

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:

В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«___» червня 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань _____ 12 Інформаційні технології

(шифр і назва галузі знань)

спеціальність _____ 125 Кібербезпека

(код і назва спеціальності)

освітній ступень _____ бакалавр

освітня програма _____ Кібербезпека

(назва освітньо-професійної програми)

на тему: _____ Програмний модуль обробки метаданих для захисту

_____ конфіденційності в мультимедійних файлах

Виконавець: студент IV курсу, групи КБ-42

_____ Павло ГОРБАТЮК

(підпис)

(ім'я, прізвище)

	Ім'я, прізвище	Підпис
Керівник	Іван ПАРХОМЕНКО	

Нормоконтроль	Лариса МИРУТЕНКО	
---------------	------------------	--

Київ 2023

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:

В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

спеціальності _____ 125 Кібербезпека
(код і назва спеціальності)
освітньої програми _____ Кібербезпека
(назва освітньо-професійної програми)

Студенту _____ **КБ-42** _____ **Горбатюку Павлу Володимировичу**
(група) (прізвище ім'я по батькові)

Тема кваліфікаційної роботи _____ Програмний модуль обробки метаданих для захисту
конфіденційності в мультимедійних файлах

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №3 від 20.10.2022 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Класифікація метаданих, стандарти та формати метаданих для мультимедійних файлів, процес забезпечення конфіденційності мультимедійних файлів

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Нормативно-правова база у сфері захисту інформації, класифікація метаданих, стандарти та формати метаданих для мультимедійних файлів, загрози конфіденційності, методи та техніки обробки метаданих, архітектура програмного модуля, програмний застосунок

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність _____ Розробка програмного модуля для підвищення захисту конфіденційності мультимедійних файлів шляхом ефективної обробки метаданих

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 24 жовтня 2022 року

Завдання видав

(підпис)

Іван ПАРХОМЕНКО

(ім'я, прізвище)

Завдання прийняв
до виконання

(підпис)

Павло ГОРБАТЮК

(ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	24.10.2022 – 03.11.2022	виконано
2	Аналіз літератури	10.01.2023 – 15.02.2023	виконано
3	Розгляд стандартів метаданих	20.02.2023 – 06.03.2023	виконано
4	Аналіз загроз конфіденційності пов'язаних з метаданими	08.03.2023 – 20.03.2023	виконано
5	Дослідження сучасних методів обробки метаданих	21.03.2023 – 02.04.2023	виконано
6	Опис розроблюваного рішення	03.04.2023 – 06.04.2023	виконано
7	Розробка програмного рішення для обробки метаданих	08.04.2023 – 12.05.2023	виконано
8	Оформлення пояснювальної записки	13.05.2023 – 30.05.2023	виконано
9	Підготовка до захисту кваліфікаційної роботи	31.05.2023 – 12.06.2023	виконано

Завдання видав

(підпис)

Іван ПАРХОМЕНКО

(ім'я, прізвище)

Завдання прийняв
до виконання

(підпис)

Павло ГОРБАТЮК

(ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 12 червня 2023 року

РЕФЕРАТ

Пояснювальна записка складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 61 сторінки, включає в себе зміст, вступ, три розділи кваліфікаційної роботи, висновки та список джерел. Крім того, робота містить 4 додатки із загальною кількістю сторінок 11. У пояснювальній записці кваліфікаційної роботи міститься 14 рисунків, 3 таблиці та 25 посилань на літературні джерела.

Метою роботи є розробка програмного модуля для забезпечення ефективної обробки та керування метаданими в мультимедійних файлах.

Об'єктом дослідження є процес надання конфіденційності мультимедійним файлам шляхом обробки їх метаданих.

Предметом дослідження є набір механізмів для забезпечення конфіденційності метаданих в мультимедійних файлах.

Методи дослідження:

- аналіз відкритих джерел;
- аналіз загроз конфіденційності метаданих;
- моделювання засобу обробки метаданих.

Практичною цінністю є розроблений програмний модуль для перегляду та обробки метаданих шляхом їх видалення у мультимедійних файлах.

Ключові слова: метадані, кіберпростір, кіберзагроза, конфіденційність, захист даних, методи обробки.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ МЕТАДАНИХ В МУЛЬТИМЕДІЙНИХ ФАЙЛАХ, ЇХ КЛАСИФІКАЦІЯ ТА СТАНДАРТИ.....	11
1.1 Законодавство з питань конфіденційності інформації, яка існує у метаданих	11
1.2 Визначення та класифікація метаданих	13
1.3 Огляд стандартів метаданих.....	17
1.4 Стандарти метаданих для мультимедійних файлів	21
1.4.1 Стандарт метаданих IPTC-ІІМ.....	23
1.4.2 Стандарт метаданих PLUS	24
1.4.3 Стандарт метаданих XMP	25
1.4.4 Стандарт метаданих IPTC Core & Extension	26
1.4.5 Стандарт метаданих EXIF	27
1.4.6 Стандарт метаданих Dublin Core.....	28
Висновки до розділу 1	29
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ЗАГРОЗ КОНФІДЕНЦІЙНОСТІ ТА МЕТОДІВ ОБРОБКИ МЕТАДАНИХ	31
2.1 Типові загрози конфіденційності, пов'язані з метаданими	31
2.1.1 Розкриття геолокації та особистої інформації	31
2.1.2 Викриття технічних характеристик.....	34
2.2 Огляд сучасних методів та технік обробки метаданих для забезпечення конфіденційності	35
Висновки до розділу 2	40

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ОБРОБКИ МЕТАДАНИХ ДЛЯ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ В МУЛЬТИМЕДІЙНИХ ФАЙЛАХ	41
3.1 Визначення основних вимог до програмного модуля.....	41
3.2 Архітектура програмного модуля.....	43
3.3 Вибір технологій та інструментів розробки	45
3.3.1 Python.....	45
3.3.2 Visual Studio Code	49
3.4 Реалізація основних функціональних компонентів.....	50
Висновки до розділу 3	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТОК А.....	62
ДОДАТОК Б.....	67
ДОДАТОК В	69
ДОДАТОК Г.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ІБ	– Інформаційна безпека;
ІС	– Інформаційна система;
ISO	– International Organization for Standardization
NISO	– National Information Standards Organization
ПЗ	– Програмне забезпечення
ІРТС-ІІМ	– Модель обміну інформацією Міжнародної ради з питань преси та телекомунікацій
XMP	– Extensible Metadata Platform
EXIF	– Exchangeable Image File Format
OSINT	– Open source intelligence
ОС	– Операційна система
GDPR	– General Data Protection Regulation

ВСТУП

Ми живемо в еру цифрових технологій, коли мультимедіа стає важливою складовою нашого повсякденного життя. Перебуваючи в епіцентрі таких областей, як комунікація, розваги, освіта, наука, охорона здоров'я, правоохоронні дії та багатьох інших, мультимедіа впливає на різноманітні аспекти суспільного життя. Тому інформаційна ера, в якій ми живемо сьогодні, розкриває перед людством неоціненні можливості в області обміну та використання даних. За даними We Are Social та Hootsuite, станом на початок 2023 року у світі проживає більше 8 мільярдів людей, з яких понад 5 мільярдів має доступ та користується інтернетом, тобто 64% населення [1]. Порівняно з 2022 роком приріст користувачів інтернету сягає приблизно 2%, що становить майже 100 мільйонів людей. У свою чергу у світі налічується 4,76 мільярда користувачів соціальних мереж, що становить трохи менше 60 відсотків населення планети. Проте протягом останніх місяців зростання кількості користувачів соціальних медіа сповільнилося, але це не заважає постійно та стрімко збільшувати кількість активних користувачів і зараз це число дорівнює 140 мільйонів в рік. Не хотілося б робити гучних припущень, але кожний рік відсоток приросту активних користувачів як інтернету так і соціальних мереж буде поступово збільшуватися, оскільки багато цифрової техніки стає більш доступною для людей, наприклад смартфони, які зараз можуть коштувати 100-200 доларів США, але навіть такі дешеві рішення зможуть кожного зробити повноцінним користувачем інтернету.

Одним з найбільш важливих і широко використовуваних видів даних, які циркулюють в мережі інтернет – мультимедіа – відео, аудіо, фотографії. Вони використовуються для комунікації, навчання, розваг, реклами та багатьох інших цілей, впливаючи на численні аспекти нашого життя [2]. Згідно з даними, наведеними в дослідженні Cisco Visual Networking Index, обсяг мультимедійного контенту в глобальній мережі досягне 4.8 зетабайтів щомісяця до 2022 року, що в 5 разів більше, ніж у 2017 році [3]. Мультимедійні файли містять не тільки безпосередньо вміст, але й додаткову інформацію – метадані. Метадані – це інформація, яка описує основні дані, і вони можуть включати різні параметри, такі як час створення файлу,

місцезнаходження, де він був створений, технічні характеристики обладнання, що було використано для його створення, і так далі.

Метадані використовуються в багатьох областях, включаючи наукові дослідження, маркетинг, розробку програмного забезпечення та інше. Проте, наявність такої інформації викликає питання про конфіденційність та ІБ. Чи можуть наші мультимедійні файли використовуватися для збору даних про нас без нашого відома? Чи можуть метадані стати інструментом в руках недобросовісних осіб або організацій?

Ці питання стають особливо актуальними в контексті глобального переходу до цифрового формату життя. Кожна дія в цифровому світі залишає сліди, які можуть бути зібрані та проаналізовані. Часто ця інформація збирається і використовується без нашої згоди. В результаті, згідно з даними Wakefield Research, приблизно 80% американців вважають, що вони дуже мало або зовсім не контролюють свої особисті дані після того, як вони передаються компаніям [4].

Особливо актуальною проблема метаданих стає в контексті сучасних військових конфліктів, таких як війна в Україні. Відома інформація може бути важливою для військових операцій та мати критичні наслідки як для ІБ так і для безпеки загалом. Особливо важливою є інформація, яка може бути використана для визначення місцезнаходження людей або об'єктів, наприклад корегування вогню для ліквідації ворогом сил ППО, що демонструється численними випадками використання геолокаційних даних з метаданих фотографій .

Також важливо пам'ятати, що метадані – це не просто "додаткова" інформація. Вони можуть надати дуже точну і детальну картину наших життів, інтересів, звичок і навіть місця проживання. Це включає в себе всі фотографії, які ми постимо в соціальних мережах, всі аудіофайли, які ми слухаємо, і всі відео, які ми дивимося. Крім того, ці дані можуть бути використані для створення точних профілів користувачів, що в свою чергу можуть бути використані для різних цілей, від реклами до спостереження.

Таким чином, виходить, що проблема метаданих в мультимедійних файлах є не лише технологічною, але й етичною. Користувачі, мають право знати, яка інформація

про них збирається, як вона використовується і як вони можуть контролювати цей процес. Тож **актуальність роботи** полягає в розробці програмного модуля для управління та видалення метаданих мультимедійних файлів, що дає змогу користувачам бути обізнаними у тому, які персональні дані вони відправляють та можливість їх видалити.

Метою роботи є розробка програмного модуля, який забезпечує ефективну обробку та керування метаданими в мультимедійних файлах.

Для досягнення зазначеної мети кваліфікаційної роботи поставлено наступні **завдання**:

- Дослідити класифікацію та стандарти метаданих.
- Провести аналіз можливих загроз на конфіденційність з використанням метаданих у файлах мультимедіа.
- Провести аналіз сучасних методів та технік обробки метаданих для забезпечення конфіденційності.
- Розробити програмний модуль для видалення метаданих з мультимедійних файлів.

Об'єктом дослідження є процес надання конфіденційності мультимедійним файлам шляхом обробки їх метаданих.

Предметом дослідження є набір механізмів, що реалізують процес надання конфіденційності мультимедійним файлам.

Методи дослідження:

- аналіз відкритих джерел;
- аналіз існуючих методів та технік обробки метаданих;
- моделювання архітектури програмного модуля.

Практичною цінністю є розроблений програмний модуль для перегляду та обробки метаданих шляхом їх видалення у мультимедійних файлах.

РОЗДІЛ 1

АНАЛІЗ МЕТАДАНИХ В МУЛЬТИМЕДІЙНИХ ФАЙЛАХ, ЇХ КЛАСИФІКАЦІЯ ТА СТАНДАРТИ

1.1 Законодавство з питань конфіденційності інформації, яка існує у метаданих

Обробка метаданих нерозривно пов'язана з використанням інформації. Закон України "Про інформацію" у своєму корінному визначенні регулює відносини, що стосуються створення, збирання, одержання, зберігання, використання, поширення, охорони та захисту інформації [5]. Метадані, що є структурованою інформацією про дані, що містять у собі різну інформацію, також підпадають під дію цього закону. Одним з важливих елементів закону є його визначення суб'єктів і об'єктів інформаційних відносин. В контексті метаданих, суб'єктами можуть бути користувачі, які створюють або використовують дані; організації, які збирають, зберігають або аналізують метадані; та провайдери послуг, які забезпечують платформи або інструменти для обробки метаданих. Об'єктом цих відносин є самі метадані.

Закон також визначає конфіденційну інформацію, яка включає інформацію про фізичну особу, та встановлює, що така інформація може бути поширена лише за бажанням або згодою відповідної особи. Це має безпосередній вплив на обробку метаданих в ІС, оскільки вони часто містять персональні дані користувачів, такі як дата та час створення файлу, місцезнаходження та інші подібні дані. Зокрема, якщо метадані містять таку інформацію, їх можна розглядати як конфіденційну інформацію, яку потрібно відповідно захищати.

Відповідно до Закону України "Про інформацію", метадані мультимедійних файлів, що знаходяться у володінні державних органів, органів місцевого самоврядування, підприємств, установ та організацій, можуть бути доступними громадянам. Однак розповсюдження таких даних може бути обмежено, особливо

якщо метадані містять конфіденційну інформацію. Закон також вимагає захисту цих даних від несанкціонованого доступу, розкриття та використання.

Використовуючи метадані, особливо ті, що містяться в мультимедійних файлах, може бути отримана значна кількість інформації про особу, включаючи географічні дані, дати, час, моделі пристрою, авторство та інше. Ці дані відповідають визначенню персональних даних в Законі України "Про захист персональних даних", як відомостей чи сукупності відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована [6]. Персональні дані – відомості чи сукупність відомостей про фізичну особу, яка ідентифікована або може бути конкретно ідентифікована.

Закон визначає правила та обмеження, яких необхідно дотримуватися при обробці персональних даних. Це стосується законності, цільових обмежень, точності, обмежень щодо зберігання та інших міркувань. Цих принципів необхідно дотримуватися і при обробці метаданих.

Також Закон України "Про захист персональних даних" контролює те, що будь-яка циркуляція даних (або метаданих, якщо дані, що містяться в них, відносяться до персональних даних) в ІС вимагає отримання згоди суб'єкта даних на обробку його персональних даних.

Будь-який суб'єкт даних, встановлений цими законом, має право доступ до власних даних, право на виправлення, видалення даних, право відмовитися від обробки даних та інше. В контексті метаданих мультимедійних файлів суб'єкт має повне право на видалення всіх можливих метаданих – місце розташування, де було здійснено фотографія або відео, ім'я автора файлу, назву пристрою за допомогою якого було зроблено файл та інше, їх зміну для збереження конфіденційності.

