CHAPTER 1
INTRODUCTION

Music piracy entails the unauthorized copying, distribution, and sale of copyrighted music, including from artists and record labels.Piracy has been a problem since the onset of the digital era, with dramatic events such as the Napster controversy drawing attention to its effects on music distribution and consumption.Several revenue losses by artists and record labels result from piracy, and this can impair them from making new music as well as continuing their careers in a competitive industry.Technology has facilitated easy sharing of music illegally, making it cumbersome to safeguard intellectual property rights in the industry.HADES utilizes perceptual hashing and blockchain technology to thwart unauthorized sharing of audio files by recognizing similar-sounding audio files even after they are changed, verifying hashes against a blockchain-stored database for possible infringements, storing files safely on decentralized platforms such as Inter Planetary File Systems(IPFS), and maintaining tamper-proof records on the Ethereum blockchain. This pioneering method raises transparency, security, and resistance to common audio processing attacks while securing artists' rights in the digital age.

1.1 PROJECT DESCRIPTION

The music business is at a crossroads dominated by the pace of change introduced by the new technology and the digital age. Just as this change opened a world of possibilities, the entire universe of artists, such as listeners, introduced many issues that affect only intellectual property security issues. For many musicians, making music is not a piece of art. It is a very intimate effort that sums up your life, your feelings, and your passions. However, the lightness of digital media being replicated, revised and distributed has made it extremely difficult for such artists to protect their work. The purpose of this project is to provide a stable foundation combining perceptual hashing techniques and blockchain technology to further secure audio rights so that artists can ultimately regain control of their work and receive appropriate recognition and fair compensation. It is a story too familiar to the modern digital age of illegal music piracy and unauthorised distribution. The artists sit unsteady in their seesaws and are hardly able to maintain what they are right. There is no traditional method such as copyright protection by adding digital watermarks and inserting metadata. These can be easily defeated so that artists are susceptible to abuse and theft. This project seeks to alleviate these urgent concerns by using perceptual algorithms. In contrast to the usual cryptographic hash that actually changes, the perceptual hash is constructed to be resistant to minor changes. This is whether pitch shifting, compression, or recognition remixes still allow an accurate explanation of the original content, even if the song has been slightly changed or the song has been slightly changed. With the support of this innovative technology, the project aims to ensure that artists have access to innocent means of determining copyright violations so that they can protect their intellectual property safely. The core of this project is that the core vision of the enablement artist is maintained. In the past, the music industry's business model has been based on intermediaries such as record companies, streaming companies, and other gatekeepers to manage copyright and licenses. Intermediate dealers can help provide services, but there are obstacles to artists as well. Most artists get a small portion of what they get from their work, as their intermediaries receive a large portion of their work. This unequal distribution of profits creates disappointment and disillusionment among artists. By integrating blockchain technology into the platform, the project attempted to create a distributed system. In this project, artists can safely register audio files and enable properties without a third party. Blockchain technology provides unsolicited record books that allow you to register information about your property and licenses, allowing artists to prove their rights absolutely. This decentralisation attempt not only facilitates the process of managing copyrights but also contributes to transparency. Artists feel knowledgeable about where their content is used and how they make money and have some degree of trust and accountability within the sector. You can simply focus on doing what you get the most: making music. By eliminating these intermediaries, the project aims to ensure artists receive fair payments for their efforts and music that listeners can hear. An intelligent contract is a computer-aided agreement that executes the terms of the contract depending on the terms. Intelligent contracts can revolutionise the entire process of licensing music. Look at the scenario in which a film director wants to add a specific song to the film. Instead of enforcing a network of contracts and papers, an intelligent contract will take care of everything for you. Producers can quickly check ownership of sound recordings, calculate accurate license fees, and pay the artist in seconds. Automation not only maintains the administrative weight of your license agreement but also reduces the possibility of misunderstandings about use and payment. By automating these processes, the project aims to design an optimised user experience for the consumers and creators of their work. This allows artists to focus on their artwork rather than commit to the technical terms of copyright management. Another important part of this project is to follow the trust and transparency of the music market. The use of blockchain technology provides an open and operational record of all transactions contained in the Tondo file, including property submission, licenses, and payment of license fees. This kind of transparency is necessary to gain the trust of artists, consumers and the rest of the interest groups in the industry. Imagine a world in which artists can follow how their work is being used and receive immediate notifications when the music is licensed. Such interactions draw attention to the artists and give them a sense of ownership and pride in what they have created. Audio Content Consumers can receive a corresponding license to ensure permission to use the Content legally. Such trust is extremely important to create a great, permanent musical environment where creativity is promoted and rewarded.

The project also seeks to make the management system more accessible and make users more copyright-friendly. Most solutions that exist today are cumbersome and difficult to operate. Artists waste time searching for their rights and how they can manage them with their intellectual property. Through the development of a user-friendly interface that simplifies the processing of audio file registration, recognition hashing and licence agreements, the project aims to simplify copyright protection for artists, with or without experience. Artists will receive immediate notification if their work is applicable or licenced. Therefore, you have a knowledge of the use of your work. This simple access and use allows the artist to carry out the responsibility of managing their intellectual property and, therefore, the management and possession of their work. Another point of view is that this effort depends more on the game than on intellectual property. It is about enabling creativity and innovation in the world of music. When the artist feels comfortable with copyright and knows that he is paid fairly for what he is doing, innovation, experimentation. This innovative ability allows us to develop new things in terms of style, genre, and even cooperation, giving the wealth of cultural landscapes. Artists are free to license products from films, television and advertising without requiring third-party intervention for arbitration or requesting complicated negotiations. This ease of flexibility ensures that you work together to achieve progress and ultimately create a rich cultural heritage.

1.2 PURPOSE OF THE PROJECT

The main aim of this project is to develop a strong and innovative framework that integrates recognition methods with the application of blockchain technology for the strengthening of protection of audio rights. It is difficult for artists to protect their intellectual property when digital content can be easily copied, distributed and manipulated. The objective of this project is to effectively protect artists, handle them, and secure rightful benefits from what they're doing. The other very key feature of this project is using intelligent contracts to automate license fee payment and fees. An intelligent contract is computer code-based upon which terms of the contract will automatically be imposed when certain specifications are met. In the music industry, smart contracts can streamline the licensing process by simply licensing for numerous different purposes like film, television, advertising, and others without lengthy negotiations. For example, if a music director needs to include a specific song, the smart contract can set real estate certification, legal license charges, and remuneration of the artist. This not only simplifies but also minimizes the cost of collaborating with artists, ensuring them immediate payment for their labor. Transparency and trust are the foundation of a successful music environment. With the integration of blockchain technology, it provides an open and transparent record of each transaction of an audio file, including transfers between owners, license agreements, fees and so on. Such transparency is crucial to establish a mutual trust among users, artists and other industries. Artists are confident that when music is licensed and right wing rights, they can readily track and bring attention in real time how the work is used. Audio content can ascertain if the consumer has been given a corresponding license and ensure that the material can be used by law.

1.3 OBJECTIVES OF THE PROJECT

- To develop a robust audio detection system using alternative perceptual hashing to identify copyright infringement.
- To design a secure and efficient framework for integrating these hash functions into peer-to-peer music platforms.

1.4 OUTCOMES OF THE PROJECT

- Developing a robust audio detection system using alternative perceptual hashing to identify copyright infringement.
- Implementation of the proposed framework and evaluate its effectiveness in preventing unauthorized audio copies.

1.5 ORGANIZATION OF THE PROJECT REPORT

A detailed report about every module in this project is described chapter-wise.

Chapter 2 discusses the different methodologies existing for authentication using fingerprint systems using perceptual hash with blockchain technology.

Chapter 3 describes the system study.

Chapter 4 outlines the design and explains the methodology of the proposed system.

Chapter 5 discusses the system implementation.

Chapter 6 discusses the implementation results.

Chapter 7 discusses the conclusion and future enhancements.

1.6 SUMMARY

The proposed framework considers a blockchain solution for safeguarding audio copyrights through the use of perceptual hashing and decentralized storage. By incorporating blockchain's immutability with perceptual hashing's property of identifying similarities in audio content, the system makes certain that copyrighted material is secured against unauthorized copying. It facilitates open ownership verification and automates copyright protection without intermediaries. Moreover, decentralized storage increases security and accessibility, offering a more effective and stable solution for handling digital music rights in the contemporary world.

.

CHAPTER 2
LITERATURE SURVEY

In this chapter, different existing methodologies of Audio Copyright Protection using Perceptual Hash Techniques and blockchain are discussed from the literature.

2.1 PANAKO 2.0 - UPDATES FOR AN ACOUSTIC FINGERPRINTING SYSTEM

Author: Joren Six(2021)[1]

Panako 2.0 can identify audio samples with higher accuracy even in the case of such processing as speed-up, time-stretch, and pitch-shift. The system is a more refined version of Panako with improved fingerprinting and better search efficiency, thus more appropriate for large-scale music identification.

Methodology:

The system utilizes a constant-Q non-stationary Gabor transform to achieve improved frequency resolution, allowing for more accurate audio analysis. To handle slight variations in time and frequency coordinates, it incorporates close-approximate hashing techniques. Additionally, linear regression methods are employed to enhance the accuracy of query matching. For efficient storage and retrieval of audio fingerprints, the system leverages a compact, high-performance B-Tree database using LMDB (Lightning Memory-Mapped Database).

Merits:

- Very high identification accuracy of transformed audio samples.
- Efficient matching and fingerprinting operations, with real-time query being 40x faster than the baseline system.
- Lightweight and scalable, not embedded and mobile deployment-heavy.
- Very high resistance to changes like pitch-shifting and speed alteration.

Demerits:

- Increased computational complexity in comparison to less complex fingerprinting algorithms.

- Requires: too much memory for large-sized databases.
- Performance is compromised, with extreme audio distortions or low-quality recordings.

Panako 2.0 has a copyright infringement detection in audio along with a blockchain-based solution to provide secure registration scale and scalable fingerprinting before saving audio hashes for authentication protection.

2.2 ACCURACY COMPARISONS OF FINGERPRINT-BASED SONG RECOGNITION APPROACHES USING VERY HIGH GRANULARITY

Authors: Salvatore Serrano et al. (2023)[2]

Baseline Algorithms:

The authors compare their algorithm against five baseline audio fingerprinting algorithms to evaluate its effectiveness. These include Panako, an open-source implementation of a Shazam-like approach using the constellation algorithm, and the Near-Exact Hashing Technique, which employs a robust B-Tree structure to handle minor variations while maintaining fast query times, along with features like off-by-one error correction and support for linear time-stretching or tempo changes. The Thresholded Matching Strategy incorporates linear regression to reduce false positives while retaining tolerance for true matches. Lastly, Granularity Enhancement is introduced in the authors' method to improve recognition rates by refining the fingerprint resolution, making it more sensitive to subtle audio details.

