

Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

Розробка захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками

Виконав: студент 4 курсу, гр. 1КІТС-206
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напрямку підготовки,
спеціальності)

Адаменко І. О.

(прізвище та ініціали)

Керівник: к.т.н., доц., зав. каф. МБІС

Карпінєць В. В.

(прізвище та ініціали)

« 6 » сервіс 2024 р.

Рецензент: к.т.н., доц., каф. ОТ

Кожем'яко А. В.

(прізвище та ініціали)

« 6 » сервіс 2024 р.

Допущено до захисту

Голова секції УБ кафедри МБІС

д.т.н., проф. Яремчук Ю.Є.

« 6 » сервіс 2024р.

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти I-й (бакалаврський)

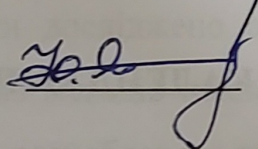
Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ

Голова секції УБ, кафедра МБІС



д.т.н., проф. Яремчук Ю.Є.

“ 22 ” березня 2024 р.

ЗАВДАННЯ

на бакалаврську дипломну роботу студенту

Адаменку Іллі Олександровичу

(прізвище, ім'я, по-батькові)

1. Тема роботи Розробка захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками.

Керівник роботи Карпинець Василь Васильович к.т.н., доц., зав. каф. МБІС
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “11” березня 2024 року № 80

2. Термін подання студентом роботи 5 червня 2024р

3. Вихідні дані до роботи Електронні джерела, наукові статті по темі, що описують тему бакалаврської дипломної роботи.

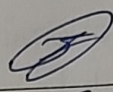
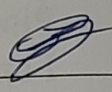
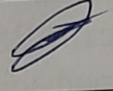
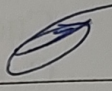
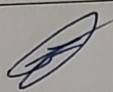
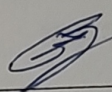
4. Зміст текстової частини:

Для досягнення мети було поставлено наступні задачі: дослідити актуальність вибраної теми; здійснити аналіз стеганографічних методів накладання цифрових водяних знаків; проаналізувати принцип роботи цих методів; розробити захищений веб-сервіс; реалізувати онлайн сервіс; протестувати розроблений сервіс;

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Діаграма алгоритму роботи користувача із розробленим онлайн сервісом, діаграма прецедентів, схема авторизації користувача, діаграма розробленої бази даних, діаграма алгоритму фазового кодування.

6. Консультанти розділів роботи

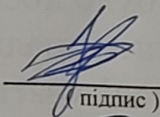
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Карпинець В. В. к.т.н., доц., зав. Каф. МБІС		
Розділ II	Карпинець В. В. к.т.н., доц., зав. Каф. МБІС		
Розділ III	Карпинець В. В. к.т.н., доц., зав. Каф. МБІС		

7. Дата видачі завдання 22 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

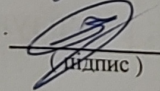
№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	27.03.2024	10.04.2024	Виконано
2	Аналіз методів розв'язання задачі	12.04.2024	20.04.2024	Виконано
3	Постановка задач розробки веб-сервісу	21.04.2024	28.04.2024	Виконано
4	Дослідження інформаційних джерел	29.04.2024	01.05.2024	Виконано
5	Аналіз варіантів роботи модуля	02.05.2024	07.05.2024	Виконано
6	Розробка алгоритму роботи	08.05.2024	15.05.2024	Виконано
7	Реалізація програмного модулю	15.05.2024	20.05.2024	Виконано
8	Тестування програмного модулю	20.05.2024	21.05.2024	Виконано
9	Оформлення матеріалів до захисту БДР	21.05.2024	26.05.2024	Виконано

Студент


(підпис)

Адамченко І.О.
(прізвище та ініціали)

Керівник роботи


(підпис)

Карпинець В.В.
(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 111 сторінки формату А4, на яких є 40 рисунків, список використаних джерел містить 35 найменування.

Метою роботи є розробка захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками з використанням бази даних MySQL, фреймворку PHP Laravel Passport та RedbeanPHP.

У першому розділі роботи досліджено актуальність розробки захищеного онлайн серісу, здійснено аналіз методів накладання ЦВЗ, розглянуто проблеми аналогів та визначено переваги розробленого продукту.

У другому розділі роботи здійснено розробку структури та функціоналу додатку, що передбачає надійну авторизацію протоколами Auth2.0; застосування PDO, проектування захисту від SQL-ін'єкцій, XSS-атак та шифрування даних; розроблена модель та описана структура розробленої бази даних; здійснено вибір та обґрунтування обраних засобів програмування для реалізації розробки.

У третьому розділі роботи здійснено розробку графічного інтерфейсу програмного засобу, описано ключові елементи програмної реалізації, описано інструкцію користувача для роботи з ресурсом, здійснено тестування розробленого веб-сервісу.

Ключові слова: онлайн сервіс, стійкі цифрові водяні знаки, захист даних, SQL-ін'єкція, XSS-атаки, HTTPS, SSL, MySQL, Laravel Passport. OTP.

ANNOTATION

The bachelor's thesis consists of 111 A4 pages, including 40 figures, and the list of references contains 35 sources.

The aim of the thesis is to develop a secure online audio sales service with copyright protection using robust digital watermarks, employing MySQL database, PHP Laravel Passport framework, and RedbeanPHP.

In the first chapter, the relevance of developing a secure online service is explored, methods of applying digital watermarks are analyzed, problems of existing solutions are considered, and the advantages of the developed product are identified.

In the second chapter, the structure and functionality of the application are developed, including reliable authorization with Auth2.0 protocols, the use of PDO, protection against SQL injection, XSS attacks, and data encryption; the model and structure of the developed database are described; the chosen programming tools for the implementation of the development are selected and justified.

In the third chapter, the graphical interface of the software tool is developed, key elements of the software implementation are described, a user manual for working with the resource is provided, and the developed web service is tested.

Keywords: online service, robust digital watermarks, data protection, SQL injection, XSS attacks, HTTPS, SSL, MySQL, Laravel Passport, OTP.

ЗМІСТ

ВСТУП.....	4
1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Актуальність дослідження обраної галузі.....	6
1.2 Аналіз актуальності захищення аудіозаписів стійким цифровими водяними знаками	10
1.3 Існуючі стеганографічні методи накладання ЦВЗ	17
1.4 Огляд існуючих сервісів із захистом авторського права.....	24
1.5 Висновки та постановка задач.....	26
2 ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ЗАХИСТУ АУДІОФАЙЛІВ.....	28
2.1 Розробка архітектури сервісу захищеного онлайн сервісу продажу аудіозаписів.....	28
2.2 Вибір та інтеграція модулів захисту для веб-сервісу захисту аудіофайлів	37
2.3 Розробка модулю накладання стійкого цифрового водяного знаку	45
2.4 Вибір середовища розробки	50
2.5 Висновки до розділу 2.....	54
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	55
3.1 Розробка користувацького інтерфейсу	55
3.2 Програмна реалізація алгоритму накладання ЦВЗ та модулів захисту	60
3.3 Тестування захищеного веб-сервісу	68
3.4 Висновки до розділу 3.....	78
ВИСНОВОК.....	79
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ.....	84
Додаток А. Технічне завдання	85
Додаток Б. Лістинг функції накладання ЦВЗ.....	89
Додаток В. Лістинг файлу index.php	91
Додаток Г. Лістинг файлів controllers	94
Додаток Д. Ілюстративний матеріал	100

Додаток Е. Протокол перевірки на антиплагіат	107
--	-----

ВСТУП

Актуальність. В музичній індустрії, забезпечення безпеки інформації є ключовою задачею для захисту конфіденційності музичних файлів та звукових записів у цифровому середовищі. Особливо це стосується аудіо зразків - цінного ресурсу для створення музики, звукового дизайну та аудіо ефектів. Проте, існуючі сервіси з продажу аудіо зразків залишаються вразливими перед можливими атаками злоумисників через недостатній рівень захисту та шифрування.

Поточна система ліцензування музичних творів вимагає значних витрат часу та спеціалізованих правових знань для ефективного захисту авторських прав. Використання водяного знаку дозволяє автоматично захистити ваш файл без потреби в вашому особистому втручанні, забезпечуючи надійність захисту.

Ця проблема стала стимулом для дослідження, спрямованого на розробку імплементації інтернет-магазину аудіо зразків, який використовує надійне шифрування для забезпечення безпеки даних користувачів. Актуальність цього дослідження підкреслюється не лише потребою в захисті авторських прав та інтелектуальної власності, а й загальною необхідністю у підвищенні рівня кібербезпеки в музичній індустрії.

У рамках дослідження буде проведено аналіз існуючих аналогів сервісів з продажу аудіо зразків, виокремлені їх переваги та недоліки, зокрема, відсутність ефективного шифрування та енергозатратність.

Новизна цієї роботи полягає у створенні інноваційного веб-сервісу для захисту аудіофайлів, який об'єднує переваги існуючих рішень та впроваджує передові технології для забезпечення високого рівня безпеки та захисту авторських прав. Зокрема, використовуються сучасні методи шифрування даних користувачів та безпечна авторизація за протоколом OAuth 2.0, що забезпечує захист від несанкціонованого доступу. Крім того, розроблено та впроваджено удосконалений механізм водяних знаків для захисту аудіофайлів, який забезпечує стійкість до видалення або зміни водяного знаку, використовуючи стеганографічні методи для накладання цифрових водяних знаків (ЦВЗ), що робить їх невидимими для слухача

та стійкими до атак. Проектування архітектури сервісу дозволяє зручно продавати та захищати аудіозаписи в онлайн-режимі, а інтуїтивно зрозумілий користувацький інтерфейс полегшує роботу користувачів із сервісом. Комплексний підхід до захисту аудіофайлів, що включає шифрування даних, безпечну авторизацію та вдосконалені методи накладання водяних знаків, значно підвищує рівень захисту аудіофайлів та сприяє розвитку захисту інтелектуальної власності у цифровому середовищі. Новий сервіс буде включати в себе інтуїтивно зрозумілий і легко доступний інтерфейс для користувачів.

В дослідженні будуть розглянуті технічні аспекти реалізації цього сервісу, зокрема вибір мови програмування, розробка архітектури сервісу та теоретична розробка модулю шифрування аудіо зразків та вбудовування водяних знаків в них.

Метою роботи є створення імплементації інтернет-магазину аудіо зразків, який не лише надасть користувачам широкий вибір, а й забезпечить безпеку їхніх даних завдяки застосуванню криптографічних та стеганографічних методів.

Предметом дослідження є розробка і реалізація інтернет-магазину аудіо зразків, спрямованого на задоволення потреб у цифрових матеріалах для різноманітних творчих та професійних проектів. Основна увага приділяється аспектам забезпечення безпеки даних користувачів шляхом використання стійкого водяного знаку.

Об'єктом дослідження є процес зберігання та обміну даними в інтернет-магазині аудіо зразків з механізмом захисту конфіденційності, цілісності та доступності цих даних під час їх передачі через мережу Інтернет. Важливою частиною дослідження є інтеграція стеганографічних методів в процес обробки та зберігання даних, а також аналіз ефективності цих методів у контексті забезпечення безпеки даних користувачів.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

В даному розділі ми проведемо аналіз предметної області, спрямований на визначення актуальності дослідження та розробки захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права, а також проаналізуємо сучасні аналоги таких систем. Паралельно буде розглянуто та проаналізовано технічні аспекти, правовий контекст і сфера аудіо продажу для отримання більш повного уявлення щодо актуальності проблеми.

1.1 Актуальність дослідження обраної галузі

При аналізі предметної області даного проекту необхідно розглянути сучасний стан ринку продажу аудіо. Також важливо вивчити потреби та побажання покупців у цій галузі, включаючи їхні вимоги до якості продукту, сервісу та захисту авторських прав. Усунення технічних перешкод і недоліків може стати ключем до успіху проекту, тому варто детально розглянути технічні аспекти, зокрема платформи для продаж та звукові формати. Крім цього, необхідно врахувати правовий контекст, зокрема використання стійких водяних знаків як підтвердження авторських прав. Це дозволяє захистити інтелектуальну власність та запобігти незаконному використанню аудіозаписів.

Предметною областю даної роботи є створення інтернет-магазину, спеціалізованого на продажі аудіозаписів для різноманітних цілей, таких як використання в музичних творах, аудіокнигах та рекламних матеріалах. Аудіозаписи у цьому контексті виступають як основний матеріал для створення звукового супроводу у різних сферах діяльності.

Метою проекту є розробка інтернет-магазину, що забезпечить користувачів зручним та надійним середовищем для продажу та захисту аудіофайлів.

Проект спрямований на створення інтернет-магазину, який забезпечить користувачів доступом до різноманітного асортименту аудіозаписів, що відповідає різноманітним творчим потребам:

- У сфері музичної індустрії, аудіозаписи використовуються для створення музичних композицій, альбомів та звукових доріжок для різних медіа проектів.

– У сфері електронних ігор, аудіозаписи використовуються для створення атмосфери, звукових ефектів та музичного супроводу, що додає глибини та імерсії гравцям.

– У галузі аудіокниг та подкастів, аудіозаписи використовуються для читання або розповіді історій, навчальних матеріалів та розважальних програм.

– У галузі кіно та відео виробництва, аудіозаписи використовуються для створення звукового супроводу та музичного супроводу для фільмів, рекламних роликів та інших візуальних проектів.

Ринок аудіозаписів зазнає значного розвитку та трансформації на сучасному етапі завдяки стрімінговим платформам, таким як Spotify, Apple Music та Amazon Music, які залишаються основними гравцями. Ці платформи пропонують масштабний каталог аудіозаписів, персоналізовані рекомендації та зручний доступ на різних пристроях. Завдяки цьому сегмент ринку постійно росте, привертаючи нових користувачів та забезпечуючи стабільний дохід для музикантів.

Також компанії піклуються і про розвиток захисту авторських прав за допомогою технічних засобів це включає в себе впровадження передових технологій шифрування, цифрових підписів та інших методів захисту контенту від несанкціонованого доступу та копіювання. Розробка та вдосконалення таких технічних засобів є ключовими для забезпечення ефективного захисту авторських прав у цифровому середовищі [1].

Потреба в якості та захисті авторських прав є важливою для авторів аудіозаписів у сучасному цифровому світі. Якість є ключовим фактором, який впливає на сприйняття твору користувачами та його успіх на ринку. Висока якість звуку може забезпечити більшу привабливість твору, його кращу розпізнаваність та сприяти кращому продажу аудіо. Захист авторських прав також є важливою аспектом для авторів, оскільки він забезпечує їхню власність над твором та відшкодовує їх за використання твору без дозволу. Один із методів захисту авторських прав - використання стійкого водяного знака.

Цей метод є ефективним, оскільки стійкий водяний знак надзвичайно складно видалити з аудіозапису без його пошкодження. Крім того, він майже не впливає на якість аудіо (рис. 1.1) [2].

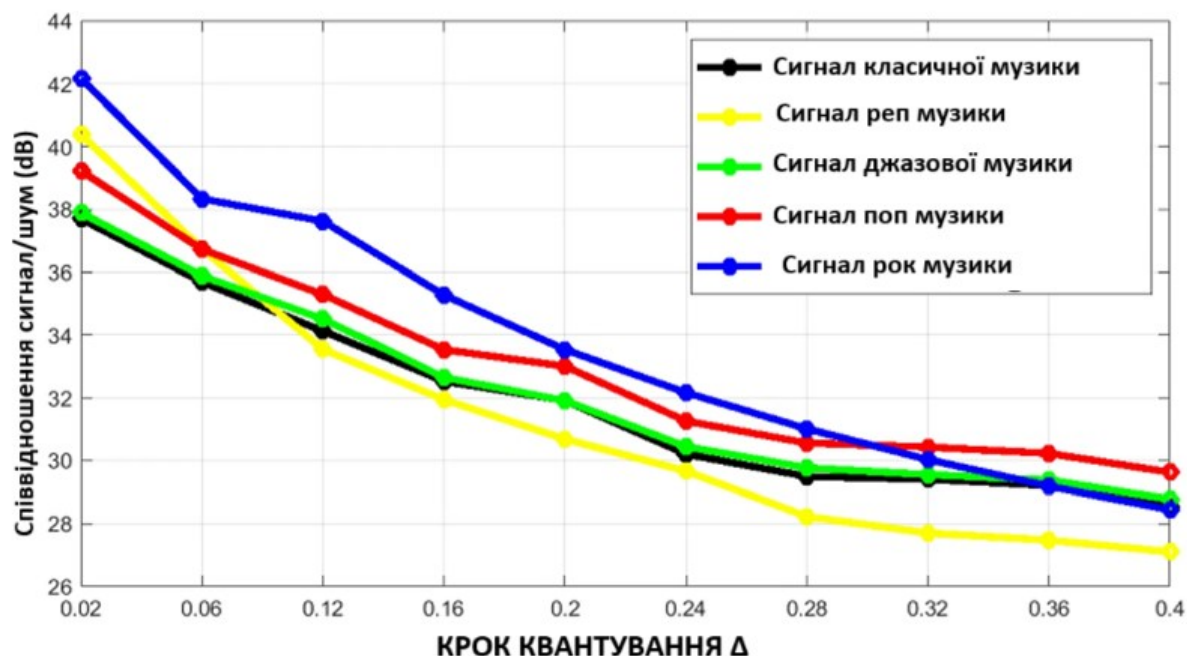


Рисунок 1.1 – Непомітність для різних аудіосигналів із водяними знаками відносно кроку квантування Δ (за [2]).

Говорячи про технічні аспекти треба зауважити що аудіозаписи представлені у різних форматах файлів, які визначають їхню якість та розмір. Найпоширенішими форматами є MP3, WAVE, FLAC тощо. Кожен з цих форматів має свої особливості: наприклад, MP3 займає мало пам'яті, що робить його ідеальним для потокового відтворення, FLAC, з іншого боку, пропонує стиснення без втрат якості звуку, що робить його популярним серед аудіофілів.

В даній роботі буде розглядатись формат WAVE через його переваги, а також популярність серед авторів аудіо.

Формат WAVE (Waveform Audio File Format) є одним з найпоширеніших форматів аудіозаписів, який відомий своєю безстислою структурою даних. WAVE зберігає аудіодані у некомпресованому, без втрат якості форматі, що робить його ідеальним аудіо контейнером для приховування інформації (рис 1.2) [3,4].

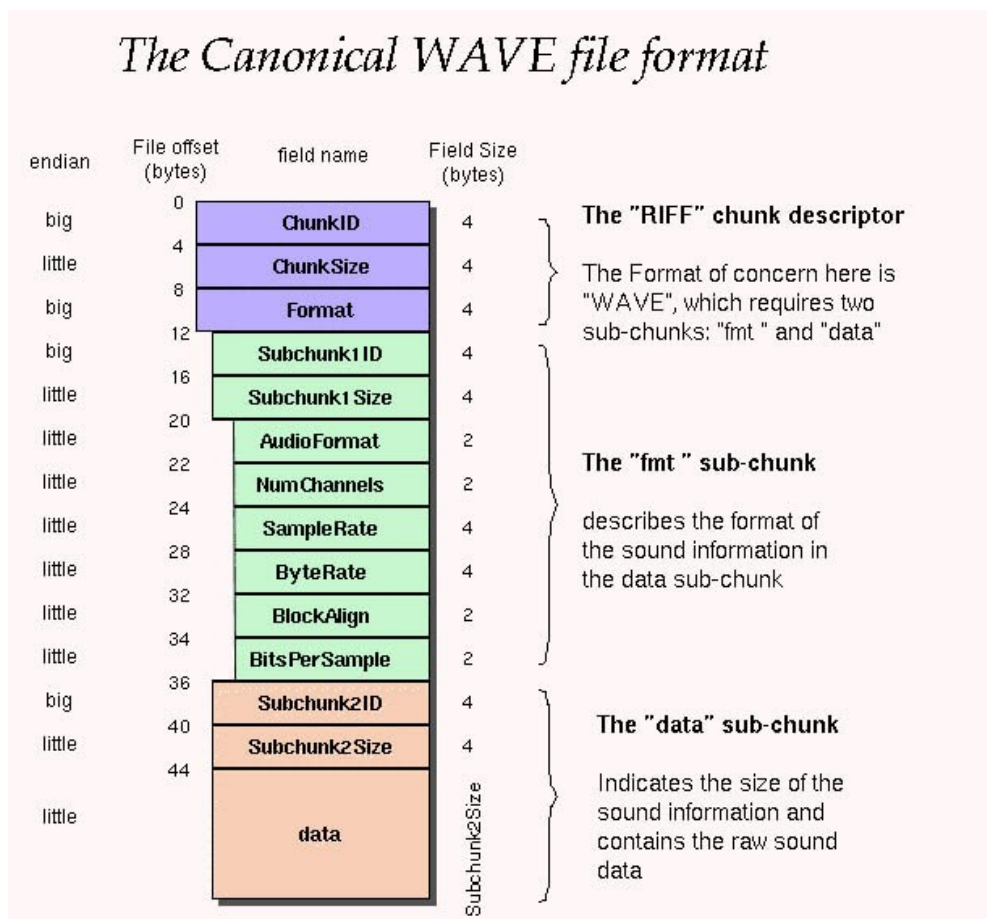


Рисунок 1.2 – Структура аудіофайлу формату WAVE

Цей формат підтримує якісну звукову інформацію з високою частотою дискретизації та бітової глибиною, що дозволяє зберігати звук в найближчому вигляді до оригіналу. WAVE часто використовується у професійному аудіо- та музичному виробництві, де важлива максимальна якість звуку.

Одне з основних завдань цієї роботи є захист авторських прав. Україна має свою законодавчу базу, що регулює авторські права на аудіозаписи та захист цих прав у цифровому середовищі. Основним правовим актом, який регулює авторські права в Україні, є Закон України "Про авторське право та суміжні права". Цей закон визначає права авторів та власників прав на аудіозаписи та встановлює механізми їх захисту [5].

Згідно з цим законом, авторське право на аудіозапис виникає автоматично з моменту створення твору і триває протягом життя автора та ще 70 років після його смерті. Після цього воно може переходити до спадкоємців або інших правонаступників.

Щодо захисту цих прав у цифровому середовищі, Україна впроваджує міжнародні стандарти та домашні закони. Зокрема, використання технологій управління цифровими правами (DRM) широко застосовується для захисту авторських прав на аудіозаписи в Інтернеті та цифрових музичних магазинах [6].

У разі порушення авторських прав на аудіозаписи в Україні існують механізми правового захисту, які включають в себе цивільні, кримінальні та адміністративні санкції. Це може включати в себе судові позови, штрафи та навіть кримінальне переслідування для порушників авторських прав.

Міжнародні стандарти та конвенції є ключовими у визначенні прав та захисту авторських прав на аудіозаписи в міжнародному масштабі. Наприклад, Бернська конвенція, прийнята у 1886 році, встановлює мінімальний стандарт захисту авторських прав для країн-учасниць. Згідно з цією конвенцією, авторське право на аудіозапис виникає автоматично з моменту створення твору і триває протягом життя автора плюс ще 50 або 70 років після його смерті, залежно від національного законодавства.

Крім того, Україна є учасницею Світової організації торгівлі (WTO) і зобов'язана виконувати положення Трипольської угоди з аспектів торговельних прав інтелектуальної власності (TRIPS). TRIPS встановлює стандарти для захисту авторських прав на аудіозаписи та надає механізми для їх захисту, включаючи правові засоби для боротьби з порушеннями авторських прав [7].

Отже, міжнародні стандарти та конвенції відіграють важливу роль у створенні міжнародного правового контексту для захисту авторських прав на аудіозаписи та забезпечення їх виконання у всьому світі.

1.2 Аналіз актуальності захищення аудіозаписів стійким цифровими водяними знаками

Захист аудіозаписів стійкими цифровими водяними знаками є важливим елементом в забезпеченні безпеки та ідентифікації цифрових аудіоданих. Ці водяні знаки, як правило, складаються з унікального шифрованого коду, який

вбудовується безпосередньо в аудіофайл. Вони дозволяють визначати автора, дату створення, джерело походження аудіоматеріалу та інші важливі метадані.

Стійкість цифрових водяних знаків полягає у їхній надійності та стійкості до різних видів обробки аудіофайлу, таких як стискання, фільтрація або зміна формату. Це робить їх ефективними засобами виявлення піратства або незаконного використання аудіоматеріалів. Такі водяні знаки дозволяють не лише захищати авторські права на аудіозаписи, але й забезпечувати їхню аутентичність та цілісність. Вони є необхідним інструментом для музикантів, продюсерів, радіостанцій, аудіокниг та інших власників аудіовмісту для визначення авторства, виявлення підробок та забезпечення прав на інтелектуальну власність.

Аудіозаписи можна захистити від несанкціонованого використання та піратства за допомогою різних методів. Ось декілька основних методів захисту аудіозаписів:

- Цифрові водяні знаки є одним з найефективніших методів захисту аудіоданих. Вони вбудовуються безпосередньо в аудіофайл та містять унікальну інформацію, таку як авторство, дата створення та інші метадані. Ці водяні знаки стійкі до різних видів обробки файлу і можуть бути використані для виявлення підробок.

- Шифрування може бути застосоване для захисту аудіозаписів під час їх передачі по мережі або зберігання на сервері. За допомогою шифрування дані у аудіофайлах можуть бути захищені від несанкціонованого доступу та перехоплення.

- Технології DRM (Digital Rights Management). Це технології управління цифровими правами надають можливість контролювати доступ до аудіозаписів шляхом обмеження кількості програвань, копіювання або пересилання файлу. DRM може бути використаний для захисту від незаконного використання та розповсюдження аудіоматеріалів.

- Методи фізичного захисту: Для додаткового захисту аудіозаписів можуть бути використані фізичні методи, такі як використання захисних оболонок або

спеціальних маркерів, що унеможливлюють несанкціоноване копіювання або викрадення.

Ці методи можуть бути використані окремо або в поєднанні з іншими технічними та правовими заходами для максимального захисту аудіозаписів від незаконного використання та піратства (мал 1.3) [8].