На міжнародному рівні існують різні закони та нормативно-правові акти, які регулюють захист метаданих. Одним з найбільш значних є Загальний регламент про захист даних (GDPR), що діє в Європейському Союзі з 2018 року. GDPR став прапором приватності даних на глобальному рівні, встановлюючи строгі вимоги до обробки даних, включаючи метадані, та надаючи особам більше контролю над їх

особистими даними. Зокрема, GDPR вимагає від організацій отримати згоду особи на обробку її даних, включаючи метадані [7].

1.2 Визначення та класифікація метаданих

В добу технологій і інформаційної ери цифрові зображення та фотографії стали невід'ємною частиною нашого життя. Вони допомагають нам документувати наше життя, висловлювати нашу творчість і спілкуватися з іншими людьми тощо. Але в то ж саме час вони створюють серйозні виклики для приватності.

Цифрові зображення і фотографії містять метадані – фактично дані про дані. Вони забезпечують структуровану описову інформацію про ресурси і, в свою чергу, сприяють збереженню, пошуку, організації та розумінню даних [8]. Метадані можна вважати за карточку каталогу в бібліотеці, яка описує, що саме міститься в книзі: вона містить назву книги, автора, видавництво, дату публікації та тематику книги. Метадані мультимедійних файлів можуть включати деталі про те, коли і де була зроблена фотографія, модель камери, яка була використана, і навіть GPS координати місця зйомки. Ця інформація може бути корисною, але також може становити серйозну загрозу для приватності.

Існує кілька різних типів метаданих, кожен з яких виконує свою власну роль. Деякі з них вбудовані в файли цифрових зображень і можуть бути прочитані або модифіковані з використанням відповідного програмного забезпечення. Інші зберігаються окремо в супутніх файлах або базах даних. Три основних класи метаданих, які використовуються в цифрових зображеннях, включають EXIF, IPTC та XMP. Кожен з цих класів метаданих містить різну інформацію та використовується для різних цілей.

EXIF метадані, наприклад, використовуються для зберігання технічної інформації про фотографію, такої як модель камери, часовий штамп і налаштування експозиції. IPTC метадані, з іншого боку, використовуються для зберігання інформації про авторство, права на копію та ключові слова. XMP метадані, в свою

чергу, є універсальним форматом, який може містити будь-яку інформацію, яка не входить до класів EXIF або IPTC.

Окрім цього, супутні файли та бази даних метаданих використовуються для зберігання додаткової інформації, яка не може бути вбудована в сам файл зображення. Це може включати дані про авторство, права на копію, відомості про ліцензію, ключові слова для пошуку та іншу інформацію, яка допомагає керувати, використовувати і поширювати цифрові зображення.

Метадані є інструментом для класифікації, організації та опису даних або вмісту. Національна організація з інформаційних стандартів (NISO) пропонує систему, що придатна для категоризації всіх видів даних або репозиторіїв, від бібліотек до веб-сайтів, текстових та не текстових матеріалів, в цифровому або фізичному форматі.

Відповідно до NISO, метадані поділяються на три типи [9]:

- Описові метадані включають інформацію, як-то контактні дані, назву або автора публікації, резюме роботи, ключові слова, географічне положення або пояснення методології. Ця інформація допомагає виявляти, збирати або групувати ресурси за спільними характеристиками.

- Структурні метадані використовуються для опису способу організації набору об'єктів. Вони можуть включати такі елементи, як "ідентифікація файлу", що допомагає визначити конкретний розділ, який є частиною більшого документа, такого як книга або звіт. "Кодування файлу" може бути використане для визначення використаних кодів для обробки файлу, включаючи використаний стандарт кодування даних, такий як ASCII. Інші приклади структурних метаданих включають методи стиснення і шифрування файлу, відтворення файлу, що вказує на те, як і з допомогою чого файл був створений (наприклад, з використанням певного програмного забезпечення, операційної системи або апаратного забезпечення), "структура вмісту", яка вказує на організацію вмісту запису (наприклад, визначення набору даних, словник даних, файли з кодами доступу), а також "джерело", що ідентифікує обставини, які спричинили збір даних.

- Адміністративні метадані використовуються для управління ресурсами. Вони можуть включати дати створення або надходження, права доступу, права або походження, або правила використання, такі як зберігання або видалення. Цей тип метаданих є корисним для адміністраторів баз даних або для тих, хто займається обробкою даних з телекомунікаційного трафіку або мереж передачі даних.

Коротка характеристика типів метаданих наведена у Таблиці 1.1.

Таблиця 1.1

Типи метаданих та їх опис

Тип метаданих	Короткий опис
Описові метадані	Пошук або розуміння ресурсу
Структурні метадані	Взаємозв'язок частин ресурсів між собою
Адміністративні метадані <ul style="list-style-type: none"> • Технічні метадані • Метадані збереження • Метадані прав 	<ul style="list-style-type: none"> • Для декодування та рендерингу файлів • Довгострокове управління файлами • Захист прав інтелектуальної власності на контент

Також існує ще один тип метаданих – мова розмітки. Вона поєднує метадані та вміст разом, що лише іноді використовується з іншими формами метаданих. Прапорці, що вставляються у вміст, позначають помітні особливості. Для текстового ресурсу це може означати позначення структурних елементів, таких як абзаци; позначення слів із семантичною інформацією, наприклад, що це слово є географічною назвою або певною частиною мови; надання інформації про форматування, наприклад курсив.

Різні категорії метаданих підтримують різні варіанти використання в ІС. Пошук є, мабуть, найпоширенішим, оскільки структуровані метадані дозволяють користувачам шукати або переглядати ресурси чи інформацію, що їх цікавлять.

Метадані підтримують управління цифровими об'єктами, надаючи інформацію, необхідну для належного відображення цифрового контенту або надання відповідної версії, що відповідає потребам користувача. Збереження досягається шляхом створення метаданих, які дозволяють перевірити цілісність контенту після перенесення та в інші важливі моменти. Метадані підтримують навігацію всередині частин об'єктів, наприклад, від однієї сторінки чи розділу до іншої, а також між різними версіями об'єктів, наприклад, з різною роздільною здатністю фотографічних зображень. Перелік властивостей, що можуть міститися у кожному типі метаданих та їх використання наведено у Таблиці 1.2.

Таблиця 1.2

Використання типів метаданих

Тип метаданих	Приклад властивостей	Використання
Описові метадані	<ul style="list-style-type: none"> • Назва • Автор • Тема • Жанр • Дата публікації 	<p>Пошук файлу</p> <p>Відображення файлу</p>
Структурні метадані	<ul style="list-style-type: none"> • Тип файлу • Розмір файлу • Дата/час створення • Метод стиснення 	<p>Сумісність з іншими системами</p> <p>Управління цифровими об'єктами</p> <p>Збереження</p>
Технічні метадані	<ul style="list-style-type: none"> • Контрольна сума • Подія збереження 	<p>Сумісність з іншими системами</p> <p>Управління цифровими об'єктами</p> <p>Збереження</p>
Метадані прав	<ul style="list-style-type: none"> • Місце в ієрархії • Послідовність 	<p>Навігація</p>

Тип метаданих	Приклад властивостей	Використання
Метадані збереження	<ul style="list-style-type: none"> • Статус авторського права • Ліцензійні умови • Правовласник 	<p>Сумісність з іншими системами</p> <p>Управління цифровими об'єктами</p>
Мова розмітки	<ul style="list-style-type: none"> • Абзац • Заголовок • Список • Ім'я • Дата 	<p>Навігація</p> <p>Сумісність з іншими системами</p>

1.3 Огляд стандартів метаданих

У сучасному світі стандарти існують майже для будь-якої галузі, зокрема для зберігання інформації, обробки даних тощо. Метадані, як вияснили раніше, це також є інформацією, яка описує дані об'єкта, притаманні всім об'єктам інформаційної діяльності, як-то авторське право у книг або pdf-файлів, кількість пікселів у зображенні, версії вашого ПЗ, коли відвідуєте сайти в інтернеті, але всі вони приховані від звичайного користувача. Тому такі елементи метаданих, можуть згруповуватися у набори, які призначені для певної мети, наприклад для певної галузі або певного типу інформаційного ресурсу і такі набори будуть мати назву схеми метаданих [10]. У таких схемах кожен елемент має власну назву та семантику, тобто значення елементу. Деякі схеми можуть вказувати на правила як повинен бути сформульований вміст, інші на правила написання великих або малих літер або взагалі на те, які значення елементів дозволені, спираючись на певний словник дозволених елементів. Тому схеми метаданих, які розробляються та підтримуються організаціями зі стандартизації, наприклад ISO та NISO, або організації, які взяли на себе таку відповідальність (наприклад, Дублінська ініціатива базових метаданих), називаються стандартами метаданих.

Взагалі існує багато схем метаданих, які розробляються для різних дисциплін, таких як освіта, бібліотечна справа, архівна справа, мистецтво та багато інших. У Таблиці 1.3 наведено частину наявних стандартів метаданих, які існують [11].

Таблиця 1.3.

Стандарти метаданих

Назва стандарту	Дисципліна, в якій використовується	Опис
DDI	Архівознавство та суспільствознавство	Ініціатива документації даних — міжнародний стандарт для опису опитувань, анкет, файлів статистичних даних та інформації на рівні дослідження соціальних наук. За стандартом ця інформація описується як метадані.
EAD	Архівація даних	Закодований архівний опис — це стандарт для кодування допоміжних засобів пошуку архівів за допомогою XML в архівних сховищах.
CDWA	Мистецтво	Категорії для опису творів мистецтва — концептуальна основа для опису та доступу до інформації про твори мистецтва, архітектури та іншої матеріальної культури.
VRA Core	Мистецтво	Забезпечує категоріальну організацію для опису творів мистецтва, а також зображень, які їх документують.
ONIX	Книжкова справа	Стандарт представлення та передачі метаданих про продукти книжкової промисловості в електронній формі.

Назва стандарту	Дисципліна, в якій використовується	Опис
Darwin Core	Біологія	Метадані, що описують інформацію про географічне поширення видів та існування зразків у вибірках.
CWM	Зберігання даних	Забезпечення легкого обміну метаданими сховищ і бізнес-аналітики в розподілених гетерогенних середовищах.
EML	Екологія	Документування типового набору даних в екологічних науках для опису цифрових ресурсів або паперових карток та інших нецифрових носіїв.
CSDGM	Географічні дані	Опис вмісту для цифрових геопросторових метаданих
ISO 19115	Географічні дані	Опис географічної інформації та пов'язані з нею дані.
GILS	Уряд/організації	Стандарт метаданих для уряду, компаній чи організацій, щоб користувачі могли швидко та зручно знайти потрібно їм інформацію.
NISO MIX	Зображення	Стандарт для XML-схеми для набору елементів технічних даних, необхідних для керування колекціями цифрових зображень.
MPEG-7	Мультимедіа	Опис мультимедійного вмісту для забезпечення швидкого та ефективного пошуку матеріалу, який цікавить користувача.

Назва стандарту	Дисципліна, в якій використовується	Опис
Dublin Core	Мережеві ресурси	Набір основних елементів метаданих для опису цифрових або фізичних ресурсів.
MARC	Бібліотечна справа	Стандарт для цифрових форматів машинозчитуваного опису елементів, каталогізованих бібліотеками, таких як книги, DVD-диски та цифрові ресурси.
METS	Бібліотечна справа	кодування описових, адміністративних і структурних метаданих об'єктів у цифровій бібліотеці. Має вигляд XML-схеми.
PBCore	ЗМІ	Стандарт для аудіовізуальних активів, зокрема цифрових бібліотек, архівів, які можуть використовуватися громадськими діячами та пов'язаними з ними спільнотами.
MEI	Запис нотного листка	Створення загальноприйнятого цифрового представлення документів нотного запису у структурі для машинного читання.
RDF	Веб-ресурси	Стандарт використовується для опису та обміну графів даних.
DOI	Мережеві ресурси	Стандарт для ідентифікації академічної, професійної та урядової інформації, як-то журнальні статті, звіти про дослідження, набори даних та офіційні публікації, комерційні відео.

Назва стандарту	Дисципліна, в якій використовується	Опис
ISO/IEC 19506	Програмні системи	Стандарт для опису об'єктів програмного забезпечення та їхніх зв'язків, дає можливість визначати архітектуру ПЗ, схеми баз даних, бізнес-правила або терміни та похідні бізнес-процеси. Використовується в основному для забезпечення безпеки, відповідності та модернізації застарілих і існуючих систем.
ISO 23081	Управління документацією	Стандарт, що складається з трьох частин і визначає метадані, необхідні для управління записами. У частині 1 розглядаються принципи управління записами, у частині 2 - концептуальні питання та питання впровадження, 3 частина методи оцінки записів.

1.4 Стандарти метаданих для мультимедійних файлів

Різні дисципліни використовують різні стандарти метаданих, мультимедійні файли також не є винятком. Кожен мультимедійний файл – будь то аудіо, відео, анімація або зображення – містить багато різних типів даних. На відміну від простого текстового файлу, який може містити тільки текстові символи, мультимедійні файли мають багато різних аспектів і вимірів, включаючи візуальний та аудіо-контент, субтитри, анотації та інше. Метадані для того і створені щоб описувати таку інформацію, як авторство, дату створення, геолокацію, короткий опис вмісту, а також технічні характеристики, наприклад, формат файлу, розмір, тривалість відтворення і

багато іншого. Важливість впорядкування цих даних не можна недооцінювати. Метадані допомагають утворювати структуру, яка дозволяє легко знаходити, організовувати, відтворювати та зберігати мультимедійні файли.

Тож значення метаданих і роль, яку вони відіграють у мультимедіа, є дуже великою. Зрозумівши це можна перейти до опису конкретних форматів метаданих, які часто використовуються в мультимедійних файлах.

Легко ілюструвати, які саме стандарти використовуються для опису метаданих мультимедійного файлу, можна за допомогою японської коробки для їжі – бенто (рисунок 1.1). Оскільки лотки в коробці для бенто містять кілька типів їжі, віртуальні відділення у файлах зображень можуть містити кілька типів даних – від пікселів, з яких складається зображення, до тексту, який ідентифікує та описує зображення.