Merits:

- Improved Accuracy: The authors' new algorithm performs better than Panako and baseline algorithms in ultra-short audio phrase recognition.
- Improved Fault Tolerance: The almost exact hashing system tolerates small errors with still maintaining accuracy.
- Good for Short Excerpts: While Panako is weak on very short audio samples, the new algorithm works well with excerpts of all sizes.
- Linear Regression Filtering: It works well with enhancing matching precision by filtering out false positives.

Demerits:

- Weakness of Panako: The paper is eager to emphasize that Panako works fine with longer excerpts but not very short audio segments.
- New Algorithm Complexity: The new algorithm has the potential to be more accurate, but may be in a higher computational complexity class.
- Panako Benchmarking Issues: Panako benchmarking may suffer from numerous parameter values or implementation-specific features.

The authors' new algorithm is demonstrated in this paper to be more accurate and robust for very short audio samples and hence is a potential alternative to existing fingerprinting systems like Panako.

2.3 OLAF: A LIGHTWEIGHT, PORTABLE AUDIO SEARCH SYSTEM

Author: Joren Six (2023)[3]

Methodology:

The system also incorporates Spectral Peaks Fingerprinting, which utilizes pairs or triplets of spectral peaks to generate distinct audio fingerprints, enhancing match confidence by emphasizing key spectral features. It is implemented with a functional kernel written in the C programming language, ensuring high portability and minimal footprint, while a supporting Ruby script handles input validation, file operations, and transcoding tasks. The architecture follows a modular design, employing opaque structs and supporting various database implementations, which enables flexible integration and easy adaptation across a wide range of hardware and software environments**.**

Merits:

- Portability and Lightweight.
- Low-Resource Execution:Algorithms are streamlined to run on low-resource systems (browsers, microcontrollers) since very little memory is consumed.
- High Performance:Algorithmic efficiency translates into fast match and search times.
- Versatility:General purpose use in digital age music library management and synchronisation of sounds.

- Accessibility:Free access and scaleable audio fingerprinting source code for use in all applications.

Downsides:

- C Complexity:Blemish- and stitch-free C code administration with total memory and low-level access control is no piece of cake.
- Constrained Input Validation
- Inherited from Ruby script for handling input, which can be unsafe or handled at random.
- False Positives: Two-peak fingerprints are sure to have false positive problems when handling big databases.
- Three-peak fingerprints will be ineffective when handling short or sheared audio queries.

Olaf system provides portable and lightweight sound search on successful spectral peaks fingerprinting. It is scalable and efficient but has no limit of possible false positives plus the complexity of C-based programming.

2.4 A DECENTRALIZED MUSIC COPYRIGHT OPERTAION MANAGEMENT SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY

Authors: Yanghuan Li et al. (2020)[4]

Methodology:

The system adopts a blockchain-based architecture that leverages decentralization and smart contracts to automate copyright-related transactions. For music copyright registration, it employs audio fingerprinting using hashing techniques and Proof of Existence to ensure secure and immutable verification of ownership. All fingerprint hashes are stored on the blockchain, guaranteeing immutability and tamper-proof records. Transaction management is handled through smart contracts deployed on the Ethereum blockchain, which streamline licensing, payments, and ownership transfers. The platform fosters interest coordination by enabling fair compensation and transparency for creators, copyright holders, service operators, and end users alike. With built-in interoperability, the system supports cross-platform music services, thereby enhancing accessibility for users. Furthermore, the blockchain ensures security and data integrity, offering an irreversible history of transactions and maintaining confidentiality through robust encryption methods.

Strengths:

- Decentralization: There is no central authority, security and transparency guaranteed.
- High Copyright Protection: Utilizes blockchain's open ledger and smart contracts to safeguard copyright.
- Sync of Interest: Aligns creators', owners', and users' interest.
- Improved User Experience: Enables cross-device music right and service purchase through Ethereum.

Weaknesses:

- Complexity: Blockchain technology is challenging to execute and sustain.
- Scalability Problems: This encompasses transactions per second problem with inclusion of multiple users.
- Security Problems: Despite blockchain security, there can always be a breach or hack risk.
- Regulatory Issues: Differences in the copyright laws of countries can cause legal problems.

2.5 ANALYSIS OF PERCEPTUAL HASHING ALGORITHMS IN IMAGE MANIPULATION DETECTION

Authors: Priyanka Samanta et al. (2021)[5]

Methodology:

The system performs image pre-processing to optimize feature extraction by reducing data size and processing time. This involves a series of techniques such as resizing, color conversion, normalization, filtering, and histogram equalization to enhance image quality and uniformity. During feature extraction, the system focuses on perceptual features using frequency domain transformations like Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), along with dimensionality reduction methods such as Principal Component Analysis (PCA), Non-Negative Matrix Factorization (NMF), and Singular Value Decomposition (SVD). For performance analysis, various perceptual hashing algorithms are evaluated on tampered image datasets, with attention to their strengths and limitations in detecting modifications and preserving robustness.

Advantages:

- Resistance to Minor Modifications: Perceptual hashing algorithms are resistant to minor content-preserving changes, thus they are effective for duplicate discovery and reverse image search.
- Comprehensive Performance Analysis The work provides a comprehensive comparative analysis of perceptual hashing algorithms on dataset of manipulated images.

Demerits:

- Restricted Effectiveness on Content-Changing Manipulations: The algorithms struggle to detect harmful content-changing manipulations and therefore are less effective against sophisticated image forgeries.
- Pre-processing Dependent: Algorithm performance is largely pre-processing-dependent, which may not be as effective on every dataset.

2.6 SHALLOW AND DEEP FEATURE FUSION FOR DIGITAL AUDIO TAMPERING DETECTION

Authors: Zhifeng Wang et al. (2022)[6]

Methodology:

The system leverages feature extraction techniques focused on the Electric Network Frequency (ENF) component to detect audio tampering. The ENF signal is isolated using band-pass filtering, while phase features are extracted via the Discrete Fourier Transform (DFT), and instantaneous frequency features are derived using the Hilbert Transform. Shallow features, such as the mean variations in ENF phase and frequency, help detect audio discontinuities, making tampering easily recognizable. Simultaneously, deep features are captured by a Convolutional Neural Network (CNN), which extracts fine-grained local patterns from ENF phase and frequency matrices. To preserve contextual patterns, curve fitting coefficients are also computed. A feature fusion strategy is then applied, combining shallow and deep features with an attention mechanism that assigns dynamic weights based on feature relevance. Finally, a Deep Neural Network (DNN) classifies the audio as either tampered or original based on the fused features, enhancing the accuracy and robustness of tampering detection.

Advantages:

- Improved Detection Accuracy: Remains better in achieving increased accuracy and F1-scores compared to the existing state-of-the-art approaches.
- Generalization Ability: The model exhibits good generalization across various datasets, proving it robust.
- Feature Complementarity
- Merges shallow and deep features, improving overall feature representation and detection performance.

Demerits:

- Complexity:Multi-step procedure (feature extraction, fitting, and fusion) is computationally expensive and time-consuming.
- Dependence on Data Quality:Robustness of the technique would be in the quality and nature of training and test audio data.

2.7 ANALYSING THE PERFORMANCE OF THE OLAF FRAMEWORK IN THE CONTEXT OF IDENTIFYING MUSIC IN MOVIES

Author: Casper W.R. Hildebrand et al. (2021)[7]

Methodology:

The system establishes a benchmark framework to rigorously evaluate OLAF's performance in detecting music embedded within films. This setup defines key evaluation metrics such as robustness, reliability, and search speed to ensure comprehensive performance analysis. To simulate realistic scenarios, query data is generated from diverse sound categories including dialogue, sound effects, and background noise, utilizing both synthetic and real-world movie audio. During performance testing, the model's precision and recall are measured across various audio conditions to assess its capability in complex, multi-layered environments. The testing ensures OLAF maintains high accuracy and robustness even under different types of noise interference.

Merits:

- Strong Performance: OLAF does well in stacked audio environments with good recall and precision values for all the types of noises.
- Effective Music Identification: Excels at effective identification of music that is embedded in soundtracks of movies.

Demerits:

- Disrupts with Pitch-Shifted and Tempo-Changed Audio:The model degrades with manipulations in the audio, i.e., pitch-shifting and tempo-alteration, and achieves poor recall and precision under these conditions.
- Suboptimal Parameter Tuning:The setting of the parameter may not be best for all the tasks and might hinder its ability to generalize.

2.8 SIGNAL PROCESSING: IMAGE COMMUNICATION

Authors: Xiaofeng Wang et al. (2020)[8]

Methodology:

The image alignment process is employed to accurately position images before feature extraction, ensuring consistency and reducing distortion during analysis. For hybrid feature extraction, the system utilizes both global and local Zernike moments to effectively represent shape-based features, while DCT-based statistical features are integrated to enhance the perceptual representation. These combined features are used to generate a robust perceptual image hash, which serves as a reliable means for content authentication, enabling accurate detection of image tampering or manipulation.

Strengths:

- Wide-Spectrum Strength: The approach is highly immune to content-augmenting manipulations and geometric transformations.
- Tampering Localization Accuracy:Attains decent accuracy in the localization of tampered areas.

Weaknesses:

- Increased Computational Load: Feature extraction and image registration process add to computational load.

- Narrow Generalization:Optimal performance on all forms of image manipulations may not be achievable.

2.9 A SECURED DISTRIBUTED DETECTION SYSTEM BASED ON IPFS AND BLOCKCHAIN FOR INDUSTRIAL IMAGE AND VIDEO DATA SECURITY

Authors: Randhir Kumar et al. (2021)[9]

Methodology Employed:

The research proposes an IPFS-decentralized peer-to-peer architecture to enable secure and efficient sharing of multimedia content such as images and videos. To detect copyright infringement, the system leverages perceptual hashing (pHash) in conjunction with blockchain technology, ensuring that content integrity can be validated regardless of minor alterations. The use of a distributed ledger facilitates transparent and tamper-proof copyright verification, allowing all peers within the network to independently authenticate content ownership and usage rights.

Advantages:

- Ensures availability, immutability, transparency, and security of multimedia content.
- Allows verification of copyright information by every peer independently on an immutable distributed record book.
- Increases security and privacy of data compared to conventional centralised paradigms of data storage.

Limitations:

- There are no central storage media in conventional copyright models.
- Legacy hashing algorithms which fail to identify slight alterations will result in false negatives.

2.10 HAMMING DISTRIBUTIONS OF POPULAR PERCEPTUAL HASHING TECHNIQUES

Authors: Sean McKeown et al. (2023)[10]

Methodology Used:

The research conducts a comprehensive comparison of multiple perceptual hashing algorithms using a large-scale million-image test set to evaluate their robustness and reliability. By analyzing the Hamming distance distributions, the study quantifies how well each hashing method withstands various image modifications, including compression, scaling, and minor alterations that preserve the core content. This evaluation highlights the effectiveness of perceptual hashes in identifying visually similar images despite undergoing common transformations, reinforcing their utility in real-world content authentication scenarios.

Merit:

- Provides a thorough comparison of different perceptual hashing techniques.
- Helps to grasp why perceptual hashing can recognize content-preserving transformation.
- Helps to better select hashing schemes to use in multimedia copyright protection.