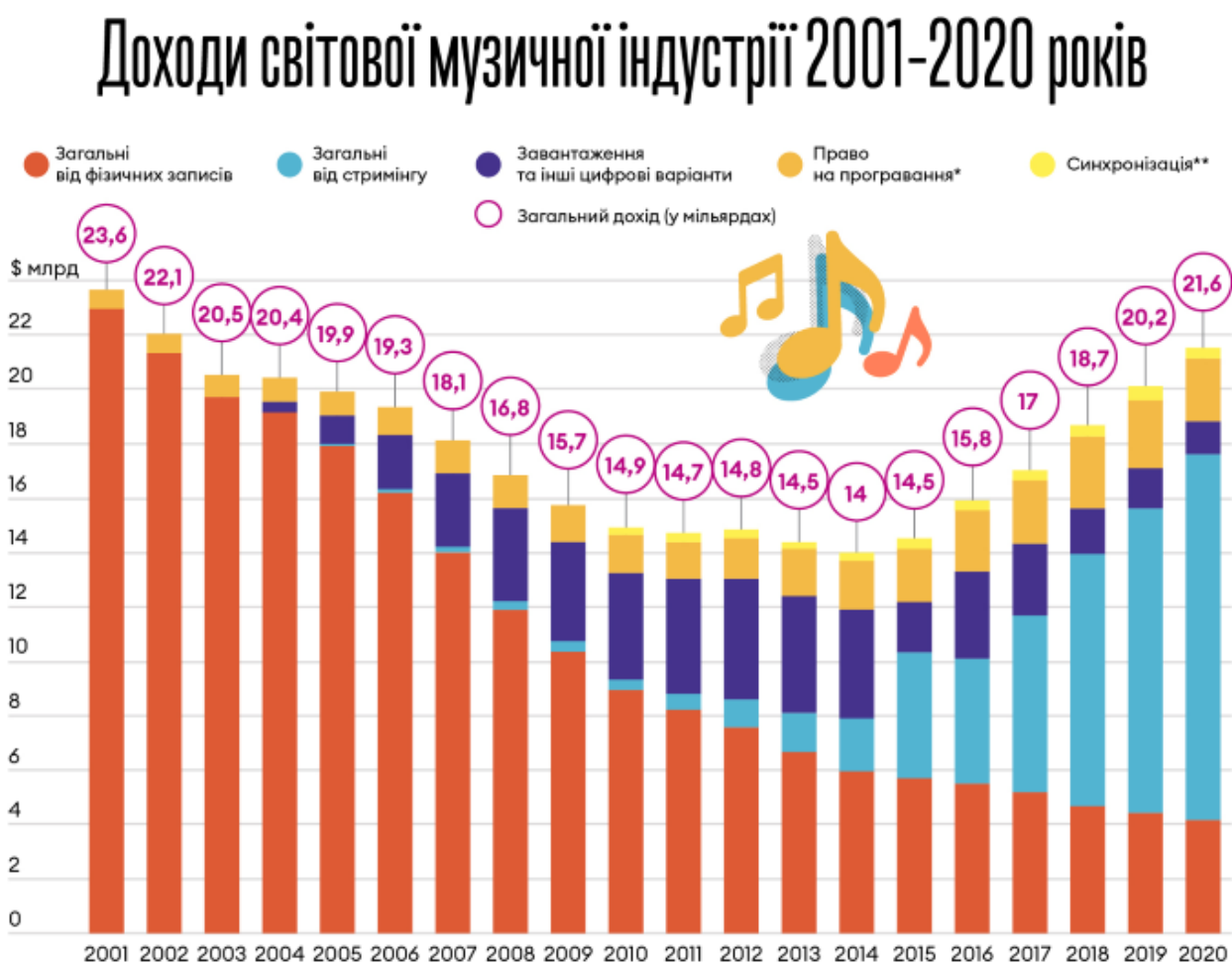


Рисунок 1.3 – Світова музична індустрія

Відповідно до даних у сучасному цифровому світі захист аудіозаписів стає ключовою проблемою, особливо в контексті зростаючої цінності інформації та інтелектуальної власності. Відновлювання інтелектуальної власності та запобігання несанкціонованому використанню аудіозаписів стає викликом для багатьох галузей, таких як музична індустрія, подкастинг та аудіо контент-провайдери.

Актуальність цієї проблеми полягає в тому, що аудіозаписи стають об'єктом прагнень для порушників авторських прав та осіб, які намагаються отримати користь з використання контенту без дозволу власника. Це може призвести до втрати прибутку для творців аудіоконтенту, порушення їхньої репутації та втрати конкурентної переваги.

Зростаюча доступність технологій для обробки звуку і редагування аудіофайлів створює нові можливості для швидкого та ефективного виявлення, копіювання та модифікації аудіозаписів. Це підсилює необхідність розвитку ефективних методів захисту, які можуть забезпечити конфіденційність, цілісність та доступність аудіоконтенту в інтернеті та інших медіа-платформах.

MUSO - компанія що спеціалізується на зборі даних про цифрове піратство та вимірюванні доступу до піратської екосистеми, опублікувала цікаві результати. Вони показують, що телевізійний контент є найбільшим медіасектором, на який припадає 46,5% всього трафіку на піратські сайти.

За перші вісім місяців 2022 року галузеві дані MUSO щодо піратства продемонстрували помітну тенденцію: У період з січня по серпень 2022 року експерти MUSO зафіксували 141,6 млрд відвідувань піратських сайтів у всіх галузях, що на 21,8% більше, ніж за період 2021 року.

Якщо розбити цю цифру на сектори медіаіндустрії, то найбільшим виявився телевізійний сектор, який включає телебачення, анімацію та прямі спортивні трансляції. У цих секторах на телевізійний контент припадає 46,5% усього трафіку на піратські сайти (мал 1.4).

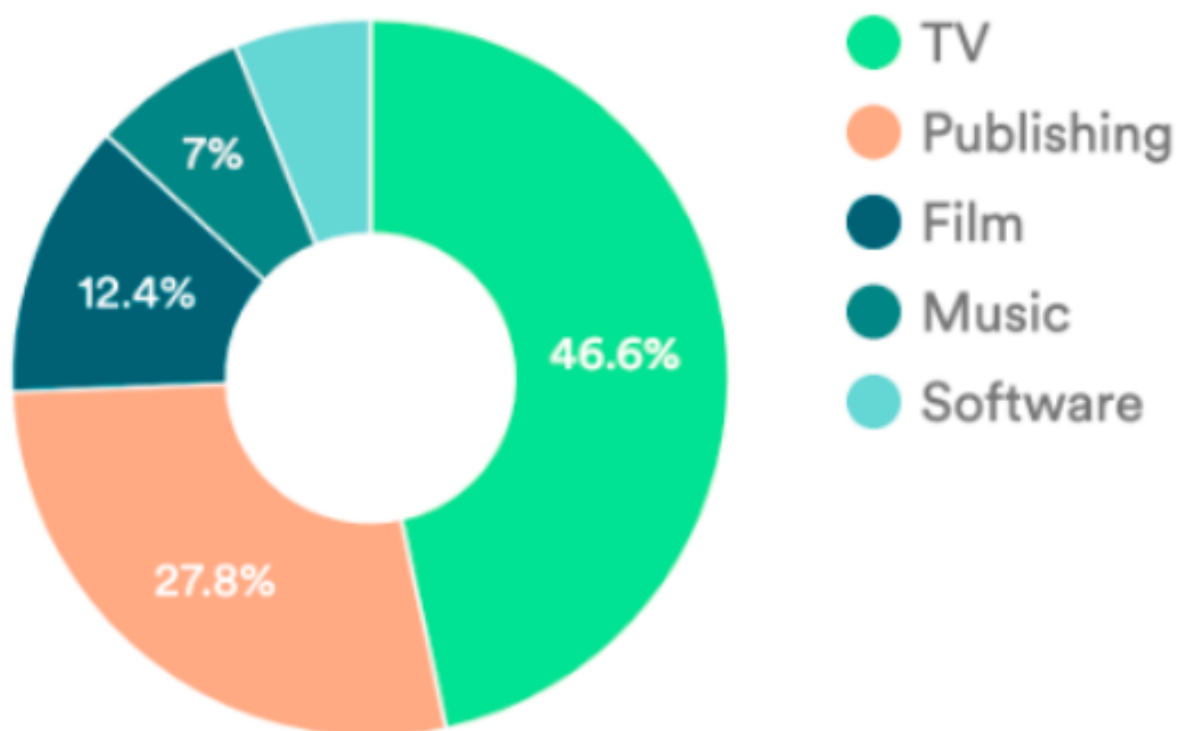


Рисунок 1.4 – Розподіл піратського контенту (за [9])

Порівнюючи дані, виміряні в кожному секторі за період січень-лютий 2022 року, з даними за той самий період 2021 року, MUSO повідомляє, що піратство зросло в усіх секторах.

Трафік піратських фільмів демонструє найбільше зростання - 49,0%. Найменший приріст у музичному секторі - 3,80%.

Що стосується способу розповсюдження неліцензійного контенту, то 53,8% трафіку припадає на неліцензійні стрімінгові сайти, тоді як 46,2% трафіку - на сайти для завантаження (включаючи торренти та кібер-блокатори).

Ось дані про розподіл піратського контенту за країнами (див. рис. 1.5) [9].

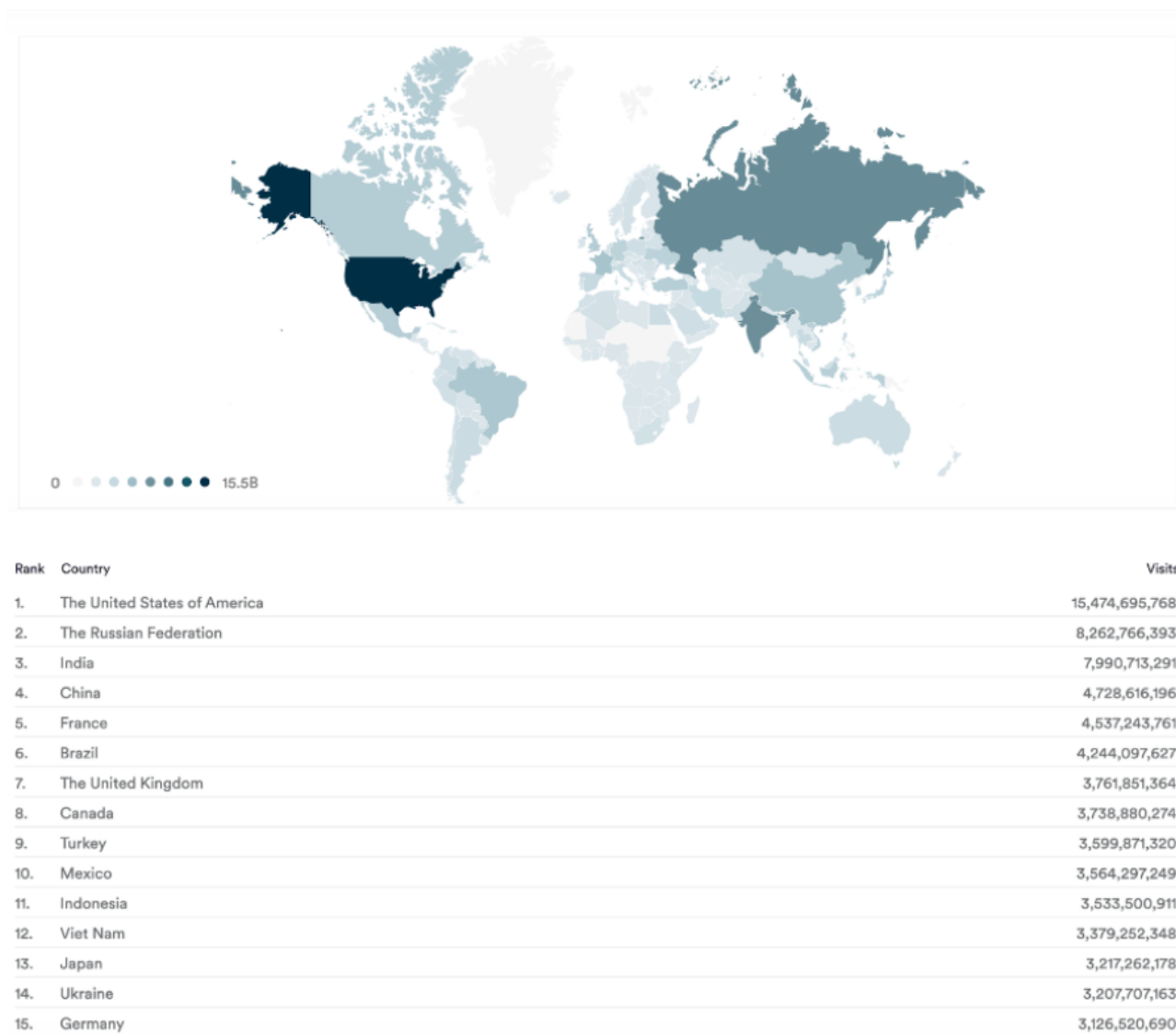


Рисунок 1.5 – Розподіл піратського контенту за країнами (за [9])

Таким чином, розробка ефективних та надійних методів збереження та захисту аудіозаписів як інтелектуальної власності стає актуальною проблемою для бізнесу та наукової спільноти. Розв'язання цієї проблеми не лише сприятиме захисту авторських прав, а й забезпечить стабільність та інноваційний розвиток у цифровому середовищі.

Під час збереження аудіозаписів виникає кілька потенційних проблем, які можуть негативно вплинути на їхнє використання та захист:

- Підробка та несанкціоноване використання. Несанкціоновані користувачі можуть використовувати аудіозаписи без дозволу власника, порушуючи його

авторські права. Це створює ризик підробки та незаконного розповсюдження контенту.

- Втрата якості при збереженні: Збереження аудіозаписів у зашифрованому або з водяним знаком форматі може призвести до втрати якості через стиснення даних або додавання зайвих артефактів, що погіршує якість звуку.

- Вразливість перед атаками хакерів: Застосування захисту може бути обхідним або вразливим перед атаками хакерів або зловмисників, що може призвести до незаконного доступу та використання аудіозаписів.

- Сумісність з платформами: Деякі методи захисту можуть бути несумісними з певними платформами або програмним забезпеченням, що ускладнює їх використання та обмін між користувачами.

- Вартість інфраструктури захисту, розробка та підтримка ефективної інфраструктури для збереження та захисту аудіозаписів може бути дорогим завданням для компаній та організацій, особливо для малих підприємств або стартапів.

Незважаючи на мінуси використання водяного знаку, який може бути менш ефективним у виявленні та захисті від піратства порівняно з іншими методами, такими як DRM, об'єднання цих технологій може мати кілька переваг:

- Комплексний захист: Об'єднання водяного знаку з технологією DRM може забезпечити комплексний підхід до захисту цифрового контенту, що зробить його більш стійким до несанкціонованого використання.

- Додаткові шари захисту: Водяні знаки можуть доповнювати захист, наданий DRM, дозволяючи встановлювати додаткові шари захисту для виявлення підробок та забезпечення автентичності аудіозаписів.

- Більша надійність: Об'єднання різних технологій захисту може зробити систему більш надійною, оскільки це ускладнює процес обходу захисту та забезпечує більш ефективний контроль над цифровим контентом.

– Гнучкість: Використання обох методів дає можливість вибору оптимального підходу до захисту для конкретних потреб та обставин, що забезпечує більшу гнучкість у впровадженні заходів захисту.

1.3 Існуючі стеганографічні методи накладання ЦВЗ

Захист авторських прав на аудіозаписи є важливим аспектом у сучасному цифровому світі. Одним із ефективних способів забезпечення цього захисту є використання стеганографії для вбудовування водяних знаків у аудіозаписи. Водяні знаки дозволяють приховано вбудовувати ідентифікаційну інформацію в аудіосигнал, що ускладнює несанкціоноване використання або копіювання контенту. У цьому розділі розглядаються основні методи стеганографії, які використовуються для вбудовування водяних знаків в аудіозаписи [10, 11].

Кодування найменш значущих бітів (LSB)

Метод кодування найменш значущих бітів (LSB) є одним із найпростіших і широко використовуваних методів стеганографії. Цей метод полягає у заміні найменш значущих бітів в аудіосигналі на біти прихованого повідомлення.

Принцип роботи:

Крок 1.

Обирається аудіосигнал формату WAVE, у якому буде вбудовуватися водяний знак.

Крок 2.

Аудіосигнал перетворюється у бінарний формат, де кожен звуковий зразок представлений у вигляді послідовності бітів.

Крок 3.

Найменш значущі біти в аудіосигналі замінюються на біти прихованого повідомлення. Це робиться шляхом запису бітів повідомлення в найменш значущі позиції кожного звукового зразка.

Крок 4.

Для витягування прихованого повідомлення з аудіосигналу використовується аналогічний процес: зчитується найменш значущі біти з аудіосигналу і перетворюються у повідомлення.

Процес вбудовування можна виразити наступною формулою (1.1):

$$S'_i = (S_i \& 0xFE) | M_i \quad S'_i = (S_i \& 0xFE) | M_i \quad (1.1)$$

де:

- S_i — i -й байт оригінального сигналу,
- S'_i — i -й байт сигналу з водяним знаком,
- M_i — i -й біт повідомлення,
- $\& 0xFE$ — побітова операція AND, яка встановлює найменш значущий біт у 0,
- $|M_i|$ — побітова операція OR, яка встановлює найменш значущий біт згідно з M_i .

Переваги:

- Висока пропускна здатність.
- Мало помітний низькочастотний шум.

Недоліки:

- Низька стійкість до атак та перетворень сигналу.
- Легкість виявлення прихованого повідомлення.

Приховування даних із використанням ехо-сигналу

Метод приховування даних із використанням ехо-сигналу полягає у вбудовуванні секретного повідомлення у сигнал-контейнер через ехо-сигнал. Це досягається зміною параметрів ехо-сигналу, таких як початкова амплітуда, швидкість затухання та зсув.

Принцип роботи:

Крок 1.

Ехо-сигнал створюється шляхом впливу основного сигналу на навколишнє середовище або об'єкт, що призводить до виникнення ехо-ефекту.

Крок 2.

Параметри ехо-сигналу, такі як початкова амплітуда, швидкість затухання та зсув, аналізуються для визначення точок, в яких можна вбудувати секретне повідомлення.

Крок 3.

Секретне повідомлення кодується та вбудовується у ехо-сигнал шляхом зміни вказаних параметрів. Наприклад, може змінюватися амплітуда ехо-сигналу, час його затухання або частота.

Крок 4.

Модифікований ехо-сигнал об'єднується з основним сигналом для створення сигналу-контейнера, в якому приховується інформація.

Крок 5.

Для відтворення секретного повідомлення ехо-сигнал повторно аналізується, і за допомогою відомих ключів або алгоритмів витягується прихована інформація.

Процес вбудовування можна виразити наступною формулою:

$$S'(t) = S(t) + \beta E(t) \quad S'(t) = S(t) + \beta E(t) \quad (1.2)$$

де:

- $S(t)$ — оригінальний сигнал у часовій області,
- $S'(t)$ — модифікований сигнал з водяним знаком,
- $E(t)$ — ехо-сигнал, який містить повідомлення,
- β — коефіцієнт, що визначає інтенсивність ехо-сигналу.

Переваги:

- Висока стійкість до більшості атак.
- Непомітність для людського слуху.

Недоліки:

- Складність у визначенні параметрів ехо-сигналу.
- Висока обчислювальна складність.

Метод розширення спектра

Метод розширення спектру використовується для вбудовування інформації у звуковий сигнал шляхом множення сигналу даних на сигнали несучої і

псевдовипадкової шумової послідовності. Це призводить до розширення спектру даних на всю доступну смугу частот. Після цього розширені дані додаються до вихідного сигналу як адитивний випадковий шум.

Принцип роботи:

Крок 1.

Спочатку виконується аналіз спектральних характеристик аудіосигналу, щоб визначити частотний діапазон, який можна використовувати для вбудовування додаткової інформації.

Крок 2.

Повідомлення, яке потрібно вбудувати в аудіосигнал, конвертується у вигляд бітової послідовності.

Крок 3.

Біти повідомлення розподіляються по різних частотних складових аудіосигналу. Це може відбуватися шляхом зміни амплітуди, фази або часу тривалості окремих частотних компонентів.

Крок 4

Отримана інформація вбудовується в аудіосигнал, зазвичай шляхом арифметичних операцій, таких як додавання або множення, щоб отримати модифікований сигнал.

Процес вбудовування можна виразити наступною формулою:

$$S'(t) = S(t) + \alpha M(t) \quad (1.3)$$

де:

- $S(t)$ — оригінальний сигнал у часовій області,
- $S'(t)$ — модифікований сигнал з водяним знаком,
- $M(t)$ — секретне повідомлення, розподілене по спектру,
- α — коефіцієнт, що визначає інтенсивність вбудованого повідомлення.

Переваги:

- Висока стійкість до шумів та атак.
- Можливість вбудовування великого обсягу інформації.

Недоліки:

- Висока обчислювальна складність.
- Потреба у значному обсязі контейнерного сигналу.

Метод фазового кодування

Метод фазового кодування використовується для вбудовування секретного повідомлення у звуковий сигнал шляхом заміни фази вихідного звукового сегмента на опорну фазу. При цьому змінюються параметри фази так, щоб відображати секретні дані. Крім того, для збереження різницевої фази між сегментами, фази узгоджуються. Хоча навіть невеликі зміни можуть бути непомітними на слух, фахівці по спектральному аналізу можуть виявити такі зміни.

Принцип роботи:

Крок 1.

Розбиття сигналу на сегменти: Вихідний аудіосигнал розбивається на короткі сегменти. Кожен сегмент обробляється окремо, що дозволяє більш точно контролювати зміни фази.

Крок 2.

Перетворення сигналу: Кожен сегмент сигналу перетворюється з тимчасової області в частотну область за допомогою, наприклад, швидкого перетворення Фур'є (FFT). Це дозволяє виділити частотні компоненти кожного сегмента.

Крок 3.

Модифікація фази: Фаза кожної частотної складової сегмента змінюється відповідно до бітів секретного повідомлення. Для цього кожному біту повідомлення призначається певна зміна фази. Зміни фази повинні бути достатньо малі, щоб не спотворити аудіосигнал на слух, але достатньо великі, щоб їх можна було надійно зчитати при декодуванні.

Крок 4.

Узгодження фаз: Для збереження різницевої фази між сусідніми сегментами фази коригуються, щоб зміни фази між сегментами були плавними і не викликали помітних розсіювань фази.

Крок 5.

Перетворення сигналу назад: Модифіковані сегменти сигналу перетворюються назад у тимчасову область з частотної області за допомогою зворотного швидкого перетворення Фур'є (IFFT).

Крок 6.

Об'єднання сегментів: Модифіковані сегменти об'єднуються в єдиний сигнал, який тепер містить приховане повідомлення.

Крок 7.

Відтворення та витягування повідомлення: Для відтворення прихованого повідомлення сигнал знову розбивається на сегменти, кожен сегмент перетворюється в частотну область, і зчитуються зміни фаз, що відповідають біту прихованого повідомлення.

Процес вбудовування можна виразити наступною формулою:

$$S'(t) = S(t) + \gamma \Delta\phi(M) S'(t) = S(t) + \gamma \Delta\phi(M) \quad (1.4)$$

де:

- $S(t)$ — оригінальний сигнал у часовій області,
- $S'(t)$ — модифікований сигнал з водяним знаком,
- $\Delta\phi(M)$ — зміна фази, яка визначається повідомленням,
- γ — коефіцієнт, що визначає інтенсивність зміни фази.

Переваги:

- Висока стійкість до атак.
- Непомітність для людського слуху.

Недоліки:

- Висока обчислювальна складність.
- Потреба у точному синхронному перетворенні для витягання водяного знака.

При аналізі стеганографічних методів важливо враховувати кількісні показники стійкості водяних знаків. Основні показники включають пропускну здатність, непомітність, робастність, стійкість до атак і психоакустичний аналіз [10].

Ось таблиця з кількісними показниками та характеристиками для чотирьох розглянутих методів стеганографії (див таблиця 1.1) [11]:

Таблиця 1.1 – Порівняння характеристик стеганографічних методів

Метод	Пропускна здатність	Непомітність	Стійкість до атак	Психоакустичний аналіз
Кодування найменш значущих бітів	$C = \frac{n}{m}$ Мала	Високий коефіцієнт сигнал/шум	Низька, проста маніпуляція може зруйнувати водяний знак	Висока непомітність через зміну найменш значущих бітів
Метод розширення спектра	$C = \frac{W}{B}$ Висока	Висока непомітність, адитивний шум	Середня, можливість виявлення досвідченими атаками	Висока непомітність через адитивний шум
Приховування даних із використанням ехо-сигналу	$C = \frac{I}{T}$ Мала	Висока непомітність, природний резонанс	Середня, можливі атаки на видалення ехо	Висока непомітність через природний ехо-ефект
Метод фазового кодування	$C = \frac{n}{m}$ Мала	Висока непомітність, зміни фази	Висока, складний для виявлення і модифікації	Висока непомітність через низьку чутливість до змін фази

Примітки:

- n кількість бітів водяного знака.
- m кількість вибірок у аудіосигналі.
- W кількість вбудованих бітів.
- B смуга пропускання сигналу.
- I обсяг вбудованої інформації.

– T тривалість сигналу.

Аналіз існуючих методів показує, що вони дозволяють приховано інтегрувати ідентифікаційну інформацію в аудіосигнал, зберігаючи при цьому якість звучання та непомітність водяного знаку. Таким чином, стеганографія водяних знаків є потужним інструментом для захисту інтелектуальної власності в аудіосфері, сприяючи розвитку безпечних і законних способів розповсюдження цифрового контенту. Саме тому для розробки даного проекту було обрано цифрові водяні знаки через їх перевагу [11-14].

1.4 Огляд існуючих сервісів із захистом авторського права

Розглянемо найпопулярніші аналоги які спеціалізуються на продажу аудіозаписів з захистом авторського права:

Content ID (YouTube) – це автоматична система розпізнавання контенту, розроблена YouTube, яка використовує цифрові водяні знаки для виявлення порушень авторських прав. Власники контенту завантажують свої аудіо- або відеофайли в базу даних Content ID, і система перевіряє завантажений контент на платформі, щоб знайти збіги з захищеним матеріалом.

Переваги: Content ID надає ефективний інструмент для виявлення та моніторингу порушень авторських прав на платформі YouTube. Система має глобальне охоплення і дозволяє правовласникам автоматично монетизувати або блокувати контент, який порушує авторські права.

Недоліки: Система обмежена лише платформою YouTube і може давати помилкові спрацьовування, що призводить до блокування легального контенту. Крім того, не всі творці мають доступ до цієї системи [15].

