VIETNAM-KOREA UNIVERSITY OF INFORMATION AND  
COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**



GRADUATION THESIS

**DECENTRALIZED MUSIC PLAYER APP DEVELOPMENT**

Student: **NGUYEN DUY SY**

**NGUYEN THI NGUYET MINH**

Class: **18IT5 + 18IT2**

Supervisor: Dr. TRAN VAN DAI

***Da Nang, December – 2022***

VIETNAM-KOREA UNIVERSITY OF INFORMATION AND

COMMUNICATION TECHNOLOGY

**Faculty of Computer Science**



GRADUATION THESIS

**DECENTRALIZED MUSIC PLAYER APP DEVELOPMENT**

Student: **NGUYEN DUY SY**

**NGUYEN THI NGUYET MINH**

Class: **18IT5 + 18IT2**

Supervisor: Dr. TRAN VAN DAI

***Da Nang, December - 2022***

**SUPERVIOR’S COMMENT***<according to the template>*

ACKNOWLEDGEMENTS

We would like to profoundly acknowledge the people who have helped me during my studies:

First and foremost, we would like to thank my research supervisors, Dr Tran Van Dai. Without his assistance and dedicated involvement in every step throughout the process, this paper could have never been accomplished. We are extremely grateful to you for supporting and understanding me during the past time.

We would like to acknowledge all lecturers at Vietnam Korea University of Information And Communication Technology, for their valuable guidance throughout my studies. You provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.

Finally, but not least, my warm and heartfelt thanks go to my family for their tremendous support and hope they had given to me. Without that hope, this thesis would not have been possible. Thank you all for the strength you gave me.

STATEMENT OF AUTHORSHIP

*Danang, December 5th, 2022*

NGUYEN THI NGUYET MINH

TABLE OF CONTENTS

[ACKNOWLEDGEMENTS i](#_Toc122275904)

[STATEMENT OF AUTHORSHIP ii](#_Toc122275905)

[TABLE OF CONTENTS iii](#_Toc122275906)

[ABBREVIATIONS v](#_Toc122275907)

[LIST OF FIGURES vi](#_Toc122275908)

[LIST OF TABLES vii](#_Toc122275909)

[INTRODUCTION 1](#_Toc122275910)

[1. Problem statement 1](#_Toc122275911)

[2. Aims and Objectives 1](#_Toc122275912)

[3. Structure of the thesis 2](#_Toc122275913)

[CHAPTER 1. DECENTRALIZED MUSIC STREAMING 3](#_Toc122275914)

[1.1 Decentralized music streaming 3](#_Toc122275915)

[1.1.1 Remaining problems on centralized music platforms 3](#_Toc122275916)

[1.1.2 Blockchain – the inevitable next stage of music? 4](#_Toc122275917)

[1.1.3 Difficulty when music blockchain application 5](#_Toc122275918)

[CHAPTER 2. THEORETICAL INFRASTRUCTURE 7](#_Toc122275919)

[2.1 Blockchain 7](#_Toc122275920)

[2.1.1 List of top Blockchain features 8](#_Toc122275921)

[2.1.2 How Does a Blockchain Work? 11](#_Toc122275922)

[2.1.3 Blockchain Decentralization 12](#_Toc122275923)

[2.1.4 Pros and Cons of Blockchain 13](#_Toc122275924)

[2.2 Overview of Next.js 17](#_Toc122275925)

[2.2.1 What Can You Build with Next.Js and When to Use Next.Js? 17](#_Toc122275926)

[2.2.2 Next.js Features 18](#_Toc122275927)

[2.2.3 Pros and Cons of Next.js 19](#_Toc122275928)

[2.3 InterPlanetary File System IPFS 20](#_Toc122275929)

[2.3.1 How does IPFS work? 21](#_Toc122275930)

[2.3.2 IPFS features 22](#_Toc122275931)

[2.3.3 Pros and Cons of IPFS 23](#_Toc122275932)

[2.3.4 Potential applications of IPFS in Blockchain 24](#_Toc122275933)

[2.4 What is Django? 24](#_Toc122275934)

[2.4.1 Learn about Django's MVT Pattern 24](#_Toc122275935)

[2.4.2 Why use Django 24](#_Toc122275936)

[2.4.3 Introduction to Django REST framework 25](#_Toc122275937)

[2.5 Alchemy in blockchain 26](#_Toc122275938)

[2.5.1 What is Alchemy? 26](#_Toc122275939)

[2.5.2 Supported networks 26](#_Toc122275940)

[2.5.3 Alchemy interface 26](#_Toc122275941)

[2.5.4 APIs, documentation, and DX 27](#_Toc122275942)

[2.5.5 Pricing 27](#_Toc122275943)

[CHAPTER 3. *SYSTEM ANALYSIS AND DESIGN* 28](#_Toc122275944)

[3.1 System analysis and design 28](#_Toc122275945)

[3.1.1 System requirements 28](#_Toc122275946)

[3.1.2 Use case diagram 30](#_Toc122275947)

[3.2 System deployment 33](#_Toc122275948)

[3.2.1 Database Design 33](#_Toc122275949)

[3.2.2 System interface 34](#_Toc122275950)

[3.3 Summary 36](#_Toc122275951)

[CONCLUSIONS AND SUGGESTIONS 37](#_Toc122275952)

[1. Conclusions 37](#_Toc122275953)

[2. Suggestions 38](#_Toc122275954)

[REFERENCES 39](#_Toc122275955)

ABBREVIATIONS

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **MEANING** |
| IPFS | Interplanetary File System |
| REST | REpresentational State Transfer |

LIST OF FIGURES

[Figure 2. 1 The concept of blockchain 7](#_Toc122275956)

[Figure 2. 2 How Does a Blockchain Work? 12](#_Toc122275957)

[Figure 2. 3 Decentralized Trust 13](#_Toc122275958)

[Figure 2. 4 Blockchain model compared to traditional model 14](#_Toc122275959)

[Figure 2. 5 Security model of blockchain 15](#_Toc122275960)

[Figure 2. 6 The chart shows the development of industries in blockchain 15](#_Toc122275961)

[Figure 2. 7 IPFS Operation Process 21](#_Toc122275962)

[Figure 2. 8 Django's MVT Model 24](#_Toc122275963)

[Figure 2. 9 Demo through REST web service 25](#_Toc122275964)

[Figure 2. 10 Alchemy interface 26](#_Toc122275965)

[Figure 3. 1 General use case diagram 27](#_Toc122271945)

[Figure 3. 2 Use case diagram for register 27](#_Toc122271946)

[Figure 3. 3 Use case diagram for login 28](#_Toc122271947)

[Figure 3. 4 Use case diagram for search 28](#_Toc122271948)

