

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту

Завідувач кафедри комп'ютерних наук,

кандидат технічних наук, доцент

_____ Ірина МАШКІНА

(підпис)

« ____ » _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»

Спеціальність 122 Комп'ютерні науки

Освітня програма 122.00.01 Інформатика

**Тема: РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ КОНВЕРТАЦІЇ ФАЙЛІВ
АУДІО ТА ВІДЕО ФОРМАТІВ**

Виконав

студент групи ІНб-2-21-4.0д

Малиженко Ярослав Михайлович

(підпис)

Науковий керівник

к.т.н. доцент кафедри комп'ютерних наук

Носенко Тетяна Іванівна

(підпис)

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук,

кандидат технічних наук, доцент

(науковий ступінь, вчене звання)

_____ Ірина МАШКІНА

(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
«Розробка мобільного додатку для конвертації файлів аудіо та відео форматів»

Виконавець – студент групи ІНб-2-21-4.0д,
спеціальності 122 Комп'ютерні науки,
освітньої програми 122.00.01 Інформатика

Малиженко Ярослав Михайлович

1. Вихідні дані: Законодавство та нормативно-правові акти України в інформаційній сфері та за напрямом дослідження, зокрема, в галузі конвертації файлів аудіо та відео форматів.
2. Основні завдання: Проаналізувати існуючі мобільні додатки для конвертації мультимедійних файлів та їх функціональні можливості. Дослідити різні бібліотеки та API для роботи з аудіо та відео на мобільних платформах. Вибрати оптимальну платформу та технології розробки мобільного додатку (Android, iOS, кросплатформна розробка). Розробити інтерфейс користувача, який забезпечить зручну та інтуїтивно зрозумілу взаємодію з додатком. Реалізувати модуль конвертації файлів з підтримкою різних форматів та налаштуванням параметрів конвертації. Розробити модуль управління файлами, що дозволить користувачам зберігати та організовувати конвертовані файли. Провести тестування мобільного додатку на різних пристроях та операційних системах.
3. Пояснювальна записка: Обсяг – до 50 стор. формату А4 комп'ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.
4. Графічні матеріали: Презентація
5. Додатки: Не передбачено
6. Строк подання роботи на кафедру: « ____ » _____ 2025 р

Науковий керівник

к.т.н. доцент кафедри комп'ютерних наук

Носенко Тетяна Іванівна

_____ дата

Виконавець

Малиженко Ярослав Михайлович

_____ дата

КАЛЕНДАРНИЙ ПЛАН КВАЛІФІКАЦІЙНОЇ РОБОТИ

№	Етапи виконання	Термін виконання	Відмітка
1	Визначення мети, об'єкта та предмета дослідження. Аналіз аналогічних програмних рішень, визначення їх переваг та недоліків.	Вересень	Виконано
2	Ознайомлення з існуючими алгоритмами конвертації аудіо та відео. Вибір технологій та інструментів для розробки мобільного додатку.	Жовтень	Виконано
3	Розробка загальної архітектури мобільного додатку. Визначення вимог до інтерфейсу користувача та його функціоналу.	Листопад	Виконано
4	Підготовка проміжного звіту з аналізу технологій та вибору архітектури. Написання теоретичної частини дипломної роботи.	Грудень	Виконано
5	Початок розробки мобільного додатку: створення базового інтерфейсу та основних функцій. Розробка механізму конвертації аудіо- та відеофайлів.	Січень	Виконано
6	Оптимізація роботи алгоритмів конвертації та тестування працездатності додатку. Додавання додаткових функцій (наприклад, вибір якості файлів, підтримка різних форматів).	Лютий	Виконано
7	Проведення тестування мобільного додатку на різних пристроях та операційних системах. Аналіз отриманих результатів, усунення недоліків.	Березень	Виконано
8	Оформлення дипломної роботи: опис розробки, аналіз проведених тестів, висновки. Підготовка до захисту: перевірка відповідності вимогам ВНЗ.	Квітень	Виконано
9	Фінальне редагування та підготовка презентації до захисту. Захист дипломної роботи.	Травень	

РЕФЕРАТ

Кваліфікаційна робота: 45 с., 11 рис., 2 табл., 37 посилань.

Актуальність: В умовах сучасного світу цифрових технологій потреба в універсальних та ефективних інструментах для роботи з мультимедійними файлами постійно зростає. Формати аудіо та відео різноманітні, і часто виникає необхідність їх конвертації для сумісності з різними пристроями або платформами. Актуальність теми розробки мобільного додатку для конвертації медіа обумовлена запитом на зручність і доступність таких рішень у повсякденному житті.

Об'єкт дослідження: конвертування файлів аудіо та відео форматів на мобільних пристроях.

Предмет дослідження: реалізація клієнт-серверної архітектури у мобільному застосунку.

Мета роботи: створити мобільний застосунок для конвертації медіа файлів, що забезпечить користувачеві швидке та зручне конвертування.

Завдання:

1. Проаналізувати і дослідити існуючі мобільні додатки для конвертації, виділити їх переваги та недоліки і врахувати у своєму мобільному застосунку
2. Сформулювати вимоги до розроблюваного мобільного додатку, включаючи функціональні можливості, інтерфейс користувача, продуктивність та сумісність з різними пристроями
3. Спроекувати архітектуру мобільного додатку, визначити основні модулі та їх взаємодію
4. Розробити зручний та інтуїтивно зрозумілий інтерфейс користувача для мобільного додатку
5. Обрати та реалізувати методи для обробки та конвертації файлів медіа форматів

6. Провести тестування мобільного додатку на різних пристроях та оцінити його ефективність та зручність використання

ОСНОВНІ РЕЗУЛЬТАТИ:

У процесі виконання кваліфікаційної роботи було розроблено мобільний додаток для конвертації аудіо та відео файлів, що дозволяє:

- Вибрати вхідний файл для конвертації.
- Налаштувати параметри конвертування такі як: формат, якість(бітрейт) та роздільна здатність.
- Виконати конвертацію.
- Отримати вихідний файл, що автоматично зберігається на пристрій.

Також було досліджено нові технології та компоненти:

- C# – це об'єктно-орієнтована мова програмування з простим зрозумілим синтаксисом та безпечною системою типізації для платформи .NET.
- Visual Studio – це інтегроване середовище розробки (IDE) від Microsoft, яке використовується для написання, відладки та компіляції програм.
- .NET MAUI – це кросплатформний фреймворк від Microsoft для створення мобільних додатків на C# та XAML. Платформа .NET MAUI є наступним поколінням після Xamarin Forms, яку Microsoft перестала підтримувати у 2024 році.
- ASP.NET Core – це платформа, створена Microsoft для проектування інтерактивних вебзастосунків, які працюють на платформі .NET, це технологія сервера, яка генерує HTML-сторінки на боці сервера, а не клієнта.
- FFmpeg – це програмна утиліта для обробки відео, аудіо та інших мультимедійних файлів. Він дозволяє записувати, конвертувати, передавати та перетворювати цифрові відео та аудіо файли в різних форматах.

- Ngrok – це потужна мережна утиліта, яка дозволяє користувачам ділитися або отримувати доступ до будь-яких локально розміщених веб-програм через загальнодоступний веб-URL.
- **Практичне значення дослідження:** полягає у створенні рішення, яке дозволяє здійснювати віддалену конвертацію мультимедійних файлів із можливістю налаштування параметрів виводу. Такий підхід забезпечує масштабованість, економію ресурсів мобільного пристрою та може бути інтегрований у різноманітні інформаційні системи або онлайн-сервіси, що потребують попередньої обробки медіафайлів.

Ключові слова: мобільний додаток, клієнт-серверна архітектура, конвертація, формати, роздільна здатність, Android, .NET MAUI, ASP.NET Core, FFmpeg, Ngrok.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ТЕМИ, ТЕХНОЛОГІЙ ТА РІШЕНЬ У СФЕРІ МЕДІАКОНВЕРТАЦІЇ	10
1.1 Визначення цільової аудиторії	10
1.2 Огляд мобільних додатків для конвертації аудіо та відео	11
1.3 Аналіз платформ для розробки мобільних додатків	12
1.4 Формати медіафайлів: особливості, переваги, обмеження та застосування	14
1.5 Інструмент обробки медіаданих: можливості, продуктивність та переваги	16
Висновки до розділу 1	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ СТРУКТУРИ ТА ВИБІР ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ	19
2.1 Постановка задачі та технічних вимог до програмного забезпечення	19
2.2 Вибір інструментів реалізації	20
2.3 Розробка архітектури додатку	22
Висновки до розділу 2	25
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ	26
3.1 Розробка клієнтської частини мобільного застосунку	26
3.2 Запуск сервера і підключення URL-адреси	30
3.3 Серверна логіка обробки і конвертації медіа	32
3.3.1 Обробка відеофайлів	32
3.3.2 Обробка аудіофайлів	33
3.3.3 Інтеграція з FFmpeg та керування файлами	34
3.3.4 Налаштування та робота зовнішнього доступу через Ngrok	35
3.4 Перевірка функціональності та результати тестування	36
Висновки до розділу 3	39
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ВСТУП

На сьогоднішній день мультимедійний контент займає важливе місце у повсякденному житті користувачів. Діти, підлітки, дорослі, старі – усі вони так чи інакше щодня зіткаються з медіа контентом в будь-якій його формі, будь то відео на ютуб, інстаграм, тік ток або музика яка постійно грає звідусіль. Всі ці фактори призводять до зростання обсягів відео- та аудіоінформації, що супроводжується потребою в ефективних засобах їх обробки, зокрема – конвертації між форматами для сумісності з різними пристроями, платформами або вимогами користувача. Попри існування численних сервісів для конвертування, більшість із них мають обмежений функціонал або вимагають потужних ресурсів пристрою. Тому актуальною є розробка рішень, здатних поєднувати мобільність, продуктивність та зручність налаштувань.