Digimarc пропонує комплексні рішення для захисту цифрових медіа, включаючи аудіо. Використовуючи стійкі цифрові водяні знаки, Digimarc дозволяє правовласникам вбудовувати невидимі метадані у свої аудіозаписи, що допомагає в ідентифікації і відстеженні контенту [16].

Переваги: Digimarc пропонує стійкі до маніпуляцій водяні знаки, які можуть бути вбудовані в різні типи медіаконтенту, включаючи аудіо. Це дозволяє

ефективно відстежувати ідентифікацію контенту навіть після його модифікації.

Недоліки: Висока вартість впровадження та обслуговування робить цю технологію менш доступною для незалежних музикантів і невеликих правовласників.

SoundCloud є популярною платформою для розповсюдження аудіоконтенту, яка також використовує технологію цифрових водяних знаків для захисту авторських прав. Завдяки цьому, SoundCloud може відстежувати нелегальне використання контенту та забезпечувати відповідні заходи захисту [17].

Переваги: Платформа забезпечує інтеграцію захисту аудіоконтенту і дозволяє користувачам розповсюджувати музику. Це сприяє охороні авторських прав і моніторингу використання контенту.

Недоліки: Обмежені функції для безкоштовних користувачів та менш розвинені можливості для моніторингу порівняно з іншими спеціалізованими сервісами.

TuneCore – це сервіс для незалежних музикантів, який дозволяє їм розповсюджувати свою музику через численні платформи та магазини. TuneCore використовує цифрові водяні знаки для захисту творів своїх користувачів, що допомагає забезпечити дотримання авторських прав [18].

Переваги: TuneCore підтримує незалежних музикантів, дозволяючи їм розповсюджувати свою музику на багатьох платформах. Це сприяє розширенню аудиторії та забезпеченню захисту авторських прав.

Недоліки: Послуги TuneCore коштують, що може бути обмеженням для деяких користувачів. Деякі функції можуть бути обмежені в залежності від обраного тарифного плану.

Audible Magic надає технологічні рішення для виявлення та управління авторськими правами на цифровий контент. Використовуючи стійкі водяні знаки, Audible Magic може ідентифікувати та відстежувати захищений контент на різних платформах і сервісах, забезпечуючи дотримання авторських прав [19].

Переваги: Audible Magic забезпечує високу точність ідентифікації контенту та підтримує масштабованість, що робить його ефективним інструментом для

великих правовласників.

Недоліки: Висока вартість впровадження та складність інтеграції можуть бути перешкодою для менших правовласників.

Технічні недоліки існуючих сервісів захисту авторських прав пов'язані здебільшого з високою вартістю впровадження та обслуговування, складністю інтеграції з існуючими системами, а також обмеженою функціональністю деяких платформ. Ці проблеми можуть ускладнювати вибір оптимального рішення для правовласників, особливо для незалежних музикантів та невеликих компаній.

Саме через високу вартість впровадження та обмеження в рамках платформи даний проект бакалаврської роботи є актуальним та вирішує вище вказані проблеми.

1.5 Висновки та постановка задач

У цьому розділі було проведено аналіз сфер застосування аудіофайлів, що охоплює їх використання у створенні музики, звукового дизайну для ігор та фільмів, а також у різних аудіовізуальних проектах. Цей огляд дозволив краще зрозуміти значення та роль аудіо у сучасному цифровому середовищі.

Була обґрунтована актуальність проблеми зберігання аудіофайлів як об'єктів авторського права та їхнього захисту від незаконного використання. Проведений аналіз існуючих сервісів по обміну та продажу аудіофайлів виявив їхню вразливість через відсутність ефективних заходів безпеки, що створює потенційну загрозу конфіденційності та цілісності аудіоматеріалів.

На основі цього аналізу було прийнято рішення про розробку власного сервісу, який поєднає переваги наявних платформ та забезпечить надійний захист аудіофайлів як інтелектуальної власності. Важливим кроком в цьому процесі є використання ефективних методів шифрування та вбудовування цифрових підписів для забезпечення безпеки аудіоданих на різних етапах їхнього обміну та зберігання.

Виходячи з отриманих результатів дослідження, було сформовано подальші задачі роботи:

- розробка структури та функціоналу веб-сервісу; а також моделей захисту веб-сервісу від SQL-ін'єкцій та XSS-атак, двохфакторної автентифікації, SSL-сертифікату;
- вибір та обґрунтування обраних засобів програмування для розробки захищеного веб-сервісу;
- проектування та розробка інтерфейсу користувача, реалізація програмного засобу у вигляді веб-сервісу;
- тестування розробки та аналіз отриманих результатів.

Виконання поставлених завдань сприятиме досягненню основної мети роботи, що полягає у розробці захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками з використанням бази даних MySQL, фреймворку PHP.

2 ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ЗАХИСТУ АУДІОФАЙЛІВ

В даному розділі ми детально розглянемо процес проектування програмного додатку, який є ключовим етапом у створенні веб-сервісу. В даному розділі розглянемо архітектуру, структуру та функціональність онлайн сервісу, а також проаналізуємо та виберемо мову програмування.

2.1 Розробка архітектури сервісу захищеного онлайн сервісу продажу аудіозаписів

Розробка програмної архітектури веб-сервісу захисту аудіофайлів планується з використанням шаблону Model-View-Controller (MVC). Це шаблон архітектури програмного забезпечення, який дозволяє розділити програму на три основні компоненти: Модель (Model), Вид (View) і Контролер (Controller).

Шаблон MVC ідеально підходить для розробки веб-сервісу захисту аудіофайлів інтелектуальної власності через кілька ключових причин: він дозволяє чітко розділити функціонал додатку на модель, представлення і контролер, що сприяє гнучкості та підтримці, спрощує тестування кожного компонента окремо і забезпечує підтримку різних пристроїв завдяки можливості створення різних представлень.

Кожен з цих компонентів відповідає за свою відповідальність і сприяє вдосконаленню модульності та підтримки коду [20].

Відповідно до шаблону MVC, структура веб-сервісу для захисту авторських прав аудіофайлів буде розроблена наступним чином.

Модель:

- Аудіофайли (Audio Files): Модель, що представляє аудіофайли, їх властивості та методи для роботи з ними.
- Цифрові підписи (Digital Signatures): Модель, що відповідає за генерацію та перевірку цифрових підписів для аудіофайлів користувачів.
- Користувачі (Users): Модель, яка представляє користувачів веб-сервісу.

Вид:

- Форма завантаження (Upload Form): Вигляд, який дозволяє користувачам

завантажувати свої аудіофайли на сервер.

– Сторінка результатів (Results Page): Вигляд, який показує результати обробки аудіофайлів, такі як успішність або помилки.

Контролер:

– Аудіофайли (Audio Files): Контролер, який обробляє запити, пов'язані з завантаженням та обробкою аудіофайлів.

– Користувачі (Users): Контролер, який відповідає за авторизацію та аутентифікацію користувачів.

MVC дозволяє розділити програму на логічно відокремлені компоненти, що полегшує розробку, розуміння та підтримку коду.

Для даного веб-сервісу захисту аудіофайлів модель має папку для моделей, в якій зберігаються класи, що визначають дані та логіку накладання ЦВЗ, окрему папку для контролерів, де розташовані файли, які обробляють запити від користувачів, і окрему для видів, де знаходяться аудіофайли, які відображаються покупцям (рис. 2.1).

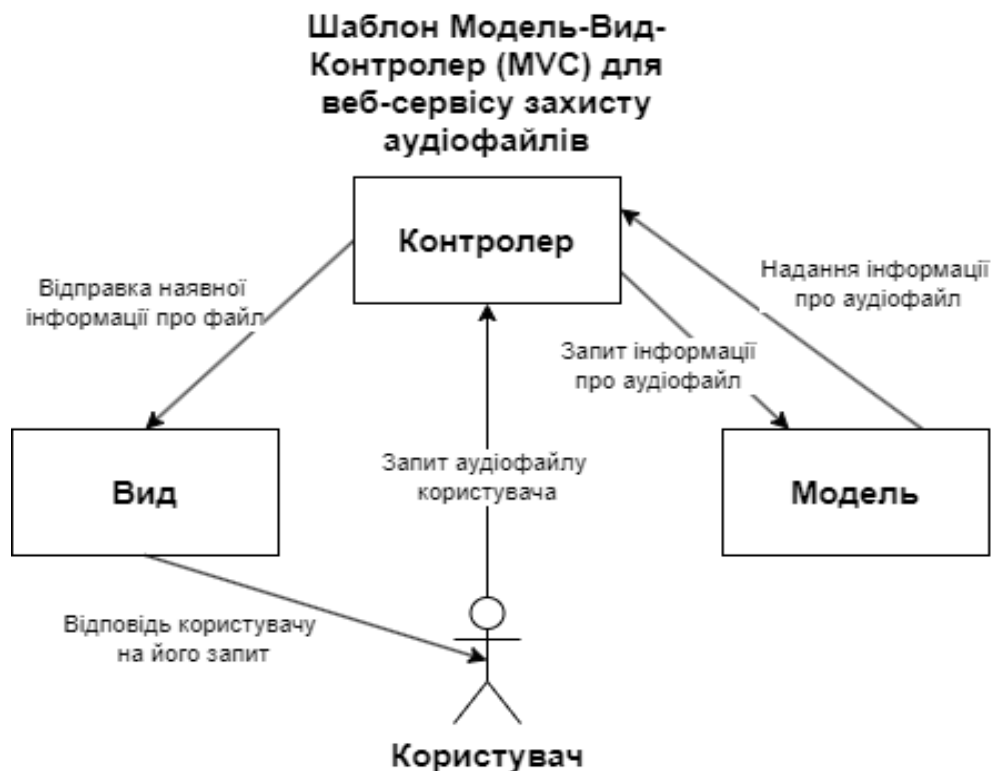


Рисунок 2.1 – MVC логіка розробленого веб-сервісу

У процесі розробки архітектури для захищеного онлайн сервісу продажу

аудіозаписів із захистом авторського права стійкими цифровими водяними знаками дотримано методології MVC.

Контролери відповідають за обробку запитів від користувачів. Створено окремі контролери для різних аспектів сервісу, наприклад, контролер для обробки запитів на покупку аудіозаписів, контролер для авторизації користувачів тощо.

Після цього розроблюються моделі. Моделі відповідають за логіку бізнес-процесів та взаємодію з базою даних.

Види відповідають за представлення даних користувачу та взаємодію з ним. Створено інтерфейс, який дозволяє користувачу переглядати доступні аудіозаписи, робити покупки та загрузати свої аудіо доріжки.

Таким чином, використовуючи методологію MVC, розділено розробку проекту на логічно відокремлені компоненти, що сприяло покращенню модульності та підтримки коду.

Діаграми варіантів використання (Use Case Diagrams) у мові моделювання UML (Unified Modeling Language) використовуються для візуалізації функціональності системи з точки зору її користувачів.

Для того, щоб зрозуміти та спростити розробку веб-сервісу захисту аудіо проаналізуємо та спроектуємо діаграму варіантів використання для того, щоб візуалізувати можливі взаємодії користувачів з системою [21].

Діаграми варіантів використання ідентифікують основні функціональність системи, представлену у вигляді варіантів використання або сценаріїв, які показують, як користувачі взаємодіють з системою для досягнення певних цілей (рис. 2.2).



Рисунок 2.2 – Основні позначення діаграми варіантів використання

Учасники системи - це стейкхолдери або ролі, які взаємодіють з системою веб-сервісу. Варіанти використання або сценарії взаємодії з проектом представлені у вигляді овалів на діаграмі. Кожен варіант використання описує конкретну функціональність або можливість, яку система надає користувачам веб-сервісу.

Діаграми варіантів використання є потужним інструментом для аналізу та моделювання функціональності системи з точки зору її користувачів. Вони дозволяють зрозуміти потреби користувачів даного проекту та визначити основні функції веб-сервісу перед розробкою.

Розглянемо деталі проектування діаграми варіантів використання для даного веб-сервісу [21]:

- Ідентифікація акторів (учасників). Першим кроком у проектуванні діаграм варіантів використання є ідентифікація усіх акторів, тобто тих, хто взаємодіє з системою. Це користувачі, зовнішні системи та інші об'єкти, що використовують систему.

- Визначення варіантів використання. Наступним кроком є ідентифікація та опис кожного варіанту використання. Кожен варіант використання представляє собою конкретний сценарій або функціональність, яку система надає користувачам для досягнення певних цілей. В даному випадку це буде авторизація, двохфакторна автентифікація, додавання аудіо, накладання ЦВЗ, покупка завантажених аудіофайлів та коментування.

- Встановлення взаємозв'язків. Після ідентифікації акторів та варіантів використання необхідно встановити взаємозв'язки між ними. Це вказує, які актори беруть участь у кожному варіанті використання та як вони взаємодіють.

- Розробка уточнень та розширень. Для кожного варіанту використання можуть бути розроблені уточнення або розширення, які деталізують дії користувача або системи у конкретних умовах чи відповідно до певних вимог. Як от покупка аудіо має розширення пошуку товарів та їх перегляд та прослуховування.

– Узагальнення та спадкування. У деяких випадках може бути використано узагальнення та спадкування для спільних варіантів використання. Це дозволяє уникнути дублювання та спростити модель.

UCD діаграма веб-сервісу захисту авторських прав аудіофайлів має таку структуру:

Актори:

- Користувач (User): Представляє звичайного користувача системи.
- Адміністратор (Administrator): Відповідає за управління системою.
- MySQL: Представляє базу даних, з якою взаємодіє система.
- Сервіс двохфакторної автентифікації: Представляє собою сервіс електронної пошти, на яку має прийти код для автентифікації користувача.

Випадки використання:

- Реєстрація: Користувач може зареєструватися в системі.
- Авторизація: Користувач може увійти в систему після реєстрації.
- Перегляд товарів: Користувач може переглядати список товарів.
- Додавання товарів: Користувач може додати свій аудіофайл та захистити його.
- Пошук товарів: Користувач може шукати товари за назвою, описом чи автором.

Зв'язки:

- Користувач взаємодіє з системою через випадки використання.
- Адміністратор може керувати користувачами та товарами.
- Система взаємодіє з базою даних MySQL для зберігання інформації.

Додаткові функціональні вимоги:

- Стійкі цифрові водяні знаки. На всі завантажені аудіозаписи автоматично накладаються водяні знаки, щоб захистити їх від піратства. Це і є основна мета веб-сервісу
- Управління авторськими правами. Система забезпечує ефективне управління авторськими правами та контроль над доступом до аудіозаписів.

Після завершення проектування діаграми варіантів використання, вона буде використана як основа для подальшого аналізу, проектування та розробки системи веб-сервісу захисту авторських прав аудіофайлів. Дана діаграма допомогла зрозуміти потреби користувачів та визначити основні функціональні вимоги до системи даного проекту перед реалізацією (рис. 2.3).

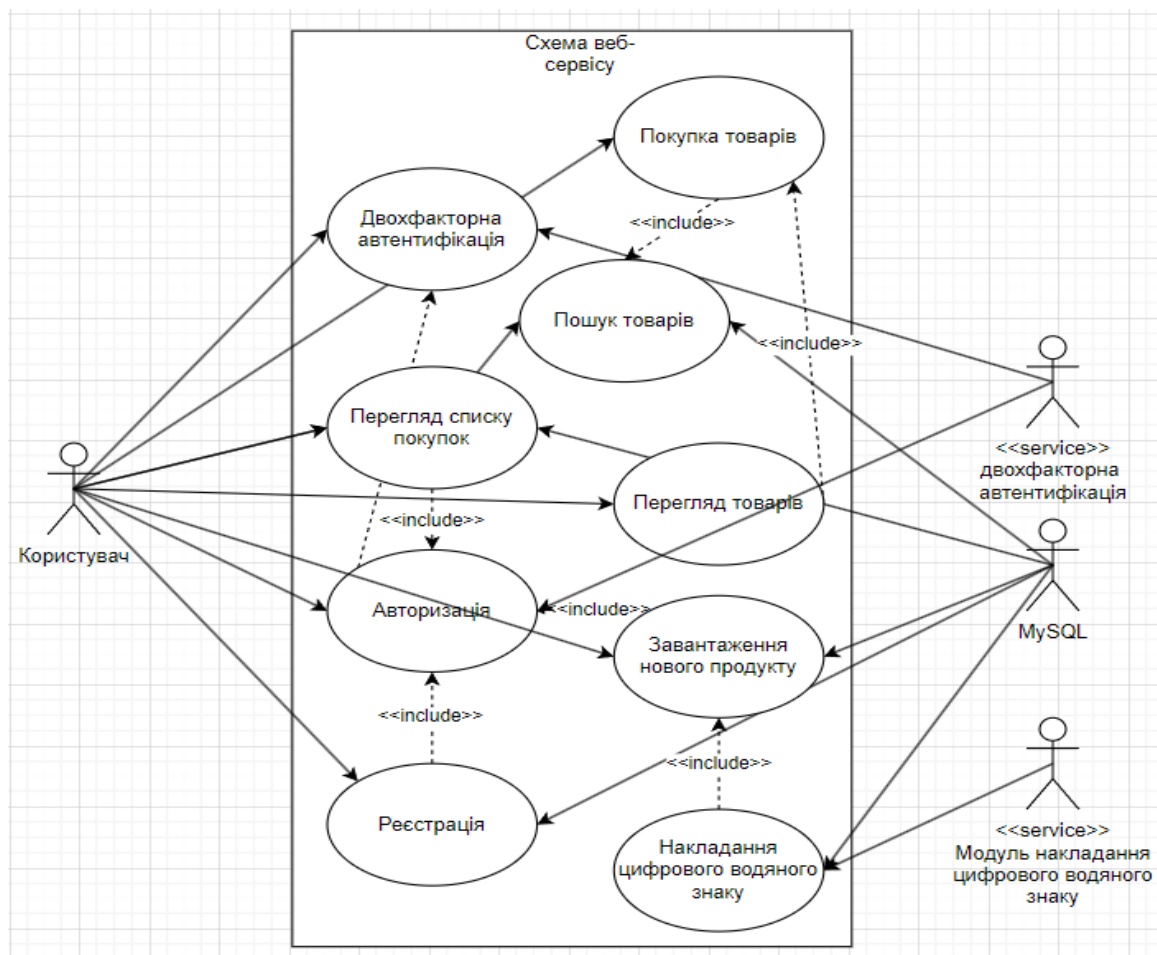


Рисунок 2.3 – Діаграма прецедентів (Use-case diagram) розробленого онлайн сервісу

Проектування захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права вимагає уважного аналізу та визначення функціональних вимог. У цьому контексті, блок-схеми є невід'ємною частиною процесу розробки. Вони є потужним інструментом для візуалізації послідовності дій або процесів, що відбуваються у веб-сервісі.

Перш за все, блок-схеми допомагають просунути розуміння процесу розробки та функціональних можливостей веб-сервісу. Вони уточнюють та документують кроки, які необхідно виконати, щоб система працювала належним чином.

Також вони дозволяють виявити можливі недоліки чи покращення в процесі розробки. Важливо мати чітку уяву про послідовність дій, які потрібно виконати, щоб гарантувати ефективну та безперебійну роботу веб-сервісу.

Блок-схеми також можуть використовуватись як частина документації процесу розробки. Вони забезпечують зручний спосіб візуалізації та пояснення алгоритмів та процесів для всіх учасників розробки [22].

Для проектування UML моделі даного веб-сервісу необхідно детально розглянути алгоритм його функціонування. Нижче наведено покроковий алгоритм, який допоможе в розробці цієї моделі:

Крок 1. Вхід:

Користувач переходить за посиланням на даний веб-сервіс та потрапляє на сторінку авторизації. На даній сторінці можна ввести дані існуючого облікового запису або зареєструвати новий.

Крок 2. Створення облікового запису:

Якщо користувач має намір створити новий обліковий запис то існує два шляхи:

- Якщо користувач відповідає “Так” – Переходимо до кроку 3.
- Якщо користувач відповідає “Ні” – Переходимо до кроку 4.

Крок 3. Реєстрація:

Користувач вводить свої облікові дані такі як – ПІБ, номер телефону, електронна пошта і звичайно пароль. Усі дані вносяться до бази даних та шифруються.

Далі генерується унікальний ідентифікатор для нового облікового запису та додається до бази даних.

Після реєстрації користувач потрапляє на головну сторінку та може користуватись усіма функціями.

Якщо користувач бажає додати аудіофайл та захистити його наклепавши на нього ЦВЗ - переходимо до кроку 5.

Якщо ні – процес закінчується

Крок 4. Авторизація:

Якщо користувач вже реєструвався у системі то йому потрібно ввести свої облікові дані – електронну пошту та пароль. Якщо дані правильні та містяться в базі даних, користувач повинен увести ще код, який надійде на його пошту, після чого авторизація буде успішною та користувач потрапить на головну сторінку.

Якщо користувач бажає додати аудіофайл та захистити його наклепавши на нього ЦВЗ - переходимо до кроку 5.

Якщо ні – процес закінчується.

Крок 5. Робота з веб-серісом:

Якщо користувач бажає завантажити свій файл він обирає вкладку “Додати” та завантажує у вікно свій файли формату WAVE. Після чого система додає цей файл до бази даних та автоматично додає до нього інформацію про користувача (автора).

На цьому процес закінчується.

Проаналізувавши алгоритм роботи користувача з сервісом створимо UML модель для нашого веб-сервісу (рис. 2.4).

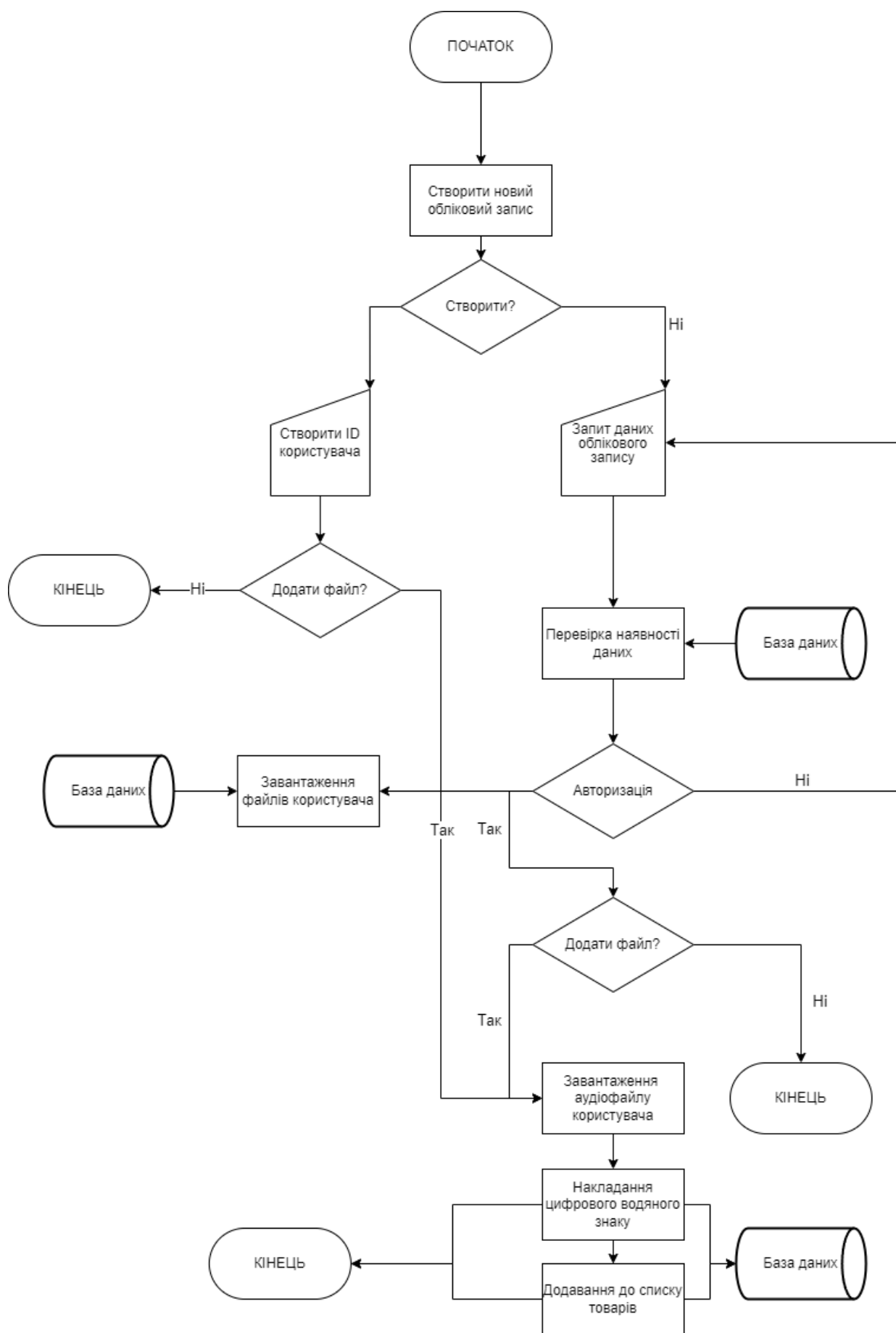


Рисунок 2.4 – Алгоритм роботи користувача із розробленим онлайн сервісом.

2.2 Вибір та інтеграція модулів захисту для веб-сервісу захисту аудіофайлів

Модель захисту для розроблюваного онлайн сервісу захисту аудіо передбачає захист даних користувачів на різних етапах взаємодії з додатком, таких як авторизація/реєстрація, збереження та обробка даних, накладання ЦВЗ і збереження аудіо.

Далі в роботі розглянемо детальніше кожен із запропонованих модулів захисту розроблюваного захищеного веб-сервісу захисту авторських прав аудіофайлів.

Застосування Laravel Passport. Laravel Passport - це офіційний пакет аутентифікації та авторизації для Laravel, одного з найпопулярніших PHP-фреймворків. Він надає простий та потужний спосіб реалізації аутентифікації на основі OAuth 2.0 та OpenID Connect (OIDC) в Laravel-додатках [23].

Далі розглянемо алгоритм процесу авторизації на основі використання Laravel Passport (див рис. 2.5) [24]:

Крок 1. Запит дозволу на авторизацію:

Клієнтський застосунок (даний веб-сервіс) звертається до власника ресурсу (користувача вбудованого в Laravel) для отримання дозволу на доступ до захищених ресурсів.