[Figure 3. 5 Use case diagram for personal library management 29](#_Toc122271949)

[Figure 3. 6 Use case diagram for discovery 29](#_Toc122271950)

[Figure 3. 7 Use case diagram for create playlist 29](#_Toc122271951)

[Figure 3. 8 Use case diagram for upload song 30](#_Toc122271952)

[Figure 3. 9 Design database system 30](#_Toc122271953)

[Figure 3. 10 Discovery page interface 31](#_Toc122271954)

[Figure 3. 11 Library page interface 31](#_Toc122271955)

[Figure 3. 12 Album page interface 32](#_Toc122271956)

[Figure 3. 13 Mint NFT page interface 32](#_Toc122271957)

[Figure 3. 14 List song page interface 33](#_Toc122271958)

[Figure 3. 15 Player music interface 33](#_Toc122271959)

LIST OF TABLES

[Table 1.1 - Related component-based approaches to the defined challenges 2](#_Toc114744541)

INTRODUCTION

##### Problem statement

Over the years, "streaming services" have exploded and changed the habits of both music listeners and creators of music.

In Vietnam, with the top 5 online music sites and hundreds of music download sites, the number of music downloads is also very large along with the percentage of Internet users. According to the current growth momentum, there will certainly be new breakthroughs because copyrighted online music is the future of the world music industry. Since then, the protection of online music copyright is also in place.

Before now, community-driven centralized streaming platforms like Spotify and YouTube gave artists more freedom to express themselves with built-in legal infrastructure support. However, the possibilities are endless with the advent of decentralized music streaming platforms, which give more to music creators and listeners.

Decentralized streaming platforms offer musicians more control over their music, ensure transparency in royalty distribution, eliminate middlemen from music sales (think record labels and distributors), boost revenue for musicians, eradicate the musician/audience divide, and establish a point of origin for music creators.

In fact, artists like Lupe Fiasco, Gramatik, and Pitbull have advocated for decentralized technologies in music, and proponents champion blockchain’s distributed ledger technology.

##### Aims and Objectives

Over the last 10 years, the music industry has changed drastically because of the internet and social media development. Artists have new mediums to share their songs, and fans have many new ways to engage with and support their favorite musicians.

The music industry has its fair share of controversies, from monopolies to the limited earning potential for up-and-coming artists. Because of this, projects are trying to utilize blockchain technology to provide new solutions for the age-old music market.

The purpose of our research at this time wants to build a decentralized music player application based on blockchain technology, revolutionizing the rights and royalties process, and ensuring artists, writers, publishers, and everyone in the music industry value chain is paid appropriately.

##### Structure of the thesis

After the *Introduction*, the thesis is structured in three chapters:

*Chapter 1,* *Decentralized music streaming.* In this chapter. In this chapter, the thesis presents the trend of decentralized music applying blockchain technology.

*Chapter 2, Theoretical infrastructure*. This chapter includes the theory of blockchains, next.js, Interplanetary File System (IPFS), Django

*Chapter 3,* *System analysis and design.* This chapter defines system analysis, system requirements, use case specification, system implementation

Finally, there are *Conclusions*, *Suggestions,* *References* and *Appendices* related to the topic.

# DECENTRALIZED MUSIC STREAMING

## Decentralized music streaming

Music streaming is one huge infrastructure right now driving the music industry globally. Let’s face it, the growth of music streaming platforms has opened up new opportunities for artists and listeners alike. Moreover, this growth is driven by music-loving Gen Z and Millennials, who prefer the subscription economy.

Notably, the U.S. alone has 82.1 million paid subscribers to on-demand music streaming platforms, with global music subscribers surging 26.4% to 523.9 million during the Covid pandemic. The overall ease of streaming music means that more music lovers will keep listening to their favorite music for many years.

Before now, community-driven centralized streaming platforms like Spotify and YouTube gave artists more freedom to express themselves with built-in legal infrastructure support. However, the possibilities are endless with the advent of decentralized music streaming platforms, which gives more to music creators and listeners.

Decentralized streaming platforms offer musicians more control over their music, ensure transparency in royalty distribution, eliminate middlemen from music sales (think record labels and distributors), boost revenue for musicians, eradicate the musician/audience divide, and establish a point of origin for music creators.

In fact, artists like Lupe Fiasco, Gramatik, and Pitbull have advocated for decentralized technologies in music, and proponents champion blockchain’s distributed ledger technology.

So if you are an upcoming musician or are a listener looking to try out decentralized music streaming platforms, this is the piece you need to read as we’ll be uncovering the top 10 decentralized music streaming platforms you need to know. Without further fuss, let’s get this show on the road.

### Remaining problems on centralized music platforms

Although, the online music industry has a lot of potential as well as room to grow in the future. However, at the present time, this art form still has many inherent weaknesses that seem to have not been untied:

Producer's benefits: Online music works are easily copied and plagiarized, so it will be difficult to protect the rights of music creators thoroughly. As a result, this producer's interests are affected, thereby reducing his interest in creating musical works.

Music Copyright: The lack of a unified and standardized online music trading platform for music copyrights makes trading music assets difficult and raises concerns about the copyright of works.

Conflict in copyright law: Currently, there are still many contradictions in the laws of intellectual property rights of musical works in many countries. Therefore, this makes transactions related to this asset class difficult with high payment costs and long verification times.

### Blockchain – the inevitable next stage of music?

Coinbase is the first crypto company to have conducted an IPO. Coinbase has built the largest centralized and regulated exchange for cryptocurrencies and digital assets. Ultimately, the success of Coinbase has raised digital asset awareness for the crypto industry, for the greater good. The same can happen with the creative industries. Digital assets, be they collectibles or digital rights. Creative IP is a huge untapped $500 billion asset class. With an exchange, this asset will benefit from a great decentralized network. At its core, the answer lies in the fact that blockchain will help manage rights better, more efficiently, and in a good protocol to deal with copyright and royalties, thereby setting the stage for future decentralization. . Smart contracts will become the application layer that turns the internet into the native settlement layer, which means fewer middlemen and less computation and royalty shuffling.

Copyright approval will also be facilitated by blockchain technology. Currently, in the world music industry, one of the biggest challenges facing the copyright sector is not having a centralized data warehouse where everyone can easily apply for a license. For example, DJs often remix music (remix). If they ask for permission, they have to find the producer, the main owner, and then determine if the artist is the one authorized to approve the copyright… It's a journey that requires a lot of time and effort. . While with a centralized data warehouse thanks to blockchain technology, everything is solved with the click of a button…

Ujo, a New York-based company, provides a decentralized data storage platform for music productions. Here artists can not only upload their works, but also keep 100% of the revenue and tips from the sale of their products without any fees. The platform can automatically divide the commission among the collaborators of each project. Another project, the Open Music Initiative, uses blockchain technology for artists to copyright their music. This project is increasingly attracting more and more artists from other platforms such as Soundcloud, YouTube, Spotify, Netflix, a testament to applying the power of technology.Bruno Guez, CEO and Founder Revelator, a blockchain-based digital asset and distribution platform for creative IP, also outlined what it takes to decentralize music.