Проведений аналіз науково-технічних джерел, таких як матеріали з Microsoft Learn, документація .NET MAUI, документація FFmpeg та інші галузеві ресурси, свідчить про інтенсивний розвиток інструментів для мультимедійної обробки. Проте виявлено, що кількість мобільних застосунків для обробки файлів є не такою великою, як здається на перший погляд. Дослідження провідних компаній, таких як Adobe, Movavi та HandBrake, показують нам зростаючий попит на мультимедійні конвертери, але більшість їх рішень є платними або складними для звичайного користувача.

Мета дослідження – створити мобільний застосунок для конвертації медіа файлів, що забезпечить користувачеві швидке та зручне конвертування.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати існуючі рішення на ринку.
2. Створити план з вимогами до розроблюваного додатку
3. Спроекувати і розробити архітектуру.
4. Реалізувати інтерфейс на клієнтській частині.
5. Реалізувати обробку файлів на серверній частині.

6. Провести тестування та оцінку ефективності роботи застосунку.

Об’єкт дослідження – конвертування файлів аудіо та відео форматів на мобільних пристроях.

Предмет дослідження – реалізація клієнт-серверної архітектури у мобільному застосунку.

Методи дослідження: аналіз літературних джерел, проектування інтерфейсу, розробка серверного API, тестування.

Практичне значення дослідження: полягає у створенні готового рішення, яке дозволить конвертувати медіафайли в мобільному застосунку, що може використовуватись у персональному користуванні, а також у професійній сфері для попередньої обробки даних.

Галузь застосування: інформаційні технології, мобільні додатки, обробка мультимедійного контенту, навчальні платформи.

Апробація: мої тези кваліфікаційної роботи було опубліковано на всеукраїнській конференції «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ –2025».

Посилання: <https://zcit.kubg.edu.ua/index.php/journal/issue/view/13/23>

Сторінки 161-163.

РОЗДІЛ 1. АНАЛІЗ ТЕМИ, ТЕХНОЛОГІЙ ТА РІШЕНЬ У СФЕРІ МЕДІАКОНВЕРТАЦІЇ

В епоху, коли мультимедійний контент є невід'ємною частиною повсякдення, питання перетворення аудіо- та відеофайлів набуває особливої уваги. Часто виникає потреба адаптувати формати медіа для їх коректного відтворення на різних пристроях або сумісності з різними програмними продуктами. Створення мобільного застосунку, є прямою відповіддю на цю потребу, надаючи користувачам практичні рішення.

1.1 Визначення цільової аудиторії

Цільовою аудиторією мобільного застосунку для конвертації може бути широке коло користувачів, які стикаються з потребою адаптації аудіо- та відеофайлів, але не обов'язково мають глибокі технічні знання. Це можуть бути:

- Звичайні юзери мобільних пристроїв, яким періодично потрібно конвертувати файли для особистих цілей – наприклад, для відтворення на різних гаджетах, зменшення розміру для зручного надсилання через месенджери чи електронну пошту, або для завантаження в соціальні мережі. Вони цінують простоту та швидкість.
- Контент-мейкери та блогери-початківці, які створюють власний медіаконтент для особистих сторінок, невеликих проектів чи соціальних мереж і потребують легкого інструменту для підготовки файлів у відповідних форматах без необхідності вивчати складне професійне програмне забезпечення.
- Студенти та освітяни, які можуть стикатися з необхідністю конвертувати навчальні відео, аудіозаписи лекцій, презентації чи інші освітні матеріали для сумісності або оптимізації.

Загалом, це користувачі, для яких є важливим баланс між функціональністю, простотою використання та ефективністю обробки файлів.

1.2 Огляд мобільних додатків для конвертації аудіо та відео

Ринок мобільних додатків пропонує широкий вибір інструментів для конвертації. Однак, багато існуючих рішень мають певні недоліки [1], такі як обмежений функціонал, не завжди інтуїтивний інтерфейс або низька швидкість обробки. Часто юзери стикаються з необхідністю встановлювати одразу декілька різних додатків для роботи з різними типами файлів або для отримання бажаного набору функцій. Аналізуючи ринок медіа конвертерів(див. табл. 1.1) видно, що хоча вони й виконують базові завдання, їм часто бракує гнучкості в налаштуваннях конвертації. Також проблемою можна вважати конвертацію безпосередньо на пристрої, це може значно навантажувати його ресурси та знизити швидкість конвертації, особливо, якщо сам пристрій не є потужним [2]. Це створює потребу в розробці мобільного додатку, який би пропонував збалансоване поєднання простоти використання та ефективної обробки файлів.

Таблиця 1.1

Порівняльний аналіз популярних мобільних додатків для конвертування

Критерії	Video Converter, Compressor	Video Format Factory	Media Converter	Timbre: Cut, Join, Convert Mp3 Audio & Mp4 Video	The Video Converter / Audio Converter
Розробник	VidSoftLab	Keerby	Antvplayer	Timbre Inc.	Float Tech
Підтримувані ОС	Android	Android	Android	Android	iOS (окремі додатки)
Основні функції	Відео/аудіо конвертація; стиснення; обрізка	Відео/аудіо конвертація; стиснення; пакетна обробка	Відео/аудіо конвертація; обрізка; зміна бітрейту	Аудіо/відео конвертація, нарізка, склеювання; зміна швидкості.	Відео конвертація; аудіо конвертація (окремо)
Підтримувані формати	Широкий спектр (MP4, MKV, MP3, AAC).	Широкий спектр (MP4, FLV, MP3, WAV).	Широкий спектр (MP4, AVI, MP3).	Широкий спектр (MP3, WAV, FLAC, MP4, AVI, MKV).	Широкий спектр (MOV, AVI, WEBM).
Інтерфейс	Простий, інтуїтивний	Функціональний, потребує звикання	Простий, мінімалістичний	Сучасний, інтуїтивний, багатофункціональний	Сучасний, інтуїтивний
Переваги	Широка підтримка форматів; стиснення	Багато форматів; пакетна обробка	Простота; швидкість	Багатофункціональність; якість	Спеціалізація; багато форматів
Недоліки	Реклама; платні Pro-функції.	Реклама; застарілий інтерфейс.	Реклама; малий функціонал.	Платні розширені функції; складність для новачків.	Реклама; обмеження безкоштовних версій; два додатки.

1.3 Аналіз платформ для розробки мобільних додатків

Вибір платформи для розробки мобільного додатку є визначальним етапом, що впливає на швидкість розробки, продуктивність, доступ до функціоналу та потенційні можливості масштабування. Детальний аналіз показано в таблиці 2.1. Для розробки мобільного додатку було обрано платформу .NET MAUI (Multi-platform App UI). .NET MAUI представляє з себе наступне покоління Xamarin.Forms і дозволяє створювати нативні мобільні та десктопні додатки з єдиної кодової бази на C# та XAML.

Ключовими перевагами .NET MAUI, що зумовили його вибір, є:

Єдина кодова база, яка закладає фундамент для потенційного розширення і на інші платформи з мінімальним доопрацюванням. Хоча поточна версія додатку орієнтована на платформу Android.

.NET MAUI надає механізми для взаємодії з нативними API функціями кожної платформи, що важливо для реалізації таких завдань, як доступ до файлової системи та робота з дозволами.

Завдяки оптимізації у .NET та архітектурі рендерингу, .NET MAUI забезпечує високу продуктивність, наближену до нативних додатків.

Для опису інтерфейсу використовується мова розмітки XAML, що дозволяє декларативно визначати візуальні елементи та їхню взаємодію.

Серверна частина додатку, відповідальна за обробку та конвертацію файлів, реалізується на платформі ASP.NET Core. Це сучасний фреймворк для створення веб-API, який забезпечує обробку HTTP-запитів від мобільного клієнта. За допомогою такого розподілу на клієнтську та серверну частини, ресурсомісткі операції конвертації переносяться на сервер, розвантажуючи мобільний пристрій.

