завантажених файлів. Особливу увагу приділено роботі системи в умовах граничних сценаріїв, таких як обробка великих файлів або завантаження файлів невірною формату.

Ось так виглядає головна сторінка системи аналізу метаданих (рис.4.1)

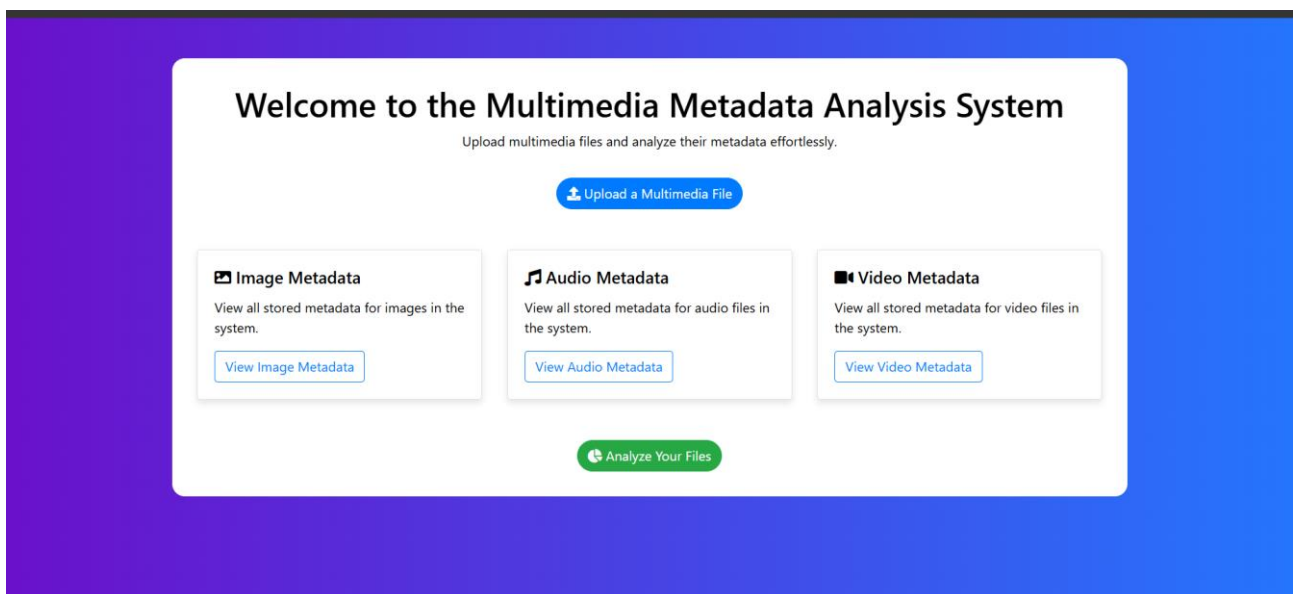


Рис.4.1 Головна сторінка

Завантаження файлу, а саме картинки формату .jpg (рис4.2-4.3) і повідомлення про успішне завантаження.