Крок 2. Надання дозволу на авторизацію:

Користувач надає дозвіл клієнтському застосунку на авторизацію від свого імені, наприклад, введенням логіна і пароля. Логін і пароль є грантом авторизації для клієнтського додатка.

Крок 3. Запит токена доступу:

Клієнтський застосунок надсилає запит на отримання токена доступу до сервера авторизації за допомогою ідентифікатора клієнта (`client_id`) та секретного ключа клієнта (`client_secret`). Окрім того, передаються реквізити користувача (логін і пароль), тип гранту (`grant_type`) та обсяг доступу (`scope`).

Крок 4. Створення та надання токена доступу:

Після успішної аутентифікації клієнта та перевірки реквізитів користувача сервер авторизації створює access-токен та необов'язковий ключ оновлення (refresh-токен). Токен доступу повертається клієнтському застосунку.

Крок 5. Доступ до захищених ресурсів:

Клієнтський застосунок може звертатися до захищених ресурсів (наприклад, захищеного Web API) та надавати токен доступу для авторизації. У випадку недійсного або простроченого токена сервер ресурсів поверне відповідь з кодом помилки 401, 403 або 498.

Крок 6. Надання даних додатку:

У випадку успішної аутентифікації та дійсності токена доступу сервер ресурсів надає клієнтському застосунку доступ до захищених даних або послуг.

При виборі Laravel Passport було проаналізовано документацію та визначені такі корисні функції для веб-сервісу захисту аудіофайлів

- OAuth 2.0 та OpenID Connect: Laravel Passport підтримує ці сучасні протоколи аутентифікації, вони в свою чергу легко інтегруються в систему веб-сервісу захисту аудіофайлів. Ці протоколи забезпечують безпеку передачі даних між клієнтом та сервером, а також гарантують, що тільки авторизовані користувачі можуть отримати доступ до захищених ресурсів.

- Шифрування даних: Laravel використовує симетричні алгоритми шифрування, такі як AES, що забезпечує надійне шифрування даних. Це гарантує, що конфіденційна інформація залишається захищеною від несанкціонованого доступу.

- HTTPS: Використання HTTPS для передачі даних захищає токени та інші чутливі дані від перехоплення та маніпуляцій під час передачі через мережу.

- XSS та CSRF-атаки: Laravel має вбудований захист від XSS (впровадження шкідливого коду) та CSRF (підробка запитів між сайтами), що забезпечує додатковий рівень безпеки для веб-додатків.

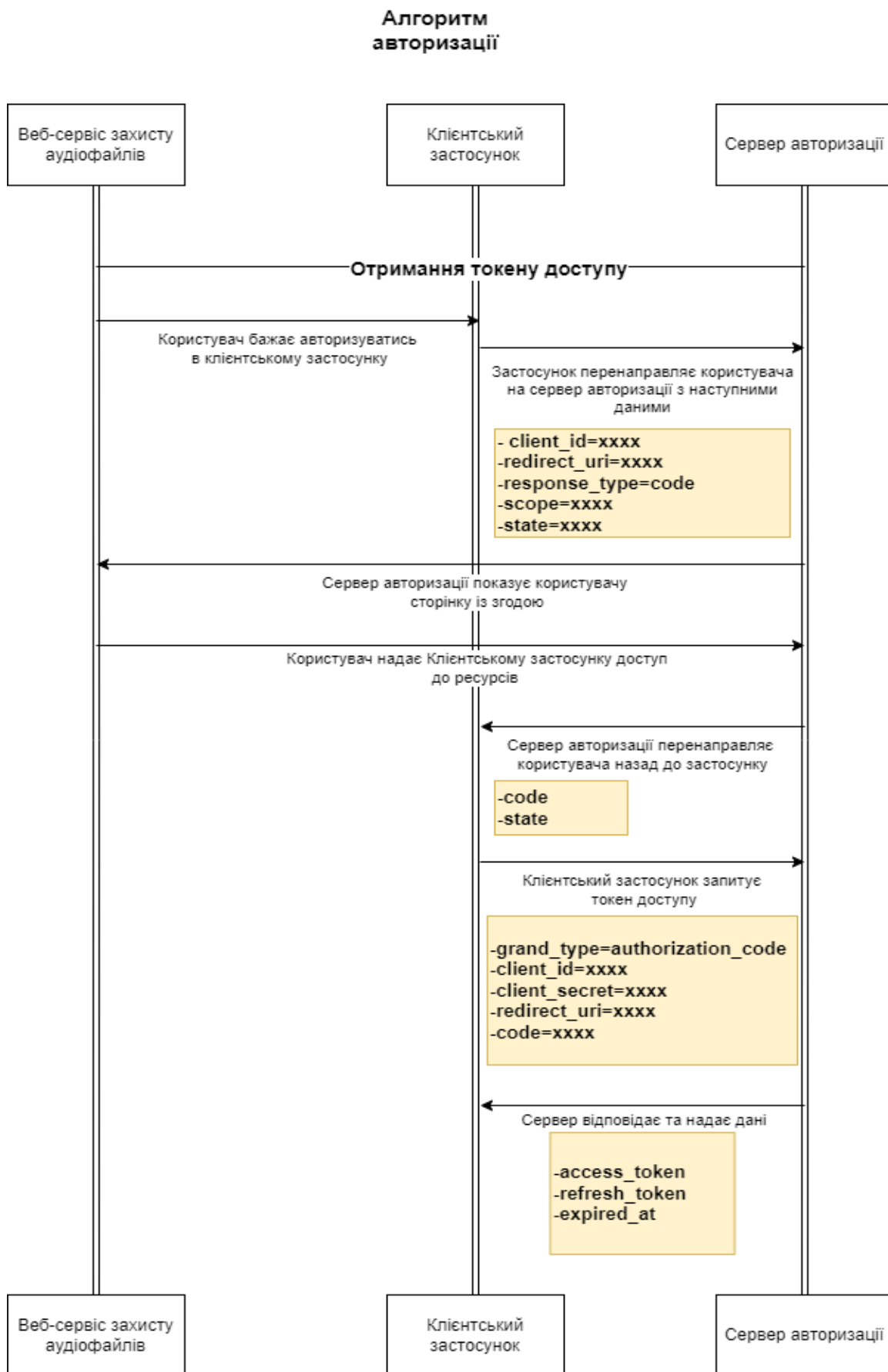


Рисунок 2.5 – Алгоритм авторизації у веб-сервісі за допомогою Laravel Passport
Розробка модулю двофакторної автентифікації

Для захисту онлайн сервісу буде розроблено ще модуль двохфакторної автентифікації за допомогою тимчасового паролю (One-time password).

ОТР (One-Time Password) — це одноразовий пароль, який використовується для автентифікації користувача тільки один раз. Він генерується автоматично і надсилається користувачеві через електронну пошту. Після введення правильного ОТР користувач отримує доступ до системи веб-сервісу [25].

Крок 1. Реєстрація користувача

Користувач заповнює форму реєстрації та вводить ім'я користувача, електронну пошту та пароль.

Система зберігає дані та пароль хешується перед збереженням у базу даних.

Створюється запис користувача в базі даних із введеними даними.

Крок 2. Вхід користувача

Користувач заповнює форму входу вводять електронну адресу та пароль.

Система перевіряє введені дані.

Перевіряє, чи існує користувач із введеним ім'ям користувача.

Перевіряє, чи правильний пароль шляхом порівняння збереженого хешу пароля.

Крок 3. Генерація та відправка ОТР

Генерується випадковий код (зазвичай 6-значний числовий код).

ОТР зберігається у базі даних разом з часом його генерації.

Система надсилає електронного листа з ОТР на адресу, вказану при реєстрації.

Крок 4. Введення ОТР користувачем

Користувач отримує електронний лист відкриває його і бачить ОТР.

Вводить отриманий код у відповідне поле на веб-сторінці.

Крок 5. Перевірка ОТР

Система перевіряє ОТР, знаходить запис користувача у базі даних за введеним ОТР.

Перевіряє, чи збігається введений код із збереженим у базі даних.

Перевіряє, чи не закінчився час дії ОТР (код дійсний 5-10 хвилин).

Якщо код правильний і дійсний, користувач отримує доступ до системи.

ОТР видається або позначається як використаний у базі даних.

Якщо код неправильний або прострочений, користувач отримує повідомлення про помилку і може спробувати ще раз.

Даний метод забезпечує усі потреби онлайн сервісу захисту аудіо через його простоту та ефективність.

Розробка захищеної бази даних.

Розробка програмного засобу для надання адміністративних послуг вимагає створення надійної та захищеної бази даних. У даному випадку для реалізації бази даних буде використано бібліотеку RedBeanPHP, яка дозволяє створювати, модифікувати та керувати базами даних у зручний та ефективний спосіб. RedBeanPHP є ORM (Object-Relational Mapping) бібліотекою для PHP, яка спрощує взаємодію з базами даних [26].

Для розроблюваного онлайн сервісу захисту аудіофайлів розробимо захищену базу даних, що є невід'ємною складовою даного програмного продукту.

Для створення бази даних потрібні такі сутності: КОРИСТУВАЧІ, АУДІОФАЙЛИ, КОРЗИНА, ДАНІ АВТОРИЗАЦІЇ, КОМЕНТАРІ.

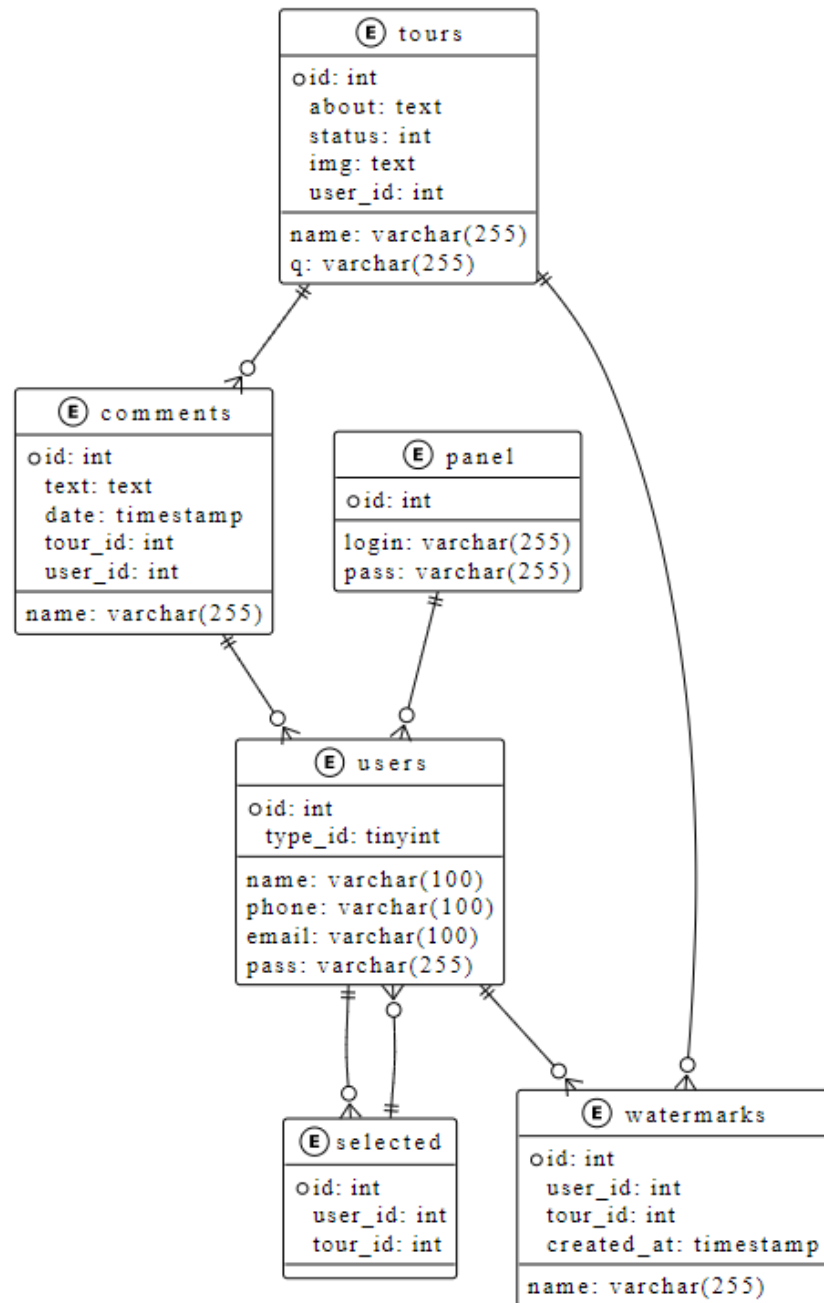


Рисунок 2.6 – UML діаграма розробленої захищеної бази даних веб-сервісу

Таким чином, можна описати такі властивості (атрибути) вказаних сутностей, що повністю розкриють їх зміст:

USERS (Id, Name, Phone, Email, Pass, Type_id);

TOURS (Id, Name, About, Status, q, img, user_id);

WATERMARKS (Id, User_id, Tour_id, Created_at);

PANEL (Id, Login, Pass);

COMMENTS (Id, Text, Date, Name, Tour_id, User_id);

SELECTED (Id, User_id, Tour_id);

При розробці бази даних для розроблюваного онлайн сервісу захисту аудіофайлів особлива увага надавалась захисту бази даних від SQL-ін'єкцій та XSS-атак. Далі в роботі опишемо детальніше особливості даних загроз та запропоновані методи захисту.

Розробка захисту веб-сервісу від Sql-ін'єкцій.

SQL-ін'єкції є однією з найпоширеніших вразливостей веб-додатків, які можуть призвести до компрометації бази даних. Щоб захистити веб-сервіс від таких атак, важливо використовувати безпечні методи взаємодії з базою даних. У цьому випадку ми будемо використовувати PDO та ORM RedBeanPHP, інтегровані у модель MVC [27].

RedBeanPHP – це ORM (Object-Relational Mapping) бібліотека для PHP, яка надає простий та гнучкий спосіб взаємодії з базою даних. Вона дозволяє працювати з базою даних за допомогою об'єктів, без необхідності писати SQL-запити вручну. RedBeanPHP автоматично створює таблиці та колонки в базі даних на основі властивостей об'єктів, що робить її дуже зручною для швидкої розробки.

PDO (PHP Data Objects) – це розширення PHP, яке забезпечує стабільний інтерфейс для доступу до баз даних. PDO підтримує підключення до MySQL, тому підходить для використання в розробці захисту для онлайн сервісу [28].

Для захисту від SQL-ін'єкцій RedBean використовує наступні можливості:

- Підготовлені запити: У підготовлених запитах SQL-код і дані передаються окремо. Це означає, що дані не можуть бути неправильно інтерпретовані як частина SQL-коду. Підготовлені запити створюються один раз, а дані передаються до них пізніше. Це дозволяє базі даних веб-сервісу розрізняти код і дані.

- Прив'язка параметрів: При використанні підготовлених запитів із прив'язкою параметрів PDO автоматично обробляє та екранує введені дані, що запобігає виконанню небезпечних SQL-інструкцій.

- Валідація даних. RedBean використовує базову валідацію даних, перш ніж вони будуть збережені у базі даних. Це допомагає перевірити, чи дані відповідають очікуваному формату і не містять шкідливих символів.

– RedBean автоматично створює таблиці в базі даних на основі моделей даних, що може запобігти деяким атакам, пов'язаним із використанням некоректних або недостатньо захищених таблиць.

Наприклад, при підготовці запиту **SELECT * FROM users WHERE email = :email** і прив'язці параметра **:email** користувацьке введення обробляється безпечно, навіть якщо користувач введе шкідливий код.

Застосування протоколів HTTPS та SSL. Протоколи HTTPS та SSL використовуються для забезпечення додаткового рівня безпеки при передачі даних від браузера користувача на сервер. Завдяки налаштуванню цих засобів на захищеному онлайн сервісі досягаються наступні переваги [29]:

– Шифрування: Дані передаються в зашифрованому вигляді, що унеможливорює зловмисникам перехопити інформацію, що обмінюється між відвідувачами та сервером. Це забезпечує конфіденційність даних.

– Аутентифікація: Користувачі можуть бути впевнені, що вони переходять на офіційний сайт компанії, а не на підроблений. SSL-сертифікат підтверджує справжність веб-сайту, забезпечуючи автентичність та довіру.

– Збереження даних: Протокол фіксує всі зміни даних, що передаються між браузером та сервером. Якщо зловмисник намагався зламати захист, це можна виявити за допомогою збережених даних про події.

Протокол HTTPS (Hypertext Transfer Protocol Secure) забезпечує безпечну передачу даних через Інтернет шляхом шифрування і аутентифікації. Ось схема його роботи:

– Ініціалізація зв'язку: Клієнт (наприклад, веб-браузер) встановлює з'єднання із сервером, ініціюючи HTTPS-запит.

– SSL Handshake (рукостискання SSL): Коли клієнт звертається до сервера за HTTPS-сторінкою, сервер відповідає SSL Handshake запитом. Під час цього процесу відбувається обмін інформацією між клієнтом і сервером для встановлення безпечного з'єднання.

– SSL сертифікат: Сервер надсилає клієнту свій SSL сертифікат для підтвердження своєї ідентичності. Цей сертифікат містить публічний ключ, який використовується для шифрування даних.

– Передача публічного ключа: Клієнт перевіряє SSL сертифікат сервера, переконуючись в його валідності та підписі відповідної авторитетної організації. Після цього клієнт генерує симетричний ключ шифрування та шифрує його за допомогою публічного ключа сервера.

– Встановлення безпечного каналу: Після успішної аутентифікації сервера та обміну ключами, клієнт і сервер встановлюють безпечний канал зв'язку, де дані будуть шифруватись за допомогою симетричного ключа.

– Передача даних: Після встановлення безпечного каналу клієнт і сервер можуть безпечно обмінюватися даними, які будуть шифруватись за допомогою симетричного ключа. Всі дані, відправлені через цей канал, будуть зашифровані і захищені від перехоплення.

– Завершення сесії: Після завершення обміну даними або після закінчення сесії з'єднання може бути закрито.

Отже, згідно поставленої задачі було розглянуто та проаналізовано всі модулі захисту веб-сервісу захисту авторських прав аудіофайлів.

2.3 Розробка модулю накладання стійкого цифрового водяного знаку

У попередньому підрозділі було відзначено, що для проектування та функціонування веб-сервісу, присвяченого захисту авторських прав на аудіофайли, важливо обрати ефективний метод накладання цифрових водяних знаків (ЦВЗ). З цією метою будуть розглянуті різні стеганографічні методи, щоб обрати найбільш відповідний для даного веб-сервісу. Вибір методу буде заснований на дослідженні їх ефективності, надійності та відповідності вимогам щодо захисту авторських прав та мінімізації впливу на якість аудіофайлів.

Розглянемо найпопулярніші методи:

Перший метод – це кодування найменш значущих бітів (LSB). Цей метод є найпростішим серед існуючих. Він полягає у заміні найменш значущого біту в

аудіосигналі на біт прихованого повідомлення. Такий підхід має високу пропускну здатність і мало помітний низькочастотний шум.

Другий метод - метод розширення спектра. Цей підхід схожий на методи приховування даних у нерухомих зображеннях. Повідомлення розподіляється по частотах несучого сигналу так, щоб співвідношення сигналу до шуму було дуже низьким. При цьому сигнал-контейнер обирається набагато більший, ніж секретне повідомлення.

Третій метод - приховування даних із використанням ехо-сигналу. Тут секретне повідомлення вбудовується у сигнал-контейнер через ехо-сигнал. Це досягається зміною параметрів ехо-сигналу, таких як початкова амплітуда, швидкість затухання та зсув. Цей метод складніший у реалізації, адже важко визначити потрібні значення параметрів.

Четвертий метод - метод фазового кодування. Він використовується для модифікації фази кожної частотної складової дискретного сигналу. Це досягається розбиттям вихідного сигналу на короткі сегменти і зміною фази кожного сегмента відповідно до повідомлення. Для збереження скритності повідомлення обчислюється різниця фаз між сусідніми сегментами [25].

Ефективність аудіо-стеганографії залежить від багатьох факторів, включаючи рівень стиснення аудіо, якість самого аудіо, специфікації кодеків та інші аспекти. Також важливо враховувати можливість втрати інформації під час передачі або обробки аудіо-сигналу.

Розглянемо вразливості та переваги всіх методів, наведені в таблиці 2.1, та визначимо відповідний метод для веб-сервісу.

Таблиця 2.1 – Вразливості та переваги стеганографічних методів

Метод	Вразливість	Перевага	Техніка приховування даних
LSB	Легко виділити	Простий та легкий	Найменш значущі біти (LSB)
Ехо-приховування	Низький рівень безпеки інформації та низька ємність	Не додає шуму в аудіо	Введення ехо-сигналу
Балансове кодування	Легко викрити	Міцніший за LSB	Зміна балансу LSB бітів
Дискретне вейвлет-перетворення	Можливість відновлення даних у разі втрати	Висока здатність до вбудовування та чіткості інформації	Перехід до області дискретно хвильового перетворення
Розподілений спектр	Займає багато діапазону	Найміцніший та якісніший	Розсіюється по всьому широкосмуговому спектру аудіосигналу
Фазове кодування	Низька ємність	Стійкий до обробки сигналу	Зміна фазових параметрів сигналу

Після ретельного аналізу таблиці, у якій були представлені різні методи накладання цифрових водяних знаків на аудіофайли, та врахування всіх їх переваг і недоліків, було прийнято рішення обрати метод фазового кодування для розробленого онлайн сервісу захисту аудіофайлів.

Цей вибір зумовлений його високою ефективністю у вбудовуванні інформації без помітного впливу на якість аудіофайлів, стійкості до обробки сигналу, а також його відповідністю вимогам щодо захисту авторських прав на контент. Такий метод забезпечує найвищий рівень конфіденційності та непомітності водяних знаків для користувачів веб-сервісу, що робить його найоптимальнішим вибором для досягнення цілей проекту.

Метод фазового кодування є одним із способів приховування даних у аудіосигналах, який ґрунтується на зміні фази кожної частотної складової сигналу.

Однією з переваг цього методу є стійкість до обробки сигналу, оскільки зміна фази може залишатися стабільною під час таких операцій, як стискання сигналу чи різноманітні ефекти обробки. Крім того, метод фазового кодування має високу

стійкість до видалення шуму, оскільки вбудована інформація може бути виявлена навіть після впливу на сигнал деякого рівня шуму.

Проте, основним недоліком методу кодування фази є низька ємність, оскільки кількість бітів, які можна приховати, обмежена кількістю фазових параметрів, які можна змінити без помітних змін у сприйнятті сигналу. Але не дивлячись на такий недолік для даної задачі не потрібна велика ємність, тому що інформації зберігатись буде не багато [30].

Для того щоб розробити програмний модуль для веб-сервісу потрібно розібрати алгоритм даного методу:

Крок 1. Розділення

Завантажене аудіо містить заголовок та дані які в ньому містяться, для початку вони розділяються та алгоритм переходить до кроку 2.

Крок 2. Семплування

Вихідний аудіосигнал розбивається на короткі сегменти, кожен з яких містить фіксовану кількість відліків.

Крок 3. Дискретне перетворення

Для кожного сегмента обчислюються фази частотних компонент за допомогою дискретного перетворення Фур'є (ДПФ).

Крок 4. Зміна фази

Потім змінюють фази частотних компонент сегментів відповідно до бітів повідомлення, яке потрібно приховати. Це може виконуватися, наприклад, шляхом додавання або віднімання фіксованих значень фази від кожного елементу ДПФ.

Крок 5. Приховування інформації.

Після зміни фаз сегментів формується новий аудіосигнал, який містить приховане повідомлення.

Крок 6. Інверсне дискретне перетворення.

Далі знову обчислюються фази частотних компонент за допомогою інверсного дискретного перетворення Фур'є (ІДПФ).

Крок 7. Результат.

Сигнал додається із заголовком файлу і в результаті отримано аудіо файл із прихованим повідомленням.

Розглянемо алгоритм дії цього методу згідно поставленої задачі веб-сервісу (рис. 2.5):

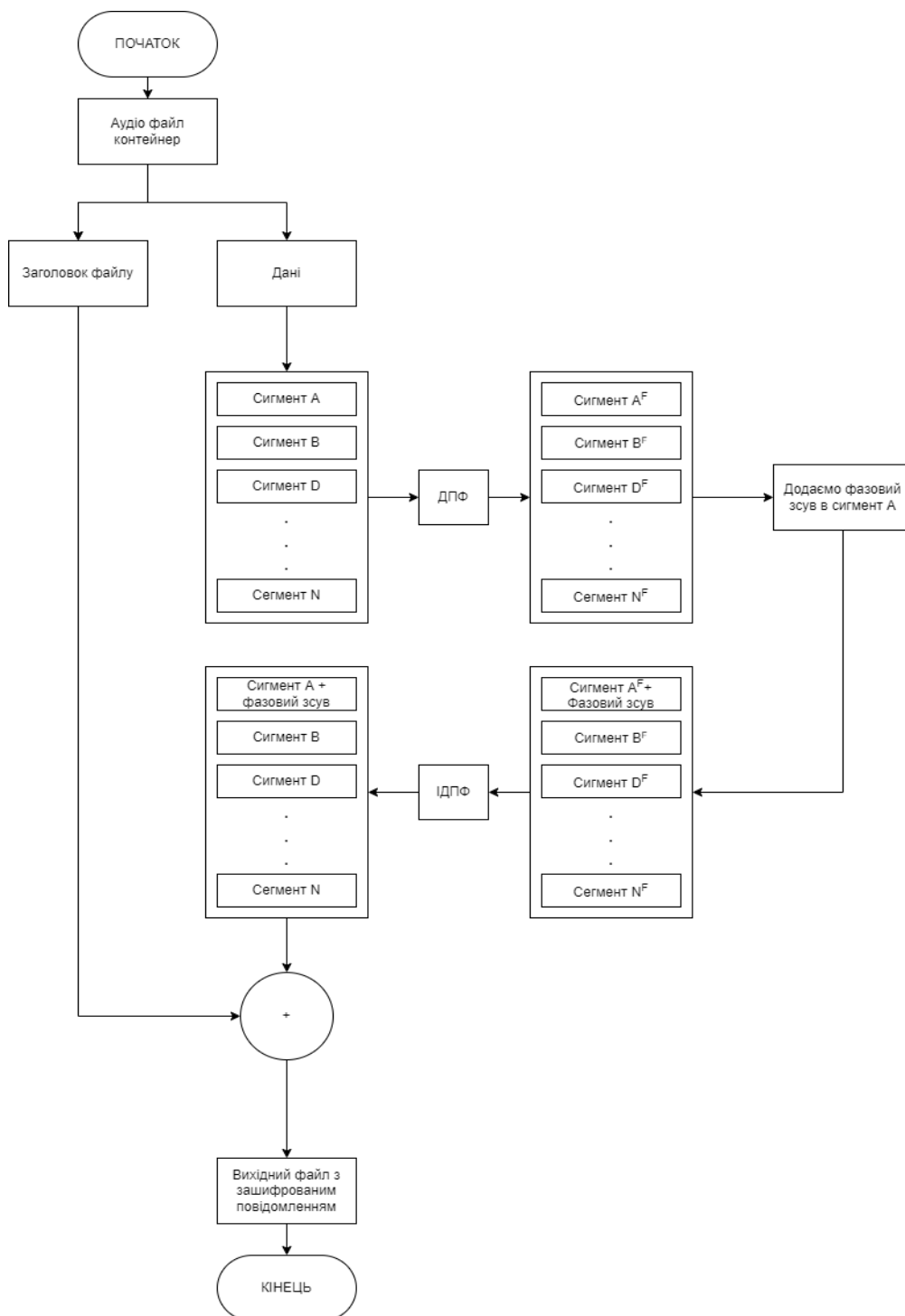


Рисунок 2.5 – Блок-схема алгоритму методу фазового кодування розробленого для веб-сервісу

2.4 Вибір середовища розробки

Після завершення розробки архітектури проектування на основі UML діаграм та обрання методу стеганографії для приховування інформації в аудіофайлах, наступним кроком є обґрунтований вибір середовища розробки та мови програмування для реалізації даного веб-сервісу. Важливо врахувати такі аспекти, як ефективність, швидкість розробки, масштабованість, підтримка засобів безпеки, доступність бібліотек та інструментів для роботи з аудіофайлами та обробки даних, а також можливість інтеграції з базою даних для зберігання та управління інформацією про авторські права та аудіофайли.