Таблиця 2.1

Порівняльний аналіз популярних платформ розробки мобільних додатків

Критерії	Swift (iOS)	Kotlin/Java (Android)	Flutter	React Native	.NET MAUI
Основна мова	Swift	Kotlin, Java	Dart	JavaScript, TypeScript	C#
Цільові платформи	iOS, iPadOS, macOS, watchOS, tvOS	Android, Android TV, Wear OS	iOS, Android, Web, Desktop	iOS, Android, Web Desktop	iOS, Android, Windows, macOS
Продуктивність	Дуже висока	Дуже висока	Висока	Середня	Висока
Інструменти (UI)	UIKit, SwiftUI	Android UI Toolkit, Jetpack Compose	Material Design, Cupertino	JavaScript	XAML
Екосистема та підтримка спільноти	Apple	Google	Google	Facebook Meta	Microsoft
Складність освоєння	Висока	Середня	Низька	Низька	Середня
Швидкість розробки	Середня	Середня	Дуже висока	Висока	Висока
Основні переваги	Продуктивність; безпека; екосистема Apple	Величезний ринок; гнучкість; відкритість	Швидка розробка; гарний UI; продуктивність	Велика спільнота; веб-технології; швидкий вихід на ринок	Єдина кодова база; інтеграція з .NET
Основні недоліки	Обмеження Apple; дорожка розробка	Фрагментація пристроїв; складне тестування	Великий розмір додатку	Проблеми продуктивності	Не велика спільнота; обмеження нативних функцій

1.4 Формати медіафайлів: особливості, переваги, обмеження та застосування

Мультимедійні формати загалом поділяються на два основні типи: аудіо та відео, кожен з яких має свої особливості, переваги, обмеження та сфери застосування.

Аудіоформати, такі як MP3, AAC та WAV, призначені для зберігання звукових даних у стиснутому або нестиснутому вигляді. Наприклад, MP3, що є найпоширенішим форматом стиснення аудіо з втратами, здобув

популярність завдяки сумісності з усіма пристроями та програмами, а також малому розміру файлів при достатньо хорошій якості, хоча й відбувається втрата частини аудіоданих [3], що може бути помітною при низьких бітрейтах. AAC, також формат стиснення аудіо з втратами, часто вважається ефективнішим за MP3, забезпечуючи кращу якість звуку при аналогічному бітрейті, особливо на низьких значеннях, але він так само є форматом із втратами. На противагу їм, WAV зазвичай зберігає нестиснуті аудіодані, що гарантує найвищу якість звуку, ідентичну оригіналу, та робить його ідеальним для професійного використання, наприклад, звукорежисерами для подальшої обробки, однак це призводить до дуже великого розміру файлу [4].

Відеоформати, такі як MP4, MOV та AVI, дозволяють зберігати як відеоряд, так і аудіодоріжки, і їх особливості визначають сфери застосування. Як приклад, MP4 (MPEG-4) є контейнерним форматом, що може містити відео (часто H.264/AVC або H.265/HEVC) та аудіо (зачасту це аудіоформат AAC), і він став стандартом для більшості пристроїв завдяки найвищій сумісності з плеєрами та веб-платформами, а також хорошему балансу між якістю та розміром файлу, хоча якість сильно залежить від використовуваних кодеків та параметрів стиснення. MOV (QuickTime File Format), розроблений Apple, також є контейнерним форматом, що часто використовується для зберігання відео високої якості, особливо в екосистемі Apple, та підтримує різноманітні кодеки, проте може мати обмежену нативну підтримку на платформах, що не належать Apple, хоча багато сучасних плеєрів його відтворюють. AVI (Audio Video Interleave), старіший контейнерний формат від Microsoft, має широку підтримку старішими пристроями та програмним забезпеченням, але може мати більший розмір файлу порівняно з MP4 при аналогічній якості та є менш ефективним для сучасних кодеків і потокового відтворення [5].

Таким чином, вибір конкретного формату залежить від пріоритетів користувача, чи то максимальна якість, мінімальний розмір файлу, чи широка сумісність, що й обумовлює їх застосування від професійних завдань до

повсякденного споживчого використання, як-от прослуховування музики у MP3 чи перегляд відео у MP4 на смартфонах [6-7].

1.5 Інструмент обробки медіаданих: можливості, продуктивність та переваги

Найголовнішим елементом системи конвертації медіафайлів є інструмент, що безпосередньо виконує перетворення даних.

FFmpeg являє собою кросплатформний набір бібліотек та інструментів командного рядка, призначений для запису, конвертації та потокової передачі аудіо й відео. Завдяки своїй винятковій універсальності та підтримці величезної кількості форматів, кодеків і протоколів, FFmpeg визнаний де-факто стандартом у галузі обробки мультимедіа. В рамках дипломної роботи, серверна частина приймає завантажений від клієнта файл разом із заданими параметрами конвертації, після чого формує відповідну команду для FFmpeg і запускає його для виконання перетворення [8-9].

Відеоконвертація:

```
ffmpeg -i {inputPath} -vf scale={resolution} -b:v {bitrate}k {outputPath}
```

Аудіоконвертація:

```
ffmpeg -i {inputPath} -b:a {bitrate}k {outputPath}
```

Після завершення процесу конвертації, оброблений вихідний файл повертається клієнту. Ключовими перевагами використання FFmpeg є його висока продуктивність, надзвичайна гнучкість налаштувань, доведена надійність та підтримка широкого спектру операцій з мультимедійними даними.

Взаємодія клієнта та сервера:

- Мобільний клієнт на .NET MAUI дозволяє користувачеві обрати файл зі сховища пристрою.

- Обраний файл разом із параметрами конвертації надсилається на сервер за допомогою HTTP POST-запиту.
- Сервер приймає запит, зберігає вхідний файл у тимчасову створену папку, викликає FFmpeg для конвертації, а потім надсилає сконвертований файл назад клієнту.
- Клієнт отримує відповідь від сервера, і якщо конвертація успішна, зберігає отриманий файл у загальнодоступну теку "Downloads" на Android-пристрої.
- Для тестування зв'язку з сервером використовується утиліта Ngrok для створення публічної URL-адреси локального сервера, яка дозволяє мобільному пристрою надсилати запити на локальний веб-сервер.

Висновки до розділу 1

Аналіз технологій та рішень для конвертації, проведений у цьому розділі, став підґрунтям для розробки мобільного застосунку. Було виявлено, що в умовах повсюдного використання мультимедійного контенту існує стійка потреба в ефективних інструментах для перетворення аудіо- та відеофайлів, адаптації їх для різних пристроїв та програмних продуктів.

Аналіз існуючих аналогів показав, що попри широкий вибір на ринку, багато рішень мають суттєві недоліки. Серед них – обмежений функціонал, не завжди інтуїтивний інтерфейс або низька швидкість обробки. Це визначило необхідність створення застосунку, що пропонував би збалансоване поєднання простоти використання, гнучкості налаштувань та ефективної обробки файлів.

Вибір платформ для розробки застосунку був зумовлений прагненням до ефективності та потенціалу для масштабування. Для цього було обрано .NET MAUI, що дозволяє створювати нативні мобільні додатки з єдиною кодовою базою, яка може використовуватись на всіх доступних операційних системах. Серверна частина, відповідальна за обробку та конвертацію, реалізується на ASP.NET Core, фреймворку для створення веб-API, що дозволяє ефективно обробляти запити від мобільного клієнта.

Основним елементом системи конвертації визначено інструмент FFmpeg, який є стандартом у галузі обробки мультимедіа. Його кросплатформність, підтримка величезної кількості форматів і кодеків, висока продуктивність та гнучкість налаштувань роблять його оптимальним вибором для розроблюваного додатку.

Таким чином, проведений аналіз дозволив не тільки виявити актуальну проблему та існуючі прогалини на ринку мобільних конвертерів, але й обґрунтувати вибір ключових технологій та підходів до реалізації мобільного застосунку, який би ефективно вирішував завдання медіаконвертації забезпечуючи зручний користувацький досвід.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СТРУКТУРИ ТА ВИБІР ІНСТРУМЕНТІВ РЕАЛІЗАЦІЇ

2.1 Постановка задачі та технічних вимог до програмного забезпечення

Основною задачею є реалізація мобільного застосунку, який надасть користувачам можливість зручно конвертувати їх медіа файлів.

Ключові завдання для програмного забезпечення:

1. Вибір файлу: Надання користувачеві можливості вибрати файл з сховища свого пристрою.
2. Налаштування параметрів конвертації: Забезпечити вибір найпопулярніших форматів, а також налаштування специфічних параметрів:
Для відео: формат (MP4, MOV, AVI), роздільна здатність, бітрейт.
Для аудіо: формат (MP3, AAC, WAV), бітрейт.
3. Передача даних на сервер: Реалізація функціоналу надсилання обраного файлу та заданих параметрів конвертації на сервер.
4. Серверна обробка: Приймання файлу та параметрів на сервері і виконання операції конвертації [10].
5. Отримання та збереження результату: Приймання сконвертованого файлу від сервера та його автоматичне збереження у доступну директорію на пристрої.
6. Інтерфейс користувача: Розробка інтерфейсу, що забезпечить користувачу легку навігацію та взаємодію з усіма функціями додатку.
7. Обробка дозволів: Реалізація запиту та перевірки необхідних дозволів на доступ до файлової системи пристрою.
8. Обробка помилок та логування: Забезпечення системи обробки можливих помилок на кожному етапі роботи (клієнт, сервер) та логування ключових подій.

Технічні вимоги до програмного забезпечення:

Платформа клієнта: Android.

Технологія розробки клієнта: .NET MAUI.

Технологія розробки сервера: ASP.NET Core.

Інструмент конвертації на сервері: FFmpeg.

Протокол взаємодії: HTTP (для передачі файлів та даних між клієнтом і сервером).

Зберігання файлів: Загальнодоступна тека на пристрої (папка "Downloads").

Безпека: Забезпечення безпечної передачі даних (HTTPS).