While implementing new technology is important, it is even more important to capture and realize more value for creators. Decentralization can help the music industry do just that, but the road ahead is long. Without protocol-level rights management, the music industry will never fully reap the benefits of blockchain or decentralization.

Blockchain technology is holding in hand a lot of potential tools, to break down the barriers inherent in the music industry, help reduce costs, bring higher efficiency and better income for artists. The advent of the Internet led to music streaming platforms like Napster, Soundcloud and now Spotify bringing music to every corner. Blockchain seems to be the next step when it comes to handing creators and listeners back to the hands of record labels like Sony Music and Warner Music.

### Difficulty when music blockchain application

It's also not easy for music companies to adopt blockchain technology, even where the technology promises, and sometimes brings real creativity and efficiency in areas where the music industry is concerned. really need it. The reason for this is clear: It's what the current music industry needs and what blockchain systems do. As Blockchain becomes popular, the first thing that must be said is that the number of decentralized applications will increase. Meanwhile, the main value of the music business comes from its ability to control creative intellectual property, and it's essentially a centralized operation.

We have to understand what needs to happen to create a truly powerful and diverse decentralized ecosystem for music. That system will be a hybrid system for years to come, regardless of what some crypto miners want or some startups believe. For the music industry to be fully decentralized and permissionless, some tough things need to happen.

First, there needs to be a global copyright framework that can be chained. Around the world, there are 240 systems of collective representation organizations (CMOs). These 240 institutions will become the validating nodes on the network, the computers that approve and ensure the correctness of transactions. Furthermore, there will be a need for an online dispute resolution, counterclaim and complaint resolution protocol that these organizations can engage with with all publishers and sub-publishers worldwide.

Alternatively, there could be a staking mechanism for the right holder to use collateral (probably virtual tokens) when they register the property, the collateral they would lose if they breach the copyright. permission. This will solve the linking problem and encourage good behavior. All of this is a big ask for the music industry. Everything looks simpler if we consider it from the creator's side. They need to register their copyrights and creative works on the blockchain. This is something we're done with, but the next issue is who will pay for this technology and who will maintain it, until it can sustain itself in a completely decentralized way.

Similar to how the Music Modernization Act established The MLC's collective funding in the US, one could argue that a global copyright consortium of 240 CMOs could jointly fund and launch run this new decentralized registry. Of course, things get more complicated on the publishing side.

# THEORETICAL INFRASTRUCTURE

## Blockchain

A blockchain is a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. It is designed in a way that makes it difficult or impossible to change, hack, or cheat the system. Nowadays, most companies run on information. The faster it is received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information that can be accessed only by permission network members. A blockchain network can track orders, payments, production and much more.

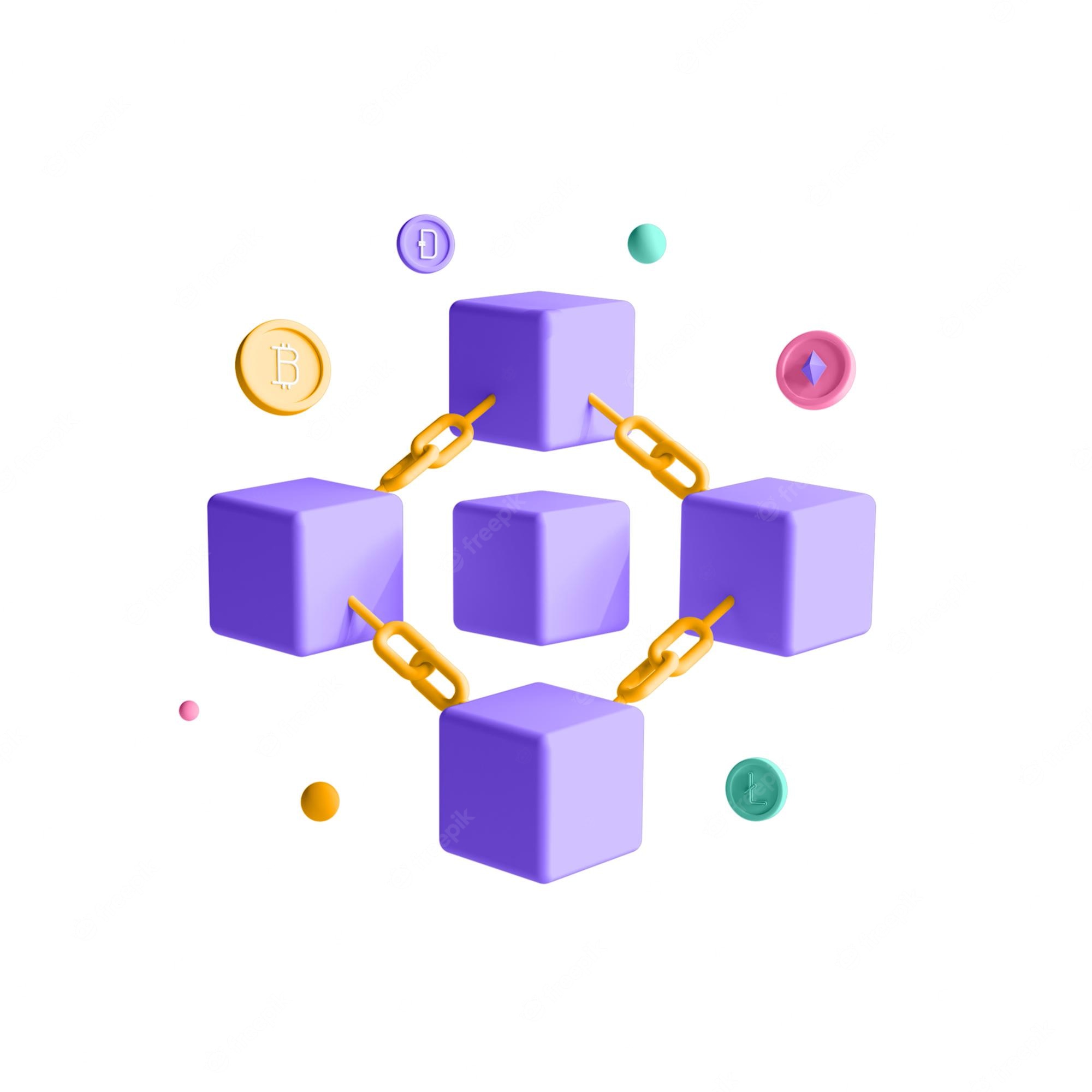


Figure 2. 1 The concept of blockchain

Blockchain has made an impact on today’s technology by revolutionizing the financial industry through the utilization of cryptocurrencies using decentralized control. This has been followed by extending Blockchain to span several other industries and applications for its capabilities in verification. With the current trend of pursuing the decentralized Internet, many methods have been proposed to achieve decentralization considering different aspects of the current Internet model ranging from infrastructure and protocols to services and applications.