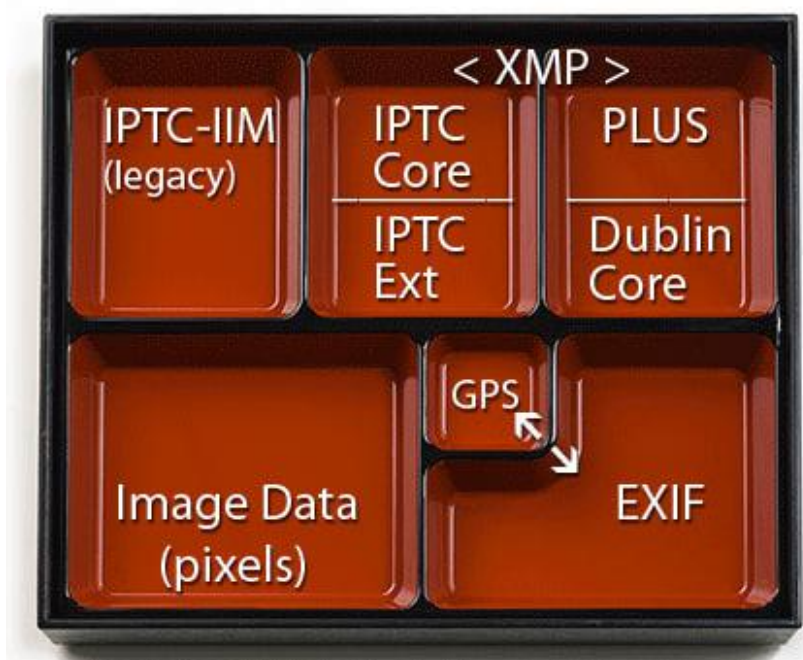


Рисунок 1.1 – Стандарти метаданих мультимедійних файлів

Не всі мультимедійні файли мають ці відділення. Але, наприклад, JPEG, TIFF, PSD, Raw і кілька інших форматів файлів можуть містити будь-які або всі стандартні типи метаданих.

1.4.1 Стандарт метаданих IPTC-ПМ

IPTC-ПМ це схема метаданих, розроблена на початку 1990-х років. Спочатку вона була створена з метою допомоги новинним агенціям у формуванні описів до перших цифрових зображень [12]. Основною перевагою цієї схеми є те, що вона широко сумісна з більшістю програм для редагування та управління зображеннями.

Незважаючи на те, що сьогодні IPTC-ПМ часто вважається "застарілою", вона продовжує бути широко використовувана і визнана. Це пов'язано з тим, що більшість програм, які працюють з метаданими, здатні читати і записувати дані за цим стандартом. Багато з цих програм не підтримують новішу схему IPTC Core/XMP.

Основою на форматі для текстових файлів, IPTC-ПМ надає спосіб обробки цифрових ресурсів з метаданими і вмістом, що зберігаються у структурованій двійковій формі. У середині 1990-х років, компанія Adobe додала можливість вставки описових метаданих в файли цифрових зображень TIFF і JPEG, що призвело до появи так званих заголовків IPTC.

Хоча ПМ був розроблений для опису всіх типів медіа (статті, зображення і т.д.), спільнота фотографів визнала цю схему особливо корисною. Однак, після випуску IPTC Core у 2005 році, вона вже не була єдиною можливістю для вбудовування метаданих у фотографії. Більшість новіших програм почали використовувати XMP для зберігання метаданих, зберігаючи при цьому зворотну сумісність із ПМ шляхом синхронізації даних.

Хоча багато програм здатні читати і записувати заголовки IPTC, вони викликають певні складнощі. Метадані часто вважаються "чорною магією", оскільки місце зберігання та структура метаданих змінюються в залежності від формату файлу зображення. Крім того, іноді виникають проблеми з відображенням діакритичних символів (наприклад, наголоси, тильди і т.д.) при обміні файлами між різними операційними системами. Схема також має обмеження щодо кількості символів, які може містити кожне поле.

1.4.2 Стандарт метаданих PLUS

Універсальна система ліцензування зображень (PLUS) це схема та набір інструментів, які дозволяють створювати рядок символів для ідентифікації власника авторського права, користувача, а також для визначення обсягу та умов використання ліцензованих зображень [13].

PLUS є універсальним стандартом метаданих, що дозволяє описувати ліцензії та права, надані для фотографій. Цей стандарт розроблено недавно і він надає інтегрований набір стандартів для передачі інформації про права та власність, що стосується як замовлених, так і існуючих зображень. PLUS розроблена міжнародною некомерційною коаліцією, яка також затверджує та підтримує набір стандартів для ліцензування. Вона є головною асоціацією, що представляє інтереси видавців, дизайнерів, рекламних агенцій, фотографів, ілюстраторів, розповсюджувачів зображень, представників художників, музеїв, бібліотек та органів стандартизації, таких як UPDIG, IPTC, IDEAlliance і багато інших.

Метадані PLUS допомагають з вирішенням проблеми з творами без власника, наприклад зображення, власник прав на яке не може бути встановлений. З усіма змінами в законодавстві про авторські права, що трапляються по всьому світу, можливість легко ідентифікувати власника прав на зображення та зв'язатися з ним становить велику важливість. Якщо законопроекти, запропоновані в США, будуть прийняті, вони дозволять використання будь-якого зображення, на яке не можна знайти власника авторського права, майже для будь-яких цілей, включаючи комерційне використання. Використання метаданих IPTC, IPTC Core або PLUS для вбудовування повної і точної інформації про права може допомогти захистити зображення від такого неліцензійного використання. Відповідні законодавчі ініціативи розглядаються в декількох країнах, а в Канаді вже існує рада з авторських прав, яка приймає рішення щодо творів, створених "невизначеними" власниками авторських прав.

1.4.3 Стандарт метаданих XMP

XMP представляє собою новий формат даних, який використовує IPTC Core and Extension для зберігання та обробки метаданих зображень. Це забезпечує можливість зберігання метаданих безпосередньо в файлі зображення або в окремому супутньому файлі додаткової інформації. Він також дозволяє користувачам створювати свої власні поля метаданих.

XMP є надійним, гнучким і універсальним інструментом для зберігання метаданих мультимедійних файлів. Це формат розширеної мови розмітки, який призначений для зберігання метаданих у цифрових фотографіях [14]. Adobe вперше представила цей формат у 2001 році з випуском Photoshop 7. У 2004 році, спільно з IPTC та IDEAlliance, Adobe представила основну схему IPTC для XMP, що переносить метадані з заголовків IPTC в сучасний і гнучкий формат XMP.

Особливою перевагою XMP є можливість створення власних полів метаданих. Ці поля не просто зберігають додаткові дані, але й організують їх інакше.

Проте, необхідно зазначити, що поки що лише Adobe Photoshop та Bridge, а також деякі інші бази даних зображень можуть імпортувати або відображати ці спеціалізовані метадані. А спеціальні панелі вимагають додаткової конфігурації, перед тим як їх зможуть використовувати інші користувачі.

XMP використовує комбінацію XML та RDF. XML дуже подібний до HTML. Однак, на відміну від HTML, який зосереджується на відображенні даних, XML має справу з "представленням" даних. XML є незалежним від операційної системи, відкритим, легким для інтерпретації, текстовим і вартісно-ефективним. RDF – це стандарт The World Wide Web Consortium (W3C), що призначений для інтеграції різноманітних додатків з використанням XML як мови обміну даними. Він може бути використаний в широкому спектрі областей, включаючи бібліотечні каталоги, всесвітні довідники, новини, програмне забезпечення, а також колекції музики, зображень та подій.

Використання XMP дозволяє не тільки ефективно управляти метаданими зображень, але і сприяє більшій інтеграції та співпраці між різними програмами та платформами.

1.4.4 Стандарт метаданих IPTC Core & Extension

Цей новітній план ґрунтується на спадкоємності ПМ, вводячи додаткові типи дескриптивної та адміністративної інформації, що включають нові області для задоволення потреб співтовариств у сфері стокової фотографії та культурної спадщини, упакованих у більш стабільний формат даних – "XMP", бо однією перевагою схеми "XMP" є її розширюваність, що дає можливість додавати нові простори імен для підтримки нових видів метаданих [15].

У 2005 році, Міжнародна рада з проблематики преси та телекомунікацій презентувала оновлений стандарт для використання даних IPTC в схемі Adobe XMP, що було названо "IPTC Core". Це дозволяє інтегрувати дані IPTC (через XMP) у більш широкий вибір форматів зображень, таких як JPEG, TIFF, JPEG2000, DNG та інші.

Враховуючи підтримку Unicode в XMP, він може відображати нелатинські алфавіти (наприклад, кирилицю, японську, китайську). Це також розв'язує проблему спотворення діакритичних символів під час передачі зображень між операційними системами Macintosh та Windows. У відмінності від застарілого формату, IPTC Core не встановлює конкретних обмежень на символи для кожного поля, окрім підтримки зворотної сумісності з оригінальною схемою IPTC.

Цей формат, спершу відомий як IPTC4XMP на честь його робочої групи, зберігає інформацію відокремлено від форми ПМ метаданих IPTC. Але незалежно від того, чи використовується IPTC Core або оригінальна схема IPTC, вони обидві дають змогу забезпечити цінну інформацію про зображення, що може допомогти в управлінні активами, пошуку та використанні контенту. Важливо, що XMP та IPTC Core підтримують стандартизовані метадані, які забезпечують універсальність та зворотну сумісність.

1.4.5 Стандарт метаданих EXIF

EXIF – це стандартизований набір метаданих, який часто генерується камерами та іншими пристроями для обробки зображень. Він охоплює технічну інформацію про зображення та метод, використаний для його створення [16]. Ці цінні дані включають такі деталі, як налаштування експозиції, час зйомки, GPS-координати, що вказують на місце розташування, і навіть специфіку моделі камери. На рисунку 1.2 зображено як саме виглядає структура EXIF метаданих на прикладі власного фото.

EXIF	
Y Resolution	72
X Resolution	72
Camera Model Name	
Make	OnePlus
Y Cb Cr Positioning	Centered
Exif Version	0220
Aperture Value	1.6
Scene Type	Directly photographed
Exposure Compensation	0
Exposure Program	Program AE
Color Space	sRGB
Max Aperture Value	1.6
Exif Image Height	3000
Brightness Value	-4.98
Date/Time Original	2023:04:29 17:30:19
Flashpix Version	0100
Sub Sec Time Original	844490
White Balance	Auto
Interoperability Index	R98 - DCF basic file (sRGB)
Exposure Mode	Auto
Exposure Time	1/11
Flash	Off, Did not fire
Sub Sec Time	844490
F Number	1.6

Рисунок 1.2 – Структура EXIF метаданих

Формат файлу, який використовується для зберігання цифрових зображень, містить важливі технічні метадані, пов'язані зі зйомкою та характеристиками самого зображення. Коли цифрові камери створюють файли зображень, вони вбудовують

технічні метадані, які називаються Exif-даними, що переважно зустрічаються у форматах JPEG і TIFF.

Однією з важливих особливостей Exif є його здатність записувати інформацію про камеру безпосередньо у файл зображення в момент зйомки. Ця інформація може охоплювати різні загальні поля даних, включаючи марку і модель камери, її серійний номер, дату і час зйомки, конкретні значення витримки і діафрагми, використаний об'єктив, а також налаштування чутливості ISO. Крім того, метадані Exif часто містять інші важливі технічні деталі, такі як налаштування балансу білого та відстань між камерою та об'єктом зйомки.

Таким чином, Exif слугує цінним сховищем технічних метаданих у файлах зображень, дозволяючи фотографам та ентузіастам отримати доступ до великої кількості інформації про налаштування камери та умови, за яких було знято зображення.

1.4.6 Стандарт метаданих Dublin Core

Dublin Core – це міжнародний стандарт метаданих, офіційно відомий як Dublin Core Metadata Element Set і включає 15 термінів метаданих (даних, що описують дані). Ці терміни надають розширену інформацію для каталогізації та покращують індексування документів пошуковими системам [17].

Dublin Core складається з 15 термінів метаданих, які забезпечують потужний каталог для опису веб-ресурсів, покращують пошукову оптимізацію (SEO) та об'єднують метадані з різних стандартів. Існує дві форми Dublin Core:

- Просте Dublin Core. Вона виражає елементи як пари "атрибут-значення", використовуючи лише 15 елементів метаданих з набору елементів метаданих Дублінського ядра.
- Кваліфіковане Dublin Core. Додає інформацію про схеми кодування, перелічені списки значень або інші підказки для обробки, щоб підвищити специфічність метаданих.

Dublin Core, як було сказано вище, має 15 термінів. Ці терміни формують схему метаданих, яку можна використовувати для опису як фізичних, так і цифрових ресурсів. Використовуються наступні терміни:

- Назва (ім'я, яке присвоєно ресурсу)
- Автор (особа або організація, відповідальна за контент)
- Тема
- Опис (текстовий виклад змісту)
- Видавець (відповідальні за доступність ресурсу)
- Той, хто додає контент до ресурсу
- Дата (коли ресурс став доступним)
- Тип (категорія для контенту)
- Формат (як представлений ресурс – книга, зображення і т.д.)
- Ідентифікатор (ідентифікатор контенту, наприклад, URL-адреса)
- Джерело (звідки було отримано ресурс)
- Мова (якою мовою написаний контент)
- Зв'язок між ресурсами (як контент пов'язаний з іншими ресурсами, наприклад, якщо це глава в книзі)
- Розміщення (де фізично розміщений ресурс)
- Права (посилання на підтвердження авторських прав)

Висновки до розділу 1

У цьому розділі було розглянуто законодавча база України та світу, яка може контролювати розповсюдження, використання, зберігання та збирання метаданих. Закони які це регулюють: Закон України «Про інформацію», Закон України «Про захист персональних даних» та регламент в межах законодавства Європейського Союзу щодо захисту персональних даних GDPR.

Було розглянуто класифікація метаданих, за якою вони поділяються на:

- Описові метадані, тобто ті метадані, які включають інформацію, як-то контактні дані, назву або автора публікації, резюме роботи, ключові слова, географічне положення або пояснення методології.

- Структурні метадані, тобто ті, які описують, як ряд об'єктів об'єднуються.

- Адміністративні метадані. Вони можуть включати дати створення або надходження, права доступу, права або походження, або правила використання, такі як зберігання або видалення.

Також, у цьому розділі було досліджено, що елементи метаданих, такі як кількість пікселів у зображенні, версії ПЗ та інші, можуть згруповуватися у схеми метаданих. Якщо такі схеми створюються та розробляються організаціями зі стандартизації, то вони вже починають називатися стандартами метаданих.

Основними стандартами для мультимедійних файлів є:

- IPTC-IIM
- PLUS
- XMP
- IPTC Core & Extension
- EXIF
- Dublin Core

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ЗАГРОЗ КОНФІДЕНЦІЙНОСТІ ТА МЕТОДІВ ОБРОБКИ МЕТАДАНИХ

2.1 Типові загрози конфіденційності, пов'язані з метаданими

З розвитком інформаційних технологій, коли мультимедійні файли стають одним з основних засобів передачі інформації, проблеми захисту конфіденційності стають все більш актуальними. В основі цих проблем знаходяться метадані – важлива частина мультимедійних файлів. Втрата контролю над метаданими може привести до витоку конфіденційної інформації і, в результаті, до загрози особистій безпеці та конфіденційності. Цим можуть скористатися зловмисники для добування персональної інформації про людину, яка пересилаючи власні фото або відео навіть і не замислюється про те, що разом з цими даними вона відправляє дані про місцезнаходження під час знімку, або модель пристрою на який було зроблено фото або відео, яка може дати можливість зловмиснику реалізувати шкідливе ПЗ для подальшого стеження або вимагання грошей.