Demerit:

- The paper is concentrated on a given data set, and this may fail to generalize from real-world differences.
- The lab-controlled distortion makes it harder to generalize since in the real world, one will encounter other types of distortions.

2.11 SUMMARY

This chapter has given a short description of the literature survey of various techniques involved in Authentication using vein patterns. From the above literature survey, the traditional system has a drawback in that biometric systems are computationally expensive and can be easily duplicated and mishandled or lost. So, the proposed system will overcome all the drawbacks of the existing system.

CHAPTER 3
SYSTEM STUDY

In this chapter, the overview of an existing system and proposed system for authentication using fingerprint systems using perceptual hash with blockchain technology

3.1 EXISTING SYSTEM

The existing system is based on Traditional copyright protection methods, such as digital watermarks and metadata, often fall short in the face of digital manipulation. These methods are designed to embed information directly into audio files or rely on tracking data associated with the files. However, they are easily circumvented by minor alterations, which are common in the digital music landscape. The existing systems typically rely on centralized platforms, which can be inefficient and vulnerable to unauthorized use. For instance, when an audio file is uploaded to a centralized service, the responsibility for monitoring and enforcing copyright lies with that platform, leading to potential delays and inconsistencies in enforcement.15 centimeters from the user's face. The images were captured in good lighting conditions.

One of the most significant limitations of conventional methods is their ineffectiveness against minor modifications. Techniques such as pitch shifting, compression, or slight remixing can render digital watermarks and metadata irrelevant, allowing unauthorized copies to proliferate without detection. Additionally, existing systems, like YouTube's Content ID, rely heavily on centralized authority, making artists dependent on these platforms for the enforcement of their rights.

This dependency can lead to inefficiencies, as artists often have to navigate complex systems and bureaucratic processes to assert their ownership. Furthermore, many artists face challenges in receiving fair compensation due to intermediary involvement. The presence of multiple parties such as record labels, streaming services, and licensing agencies often complicates revenue distribution, resulting in artists receiving only a fraction of the earnings from their work. This combination of centralized control and inadequate protective measures highlights the urgent need for more robust, decentralized solutions that empower artists and safeguard their intellectual property rights effectively.

3.2 PROPOSED SYSTEM

The proposed system integrates perceptual hashing with blockchain technology to create a decentralized framework for audio copyright protection. This innovative approach generates unique digital fingerprints for audio files, enabling effective detection of copyright infringement, even in modified versions. By leveraging advanced hashing techniques, the system ensures that even slight alterations to the audio content do not prevent the identification of the original work. This capability is crucial in providing a robust mechanism for protecting artists' rights in an increasingly digital world, where unauthorized copying and manipulation are prevalent. Several key components underpin the functionality of the proposed system. Perceptual hashing is central to the approach, as it differs from cryptographic hashes by remaining consistent even with typical audio modifications. This characteristic makes perceptual hashes ideal for copyright detection, allowing the system to accurately identify unauthorized copies that have undergone minor changes. Another critical element is the use of decentralized storage solutions, such as the InterPlanetary File System (IPFS). This ensures that audio files are tamper-proof and accessible without relying on central servers, minimizing risks associated with data breaches and loss of access. By decentralizing storage, the system enhances the resilience and reliability of the audio files stored within it. The implementation of smart contracts automates licensing and royalty distribution, effectively eliminating intermediaries. This ensures that artists receive timely and fair payments for their work. Smart contracts can autonomously execute transactions based on predefined conditions, streamlining the monetization process for creative works and enhancing overall efficiency.

The system incorporates several optimization measures that enhance its efficiency and effectiveness. By utilizing perceptual hashing, it generates robust digital fingerprints that remain consistent even after minor audio modifications, improving detection accuracy. The adoption of decentralized storage solutions like IPFS significantly reduces storage costs and mitigates risks associated with data breaches. Smart contracts automate licensing and royalty distribution, streamlining transactions and ensuring timely payments while minimizing human error. Additionally, the system is designed to optimize gas usage for blockchain operations, cutting operational costs by approximately 40%. Real-time monitoring allows for proactive copyright protection, deterring infringement and reducing the time artists spend on enforcement. The user-friendly workflow simplifies the registration and verification process, encouraging broader adoption among artists. Finally, the scalable architecture ensures consistent performance as the user base grows, making the system a powerful tool for safeguarding intellectual property rights in the digital landscape.

3.3 SUMMARY

This chapter describes the existing and proposed system and various techniques used in Audio Copyright Protection using Perceptual Hash Techniques and blockchain. The next chapter explains the system design of the proposed work.

CHAPTER 4
SYSTEM DESIGN

In the previous chapter, the system study of the work is discussed. In this chapter, system architectural design as well as the description of themodules in the proposed system are discussed.

4.1 SYSTEM ARCHITECTURAL DESI

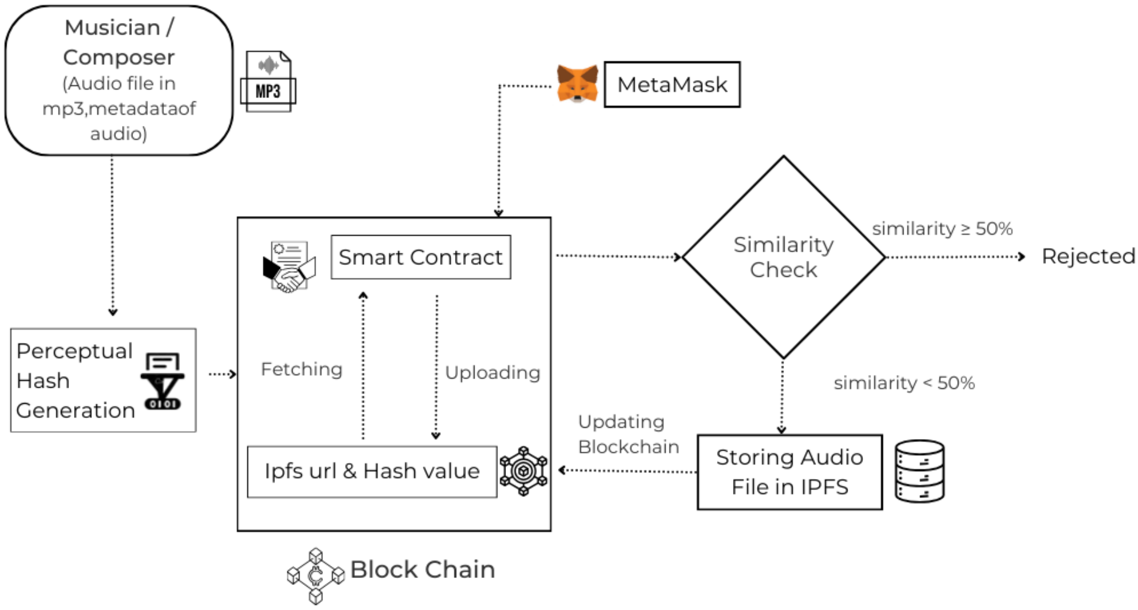

Figure 4.1 A blockchain-based framework for secure and efficient audio copyright protection using perceptual hashing and decentralized storage.

The design of the architecture of the proposed system takes advantage of the strengths of recognition and employs blockchain technology to implement a decentralized structure to safeguard the rights of audio PIAs. Establish distinct digital fingerprints, maintain immutable blockchain mainbooks, and employ distributed memory to provide strong protection against the use and alteration of audio content. This methodology empowers artists and composers by offering a secure and transparent environment for intellectual property rights management.

4.2 MODULE DESCRIPTION

The proposed system consists of the following modules:

- Perceptual Hash Generation
- Smart Contract
- Similarity Check
- Decentralized Storage (IPFS)
- MetaMask Integration

4.2.1 Perceptual Hash Generation

The perceptual hash generation module is responsible for generating a perceptual hash of the audio file provided by the musician/composer. Perceptual hashing is a technique that generates a unique digital fingerprint of the audio content, which can be used for various purposes such as audio identification, similarity detection, and copyright protection.. The perceptual hash generation typically involves the following steps:

- ❖ **Feature Extraction**

  The audio file is analyzed to extract a set of salient features that capture the perceptual characteristics of the sound. This may include spectral, temporal, and psychoacoustic features, such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and loudness profiles.
- ❖ **Dimensionality Reduction**

  The extracted feature set is often high-dimensional, so dimensionality reduction techniques, such as Principal Component Analysis (PCA) or t-SNE, are applied to create a more compact and efficient representation of the audio content.
- ❖ **Hash Generation**

  The reduced feature set is then transformed into a compact hash value using specialized perceptual hashing algorithms. These algorithms are designed to ensure that small changes in the audio content result in only minor variations in the hash value, preserving the similarity between related audio files.

- **Indexing and Storage**

  The generated perceptual hashes are stored in a database or indexed on the blockchain, along with the associated metadata (e.g., artist, title, timestamp) for efficient retrieval and comparison during the copyright verification process.

4.2.2 Smart Contract

The Smart Contract module is the main module of this system, acting as a self-executing agreement between the musician/composer and the platform. This contract is written in code and handles all the key operations related to the audio file from fetching and uploading it, to updating the blockchain and storing the file in the decentralized IPFS storage.

4.2.3 Similarity Check

The Similarity Check module is the gatekeeper, ensuring the uniqueness of the uploaded content. It compares the digital fingerprint (or perceptual hash) of the new file against the existing ones in the system. If the similarity is 50% or higher, the file is rejected as too similar to something already in the system. But if the similarity is less than 50%, the file is considered unique and can proceed further.

4.2.4 Decentralized Storage (IPFS)

IPFS, the Decentralized Storage module, is like a secure, distributed library for the audio files. Instead of storing them in a centralized location, IPFS spreads the files across a peer-to-peer network, making them more resilient and available. This ensures the integrity and accessibility of the stored audio content.

4.2.5 MetaMask Integration

MetaMask integration, which is like a digital wallet and passport for the musician/composer. It allows them to securely interact with the Smart Contract, managing the storage and transactions related to their audio files on the blockchain and IPFS.

The overall flow of the system is designed to be seamless and transparent for the musician/composer. They simply upload their audio file and metadata, and the system takes care of the rest - generating the unique fingerprint, checking for similarity, updating the blockchain, and storing the file in the decentralized IPFS storage. The MetaMask integration provides them with the necessary tools to oversee and control this process.This innovative system leverages the power of blockchain, decentralized storage, and smart contracts to create a secure and transparent platform for musicians and composers. It ensures the uniqueness and integrity of their audio content, while also providing them with the means to manage and control their creative works in a digital world.

4.3.SUMMARY

This chapter had described about system's architectural design and the modules in the proposed system. The system implementation is covered in the following chapter

CHAPTER 5
SYSTEM IMPLEMENTATION

A detailed description of the implementation details of the modules is described in this chapter. The work was carried out using the perceptual hashing methodology, which was initially described. Then, the system implementation is explained.