The blockchain was invented by Satoshi Nakamoto (alias of a person or a group of 1people) in 2008 to deliver the public transaction ledger of the cryptocurrency bitcoin. Based on its whitepaper, Bitcoin is "a purely peer-to-peer version of electronic cash that would allow online payments to be sent directly from one party to another without going through a financial institution". The invention of the blockchain made it the first digital currency to solve the double-spending problem without the need for an intermediary, such as a trusted authority or central server. Since Bitcoin was raised, new blockchains and cryptocurrencies emerged.

The second-largest blockchain is Ethereum, a decentralized, open-source blockchain with smart contract functionality. A smart contract is a simple program that runs on Ethereum and it enables different activities, including decentralized applications.

The key elements a basic blockchain is built on are transactions and blocks. The transactions are used to keep track of the movement of an asset that can be tangible (a product) or intangible (intellectual). For instance, we can use transactions to record the state of an item during its production. A set of transactions is then collected into a new block which is attached to the latest one inside the network. These blocks form a chain of data as an asset moves from place to place or ownership changes. The blocks confirm the exact time and sequence of transactions, and the blocks link securely together to prevent any block from being altered or a new block being inserted between two existing blocks.

A new block is linked to the previous one through the latter hash. A cryptographic hash function is an algorithm that takes an arbitrary amount of data as input and produces a fixed-size output of enciphered text called hash. A slight difference in the input generates a completely different output. In this way, building a new block based on the previous block hash ensures that middle blocks inside the blockchain can not be altered.

Every time a new transaction occurs on the blockchain, it is added to every node in the network. This results in a decentralised database managed by multiple participants, known as Distributed Ledger Technology (DLT).

### List of top Blockchain features

Blockchain technology has been around for quite some time now, still actively being in the spotlight. Even though there are some mixed feelings toward this technology, yet no one can entirely underestimate its role in the global economic landscape. Blockchain technology isn’t just a backup network for cryptocurrencies, but it offers a lot more.

**a) Immutability**

There are some exciting blockchain features but among them “Immutability” is undoubtedly one of the key features of blockchain technology. But why is this technology uncorrupted? Let’s start with a connecting blockchain with immutability.

Immutability means something that can’t be changed or altered. This is one of the top blockchain features that help to ensure that the technology will remain as it is – a permanent, unalterable network. But how does it maintain that way?

Blockchain technology works slightly different than the typical banking system. Instead of relying on centralized authorities, it ensures the blockchain features through a collection of nodes.

Every node on the system has a copy of the digital ledger. To add a transaction every node needs to check its validity. If the majority thinks it’s valid, then it’s added to the ledger. This promotes transparency and makes it corruption-proof.

So, without the consent from the majority of nodes, no one can add any transaction blocks to the ledger.

Another fact, that backs up the list of key blockchain features is that, once the transaction blocks get added on the ledger, no one can just go back and change it. Thus, any user on the network won’t be able to edit, delete or update it.

**b) Decentralized**

The network is decentralized meaning it doesn’t have any governing authority or a single person looking after the framework. Rather a group of nodes maintains the network making it decentralized.

This is one of the key features of blockchain technology that works perfectly. Let me make it simpler. Blockchain puts us users in a straightforward position. As the system doesn’t require any governing authority, we can directly access it from the web and store our assets there.

You can store anything starting from cryptocurrencies, important documents, contracts or other valuable digital assets. And with the help of blockchain, you’ll have direct control over them using your private key. So, you see the decentralized structure is giving the common people their power and rights back on their assets.

**c) Enhanced Security**

As it gets rid of the need for a central authority, no one can just simply change any characteristics of the network for their benefit. Using encryption ensures another layer of security for the system.

But how does it offer so much security compared to already existing techs? It’s extremely secure because it offers a special disguise – Cryptography.

Added with decentralization, cryptography lays another layer of protection for users. Cryptography is a rather complex mathematical algorithm that acts as a firewall for attacks.

Every information on the blockchain is hashed cryptographically. In simple terms, the information on the network hides the true nature of the data. For this process, any input data gets through a mathematical algorithm that produces a different kind of value, but the length is always fixed.

You could think of it as a unique identification for every data. All the blocks in the ledger come with a unique hash of its own and contain the hash of the previous block. So, changing or trying to tamper with the data will mean changing all the hash IDs. And that’s kind of impossible.

You’ll have a private key to access the data but will have a public key to make transactions.

**d) Distributed Ledgers**

Usually, a public ledger will provide every information about a transaction and the participant. It’s all out in the open, nowhere to hide. Although the case for private or federated blockchain is a bit different. But still, in those cases, many people can see what really goes on in the ledger.

That’s because the ledger on the network is maintained by all other users on the system. This distributed computational power across the computers to ensure a better outcome.

This is the reason it’s considered one of the blockchain essential features. The result will always be a higher efficient ledger system that can take on the traditional ones.

**e) Consensus**

Every blockchain thrives because of the consensus algorithms. The architecture is cleverly designed, and consensus algorithms are at the core of this architecture. Every blockchain has a consensus to help the network make decisions.

In simple terms, the consensus is a decision-making process for the group of nodes active on the network. Here, the nodes can come to an agreement quickly and relatively faster. When millions of nodes are validating a transaction, a consensus is absolutely necessary for a system to run smoothly. You could think of it as kind of a voting system, where the majority wins, and the minority has to support it.

The consensus is responsible for the network being trustless. Nodes might not trust each other, but they can trust the algorithms that run at the core of it. That’s why every decision on the network is a winning scenario for the blockchain. It’s one of the benefits of blockchain features.

There are lots of different consensus algorithms for blockchains over the globe. Each has its own unique way to make decisions and perfecting previously introduces mistakes. The architecture creates a realm of fairness on the web.

However, to keep the decentralization going every blockchain must have a consensus algorithm, or else the core value of it is lost.

**f) Faster Settlement**

Traditional banking systems are quite slow. Sometimes it can take days to process a transaction after finalizing all settlements. It also can be corrupted quite easily. Blockchain offers a faster settlement compared to traditional banking systems. This way a user can transfer money relatively faster, which saves a lot of time in the long run.

These blockchain features make life easier for foreign workers and help to understand Why Blockchain is Important. Many people travel to another country in search of a better life and job and leave families behind. However, sending money to their families overseas takes a lot of time and could become fatal in times of need.