Проблема з метаданими полягає в тому, що вони часто створюються і зберігаються без відома користувача. Саме це і призводить до ряду потенційних загроз конфіденційності, таких як:

- Розкриття геолокації та особистої інформації
- Викриття технічних характеристик [18]

2.1.1 Розкриття геолокації та особистої інформації

Метадані, які містяться в мультимедійних файлах, мають потенціал стати значною загрозою конфіденційності користувачів. Вони можуть включати широкий спектр інформації, яка, в свою чергу, може бути використана для небажаного доступу до особистої інформації користувача або її зловмисного використання.

Прикладом того, як метадані можуть стати джерелом витоків інформації, є обробка фотографій або інших мультимедійних файлів, як-то відео або аудіо. Багато сучасних камер (включаючи ті, що вбудовані в смартфони) автоматично додають до фотографій EXIF-дані, що містять різноманітну інформацію про умови зйомки (час зйомки, параметри експозиції тощо), а також, потенційно, інформацію про місцезнаходження зйомки в форматі геотегів. Якщо такі фотографії оприлюднюються в Інтернеті без відповідної обробки, ці дані стають доступними для всіх, хто має доступ до фотографії.

Тому розкриття даних про місцезнаходження в метаданих мультимедійних файлів є однією з найбільш критичних загроз конфіденційності. Геолокаційні метадані, як правило, є даними, які вказують на фізичне місце, де був створений чи модифікований файл. Вони можуть містити конкретні географічні координати (широту і довготу), а також інформацію про висоту, швидкість, напрямок та інші геопросторові дані.

Метадані геолокації можуть надавати достатньо інформації для виявлення шаблонів поведінки користувача. Зокрема, аналіз місцезнаходження, зафіксованого на фотографіях, може вказувати на місця, де користувач найчастіше перебуває, такі як дім, робота, школа, спортзал, ресторани та інші місця відпочинку [19]. Знання про ці шаблони поведінки може бути використано зловмисниками для різних цілей, включаючи, але не обмежуючись, шахрайство, крадіжку, або навіть фізичні загрози. Розкриття геолокації може стати особливо небезпечним для осіб з публічним статусом, таких як політики, знаменитості або журналісти, які можуть стати ціллю для незаконних дій.

Також отримання інформації про місцезнаходження під час фото або відео-зйомки може використовуватися ворогом у воєнних конфліктах. Яскравим прикладом є повномасштабне вторгнення в Україну яке розпочала російська федерація у лютому 2022 року. Агресор, в обличчі росії, влаштовуючи масовані обстріли, постійно веде пошук фото або відеоматеріалів у соціальних мережах для корегування своїх цілей. Це відбувається тому що багато людей не обізнані в тому, що, викладаючи мультимедійний файл у мережу інтернет або у соціальну мережу, викладають також

і інформацію про місцезнаходження техніки або інших важливих військових об'єктів [20].

Метадані місцезнаходження можуть вказувати на специфічні події, в яких брав участь користувач. Зловмисники можуть використовувати цю інформацію для встановлення зв'язків між людьми, що може призвести до розкриття інформації про особисті відносини або професійні стосунки. Для прикладу можна взяти будь-який захід, на якому зібралося багато людей, людина сфотографувала щось та виклала у мережу, зловмисник знайшов за допомогою метаданих місцезнаходження почав шукати інші фото та відео з цього заходу, потім за допомогою іншого ПЗ він може знайти імена людей, що відвідували цю подію та за допомогою цієї інформації багато інших персональних даних. Така інформація може бути використана для шантажу або гірше – фізичного насилля.

Взагалі, розкриття місцезнаходження призводить до наступних загроз конфіденційності користувачів:

- Ідентифікація особи. Ненавмисне розкриття геолокаційних метаданих може призвести до виявлення особистості користувача. Ця проблема стає особливо актуальною, коли користувач бажає залишитися в анонімності.
- Розкриття місця проживання. Постійне публікування контенту з конкретної локації може непрямо вказувати на домашню адресу користувача. Це може становити проблему, якщо користувач не прагне розголошувати цю інформацію.
- Визначення життєвих звичок. Регулярне публікування контенту з певних місць може дозволити іншим зрозуміти повторювані моделі поведінки користувача. Наприклад, якщо користувач часто ділиться контентом з конкретної кав'ярні, це може свідчити про його звичай там відвідувати.
- Використання даних третіми особами. Геолокаційні дані можуть бути використані третіми сторонами для різноманітних цілей, включаючи маркетинг, слідкування, які можуть порушувати приватність людини.

Також особливо важливими є метадані, які можуть містити особисту інформацію, таку як ім'я користувача або контактні дані. Всі ці види інформації можуть бути використані зловмисниками в різних контекстах.

Ім'я користувача, закодоване в метаданих, може стати критичною вихідною точкою для проведення цільових атак. Поняття цільової атаки вказує на специфічну взаємодію зловмисника з жертвою, при якій атакується конкретна особа або організація. У контексті соціальної інженерії зловмисники можуть використовувати ім'я, знайдене в метаданих, як початковий пункт для збирання додаткової інформації. Це може бути пошук в соціальних мережах, де більшість сучасних людей виставляють свої персональні дані, що і допомагає виявляти більше особистої інформації, такої як дата народження, адреса, робота, хобі та інше. Також можливий пошук даних про особу на форумах або інших платформах, на яких вона може брати участь. Ця інформація потім може бути використана для підготовки фішингових атак, які виглядають більш достовірно, тому що містять конкретні деталі, знайомі жертві.

У разі наявності контактної інформації в метаданих, такої як електронна адреса або номер телефону, зловмисники можуть ініціювати безпосередній контакт з жертвою. Це може призвести до різних форм шахрайства, такі як, наприклад, підроблені повідомлення від "банку" або "постачальника послуг", які вимагають від жертви поділитися додатковими даними або ввести свої дані на підробленому веб-сайті. Крім того, зловмисники можуть намагатися маніпулювати жертвою, створюючи враження термінової потреби або критичної ситуації, в якій жертва має відгукнутися швидко, без належної обачності.

В решті решт, зловмисники можуть використовувати особисту інформацію для цілком недемократичних форм впливу, таких як погрози або шантаж. Наприклад, зловмисник може стверджувати, що він має компрометуючу інформацію про жертву або що він може нашкодити жертві або її близьким, якщо жертва не виконає певні вимоги.

2.1.2 Викриття технічних характеристик

Іншою значимою проблемою конфіденційності через несанкціонований доступ до метаданих мультимедійних файлів є викриття технічних характеристик пристрою на якому було зроблено або редаговано цей файл. Ця інформація може бути моделлю

камери, версією ПЗ, деякі налаштування системи або унікальні ідентифікатори обладнання та інші технічні параметри. Використовуючи ці дані зловмисники можуть проводити атаки, спрямовані на специфічне ПЗ або обладнання.

Розкриття технічних характеристик може збільшити ризики людини на двох рівнях. По-перше, це допомога зловмисникам у визначенні потенційних вразливостей, які є відомими для певних моделей обладнання або версій програмного забезпечення і подальше його впровадження до системи своєї жертви. Знання про це дає можливість досить легко визначити що і як реалізовувати задля отримання бажаного результату. Наприклад застаріле обладнання має більше різних вразливостей та згідно статистики OWASP займає 6 місце у OWASP Top 10 [21].

Для пошуку таких технічних даних зловмисник може використовувати методи OSINT задля знаходження компрометуючої інформації, зокрема і мультимедійних файлів. Знайшовши файли, які належать жертві він починає аналізувати метадані, які в них зберігаються та виявляє якою операційною системою вона користується, моделі пристроїв які є у жертви та інші технічні дані [22]. Далі він реалізовує або шкідливе ПЗ для стеження, щоб зібрати більше інформації про жертву, або програми вимагачі задля вимагання грошових коштів від жертви. І після цього зловмисник переходить на інший рівень – знайдена ним інформації про технічні характеристики може бути використана для створення більш переконливих спроб шахрайства або соціальної інженерії. Тобто, зловмисник, наприклад, може використовувати відомості про модель камери користувача для створення електронного листа, який виглядає як офіційне повідомлення від виробника цієї камери, в якому користувача просять надати особисту інформацію або завантажити "оновлення" програмного забезпечення, яке насправді є шкідливим ПЗ, яке він розробив.

2.2 Огляд сучасних методів та технік обробки метаданих для забезпечення конфіденційності

В рамках сучасних технологій інформаційної безпеки, обробка метаданих є критичним елементом для захисту конфіденційності, особливо в контексті

мультимедійних файлів. Одним з основних напрямків в обробці метаданих є їх анонімізація [23] та псевдонімізація [24]. Анонімізація відноситься до процесу видалення або заміни ідентифікуючої інформації в метаданих, тоді як псевдонімізація включає заміну ідентифікуючої інформації на синтетичну, яка не може бути безпосередньо пов'язана з реальною особою.

Анонімізація метаданих часто використовується для зменшення ризику втрати конфіденційності. Це включає видалення таких даних, як ім'я користувача, контактна інформація та інша інформація, яка може вказувати на особу. Проте анонімізація не є повністю бездоганною. За деяких обставин, зловмисники можуть провести відновлення анонімізованої інформації через методи де-анонімізації. Псевдонімізація метаданих – це ще один метод, який часто використовується для захисту конфіденційності. Він включає в себе заміну особистої інформації на псевдоніми, які не містять прямих посилань на реальну особу. Цей метод дозволяє захистити особисту інформацію.

Крім анонімізації та псевдонімізації, існує ще ряд технологій для захисту метаданих, включаючи шифрування та управління доступом. Шифрування допомагає захистити метадані від несанкціонованого доступу та витоку. Управління доступом, у свою чергу, дозволяє контролювати, хто має право переглядати та модифікувати метадані.

Також машинне навчання та штучний інтелект можуть бути ефективними інструментами для автоматичної обробки метаданих в мультимедійних файлах. Зокрема, ці технології можуть бути використані для автоматичного виявлення та видалення чутливої інформації, що міститься в метаданих.

У контексті мультимедійних файлів, захист метаданих вимагає особливого підходу. Оскільки метадані можуть містити велику кількість інформації, яку не бажано знайти зловмиснику, її захист від витоку та зловмисного використання є важливим завданням. Для ефективного захисту метаданих в мультимедійних файлах можна використовувати комбінацію різних методів і технологій. Це може бути використання різних технік анонімізації та псевдонімізації разом зі штучним

інтелектом, шифруванням метаданих або впровадження жорсткого контролю доступу до метаданих.

Щоб зрозуміти як саме працюють ці методи обробки метаданих для забезпечення конфіденційності краще всього буде пояснити на прикладах.

Для пояснення методів анонімізації та псевдонімізації візьмемо фотографію, яка була зроблена на смартфон. Вона містить метадані EXIF, що включають точну геолокацію, де фото було зроблено. Виконати процес анонімізації в цьому контексті б означало видалення цих даних геолокації з метаданих за допомогою завантаженого програмного забезпечення або використання спеціальних веб-ресурсів, які вміють видаляти метадані у фото. Псевдонімізація, у свою чергу, може замінити точні координати на більш загальне місцезнаходження, наприклад, місто або країну. При використанні псевдонімізації не бажано вказувати схожі з достовірними дані, у випадку зміни інформації про місцезнаходження під час зйомки буде кращим не вказувати сусідню вулицю або район.

Переваги та недоліки анонімізації як методу обробки даних для забезпечення їх конфіденційності наведено нижче.

Переваги:

- Анонімізація запобігає витік особистої інформації, оскільки вона видаляє всі ідентифікаційні дані.
- Видалені дані не можуть бути відновлені.

Недоліки:

- Інколи процес анонімізації видаляє не всі метадані і тому всі дані, що залишилися можуть бути скомбіновані для виявлення особистості користувача.

Переваги та недоліки псевдонімізації наведено нижче.

Переваги:

- Баланс між конфіденційністю та корисністю даних
- Порівняно з анонімізацією, псевдонімізація зберігає контекст. Як приклад, у псевдонімізації географічні дані замінюються на більш загальні, що і ускладнює отримання персональної інформації зловмисником.

Недоліки:

- При псевдонімізації є можливість, що при достатньому аналізі можливо відновити оригінальні дані.

Шифрування також можна вважати ефективним засобом захисту будь-якої інформації, в тому числі метаданих. Наприклад, візьмемо відеофайл, метадані якого містять дані про ім'я автора, програмне забезпечення, що було використано для створення цього відео, а також час і дату його створення. Шифрування цих метаданих буде означати, що лише особи з відповідним ключем дешифрування можуть доступитись до цієї інформації. Тобто шифрування можна використовувати разом з методами псевдонімізації, що дозволить максимально збільшити безпеку людини, яка зробила це відео та завантажила його до мережі інтернет.

Переваги та недоліки шифрування метаданих наведено нижче.

Переваги:

- Шифрування є одним з найдієвіших способів захисту даних, тим самим забезпечує та майже унеможлиблює розшифрування зашифрованих полів метаданих.
- Отримати зашифровані метадані зможе тільки та людина, яка має ключ для розшифрування.

Недоліки:

- Для звичайних користувачів шифрування може здатися складним для використання.
- У випадку втрати ключа дешифрування, поля метаданих, які були зашифровані, стають недоступні для всіх, навіть автора файлу.

Стосовно управління доступом можна розглянути сценарій, коли ми маємо власно створений pdf- або doc-файл, який містить, автоматично згенеровані програмою редактором, метадані про автора, дату створення та які зміни могли відбуватися з ним. Використання методів та технік управління доступом до метаданих може встановлювати обмеження на те, хто може переглядати або змінювати ці метадані. Наприклад, створений вами файл може бути налаштований таким чином, що лише автор документа може переглядати або змінювати метадані або взагалі прибрати можливість перевірки файлу на вміст метаданих.

Переваги та недоліки при управлінні доступом до метаданих наведено нижче.

Переваги:

- Тільки уповноважені особи зможуть мати доступ до файлу, тим самим і до чутливої інформації у метаданих.

Недоліки:

- При неправильних налаштуваннях системи для управління доступом, є можливість витоку файлів а з ними і метаданих.

Машинне навчання та штучний інтелект – два популярних напрямки, що мають дуже стрімкий зріст у сучасному світі, які можуть використовуватися майже не в усіх сферах нашого життя, оскільки їх розробка та навчання не є дуже складним процесом. Їх також можна використовувати для обробки метаданих, щоб забезпечити конфіденційність тієї інформації, яку можна дізнатися проаналізувавши метадані. Зокрема, вони можуть бути використані для виявлення та видалення чутливих даних, що містяться в метаданих. Наприклад, за допомогою машинного навчання штучний інтелект може бути навчений таким чином, що буде використовуватися для аналізу метаданих зображень, відео та інших мультимедійних файлів щоб виявляти та видаляти інформацію про особу, які створила або редагувала цей файл.

Штучний інтелект та машинне навчання як і всі інші методи мають свої переваги та недоліки. Деякі з них наведено нижче.

Переваги:

- Має високу ефективність, оскільки можуть аналізувати великі об'єми метаданих швидше, ніж людина, та виявляти поля метаданих з чутливою інформацією.
- Має високу адаптивність до нових умов, такі як зміни в метаданих або шаблонах поведінки користувачів.