Обмеження розміру файлу: Сервер має приймання файли великого розміру (зняти обмеження `MaxRequestBodySize` в Kestrel).

2.2 Вибір інструментів реалізації

Обраний технологічний стек для реалізації програми:

1. C#: представляє з себе сучасну мову програмування з дуже великим попитом на сьогоднішній день у великої кількості компаній, це об'єктно-орієнтована мова програмування, розроблена Microsoft. Вона є основною мовою для розробки на платформі .NET. Її перевагами є висока продуктивність, строга типізація, уніфікація, велика стандартна бібліотека класів, а також потужна підтримка з боку інтегрованих середовищ розробки [11-12].
2. .NET MAUI (Multi-platform App UI): кросплатформний фреймворк, що дозволяє створювати нативні інтерфейси користувача з єдиної кодової бази C# та XAML [13]. Він транслює елементи інтерфейсу в нативні компоненти платформи, що забезпечує високу продуктивність та природний вигляд додатків на будь-яких пристроях.

Фреймворк також забезпечує зручний доступ до нативних API платформи, необхідних для роботи з файловою системою, дозволами та іншими специфічними функціями пристрою [14]. Інтеграція з .NET дає змогу використовувати популярні бібліотеки та інструменти, що

розширює можливості розробки [15]. Завдяки архітектурі .NET MAUI, можна створювати застосунки, що працюють на всіх платформах, забезпечуючи повну кросплатформність.

3. ASP.NET Core: кросплатформний фреймворк, який забезпечує можливість створення веб-додатків та API на стороні сервера [16]. Він надає можливість використовувати Minimal APIs [17], що дозволяють швидко створювати HTTP-ендпоінти з мінімальною кількістю коду, що ідеально підходить для реалізації API конвертації. Фреймворк має вбудовані засоби для роботи з HTTP-запитами та відповідями, включаючи підтримку завантаження файлів через IFormFile, а також за замовчуванням інтегрується з веб-сервером Kestrel і дозволяє налаштовувати максимальний розмір тіла запиту.

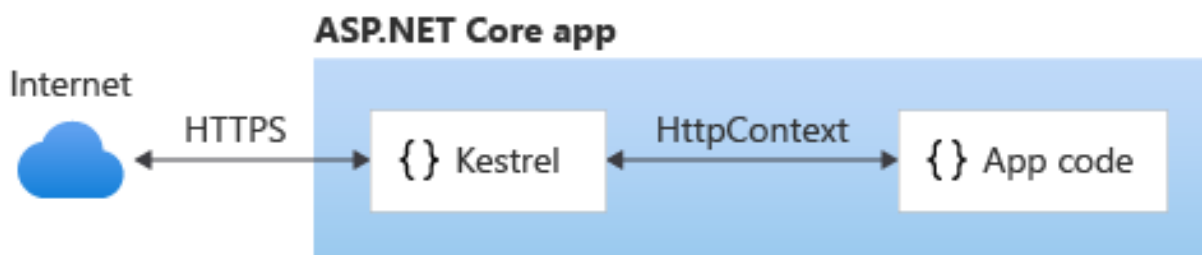


Рис 2.1 Принцип роботи Kestrel

4. FFmpeg: потужний та універсальний інструмент командного рядка для виконання операцій з аудіо- та відеофайлами. Він підтримує практично всі відомі аудіо- та відеоформати й кодеки [18], що робить його надзвичайно гнучким і придатним для роботи з будь-яким мультимедійним вмістом. FFmpeg дозволяє точно налаштовувати параметри конвертації, що важливо для досягнення бажаної якості та розміру файлів [19]. Це перевіроне часом рішення, яке широко використовується в індустрії завдяки своїй надійності. У серверному середовищі ASP.NET Core FFmpeg викликається як окремий системний процес (через `System.Diagnostics.Process`), що дозволяє керувати його виконанням і отримувати результати або повідомлення про помилки.

5. Ngrok: сервіс, що дозволяє розгорнути локальний сервер ASP.NET Core в Інтернет, роблячи його доступним для мобільного клієнта, що працює на фізичному пристрої або емуляторі, без необхідності розгортання сервера на публічному хостингу на етапі розробки [20-21].

2.3 Розробка архітектури додатку

Для забезпечення ефективної роботи та чіткого розподілу функцій, архітектура додатку будується за клієнт-серверним принципом, що передбачає два основні взаємодіючі компоненти [22].

Клієнтська частина:

Бере на себе відповідальність за надання зручного та інтуїтивно зрозумілого користувацького інтерфейсу, через який здійснюється вся взаємодія та навігація. Її ключовими завданнями є надання користувачеві можливості вибору файлів для конвертації, налаштування необхідних параметрів, формування та надсилання структурованих запитів на сервер, а також отримання сконвертованих файлів та їх збереження на пристрої. Увесь користувацький інтерфейс клієнтської частини планується реалізувати з використанням технологій XAML опису візуальних елементів та C# для програмної логіки [23]. Структурно інтерфейс програми складатиметься з 3 сторінок – головної сторінки, яка є початковою точкою при запуску програми і де користувачеві надається вибір типу конвертації – аудіо або відео. Залежно від вибору, користувач переходить на відповідні сторінки конвертації. Ці сторінки міститимуть усі необхідні елементи управління: кнопку для вибору файлу з пристрою, списки для налаштування параметрів, кнопка запуску процесу конвертування та кнопка повернення на головну для зручної навігації по застосунку. Логіка взаємодії з користувачем на клієнтській стороні охоплює повний цикл: від вибору файлу, через автоматичне формування запиту з усіма параметрами, його надсилання на сервер, подальшу обробку відповіді від сервера (успішної чи з помилкою),

збереження отриманого сконвертованого файлу на пристрої та інформування користувача про результат операції через відповідні сповіщення.

Серверна частина:

Відповідає за приймання запитів, що надходять від клієнтської частини. Її функціонал включає валідацію отриманих даних (файлу та параметрів конвертації) на коректність, безпосереднє виконання операції конвертації файлів за допомогою спеціалізованих інструментів, та повернення фінального результату (сконвертованого файлу або повідомлення про помилку) назад клієнту. Для взаємодії з клієнтом серверна частина надаватиме чітко визначені ендпоінти API. Зокрема, передбачається ендпоінт POST /convert/video, який прийматиме відеофайл, бажаний цільовий формат, роздільну здатність та бітрейт. Аналогічно, для аудіофайлів буде реалізований ендпоінт POST /convert/audio, що прийматиме аудіофайл, цільовий формат та бітрейт. Процес обробки запитів на сервері буде стандартизованим: після приймання файлу та параметрів він тимчасово зберігатиметься на сервері. Далі, на основі отриманих параметрів, буде формуватися відповідна команда для FFmpeg. Після цього запускатиметься сам процес конвертації. Сервер відслідковуватиме результат виконання команди FFmpeg, і залежно від успішності операції, надсилатиме клієнту або сконвертований файл, або повідомлення про помилку. Завершальним етапом обробки кожного запиту буде очищення тимчасових файлів, що були створені під час конвертації, для звільнення ресурсів сервера [24].