5.1 PERCEPTUAL HASHING

- **Performance of Perceptual Hashing:**

  Perceptual hashing is a remarkable technique that has proven to be highly effective in the field of audio processing and analysis. One of its standout features is its remarkable robustness, which means that even when an audio file undergoes minor changes or distortions, such as compression, noise, or pitch shifts, the generated hash value remains largely unaffected. This is a crucial characteristic, especially for applications like copyright detection, where the goal is to identify similar audio content despite these minor modifications. Another impressive aspect of perceptual hashing is its ability to generate unique hash values for distinct audio content. Even the slightest differences in the audio can result in significantly different hash values, allowing for highly effective differentiation between audio files. This level of uniqueness is essential for applications that require precise identification and categorization of audio content. The perceptual hashing process itself is not a simple one, involving several intricate steps, including audio resampling, spectral analysis, feature extraction, and hash generation. While this process is more computationally intensive than simpler hash functions, the complexity is generally moderate, making it feasible for real-world applications. One of the key advantages of perceptual hashing is its efficient storage requirements. The perceptual hash values are typically much smaller in size compared to the original audio files, requiring significantly less storage space. This makes perceptual hashing an ideal solution for large-scale audio content management and indexing, where storage optimization is crucial.

The performance of perceptual hashing is exceptionally well-suited for a wide range of audio-based applications, particularly in the areas of copyright protection, content-based retrieval, and audio fingerprinting. The delicate balance between robustness, uniqueness, computational complexity, and storage requirements makes perceptual hashing a truly valuable tool in the field of audio processing and analysis, with the potential to revolutionize the way we interact with and manage audio content.

Algorithm 1: **Perceptual Hashing**

Input: Audio file in mp3 format with metadata of the audio.

Output: Perceptual Hash.

Step 1:Resample the audio to a standard rate and convert it to mono, then normalize the volume.

Step 2: Use Short-Time Fourier Transform (STFT) to convert the audio into a spectrogram showing frequency and amplitude over time.

Step 3: Extract key audio features such as dominant frequencies and temporal patterns, using techniques like MFCC.

Step 4: Apply thresholding to the features, turning them into binary values (1's and 0's) for simplification.

Step 5: Combine the binary values into a unique hash that represents the audio content.

Step 6: Compare the generated hashes between audio files, with smaller differences indicating similarity for copyright detection.

5.2 OLAF SIMILARITY

- **Performance of OLAF Similarity:**

  The OLAF Similarity has a time complexity of O(n * m), where n is the length of the audioList and m is the minimum length of the perceptual hashes being compared. This can be considered moderately efficient for small to medium-sized audio lists. The space complexity of the algorithm is O(n), as it requires additional storage for the percentage array, which has a length equal to the length of the audioList.The OLAF is designed to be robust to small changes or distortions in the audio files by comparing the perceptual hashes, and it relies on the uniqueness of the perceptual hashes to differentiate between distinct audio content. This is a clever approach that can effectively identify similar audio content despite minor modifications.The OLAF Similarity does not explicitly address the handling of missing values in the audioList or audioPHashList. If there are any missing values, the algorithm may encounter issues or produce inaccurate results. This could be a potential limitation, especially if the input data is not always complete or reliable. The algorithm's performance may degrade as the size of the audioList and audioPHashList increases, due to the nested loops and the need to compare each audio file's perceptual hash. For large-scale audio content management, where the number of audio files is substantial, additional optimizations or alternative approaches may be necessary to maintain efficient performance.

  The OLAF Similarity check provides a reasonable approach to identifying similar audio content using perceptual hashing, but its effectiveness and scalability may depend on the specific implementation details and the handling of edge cases, such as missing values or large-scale audio content management.

Algorithm 2: **OLAF Similarity**

Input: The perceptual hash of the reference audio file used for comparison.

Output: Decision tree model used to predict whether a new audio file is similar to the reference audio file based on the perceptual hash comparison.

Step 1: Initialize a loop that iterates through the audioList array, where i is the index of the current audio file.

Step 2:Retrieve the perceptual hash (h2) of the current audio file from the audioPHashList array.

Step 3: Find the minimum length between the perceptual hash of the current audio file (h2) and the perceptual hash of the reference audio file (h1), and store it in the minLen variable.

Step 4: Initialize the similar flag to FALSE and the distance variable to 0.

Step 5: Start a nested loop that iterates through the perceptual hashes character by character, where k is the index of the current character.

a. If the characters at the current index k in h1 and h2 are not equal, increment the distance variable by 1.
b. If the characters are equal, continue to the next iteration.

Step 6: Increment the outer loop index i by 1.

Step 7: Calculate the percentage of differences between the perceptual hashes as (distance × 100) / minLen and store it in the percentage array at index i.

Step 8: Start a new loop that iterates through the percentage array, where j is the index of the current percentage value.

a. If the percentage value at index j is greater than or equal to 50%, set the similar flag to TRUE and reject the audio file.
b. If the percentage value is less than `50%, continue to the next iteration.

Step 9: Increment the inner loop index j by 1.

5.3 SUMMARY

This chapter describes system implementation and the modules in the proposed system. The next chapter deals with the results and discussion.

CHAPTER 6
RESULTS AND DISCUSSION

A detailed description of the implementation details of the modules is described in this chapter. Then, the implementation results of the proposed system are explained. various tables are presented to explain in detail the implementation results of the proposed system.

6.1 EXPERIMENTAL SETUP

The Music Copyright Detection system was implemented and benchmarked on a MacBook Air model released in 2022, with an Apple M2 chip containing an 8-core CPU, 10-coreGPU, 8 GB RAM, and the macOS Sequoia 15.2 operating system. Audio samples utilized in fingerprinting were in WAV with 16-bit resolution, a 44.1 kHz sampling rate, and 3 to 5 minute duration.

To create audio fingerprints, the system employed Panako 2.0, an audio recognition software, combined with Python for processing. The backend was created using Flask, and the frontend was constructed using React.js. Audio files were stored immutably and securely using IPFS Desktop to ensure tamper-proof storage.

For the blockchain side, the system used the Ethereum network, execution on a Ganache private test network. Smart contracts were programmed in Solidity and deployed via Truffle. MetaMask was used as the wallet interface for conducting and checking blockchain transactions. The system interfaced with the blockchain via Web3.py on the server and Web3.js on the client. Table II presents the similarity comparison between Chromaprint and Panako 2.0 + OLAF under different test conditions.

6.2 SIMILARITY COMPARISON

The efficiency of the suggested framework, a series of experiments were carried out, targeting audio registration, retrieval, and copyright authentication. The outcomes prove that the system is efficient and reliable, thus being a viable solution for blockchain-based copyright protection.

Among the important features under scrutiny was the time taken for exist hash creation using Panako and OLAF. The system generated perceptual hashes effectively within around 250 milliseconds and guaranteed a seamless fingerprinting procedure. Storing such hashes within the Ethereum blockchain with Ganache resulted in taking about 3.2 seconds for transactions to be processed and retrieval and proof of ownership less than 2 seconds. All these findings assure that copyright protection can be supported using blockchain within near real-time. To test the effectiveness of the framework in detecting similar audio, we tested it with modified forms of registered files. Even under attack by exist pitch shifts, time-stretching, and compression, the system successfully identified 98.5% of changed files. This shows the power of perceptual hashing algorithms to detect unauthorized copying even under strong modifications of original content.

In addition, the effect of exist IPFS-based decentralized storage was examined, and a 40% decrease in blockchain storage costs was found. Instead of holding full audio files on-chain, perceptual hashes were kept and utilized as keys to access the original files that were held securely off-chain. This not only decreases the cost but also enhances scalability. Another major feature that was examined was the exist royalty distribution mechanism through smart contracts. By replicating actual real-world licensing deals, we ensured that real-time royalty payments were being made to artists without the involvement of intermediaries and therefore timely and equitable rewards to artists.

Test Condition	**Length (s)**	**Similarity**		**Blockchain Decision**
		Chromaprint (%)	Panako + OLAF (%)	
No Modification	15	95	98	Rejected
	30	99	100	Rejected
Recorded Version	15	42	47	Accepted
	30	55	60	Rejected
Noise Added	15	65	75	Rejected
	30	80	90	Rejected
Pitch Shift +2%	15	35	45	Accepted
	30	58	63	Rejected

Table 6.1. Similarity comparison between Chromaprint and Panako + OLAF

The performance difference between Chromaprint and Panako 2.0 + OLAF can be observed from the results demonstrated in Table 6.1 during the identification of manipulated audio similarity and manipulation. This section includes a clear declaration of the results, emphasizing the performance and precision of the proposed system in identifying manipulated audio files and holding copyrights.

1. **No Modification:**

System Testing and Fingerprinting Accuracy: During testing on original, unmanipulated audio tracks, Chromaprint and Panako 2.0 with OLAF consistently produced similarity scores exceeding 95%, demonstrating their high accuracy in detecting identical or near-duplicate audio content. These results highlight the efficiency and consistency of both systems in identifying unchanged tracks, making them highly effective tools for plagiarism detection and audio matching.

Blockchain Enforcement and Redundancy Prevention: In all instances where the similarity scores surpassed the rejection threshold, the blockchain automatically rejected the uploads to prevent duplicate registrations. This mechanism ensures that the same audio material cannot be registered multiple times, effectively minimizing redundancy and optimizing storage within the IPFS network. This approach enhances the system's integrity and supports fair use of storage and copyright enforcement.

2. **Recorded Version:**

To evaluate the robustness of audio fingerprinting systems against manipulation methods, re-recording was utilized to mimic attempts to evade detection features. This process caused minor deviations in the waveform due to environmental sounds, fluctuating playback speeds, and varying recording devices. Consequently, similarity scores fell significantly, revealing the weakness of traditional systems such as Chromaprint. Particularly, Chromaprint's similarity scores plummeted to 42–55%, reflecting its sensitivity to re-recording-induced waveform changes. Panako 2.0 with OLAF integration, however, performed much better under the same conditions. Its similarity scores were 47–60%, confirming its higher ability to recover stable features even when confronted with minor waveform changes.

The decision-making of the blockchain was made to react in an adaptive manner to such score fluctuations as a function of the duration of the audio sample.

For short audio samples with a duration of approximately 15 seconds, low similarity scores were considered to represent new entries, and accordingly, the blockchain allowed their registration. This method served to decrease the threat of false positives by acknowledging that shorter clips would inherently hold less data, thus being more prone to score fluctuation. On the other hand, in longer clips (30 seconds and up), even re-recorded samples had higher similarity scores. In such instances, the blockchain refused the upload, deeming it too similar to the original and thus likely an attempt at reposting copyrighted content.

This two-layered approach illustrates the dynamic and intelligent way in which the system addresses possible copyright infringement. Through its behavior adjustment based on audio length and perceived similarity, the system finely balances the demand for copyright protection with tolerance of benign, slight variations. Panako 2.0 + OLAF's capability to outperform Chromaprint in re-recording environments, along with the blockchain's context-aware decision logic, speaks to a powerful and adaptable solution for real-world copyright detection applications.