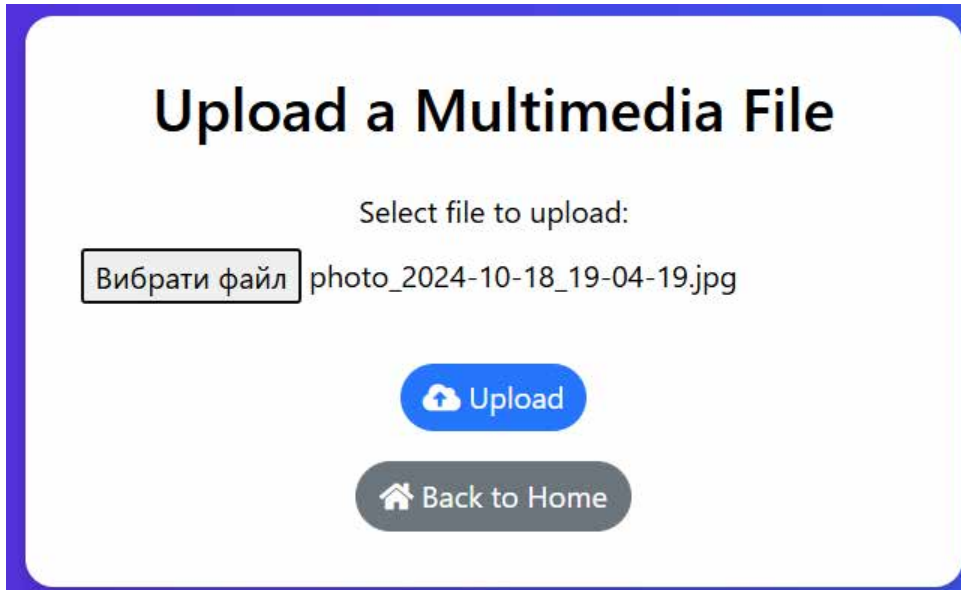


Рис.4.2

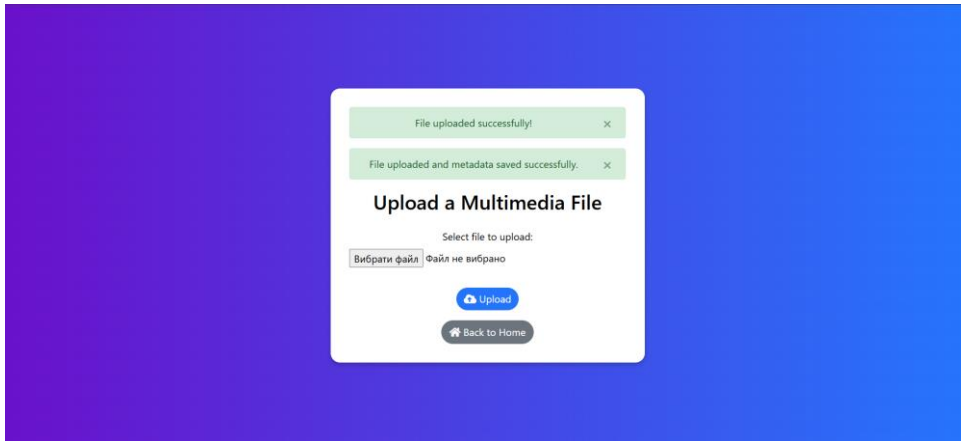


Рис.4.3

Далі переглянемо метадані з наших завантажених зображень (рис.4.4).

photo_2024-10-18_19-04-19.jpg	<a href="#">View Details</a>
<b>SourceFile:</b> uploads/photo_2024-10-18_19-04-19.jpg	
<b>ExifToolVersion:</b> 12.98	
<b>FileName:</b> photo_2024-10-18_19-04-19.jpg	
<b>Directory:</b> uploads	
<b>FileSize:</b> 148 kB	
<b>FileModifyDate:</b> 2024:11:18 01:49:28+02:00	
<b>FileAccessDate:</b> 2024:11:18 01:49:29+02:00	
<b>FileCreateDate:</b> 2024:10:18 19:04:48+03:00	
<b>FilePermissions:</b> -rw-rw-rw-	
<b>FileType:</b> JPEG	
<b>FileTypeExtension:</b> jpg	
<b>MIMEType:</b> image/jpeg	

Рис.4.4

Висвітлення помилки в повідомленні якщо завантажено файл невірною формату (рис.4.5).

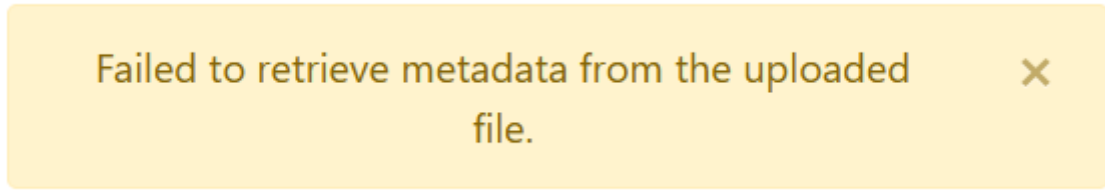


Рис.4.5

Генерація графіка у розділі аналітики (рис.4.6).