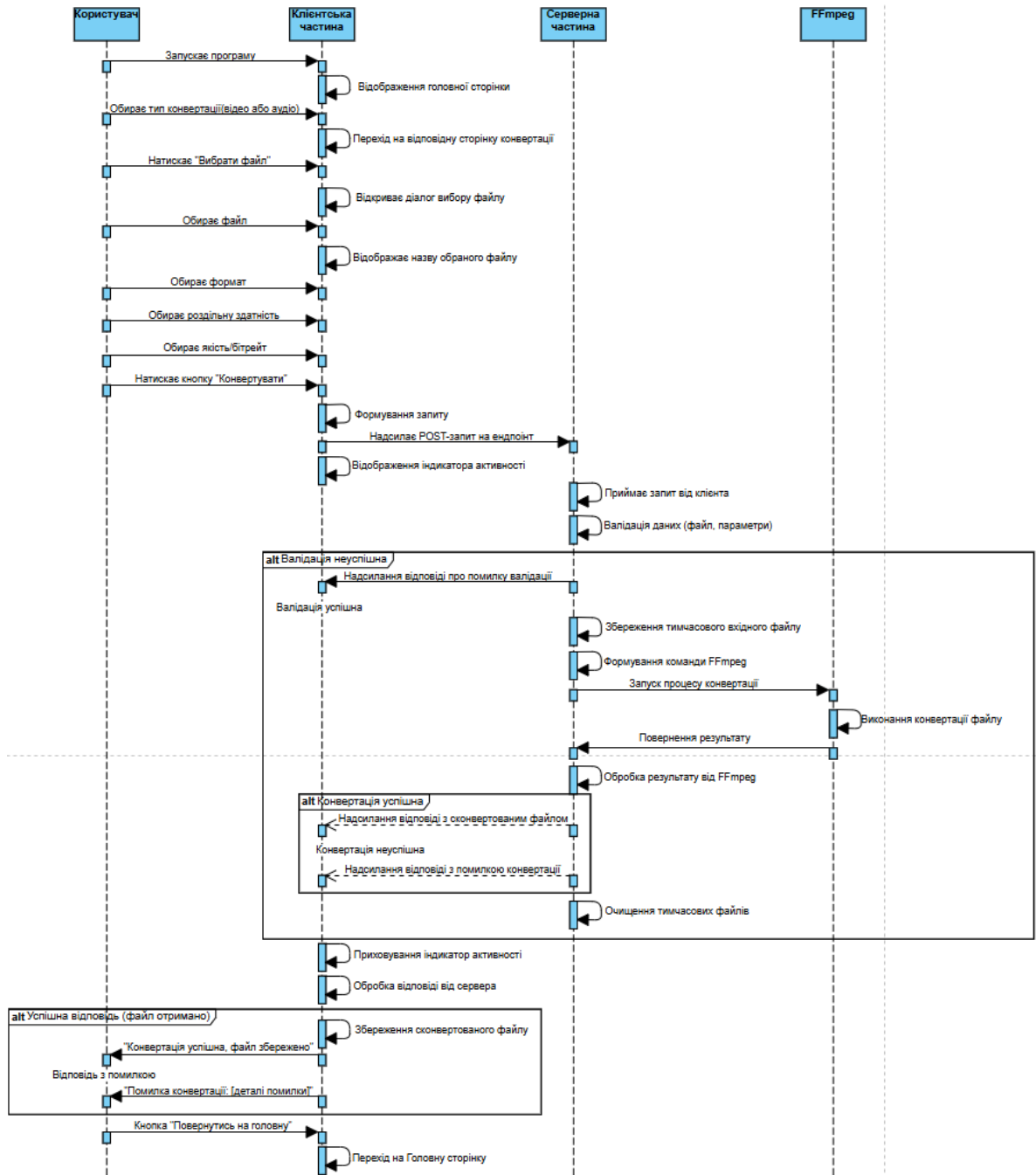


Рис 2.2 Схеми взаємодії

На цій схемі представлений повний процес конвертації, що реалізований за клієнт-серверною архітектурою. Користувач взаємодіє з клієнтською частиною додатку, після чого клієнт формує та надсилає запит на сервер. Серверна частина приймає цей запит, обробляє та звертається до FFmpeg, який в свою чергу виконує перетворення файлу, і після завершення повертає сконвертований файл або повідомлення про результат назад клієнту.

Завершується процес тим, що клієнтська частина обробляє отриману відповідь, зберігає файл (у разі успіху) та інформує користувача.

Висновки до розділу 2

Було сформульовано основні завдання та визначено технічні вимоги до програмного забезпечення, включаючи підтримку найпоширеніших форматів, можливість налаштування параметрів конвертації та реалізацію клієнт-серверної взаємодії.

Були обрані та обгрунтовані інструменти для реалізації. Сформовано клієнт-серверну архітектуру, що передбачає чіткий розподіл відповідальності: мобільний клієнт забезпечує взаємодію з користувачем та надсилання даних, тоді як сервер виконує усі операції і повертає клієнту. Це дозволяє оптимізувати використання ресурсів мобільного пристрою та забезпечити вищу продуктивність процесу конвертації.

Підводячи підсумок, цей етап проектування дозволив закласти міцний фундамент для подальшої роботи та розробки програмного продукту, визначивши його структуру, функціональні компоненти та технологічну базу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ОЦІНКА РЕЗУЛЬТАТІВ

3.1 Розробка клієнтської частини мобільного застосунку

Клієнтська частина реалізована на платформі .NET MAUI для побудови інтерфейсу користувача. Основний акцент було зроблено на створенні простого, інтуїтивно зрозумілого та естетично привабливого дизайну для забезпечення комфортної взаємодії користувача з функціоналом, на основі результатів аналізу інших подібних додатків для конвертації.

Загальні принципи UI/UX-дизайну:

Інтерфейс користувача виконаний у спокійних темнуватих тонах з використанням градієнтного фону, що сприяє зручному використанню в умовах різного освітлення [25]. Основні інтерактивні елементи, такі як кнопки, мають яскраві кольори для полегшення навігації та візуального виділення доступних дій. Структура додатку є багатосторінковою, де користувач з головного екрана переходить до відповідних інтерфейсів для конвертації аудіо або відео файлів [26].

Головна сторінка (MainPage):

Стартовий екран програми відіграє роль візитівки, яка зустрічає користувача, на ній було зосереджено велику увагу, щоб зробити її максимально лаконічною і зрозумілою у використанні. На головну сторінку виведено назву застосунку, короткий, ємний слоган, що пояснює його суть та кнопки навігації до сторінок конвертування. Головна функція екрана – надати вибір типу конвертації через дві розмежовані кнопки: "Відео" та "Аудіо". Для коректного відображення на всіх мобільних пристроях, усі елементи розміщені симетрично, забезпечуючи інтуїтивно зрозумілу навігацію.

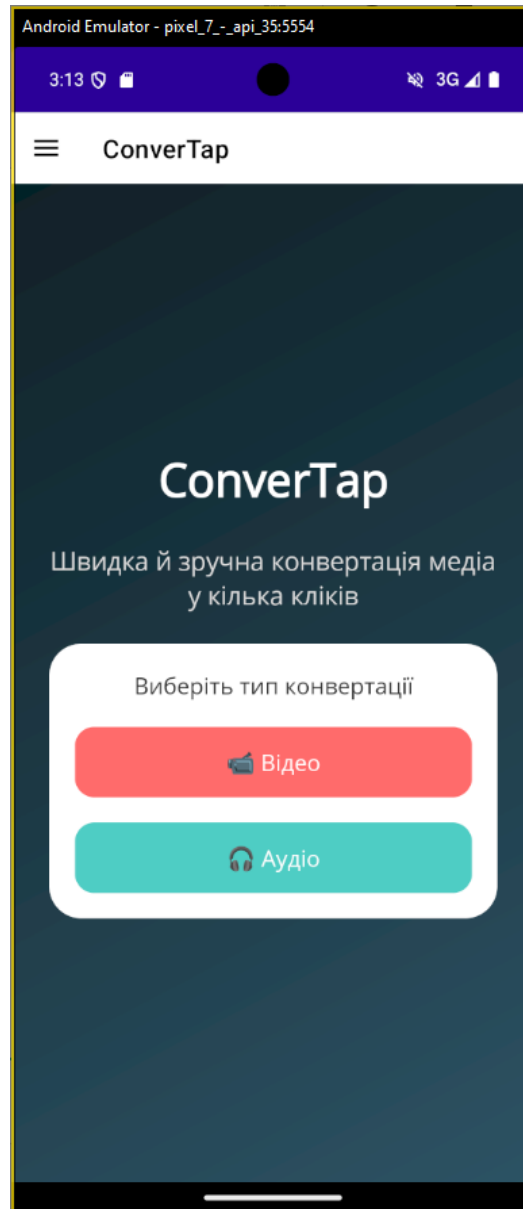


Рис 3.1 Головна сторінка застосунку

Сторінка конвертації відео (VideoConversionPage):

Цей екран призначений для налаштування та запуску процесу конвертації відеофайлів. Користувач починає з натискання кнопки "Вибрати файл", після чого під нею відображається ім'я завантаженого файлу або напис "Файл не вибрано". Далі, за допомогою випадаючих списків, можна обрати цільовий формат відео, необхідну роздільну здатність та бажану якість(бітрейт). Після встановлення всіх параметрів, натискання кнопки "Конвертувати" ініціює процес перетворення, під час якого з'являється

індикатор активності, який візуально відображає хід конвертації. Для повернення на попередній екран передбачена кнопка "Повернутись на головну".

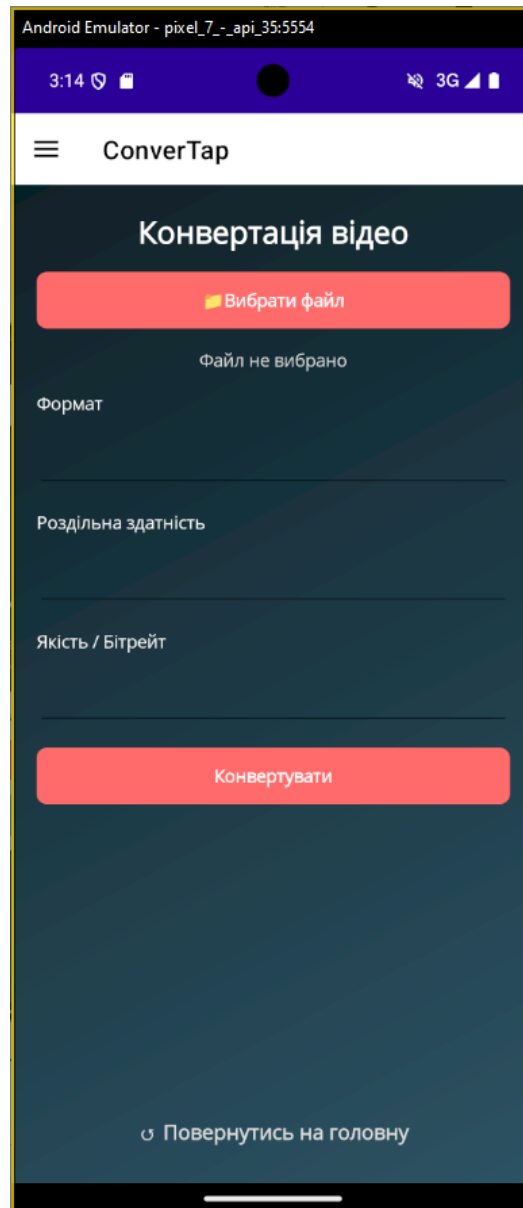


Рис 3.2 Сторінка конвертації відео

Сторінка конвертації аудіо (AudioConversionPage):

Цей екран аналогічний як і на сторінці конвертації відео тільки зі змінами для аудіо, призначений для налаштування та запуску процесу конвертації аудіофайлів. Користувач вибирає свій та налаштовує параметри, формат та бажану якість звуку. Після налаштувань, натискає кнопку

"Конвертувати" та очікує на результат конвертації. Після за потреби натискає кнопку "Повернутись на головну".