Now, blockchains are way too fast, and they can easily use it to send money to their loved ones. Another fun fact is the smart contract system. This can allow making faster settlements for any kind of contract. This is one of the best benefits of blockchain features to this day. And with the third party out of the way, people can send money with a minimal fee. Seems intriguing, right?

### How Does a Blockchain Work?

The goal of blockchain is to allow digital information to be recorded and distributed, but not edited. In this way, a blockchain is the foundation for immutable ledgers, or records of transactions that cannot be altered, deleted, or destroyed. This is why blockchains are also known as a distributed ledger technology (DLT).

First proposed as a research project in 1991, the blockchain concept predated its first widespread application in use: Bitcoin, in 2009. In the years since, the use of blockchains has exploded via the creation of various cryptocurrencies, decentralized finance (DeFi) applications, non-fungible tokens (NFTs), and smart contracts.

Transaction Process

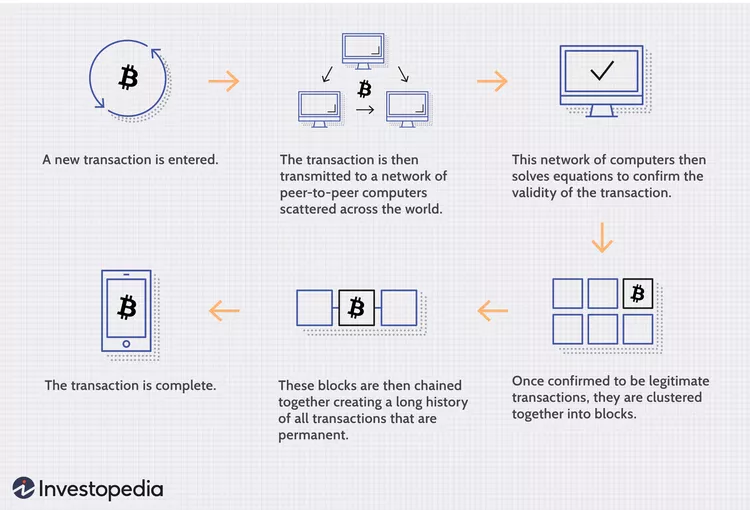


Figure 2. 2 How Does a Blockchain Work?

### Blockchain Decentralization

Imagine that a company owns a server farm with 10,000 computers used to maintain a database holding all of its client’s account information. This company owns a warehouse building that contains all of these computers under one roof and has full control of each of these computers and all of the information contained within them. This, however, provides a single point of failure. What happens if the electricity at that location goes out? What if its Internet connection is severed? What if it burns to the ground? What if a bad actor erases everything with a single keystroke? In any case, the data is lost or corrupted.

What a blockchain does is to allow the data held in that database to be spread out among several network nodes at various locations. This not only creates redundancy but also maintains the fidelity of the data stored therein—if somebody tries to alter a record at one instance of the database, the other nodes would not be altered and thus would prevent a bad actor from doing so. If one user tampers with Bitcoin’s record of transactions, all other nodes would cross-reference each other and easily pinpoint the node with the incorrect information. This system helps to establish an exact and transparent order of events. This way, no single node within the network can alter information held within it.

Because of this, the information and history (such as of transactions of a cryptocurrency) are irreversible. Such a record could be a list of transactions (such as with a cryptocurrency), but it also is possible for a blockchain to hold a variety of other information like legal contracts, state identifications, or a company’s product inventory.

### Pros and Cons of Blockchain

**a) Pros of Blockchain**

**Decentralized Trust**

One of its biggest strengths is that you no longer need to trust a third party to make any transaction. People using blockchain worldwide are confident that no single party is manipulating transactions, viewing personal information or performing any other activity breaching their privacy and security.

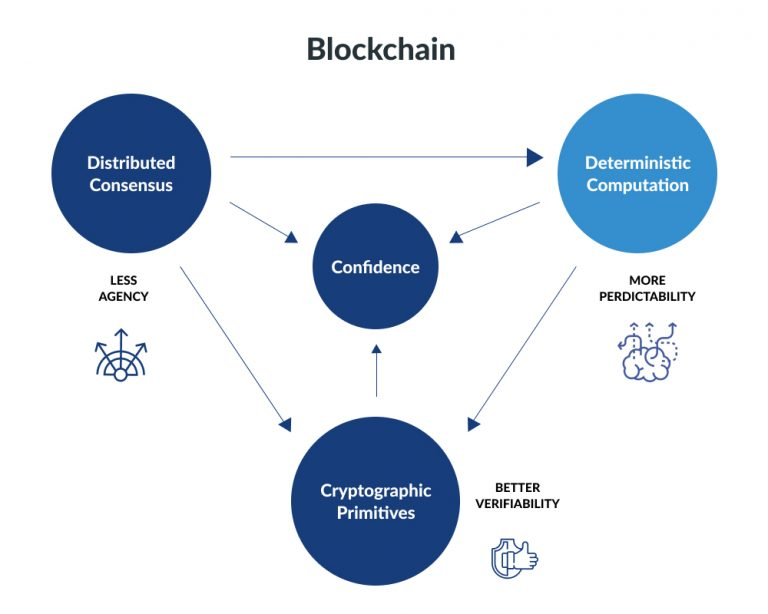


Figure 2. 3 Decentralized Trust

That doesn’t mean blockchain-based applications are always secure—that depends on how good developers are at creating secure code—but it does mean there are opportunities for better security than conventional applications. With blockchain, you can feel more confident about your data and identity.

You only share what you want; companies cannot see your data without your permission. You can also feel more confident about getting paid for providing services. With blockchain, payment is instant; there’s no need to wait days for money orders or checks to clear.

**Low Operational Cost**

Blockchain reduces overhead costs as it has no centralized authority or servers to maintain operations. There are no payment processing or banking fees as it opts for peer-to-peer transactions without third-party approval. It embeds documents, agreements, or transactions within the system.

Blockchain encryptions are more secure against identity theft than conventional payment systems.