Виходячи із поставленої мети та задач роботи, для розробки захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками було обрані такі засоби програмування:

- PHP - це мова програмування, яка призначена для розробки веб-застосунків та динамічних веб-сайтів;
- MySQL - це безкоштовна реляційна система управління базами даних, яка використовується для зберігання та управління даними веб-додатків;
- PhpStorm - це інтегроване середовище розробки для PHP, яке надає широкий набір інструментів для зручного програмування, налагодження та відлагодження веб-застосунків на PHP.

Далі в підрозділі розглянемо детальніше наведені засоби програмування та проаналізуємо основні переваги, що зумовили їх вибір для даної роботи.

PHP – це мова програмування, яка призначена переважно для веб-розробки. Вона добре підходить для створення веб-сайтів середнього та невеликого розміру, таких як блоги, форуми, корпоративні веб-сайти. Використовуючи PHP, можна швидко розробляти прототипи та мінімізувати час від ідеї до реалізації завдяки великій кількості готових рішень та фреймворків [31].

Крім того, PHP має широкі можливості завдяки великій кількості розширень та бібліотек. Вона також має велику та активну спільноту розробників, що дозволяє легко знаходити ресурси та отримувати підтримку. PHP широко використовується

в популярних веб-платформах, таких як WordPress, Drupal, Joomla, що підтверджує його ефективність та відповідність для різноманітних завдань веб-розробки.

Вона призначена для створення HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов для веб-розробки PHP підтримується більшістю хостинг-провайдерів PHP є програмним проектом з відкритим вихідним кодом.

PHP інтерпретується веб-сервером і перетворюється на HTML-код, який надсилається клієнтській стороні; на відміну від мови сценаріїв JavaScript, код PHP не видно користувачеві. Це є перевагою з точки зору безпеки, але знижує інтерактивність сторінки. Ключові особливості PHP [32]:

- Вбудована підтримка веб-розробки. PHP розроблений спеціально для веб-розробки. Він має вбудовані функції для роботи з HTTP-запитами, формами, сесіями, кукісами та іншими веб-технологіями.

- Легкість програмування адже PHP має простий синтаксис, що робить його досить доступним для новачків. Велика спільнота розробників PHP також забезпечує багато матеріалів для вивчення та підтримки.

- Підтримка баз даних. PHP має вбудовану підтримку для багатьох систем керування базами даних (СКБД), таких як MySQL, PostgreSQL, SQLite і багатьох інших. Це дозволяє легко працювати з даними в вашому веб-додатку.

- Динамічність дозволяє легко генерувати динамічний вміст веб-сторінок. Ви можете вбудовувати PHP-код безпосередньо в HTML-сторінки, щоб створювати персоналізований контент для користувачів.

- Розширюваність - мова має широкий вибір розширень (extensions) і фреймворків, які полегшують розробку складних веб-додатків і забезпечують більшу функціональність.

У контексті створення веб-сервісу для захисту авторських прав на аудіофайли PHP виявляється відмінним вибором для програмної розробки. Висока швидкість розробки, широкі можливості для інтеграції з базами даних та підтримка різноманітних фреймворків роблять PHP ефективним інструментом для реалізації

функціональності веб-сервісу. Його простий синтаксис і доступність для новачків також сприяють швидкому старту та розвитку проекту.

Узагальнюючи, PHP відповідає вимогам ефективності, масштабованості та гнучкості, необхідних для успішної розробки та функціонування веб-сервісу захисту авторських прав на аудіофайли.

Однією з переваг мови програмування PHP є широке використання мови для розробки веб-додатків, що забезпечує гнучкість та продуктивність у розробці. Його простий синтаксис дозволяє швидко створювати та підтримувати веб-додатки.

Функції шифрування в PHP дозволяють забезпечити безпеку даних шляхом захисту їх від несанкціонованого доступу та перегляду. Деякі з найбільш важливих функцій шифрування в PHP:

- PHP має вбудовані функції для хешування паролів, такі як `password_hash()` та `password_verify()`. Вони дозволяють зберігати паролі у вигляді хешів, що робить їх важкими для відновлення навіть у випадку злому бази даних.

- PHP надає доступ до різних алгоритмів шифрування, таких як AES (Advanced Encryption Standard), DES (Data Encryption Standard) та RSA (Rivest–Shamir–Adleman). Це дозволяє зашифровувати конфіденційні дані перед їх збереженням в базі даних або передачею по мережі.

- PHP може використовувати HTTPS для зашифрованої передачі даних між сервером та клієнтом за допомогою SSL (Secure Sockets Layer) або TLS (Transport Layer Security) протоколів. Це забезпечує безпеку під час передачі конфіденційних даних, таких як паролі або фінансова інформація.

Вона є вільним програмним забезпеченням з відкритим вихідним кодом, що дозволяє розробникам досліджувати, аудитувати та вдосконалювати його функції безпеки. Для роботи може використовувати MySQL. Ця база даних є однією з найпопулярніших відкритих систем керування базами даних. Її надійність, продуктивність та широкі можливості роблять її відмінним вибором для багатьох проектів. MySQL забезпечує швидкий доступ до даних та ефективне управління базами даних будь-якого розміру. Крім того, її сумісність з PHP дозволяє легко

інтегрувати бази даних MySQL у веб-додатки, що робить їх швидкими та надійними.

MySQL - це безкоштовна реляційна система управління базами даних, розроблена та підтримується корпорацією Oracle. Розглянемо переваги та особливості цієї системи управління базами даних для даного онлайн сервісу.

Універсальність та функціональність:

- MySQL надає засоби для створення, налаштування та адміністрування баз даних різного розміру та складності.
- Система пропонує різні типи таблиць, такі як MyISAM та InnoDB, з різними можливостями, що розширюють функціональність та ефективність.

Інструменти розробки:

- MySQL має інтегровані засоби для роботи з базами даних, включаючи редактори запитів, візуальні засоби керування об'єктами та інші інструменти, які спрощують розробку та адміністрування.

Надійність та продуктивність:

- MySQL відомий своєю стабільністю та продуктивністю, що дозволяє використовувати його для розробки навіть великих та вимогливих додатків.

MySQL є потужним та універсальним інструментом для управління базами даних, який відповідає вимогам розробки БД для веб-сервісу. Його широкий функціонал, надійність та підтримка роблять його відмінним вибором для розробки та адміністрування баз даних [33-35].

У сучасному світі розробка програмного забезпечення стала важливою складовою бізнесу та технологічного прогресу. Вибір мови програмування впливає на ефективність розробки, зручність підтримки та швидкість виконання програми. При виборі мови програмування та системи керування базами даних (СКБД) необхідно враховувати ряд факторів, таких як:

- Продуктивність
- Надійність
- Швидкість розробки

– Сумісність із завданнями проекту.

PHP разом з MySQL гарний вибір для створення ефективних, надійних та продуктивних веб-додатків, що відповідають потребам сучасного бізнесу та користувачів.

Отже, враховуючи функціональні можливості обраних мов програмування, їх доступність та зрозумілість при наявності відповідних навичок, ми плануємо використовувати ці засоби для реалізації програмної частини даного проекту.

2.5 Висновки до розділу 2

Мова програмування PHP була обрана для розробки сервісу завдяки своїй широкій популярності та використанню у веб-розробці. Вона володіє високою швидкістю виконання коду, має велику спільноту розробників та надає доступ до багатьох готових рішень і фреймворків, що спрощують розробку та підтримку проектів.

Архітектура сервісу ґрунтується на принципі клієнт-серверної взаємодії в стилі MVC (Model-View-Controller). Серверна частина, написана на мові програмування PHP, відповідає за обробку запитів користувачів, зберігання даних та відправку їх в базу даних.

Вибір методу шифрування аудіофайлів та накладання водяного знаку ґрунтується на критеріях ефективності та безпеки. Для забезпечення конфіденційності та цілісності даних був обраний метод фазового кодування, який застосовується на етапі завантаження аудіофайлу та його зберігання в базі даних.

Метод фазового кодування виявився не тільки ефективним, але й надійним. Він забезпечує захист від несанкціонованого використання аудіофайлів на всіх етапах обміну даними, зберігаючи конфіденційність інформації.

Таким чином, вибір PHP для серверної розробки разом з методом фазового кодування аудіофайлів відповідає вимогам ефективності, безпеки та забезпечує надійність обробки даних у розробленому сервісі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

В даному розділі зосередимося на програмній реалізації обраних підходів та концепцій, що були описані та проаналізовані у попередніх розділах. Буде розроблено інтерфейс веб-сервісу, модулі захисту а також модуль накладання ЦВЗ.

3.1 Розробка користувацького інтерфейсу

Розробка інтерфейсу користувача – це процес створення візуальної та функціональної складової сторінки, яка буде доступна для користувачів додатку. Інтерфейс користувача (UI) охоплює не лише зовнішній вигляд сторінки, але й її інтерактивність. До елементів UI відносяться кольори, які бачать користувачі, тексти, які вони читають, кнопки, на які натискають, і анімації, з якими вони взаємодіють. Відповідальність за ці елементи лежить на дизайнері інтерфейсу користувача.

Для створення зручного та інтуїтивно зрозумілого інтерфейсу слід дотримуватись декількох важливих принципів:

- Консистентність. Використання однакових елементів і стилів на всіх сторінках додатку допомагає користувачам швидко адаптуватися до інтерфейсу та легше взаємодіяти з ним.

- Зрозумілість. Інтерфейс повинен бути інтуїтивно зрозумілим, щоб користувачі могли легко знайти потрібні функції без додаткових інструкцій.

- Простота. Слід уникати перевантаження інтерфейсу зайвими елементами. Кожен елемент повинен мати чітке призначення і бути необхідним для користувача.

- Зворотний зв'язок. Інтерфейс повинен надавати користувачам зворотний зв'язок на їхні дії. Це можуть бути повідомлення про помилки, сповіщення про успішні дії або анімації, що вказують на поточний стан системи.

- Адаптивність. Інтерфейс повинен коректно відображатися на різних пристроях і бути зручним для користувачів як на великих екранах комп'ютерів, так і на мобільних пристроях.

– Доступність. Інтерфейс повинен бути доступним для користувачів з різними можливостями, включаючи тих, хто має обмеження зору або моторики. Використання правильних кольорових контрастів, масштабованого тексту та підтримка клавіатурної навігації сприяють досягненню цієї мети.

Дотримання цих принципів допоможе створити інтерфейс, який буде не тільки привабливим, але й зручним та ефективним у використанні.

Враховуючи, що розроблюваний веб-сервіс буде використовуватись в сфері державних послуг – його інтерфейс повинен бути яскравим, унікальним та інтуїтивно зрозумілим для типового користувача. З даних причин, в розроблюваному додатку будуть використовуватись яскраві кольори та елементи анімації.

Розглянемо проектування деяких вікон додатку для розуміння загальної його структури.

Сторінка авторизації / реєстрації користувача – її основне призначення – це розміщення полів для введення даних авторизації користувачем та перевірка доступності його облікового запису.

На даній сторінці у верхній стрічці меню будуть розміщені функціональні кнопки «Головна», «Товари», «Контакти» та «Особистий кабінет».

Основна частина сторінки – це назва сторінки та інструкція для користувача. Далі буде розміщено поле для введення електронної пошти користувача, нижче буде розміщено поле для введення паролю та функціональна кнопка «Зареєструватись» або

«Авторизуватись». Окрім того, передбачені ситуативні надписи «Пароль введено невірно», «Поле обов'язкове до заповнення», «Обліковий запис з такими даними не знайдено» і т.д. Зразок даної сторінки наведено на рис. 3.1.

Рисунок 3.1 – Макет авторизації

Згідно поставленої задачі має відбуватись також двофакторна автентифікація за допомогою Laravel Passport.

Вікно клієнтського застосунку дуже лаконічне та зрозуміле (див рис 3.2)

Authorization Request

My Client is requesting permission to access your account.

Authorize

Cancel

Рисунок 3.2 – Макет авторизації Laravel Passport

Після авторизації йде головна сторінка яка має наступний вигляд (рис. 3.3).

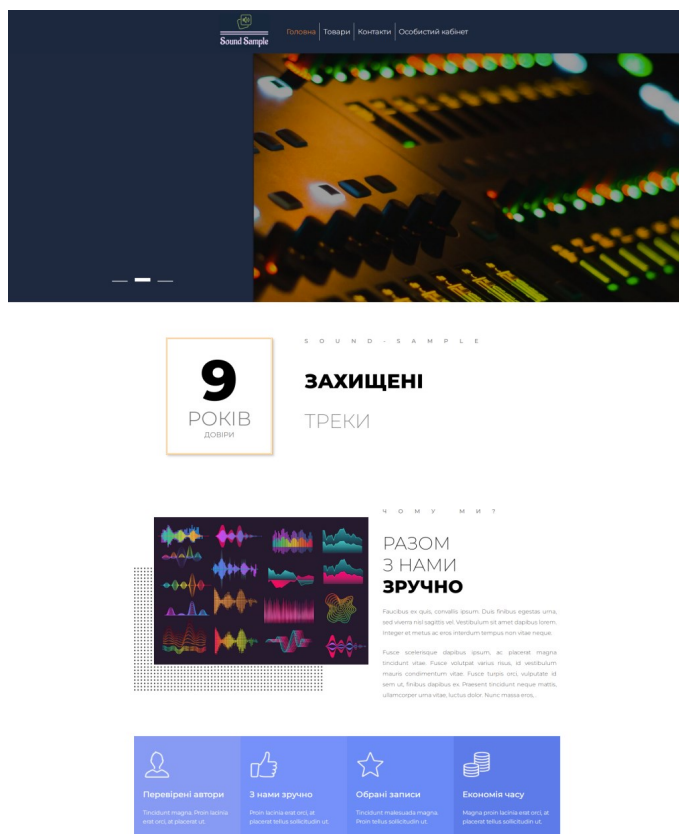


Рисунок 3.3 – Макет головної сторінки веб-сервісу

Далі користувач отримує можливість переглядати товари, коментувати, завантажувати аудіо та знайти контакти (див. рис. 3.4, 3.5, 3.6).

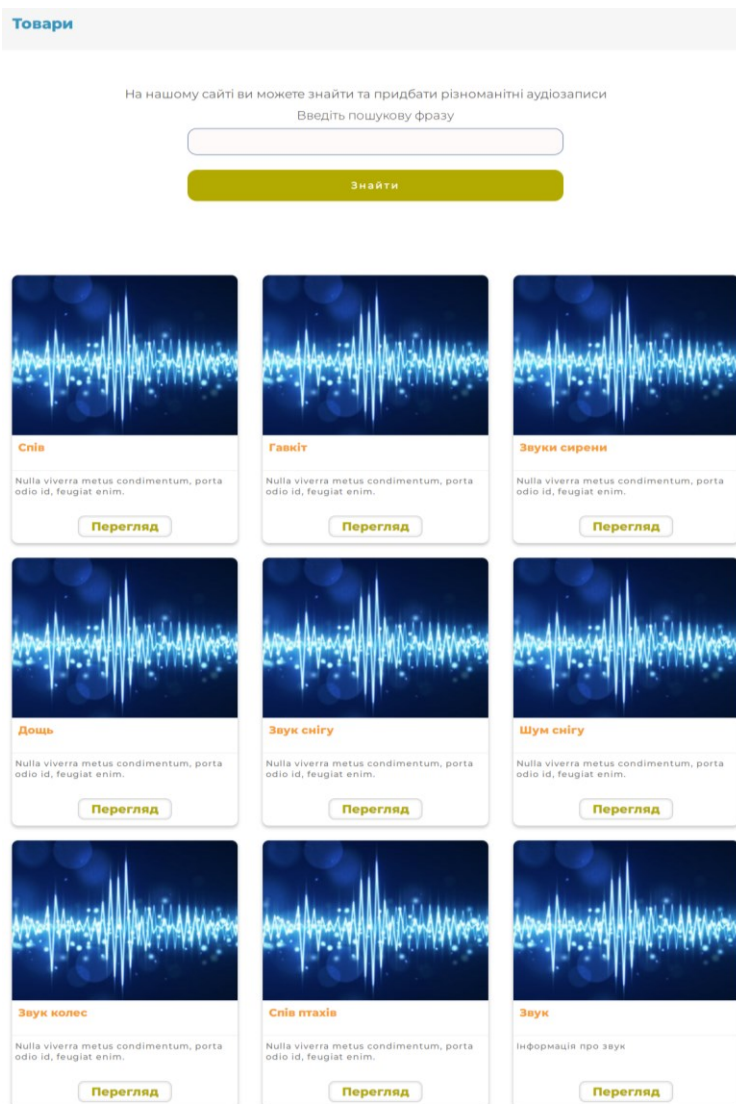


Рисунок 3.4 – Макет сторінки перегляду товарів

Коментарі

Автор: Юрій Петренко	Дата: 2024-05-10 21:40:41
Коментар	
Залишити відгук	
Надіслати	

Рисунок 3.5 – Макет полів для додавання коментарів

Додати аудіо

Назва запису

Детальна інформація

Короткий опис

Завантажте аудіо

Выберите файл Файл не выбран

Зберегти

Рисунок 3.6 – Макет сторінки додавання аудіо

Ім'я

Ваше ім'я

Телефон

Контактний номер телефону

Пошта

Електронна пошту

Повідомлення

Введіть ваше повідомлення

Надіслати

Рисунок 3.7 – Макет сторінки контактів

Даний інтерфейс відповідає усі вище зазначеним вимогам до веб-сервісу захисту аудіо та має яскравий анімований інтерфейс.

Усі функціональні та спроектовані в другому розділі сторінки були розроблені та успішно додані.

3.2 Програмна реалізація алгоритму накладання ЦВЗ та модулів захисту

У даному сервісі повинен бути присутній модуль захисту який додає до аудіофайлу інформацію про автора і покупця але додатково кожен аудіофайл

повинен бути помічений таким чином що б можна було виявити з якого саме сервісу (веб-сайту) він був завантажений. Тому реалізуємо алгоритм вбудовування стійкого цифрового водяного знаку в аудіофайл.

Опишемо усі функції модулю накладання:

Функція `readAudioSamples` виконує читання та розбивання аудіосемплів з WAV-файлу:

- Аналізує WAV-файл з використанням бібліотеки `getID3`.
- Відкриває WAV-файл для читання в режимі `rb` (читання в бінарному режимі).
- Переміщається до початку аудіоданих у файлі (зміщення даних аудіо).
- Читає семпли у форматі `float` з файлу, доки не досягне кінця файлу.
- Повертає масив із зчитаними семплами.

Ось відповідний блок коду:

```
<?php
require 'vendor/autoload.php';
use Getid3\Getid3;
use Getid3\Write\Writer;
function readAudioSamples($path) {
    $getID3 = new Getid3();
    $fileInfo = $getID3->analyze($path);
    $samples = [];
    $waveFile = fopen($path, 'rb');
    fseek($waveFile, $fileInfo['avdataoffset']);
    while (!feof($waveFile)) {
        $sample = unpack('f', fread($waveFile, 4))[1];
        $samples[] = $sample;
    }
    fclose($waveFile);
    return $samples;
}
```

Функція `writeAudioSamples`

Запис аудіосемплів у новий WAV-файл:

- Формує заголовок WAV-файлу (RIFF header) з необхідними параметрами (моно, 44100 Гц).
- Відкриває новий WAV-файл для запису в режимі `wb` (запис у бінарному режимі).
- Записує заголовок у файл.
- Записує семпли в форматі `float` у файл.
- Закриває файл після запису.

Ось відповідний блок коду:

```
function writeAudioSamples($path, $samples) {
    $header = pack('N', 0x52494646) // 'RIFF'
        . pack('V', 36 + count($samples) * 4) // file size
        . pack('N', 0x57415645) // 'WAVE'
        . pack('N', 0x666D7420) // 'fmt '
        . pack('V', 16) // PCM chunk size
        . pack('v', 1) // PCM format
        . pack('v', 1) // mono
        . pack('V', 44100) // sample rate
        . pack('V', 44100 * 4) // byte rate
        . pack('v', 4) // block align
        . pack('v', 32) // bits per sample
        . pack('N', 0x64617461) // 'data'
        . pack('V', count($samples) * 4); // data size
    $waveFile = fopen($path, 'wb');
    fwrite($waveFile, $header);
    foreach ($samples as $sample) {
        fwrite($waveFile, pack('f', $sample));
    }
    fclose($waveFile);
}
```

Функція stringToBinary

Конвертація текстового рядка в бінарний рядок:

- Ініціалізує порожній рядок для зберігання бінарного представлення.
- Проходить через кожен символ у рядку.
- Конвертує кожен символ у 16-бітове бінарне представлення.
- Додає бінарне представлення кожного символу до загального бінарного рядка.
- Повертає бінарний рядок.

Ось відповідний блок коду:

```
function stringToBinary($data) {
    $binary = '';
    for ($i = 0; $i < strlen($data); $i++) {
        $binary .= str_pad(decbin(ord($data[$i])), 16, '0', STR_PAD_LEFT);
    }
    return $binary;
}
```

Функція encodePhase

Кодування фазової інформації в аудіосемплах:

- Конвертує вхідний текстовий рядок у бінарне представлення.
- Додає метадані (довжину бінарного рядка як 32-бітове ціле число) до початку бінарного рядка.

- Обчислює кількість бітів для кодування.
- Використовує статичне значення Δphase для зміщення фази.
- Проходить через кожен семпл і модифікує його фазу відповідно до поточного біта в бінарному рядку.

- Зберігає модифіковані семпли.

Функція `decodePhase`

Декодування фазової інформації з аудіосемплів:

- Перевіряє, чи достатньо довжини вхідних семплів для декодування метаданих.

- Ініціалізує рядок для зберігання метаданих.
- Проходить через перші 32 семпли, щоб отримати метадані (довжину бінарного рядка).

- Конвертує метадані в довжину бінарного рядка.
- Перевіряє, чи достатньо семплів для декодування даних.
- Проходить через семпли, щоб декодувати фазову інформацію.
- Повертає декодований бінарний рядок.

Ось відповідний блок коду:

```
function encodePhase(&$samples, $data) {
    $binaryData = stringToBinary($data);
    $metadata = str_pad(decbin(strlen($binaryData)), 32, '0', STR_PAD_LEFT);
    $binaryData = $metadata . $binaryData;
    $bitLength = strlen($binaryData);
    $sampleCount = min($bitLength, count($samples));
    $deltaPhase = pi() / 8;
    for ($i = 0; $i < $sampleCount; $i++) {
        $sample = $samples[$i];
        $phase = acos($sample);
        if ($binaryData[$i] == '1') {
            $phase += $deltaPhase;
        } else {
            $phase -= $deltaPhase;
        }
        $samples[$i] = cos($phase);
    }
}

$audioPath = 'input.wav';
$samples = readAudioSamples($audioPath);
echo "Enter the string: ";
$data = trim(fgets(STDIN));
if (count($samples) < strlen($data) * 32) {
    echo "Not enough audio samples to encode the data.\n";
    exit; }
```

Також опишемо функцію створення нових користувачів.

Фрагмент коду є частиною функції яка, виконує обробку даних, введених користувачем, при створенні нового користувача на сайті. Розглянемо кожний рядок цього коду:

- `$name = safe($_POST['name']);` - звертаємося до POST-змінної name із форми, яка відправляє дані на сервер, та очищуємо її від можливих небажаних символів за допомогою функції `safe`.

- Аналогічно отримуємо інші дані: `email`, `phone`.

```
public function create(){
    $name = safe($_POST['name']);
    $email = safe($_POST['email']);
    $phone = safe($_POST['phone']);
    $pass = $_POST['pass'];
    $pas = $_POST['pas'];
```

- `$pass = $_POST['pass'];` - отримуємо пароль, введений користувачем, без очищення.

- `$pas = $_POST['pas'];` - отримуємо підтвердження пароля, також без очищення.

- `$user = $this->findOne( 'users', 'email = ?', [ $email ] );`: Звертаємося до бази даних і перевіряємо, чи існує вже користувач з введеним `email`.

- Проводяться перевірки на наявність помилок:

- Перевіряється, чи існує користувач з введеним `email`.

```
$user = $this->findOne( 'users', 'email = ?', [ $email ] );
if( isset($user) ){
    $_SESSION['form_err'][] = 'Вказаний email вже використовується.';
}
```

- Перевіряється, чи співпадають введені паролі.

- Перевіряється, чи пароль має не менше 6 символів.

- Перевіряється, чи всі поля форми заповнені.