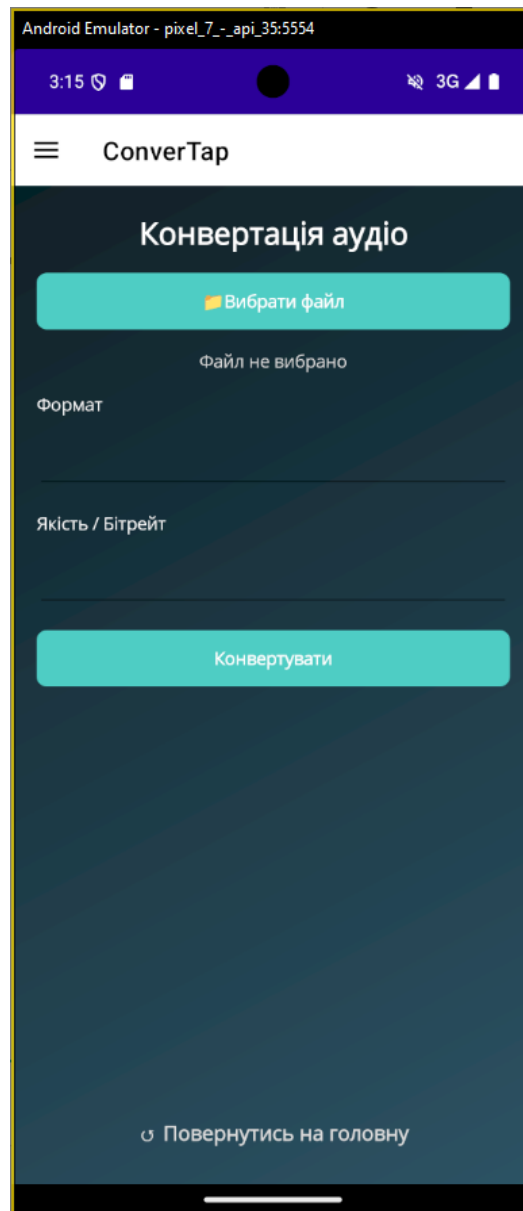


Рис 3.3 Сторінка конвертації аудіо

Логіка інтерфейсу користувача:

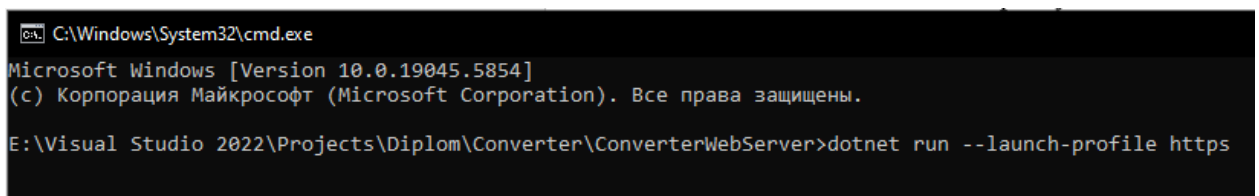
- Навігація реалізована за допомогою .NET MAUI Shell. Перехід між сторінками здійснюється асинхронно через `Shell.Current.GoToAsync("<Сторінка переходу>")`.
- Перед вибором файлу додаток перевіряє наявність дозволу на читання файлів зі сховища (`Permissions.StorageRead`). Якщо дозвіл відсутній, він

запитується у користувача за допомогою `Permissions.RequestAsync<Permissions.StorageRead>()`. Це забезпечує коректний доступ до файлової системи.

- Для вибору файлу використовується `FilePicker.PickAsync()`, що відкриває стандартний системний діалог вибору файлів.
- Після вибору налаштувань конвертації, формується HTTP POST-запит на відповідний ендпоінт сервера (`/convert/video` або `/convert/audio`). Файл та параметри надсилаються на сервер за допомогою `HttpClient` упаковані в `MultipartFormDataContent`.
- Під час надсилання запиту на сервер, під кнопкою конвертації з'являється індикація процесу (`ActivityIndicator`) інформуючи користувача, що операція виконується.
- Після успішного отримання сконвертованого файлу від сервера, він зберігається в папку "Downloads" на Android-пристрої. Шлях до директорії отримується за допомогою `Android.OS.Environment.GetExternalStoragePublicDirectory(Android.OS.Environment.DirectoryDownloads)?.AbsolutePath`.
- Користувачеві виводяться інформаційні повідомлення (`DisplayAlert`) про успішне завершення конвертації та збереження файлу, або про помилки, що могли виникнути.

3.2 Запуск сервера і підключення URL-адреси

ASP.NET Core сервер запускається через консольну команду з налаштуванням профілю запуску протоколу HTTPS для захищеної передачі даних між клієнтом та сервером [27].



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.5854]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
E:\Visual Studio 2022\Projects\Diplom\Converter\ConverterWebServer>dotnet run --launch-profile https
  
```

Рис 3.4 Налаштування запуску сервера

```

C:\Windows\System32\cmd.exe - dotnet run --launch-profile https
Microsoft Windows [Version 10.0.19045.5854]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

E:\Visual Studio 2022\Projects\Diplom\Converter\ConverterWebServer>dotnet run --launch-profile https
Используются параметры запуска из E:\Visual Studio 2022\Projects\Diplom\Converter\ConverterWebServer\Properties\launchSettings.json...
Сборка...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7248
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5062
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: E:\Visual Studio 2022\Projects\Diplom\Converter\ConverterWebServer
  
```

Рис 3.5 Запущений локальний сервер

Після того як сервер було запущено ним вже можна користуватися але тільки на локальному пристрої, щоб зробити адресу публічною і доступною в Інтернеті її потрібно опублікувати за допомогою Ngrok через командний рядок [28].

```

Администратор: Командная строка
Microsoft Windows [Version 10.0.19045.5854]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>ngrok http https://localhost:7248
  
```

Рис 3.6 Виклик Ngrok

```

Администратор: Командная строка - ngrok http https://localhost:7248
ngrok
ngrok is also now your Kubernetes-native ingress: https://ngrok.com/r/k8s

Session Status      online
Account             ymmalyzhenko.fitu21@kubg.edu.ua (Plan: Free)
Version             3.22.1
Region              Europe (eu)
Latency             35ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://1fbd-95-215-223-222.ngrok-free.app -> https://localhost:7248
  
```

Рис 3.7 Створена публічна URL-адреса

Коли всі необхідні підготовчі кроки було зроблено отриману адресу потрібно скопіювати і вставити у програмний код, для подальшої роботи програми з нею.

3.3 Серверна логіка обробки і конвертації медіа

Серверна частина додатку реалізована на платформі ASP.NET Core і відповідає за приймання вхідних файлів від мобільного клієнта, їх обробку за допомогою утиліти FFmpeg та повернення обробленого результату. Сервер містить логіку обробки POST-запитів на ендпоінти /convert/video та /convert/audio [29].

3.3.1 Обробка відеофайлів

При надходженні POST-запиту на ендпоінт /convert/video, сервер виконує наступні кроки:

- Перевірка типу запиту він повинен бути типу multipart/form-data та містити файл.
- Отримання даних запиту, з нього зчитується завантажений відеофайл (IFormFile) та параметри: цільовий формат (format), роздільна здатність (resolution) і бітрейт (bitrate).
- Завантажений користувачем файл тимчасово зберігається на сервері у спеціально створеній директорії (Temp/Uploads/).
- На основі отриманих параметрів для відео файлу генерується команда: `ffmpeg -i {inputVideoPath} -vf scale={resolution} -b:v {bitrate}k {outputVideoPath}`
- Окремим процесом запускається FFmpeg, який виконує згенеровану команду.
- Після завершення роботи FFmpeg, якщо конвертація пройшла успішно, сконвертований відеофайл зберігається у Temp/Outputs/, зчитується та надсилається.
- Сконвертований файл надсилається клієнту і виводить повідомлення з повним шляхом збереженого файлу.
- Після завершення обробки запиту, тимчасові вхідний та вихідний файли видаляються з сервера.

3.3.2 Обробка аудіофайлів

Обробка POST-запитів на ендпоінт `/convert/audio` відбувається аналогічно до відеофайлів, але з урахуванням специфіки аудіо:

- Перевірка типу запиту він повинен бути типу `multipart/form-data` та містити файл.
- Отримання даних запиту, з нього зчитується завантажений відеофайл (`IFormFile`) та параметри: цільовий формат (`format`) і бітрейт (`bitrate`).
- Завантажений користувачем файл тимчасово зберігається на сервері у спеціально створеній директорії (`Temp/Uploads/`).
- На основі отриманих параметрів для аудіо файлу генерується команда: `ffmpeg -i {inputAudioPath} -b:a {bitrate}k {outputAudioPath}`
- Окремим процесом запускається `FFmpeg`, який виконує згенеровану команду.
- Після завершення роботи `FFmpeg`, якщо конвертація пройшла успішно, сконвертований відеофайл зберігається у `Temp/Outputs/`, зчитується та надсилається.
- Сконвертований файл надсилається клієнту і виводить повідомлення з повним шляхом збереженого файлу.
- Після завершення обробки запиту, тимчасові вхідний та вихідний файли видаляються з сервера.

У випадку виникнення помилок на будь-якому етапі (наприклад, відсутність необхідних параметрів, помилка під час виконання `FFmpeg`), сервер повертає клієнту відповідне повідомлення про помилку та HTTP статус помилки.

3.3.3 Інтеграція з FFmpeg та керування файлами

Інтеграція з FFmpeg є ключовою частиною серверної логіки. FFmpeg запускається як зовнішній процес за допомогою класу `System.Diagnostics.Process`. У `ProcessStartInfo` вказується ім'я виконуваного файлу (`ffmpeg`), аргументи командного рядка (сформована команда `ffmpegArgs`) [30]:

```
var    ffmpegArgs    =    $"-i    \"{inputPath}\"    -vf
    scale={resolution} -b:v {bitrate} -y \"{outputPath}\"";
```

А також налаштовується перенаправлення стандартного виводу (`RedirectStandardOutput = true`) та стандартного потоку помилок (`RedirectStandardError = true`). Це дозволяє отримувати інформацію про хід виконання та можливі помилки від FFmpeg. `UseShellExecute` встановлюється в `false`, а `CreateNoWindow` в `true` для запуску процесу у фоновому режимі без створення вікна:

```
var process = new Process
{
    StartInfo = new ProcessStartInfo
    {
        FileName = "ffmpeg",
        Arguments = ffmpegArgs,
        RedirectStandardOutput = true,
        RedirectStandardError = true,
        UseShellExecute = false,
        CreateNoWindow = true
    }
};
process.Start();
```

Для логування виводу сервер зчитує вивід з `StandardError` процесу FFmpeg. Ця інформація виводиться в консоль сервера, для моніторингу всіх процесів виконання конвертації, від POST-запиту до відправки готового файлу користувачу.

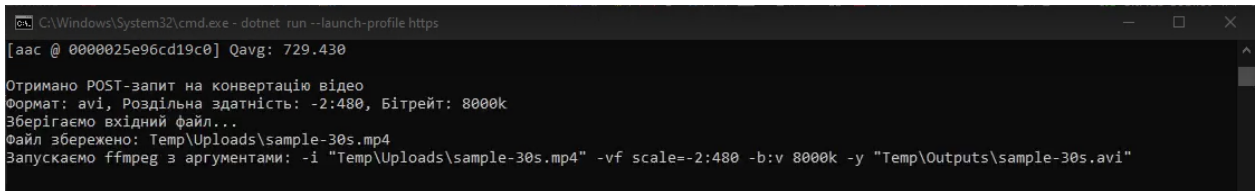


Рис 3.8 Логування процесу конвертації на кожному кроці

Керування тимчасовими файлами:

Вхідні файли, отримані від клієнта, зберігаються у тимчасовій піддиректорії Temp/Uploads.

Вихідні файли, від FFmpeg зберігаються у тимчасовій піддиректорії Temp/Outputs.

Для уникнення накопичення файлів на сервері та забезпечення чистоти середовища, обидва файли (вхідний та вихідний) видаляються після завершення обробки запиту (незалежно від того, успішно він завершився чи з помилкою). Це реалізовано за допомогою блоку `finally` в `try-catch-finally` конструкції:

```
finally
{
    if (File.Exists(inputPath)) File.Delete(inputPath);
    if (File.Exists(outputPath)) File.Delete(outputPath);
}
```

3.3.4 Налаштування та робота зовнішнього доступу через Ngrok

Оскільки мобільний додаток (клієнт) та сервер розроблялися та тестувалися на одному локальному комп'ютері, для забезпечення доступу мобільного клієнта до локально запущеного ASP.NET Core сервера використовувалася утиліта Ngrok. Вона створює безпечний тунель від публічної URL-адреси в Інтернеті до локального порту, на якому працює веб-сервер. Після запуску локального ASP.NET Core сервера (<https://localhost:7248>), через командний рядок запускається Ngrok з командою:

```
ngrok http https://localhost:7248
```

Ngrok генерує унікальну публічну HTTPS-адресу ([https://<унікальний ідентифікатор>.ngrok-free.app](https://<унікальний_ідентифікатор>.ngrok-free.app)). Ця згенерована URL-адреса використовується в мобільному клієнті як базовий URL для надсилання запитів на сервер. Такий підхід дозволяє проводити повноцінне тестування клієнт-серверної взаємодії на реальному пристрої без необхідності розгортання сервера на публічному хостингу на ранніх етапах розробки. Ngrok також надає веб-інтерфейс для моніторингу HTTP-трафіку, що проходить через тунель [31].

```

Администратор: Командная строка - ngrok http https://localhost:7248
ngrok (Ctrl+C to quit)
ngrok is also now your Kubernetes-native ingress: https://ngrok.com/r/k8s
Session Status      online
Account             ymmalyzhenko.fitu21@kubg.edu.ua (Plan: Free)
Version             3.22.1
Region              Europe (eu)
Latency             35ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://1fbd-95-215-222.ngrok-free.app -> https://localhost:7248

Connections
  ttl    opn    rt1    rt5    p50    p90
    2     1     0.01   0.01   103.24 112.22