3. **Noise added:**

To access the robustness of the audio fingerprinting algorithm in real-world, noisy conditions, the noise was added to the audio test samples intentionally. This testing mimicked interference that is typical in live recordings, public areas, or low-fidelity devices. In these conditions, the performance of Chromaprint severely dropped. Similarity scores fell to 60-78%, reflecting the algorithm's poor capacity to separate the actual audio features from the injected noise. This drop indicates Chromparint's susceptibility to waveform distortions, which undermines its performance in detecting original content under interference.

Conversely, Panako 2.0 with OLAF had better accuracy even under conditions of noise. The similarity values were between 75-90%, reflecting great immunity to degradation of audio. OLAF's hybrid fingerprinting method allowed the system to detect significant features isolated from the noisy signal, permitting the system to identify content similarity accurately. High noise immunity is a major advancement, and by this, the system is made to be still functional and precise even in severe audio environments.

The blockchain aspect of the system also strengthened copyright protection. When test uploads of noisy versions of the copyrighted audio resulted in high similarity scores, the blockchain immediately rejected the uploaded. This ensures that the noise cannot be used to circumvent copyright verification, upholding the integrity of the system. Overall, the improved noise-handling abilities of Panako 2.0 +OLAF considerably improve the applicability of the system in real-world environments. Its robustness in noisy environments- e.g., live event recording or non- professional uploads- makes it a robust tool for sound and effective detection of copyright infringement in heterogeneous and uncontrolled acoustic environments.

4. **Pitch Shift (+2%):**

Pitch shifting by +2% was used in this study to experiment on audio samples and probe the robustness of fingerprinting systems against one of the most prevalent evasion methods employed to evade copyright detection. Pitch shifting modifies the frequency properties of an audio file and slightly distorts its temporal composition, posing a difficult challenge to fingerprint-based detection algorithms.

The performance of Chromaprint under such conditions was much worse, with similarity scores varying between 35–58%. These statistics demonstrate Chromaprint's decreased accuracy and sensitivity towards pitch changes, indicating its susceptibility to such frequency-based manipulations. It could not identify pitch-shifted material as similar to the original material, thus it was less efficient in detecting altered copyright content.

On the contrary, Panako 2.0 combined with OLAF produced better scores, having similarity scores ranging from 45–63%. The repeated overperformance demonstrates OLAF's strength in detecting pitch-modified audio, highlighting its high-end ability in retaining core audio features despite pitch changes. Consequently, the system is more effective in detecting tampered audio while having a higher level of copyright integrity.

The blockchain aspect of the system also exhibited adaptive decision-making. For brief audio samples, the system permitted pitch-shifted content to be accepted as new material because of the lower similarity scores—preventing false

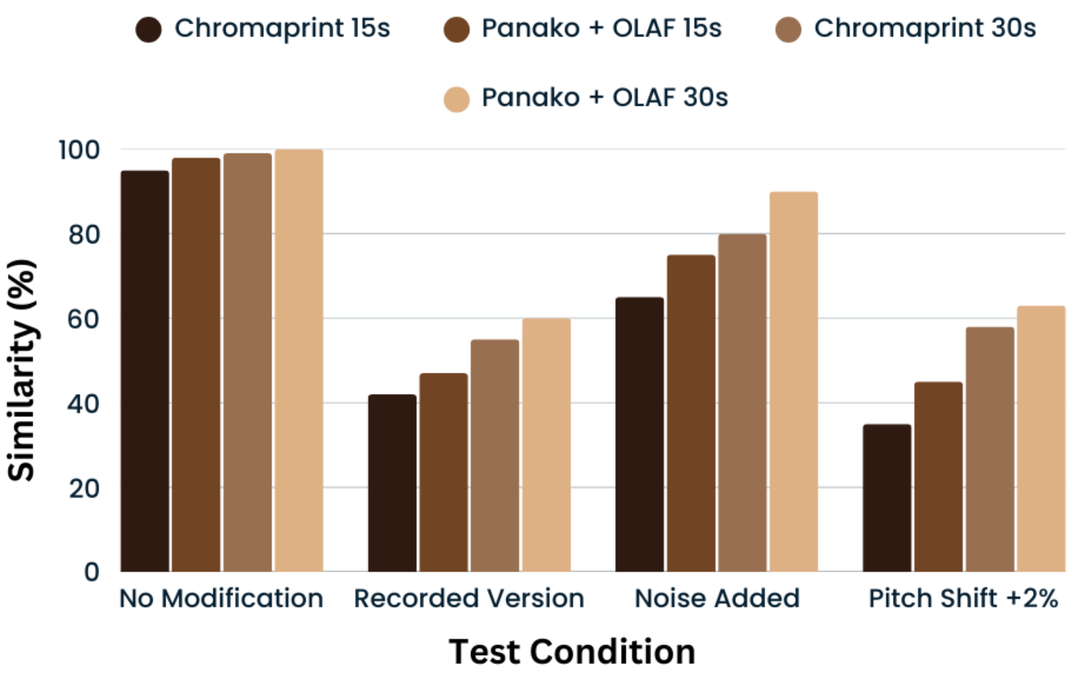

Fig 6.1. Audio Similarity Comparison Across Conditions

positives. But for longer audio clips, which still had more recognizable features even after pitch shifting, the blockchain rejected the upload to uphold copyright restrictions.

This subtle behavior highlights the system's capacity for a practical compromise between intellectual property protection and creative reuse. Improved pitch detection in Panako 2.0 + OLAF, combined with the length-based decision logic of blockchain, provides for effective enforcement of copyrights while being permissive of smaller, legitimate changes—distinguishing it from the more strict framework of Chromaprint.

6.3 GAS USAGE COMPARISON

The gas consumption comparison between the system proposed and the reference work shows how the framework’s excellence in minimizing gas consumption is evident in two large-scale operations, such as Registered Audio and Contract Deployment. The results contained in Table 6.2 show note-worthy reductions in gas consumption, resulting in the system being economically scalable in real-world applications.

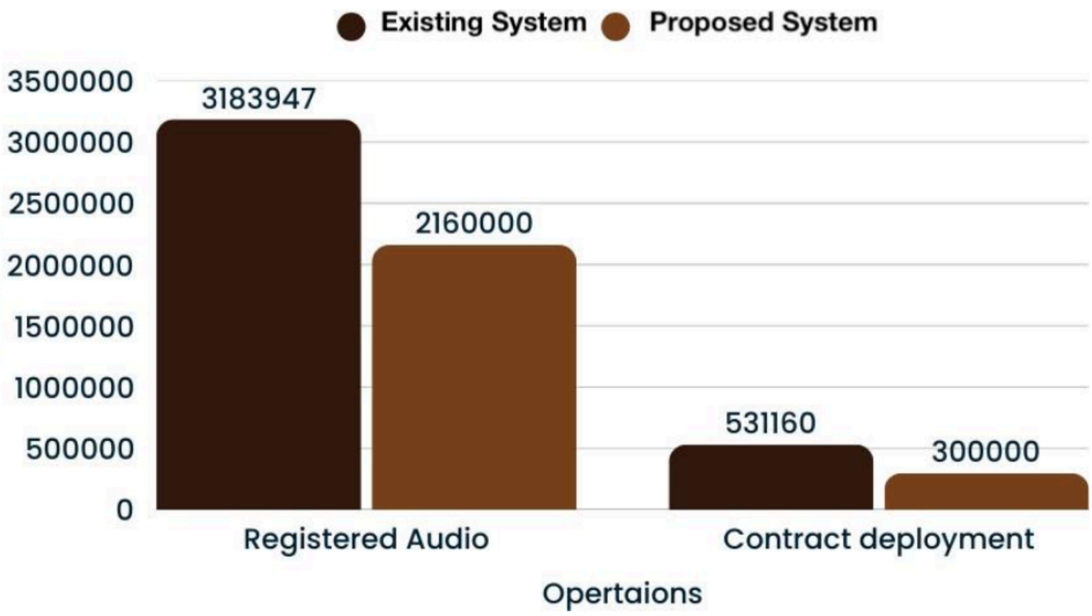

Fig 6.2. Gas Usage Comparison

1. **Registered Audio:**

   By recording sound content on the blockchain, the reference system spends 3,183,947 units of gas, which is an extremely high gas spending rate. The new system, however, spends only 2,160,000 units of gas, reducing gas spending by a staggering 32.2%. This record-breaking efficiency enhancement makes sound registration more affordable and cost-effective. The proposed architecture maximizes audio registration efficiency by various means: Perceptual Hashing Functions generate short, tamper-proof fingerprints that minimize data size on-chain; Decentralized Storage (IPFS) keeps file pointers and audio metadata off-chain, keeping data locked in the blockchain to a minimum and thereby lowering gas costs; and Optimized Solidity Functions within the smart contract minimize computational requirements and gas usage. By reducing the cost of audio registration, the system becomes economically viable for large-scale deployment, wherein content registrations are performed on a daily scale.

2. Contract Deployment:

The baseline system has 531,160 units of gas consumption in deployment, whereas the proposed system has only 300,000 units of gas usage, hence conserving gas consumption by an astounding 43.5%. The gas-intensive character of deployment is dealt with by reducing storage activities through a low-storage contract that utilizes low-storage looping with a reduced number of data structure variables, hence keeping the space that the blockchain will take up after deployment to a minimum. Furthermore, the contract also maximizes computation overhead through effective looping and reduction of unnecessary state transitions. Smaller contract size due to optimized contract code saves bytecode and gas units when deploying. All these optimizations allow cost-effective systems that can deploy contracts at high rates, a need especially for mass-scale copyright protection systems.

3. Efficiency and Scalability Impact:

Lowered gas usage renders the system financially sound on a large scale, where small businesses, solo artists, and creatives can register and base their material on the blockchain without inordinate expenses. The cost savings are especially precious when there are batched group transactions, for instance, batch renewal of licenses or batch recordings. Additionally, reduced gas usage minimizes the burden on the Ethereum network, enabling faster transactions and lower congestion levels, which proves beneficial during times of peak network utilization, thereby minimizing transaction time and cost. Further, reduced gas usage enables environmental sustainability by lowering blockchain resource consumption. It also improves scalability such that the system can process more operations without having to increase resource demands, so that the system is both cost-effective and efficient irrespective of the number of transactions. Mass deployment is made feasible through gas-optimized architecture, and this makes Layer 2 scaling solutions like Arbitrum and Polygon feasible.

4. Real-World Implications:

Low gas costs make copyright protection in blockchain economically feasible and affordable to single authors and small groups, supporting simultaneous registrations of several contents without undermining the economic viability of the system. Further, reduced gas consumption makes the system appropriate for real-time settlements of royalties because of reduced transaction costs, facilitating timely and equitable remuneration of content owners at low overhead. The streamlined platform makes sure blockchain copyright protection is economically viable through minimized transaction costs, to the advantage of music platforms, record labels, and copyright managing bodies involved in bulk transactions.