Недоліки:

- Розробка, тренування і впровадження штучного інтелекту вимагає значних ресурсів та спеціалістів.
- Тренування штучного інтелекту за допомогою машинного навчання на основі чутливих метаданих може порушувати приватність людей.

Висновки до розділу 2

У цьому розділі було розглянуто основні загрози конфіденційності, які пов'язані з метаданими і сучасні методи та техніки обробки метаданих для забезпечення конфіденційності інформації, яку вони можуть містити.

Розглянутими та найголовнішими проблемами безпеки конфіденційності метаданих є:

- Розкриття геолокації та особистої інформації, тобто розкриття інформації про місцезнаходження під час зйомки фото або відео, а також викриття імені автора файлу.
- Викриття технічних характеристик, тобто визначення версії ПЗ, типу пристрою і т.п., що дає змогу зловмиснику легше визначитися з типом атаки: соціальна інженерія або розповсюдження зловмисного програмного забезпечення, яке цілеспрямоване на жертву.

Сучасними методами та техніками для обробки метаданих можна вважати:

- Анонімізація та псевдонімізація, для видалення або модифікації чутливих даних.
- Шифрування інформації, яка міститься у метаданих
- Управління доступом – введення обмежень на те хто, та що саме може отримати з метаданих.
- Використання машинного навчання та штучного інтелекту для швидкого аналізу, виявлення та видалення чутливої інформації, що міститься у метаданих мультимедійних файлів.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ОБРОБКИ МЕТАДАНИХ ДЛЯ ЗАХИСТУ КОНФІДЕНЦІЙНОСТІ В МУЛЬТИМЕДІЙНИХ ФАЙЛАХ

3.1 Визначення основних вимог до програмного модуля

Головним завданням розробленого програмного модуля є захист приватності користувача шляхом анонімізації метаданих. Оскільки, як раніше вже було розглянуто, метадані можуть містити багато інформації про користувача, включаючи деталі, як-от географічне положення, час створення файлу, тип пристрою, на якому було створено файл, ім'я автора файлу і багато іншого. Саме тому програмний модуль спрямований на мінімізацію таких ризиків, надаючи користувачам інструменти для захисту їхньої приватності.

Функціональні вимоги

- Перегляд метаданих

Програмний модуль повинен надавати можливість користувачам легко і швидко переглядати метадані мультимедійних файлів. Тобто програма повинна вміти аналізувати файл, виявляти всю інформацію, що міститься в метаданих, і відобразити її в зручному форматі. Цей процес включає в себе зчитування даних про автора, дату створення файлу, розмір файлу, тип пристрою, на якому був створений файл, географічне положення, де файл був створений (якщо ця інформація доступна), і так далі.

- Анонімізація

Однак основною метою розробленого програмного модулю повинно бути не просто виявлення метаданих, а надати можливість користувачам анонімізувати цю інформацію для захисту їхньої приватності. Це може бути простим видаленням чутливої інформації з метаданих або видалення всіх метаданих мультимедійного файлу (які можливо видаляти).

Нефункціональні вимоги

- Модуль повинен бути інтуїтивно зрозумілим і легким у використанні для простих користувачів без глибоких технічних знань.
- Програмний застосунок повинен бути сумісний з ОС Windows та ефективно працювати в цьому середовищі.
- Програма повинна забезпечувати швидку обробку метаданих, незалежно від розміру мультимедійних файлів.

Вимоги до типів файлів

Програмний модуль повинен підтримувати різні типи мультимедійних файлів. Це дозволить користувачам використовувати його з більшою кількістю даних і розширить горизонт можливостей його використання.

- Файли зображень

Першим типом файлів, які повинен підтримувати розроблена програма, є файли зображень. Формати зображень, які повинні підтримуватися мають бути наступні: JPEG, PNG, TIFF, HEIC, BMP та інші.

- Відеофайли

Другим типом файлів, які повинен підтримувати програмний застосунок, є відеофайли. Формати відеофайлів, які повинні підтримуватися мають бути наступні: MP4, MOV, WEBM, WMV, FLV, MKV, OGG, AVI та інші.

- PDF файли

Третім типом файлів, які повинен підтримувати програмний модуль, є PDF файли. PDF файли часто використовуються для обміну документами, науковими статтями або авторськими науковими або іншими книгами. Саме тому інформація з метаданих pdf-файлів може дати зловмиснику ім'я автора документа або тип пристрою з якого було створено або редаговано його.

Технічні вимоги

Програмний модуль повинен бути розроблений для роботи на ОС Windows і написаний на мові програмування Python. Це забезпечує те, що модуль буде сумісний з великою кількістю систем і може бути легко інтегрований в існуючі процеси.

3.2 Архітектура програмного модуля

Архітектура ПЗ це комплекс внутрішніх структур, що містять компоненти, їх взаємозв'язки та потенційні взаємодії, а також зовнішні характеристики цих компонентів.

У такому контексті компонент – це гнучкий структурний елемент ПЗ, який можна визначити, встановивши його спосіб взаємодії з оточуючим середовищем. При розробці програмного забезпечення "компонент" часто використовується як найменша одиниця розгортання, що може бути включена або виключена з системи. Такий компонент має визначений інтерфейс і відповідає певному набору правил.

Архітектура ПЗ – це комплекс структур або представлень, які мають різні рівні абстракції та відображають різні аспекти (структуру класів ПЗ, структуру розгортання, тобто відношення компонентів ПЗ до фізичних машин, можливі сценарії взаємодій компонентів), що поєднуються через відповідність всіх представлених даних структурним елементам програмного забезпечення.

Архітектура ПЗ відіграє ключову роль, оскільки вона визначає більшість якісних характеристик програмного забезпечення. Вона також є головним інструментом комунікації між розробниками, а також між розробниками і всіма зацікавленими сторонами.

Програмний застосунок для обробки метаданих мультимедійних файлів містить декілька модулів, які взаємодіють між собою та між компонентами інших модулів.

Основні модулі наведено нижче:

- Видалення метаданих
- Допоміжні функції
- Графічний інтерфейс
- Перелік можливих розширень файлів

Структура взаємодії програмних модулів, які реалізують функції програмного застосунку по видаленню метаданих з мультимедійних файлів представлена на рисунку 3.1.

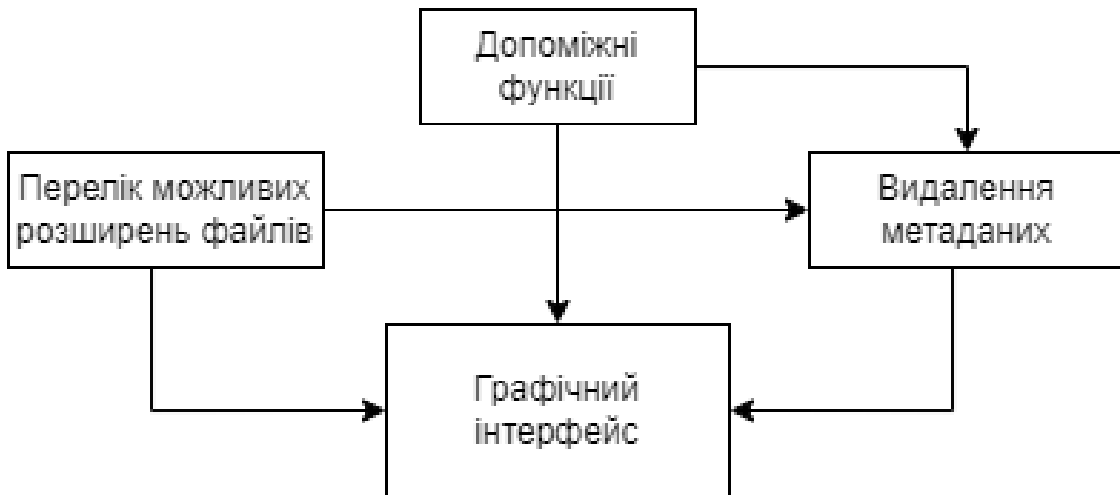


Рисунок 3.1 – Структура взаємодії модулів програми

Оскільки програмний застосунок реалізований на мові програмування Python, то всі модулі можна як групувати за допомогою папок, так і просто зберігати всі файли у одній директорії. Групування у папки більше підходить до проектів, які мають велику кількість функціоналу. Це сприяє покращенню навігації між окремими модулями та частинами програми. У випадку мого програмного застосунку з видалення метаданих мультимедійних файлів, то цей проект є невеликим за розміром, тому що виконує невелику кількість функціональних потреб. Тому всі файли мого проекту лежать у одній директорії, як зображено на рисунку 3.2.

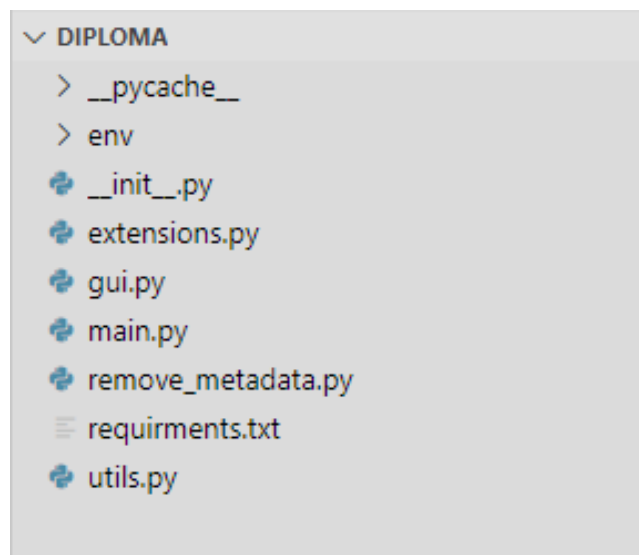


Рисунок 3.2 – Структура програмного застосунку

Пояснення кожного файлу та за що він відповідає:

`__init__.py` – конструктор проекту, який визначає, які залежні модулі будуть використовуватися у додатку.

`extensions.py` – файл, у якому прописані розширення файлів, які будуть розпізнаватися застосунком для перегляду та видалення з них метаданих.

`gui.py` – файл, у якому описується графічний інтерфейс.

`main.py` – головний файл програми з якого вона вся запускається.

`remove_metadata.py` – основний модуль для видалення метаданих.

`utils.py` – файл з допоміжними функціями, наприклад відображення метаданих на екрані і т.п.

3.3 Вибір технологій та інструментів розробки

При розробці будь-якого програмного застосунку потрібно з самого початку виділити якої мовою він буде написаний, які технології можуть використовуватися у ньому та вибрати зручний для себе або редактор коду або середовище розробки. Для виконання поставленого завдання розробки модуля для видалення метаданих у мультимедійних файлах мною було обрано Python, як основну мову програмування, оскільки він простий у вивченні та на ньому легко писати велику кількість програмних додатків різної складності та напрямлення, VS Code, як головне середовище де буде писатися програма.

3.3.1 Python

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання як мови сценаріїв (скриптів) або з'єднувальної мови для з'єднання існуючих компонентів. Простий, легкий для вивчення синтаксис Python підкреслює читабельність і, отже,

знижує вартість обслуговування програми. Python підтримує модулі та пакети, що створює модульність програми та покращує повторне використання одного і того самого блоку коду (наприклад функції додавання чисел) [25].

Python з'явився три десятиліття тому. Його винахідник, голландський програміст Гвідо ван Россум, назвав її на честь свого улюбленого комедійного гурту того часу – "Летючий цирк Монті Пайтона" (Monty Python's Flying Circus). З того часу вона привернула до себе енергійну спільноту ентузіастів, які працюють над виправленням потенційних помилок і розширенням можливостей мови програмування.

Замість того, щоб вбудовувати всю свою функціональність в ядро, Python був розроблений з можливістю розширення за допомогою модулів. Ця компактна модульність зробила її особливо популярною як засіб додавання програмованих інтерфейсів до існуючих додатків. Бачення Ван Россама про невелику основну мову з великою стандартною бібліотекою та інтерпретатором, що легко розширюється, впливало з його розчарування в АВС, який сповідував протилежний підхід.

Python відомий тим, що він потужний, швидкий і робить програмування більш цікавим. Програмісти на Python можуть динамічно вводити змінні без необхідності пояснювати, якого типу має бути змінна. Користувачі можуть завантажити Python безкоштовно і одразу ж почати програмувати. Для цього потрібно лише зайти на офіційний сайт Python, вибрати потрібно версію та завантажити її, оскільки щоб програмувати на цій мові не потрібен сторонній інтерпретатор – мова Python має власний, який встановлюється разом з нею.

Впровадження Python швидко розповсюджується завдяки його зрозумілому синтаксису та читабельності. Часто використовується в аналітиці даних, машинному навчанні (ML) та веб-розробці. Додатки, розроблені за допомогою мови програмування Python, як правило, мають менший розмір, ніж програмне забезпечення, створене за допомогою мов програмування, таких як Java або C++.

Для написання програмного модуля видалення метаданих з мультимедійних файлів були використані наступні модулі (або бібліотеки) мови програмування Python:

- os
- pikepdf
- shutil
- re
- subprocess
- tkinter
- platform

Python.os

Модуль OS у Python є частиною стандартної бібліотеки мови програмування. Модуль дозволяє користувачеві взаємодіяти з рідною ОС, на якій зараз працює Python. Якщо коротко, то він надає користувачеві простий спосіб взаємодії з деякими функціями ОС, які стають в нагоді у повсякденному програмуванні, наприклад створювати або видаляти папки і файли, переходити між директоріями і т.п.

Python.pikepdf

pikepdf – це бібліотека, призначена для розробників, які хочуть створювати, маніпулювати, аналізувати, виправляти та зловживати форматом PDF. Вона підтримує читання і запис PDF-файлів, включаючи створення з нуля. Завдяки QPDF підтримує доступ до зашифрованих PDF-файлів.

Найголовніше що може робити бібліотека pikepdf:

- Копіювання сторінок з одного PDF-файлу в інший
- Поділ та об'єднання PDF-файлів
- Витягування вмісту з PDF-файлів, наприклад зображень
- Замінювати вміст, наприклад, замінювати зображення, не змінюючи решту файлу
- Додавання або зміна метаданих PDF-файлів

Python.shutil

Модуль Shutil у Python надає багато функцій високорівневих операцій над файлами та колекціями файлів. Він входить до складу стандартних модулів Python. Цей модуль допомагає автоматизувати процес копіювання та видалення файлів і каталогів.

Джерелом має бути файл, а місцем призначення може бути як файл, так і каталог. Якщо місцем призначення є каталог, то файл буде скопійовано до нього з використанням базового імені файлу з джерела. Крім того, місце призначення має бути доступним для запису. Якщо місце призначення є файлом і вже існує, то його буде замінено файлом-джерелом, інакше буде створено новий файл.

Python.re

Модуль, що забезпечує операції для створення регулярних виразів.

Регулярний вираз (або RE) визначає набір рядків, який йому відповідає; функції в цьому модулі дозволяють перевірити, чи збігається певний рядок з даним регулярним виразом (або чи збігається даний регулярний вираз з певним рядком, що зводиться до того самого).

Python.subprocess

Subprocess – це інструмент, який дозволяє запускати інші програми або команди з коду на Python. З його допомогою можна відкривати нові програми, передавати їм дані та отримувати результати назад.