**No Single Point of Failure**

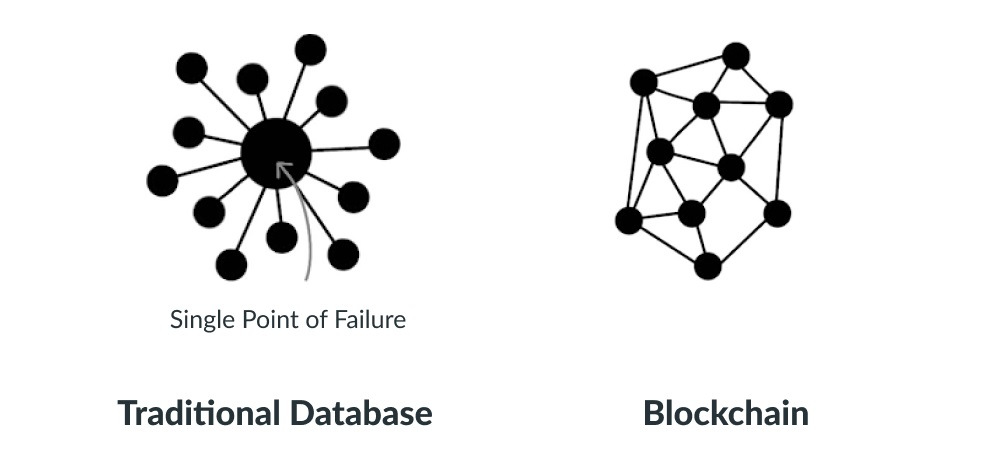
****

Figure 2. 4 Blockchain model compared to traditional model

With blockchain technology, there’s no single point of failure. If a hacker were to gain access to your business’s server or database, they could very easily wipe out your entire network—all at once.

Blockchain technology is not centralized; instead, it is in distributed form. It saves your data if the network goes down as hackers cannot break into the central grid and affect any connected account.

You can create passwords up to 100 characters long, making it impossible for hackers to guess or decode. It gives better security than regular networks with the option of up to 8-character long passwords (including letters and numbers)

**Enhanced Security And Confidentiality**

Being distributed across a global network of computers and protected by cryptography, blockchain technology is inherently more secure than centralized systems.

It has an added layer of confidentiality that secures your data from hackers. Transactions are impossible to trace or link back to an individual user. The user can select their names and e-mail addresses during transactions. You get the option to complete your transactions while remaining anonymous.

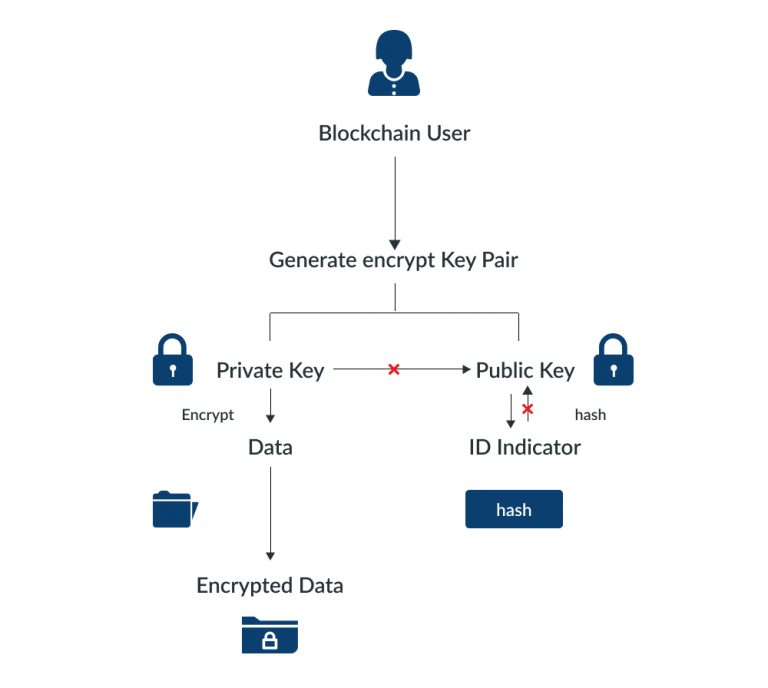
****

Figure 2. 5 Security model of blockchain

So, you can use blockchain-based services without worrying about advertisers tracking your activity or identity thieves accessing sensitive information such as credit card numbers.

**Reduces Fraud**

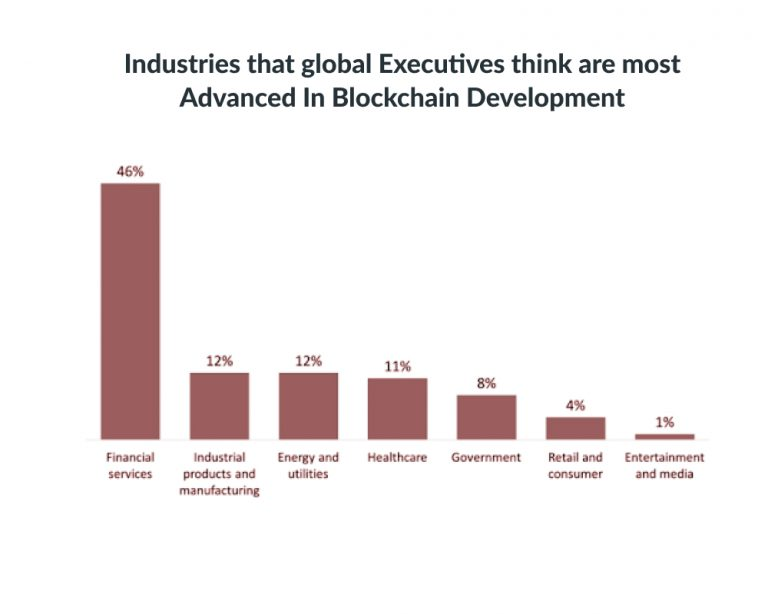
****

Figure 2. 6 The chart shows the development of industries in blockchain

Blockchain technology has some fantastic attributes that make it ideal for financial institutions to reduce forgery. It records every activity, making it impossible for anyone to make duplicate transactions.

Each block stores the financial information, and if any modification is made to a previous block, other nodes on the network rejected it.

Once your bank confirms your transaction, they can’t deny receiving the funds. On top of that, you could see that fraudulent activity happened when another node changed transactions.

**b) Cons of Blockchain**

**Scalability**

Blockchain is capable of handling fewer transactions per second. It causes delays in finalizing the massive volume of transactions resulting in poor scalability. However, several methods have been proposed to overcome this shortcoming, but none has been implemented till now.

**Security**

Blockchain is publicly accessible as a distributed ledger. It may attract any unknown visitor monitoring your wallet. Though there are several provisions to add privacy and encryption layers to enable your preferred privacy, all are not commonplace yet.

Moreover, much of your data is linked directly to your digital identity, so it could potentially expose parts of your private life that you wouldn’t necessarily want online. Security concerns often lead people to trust third-party solutions (like exchanges) over direct blockchain transactions, relinquishing control over personal assets.

**Cost**

One of the biggest problems with blockchain technology is that it requires enormous energy. Because miners have to solve complicated math problems to get a payout, they need powerful rigs that consume tons of electricity.

As a result, some blockchains are incredibly costly to run, especially for smaller businesses or individuals. You cannot make changes later; if you want your blockchain online, you must pay for it up front!