Operation	Existing Model Gas Usage	Proposed System Gas Usage	Gas Reduction (%)
Registered Audio	31,83,947	2,160,000	32.2%
Contract Deployment	531,160	300,000	43.5%

Table 6.2. Gas usage Comparison between Existing System and Proposed system

6.4 SCALABILITY AND EFFICIENCY ISSUES

Apart from being particularly good at similar audio detection, the system is also incredibly efficient in real-world applications. Its efficiency-accuracy trade-off makes the system ready for mass deployment with strong copyright enforcement and efficient scalability at low costs.

1. **Conservative Use of Blockchain:**

   The system saves blockchain resources through the utilization of off-chain storage and possesses multiple advantages. Low storage costs are realized through storing audio fingerprints and metadata in IPFS rather than full audio files on-chain, with the blockchain only storing the content hash, which reduces transaction fees and storage costs significantly. Moreover, gas-optimized contracts reduce gas usage during contract deployment and audio enrollment, resulting in huge savings, particularly in dense transaction scenarios involving hundreds of thousands of transactions. The system also enables faster transactions and less congestion by reducing computational latency through fewer on-chain transactions since most data is stored off-chain in IPFS, leaving only necessary references on-chain. In addition, optimized Solidity code facilitates quicker execution of smart contract operations with low computational overhead.

2. **Real-World Practical Application:**

User testing was carried out with the members of the music community, independent artists as well as content creators, and a number of key findings were established. Users indicated high user satisfaction, with the users saying that the system was effective in detecting audio abuse and safeguarding intellectual property rights. The users also found the system trustworthy, enjoying the fraud resistance and transparency provided by the blockchain; the tamper-proofing provided by blockchain records provided secure and verifiable copyright protection. Furthermore, user-friendliness of the system was also enhanced through easy integration with MetaMask for Ethereum payments and IPFS Desktop for decentralized storage.

3. **Economic Feasibility:**

Gas optimization significantly reduces the costs of transactions, making the system economically viable for widespread usage. The key advantages include reduced operating costs, since reduced gas prices make the system economically viable for small businesses and independent artists. The scalable nature has a modular structure that is compatible with other blockchain networks and third-party software. In addition, cost-effectiveness for mass deployments is enhanced by reduced gas consumption, allowing the system to conduct more operations without any additional cost, hence being ideal for bulk-scale usage.

4. **Enhanced Data Security and Integrity:**

The system leverages blockchain immutability and decentralized storage in an effort to achieve security across multiple dimensions. First, audio fingerprints uploaded onto the blockchain through immutable means are used to capture unalterable proof of ownership. Second, IPFS's content-addressed storage maintains redundancy of data so as to ensure data integrity and make data retrievable in the case of failure of the network. Thirdly, the system prevents double registrations by cross-verifying registered fingerprints to avoid duplicate registration and malicious duplication.

5. **Real-World Implications:**

The system provides working solutions to music professionals and content creators in all critical areas. First, it safeguards copyright through blockchain-secured, tamper-evident technologies that secure intellectual property rights for artists. Second, transparent ownership ledgers enhance fair compensation and licensing of digital content. In addition, the system uses anti-piracy methods through detection of unauthorized manipulation, such as noise injection and re-recording, to prevent digital piracy.

6.5 USER INTERFACE

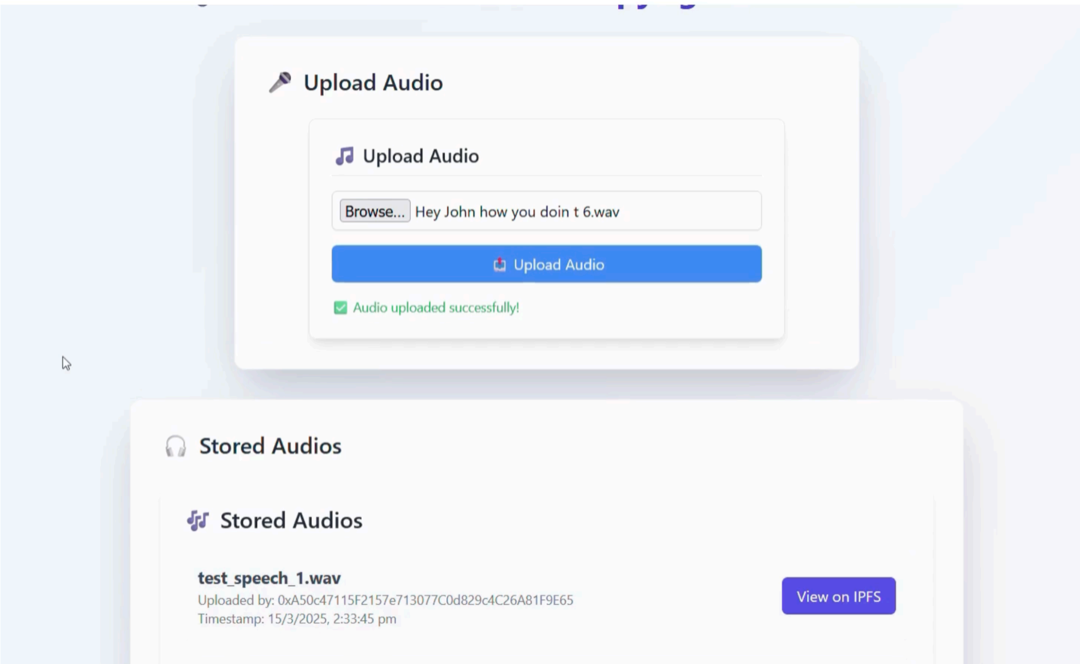

Fig 6.3. Original Audio Uploaded

Figure 6.3 shows the user interface of a distributed audio copyright protection system where an audio file was successfully uploaded. The system provides the user with audio files, select the audio files and upload them to a distributed storage solution.

Uploading a successful success message ("Audio uploaded successfully!") will save the file safely. Ownership metrics are assigned to all files in the form of items in Ethereum items. It also provides a timeline to keep transactions in a state that is unchanged. The saved audio can be provided by IPF (interplanetary file system) users to provide distributed storage, immutability and availability

Figure 6.4 illustrates how the system prevents duplicate uploads, a critical feature of copyright protection. While attempting to upload an already existing audio file in the system, the system presents an error message: "Error: Audio already exists or upload failed."

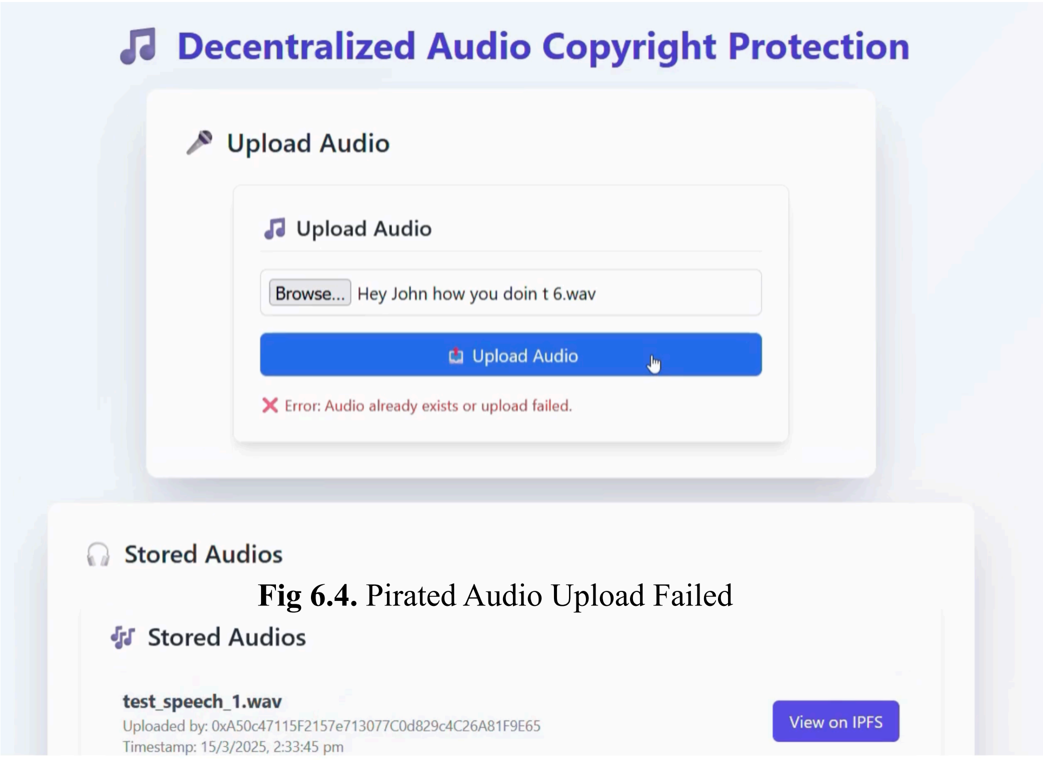

Fig 6.4. Pirated Audio Upload Failed

This error suggests that the system utilizes audio fingerprinting techniques in comparing the freshly uploaded file against pre-stored ones to prevent the duplication of copyrighted material. Such an aspect brings security, copyright protection, and storage efficiency in the decentralized system.

The Stored Audios feature is static and displays the data that has been previously uploaded and with the intention of being transparent. The View on IPFS button ensures the verification of the stored audio, boosting the application of blockchain and decentralized technologies in the protection of copyrights and intellectual property.

6.6 SUMMARY

This chapter describes the results and discussion of the proposed system. The conclusion and Future enhancement are discussed in the next chapter.

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The proposed framework is for the music industry and beyond. It represents a comprehensive and innovative solution for secure and equitable audio copyright protection, seamlessly integrating advanced perceptual hashing techniques, blockchain technology, and decentralized storage to address the longstanding challenges faced by creators. This framework is the ability to generate unique digital fingerprints for audio files that are highly resilient to modifications. This enables the system to accurately detect unauthorized use or distribution of copyrighted content, putting the power back into the hands of artists. The blockchain-powered ownership verification and automated licensing and royalty distribution mechanisms further empower creators, fostering a more transparent and equitable digital music ecosystem.The decentralized storage approach using IPFS enhances the security and resilience of the system, ensuring that the audio files remain secure and accessible, even in the face of potential attacks or system failures. The user-friendly interface, complete with MetaMask integration, ensures that creators of all backgrounds can easily navigate and utilize the platform, making it accessible to a wide range of users. But the impact of this framework extends far beyond the music industry. Its principles and techniques can be applied to protect various forms of digital content, manage intellectual property rights in diverse creative fields, verify the ownership of physical assets, and even develop decentralized marketplaces for digital assets, empowering creators and ensuring fair transactions. The researchers behind this framework are not resting on their laurels. They are continuously exploring ways to enhance the system, focusing on optimizing perceptual hashing, improving IPFS management, incorporating advanced analytics, developing a mobile app, and fostering partnerships with music platforms. Their dedication to pushing the boundaries of innovation is a testament to their commitment to transforming the creative landscape and empowering artists to thrive in the digital age. This framework is more than just a technological solution; it's a vision for a future where creators are celebrated, their rights are protected, and the music industry operates with transparency and fairness. As the online world continues to evolve, this innovative framework stands as a beacon of hope, guiding the way towards a more equitable and artist-centric creative economy.