Однією з найкорисніших можливостей модуля subprocess є те, що він дозволяє користувачеві керувати входами, виходами і навіть помилками, які генеруються дочірнім процесом, з коду Python.

Ця функція вважається одним з найпотужніших аспектів модуля. Завдяки чому можна зробити процес виклику підпроцесів більш потужним та універсальним. Наприклад, тепер можна використовувати вихідні дані підпроцесу як змінну в іншій частині коду на Python.

Python.tkinter

Tkinter – багатоплатформна графічна бібліотека інтерфейсів на основі засобів Tk (крос-платформенна бібліотека базових елементів графічного інтерфейсу, поширювана з відкритим вихідним кодом). Tkinter поширюється з відкритим вихідним кодом, був написаний Стіном Лумхольтом і Гвідо ван Россумом. Входить у стандартну бібліотеку Python.

Модуль Tkinter не реалізує власний інтерфейс до бібліотеки Tk, а забезпечує конвертування звернень Python в звернення Tcl — мови, яка тісно інтегрована з Tk. Таким чином Tkinter є обгорткою для Tcl/Tk.

Python. platform

Модуль Platform використовується для отримання якомога більшої кількості інформації про платформу, на якій виконується програма. Під інформацією про платформу вважається інформація про пристрій, його ОС, версію ОС, версію Python тощо. Цей модуль відіграє вирішальну роль, коли потрібно перевірити, чи сумісна написана програма з версією python, встановленою на певній системі, або чи відповідають апаратні специфікації вимогам програми.

3.3.2 Visual Studio Code

Для зручного та швидкого редагування коду мною було обрано Visual Studio Code, як основний редактор коду.

Visual Studio Code – це редактор вихідного коду, створений Microsoft із Electron Framework для Windows, Linux і macOS. Функції включають підтримку налагодження, підсвічування синтаксису, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, параметри та встановлювати розширення, які додають функціональність.

VS Code має вбудовану підтримку JavaScript, TypeScript та Node.js, а також багату екосистему розширень для інших мов програмування (таких як C++, C#, Java, Python, PHP та Go), середовищ виконання (таких як .NET та Unity), середовищ (таких як Docker та Kubernetes) та хмарних технологій (таких як Amazon Web Services, Microsoft Azure та Google Cloud Platform).

VS Code має велику кількість переваг порівняно з іншими редакторами коду або середовищ розробки. Деякі з них наведені нижче:

- Простота освоєння. Редактор має інтуїтивно зрозумілий інтерфейс, його легко встановити і налаштувати. А після цього ним відразу можна починати користуватися.

- Мала вага. Порівняно з IDE редактор швидко працює і мало важить, не такий вимогливий до ресурсів комп'ютера. Водночас він досить функціональний, а величезна кількість розширень дає змогу додавати в нього нові можливості.
- Кросплатформеність. Visual Studio Code є для всіх популярних операційних систем, тож з ним можна працювати незалежно від ОС.
- Підтримка різних мов. Є інструменти, які призначені спеціально для якоїсь мови, наприклад Python або C++. Visual Studio Code підтримує величезну кількість мов програмування, і вони легко підключаються. Тому з його допомогою можна вести проекти, написані кількома мовами одночасно.

3.4 Реалізація основних функціональних компонентів

Для реалізації програмного модуля видалення метаданих було написано код програми на мові Python з використанням допоміжних програм – Exiftool та ffmpeg, які використовуються для перегляду та видалення метаданих з фото та відео відповідно.

Для взаємодії додатку з користувачем був розроблений мінімальний графічний інтерфейс задля зручності користування (рисунок 3.3).

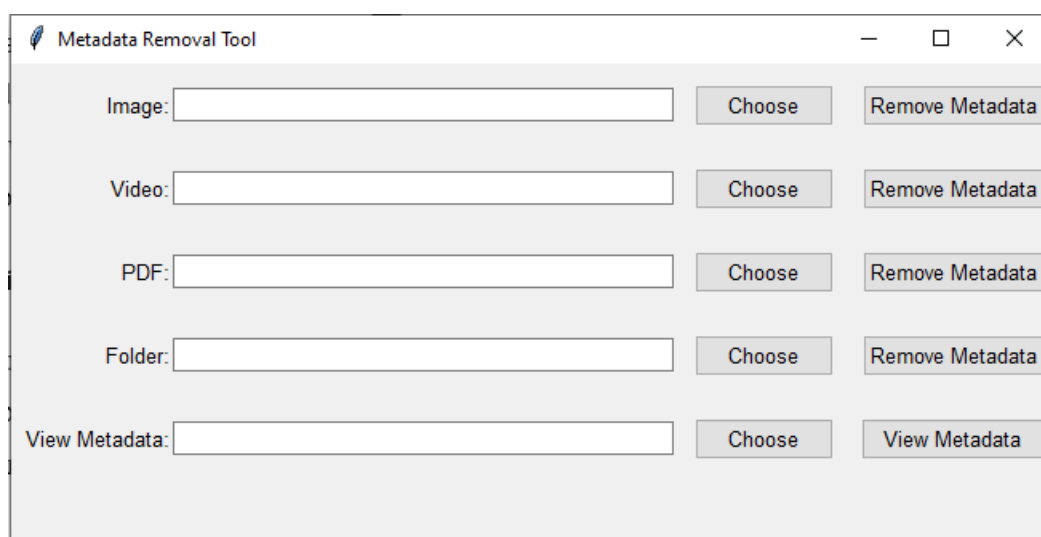


Рисунок 3.3 – Зовнішній вигляд інтерфейсу

Користувач має можливість самостійно ввести шлях до файлу або папки, а також може натиснути кнопку “Choose”, щоб відкрити нове вікно провідника Windows та вже там обрати потрібне фото, відео, pdf-файл або цілу директорію (рисунок 3.4). Натискаючи “Remove Metadata” користувач видаляє метадані вибраного файлу про що, сповіщає напис внизу вікна програми (рисунок 3.5).

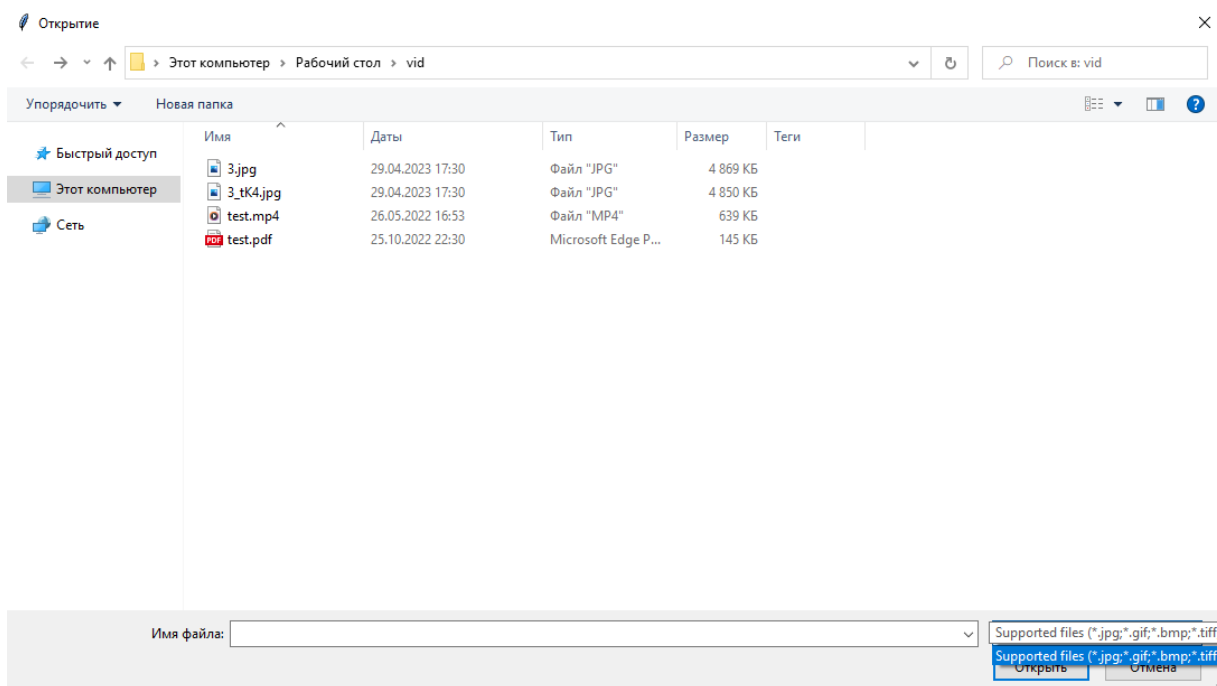


Рисунок 3.4 – Демонстрація можливості вибрати файл у провіднику Windows

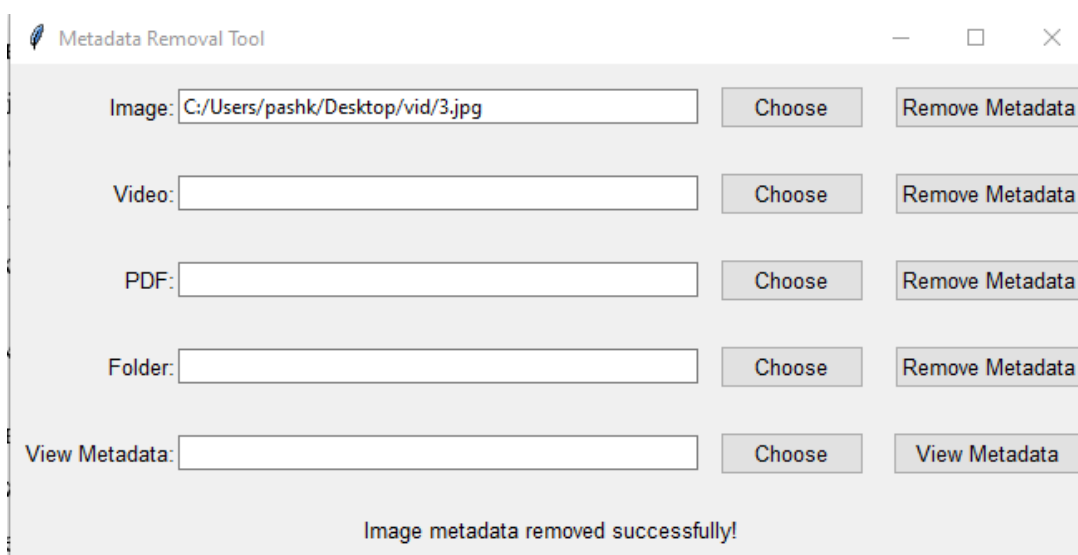


Рисунок 3.5 – Сповідження про успішне видалення метаданих

Інтерфейс додатку дуже простий і зрозумілий, тому перейдемо до більш детального розбору функцій, які відповідальні за видалення метаданих, їх перегляду та інші.

Для видалення метаданих зображень була написана функція, яка приймає назву файлу як вхідний параметр, обробляє це ім'я файлу за допомогою допоміжної функції `file_strip_name()`, щоб створити нову назву копії файлу без метаданих, а потім використовує утиліту `exiftool` та потрібні аргументи для видалення усіх метаданих з оригінального файлу. Код самої функції зображено на рисунку 3.6.

```
def img(x):  
    new_file = file_strip_name(x)  
    os.system(f"exiftool -all= -o {new_file} {x}")  
    return new_file
```

Рисунок 3.6 – Функція з видалення метаданих у фото

Функція видалення метаданих з відеофайлів має схожу реалізацію, але замість використання `exiftool`, використовується `ffmpeg`. Тобто, вона також приймає назву файлу вхідним параметром, створює нове ім'я для файлу без метаданих, і потім за допомогою програми `ffmpeg` видаляє метадані і зберігає “чисте” відео з новою назвою. Програмний код функції зображено на рисунку 3.7

```
def vid(file):  
    new_file = file_strip_name(file)  
    os.system('ffmpeg -i '+file+' -map_metadata -1 -c:v copy -c:a copy '+new_file)  
    return new_file
```

Рисунок 3.7 – Функція видалення метаданих з відео

Для видалення метаданих pdf-файлів була створена функція, зображена на рисунку 3.8, яка спочатку створюється нове ім'я для очищеного файлу, потім pdf-файл відкривається за допомогою `pikempdf.open()`. За допомогою команди `del`, бібліотеки `pikempdf`, видаляються метадані XMP (`pdf.Root.Metadata`) та метадані інформації про документ (`pdf.docinfo`). Після видалення метаданих pdf-файл зберігається під новим ім'ям і закривається.

```
def delete_pdf(x):
    new_name = file_strip_name(x)
    with pikepdf.open(x) as pdf:
        del pdf.Root.Metadata
        del pdf.docinfo
        pdf.save(new_name)
        pdf.close()
```

Рисунок 3.8 – Функція видалення метаданих з pdf-файлу

Далі не менш важливою є функція (рисунок 3.9), яка виконує видалення метаданих з усіх файлів у папці, які підтримуються. Спочатку вона отримує шлях до директорії з об'єкта `entry_folder` (об'єкт графічного інтерфейсу для вводу шляху до файлу). Якщо ця директорія не існує, відображається повідомлення про помилку. Потім вона ініціалізує лічильники для зображень, відео і PDF-файлів, які потім використовуються для підрахунку кількості файлів кожного типу, з яких були видалені метадані. За допомогою функції `os.walk()`, функція проходить через всі файли в директорії і її піддиректоріях. Для кожного файлу вона перевіряє, чи відповідає розширення файлу одному з підтримуваних розширень для зображень, відео або PDF. Якщо це так, відповідна функція викликається для видалення метаданих з файлу, а відповідний лічильник збільшується. Після того, як всі файли були оброблені, відображається повідомлення з кількістю зображень, відео і PDF-файлів, з яких були видалені метадані. Код самої функції наведено нижче та як саме виводиться повідомлення про кількість файлів, з яких видалили метадані, зображено на рисунку 3.10.

```
def process_folder():
    folder = entry_folder.get()
    if not os.path.exists(folder):
        result_label.config(text=f"Error: Folder not found")
        return

    image_count, video_count, pdf_count = 0, 0, 0

    for root, _, files in os.walk(folder):
        for file in files:
            file_path = os.path.join(root, file)

            if file.lower().endswith(ext.SUPPORTED_IMAGE_EXTENSIONS):
                delete_and_log_metadata(file_path, 'i')
                image_count += 1
            elif file.lower().endswith(ext.SUPPORTED_VIDEO_EXTENSIONS):
                delete_and_log_metadata(file_path, 'v')
                video_count += 1
            elif file.lower().endswith(ext.SUPPORTED_PDF_EXTENSIONS):
                delete_pdf(file_path)
                pdf_count += 1

    result_label.config(text=f'Metadata removed from {image_count} images, {video_count} videos, and {pdf_count} PDFs')
```