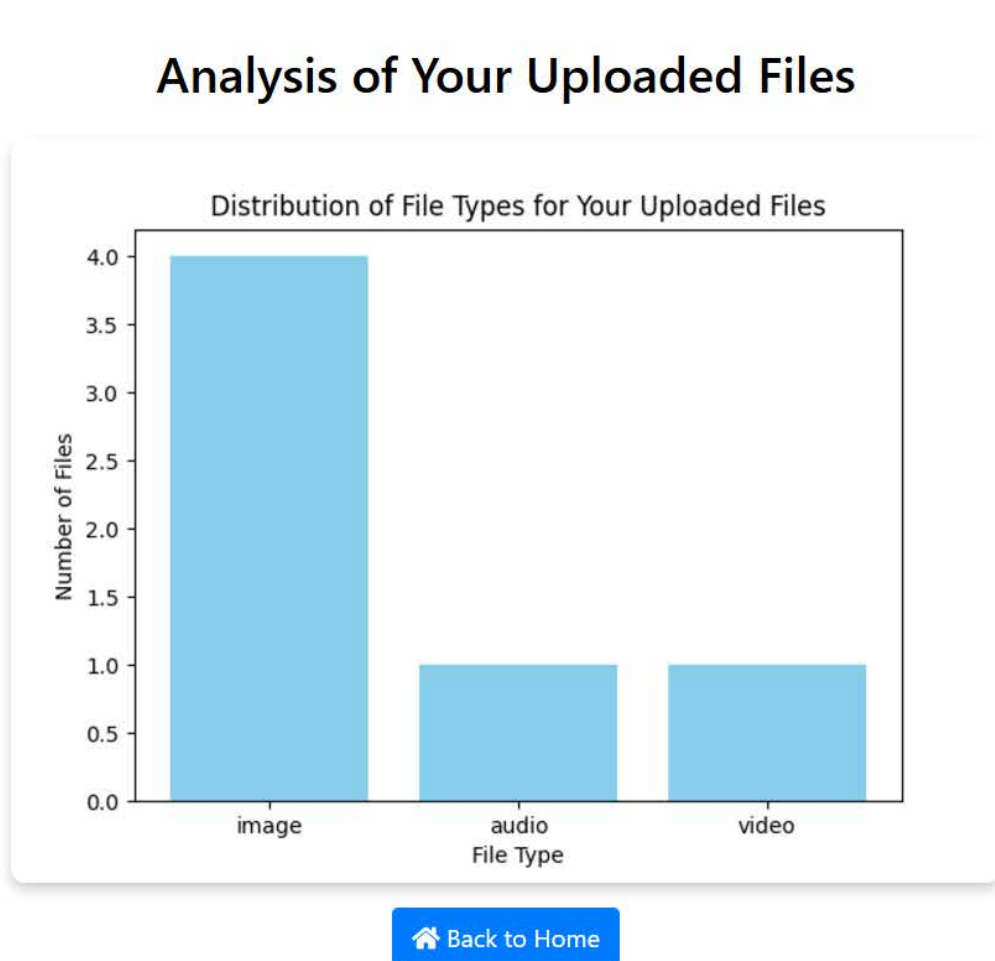


Рис.4.6

Колекції де зберігаються завантажені файли користувачем (рис.4.7), а також вміст однієї з колекцій (рис.4.8).

audio				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	124.00 B	1	36.86 kB
files				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	6	753.00 B	1	36.86 kB
images				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
24.58 kB	15	987.00 B	1	36.86 kB
video				
Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	2	345.00 B	1	36.86 kB

Рис.4.7

```

▼ Show 2 more fields

_id: ObjectId('673a82cb87f449861450e46e')
Artist: "The HARDKISS"
TENC: "iTunes 11.0.4"
Title: "Rain"
Year: "2016"
FileName: "uploads\The_Hardkiss_Rain.mp3"
session_id: "0b764a13-ffb6-4104-a86b-e616990d999c"
type: "audio"

_id: ObjectId('673a82d687f449861450e472')
SourceFile: "uploads/IMG_8467.JPG"
ExifToolVersion: 12.98
FileName: "IMG_8467.JPG"
Directory: "uploads"
FileSize: "1035 kB"
FileModifyDate: "2024:11:18 01:57:09+02:00"
FileAccessDate: "2024:11:18 01:57:09+02:00"
FileCreateDate: "2024:10:22 19:24:55+03:00"
FilePermissions: "-rw-rw-rw-"
FileType: "JPEG"
FileTypeExtension: ".jpg"
MIMEType: "image/jpeg"
JFIFVersion: 1.01
ExifByteOrder: "Big-endian (Motorola, MM)"

```

Рис.4.8

Таблиця із тестовими сценаріями може бути використана для організації процесу тестування.

Таблиця 4.1

№	Сценарій	Очікуваний результат	Результат тесту
1.	Завантаження файлу формату .jpg	Файл успішно завантажено, метадані збережено в базі даних.	Успішно
2.	Завантаження файлу невірною формату	Помилка з повідомленням "Не вдалося визначити метадані".	Успішно
3.	Відображення метаданих для зображень	Виведення основних параметрів: розмір, тип, роздільна здатність.	Успішно
4.	Генерація графіків у розділі аналітики	Графік створено, розподіл типів файлів візуалізовано.	Успішно

Усі ключові сценарії успішно пройдено. Система коректно працює з різними форматами файлів, виконує збереження та обробку метаданих, генерує графіки для аналітики.