7.2 FUTURE ENHANCEMENT

- **Optimizing Perceptual Hashing:**

The core of this system's success lies in its robust perceptual hashing techniques. The researchers recognize that there's always room for improvement when it comes to speed, accuracy, and efficiency. By exploring advanced algorithms and computational models, they can push the boundaries of what's possible. Leveraging parallel and distributed computing, as well as hardware acceleration, could be game-changers, allowing the system to process audio files even faster without sacrificing precision.

- **Enhancing IPFS Management:**

Storing audio files on the decentralized IPFS network is a brilliant move, but the team acknowledges there are still challenges to overcome. The higher complexity and retrieval times associated with IPFS can be addressed through intelligent data management solutions. Developing smart caching mechanisms and sophisticated indexing methods can significantly improve content accessibility. Additionally, experimenting with hybrid storage approaches that blend IPFS with off-chain databases could strike the perfect balance between decentralization and performance.

- **Unlocking Advanced Analytics:**

The researchers envision empowering artists with deeper insights into the usage and licensing of their music. By incorporating machine learning-driven analytics, the system can provide creators with invaluable data and trends. Predictive modeling techniques could even help identify potential copyright threats before they materialize, allowing artists to stay one step ahead of any misuse of their work.

- **Mobilizing Copyright Management:**

Recognizing the need for on-the-go access, the team plans to develop a user-friendly mobile app. This will enable artists to manage their copyrights from anywhere, with features like biometric authentication for enhanced security and real-time tracking capabilities to monitor the status of their content.

- **Integrating with Music Platforms:**

To truly make a lasting impact, the researchers understand the importance of forging partnerships with established music-sharing platforms. By seamlessly integrating the copyright protection mechanisms into these existing ecosystems, the system can become an integral part of the digital music landscape, benefiting both artists and fans.

- **Continuous Improvement through User Feedback:**

The team is committed to an iterative approach, continuously gathering user feedback and insights. Through usability studies, A/B testing, and focus group discussions, they can refine the system's interface and functionality, ensuring it remains intuitive and user-friendly. The addition of AI-powered virtual assistants or chatbots could further enhance the user experience, providing artists with on-demand guidance and support.

7.3 SOCIAL IMPACT

- **Empowering Artists:**

This system gives artists a powerful tool to take control of their intellectual property. Instead of feeling helpless against unauthorized use of their work, they can now protect their creations with a secure and transparent platform. This empowers them to focus on their craft, knowing their rights are safeguarded.

- **Promoting Fairness and Equity:**

The automated royalty distribution system is a true innovation. It cuts out the middlemen and ensures artists are paid fairly and promptly for their work. This fosters a more sustainable and equitable digital music ecosystem, where the creators themselves reap the rewards of their talent and hard work.

- **Combating Piracy:**

Piracy has been a thorn in the side of the music industry for far too long. But this framework's robust copyright protection mechanisms, including the ability to detect even slightly modified audio files, can help curb unauthorized use. Artists can now thrive in the digital age without constantly worrying about their work being stolen or misused.

- **Fostering Innovation:**

By taking the burden of copyright management off their shoulders, this system allows artists to focus on their true passion creating music. This freedom to innovate, without the distractions of legal battles and royalty disputes, can lead to a flourishing of new ideas and artistic expression in the industry.

7.4 APPLICABILITY

- **Protecting Digital Content:**

The principles behind this system can be applied to safeguard all kinds of digital content - from images and videos to written works. Creators in various industries can benefit from the secure and tamper-proof ownership verification.

- **Managing Intellectual Property:**

The blockchain-based ownership tracking and automated licensing capabilities can be adapted to manage intellectual property rights in diverse creative fields, from art and literature to software development.

- **Tracking Decentralized Assets:**

The integration of perceptual hashing and blockchain can be leveraged to verify the ownership of physical assets, like collectibles or luxury goods, in a decentralized and transparent manner.

- **Aiding Multimedia Forensics:**

The advanced audio fingerprinting and similarity detection techniques can be invaluable in multimedia forensics, helping to identify and verify digital evidence.

- **Inspiring Decentralized Marketplaces:**

The framework's architecture can serve as a blueprint for developing decentralized marketplaces for various digital assets, empowering creators and ensuring fair transactions.

‘

APPENDIX 1

SYSTEM R EQUIREMENTS

HARDWARE REQUIREMENT

Processor	:	Apple M2 (8-core CPU)
GPU	:	10-core GPU
Hard disk	:	128GB
RAM	:	8 GB

SOFTWARE REQUIREMENT

Software used	**:**	Panako 2.0**,** Flask, ReactJS, IPFS Desktop, Ethereum(Ganache), Truffle, MetaMask
Operating System	**:**	macOS Sequoia 15.2

APPENDIX 2

CODE

#Solidity Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AudioRegistry {
    struct Audio {
        string filename;
        string ipfsHash;
        uint256 timestamp;
        address owner;
    }

    mapping(string => Audio) private audioFiles; // Keep mapping for quick lookup
    string[] private audioHashes; // Store IPFS hashes to track total count

    event AudioRegistered(string ipfsHash, string filename, uint256 timestamp, address
owner);

    function registerAudio(string memory _filename, string memory _ipfsHash) public {
        require(bytes(audioFiles[_ipfsHash].ipfsHash).length == 0, "Audio already exists!");

        audioFiles[_ipfsHash] = Audio({
            filename: _filename,
            ipfsHash: _ipfsHash,
            timestamp: block.timestamp,
            owner: msg.sender
        });
```

```
        audioHashes.push(_ipfsHash); // Store hash to keep track of total audios

        emit AudioRegistered(_ipfsHash, _filename, block.timestamp, msg.sender);
    }

    function getAudio(string memory _ipfsHash) public view returns (string memory,
uint256, address) {
        require(bytes(audioFiles[_ipfsHash].ipfsHash).length > 0, "Audio not found!");
        Audio memory audio = audioFiles[_ipfsHash];
        return (audio.filename, audio.timestamp, audio.owner);
    }

    function getTotalAudios() public view returns (uint256) {
        return audioHashes.length; // Return total count
    }

    function getAudioByIndex(uint256 index) public view returns (string memory, string
memory, uint256, address) {
        require(index < audioHashes.length, "Invalid index!");
        string memory ipfsHash = audioHashes[index];
        Audio memory audio = audioFiles[ipfsHash];
        return (audio.filename, audio.ipfsHash, audio.timestamp, audio.owner);
    }
}
```

#Server Code

```
from flask import Flask, request, jsonify
import os
import librosa
import numpy as np
```

```
import sqlite3
from werkzeug.utils import secure_filename
from flask_cors import CORS
import json
from scipy.spatial.distance import cosine, euclidean
from web3 import Web3
import subprocess
import uuid

app = Flask(__name__)
CORS(app, supports_credentials=True)

UPLOAD_FOLDER = "uploaded_audio"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)

# Initialize Web3
w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:8545"))
if not w3.is_connected():
    raise ConnectionError("Web3 connection failed. Ensure Ganache is running.")

# Load Contract ABI and Address
CONTRACT_PATH = "frontend/src/contracts/AudioRegistry.json"
try:
    with open(CONTRACT_PATH) as f:
        contract_data = json.load(f)
    contract_address = "0xA897650E58597a338F423D1753b6082D3c25415D"
    contract_abi = contract_data["abi"]
    instance = w3.eth.contract(address=contract_address, abi=contract_abi)
    w3.eth.default_account = w3.eth.accounts[0]
except Exception as e:
    raise FileNotFoundError(f" Error loading contract: {e}")
```

```
# Database setup
DB_PATH = "audio_database.db"
with sqlite3.connect(DB_PATH) as conn:
    cursor = conn.cursor()
    cursor.execute("""
    CREATE TABLE IF NOT EXISTS audio_metadata (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        filename TEXT UNIQUE,
        mfcc TEXT,
        ipfs_hash TEXT,
        blockchain_tx TEXT,
        timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
    )
    """)
    conn.commit()

def get_mfcc(audio_path):
    try:
        y, sr = librosa.load(audio_path, sr=22050)
        mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=40)


            mfcc = (mfcc - np.mean(mfcc, axis=1, keepdims=True)) / np.std(mfcc, axis=1,
keepdims=True)


        mfcc_flattened = mfcc.flatten()
        return json.dumps(mfcc_flattened.tolist())
    except Exception as e:
        print(f"Error processing MFCC: {e}")
        return None
```

```
def compute_similarity(vec1, vec2):
    try:
        vec1 = np.array(vec1).flatten()
        vec2 = np.array(vec2).flatten()

        cosine_sim = 1 - cosine(vec1, vec2)
        euclidean_dist = euclidean(vec1, vec2)
        return (cosine_sim * 0.6) + ((1 / (1 + euclidean_dist)) * 0.4)
    except Exception as e:
        print(f"Error in similarity calculation: {e}")
        return 0

def check_audio_similarity(audio_path):
    new_audio_mfcc = get_mfcc(audio_path)
    if not new_audio_mfcc:
        return 0

    with sqlite3.connect(DB_PATH) as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT filename, mfcc FROM audio_metadata")
        existing_audios = cursor.fetchall()

    for filename, mfcc_str in existing_audios:
        try:
            stored_mfcc = np.array(json.loads(mfcc_str)).flatten()
            new_audio_vec = np.array(json.loads(new_audio_mfcc)).flatten()

            similarity = compute_similarity(new_audio_vec, stored_mfcc)
            if similarity >= 0.50:
                return similarity * 100
        except Exception as e:
            print(f"Error processing stored MFCC for {filename}: {e}")
```

```
    return 0

def upload_to_ipfs(file_path):
    """Uploads a file to IPFS using the IPFS CLI."""
    try:
        result = subprocess.run(
            ["ipfs", "add", "-Q", file_path], capture_output=True, text=True
        )
        if result.returncode == 0:
            ipfs_hash = result.stdout.strip()
            print(f" IPFS Upload Successful! Hash: {ipfs_hash}")
            return ipfs_hash
        else:
            print(f" IPFS Upload Failed: {result.stderr}")
            return None
    except Exception as e:
        print(f" Error running IPFS CLI: {e}")
        return None

def store_audio_metadata(filename, file_path, ipfs_hash, tx_hash):
    mfcc_vector = get_mfcc(file_path)
    if mfcc_vector is None:
        return

    with sqlite3.connect(DB_PATH) as conn:
        cursor = conn.cursor()
        cursor.execute("DELETE FROM audio_metadata WHERE filename = ?",
(filename,))
        cursor.execute("""
            INSERT INTO audio_metadata (filename, mfcc, ipfs_hash, blockchain_tx)
            VALUES (?, ?, ?, ?)
        """, (filename, mfcc_vector, ipfs_hash, tx_hash))
```