Рисунок 3.9 – Код функції для обробки всіх файлів у директорії

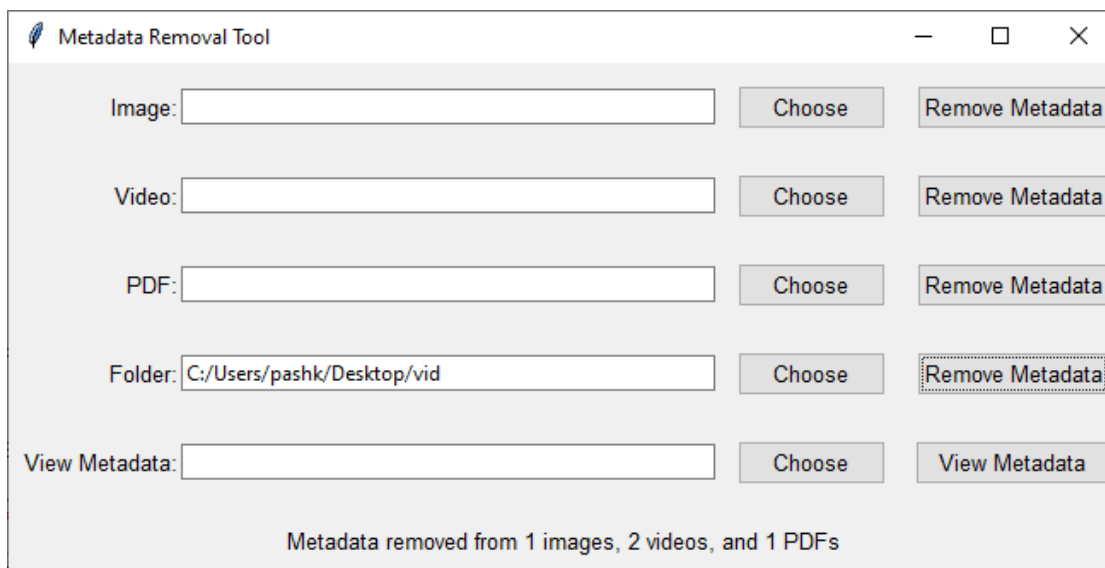


Рисунок 3.10 – Сповіщення про кількість файлів у яких було видалено метадані

У розробленому додатку користувач може не тільки видаляти метадані, а також їх дивитися. Для цього було реалізовано функцію (рисунок 3.11), яка відображає вміст метаданих у файлі у графічному інтерфейсі. Спочатку вона отримує шлях до файлу з об'єкта `entry_metadata` і перевіряє його на валідність, використовуючи функцію `is_valid_file()`. Якщо, наприклад, такого файлу не існує, то виводиться відповідне повідомлення про помилку.

Далі функція перевіряє розширення файлу. Якщо воно відповідає підтримуваним розширенням зображень, відео або pdf-файлу викликається функція `display_metadata()` для отримання метаданих файлу.

Якщо функція `display_metadata()` не повернула метадані, виводиться повідомлення про помилку. Якщо метадані були успішно отримані, створюється нове вікно (так зване "Toplevel" вікно в Tkinter) з заголовком "Metadata". В цьому вікні створюється текстовий об'єкт, де буде відображатися метадані (рисунок 3.12).

Метадані перетворюються на рядок, де кожна пара ключ-значення з метаданих знаходиться на окремому рядку, і цей рядок вставляється в текстовий об'єкт. Нарешті, до вікна додається вертикальна смуга прокрутки (`scrollbar`), що дозволяє прокручувати метадані, якщо вони не поміщаються в вікно.

```

def view_metadata():
    file = entry_metadata.get()
    valid, message = is_valid_file(file, ext.SUPPORTED_EXTENSIONS)

    if not valid:
        result_label.config(text=f"Error: {message}")
        return

    if file.lower().endswith(ext.SUPPORTED_EXTENSIONS):
        metadata = display_metadata(file)

    if not metadata:
        result_label.config(text="Error: Unable to read metadata")
        return

    metadata_window = tk.Toplevel(window)
    metadata_window.title("Metadata")

    metadata_text = tk.Text(metadata_window, wrap=tk.WORD, width=80, height=20)
    metadata_text.pack(expand=True, fill=tk.BOTH)

    metadata_str = '\n'.join([f"{k}: {v}" for k, v in metadata.items()])
    metadata_text.insert(tk.END, metadata_str)

    scrollbar = tk.Scrollbar(metadata_window, command=metadata_text.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
    metadata_text.config(yscrollcommand=scrollbar.set)

```

Рисунок 3.11 – Програмний код функції для перегляду метаданих

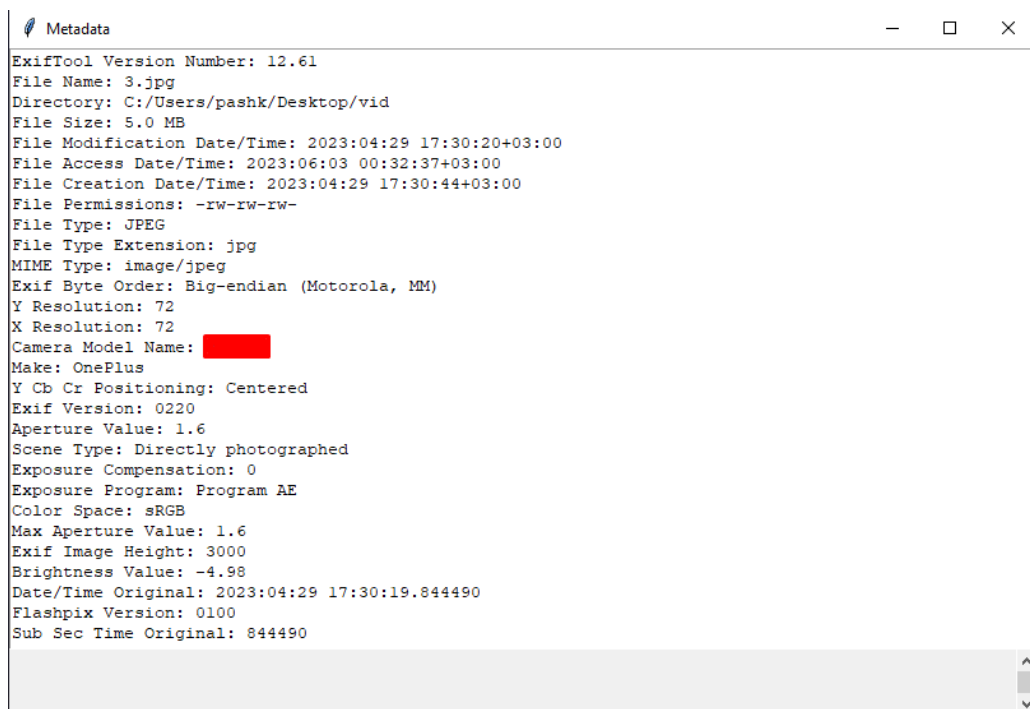


Рисунок 3.12 – Вікно з відображенням метаданих файлу

Наведені та проаналізовані функції були одними з найголовніших у розробленому додатку. Весь програмний код наведений у Додатках А, Б, В.

Висновки до розділу 3

У цьому розділі було описано архітектуру програмного модуля, які технології та інструменти були використанні при розробці, та описано найголовніші функції додатку, для чого вони і що саме роблять. Також було наведено яких функціональних та нефункціональних вимог повинна була дотримуватися розроблена програма.

Програмний модуль реалізовано на мові програмування Python, з використанням деяких базових та сторонніх бібліотек. Застосунок дозволяє користувачу швидко та легко видаляти або продивлятися метадані файлів, завдяки простому графічному інтерфейсу.

ВИСНОВКИ

Результатом виконаної роботи є проаналізована законодавча база України та світу з питань регулювання даних, які містяться у метаданих мультимедійних файлів. Було проведено дослідження з пошуком загроз конфіденційності метаданих мультимедійних файлів та сучасних засобів і механік їх обробки для покращення захисту персональних даних. Також реалізовано програмний застосунок, який забезпечує перегляд та видалення метаданих з мультимедійних файлів для забезпечення їх безпечного використання у мережі.

У першій частині кваліфікаційної роботи було проведено аналіз стандартів метаданих мультимедійних файлів та законів, які регулюють інформацію, що зберігають метадані. Аналіз складався з дослідження нормативно-правової бази в галузі інформаційної безпеки, а саме було виділено декілька основних законів, це: Закон України «Про інформацію», Закон України «Про Захист персональних даних» та GDPR (Загальний регламент захисту даних, який діє на території Європейського союзу). Було розглянуто класифікацію метаданих відповідно до NIST (адміністративні метадані, описові, та структурні). Також було описано основні стандарти метаданих, які використовуються у мультимедійних файлах. Розглядалися наступні стандарти:

- IPTC-IIM
- PLUS
- XMP
- IPTC Core & Extension
- EXIF
- Dublin Core

У другій частині роботи було розглянуто найпоширеніші та найголовніші загрози конфіденційності, які присутні для метаданих мультимедійних файлів. Цими загрозами є розкриття геолокації та особистої інформації і викриття технічних характеристик пристроїв. Також було проведено огляд сучасних методів та технік

обробки метаданих для запобігання цим загрозам. Розглядалися наступні методи та техніки:

- Анонімізація та псевдонімізація
- Шифрування
- Управління доступом
- Використання машинного навчання та штучного інтелекту

У третій частині було розроблено програмний застосунок з простим та інтуїтивним графічним інтерфейсом, який дозволяє переглядати та видаляти метадані мультимедійних файлів (фото, відео, pdf), використовуючи метод анонімізації метаданих.

Враховуючи встановлену мету кваліфікаційної роботи, були виконані наступні завдання:

- Досліджено класифікацію та стандарти метаданих.
- Проведено аналіз можливих загроз на конфіденційність з використанням метаданих у файлах мультимедіа.
- Проведено аналіз сучасних методів та технік обробки метаданих для забезпечення конфіденційності.
- Реалізовано програмний модуль для видалення метаданих з мультимедійних файлів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital 2023 april global statshot report – datareportal – global digital insights [Електронний ресурс] // DataReportal – Global Digital Insights. – Режим доступу: <https://datareportal.com/reports/digital-2023-april-global-statshot>.
2. Meng R. A survey of image information hiding algorithms based on deep learning [Електронний ресурс] / Ruohan Meng, Qi Cui, Chengsheng Yuan // Computer modeling in engineering & sciences. – 2018. – Т. 117, № 3. – С. 425–454. – Режим доступу: <https://doi.org/10.31614/cmcs.2018.04765>.
3. Cisco visual networking index: forecast and trends, 2017–2022 [Електронний ресурс]. – 2019. – Режим доступу: <https://twiki.cern.ch/twiki/pub/HEPIX/TechwatchNetwork/HtwNetworkDocuments/white-paper-c11-741490.pdf>.
4. Americans and privacy: new study finds u.s. consumers overwhelmingly concerned about the lack of control over their personal information [Електронний ресурс] // PR Newswire: press release distribution, targeting, monitoring and marketing. – Режим доступу: <https://www.prnewswire.com/il/news-releases/americans-and-privacy-new-study-finds-us-consumers-overwhelmingly-concerned-about-the-lack-of-control-over-their-personal-information-301532730.html>.
5. Про інформацію [Електронний ресурс] : Закон України від 02.10.1992 р. № 2657-ХІІ : станом на 31 берез. 2023 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2657-12#Text>.
6. Про захист персональних даних [Електронний ресурс] : Закон України від 01.06.2010 р. № 2297-VI : станом на 27 жовтня. 2022 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>.
7. General data protection regulation (GDPR) – official legal text [Електронний ресурс] // General Data Protection Regulation (GDPR). – Режим доступу: <https://gdpr-info.eu/>.
8. Метадані - це що? [Електронний ресурс] // «Pedia» Україна. – Режим доступу: <https://pedia.com.ua/tekhnohii/metadani-tse-shcho.html>.

9. Understanding metadata what is metadata, and what is it for? [Электронный ресурс]. – Режим доступа: <https://groups.niso.org/higherlogic/ws/public/download/17446/Understanding%20Metadata>.
10. Metadata standard - Wikipedia [Электронный ресурс] // Wikipedia, the free encyclopedia. – Режим доступа: https://en.wikipedia.org/wiki/Metadata_standard.
11. Metadata standards | VRA [Электронный ресурс] // VRA |. – Режим доступа: <https://vraweb.org/resources/cataloging-metadata-and-data-management/metadata-standards/>.
12. The IPTC-NAA standards [Электронный ресурс] – Режим доступа: https://controlledvocabulary.com/imagedatabases/iptc_naa.html.
13. Plus [Электронный ресурс] // PLUS – Режим доступа: <https://www.useplus.com/>.
14. XMP metadata [Электронный ресурс] // Adobe Help Center. – Режим доступа: <https://helpx.adobe.com/after-effects/using/xmp-metadata.html>.
15. Standards: IPTC core & extensions | photometadata.org [Электронный ресурс] // Are you meta-smart? | Photometadata.org. – Режим доступа: <https://photometadata.org/META-Resources-metadata-types-standards-IPTC-Core-and-extensions?subject=IPTC%20Core>.
16. What is an EXIF file? [Электронный ресурс]. – Режим доступа: <https://www.adobe.com/creativecloud/file-types/image/raster/exif-file.html>.
17. Dublin core [Электронный ресурс]. – Режим доступа: <https://www.dublincore.org>.
18. Metadata and privacy - A technical and legal overview [Электронный ресурс]. – Режим доступа: https://www.priv.gc.ca/en/opc-actions-and-decisions/research/explore-privacy-research/2014/md_201410/.
19. Face/Off: preventing privacy leakage from photos in social networks [Электронный ресурс] / Elias Athanadoroulos [та ін.]. – 2015. – Режим доступа: https://www.researchgate.net/publication/301419612_FaceOff_Preventing_Privacy_Leakage_From_Photos_in_Social_Networks.

20. Menfors M. Geotagging in social media : exploring the privacy paradox [Електронний ресурс] : thesis / Menfors Martina, Fernstedt Felicia. – [Б. м.], 2015. – Режим доступу: <http://urn.kb.se/resolve?urn=urn:nbn:se:hb:diva-8685>.
21. OWASP top 10:2021 [Електронний ресурс] // OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. – Режим доступу: <https://owasp.org/Top10/>.
22. Carnogursky M. Metadata: a hacker's best friend [Електронний ресурс] / Martin Carnogursky // Sweepatic Blog. – Режим доступу: <https://blog.sweepatic.com/metadata-hackers-best-friend/>.
23. Anonymisation: managing data protection risk code of practice [Електронний ресурс]. – 2012. – Режим доступу: <https://ico.org.uk/media/1061/anonymisation-code>.
24. Агенція Європейського Союзу з питань основоположних прав та Рада Європи Посібник з європейського права у сфері захисту персональних даних [Електронний ресурс] / Агенція Європейського Союзу з питань основоположних прав та Рада Європи, 2018. – 436 с. – Режим доступу: https://www.echr.coe.int/Documents/Handbook_data_protection_UKR.
25. What is python? Executive summary [Електронний ресурс] – Режим доступу: <https://www.python.org/doc/essays/blurb/>.