- Якщо є помилки, вони зберігаються в масиві `$_SESSION['form_err']`, після чого користувач перенаправляється на іншу сторінку і виконання функції завершується.

- Якщо помилок немає, створюється новий запис в таблиці бази даних `users`, у якому зберігаються введені дані користувача, включаючи зашифрований пароль.

Після збереження даних до бази даних, встановлюється сесія для зареєстрованого користувача, після чого користувач перенаправляється на іншу сторінку.

```
public function create(){
    $name = safe($_POST['name']);
    $email = safe($_POST['email']);
    $phone = safe($_POST['phone']);
    $pass = $_POST['pass'];
    $pas = $_POST['pas'];

    $user = $this->findOne( 'users', 'email = ?', [ $email ] );
    if( isset($user) ){
        $_SESSION['form_err'][] = 'Вказаний email вже використовується.';
    }
    if( $pass != $pas ){
        $_SESSION['form_err'][] = 'Паролі не співпадають.';
    }
    if( mb_strlen($pass) < 6 ){
        $_SESSION['form_err'][] = 'Пароль повинен містити не менше 6 символів.';
    }
    if( empty($pas) || empty($pass) || empty($phone) || empty($email) ){
        $_SESSION['form_err'][] = 'Заповніть всі поля';
    }

    if( isset($_SESSION['form_err']) ){
        redirect();
        return;
    }
    $item = $this->create( 'users' );
    $item->name = $name;
    $item->email = $email;
    $item->phone = $phone;
    $item->pass = md5($pass);
    $this->store( $item );
    $_SESSION['user_id'] = $item->id;
    $_SESSION['login'] = 'user'; }
```

Ця функція забезпечує метод верифікації користувачів.

Дані про покупця не містять конфіденційної інформації але всеодно не повинні бути доступні у відкритому доступу тому перед додаванням їх до файлу проведемо нескладне шифрування.

Для цього розробимо алгоритм шифрування, який використовує заміну символів згідно з певним ключем.

```
public function subEncrypt($text, $key)
{
    $encryptedText = '';
    $keyLength = strlen($key);
    $textLength = strlen($text);
    for ($i = 0; $i < $textLength; $i++) {
```

```

    $position = ord($text[$i]) % $keyLength;
    $encryptedText .= $key[$position];
}
return $encryptedText;
}
public function subDecrypt($encryptedText, $key)
{
    $decryptedText = "";
    $keyLength = strlen($key);
    $textLength = strlen($encryptedText);
    for ($i = 0; $i < $textLength; $i++) {
        $position = strpos($key, $encryptedText[$i]);
        $decryptedText .= chr($position);
    }
    return $decryptedText;
}

```

У цьому алгоритмі кожен символ вхідного тексту замінюється на відповідний символ з ключа. Ключ визначає відображення символів. Це досить простий але ефективний алгоритм захисту.

Реалізація модулю авторизації Laravel Passport.

Після встановлення даної бібліотеки потрібно провести ряд налаштування перед повним використанням.

Налаштування моделі User:

Додаймо трейт HasApiTokens до вашої моделі User. Це надасть вашій моделі методи Passport, необхідні для роботи з токенами:

```

use Laravel\Passport\HasApiTokens;
class User extends Authenticatable
{
    use HasApiTokens, Notifiable;
}

```

Налаштування аутентифікації API:

Налаштуємо драйвер охорони для API, щоб використовувати токени Passport:

```

'api' => [
    'driver' => 'passport',
    'provider' => 'users',
],

```

Захист API маршрутів:

Тепер можна захистити API маршрути, використовуючи middleware

```

auth:api:
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

```

Маршрути Passport:

Далі потрібно додати маршрути Passport:

```
use Laravel\Passport\Passport;
public function boot()
{
    $this->registerPolicies();
    Passport::routes();
}
```

Налаштування контролерів для авторизації:

Додайте маршрути для реєстрації та авторизації користувачів:

```
Route::post('register', 'AuthController@register');
Route::post('login', 'AuthController@login');
```

Далі створіть контролер AuthController:

```
php artisan make:controller AuthController
```

У AuthController, реалізуємо методи для реєстрації та входу користувачів:

```
namespace App\Http\Controllers;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Validator;
class AuthController extends Controller
{
    public function register(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:8',
        ]);

        if ($validator->fails()) {
            return response()->json($validator->errors(), 400);
        }

        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => bcrypt($request->password)
        ]);

        $token = $user->createToken('LaravelAuthApp')->accessToken;

        return response()->json(['token' => $token], 200);
    }

    public function login(Request $request)
    {
        $credentials = $request->only('email', 'password');

        if (Auth::attempt($credentials)) {
            $user = Auth::user();
            $token = $user->createToken('LaravelAuthApp')->accessToken;
            return response()->json(['token' => $token], 200);
        }
    }
}
```

```

    } else {
        return response()->json(['error' => 'Unauthorised'], 401);
    }
}
}

```

3.3 Тестування захищеного веб-сервісу

На головній сторінці знаходиться навігаційне меню а також додаткова інформація про сервіс.

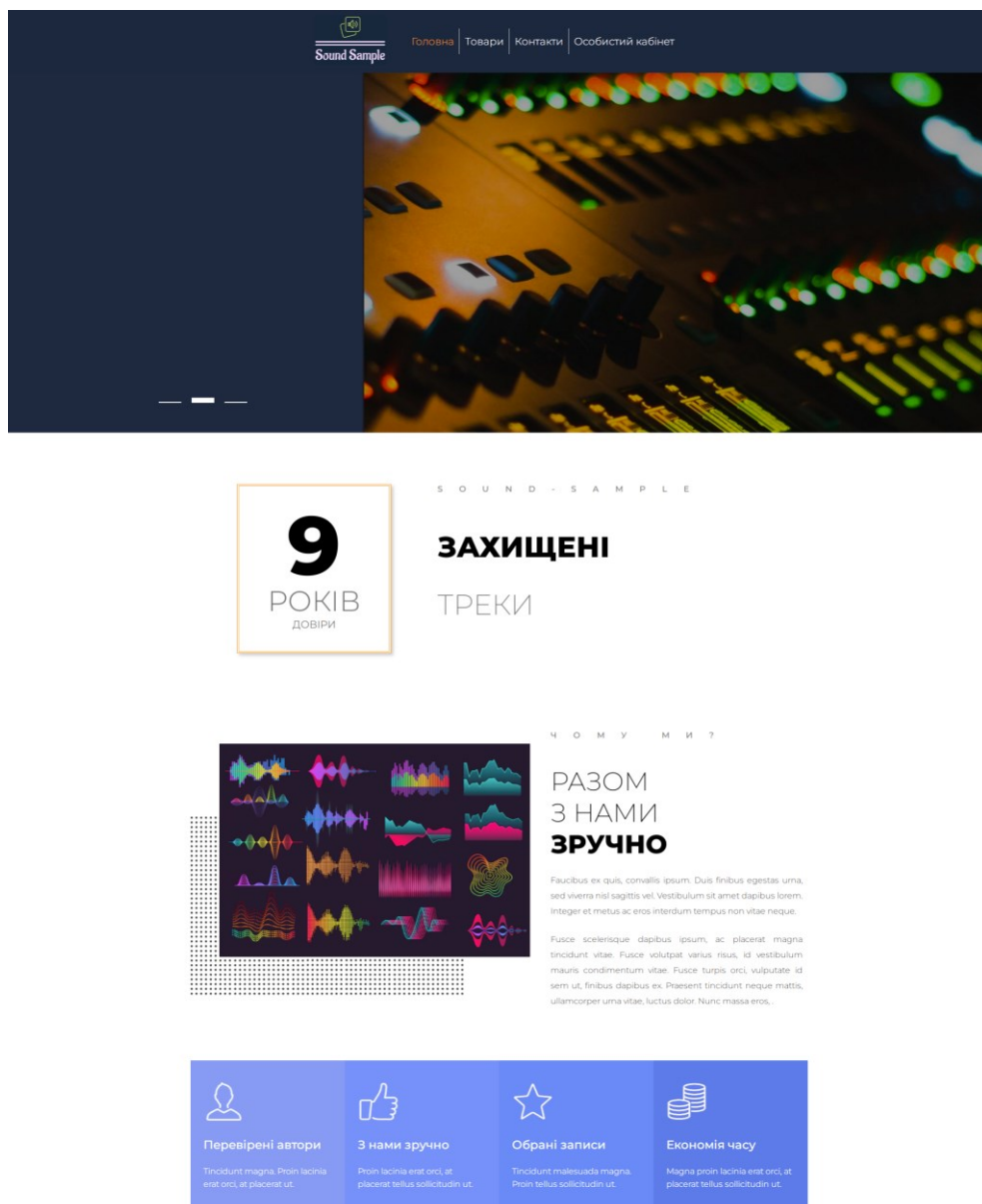


Рисунок 3.8 – Частина головної сторінки

В нижній частині сторінки знаходиться форма для зручності звернення до адміністрації сайту.

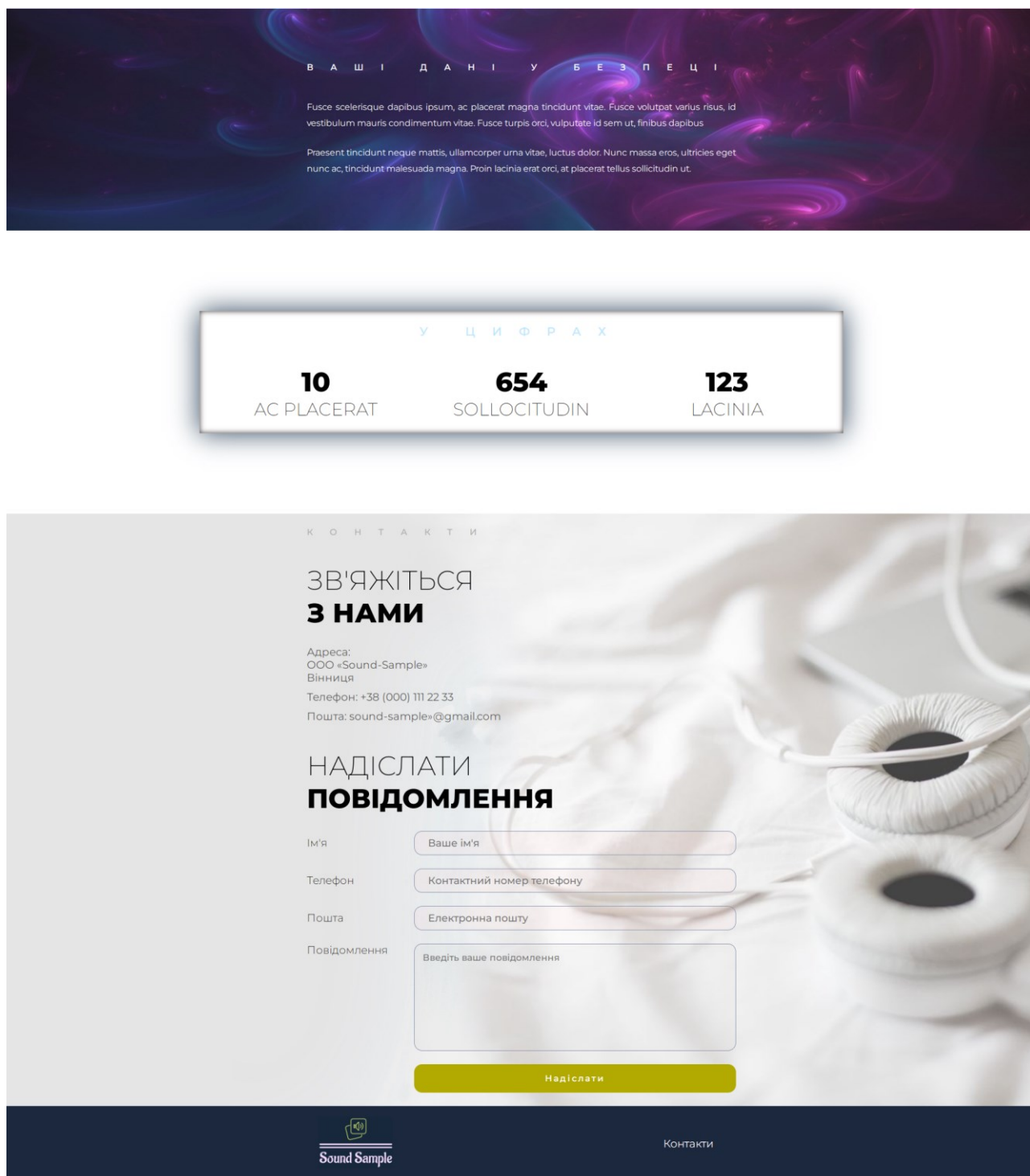


Рисунок 3.9 – Нижня частина головної сторінки

Для входу в особистий кабінет необхідно заповнити форму «логіну» або зареєструватися на сайті.



Головна | Товари | Контакти | **Особистий кабінет**

Вхід до особистого кабінету

Email

Пароль

Увійти

[Реєстрація нового користувача](#)



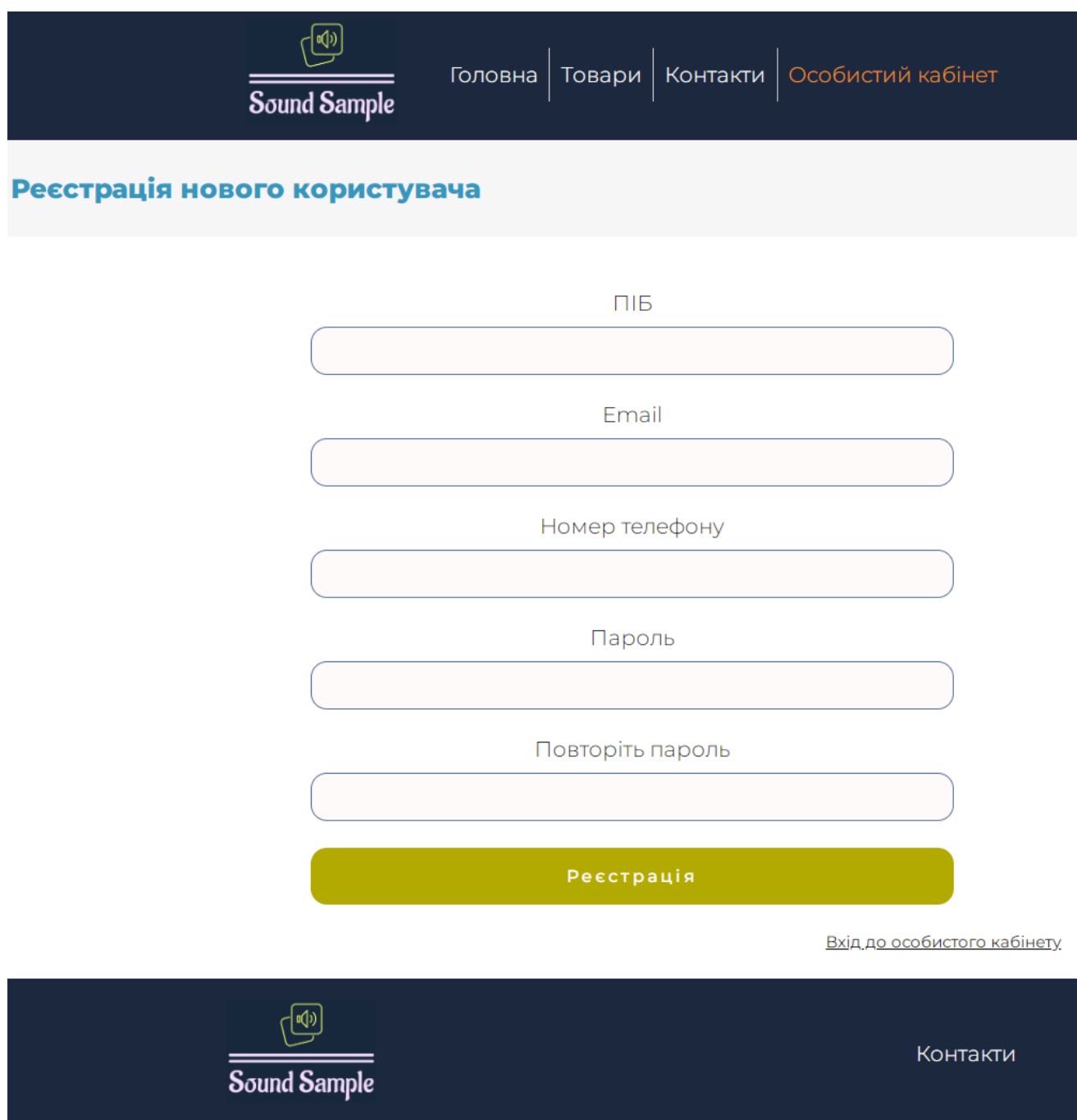
Контакти

Рисунок 3.10 – Форма входу

Форма реєстрації потребує заповнення інформації про:

- ПІБ
- Пошту
- Номер телефону
- Пароль

Довжина паролю повинна бути мінімум шість символів.



The image shows a registration form for the 'Sound Sample' website. The form is set against a dark blue header and footer. The header contains the 'Sound Sample' logo (a green speech bubble with a sound wave) and navigation links: 'Головна', 'Товари', 'Контакти', and 'Особистий кабінет'. The main content area has a light gray background with the title 'Реєстрація нового користувача' in blue. The form consists of several input fields: 'ПІБ', 'Email', 'Номер телефону', 'Пароль', and 'Повторіть пароль', each with a light pink rounded rectangle. Below these is a green 'Реєстрація' button. A link '[Вхід до особистого кабінету](#)' is located at the bottom right of the form area. The footer contains the 'Sound Sample' logo and a 'Контакти' link.

Головна | Товари | Контакти | **Особистий кабінет**

Реєстрація нового користувача

ПІБ

Email

Номер телефону

Пароль

Повторіть пароль

Реєстрація

[Вхід до особистого кабінету](#)

Контакти

Рисунок 3.11 – Форма реєстрації

На вкладці товари знаходиться список аудіо-фалів доступних для покупки. Також присутній пошук для зручності.

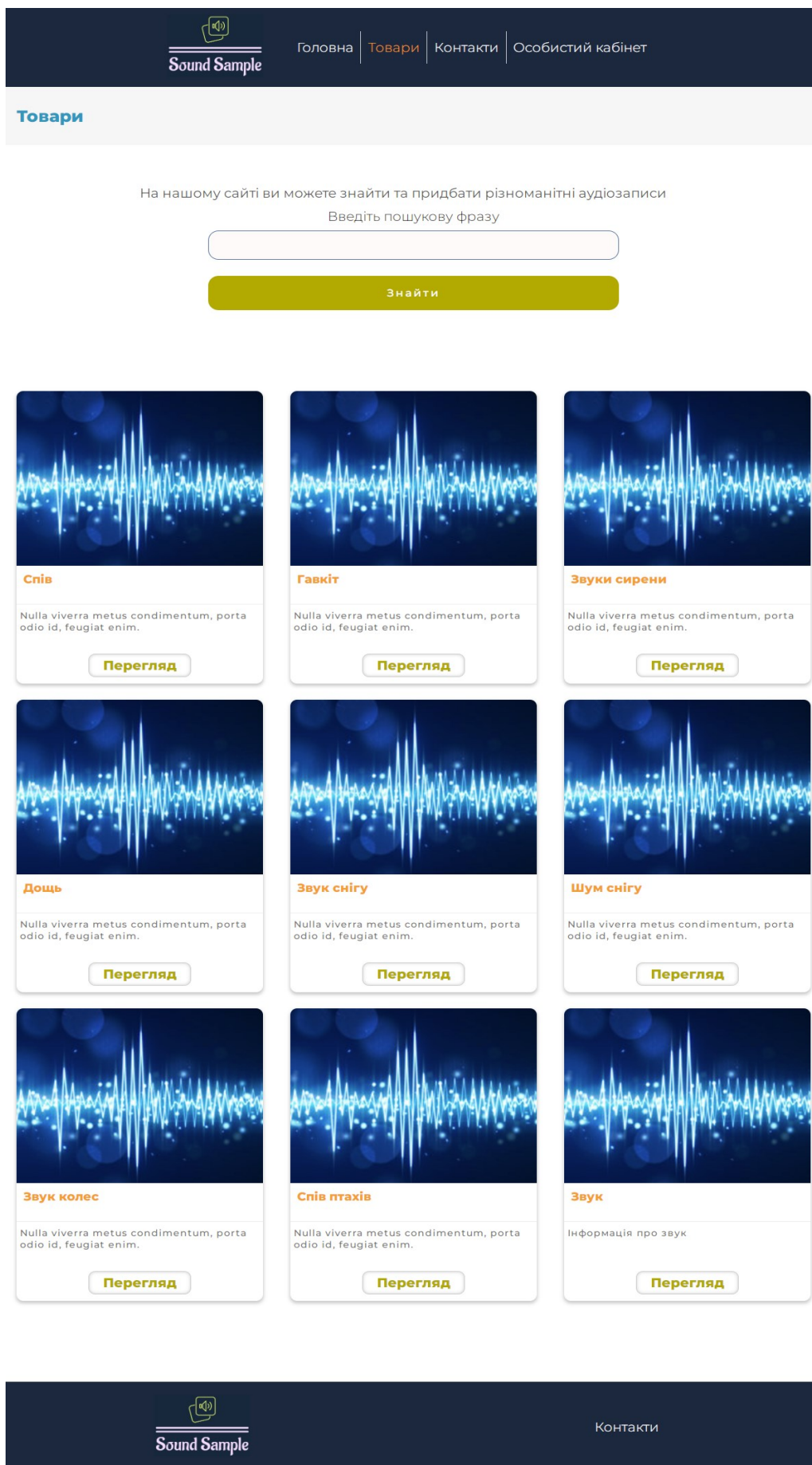


Рисунок 3.12 – Список товарів

Відкривши сторінку можна прочитати про нього детальну інформацію, придбати або залишити коментар.

Деталі подорожі

Спів

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut finibus tristique ipsum ut venenatis. Proin et tortor viverra tortor commodo eleifend. Sed iaculis facilisis mollis. Fusce dapibus, metus nec ornare pharetra, ligula nisl ultrices orci, id vulputate ante sem ac nisl. Nullam diam erat, hendrerit at pulvinar sed, consequat vel leo.

▶ 0:00 / 4:21 — 🔊 ⋮

Покупка

Коментарі

Автор: Юрій Петренко Дата: 2024-05-10 21:40:41

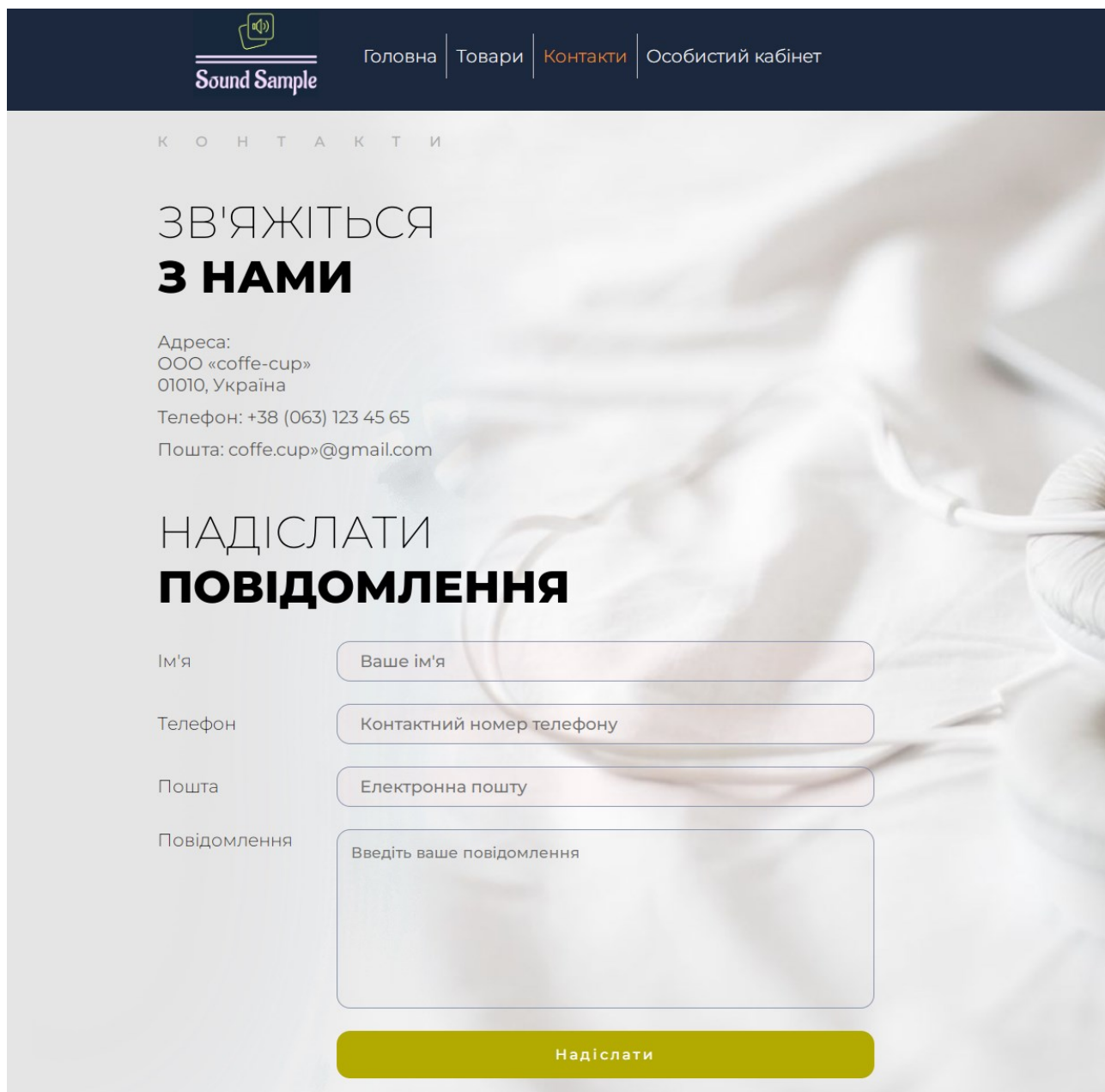
Коментар

Залишити відгук

Надіслати

Рисунок 3.13 – Сторінка товару

Також у головному меню присутній пункт «контакти» де знаходиться форма для звернення до адміністрації сайту і інформація про них.