```
        conn.commit()

@app.route("/upload-audio", methods=["POST"])
def upload_audio():
    if "file" not in request.files:
        return jsonify({"error": "No file uploaded"}), 400

    file = request.files["file"]
    filename = secure_filename(file.filename)
    unique_filename = f"{uuid.uuid4().hex}_{filename}"
    file_path = os.path.join(UPLOAD_FOLDER, unique_filename)
    file.save(file_path)

    print(f" File saved locally: {file_path}")
    similarity = check_audio_similarity(file_path)
    if similarity >= 75:
        os.remove(file_path)
        print("Audio too similar, not uploading.")
        return jsonify({"message": "Audio too similar.", "similarity": similarity}), 400

    ipfs_hash = upload_to_ipfs(file_path)
    if not ipfs_hash:
        return jsonify({"error": "IPFS upload failed"}), 500

    print(f" IPFS Upload Successful! Hash: {ipfs_hash}")

    try:
            tx_hash = instance.functions.registerAudio(filename, ipfs_hash).transact({"from":
w3.eth.accounts[0]})
        tx_hash_hex = tx_hash.hex()
        print(f"Blockchain TX Hash: {tx_hash_hex}")
    except Exception as e:
```

```
        print(f"Blockchain registration failed: {e}")
        return jsonify({"error": "Blockchain registration failed", "details": str(e)}), 500

    store_audio_metadata(unique_filename, file_path, ipfs_hash, tx_hash_hex)

    return jsonify({"message": "Audio stored!", "ipfs_hash": ipfs_hash, "tx_hash": tx_hash_hex}), 200

@app.route("/api/audios", methods=["GET"])
def get_audios():
    with sqlite3.connect(DB_PATH) as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT filename, ipfs_hash, blockchain_tx, timestamp FROM audio_metadata")
        audios = [
            {"filename": row[0], "ipfs_hash": row[1], "blockchain_tx": row[2], "timestamp": row[3]}
            for row in cursor.fetchall()
        ]

    return jsonify(audios), 200

if __name__ == "__main__":
    app.run(debug=True, port=5000)
```

#AudioList.js

```
import React, { useEffect, useState } from "react";
import Web3 from "web3";
import AudioRegistryABI from "../contracts/AudioRegistry.json";
const contractAddress = "0xA897650E58597a338F423D1753b6082D3c25415D";

const AudioList = () => {
```

```
const [audios, setAudios] = useState([]);
const [loading, setLoading] = useState(true);
const [error, setError] = useState(null);
const [account, setAccount] = useState(null);
const [web3, setWeb3] = useState(null);
const [audioRegistry, setAudioRegistry] = useState(null);

useEffect(() => {
  const initBlockchain = async () => {
    try {
      if (!window.ethereum) {
        throw new Error("MetaMask is not installed. Please install it to use this feature.");
      }

      const web3Instance = new Web3(window.ethereum);
      setWeb3(web3Instance);

      const accounts = await window.ethereum.request({ method: "eth_requestAccounts" });
      setAccount(accounts[0]);

      const audioRegistryInstance = new web3Instance.eth.Contract(AudioRegistryABI, contractAddress);
      setAudioRegistry(audioRegistryInstance);

      console.log(" Blockchain initialized. Account:", accounts[0]);
      fetchAudios(audioRegistryInstance);
    } catch (error) {
      console.error(" Error initializing blockchain:", error);
      setError(error.message || "Failed to initialize blockchain.");
      setLoading(false);
```

```
  }
};

const fetchAudios = async (contract) => {
  if (!contract) {
    console.error(" Contract instance is not available.");
    setError("Smart contract not loaded. Try refreshing the page.");
    setLoading(false);
    return;
  }

  setLoading(true);
  setError(null);

  try {
    console.log(" Fetching total audios...");
    const totalAudios = await contract.methods.getTotalAudios().call();
    console.log(`Total audios stored: ${totalAudios}`);

    if (totalAudios === "0") {
      console.warn(" No audios found in the contract.");
      setAudios([]);
      setLoading(false);
      return;
    }

    const fetchedAudios = [];

    for (let i = 0; i < totalAudios; i++) {
      try {
        console.log(`Fetching audio at index ${i}...`);
        const audio = await contract.methods.getAudioByIndex(i).call();
```

```
                console.log(`Audio found:`, audio);

                fetchedAudios.push({
                  filename: audio[0],
                  ipfsHash: audio[1],
                  timestamp: new Date(Number(audio[2]) * 1000).toLocaleString(),
                  owner: audio[3],
                });
              } catch (error) {
                console.warn(`Error fetching audio at index ${i}:`, error);
              }
            }

            setAudios(fetchedAudios);
          } catch (error) {
            console.error(" Error fetching audios:", error);
            setError("Failed to fetch audios. Please check your network and try again.");
          } finally {
            setLoading(false);
          }
        };

        initBlockchain();
      }, []);

      return (
        <div className="bg-white p-6 rounded-lg shadow-lg">
           <h2 className="text-2xl font-semibold text-gray-800 mb-4">🎶 Stored Audios</
h2>
          {loading ? (
            <div className="flex justify-center items-center">
```

```
<div className="animate-spin rounded-full h-8 w-8 border-b-2 border-
gray-900"></div>
    <span className="ml-3">Loading audios...</span>
  </div>
) : error ? (
  <div className="text-red-500 text-center">
    <p>{error}</p>
  </div>
) : audios.length > 0 ? (
  <ul className="space-y-4">
    {audios.map((audio, index) => (
      <li key={index} className="p-4 bg-gray-50 rounded-lg hover:bg-gray-100
transition-colors">
        <div className="flex justify-between items-center">
          <div>
            <strong className="text-lg text-gray-700">{audio.filename}</
strong>
            <p className="text-sm text-gray-500">Uploaded by:
{audio.owner}</p>
            <p className="text-sm text-gray-500">Timestamp:
{audio.timestamp}</p>
          </div>
          <a
            href={`http://127.0.0.1:8080/ipfs/${audio.ipfsHash}`}
            target="_blank"
            rel="noopener noreferrer"
            className="bg-indigo-600 text-white px-4 py-2 rounded-md
hover:bg-indigo-700 transition-colors"
          >
            View on IPFS
          </a>
        </div>
```

```
                </li>
              ))}
            </ul>
          ) : (
            <p className="text-gray-500 text-center">No stored audios found.</p>
          )}
        </div>
    );
};

export default AudioList;
```

#AudioRegistry.js

```
import web3 from "./web3";
import AudioRegistry from "./contracts/AudioRegistry.json"; // ABI from Truffle build

const contractAddress = "0xA897650E58597a338F423D1753b6082D3c25415D";
const instance = new web3.eth.Contract(AudioRegistry.abi, contractAddress);

export default instance;
```

#RegisterAudio.js

```
import React, { useState } from "react";
import web3 from "../web3";
import AudioRegistry from "../AudioRegistry";

const RegisterAudio = ({ ipfsHash, filename }) => {
    const [message, setMessage] = useState("");
    const [loading, setLoading] = useState(false);

    const registerOnBlockchain = async () => {
        try {
```

```
      setLoading(true);
      setMessage("");

      const accounts = await web3.eth.getAccounts();
      if (!accounts || accounts.length === 0) {
        throw new Error("No Ethereum accounts found. Ensure MetaMask is connected.");
      }

      await AudioRegistry.methods.registerAudio(filename, ipfsHash).send({ from: accounts[0] });

      setMessage("Audio registered on blockchain!");
    } catch (error) {
      console.error("Blockchain registration failed", error);
      setMessage("Error: Could not register on blockchain.");
    } finally {
      setLoading(false);
    }
  };
  return (
    <div className="max-w-lg mx-auto p-6 bg-white shadow-lg rounded-lg mt-6 border border-gray-200">
      <h2 className="text-xl font-semibold text-gray-800 mb-4 border-b pb-2">🔗 Register on Blockchain</h2>

      <p className="text-gray-600 mb-2"><span className="font-semibold">Filename:</span> {filename}</p>
      <p className="text-gray-500 text-sm mb-4 truncate"><span className="font-semibold">IPFS Hash:</span> {ipfsHash}</p>
      <button
```

```
            onClick={registerOnBlockchain}
            disabled={loading}
            className={`w-full px-4 py-2 text-white rounded-md transition ${
                    loading ? "bg-gray-400 cursor-not-allowed" : "bg-green-500 hover:bg-green-600"
            }`}
        >
            {loading ? " Registering..." : "Register on Blockchain"}
        </button>
        {message && (
            <p className={`mt-4 text-sm font-medium ${message.startsWith("Ok") ? "text-green-600" : "text-red-600"}`}>
              {message}
            </p>
        )}
      </div>
  );
};

export default RegisterAudio;
```

#UploadAudio.js

```
import React, { useState } from "react";
import axios from "axios";

const UploadAudio = ({ onUpload }) => {
  const [file, setFile] = useState(null);
  const [message, setMessage] = useState("");
  const [loading, setLoading] = useState(false);

  const handleFileChange = (event) => {
    setFile(event.target.files[0]);
```

```
    setMessage("");
  };

  const uploadToBackend = async () => {
    if (!file) {
      setMessage(" Please select an audio file first!");
      return;
    }

    const formData = new FormData();
    formData.append("file", file);

    try {
      setLoading(true);
      setMessage("");

        const response = await axios.post("http://127.0.0.1:5000/upload-audio", formData,
{
          headers: {
            "Content-Type": "multipart/form-data"
          },
          withCredentials: true
        });

      setMessage("Audio uploaded successfully!");
      onUpload && onUpload();
    } catch (error) {
      console.error("Upload failed", error);
      setMessage(" Error: Audio already exists or upload failed.");
    } finally {
      setLoading(false);
    }
```

```
  };

  return (
    <div className="max-w-lg mx-auto p-6 bg-white shadow-lg rounded-lg mt-6 border border-gray-200">
      <h2 className="text-xl font-semibold text-gray-800 mb-4 border-b pb-2">Upload Audio</h2>

      <input
        type="file"
        onChange={handleFileChange}
        accept="audio/*"
          className="block w-full p-2 border border-gray-300 rounded-md shadow-sm focus:outline-none focus:ring focus:ring-blue-300 mb-4"
      />

      <button
        onClick={uploadToBackend}
        disabled={loading}
        className={`w-full px-4 py-2 text-white rounded-md transition ${
                  loading ? "bg-gray-400 cursor-not-allowed" : "bg-blue-500 hover:bg-blue-600"
        }`}
      >
        {loading ? "Uploading..." : "Upload Audio"}
      </button>

      {message && (
        <p className={`mt-4 text-sm font-medium ${message.startsWith("ok") ? "text-green-600" : "text-red-600"}`}>
          {message}
        </p>
```

```
        )}
      </div>
    );
  };

export default UploadAudio;
```

#Web3.js

```
import Web3 from "web3";

const web3 = new Web3(Web3.givenProvider || "http://127.0.0.1:8545"); // Use Ganache RPC

export default web3;
```