ДОДАТОК А

Програмний код файлу gui.py

```
import os
import tkinter as tk
from tkinter import filedialog, ttk

from utils import *
from remove_metadata import *
import extensions as ext

def check_exiftool_and_process(function):
    def wrapper(*args, **kwargs):
        if platform.system() == 'Windows':
            install_exiftool()
        return function(*args, **kwargs)
    return wrapper

def check_ffmpegtool_and_process(function):
    def wrapper(*args, **kwargs):
        if platform.system() == 'Windows':
            install_ffmpeg()
        return function(*args, **kwargs)
    return wrapper

def is_valid_file(file, extensions):
    if not os.path.exists(file):
        return False, "File not found"
    elif not file.lower().endswith(extensions):
        return False, "Unsupported file format"
    else:
        return True, "Valid file"

def choose_folder():
    folder = filedialog.askdirectory()
    entry_folder.delete(0, tk.END)
    entry_folder.insert(0, folder)

def choose_file(entry):
    file = filedialog.askopenfilename(filetypes=[("Supported files", ext.SUPPORTED_EXTENSIONS)])
    entry.delete(0, tk.END)
    entry.insert(0, file)

def choose_image_file(entry):
    file = filedialog.askopenfilename(filetypes=[("Image files",
ext.SUPPORTED_IMAGE_EXTENSIONS)])
    entry.delete(0, tk.END)
    entry.insert(0, file)
```

```

def choose_video_file(entry):
    file = filedialog.askopenfilename(filetypes=[("Video files",
ext.SUPPORTED_VIDEO_EXTENSIONS)])
    entry.delete(0, tk.END)
    entry.insert(0, file)

def choose_pdf():
    file = filedialog.askopenfilename(filetypes=[("PDF files", ext.SUPPORTED_PDF_EXTENSIONS)])
    entry_pdf.delete(0, tk.END)
    entry_pdf.insert(0, file)

@check_exiftool_and_process
@check_ffmpegtool_and_process
def process_folder():
    folder = entry_folder.get()
    if not os.path.exists(folder):
        result_label.config(text=f"Error: Folder not found")
        return

    image_count, video_count, pdf_count = 0, 0, 0

    for root, _, files in os.walk(folder):
        for file in files:
            file_path = os.path.join(root, file)

            if file.lower().endswith(ext.SUPPORTED_IMAGE_EXTENSIONS):
                delete_and_log_metadata(file_path, 'i')
                image_count += 1
            elif file.lower().endswith(ext.SUPPORTED_VIDEO_EXTENSIONS):
                delete_and_log_metadata(file_path, 'v')
                video_count += 1
            elif file.lower().endswith(ext.SUPPORTED_PDF_EXTENSIONS):
                delete_pdf(file_path)
                pdf_count += 1

    result_label.config(text=f'Metadata removed from {image_count} images, {video_count} videos, and
{pdf_count} PDFs')

@check_exiftool_and_process
def process_image():

    file = entry_image.get()
    valid, message = is_valid_file(file, ext.SUPPORTED_IMAGE_EXTENSIONS)

    if valid:
        delete_and_log_metadata(file, 'i')
        result_label.config(text='Image metadata removed successfully!')
    else:
        result_label.config(text=f'Error: {message}')

@check_ffmpegtool_and_process

```

```

def process_video():
    file = entry_video.get()
    valid, message = is_valid_file(file, ext.SUPPORTED_VIDEO_EXTENSIONS)

    if valid:
        delete_and_log_metadata(file, 'v')
        result_label.config(text='Video metadata removed successfully!')
    else:
        result_label.config(text=f"Error: {message}")

@check_exiftool_and_process
def process_pdf():
    file = entry_pdf.get()
    valid, message = is_valid_file(file, ext.SUPPORTED_PDF_EXTENSIONS)

    if valid:
        delete_pdf(file)
        result_label.config(text='PDF metadata removed successfully!')
    else:
        result_label.config(text=f"Error: {message}")

@check_exiftool_and_process
def view_metadata():
    file = entry_metadata.get()
    valid, message = is_valid_file(file, ext.SUPPORTED_EXTENSIONS)

    if not valid:
        result_label.config(text=f"Error: {message}")
        return

    if file.lower().endswith(ext.SUPPORTED_EXTENSIONS):
        metadata = display_metadata(file)

    if not metadata:
        result_label.config(text="Error: Unable to read metadata")
        return

    metadata_window = tk.Toplevel(window)
    metadata_window.title("Metadata")

    metadata_text = tk.Text(metadata_window, wrap=tk.WORD, width=80, height=20)
    metadata_text.pack(expand=True, fill=tk.BOTH)

    metadata_str = '\n'.join([f"{k}: {v}" for k, v in metadata.items()])
    metadata_text.insert(tk.END, metadata_str)

    scrollbar = tk.Scrollbar(metadata_window, command=metadata_text.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
    metadata_text.config(yscrollcommand=scrollbar.set)

window = tk.Tk()

```

```

window.title("Metadata Removal Tool")
window.geometry("650x300")

s = ttk.Style()
s.configure('TButton', font=('Helvetica', 10), foreground='black')
s.configure('TLabel', font=('Helvetica', 10), foreground='black')
s.configure('TEntry', background='white')

for i in range(4):
    window.columnconfigure(i, weight=1)

for i in range(5):
    window.rowconfigure(i, weight=1)

label_image = ttk.Label(window, text="Image:")
label_image.grid(row=0, column=0, sticky='e')
entry_image = ttk.Entry(window, width=50)
entry_image.grid(row=0, column=1, sticky='ew')
button_choose_image = ttk.Button(window, text="Choose", command=lambda:
choose_image_file(entry_image))
button_choose_image.grid(row=0, column=2, padx=10)
button_image = ttk.Button(window, text="Remove Metadata", command=process_image)
button_image.grid(row=0, column=3)

label_video = ttk.Label(window, text="Video:")
label_video.grid(row=1, column=0, sticky='e')
entry_video = ttk.Entry(window, width=50)
entry_video.grid(row=1, column=1, sticky='ew')
button_choose_video = ttk.Button(window, text="Choose", command=lambda:
choose_video_file(entry_video))
button_choose_video.grid(row=1, column=2, padx=10)
button_video = ttk.Button(window, text="Remove Metadata", command=process_video)
button_video.grid(row=1, column=3)

label_pdf = ttk.Label(window, text="PDF:")
label_pdf.grid(row=2, column=0, sticky='e')
entry_pdf = ttk.Entry(window, width=50)
entry_pdf.grid(row=2, column=1, sticky='ew')
button_choose_pdf = ttk.Button(window, text="Choose", command=choose_pdf)
button_choose_pdf.grid(row=2, column=2, padx=10)
button_pdf = ttk.Button(window, text="Remove Metadata", command=process_pdf)
button_pdf.grid(row=2, column=3)

label_folder = ttk.Label(window, text="Folder:")
label_folder.grid(row=3, column=0, sticky='e')
entry_folder = ttk.Entry(window, width=50)
entry_folder.grid(row=3, column=1, sticky='ew')
button_choose_folder = ttk.Button(window, text="Choose", command=choose_folder)
button_choose_folder.grid(row=3, column=2, padx=10)
button_folder = ttk.Button(window, text="Remove Metadata", command=process_folder)

```

```
button_folder.grid(row=3, column=3)

label_metadata = ttk.Label(window, text="View Metadata:")
label_metadata.grid(row=4, column=0, sticky='e')
entry_metadata = ttk.Entry(window, width=50)
entry_metadata.grid(row=4, column=1, sticky='ew')
button_choose_metadata = ttk.Button(window, text="Choose", command=lambda:
choose_file(entry_metadata))
button_choose_metadata.grid(row=4, column=2, padx=10)
button_view_metadata = ttk.Button(window, text="View Metadata", command=view_metadata,
width=15)
button_view_metadata.grid(row=4, column=3)

result_label = ttk.Label(window, text="")
result_label.grid(row=5, column=0, pady=10, colspan=4)

def main():
    window.mainloop()
    remove_temp_files()
```

ДОДАТОК Б

Програмний код файлу remove_metadata.py

```

from utils import *
import pikepdf
import time
import extensions as ext

# Function to remove metadata from image file.
def img(x):
    new_file = file_strip_name(x)
    os.system(f"exiftool -all= -o {new_file} {x}")
    return new_file

def delete_pdf(x):
    os.system("exiftool " + x + ">input.txt")
    new_name = file_strip_name(x)

    with pikepdf.open(x) as pdf:
        del pdf.Root.Metadata
        del pdf.docinfo
        pdf.save(new_name)
        pdf.close()

    os.system("exiftool " + new_name + ">output.txt")

# Function to remove metadata from video file.
def vid(file):

    new_file = file_strip_name(file)
    os.system('ffmpeg -i '+file+' -map_metadata -1 -c:v copy -c:a copy '+new_file)
    cls()
    return new_file

# Function to remove metadata from single image or video file.
# file - filename given as input; type: i for images, v - for videos;
def delete_and_log_metadata(file,type):

    success = False
    if not os.path.exists(file):
        print("File Doesn't Exist!\n")
        time.sleep(3)
        exit(0)

    if type.lower()=='i':
        supported_extensions=ext.SUPPORTED_IMAGE_EXTENSIONS
        for ex in supported_extensions:
            if file.lower().endswith(ex):
                os.system("exiftool "+file+">input.txt")

```

```
os.system("exiftool "+img(file)+">output.txt")
success = True
break

elif type.lower()=='v':
    supported_extensions=ext.SUPPORTED_VIDEO_EXTENSIONS
    for ex in supported_extensions:
        if file.lower().endswith(ex):
            os.system("exiftool "+file+">input.txt")
            os.system("exiftool "+vid(file)+">output.txt")
            success = True
            break
        else:
            return 0

return success
```

ДОДАТОК В

Програмний код файлу utils.py

```
try:
    import zipfile
    import platform
    import os
    import urllib.request
    import shutil
    from tkinter import messagebox
    import string
    import random
    import subprocess
    import re

except ImportError:
    print('Critical Error: Necessary Python Modules are missing!')
    print('Commons module couldnot satisfy all the dependencies!')
    input('\nPress any key to exit...')
    exit()

# Function to read and display metadata of a file.
def display_metadata(file):
    metadata = { }
    cmd = f'exiftool {file}'
    process = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True,
text=True)
    stdout, _ = process.communicate()

    for line in stdout.split('\n'):
        match = re.match(r'^(.+)\s+:\s+(.+)$', line)
        if match:
            key, value = match.groups()
            metadata[key.strip()] = value.strip()

    return metadata

def cleanup_temp_files():
    temp_files = ["exiftool-12.61.zip", "ffmpeg-n6.0-latest-win64-gpl-6.0.zip"]

    for temp_file in temp_files:
        if os.path.exists(temp_file):
            os.remove(temp_file)

    temp_folders = ["ffmpeg-n6.0-latest-win64-gpl-6.0"]

    for temp_folder in temp_folders:
        if os.path.exists(temp_folder):
            shutil.rmtree(temp_folder)
```

```

def file_strip_name(filename):
    count = filename.find('.')
    file = filename[:count]
    ext = filename[count:]
    random_string = ''.join(random.choices(string.ascii_letters + string.digits, k=3))
    y = f"{file}_{random_string}{ext}"
    return y

# Clearscreen
def cls():
    if platform.system().lower()=='windows':
        os.system('cls')
    else:
        os.system('clear')

def check_installed(executable):
    for path in os.environ["PATH"].split(os.pathsep):
        path = path.strip("")
        exe_file = os.path.join(path, executable)
        if os.path.isfile(exe_file) and os.access(exe_file, os.X_OK):
            return True
    return False

def download_file(url, file_path):
    with urllib.request.urlopen(url) as response, open(file_path, 'wb') as out_file:
        out_file.write(response.read())

def unzip_and_rename_exiftool(zip_file):
    with zipfile.ZipFile(zip_file, 'r') as zip_ref:
        zip_ref.extractall()

    os.rename("exiftool(-k).exe", "exiftool.exe")

def move_to_windows_folder(exe_file, windows_folder):
    if not os.path.exists(windows_folder):
        os.makedirs(windows_folder)

    abs_exe_file = os.path.abspath(exe_file)
    shutil.move(abs_exe_file, os.path.join(windows_folder, os.path.basename(exe_file)))

def find_ffmpeg_in_folder(folder):
    for root, dirs, files in os.walk(folder):
        if "ffmpeg.exe" in files:
            return os.path.join(root, "ffmpeg.exe")
        if "bin" in dirs:
            bin_folder = os.path.join(root, "bin")
            if "ffmpeg.exe" in os.listdir(bin_folder):
                return os.path.join(bin_folder, "ffmpeg.exe")
    return None

```

```

def move_to_project_folder(ffmpeg_exe, project_folder):
    destination = os.path.join(project_folder, "ffmpeg.exe")
    if not os.path.exists(destination):
        shutil.move(ffmpeg_exe, destination)
    return destination

def install_exiftool():
    exiftool_installed = check_installed("exiftool.exe")

    if not exiftool_installed:
        if messagebox.askyesno("Install exiftool", "Exiftool is not installed. Do you want to install it?"):
            exiftool_url = "https://exiftool.org/exiftool-12.61.zip"
            download_file(exiftool_url, "exiftool-12.61.zip")
            unzip_and_rename_exiftool("exiftool-12.61.zip")
            move_to_windows_folder("exiftool.exe", "C:\Windows")
        cleanup_temp_files()

def install_ffmpeg():
    ffmpeg_installed = check_installed("ffmpeg.exe")

    if not ffmpeg_installed:
        if messagebox.askyesno("Install ffmpeg", "FFmpeg is not installed. Do you want to install it?"):
            ffmpeg_url = "https://github.com/BtbN/FFmpeg-Builds/releases/download/latest/ffmpeg-n6.0-
latest-win64-gpl-6.0.zip"
            download_file(ffmpeg_url, "ffmpeg-n6.0-latest-win64-gpl-6.0.zip")

            with zipfile.ZipFile("ffmpeg-n6.0-latest-win64-gpl-6.0.zip", 'r') as zip_ref:
                zip_ref.extractall()

            ffmpeg_exe = find_ffmpeg_in_folder(".")
            if ffmpeg_exe:
                project_folder = os.path.dirname(os.path.realpath(__file__))
                ffmpeg_in_project = move_to_project_folder(ffmpeg_exe, project_folder)
                move_to_windows_folder(ffmpeg_in_project, "C:\Windows")
            else:
                messagebox.showerror("Error", "ffmpeg.exe not found in the extracted folder.")
        cleanup_temp_files()

def remove_temp_files():
    if os.path.exists("log"):
        os.remove("log")
    if os.path.exists("chk"):
        os.remove("chk")
    if os.path.exists("input.txt"):
        os.remove("input.txt")
    if os.path.exists("output.txt"):
        os.remove("output.txt")

```

ДОДАТОК Г

Список опублікованих праць за темою кваліфікаційної роботи

1. Horbatiuk P. The Crucial Role of Securing Digital Image Metadata: Protecting Privacy, Authenticity, and Integrity / P. Horbatiuk, V. Kostiuchenko, I. Parkhomenko. // VI Міжнародна науково-практична конференція “Проблеми кібербезпеки інформаційно-телекомунікаційних систем” (PCSITS) 27-28 квітня 2023, Київ, Україна, стр.59-60