HTTP Requests
-----
22:14:02.488 EEST POST /convert/video      200 OK
22:11:52.647 EEST POST /convert/audio      200 OK
22:11:30.050 EEST POST /convert/video      200 OK
  
```

Рис 3.9 Робота Ngrok з результатами звернення клієнта на сервер

3.4 Перевірка функціональності та результати тестування

Тестування розробленого програмного забезпечення проводилося на кількох рівнях для перевірки коректності роботи як клієнтської, так і серверної частин, а також їхньої взаємодії [32].

1. Тестування клієнтської частини:

Перевірка коректності відображення всіх елементів на різних екранах, роботи навігації, випадаючих списків, кнопок;

Тестування функціоналу вибору аудіо- та відеофайлів, коректності запиту та обробки дозволів на доступ до сховища;

Перевірка правильності формування та надсилання запитів на сервер з обраними файлами та параметрами;

Тестування приймання відповіді від сервера та коректного збереження сконвертованих файлів у папку "Downloads";

Перевірка відображення повідомлень про помилки, що надходять від сервера або виникають на клієнті.

Пристрої для тестування:

- Android-емулятор (Pixel 7 – API 35, Android 15.0)
- Android-пристрій (Meizu note 8 – API 27, Android 8.1)

2. Тестування серверної частини:

Перевірка коректності приймання файлів та параметрів на ендпоінтах /convert/video та /convert/audio;

Тестування правильності формування команд для FFmpeg та його запуску, обробки виводу;

Перевірка відповідності сконвертованих файлів заданим параметрам (формат, роздільна здатність, бітрейт);

Перевірка повернення коректних HTTP статусів та повідомлень про помилки;

Аналіз логів сервера для виявлення проблем та моніторингу процесу.

3. Інтеграційне тестування (клієнт-сервер):

Перевірка всього циклу: вибір файлу на клієнті > надсилання на сервер > конвертація на сервері > отримання результату клієнтом > збереження файлу.

Тестування з різними типами та розмірами файлів.

Результати тестування:

Розроблений мобільний додаток працює коректно, як на емуляторі так і на фізичному пристрої, інтерфейс додатку виглядає правильно на обох пристроях без артефактів, або ще якихось нарікань, окремо від вручну

налаштованих параметрів (розмірів кнопок, відступів, заокруглень, тощо) сама платформа .NET MAUI має свої внутрішні налаштування, щоб будь-який додаток виглядав правильно в рамках розміру екрану. Кожна кнопка прив'язана до своєї сторінки і правильно перенаправляє користувача. На сторінках конвертації будь то відео чи аудіо конвертація, все працює коректно, успішно обираються файли, правильно відображаються випадаючі списки і вибираються параметри, при натисканні кнопки конвертації дані без помилок відправляються на сервер, а на екрані з'являється іконка виконання процесу конвертації. Серверна частина правильно приймає запити, викликає FFmpeg для конвертації, обробляє їх та повертає результат зберігаючи на пристрій користувача.

Можливості для подальшого розширення:

1. Додавання підтримки інших популярних або специфічних аудіо- та відеоформатів і кодеків.
2. Можливість пакетної конвертації (обробка кількох файлів одночасно) [33], додавання опцій для обрізки відео чи аудіо або зміни співвідношення сторін.
3. Деталізація інтерфейсу додатку для більшої зручності користування, можливість перегляду історії конвертацій та налаштування теми оформлення інтерфейсу.
4. Оптимізація продуктивності з можливістю паралельної обробки запитів на сервері та оптимізація додатку для пришвидшення конвертації без значної втрати якості [34].
5. Повноцінна реалізація та тестування додатку для інших платформ, підтримуваних .NET MAUI (iOS, Windows) [35].
6. Впровадження механізмів автентифікації та авторизації, для розширення функціоналу (наприклад, збереження файлів у хмарі користувача).
7. Можливість завантаження файлів з хмарного сховища (наприклад Google Drive) та збереження результатів конвертації туди ж.

Висновки до розділу 3

Було успішно реалізовано та протестовано мобільний застосунок "ConverTap" для конвертації медіафайлів. У повній мірі був застосований стек обраних технологій та архітектурних рішень при яких, клієнтська частина, забезпечує зручний інтерфейс для користувачів Android, дозволяючи легко обирати файли та налаштовувати параметри конвертації, а серверна частина ефективно обробляє запити та виконує конвертацію.

Серед основних досягнень реалізації варто відзначити створення повнофункціонального мобільного додатка для Android, що виконує конвертацію відео- та аудіофайлів, з підтримкою популярних форматів, зокрема MP4, MOV, AVI для відео та MP3, AAC, WAV для аудіо, з можливістю налаштування ключових параметрів, таких як роздільна здатність і бітрейт. Також забезпечено стабільну клієнт-серверну взаємодію для передачі файлів та обробки запитів. Для виконання якісної та гнучкої конвертації на сервері було інтегровано FFmpeg, та реалізовано збереження результатів у стандартну директорію "Downloads" на Android-пристрої.

Загалом, розроблений програмний комплекс є масштабованим та готовим до подальшого розвитку, що дозволить розширити його функціональні можливості та покращити користувацький досвід.

ВИСНОВКИ

У рамках виконання кваліфікаційної роботи було успішно розроблено та протестовано мобільний додаток для конвертації файлів аудіо та відео форматів. Основною метою роботи було створення зручного та ефективного інструменту, який би дозволяв користувачам легко оперувати своїми відео або аудіо файлами, змінюючи їх формати та налаштовуючи параметри конвертації і це все в одному додатку на їх мобільному пристрої.

Провівши аналіз існуючих рішень на ринку мобільних конвертерів, було виявлено потребу в додатку, який би поєднував простий інтерфейс з гнучкими налаштуваннями конвертації та високою продуктивністю, не перевантажуючи при цьому мобільний пристрій. Для досягнення цієї мети було обрано сучасний стек технологій: клієнтська частина розроблена на сучасній платформі .NET MAUI, що забезпечило створення нативного інтерфейсу для Android та заклало потенціал для майбутньої кросплатформності. Серверна частина реалізована за допомогою фреймворку ASP.NET Core, яка обробляє запити від клієнта та керує процесом конвертації за допомогою потужної утиліти FFmpeg.

У ході проектування було визначено та реалізовано ключові функціональні та технічні вимоги. Клієнтський додаток надає користувачеві можливість обирати локальні файли та задавати параметри конвертації. Після налаштування, файл та параметри надсилаються на сервер, який виконує конвертацію і повертає готовий файл клієнту, зберігаючи у папку "Downloads".

Реалізація програмного забезпечення включала розробку багатосторінкового інтерфейсу користувача, логіку вибору файлів, обробку дозволів доступу до сховища, формування та надсилання HTTP-запитів, а також обробку відповідей від сервера та збереження файлів. На сервері було реалізовано ендпоінти для приймання відео- та аудіофайлів, формування команд та запуск процесу конвертації, а також

покрокове логування кожного процесу для моніторингу несправностей, якщо вони виникали.

Тестування функціональності проводилося на Android-емуляторі та фізичному пристрої, і підтвердило коректну роботу всіх основних компонентів системи. Додаток продемонстрував стабільну роботу та відповідність заявленим вимогам.

Таким чином, розроблений мобільний додаток "ConverTap" є повноцінним програмним продуктом, що вирішує поставлене завдання конвертування медіафайлів. Програма має значний потенціал для подальшого розвитку, включаючи розширення списку підтримуваних форматів, додавання нових функцій, покращення інтерфейсу користувача, оптимізацію продуктивності, а також розширення на інші платформи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Засоби перетворення аудіо- й відеоформатів [Електронний ресурс] – https://alextexnok.blogspot.com/p/blog-page_852.html
2. Як конвертувати відеофайли. [Електронний ресурс] – <https://experience.dropbox.com/uk-ua/resources/convert-video-files>
3. Програми для конвертації відео [Електронний ресурс] – <https://uk.soringpcrepair.com/programs-for-converting-video/>
4. What is a video file format? [Електронний ресурс] – <https://www.adobe.com/creativecloud/file-types/video.html>
5. Формати аудіо- та відеофайлів [Електронний ресурс] – <https://ua5.org/information/2570-formaty-audio-ta-videofajliv.html>
6. Конвертування аудіо-, відеоінформації [Електронний ресурс] – https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B0%D1%83%D0%B4%D1%96%D0%BE-_%D0%B2%D1%96%D0%B4%D0%B5%D0%BE%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97
7. Засоби перетворення аудіо- й відеоформатів [Електронний ресурс] – <https://ua5.org/ofpac/2156-zasoby-peretvorennya-audio-j-videoformativ.html>
8. ffmpeg Documentation [Електронний ресурс] – <https://ffmpeg.org/ffmpeg.html>
9. Recipes in FFmpeg [Електронний ресурс] – <https://ottverse.com/recipes-in-ffmpeg/>
10. How to Integrate Video Conversion into Your App: Tips and Best Practices [Електронний ресурс] – <https://www.devprojournal.com/technology-trends/how-to-integrate-video-conversion-into-your-app-tips-and-best-practices/>

11. C#.NET Tutorials For Beginners and Professionals [Електронний ресурс] – <https://dotnettutorials.net/course/csharp-dot-net-tutorials/>
12. Introduction to .NET [Електронний ресурс] – <https://learn.microsoft.com/en-us/dotnet/core/introduction>
13. What is .NET MAUI? [Електронний ресурс] – <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-9.0>
14. .NET MAUI Tutorial – Build your first multi-platform app in C# [Електронний ресурс] – <https://dotnet.microsoft.com/en-us/learn/maui/first-app-tutorial/device-setup>
15. Додаток на .NET MAUI [Електронний ресурс] – <https://abitap.com/2-1-dodatok-na-net-maui/>
16. Upload files in ASP.NET Core [Електронний ресурс] – <https://learn.microsoft.com/en-us/aspnet/core/mvc/models/file-uploads>
17. Minimal APIs quick reference [Електронний ресурс] – <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/minimal-apis>
18. Converting audio and video files [Електронний ресурс] – <https://medium.com/@ioannidis.au/converting-audio-and-video-files-9a7c0adad804>
19. A quick guide to using FFmpeg to convert media files [Електронний ресурс] – <https://opensource.com/article/17/6/ffmpeg-convert-media-file-formats>
20. Ngrok documentation [Електронний ресурс] – <https://ngrok.com/docs>
21. Putting your APIs online shouldn't be hard [Електронний ресурс] – <https://ngrok.com/>
22. Клієнт-серверна архітектура [Електронний ресурс] – <https://training.gatestlab.com/blog/technical-articles/client-server-architecture/>

23. Взаємодія клієнт-сервер [Електронний ресурс] –
<https://ua5.org/technol/2094-vzayemodiya-kliiyent-server.html>
24. Клієнт-серверна архітектура та ролі серверів. [Електронний ресурс] –
<https://medium.com/@IvanZmerzlyi/клієнт-серверна-архітектура-та-ролі-серверів-9893d8048229>
25. Create a .NET MAUI app [Електронний ресурс] –
<https://learn.microsoft.com/en-us/dotnet/maui/tutorials/notes-app/?view=net-maui-9.0>
26. Розробка мобільних додатків від А до Я [Електронний ресурс] –
<https://dan-it.com.ua/uk/blog/rozrobka-mobilnih-dodatkov-vid-a-do-ja-povni-j-gajd/>
27. App startup in ASP.NET Core [Електронний ресурс] –
<https://learn.microsoft.com/uk-ua/aspnet/core/fundamentals/startup?view=aspnetcore-9.0>
28. Ngrok через командний рядок [Електронний ресурс] –
<https://rubydevelopers.org/t/ngrok/436>
29. Routing in ASP.NET Core [Електронний ресурс] –
<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-9.0>
30. Конвертація медіафайлів за допомогою FFmpeg [Електронний ресурс] –
<https://ivaniura.org.ua/soft/process-media-with-ffmpeg>
31. Using ngrok to get a public HTTPS address for a local server already serving HTTPS [Електронний ресурс] –
<https://camerondwyer.com/2019/09/23/using-ngrok-to-get-a-public-https-address-for-a-local-server-already-serving-https-for-free/>
32. Android emulator [Електронний ресурс] –
<https://learn.microsoft.com/en-us/dotnet/maui/android/emulator/?view=net-maui-9.0>
33. Batch convert videos using FFmpeg [Електронний ресурс] –
<https://shotstack.io/learn/ffmpeg-batch-convert/>

- 34.How to convert video to audio files without any quality loss (free & paid)
[Електронний ресурс] –
<https://techwithmuchiri.com/convert-video-to-audio-without-quality-loss/>
- 35..NET MAUI: The Future of Cross-Platform Development [Електронний ресурс] –
<https://medium.com/@BlueflameLabs/net-maui-the-future-of-cross-platform-development-0ca7a3461947>
- 36.В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко.Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.
- 37.Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці:моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка,2021. — 332 с.