Sound Sample

Головна | Товари | **Контакти** | Особистий кабінет

К О Н Т А К Т И

ЗВ'ЯЖІТЬСЯ З НАМИ

Адреса:
ООО «coffe-cup»
01010, Україна

Телефон: +38 (063) 123 45 65

Пошта: coffe.cup@gmail.com

НАДІСЛАТИ ПОВІДОМЛЕННЯ

Ім'я

Телефон

Пошта

Повідомлення

Надіслати

Рисунок 3.14 – Сторінка контакти

Після авторизації користувач може переглядати свої покупки якщо вони були здійснені.

Також присутній функціонал створення нового товару. Заповнивши інформацію можна завантажити новий аудіо-файл.

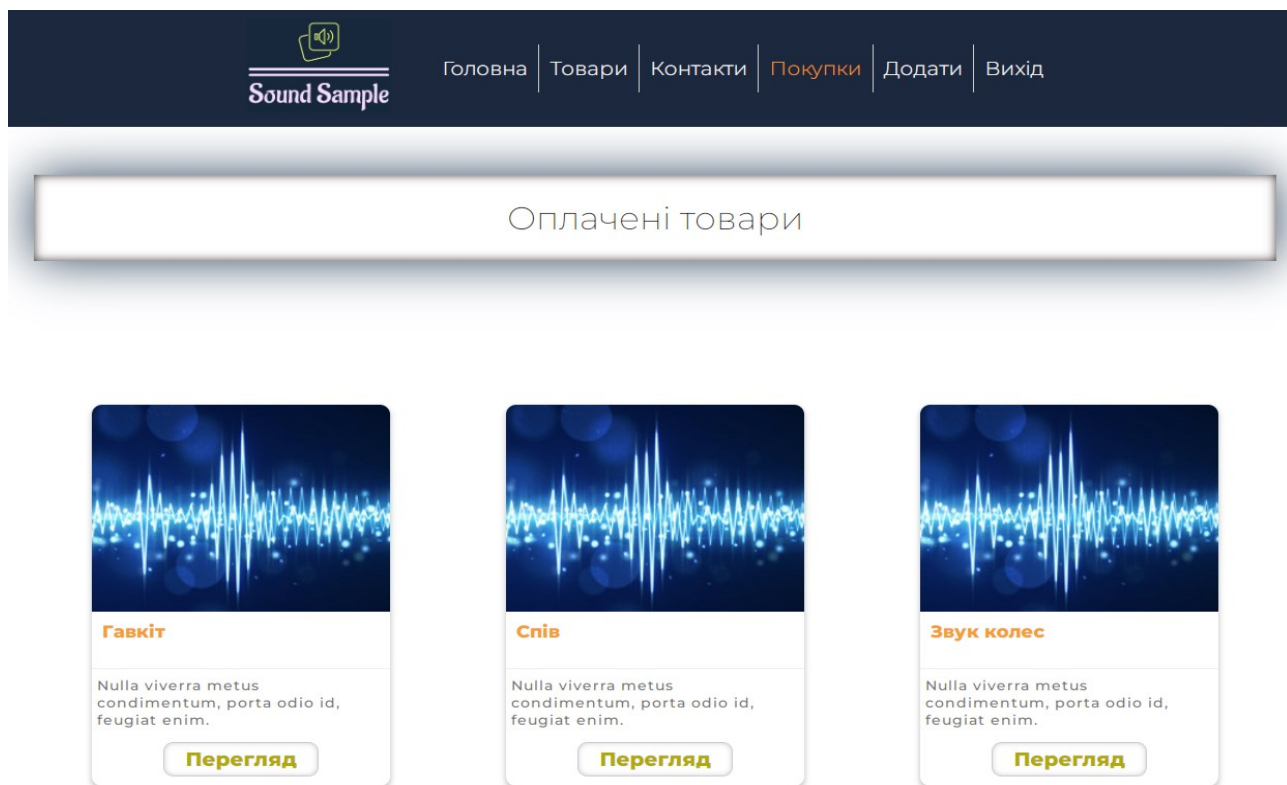


Рисунок 3.15 – Куплені товари

Sound Sample

Головна | Товари | Контакти | Покупки | Додати | Вихід

Додати аудіо

Назва запису

Детальна інформація

Короткий опис

Завантажте аудіо Файл не вибран

Рисунок 3.16 – Форма завантаження

Для адміністратора передбачена окрема сторінка для входу

The image shows a login form with a light gray background. At the top, the word "Логін" (Login) is centered. Below it is a white rectangular input field. Underneath that, the word "Пароль" (Password) is centered. Below it is another white rectangular input field. At the bottom of the form is a red rectangular button with the white text "Увійти" (Login).

Рисунок 3.17 – Форма логіну

Після авторизації в адміністративній панелі доступний функціонал:

- Перегляду користувачів
- Додавання аудіо-товарів
- Переглядати всі аудіо-товари
- Переглядати і видаляти коментарі
- Змінювати облікові дані
- Вихід з акаунту

Головна	Додати аудіо	Список	Коментарі	Пароль	Вихід
КОРИСТУВАЧІ					
Імя	Номер		Пошта		
ПІБ	0631234567		admin@gmail.com		
Юрій Петренко	0637485741		test@meta.ua		

Рисунок 3.18 – Адміністративна сторінка користувачів

ГоловнаДодати аудіоСписокКоментаріПарольВихід

ЗАВАНТАЖИТИ АУДІО

Назва запису

Детальна інформація

Короткий опис

Завантажте аудіо

Виберіть файл

Файл не вибран

Зберегти

Рисунок 3.19 – Адміністративна сторінка створення товару

ГоловнаДодати аудіоСписокКоментаріПарольВихід

СПИСОК ТОВАРІВ

Спів

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Гавкіт

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Звуки сирени

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Дощ

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Звук снігу

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Шум снігу

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Звук колес

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Спів птахів

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Звук

Інформація про звук

Перегляд

Рисунок 3.20 – Адміністративна сторінка перегляду товарів

ГоловнаДодати аудіоСписокКоментаріПарольВихід

КОМЕНТАРІ

Автор: Юрій ПетренкоДата: 2024-05-10 21:40:41

Коментар

Рисунок 3.21 – Адміністративна сторінка коментарів

Головна	Додати аудіо	Список	Коментарі	Пароль	Вихід
ЗМІНИТИ ДАНІ					
Логін					
Пароль					
Змінити					

Рисунок 3.22 – Адміністративна сторінка редагування пароля

3.4 Висновки до розділу 3

Під час аналізу функціоналу сервісу було виявлено, що процес завантаження аудіо зразків та користування платформою в цілому є інтуїтивно зрозумілим та зручним для кінцевого користувача. Описано всі етапи взаємодії з сервісом, від вибору та завантаження аудіо зразків до їхнього збереження та використання.

У частині розробки онлайн сервісу представлено детальний опис процесу розробки самого сервісу. Це включає розробку інтерфейсу, серверної частини та бази даних. Надано скріншоти та приклади коду для кращого розуміння та візуалізації проекту.

У розділі "Розробка модулю захисту" були представлені кроки написання коду для шифрування та вбудовування водяних знаків. Ці модулі були інтегровані в основний функціонал сервісу з метою забезпечення безпеки та захисту власності користувачів.

У підсумку, на основі аналізу та розробки функціоналу, а також інтеграції модулів захисту, можна вважати, що розроблений сервіс є ефективним та забезпечує високий рівень зручності та безпеки для користувачів.

ВИСНОВОК

Було успішно розроблено та реалізовано онлайн сервісу продажу аудіо та захисту їх стійкими цифровими водяними знаками, який забезпечує високий рівень безпеки даних користувачів.

Крім того, кожен аудіо зразок, що розміщується в магазині, забезпечений водяним знаком, що дозволяє відстежувати та ідентифікувати автора чи правовласника цих зразків. Це не лише підвищує безпеку продукту, а й додає додатковий шар захисту від піратства та незаконного використання.

У першому розділі було ретельно проаналізовано предметну область та сфери застосування аудіо, а також обґрунтована актуальність проблеми зберігання цих даних як наукової власності та її захисту. Проаналізовано існуючі методи стеганографічного вбудовування ЦВЗ. Проведено порівняльний аналіз існуючих аналогів інтернет-магазинів аудіо зразків, їх переваги та недоліки, що дало змогу виокремити недоліки та обґрунтувати необхідність створення власного сервісу з вдосконаленим захистом та накладанням ЦВЗ.

У другому розділі було представлено обґрунтування вибору мови програмування та описана архітектура сервісу. Розглянуто принцип роботи сервісу, включаючи перенаправлення даних на сервер, їх опрацювання та передачу до бази даних. Додатково була надана розробка модулю шифрування аудіо зразків та вбудовування водяних знаків в них, з обґрунтуванням їх переваг та визначенням етапів їх роботи у системі.

У третьому розділі було детально описано процес завантаження аудіо зразків та інші робочі процеси зі сторони користувача. Також було представлено процес розробки самого сервісу, включаючи інтерфейс, сервер та базу даних, з додатковими скріншотами та частинами коду. На завершення були представлені етапи написання коду, який забезпечує шифрування та вбудовування водяних знаків, а також їх опис та призначення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стрімінг. URL: <https://www.goldmansachs.com/intelligence/pages/infographics/music-streaming/> (дата звернення: 22.04.2024).
2. Ямні М., Дауї А., Кармуні Х. та ін. Ефективний алгоритм водяних знаків для цифрових аудіоданих у програмах безпеки // Sci Rep. 2023. Т. 13. С. 18432. DOI: 10.1038/s41598-023-45619-w. URL: <https://www.nature.com/articles/s41598-023-45619-w> (дата звернення: 22.04.2024).
3. WAVE. URL: <https://datafireball.com/2016/08/29/wav-deepdive-into-file-format/> (дата звернення: 23.04.2024).
4. WAVE. URL: <https://docs.fileformat.com/audio/wav/> (дата звернення: 25.04.2024).
5. Про авторське право і суміжні права : Закон України від 23.12.1993 № 3792-ХІІ // База даних «Законодавство України» / Верховна Рада України. URL: <https://zakon.rada.gov.ua/go/3792-12> (дата звернення: 28.04.2024).
6. Братасюк О. Б., Ментух Н. Ф. Поняття та класифікація цифрових прав в Україні // Юрид. наук. електрон. журн. 2021. № 10. С. 58-61.
7. WTO. URL: https://www.wto.org/english/thewto_e/countries_e/ukraine_e.htm (дата звернення: 30.04.2024).
8. Spotify хоче стати Instagram і TikTok для аудіоформату. URL: <https://forbes.ua/richest/spotify-khoche-stati-instagram-i-tiktok-dlya-audioformatu-02122021-2884> (дата звернення: 01.05.2024).
9. MUSO Discover Q1 2022: Digital Piracy Data Insights. URL: <https://www.muso.com/magazine/muso-discover-q1-2022-digital-piracy-data-insights> (дата звернення: 02.05.2024).
10. Коняхович Г. Ф., Прогонов Д. О. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних: підруч. Київ: РИДМИ, 2019. 268 с.

11. Oleshchenko, Vladimir, and Vladimir Pevnev. "Development of digital steganography techniques for copyright protection, based on the watermark." *Сучасні інформаційні системи* 1, № 1 (2017): 57-60.
12. Dutta, Hrishikesh, et al. "An overview of digital audio steganography." *IETE Technical Review* 37.6 (2020): 632-650.
13. Varaprasada Rao, K., and Sandeep Kumar Panda. "A design model of copyright protection system based on distributed ledger technology." *Computer Communication, Networking and IoT: Proceedings of 5th ICICC 2021, Volume 2*. Singapore: Springer Nature Singapore, 2022. 127-141.
14. Aslantaş F., Hanilçi C. Comparative Analysis of Audio Steganography Methods // *J. Innov. Sci. Eng.* 2022. T. 6, № 1. С. 122-137. DOI: 10.38088/jise.932549.
15. SOLOMON, Leron. Fair users or content abusers: The automatic flagging of non-infringing videos by content id on youtube. *Hofstra L. Rev.*, 2015, 44: 237.
16. PERRY, Burt; MACINTOSH, Brian; CUSHMAN, David. Digimarc MediaBridge: the birth of a consumer product from concept to commercial application. In: *Security and Watermarking of Multimedia Contents IV*. SPIE, 2002. p. 118-123.
17. Hesmondhalgh, David, Ellis Jones, and Andreas Rauh. "SoundCloud and Bandcamp as alternative music platforms." *Social Media+ Society* 5.4 (2019): 2056305119883429.
18. Mortensen, Bryson. "Choral Music in the Digital Age: Distributing your Music Online." *The Choral Journal* 50.10 (2010): 63.
19. WANG, Avery, et al. An industrial strength audio search algorithm. In: *Ismir*. 2003. p. 7-13.
20. MVC Tutorial. URL: <https://www.guru99.com/uk/mvc-tutorial.html> (дата звернення: 22.05.2024).
21. Use Case Diagrams Example. URL: <https://www.guru99.com/uk/use-case-diagrams-example.html> (дата звернення: 22.05.2024).
22. UML Diagrams. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 22.05.2024).

23. Laravel Passport. URL: <https://laravel.com/docs/11.x/passport> (дата звернення: 22.05.2024).
24. AL-ATRAQCHI, Osama MA. A proposed model for build a secure restful api to connect between server side and mobile application using laravel framework with flutter toolkits. Cihan University-Erbil Scientific Journal, 2022, 6.2: 28-35.
25. One-time password and other authentication algorithms: How are they used in digital security? URL: <https://introserv.com/ua/blog/one-time-password-and-other-authentication-algorithms-how-are-they-used-in-digital-security/> (дата звернення: 22.05.2024).
26. RedBeanPHP. URL: <https://www.redbeanphp.com/index.php> (дата звернення: 25.05.2024).
27. BALASUNDRAM, Indrani; RAMARAJ, E. Prevention of SQL Injection attacks by using service oriented authentication technique. International Journal of modeling and optimization, 2013, 3.3: 302.
28. How to use the PDO PHP extension to perform MySQL transactions in PHP on Ubuntu 18.04. URL: <https://www.digitalocean.com/community/tutorials/how-to-use-the-pdo-php-extension-to-perform-mysql-transactions-in-php-on-ubuntu-18-04> (дата звернення: 26.05.2024).
29. What is an SSL Certificate? URL: <https://www.cloudflare.com/learning/ssl/what-is-an-ssl-certificate/> (дата звернення: 27.05.2024).
30. Саїд М. Х., Вахбі Т. М. Інформаційна безпека для аудіостеганографії з використанням методу кодування фази // Eur. J. Theor. Appl. Sci. 2024. Т. 2, № 1. С. 634-647.
31. Івашкевич Д., Сімдянов І. PHP 7. Одеса: БХВ, 2014. 311 с.
32. Коняхович Г. Ф., Пузиренко А. Ю. Комп'ютерна стеганографія. Київ: МК-Прес, 2006. 288 с.
33. MELNYK, V.; NAHORNIUK, A.; BORTNIK, K. Дослідження швидкодії виконання запиту в СУБД MySQL та MariaDB засобами Python та PHP.

COMPUTER-INTEGRATED TECHNOLOGIES: EDUCATION, SCIENCE, PRODUCTION, 2019, 36: 123-127.

34. Що таке MySQL? URL: <https://freehost.com.ua/ukr/faq/wiki/hto-takoe-mysql/> (дата звернення: 28.05.2024).

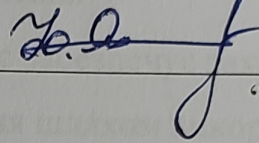
35. МЕЛЬНИЧУК, М. М. Перспективи застосування баз даних mysql в сучасних Інтернет ресурсах. Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку, 2015, 220-221.

ДОДАТКИ

Додаток А. Технічне завдання
Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції “Управління інформаційною
безпекою” кафедри МБІС
д.т.н., професор



Юрій ЯРЕМЧУК
“22” березня 2024 р.

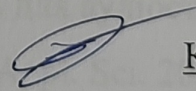
ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:
Розробка захищеного онлайн сервісу продажу аудіозаписів із захистом
авторського права стійкими цифровими водяними знаками

08-72.БДР.001.00.083 ТЗ

Керівник бакалаврської дипломної роботи

к.т.н., доцент



Карпінець В. В.

“ ”

Вінниця – 2024 р.

1. Найменування та область застосування

Захищений веб-сервіс продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками. Область застосування: захист аудіозаписів від несанкціонованого копіювання та розповсюдження у розробленому веб-сервісі продажу.

2. Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного методу захисту аудіозаписів від несанкціонованого копіювання та розповсюдження.

3.2 Призначення: розроблений веб-сервіс забезпечує захист аудіозаписів від несанкціонованого доступу та розповсюдження шляхом використання стійких цифрових водяних знаків.

4. Джерела розробки

4.1. Ямні М., Дауї А., Кармуні Х. та ін. Ефективний алгоритм водяних знаків для цифрових аудіоданих у програмах безпеки // Sci Rep. 2023. Т. 13. С. 18432. DOI: 10.1038/s41598-023-45619-w.

4.2. Zakaria Omar, Zangooei Toomaj, Mohd Afizi Mohd Shukran. Enhancing Mixing Recognition-Based and Recall-Based Approach in Graphical Password Scheme. ІАСТ, Vol. 4, No. 15, pp. 189-197, 2012.

4.3. Саїд М. Х., Вахбі Т. М. Інформаційна безпека для аудіостеганографії з використанням методу кодування фази // Eur. J. Theor. Appl. Sci. 2024. Т. 2, № 1. С. 634-647.

4.4. Dutta, Hrishikesh, et al. "An overview of digital audio steganography." IETE Technical Review 37.6 (2020): 632-650.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

- процесор – Pentium 1500 МГц і подібні до них;
- оперативна пам'ять – не менше 512 Мб;
- середовище функціонування – операційна система сімейство Windows;
- вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп'ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов'язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

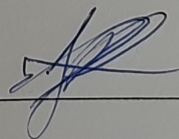
№ з/п	Назва етапів бакалаврської дипломної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми	15.03.2024	22.03.2024
2.	Аналіз предметної області обраної теми	27.03.2024	10.04.2024
3.	Розробка алгоритму роботи	08.05.2024	15.05.2024
4.	Написання бакалаврської роботи на основі розробленої теми	15.05.2024	22.05.2024
5.	Передзахист бакалаврської дипломної роботи	23.05.2024	24.05.2024
6.	Виправлення, уточнення, корегування бакалаврської дипломної роботи	25.05.2024	04.06.2024
7.	Захист бакалаврської дипломної роботи		

10. Порядок контролю та прийому

10.1 До приймання бакалаврської дипломної роботи надається:

- ПЗ до бакалаврської дипломної роботи;
- демонстрація результату бакалаврської дипломної роботи;
- презентація;
- відзив керівника роботи;
- відзив рецензента

Технічне завдання до виконання прийняв _____ Адаменко І. О.



Додаток Б. Лістинг функції накладання ЦВЗ

```

<?php
require 'vendor/autoload.php';
use GetId3\GetId3;
use GetId3\Write\Writer;
use RedBeanPHP\R;
R::setup('mysql:host=127.0.0.1;dbname=sound', 'admin', 'sdghhsfh4t3dfhfgj43');
R::ext('xdispense', function($type) {
    return R::getRedBean()->dispense($type);
});
function readAudioSamples($path) {
    $getId3 = new GetId3();
    $fileInfo = $getId3->analyze($path);
    $samples = [];
    $waveFile = fopen($path, 'rb');
    fseek($waveFile, $fileInfo['avdataoffset']);
    while (!feof($waveFile)) {
        $sample = unpack('f', fread($waveFile, 4))[1];
        $samples[] = $sample;
    }
    fclose($waveFile);
    return $samples;
}
function writeAudioSamples($path, $samples) {
    $header = pack('N', 0x52494646) // 'RIFF'
        . pack('V', 36 + count($samples) * 4) // file size
        . pack('N', 0x57415645) // 'WAVE'
        . pack('N', 0x666D7420) // 'fmt '
        . pack('V', 16) // PCM chunk size
        . pack('v', 1) // PCM format
        . pack('v', 1) // mono
        . pack('V', 44100) // sample rate
        . pack('V', 44100 * 4) // byte rate
        . pack('v', 4) // block align
        . pack('v', 32) // bits per sample
        . pack('N', 0x64617461) // 'data'
        . pack('V', count($samples) * 4); // data size
    $waveFile = fopen($path, 'wb');
    fwrite($waveFile, $header);
    foreach ($samples as $sample) {
        fwrite($waveFile, pack('f', $sample));
    }
    fclose($waveFile);
}
function stringToBinary($data) {
    $binary = '';
    for ($i = 0; $i < strlen($data); $i++) {
        $binary .= str_pad(decbin(ord($data[$i])), 16, '0', STR_PAD_LEFT);
    }
    return $binary;
}
function encodePhase(&$samples, $data) {
    $binaryData = stringToBinary($data);
    $metadata = str_pad(decbin(strlen($binaryData)), 32, '0', STR_PAD_LEFT);
    $binaryData = $metadata . $binaryData;
    $bitLength = strlen($binaryData);
}

```

```

$sampleCount = min($bitLength, count($samples));
$deltaPhase = pi() / 8;
for ($i = 0; $i < $sampleCount; $i++) {
    $sample = $samples[$i];
    $phase = acos($sample);
    if ($binaryData[$i] == '1') {
        $phase += $deltaPhase;
    } else {
        $phase -= $deltaPhase;
    }
    $samples[$i] = cos($phase);
}
}

function saveToDatabase($filePath, $userId, $tourId) {
    $watermark = R::xdispense('watermarks');
    $watermark->user_id = $userId;
    $watermark->name = basename($filePath);
    $watermark->tour_id = $tourId;
    $watermark->created_at = R::isoDateTime();
    R::store($watermark);
}

$audioPath = 'input.wav';
$samples = readAudioSamples($audioPath);

echo "Enter the string: ";
$data = trim(fgets(STDIN));

if (count($samples) < strlen($data) * 32) {
    echo "Not enough audio samples to encode the data.\n";
    exit;
}

encodePhase($samples, $data);
$encodedFilePath = 'encoded.wav';
writeAudioSamples($encodedFilePath, $samples);
$userId = 1;
$tourId = 1;
saveToDatabase($encodedFilePath, $userId, $tourId);
echo "Encoding completed and written to 'encoded.wav'. Data saved to database.\n";
?>

```

Додаток В. Лістинг файлу index.php

```

<section class="header">
  <div class="slider-container">
    <div class="slider">
      <div class="slider__item">
        <h1 class="slider__caption">
          <span class="top caption-item">Lorem ipsum dolor</span>
          <span class="big caption-item">faucibu</span>
          <span class="spacing caption-item">Cras faucibu </span>
        </h1>
      </div>
      <div class="slider__item">
        <h2 class="slider__caption">
          <span class="top caption-item">Lorem ipsum dolor </span>
          <span class="big caption-item">sitamet</span>
          <span class="spacing caption-item">amet iaculis</span>
        </h2>
      </div>
      <div class="slider__item">
        <h2 class="slider__caption">
          <span class="top caption-item">consectetur adipiscing elit</span>
          <span class="big caption-item">Proin. vitae.</span>
          <span class="spacing caption-item">tellus sagittis </span>
        </h2>
      </div>
    </div>
    <div class="slider__switch slider__switch--prev" data-ikslider-dir="prev"><span> </span></div>
    <div class="slider__switch slider__switch--next" data-ikslider-dir="next"><span> </span></div>
  </div>
</section>
<section class="year wrap">
  <div class="left">
    <div class="border">
      <p class="big">9</p>
      <p class="med">ПОКІВ</span>
      <p class="bot">ДОВІРИ</p>
    </div>
  </div>
  <div class="right">
    <p class="spacing">Sound-Sample</p>
    <p class="big">ЗАХИЩЕНІ</p>
    <p class="med">ТРЕКИ</p>
  </div>
</section>
<section class="why wrap">
  <div class="left">
    
  </div>
  <article class="right">
    <div class="article">
      <h3 class="spacing">ЧОМУ МИ?</h3>
      <p class="med">РАЗОМ</span>
      <p class="med">З НАМИ</p>
      <p class="big">ЗРУЧНО</p>
    </div>
  </div>
  <p class="text">
    Faucibus ex quis, convallis ipsum. Duis finibus egestas urna, sed viverra nisl sagittis vel.
  </p>

```

```

    Vestibulum sit amet dapibus lorem. Integer et metus ac eros interdum tempus non vitae neque.
  </p>
  <p class="text">
    Fusce scelerisque dapibus ipsum, ac placerat magna tincidunt vitae. Fusce volutpat varius risus,
    id vestibulum mauris condimentum vitae. Fusce turpis orci, vulputate id sem ut, finibus dapibus
    ex. Praesent tincidunt neque mattis, ullamcorper urna vitae, luctus dolor. Nunc massa eros, .
  </p>
</div>
</article>
</section>
<section class="labels wrap">
  <div class="item">
    <div class="img">
      
    </div>
    <h4 class="title">Перевірені автори</h4>
    <p>Tincidunt magna. Proin lacinia erat orci, at placerat ut.</p>
  </div>
  <div class="item">
    <div class="img">
      
    </div>
    <h4 class="title">3 нами зручно</h4>
    <p>Proin lacinia erat orci, at placerat tellus sollicitudin ut.</p>
  </div>
  <div class="item">
    <div class="img">
      
    </div>
    <h4 class="title">Обрані записи</h4>
    <p>Tincidunt malesuada magna. Proin tellus sollicitudin ut.</p>
  </div>
  <div class="item">
    <div class="img">
      
    </div>
    <h4 class="title">Економія часу</h4>
    <p>Magna proin lacinia erat orci, at placerat tellus sollicitudin ut.</p>
  </div>
</section>
<section class="care">
  <article class="article">
    <h3 class="spacing">ваші дані у безпеці</h3>
    <p class="text">
      Fusce scelerisque dapibus ipsum, ac placerat magna tincidunt vitae. Fusce volutpat varius risus,
      id vestibulum mauris condimentum vitae. Fusce turpis orci, vulputate id sem ut, finibus dapibus
    </p>
    <p class="text">
      Praesent tincidunt neque mattis, ullamcorper urna vitae, luctus dolor. Nunc massa eros,
      ultricies eget nunc ac, tincidunt malesuada magna.
      Proin lacinia erat orci, at placerat tellus sollicitudin ut.
    </p>
  </article>
</section>
<section class="num">
  <div class="inner wrap">
    <h4 class="spacing">У ЦИФРАХ</h4>
    <div class="item top">