**Competitiveness**

There is a lot of hype surrounding these industries trying to use blockchain. It leads to unnecessary competition between businesses as they opt for this technology and waste their time, money, and efforts even when it is useless for their business. Companies will have no alternative but to invest heavily to keep up with their competitors.

**Speed**

The other significant con to blockchain technology is its speed. Unlike a centralized database, blockchains require miners—or people with high-end computers and dedicated software that solve computational puzzles in exchange for new crypto tokens.

In simple terms, blockchain transactions take longer than traditional payment methods like cash or credit cards. This can be discouraging if you’re interested in using blockchain technology as a daily payment method.

## Overview of Next.js

Next.js is a React framework that allows users to create single-page Javascript apps. This framework has various advantages for both clients’ apps and the development team. As users, one can get increasingly irritated as our expectations are not met by websites and apps that take longer than milliseconds to load. Next.js is widespread and a fine choice for various reasons; most are speed and performance-related.

**Core features:**

* Rich User Experience
* Great performance
* Faster feature development

### What Can You Build with Next.Js and When to Use Next.Js?

BY using Next.js, a user can build numerous digital products and interfaces such as:

**MVP (Minimum Viable Product):** MVP is a product version that encompasses enough features to be used by the early customer and add validation to the product idea in the product development cycle. For instance, the MVP can assist the product team in altering and integrating products as early user feedback in the software industry.

**Jamstack websites:** Jamstack is referred to as the new standard architecture in the web context. JAMstac is a modern web development architecture elicited from Markup (JAM), JavaScript, and APIs. JAMstack cannot be specified as a discrete technology; on the other hand, it is a  different technique for building apps and websites.

**Web Portals:** A portal is a web-based platform that gathers data from several sources into a single user interface and displays it to users in the most appropriate way for their situation. Simple web portals have grown into systems that enable digital customer experience efforts throughout time.

**Single web pages:** A single-page website, sometimes known as a one-page website, is one that only has one HTML page. There are no extra pages like Contact Us, About, or Features page.

**Static websites:** A static website (also known as a flat or stationary page) is displayed precisely as stored in a computer browser. It is made up of HTML-coded web pages kept on a web server. It does not change; it remains the same, or “static,” for all site visitors.

Web programming or database design are not required for a static website. Static websites are the most basic and easiest to design, making them ideal for small-scale sites. Maintaining many static pages can soon become a time-consuming and inefficient chore.

**SaaS products:** SaaS products are internet software hosted by a central provider and allow access to all users, for instance, text expander. DropBox, Google Apps, and Canva are some of the prominent illustrations of SaaS products.

**eCommerce and retail websites:** A website allows individuals to purchase and sell physical goods, services, and digital products online rather than at an actual store. A business can handle orders, receive payments, manage shipping and logistics, and provide customer care through an e-commerce website.

**Dashboards:** A dashboard is a visual representation or display of a user’s data. It provides links to valuable tools and critical information displayed on the website.

**Progressive web applications (PWA):** PWA is a software application built using standard web technologies like Javascript and HTML.

**Interactive user interfaces:** The user interface (UI) is the point of interaction and communication between humans and computers in a device. Display screens, keyboards, mice, and the appearance of a desktop are all examples of this. It can also refer to how a user interacts with a program or a website.

**Blog:** A blog is a reverse chronological online diary, regularly updated web page, or informational website that displays material in reverse chronological order, with the most recent posts at the top. It’s a platform where a writer or a group of authors may express their thoughts on a specific topic.

### Next.js Features

**Hot Code Reloading**: When modifications in the code are saved, it automatically reloads the program.

**Automatic Code Splitting**: This functionality bundles and serves each page with every import in the code. It signifies that no more code is loaded onto the website.

**Ecosystem Compatibility**: JavaScript, Node, and react are compatible with the Next.js ecosystem.

**Server Rendering**: Render react components on the server efficiently before delivering HTML to the client.

**Styled-JSX**: Styled-JSX styled-jsx is a JavaScript extension that allows you to write CSS right into the code.

**Automatic Prefetching**: Only links shown in the viewport are prefetched by Next.js, which uses the Intersection Observer API to detect them. Next.js also disables prefetching in case of a slow network connection or when the user has (Save-Data) on. Based on the following checks, it dynamically injects <link rel="preload"> tags to download components for subsequent navigations. Next.js doesn’t execute JavaScript; it only fetches it. This prevents it from downloading any other stuff that the prefetched page may request until the user clicks on the link.

**Static Exports**: The user can use next export to export Next. Js application to static HTML can be run without a Node. Js server.

**TypeScript Support** : Next.js provided an integrated TypeScript experience out of the box.

### Pros and Cons of Next.js

**a) Pros of Next.js**

Regardless of whether you are looking for benefits from a business perspective or a technical one, you will find some reasons to seriously consider choosing Next.js.

If you want to build a complex and demanding application, React development nature of Next.js allows for saving a lot of time. Developers especially favor  features like:

* **Zero Config**: Next allows you to focus on the business logic of your application instead of the application logic. And to help you, it provides automatic compilation and bundling. In other words, Next is optimized for production right from the start.
* **Incremental Static Regeneration**: it allows you to update the pages by re-rendering them in the background as traffic comes in. So in other words, static content can become dynamic.
* **Hybrid of server side rendering SSR and static site generation SSG**: prerender pages at build time or request time in a single project.
* **TypeScript support**: automatic TypeScript configuration and compilation.
* **Fast Refresh**: fast, live-editing experience – edits made on React components are live within seconds. It works analogically to Hot Module Replacement (HMR).
* **CSS parsers:** possibility to import CSS files from a JavaScript file. New parses improved handling of CSS.
* **Built-in Image Component and Automatic Image Optimization**: this feature automatically optimizes images
* **Automatic code splitting:** automatically reduce the size of the page by splitting the code and serving components only when needed. Modules can be automatically imported too, thanks to the dynamic import option.
* **Data fetching:** this option allows rendering the content in different ways, accordingly to the app’s use case. It can be done by pre-rendering with server side rendering SSR or static site generation and by updating or creating content with ISR.

**b) Cons of using Next.js**

The number of Next benefits is huge and clearly outweighs its cons. However, let’s write them down to be as objective as it’s possible.

* Development and management: the flexibility, given by Next, has its cost – continuous management. To make all desired changes properly, you will need a dedicated person with proper knowledge. The good news is that this person doesn’t have to be a developer.
* Ongoing cost: since Next.js does not provide many built-in front pages, you have to create your own front-end, which will require changes from time to time. It means that you will have to pay a frontend developer to get the job done.
* Lack of built-in state manager: so if you need a state manager in your app, you have to add Redux, MobX or something else.
* Low on plug-ins: you cannot use much of easy-to-adapt plugins.