Під час тестування було встановлено, що всі основні функції системи працюють коректно. Завантаження файлів, обробка метаданих та їх виведення на екран здійснюються без помилок. У розділі аналітики дані успішно відображаються у вигляді графіків, що дозволяє користувачеві отримати уявлення про розподіл файлів за типами. Було виявлено кілька недоліків, зокрема

некоректна обробка випадків відсутності певних типів файлів у базі даних, але вони були оперативно виправлені.

Проведене тестування також дозволило оцінити зручність використання системи, що є важливим показником для кінцевого користувача. У майбутньому рекомендується впровадити автоматизоване тестування для забезпечення ще більшої надійності системи та спрощення перевірки нових функцій. Таким чином, система готова до використання і може слугувати як надійний інструмент для роботи з метаданими мультимедійних файлів.

## **4.2 Вимоги до апаратного та програмного забезпечення**

Для забезпечення стабільної та ефективної роботи системи аналізу метаданих мультимедійних файлів важливо дотримуватися певних вимог до апаратного та програмного забезпечення. Ці вимоги дозволяють гарантувати коректну роботу системи, забезпечити обробку файлів різних форматів та обсягу.

На рівні апаратного забезпечення мінімальними вимогами є наявність комп'ютера або ноутбука із процесором не нижче Intel Core i3 або його еквівалентом, оперативною пам'яттю об'ємом від 4 ГБ, та накопичувачем із вільним місцем щонайменше 10 ГБ для збереження завантажених файлів. Рекомендованим є використання комп'ютера із більш потужним процесором, таким як Intel Core i5 або i7, 8 ГБ оперативної пам'яті та SSD-накопичувачем для забезпечення швидкого доступу до даних. Крім того, бажано мати монітор із роздільною здатністю Full HD (1920x1080) для комфортного відображення графічного інтерфейсу та результатів аналітики. З програмного забезпечення кінцевий користувач повинен мати операційну систему, сумісну з інструментами розробки, такими як Windows 10 або вище, macOS, або одна з популярних дистрибуцій Linux. Для роботи веб-застосунку необхідно мати встановлений сучасний веб-браузер, наприклад Google Chrome, Mozilla Firefox або Microsoft Edge, із підтримкою HTML5 і CSS3. Для роботи веб-застосунку користувачеві не потрібно окремо встановлювати додаткове програмне забезпечення, оскільки всі необхідні компоненти інтегровані у веб-систему. Система створена на основі

Flask, що забезпечує легкий запуск і доступ до функціональності через веб-інтерфейс. Бажано також мати стабільне інтернет-з'єднання, якщо система передбачає використання віддалених серверів для зберігання даних або хостингу.

Таким чином, ці вимоги до апаратного та програмного забезпечення дозволяють забезпечити плавну і стабільну роботу системи для кінцевого користувача, гарантують коректну обробку даних і доступ до результатів аналізу через зручний інтерфейс.

## ВИСНОВКИ

На основі виконаного проекту, система аналізу метаданих мультимедійних файлів демонструє значний практичний потенціал для автоматизації, обробки та аналізу метаданих різних типів мультимедіа. У межах дослідження були виявлені актуальні аспекти використання метаданих, зокрема їх важливість для організації, управління й аналізу великих обсягів інформації.

Розроблена система забезпечує обробку зображень, аудіо та відео, включаючи можливість детального аналізу ключових параметрів, таких як роздільна здатність, бітрейт, тривалість, частота кадрів тощо. Крім цього, завдяки використанню сучасних інструментів, таких як Flask для створення веб-інтерфейсу та MongoDB для зберігання метаданих, система є гнучкою й масштабованою.

У межах проекту ми не лише реалізували технічну складову, а й проаналізували стандарти метаданих, такі як EXIF, ID3 та XMP, а також інструменти, які використовуються для роботи з ними, наприклад, ExifTool і Mutagen. Це дало змогу оцінити ефективність різних підходів до обробки метаданих, забезпечити захист даних і покращити продуктивність системи. Виконаний аналіз охоплює ключові аспекти роботи з метаданими, включаючи їхню безпеку, конфіденційність і правові аспекти. Це дозволяє системі відповідати сучасним вимогам щодо обробки даних і забезпечує її застосування у таких сферах, як журналістика, маркетинг, цифрова криміналістика й медіа-аналітика.

На основі отриманих результатів можна зробити висновок, що розроблена система має значний потенціал для подальшого вдосконалення. Наприклад, можна інтегрувати можливості візуалізації даних для глибшого аналізу взаємозв'язків між різними параметрами метаданих або застосувати машинне навчання для автоматичного виявлення аномалій у даних. Таким чином, ця система є важливим внеском у розвиток технологій аналізу мультимедійних даних.

## СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What Is Metadata and Why Is It Critical to a Forensic Investigation? [Електронний ресурс] – Режим доступу до ресурсу: <https://cornerstonediscovery.com/what-is-metadata-and-why-is-it-critical-to-a-forensic-investigation/>
2. ExifTool Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://exiftool.org/>
3. Adobe XMP Specification [Електронний ресурс] – Режим доступу до ресурсу: <https://www.adobe.com/products/xmp.html>
4. FFmpeg Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://ffmpeg.org/documentation.html>
5. Mutagen Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://mutagen.readthedocs.io>
6. IPTC Metadata Standard [Електронний ресурс] – Режим доступу до ресурсу: <https://iptc.org/standards/photo-metadata/iptc-standard/>
7. Dublin Core Metadata Initiative [Електронний ресурс] – Режим доступу до ресурсу: <https://dublincore.org/>
8. Python Hachoir Library Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://hachoir.readthedocs.io>
9. Elasticsearch Reference Guide [Електронний ресурс] – Режим доступу до ресурсу: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
10. Коваленко А. В. Використання цифрових технологій в криміналістиці і судовій експертизі: матеріали міжнар. наук.-практ. круглого столу. Метадані як джерело криміналістично значущої інформації. 2024. С. 82–85.
11. Технічний експерт: Журналісти повинні знати про небезпеку метаданих [Електронний ресурс] – Режим доступу до ресурсу: <https://safety.rsf.org/ukr/tech-expert-journalists-must-be-aware-of-the-dangers-of-metadata/>

12. Використання метаданих в сучасному світі [Електронний ресурс] – Режим доступу до ресурсу: <https://jpasmd.donnu.edu.ua/article/view/12969>
13. General Data Protection Regulation (GDPR) [Електронний ресурс] – Режим доступу до ресурсу: <https://gdpr-info.eu/>
14. Media forensics on social media platforms: a survey [Електронний ресурс] – Режим доступу до ресурсу: <https://jis-urasipjournals.springeropen.com/articles/10.1186/s13635-021-00117-2>
15. AWS Lambda Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/lambda/documentation/>
16. Все, що потрібно знати про шифрування [Електронний ресурс] – Режим доступу до ресурсу: <https://safety.rsf.org/ukr/all-you-need-to-know-about-encryption/>
17. Реалізація системи розподіленого кешу з Redis та Memcached [Електронний ресурс] – Режим доступу до ресурсу: <https://peerdh.com/uk/blogs/programming-insights/implementing-a-distributed-cache-system-with-redis-and-memcached>
18. Ядро процесора, багатоядерність, потоки, ядро проти потоків, гіперпотоківість [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/cpu-core-multicore-thread.html>
19. The Difference Between Brotli And Gzip Compression Algorithms To Speed Up Your Site [Електронний ресурс] – Режим доступу до ресурсу: <https://www.siteground.com/blog/brotli-vs-gzip-compression>
20. Загальний регламент про захист даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gdpr.org.ua/>
21. Що таке фотостоки і чому ви маєте працювати з ними? [Електронний ресурс] – Режим доступу до ресурсу: <https://fotopark.ua/stock-photography/>
22. Гучні штрафи за порушення GDPR: що пішло не так? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.avitar.legal/post/guchni-shtrafi-za-porushennya-gdpr-shcho-pishlo-ne-tak>

23. Міжнародні угоди у сфері інтелектуальної власності [Електронний ресурс] – Режим доступу до ресурсу: <https://patent.km.ua/ukr/articles/i177?nc=1>

24. Проекція у майбутнє. Які п'ять ключових трендів медіаринку визначають споживання контенту та масштабуються на будь-який бізнес? [Електронний ресурс] – Режим доступу до ресурсу: <https://forbes.ua/lifestyle/proektsiya-u-maybutne-pyat-klyuchovikh-trendiv-mediarynku-yaki-viznachat-spozhyvannya-kontentu-kolonka-smo-starlight-media-oleni-martinovoi-05032024-19636>

25. E-commerce: що треба знати і які зараз тренди? [Електронний ресурс] – Режим доступу до ресурсу: <https://online.novaposhta.education/blog/e-commerce-shho-treba-znati-i-yaki-zaraz-trendi>

26. Використання метаданих для вирішення задач великих даних [Електронний ресурс] – Режим доступу до ресурсу: <https://dspace.nbu.gov.ua/handle/123456789/161489>

27. Smart cities: the role of Internet of Things and machine learning in realizing a data-centric smart environment [Електронний ресурс] – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s40747-023-01175-4>

28. MongoDB Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs>

29. What is Python? [Електронний ресурс] – Режим доступу до ресурсу: <https://opensource.com/resources/python>