```

```

</div>
</section>
<section class="contact">
  <div class="small">
    <div class="top">
      <h4 class="spacing">КОНТАКТИ</h4>
      <p class="med">ЗВ'ЯЖІТЬСЯ</p>
      <p class="big">З НАМИ</p>
      <p class="text">
        Адреса:<br>
        ООО «Sound-Sample»<br>
        Вінниця
      </p>
      <a class="text" href="tel:0959121092">Телефон: +38 (000) 111 22 33</a>
      <a class="text" href="mailto:coffe.cup@gmail.com">Пошта: sound-sample@gmail.com</a>
    </div>
    <div class="form">
      <p class="med">надіслати</p>
      <p class="big">повідомлення</p>
      <form class="msg" method="POST" action="/contact/send">
        <div class="item">
          <span>Ім'я</span>
          <input class="name" type="text" name="name" required placeholder="Ваше ім'я">
        </div>
        <div class="item">
          <span>Телефон</span>
          <input class="phone" type="text" name="phone" required placeholder="Контактний номер телефону">
        </div>
        <div class="item">
          <span>Пошта</span>
          <input class="email" type="email" name="email" placeholder="Електронна пошта">
        </div>
        <div class="item">
          <span class="top">Повідомлення</span>
          <textarea class="msg" name="text" required placeholder="Введіть ваше повідомлення"></textarea>
        </div>
        <div class="btn">
          <input class="send" type="button" value="Надіслати">
        </div>
      </form>
    </div>
  </div>
</section>
<footer class="footer">
  <div class="inner wrap">
    <div class="logo">
      
    </div>
    <div class="links">
      <a href="/contact">Контакти</a>
    </div>
  </div>
</footer>

```

Додаток Г. Лістинг файлів controllers

```

<?php
namespace controllers;
class UserController extends Controller {
    public function indexAction(){
        if( !isset($_SESSION['user_id']) ){
            redirect('/user/login');
        }
        $selected = \R::findAll( 'selected' );
        $tours = \R::findAll( 'tours' );
        $this->set(compact('tours', 'selected'));
    }
    public function loginAction(){
        $_SESSION['page'] = 'user';
        if( isset($_SESSION['user_id']) && $_SESSION['login'] == 'user'){
            redirect('/user');
        }
    }
    public function registerAction(){
        $_SESSION['page'] = 'user';
    }
    public function checkAction(){
        $email = clear($_POST['email'], 'text');
        $pass = $_POST['pass'];
        if( empty($email) || empty($pass) ){
            $_SESSION['form_err'][] = 'Заповніть всі поля.';
        }
        $user = \R::findOne( 'users', 'email = ?', [ $email ] );
        if( !$user || md5($pass) != $user->pass){
            $_SESSION['form_err'][] = 'Невірний логін/пароль.';
            redirect();
        }else{
            $_SESSION['user_id'] = $user->id;
            $_SESSION['login'] = 'user';
            redirect('/user');
        }
    }
    public function createAction(){
        $name = clear($_POST['name'], 'text');
        $email = clear($_POST['email'], 'text');
        $phone = clear($_POST['phone'], 'text');
        $pass = $_POST['pass'];
        $pas = $_POST['pas'];
        $user = \R::findOne( 'users', 'email = ?', [ $email ] );
        if( isset($user) ){
            $_SESSION['form_err'][] = 'Вказаний email вже використовується.';
        }
        if( $pass != $pas ){
            $_SESSION['form_err'][] = 'Паролі не співпадають.';
        }
        if( mb_strlen($pass) < 6 ){
            $_SESSION['form_err'][] = 'Пароль повинен містити не менше 6 символів.';
        }
        if( empty($pas) || empty($pass) || empty($phone) || empty($email) ){
            $_SESSION['form_err'][] = 'Заповніть всі поля';
        }
        if( isset($_SESSION['form_err']) ){

```

```

    redirect();
    return;
}
$item = \R::dispense( 'users' );
$item->name = $name;
$item->email = $email;
$item->phone = $phone;
$item->pass = md5($pass);
\R::store( $item );
$_SESSION['user_id'] = $item->id;
$_SESSION['login'] = 'user';
redirect('/user');
}
public function exitAction(){
    unset($_SESSION['user_id']);
    unset($_SESSION['login']);
    redirect();
}
public function addTourAction(){
    $_SESSION['page'] = 'addTour';
}
public function saveUserAction(){
    $name = clear($_POST['name'], 'text');
    $about = clear($_POST['about'], 'text');
    $q = clear($_POST['q'], 'text')
    $q = clear($_POST['q'], 'text');
    $qq = $this->loadImg('q-img');
    $tours = \R::dispense( 'tours' );
    $tours->name = $name;
    $tours->about = $about;
    $tours->q = $q;
    $tours->img = $qq;
    \R::store( $tours );
    redirect('/user');
}
public function loadImg($name){
    $uploaddir = 'load/';
    $file = basename($_FILES[$name]['name']);
    $existFile = $uploaddir . time() . $file;
    if( $_FILES[$name]['error'] == 0 ){
        if( is_uploaded_file($_FILES[$name]['tmp_name'])) {
            if( !file_exists($existFile) ){
                if( (move_uploaded_file($_FILES[$name]['tmp_name'], $existFile)) ) {
                }
            }
        }
    }
    $audio_file = 'input_audio.wav';
    $text = 'Цифровой водяной знак';
    $file = time() . '_audio.wav';
    $this->embedTextWatermark($existFile, $text, $file);
    return $existFile;
}
}
public function embedTextWatermark($audio_file, $watermark_text, $output_file) {
    $audio_data = file_get_contents($audio_file);
    $watermark_bytes = unpack('C*', $watermark_text);
    $watermarked_bytes = array();
    $watermark_index = 0;

```

```

foreach ($audio_data as $byte) {
    $watermarked_byte = $byte;
    if ($watermark_index < count($watermark_bytes)) {
        $watermarked_byte = $watermark_bytes[$watermark_index];
        $watermark_index++;
    }
    $watermarked_bytes[] = $watermarked_byte;
}
$watermarked_audio_data = pack('C*', ...$watermarked_bytes);
file_put_contents($output_file, $watermarked_audio_data);
}
}
<?php
namespace controllers;
class PriceController extends Controller {
    public function indexAction(){
        $_SESSION['page'] = 'price';
        $tours = \R::findAll( 'tours' );
        $this->set(compact('tours'));
    }
    public function searchAction(){
        $s = clear($_POST['search'], 'text');
        $s = '%' . $s . '%';
        $tours = \R::findAll( 'tours', 'WHERE name like ? OR q like ? ', [$s,$s] );
        $this->set(compact('tours'));
    }
    public function viewAction(){
        $_SESSION['page'] = 'price';
        $id = clear($_GET['id'], 'int');
        $comments = \R::findAll( 'comments', 'tour_id = ? ORDER BY date DESC', [ $id ] );
        $tour = \R::findOne( 'tours', 'id = ?', [ $id ] );
        $this->set(compact('tour', 'labels', 'comments'));
    }
    public function addCommentAction(){
        if( empty($_SESSION['user_id']) ){
            redirect('/user/login');
        }
        $id = clear($_POST['id'], 'int');
        $text = clear($_POST['text'], 'text');
        $user = \R::findOne( 'users', 'id = ?', [ $_SESSION['user_id'] ] );
        $comment = \R::dispense( 'comments' );
        $comment->text = $text;
        $comment->name = $user->name;
        $comment->tour_id = $id;
        $comment->user_id = $user->id;
        \R::store( $comment );
        redirect();
    }
    public function toFavAction(){
        $id = clear($_GET['id'], 'int');
        if( empty($_SESSION['user_id']) ){
            redirect('/user');
            die;
        }
        $present = \R::findOne( 'selected', 'tour_id = ? AND user_id = ?', [ $id, $_SESSION['user_id'] ] );
        if( empty($present) ){
            $user = \R::findOne( 'users', 'id = ?', [ $_SESSION['user_id'] ] );
            $selected = \R::dispense( 'selected' );
            $selected->tour_id = $id;

```

```

        $selected->user_id = $user->id;
        \R::store( $selected );
        redirect('/user');
    }
    redirect();
}
public function fromFavAction(){
    $id = clear($_GET['id'], 'int');
    $present = \R::findOne( 'selected', 'id = ? ', [ $id ] );
    \R::trash( $present );
    redirect();
}
}
<?php
namespace controllers;
use View;
use models\Model;
use Exception;
abstract class Controller {
    public $route = [];
    public $view;
    public $layout;
    public $vars = [];
    public function __construct($route) {
        Model::instance();
        if( $route['prefix'] != '' && strtolower($route['controller']) != 'welcome' ){
            if( empty($_SESSION['user_id']) || $_SESSION['login'] != 'panel'){
                redirect('/panel/welcome');
            }else{
                $this->route = $route;
                $this->view = $route['action'];
            }
        }else{
            $this->route = $route;
            $this->view = $route['action'];
        }
    }
    public function getView(){
        $vObj = new View($this->route, $this->layout, $this->view);
        $vObj->render($this->vars);
    }
    public function set($vars){
        $this->vars = $vars;
    }
    public function isAjax() {
        return isset($_SERVER['HTTP_X_REQUESTED_WITH']) && $_SERVER['HTTP_X_REQUESTED_WITH'] ===
'XMLHttpRequest';
    }
    public function loadView($view, $vars = []){
        extract($vars);
        require ROOT . "/views/{$this->route['prefix']}/{ $this->route['controller'] }/{ $view }.php";
    }
}
<?php
namespace controllers;
class WelcomeController extends Controller {
    public function indexAction(){
    }
}

```

```

public function authorizeAction(){
    if( empty($_POST['login']) || empty($_POST['pass']) ){
        redirect('/');
    }
    $login = clear($_POST['login'], 'text');
    $user = \R::findOne( 'users', 'login = ?', [$login] );
    if( password_verify($_POST['pass'], $user['pass']) ){
        $_SESSION['id'] = $user['id'];
        if( $user['role'] ){
            unset( $_SESSION['user'] );
            $_SESSION['admin'] = 'ok';
            redirect('/panel');
        }else{
            unset( $_SESSION['admin'] );
            $_SESSION['user'] = 'ok';
            redirect('/');
        }
    }
    redirect('/welcome');
}

public function logoutAction(){
    unset( $_SESSION['admin'] );
    unset( $_SESSION['user'] );
    unset( $_SESSION['id'] );
    redirect('/');
}
}
<?php
namespace controllers\panel;
use controllers\Controller;
class AdminController extends Controller {
    public $layout = 'admin';
    public function indexAction(){
        $condition = ' ORDER BY id DESC ';
        $item_for_page = 50;
        !empty( $_GET['page'] ) ? $page = clear($_GET['page'], 'int') : $page = 0;
        $numOfOrder = \R::count( 'users' );
        if( $numOfOrder > $item_for_page ){
            $page -= 1;
            if( $page < 0 ){
                $page = 0;
            }
            $start_get_from = $page * $item_for_page;
            $condition .= ' limit ' . $start_get_from . ', ' . $item_for_page;
            $page_quantity = $numOfOrder / $item_for_page;
            $page_quantity = ceil($page_quantity);
            if( $page_quantity >= ($page + 1) ){
                $users = \R::find( 'users', $condition );
            }else{
                $page_quantity = 0;
            }else{
                $users = \R::find( 'users', $condition );
            }
            $this->set(compact('users', 'page_quantity', 'page'));
        }
    }
    public function userAction(){
        $user = \R::findOne( 'panel', ' id = 1 ' );
        $this->set(compact('user')); }
}

```

```

public function saveUserAction(){
    $name = clear($_POST['name'], 'text');
    $about = clear($_POST['about'], 'text');
    $q = clear($_POST['q'], 'text');
    $q = clear($_POST['q'], 'text');
    $qq = $this->loadImg('q-img');
    $tours = \R::dispense( 'tours' );
    $tours->name = $name;
    $tours->about = $about;
    $tours->q = $q;
    $tours->img = $qq;
    \R::store( $tours );
    redirect(); }

public function toursAction(){
    $tours = \R::findAll( 'tours' );
    $this->set(compact('tours')); }

public function placesAction(){
    $places = \R::findAll( 'places' );
    $this->set(compact('places')); }

public function tourDelAction(){
    $id = clear($_GET['id'], 'int');
    $present = \R::findOne( 'tours', 'id = ? ', [ $id ] );
    \R::trash( $present );
    $selected = \R::findAll( 'selected', 'tour_id = ? ', [ $id ] );
    \R::trashAll( $selected );
    redirect();

public function commentsAction(){
    $comments = \R::findAll( 'comments' );
    $this->set(compact( 'comments' )); }

public function comDelAction(){
    $id = clear($_GET['id'], 'int');
    $comment = \R::findOne( 'comments', 'id = ? ', [ $id ] );
    \R::trash( $comment )
    redirect(); }

public function addAction(){ }

public function activeTourAction(){
    $id = clear($_GET['id'], 'int');
    $active = clear($_GET['active'], 'int');
    $tours = \R::findOne( 'tours', 'id = ? ', [ $id ] );
    $tours->active = $active;
    \R::store( $tours
    redirect(); }

public function loadImg($name){
    $uploaddir = 'load/';
    $file = basename($_FILES[$name]['name']);
    $existFile = $uploaddir . time() . $file;

    if( $_FILES[$name]['error'] == 0 ){
        if( is_uploaded_file($_FILES[$name]['tmp_name'])) {
            if( !file_exists($existFile) ){
                if( (move_uploaded_file($_FILES[$name]['tmp_name'], $existFile)) ) {
                }
            }
        }
    }
    return $existFile; }
}
}

```

Розробка захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками

Виконав студент групи 1КІТС-20Б Адаменко І. О.

Науковий керівник: к.т.н., доц., зав. каф. МБІС Карпінєць В. В.

Актуальність теми

Актуальність роботи полягає в необхідності розв'язання проблеми захисту конфіденційності та безпеки музичних файлів у цифровому середовищі. Зростання кількості цифрових атак і порушень безпеки створює загрозу для авторських прав та інтелектуальної власності у музичній індустрії. Розробка і впровадження інноваційних заходів безпеки, таких як надійне шифрування та стеганографічні методи, стає надзвичайно актуальною для забезпечення безпеки даних користувачів та захисту їхніх авторських прав в цифровому середовищі.

Мета роботи

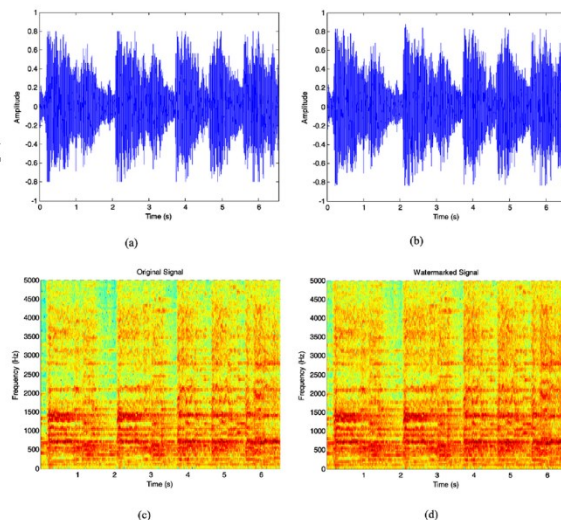
Метою роботи є створення інтернет-магазину аудіо зразків з використанням надійного шифрування для забезпечення безпеки даних користувачів. Проект спрямований на розробку нового веб-сервісу, який об'єднує переваги існуючих рішень та технології накладання ЦВЗ для забезпечення високого рівня безпеки та захисту авторських прав.

Новизна

Новизна цієї роботи полягає в розробці інтернет-магазину аудіо зразків, який поєднує в собі технології забезпечення безпеки даних та захисту авторських прав. Інноваційний підхід полягає у використанні сучасних методів шифрування даних користувачів, безпечної авторизації через протокол OAuth 2.0 та впровадженні вдосконалених методів стеганографії для захисту аудіофайлів.

Стійкий цифровий водяний знак

Стійкий цифровий водяний знак - це невидимий індивідуальний код, який вбудовується в цифровий контент (в даному випадку аудіофайли) для захисту від несанкціонованого використання та підробки. Він надійно вбудовується у контент і залишається стійким навіть при зміні формату або компресії файлу.

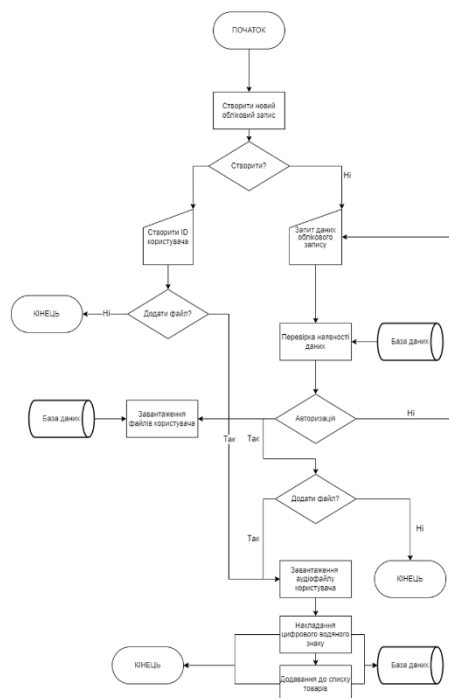
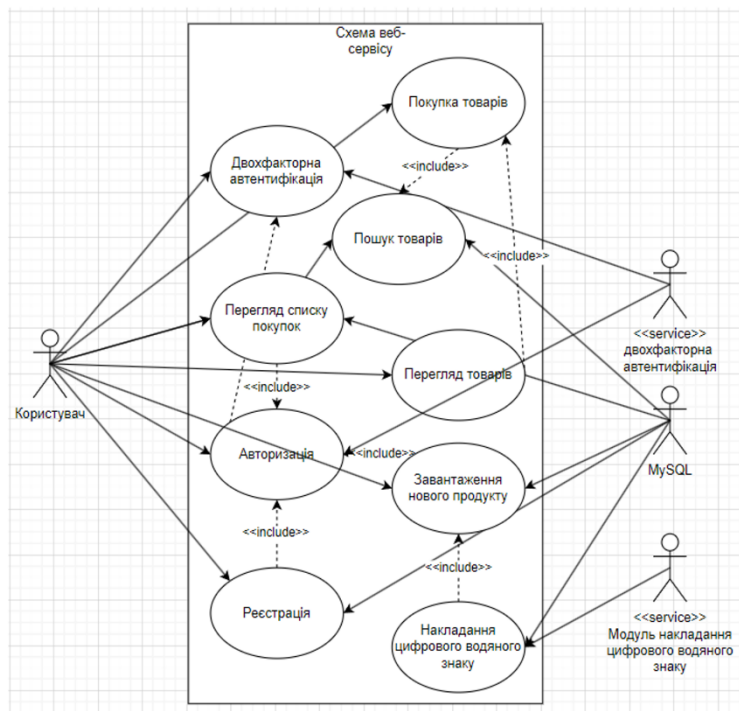


Порівняння стеганографічних методів

Метод	Пропускна здатність	Непомітність	Стійкість до атак	<u>Психоакустичний аналіз</u>
Кодування найменш значущих бітів	$C = \frac{n}{m}$ Мала	Високий коефіцієнт сигнал/шум	Низька, проста маніпуляція може зруйнувати водяний знак	Висока непомітність через зміну найменш значущих бітів
Метод розширення спектра	$C = \frac{W}{B}$ Висока	Висока непомітність, адитивний шум	<u>Середня, можливість виявлення досвідченими атаками</u>	Висока непомітність через адитивний шум
<u>Приховування даних із використанням ехо-сигналу</u>	$C = \frac{I}{T}$ Мала	Висока непомітність, природний резонанс	Середня, можливі атаки на видалення ехо	Висока непомітність через <u>природний ехо-ефект</u>
Метод фазового кодування	$C = \frac{n}{m}$ Мала	Висока непомітність, зміни фази	Висока, складний для виявлення і модифікації	Висока непомітність через низьку чутливість до змін фази

Примітки:

- n кількість бітів водяного знака.
- m кількість вибірок у аудіосигналі.
- W кількість вбудованих бітів.
- B смуга пропускання сигналу.
- I обсяг вбудованої інформації.



Модулі захисту веб-сервісу

Авторизація та аутентифікація
з використанням Laravel
Passport:



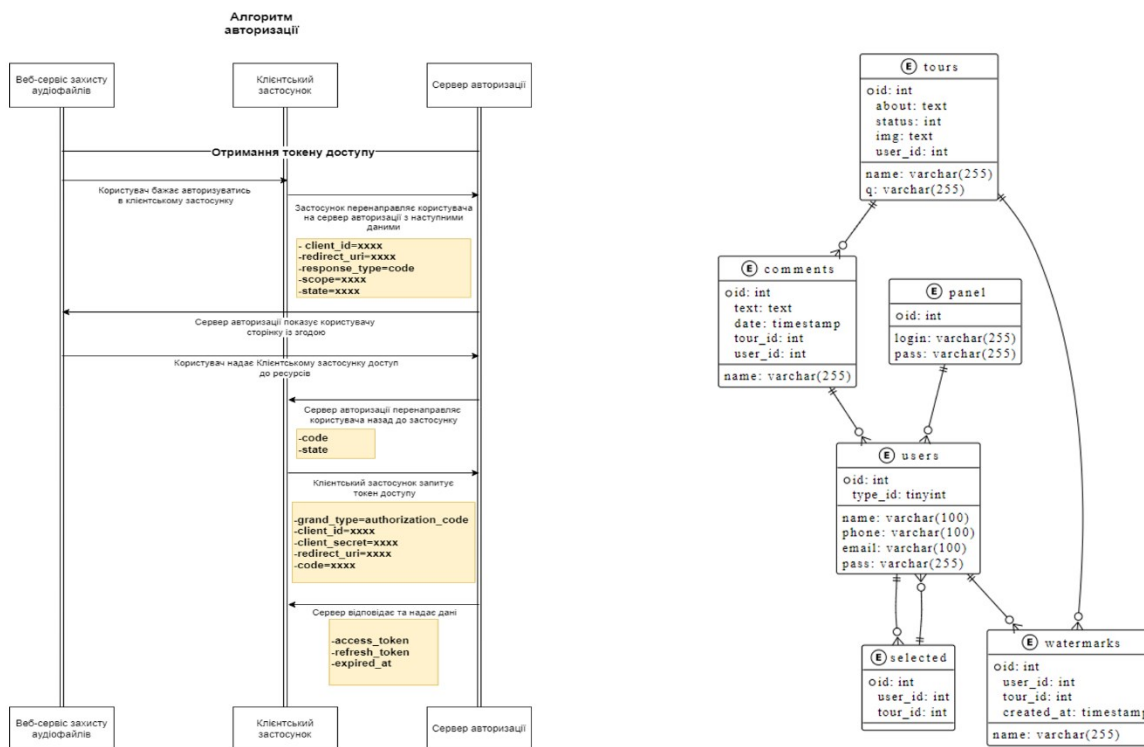
Захист від XSS-атак та
Sql-in'єкцій:



Розробка захищеної
бази даних

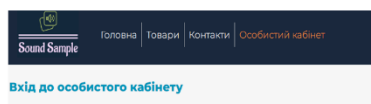
Модуль двофакторної
автентифікації з
використанням OTP:





Приклад авторизації веб-сервісу

Захищена авторизація:



Authorization Request

My Client is requesting permission to access your account.

Authorize

Cancel

Email

Пароль

[Встановити новий користувач](#)



Ilya Adamenko KITC 206 <ilysha.adamenko@gmail.com>

кому: МНЕ ▼

Ваш одноразовий пароль - 15487рв23



Приклад завантаження аудіо

Форма завантаження аудіо:

Додавання аудіо у список товарів:

Додати аудіо

Назва запису

Детальна інформація

Короткий опис

Завантажте аудіо

Cnle

Выберите файл cnle.wav

Зберегти

Sound Sample

ГоловнаТовариКонтактиОсобистий кабінет

Товари

На нашому сайті ви можете знайти та придбати різноманітні аудіозаписи

Введіть пошукову фразу

Знайти

Cnle

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Canis

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Зарок сирени

Nulla viverra metus condimentum, porta odio id, feugiat enim.

Перегляд

Порівняння з аналогами

Сервіс	Особливості	Види захисту
Даний веб-сервіс	Захист авторських прав, використання водяних знаків, стрімінг музики	Водяні знаки, DRM
Content ID (YouTube)	Автоматичне виявлення порушень авторських прав на YouTube	Цифрові водяні знаки, автоматизоване виявлення контенту
Digimarc	Вбудовання невидимих метаданих для ідентифікації контенту	Стійкі цифрові водяні знаки, аналіз метаданих
SoundCloud	Розповсюдження аудіоконтенту з використанням водяних знаків	Цифрові водяні знаки, автоматичне виявлення порушень
TuneCore	Розповсюдження музики для незалежних музикантів з захистом авторських прав	Водяні знаки, DRM, авторські ліцензії
Audible Magic	Виявлення та управління авторськими правами на цифровий контент	Цифрові водяні знаки, автоматичне виявлення порушень, блокування контенту

Висновки

У результаті проведеного дослідження та розробки було успішно створено онлайн сервіс продажу аудіо з вдосконаленням захистом та водяними знаками.

Порівняльний аналіз аналогів та методів стеганографічного вбудовування ЦВЗ підтвердив необхідність створення власного сервісу

Обґрунтування вибору технологій та архітектури сервісу, разом з розробкою модулів шифрування та водяних знаків, підтверджують його ефективність та надійність.

ДЯКУЮ ЗА УВАГУ!

Додаток Е. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка захищеного онлайн сервісу продажу аудіозаписів із захистом авторського права стійкими цифровими водяними знаками

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

ПОКАЗНИКИ ЗВІТУ ПОДІБНОСТІ UNISCHECK

Оригінальність 97 %

Схожість 3 %

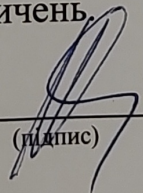
Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

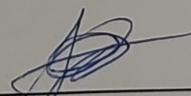
Особа, відповідальна за перевірку


(підпис)

Коваль Н.П.
(прізвище, ініціали)

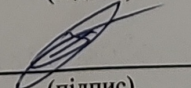
Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи


(підпис)

Адаменко І.О.
(прізвище, ініціали)

Керівник роботи


(підпис)

Карпінєць В.В.
(прізвище, ініціали)