## InterPlanetary File System IPFS

IPFS stands for Interplanetary File System. This is a peer-to-peer network distributed file system. In essence, it is used to connect devices together. More specifically, IPFS is a network that can transfer decentralized content and allows flexible data management and storage. This is clearer when each computer participating in the network will receive the task of uploading and downloading data without any intermediary or server intervention.

### How does IPFS work?

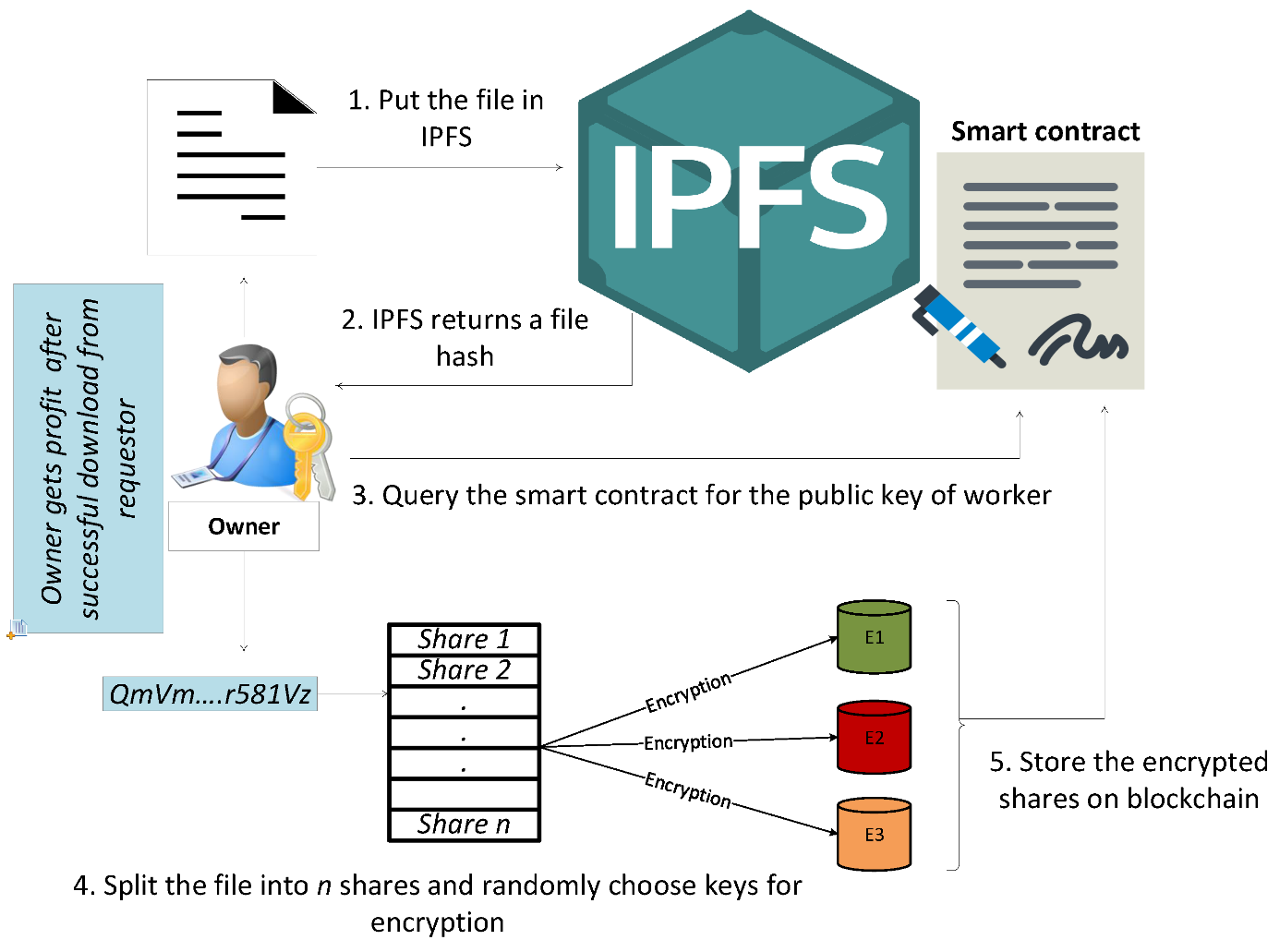


Figure 2. 7 IPFS Operation Process

IPFS seeks to create a permanent and distributed web. It does this by using a content addressing system instead of HTTP's location-based system.

• The HTTP request should look like http://10.20.30.40/folder/file.txt

• The IPFS request will look like /ipfs/QmT5NvUtoM5n/folder/file.txt

How it works:

* Content identification: When you add a file to IPFS, your file is split into smaller pieces, cryptographically hashed, and provided with a unique fingerprint known as a content identifier (CID) . This CID acts as a permanent record of your file as it exists at that time.
* Content Search: When other nodes search your content file, they will ask their peers who are storing the content referenced by the file's CID. When they view or download your content file, they cache a copy and become your other content provider until their cache is cleared.
* Pin content: A node can pin content to make it available forever, or destroy content it hasn't used for a while to save space. This means that each node in the network stores only the content it is interested in, plus some indexing information that helps figure out which node is hosting what.
* Anti-tampering and content moderation: If you add a new version of a file to IPFS, its cryptographic hash function will be different, and so it will receive a new CID. This means that files stored on IPFS are resistant to tampering and censorship. Any changes to the file do not overwrite the original file, and common parts across the files can be reused to minimize storage costs.
* Name the content file: You need to remember a long string of CIDs. IPFS can find the latest version of your file using the IPNS hierarchical naming system, and DNSLink can be used to map CIDs to human-readable DNS names.

Instead of using a location address, IPFS uses a representation of the content itself to resolve the content. This is done using a cryptographic hash function on a file and used as the address. The hash represents a root object and other objects that can be found in its path. Instead of communicating with a server, you have access to the “originating point” of this data. In this way, the system promotes physical connectivity. If someone very close to me has something I want, I get it directly from them instead of connecting to a central server. To do this, IPFS aggregates a number of successful ideas from other peer-to-peer systems.

### IPFS features

Reduce costs: The IPFS model is about reducing costs for both content providers and regular users. IFPS will allow content to be completely downloaded to the IFPS intranet no matter who you are and where you are.

Where to Store: Storing data using IPFS enables deduplication, cluster persistence, and high performance, allowing users to store the world's information for generations to come.

Service Delivery: Delivering massive amounts of data to users, hosted on IPFS can help you cut bandwidth costs with their secure peer-to-peer content delivery. If you're partnering or distributing large data sets, storing that data using IPFS can help speed up performance and unlock decentralized storage.

Off-chain development: The IPFS asset address format allows you to store large files off-chain and place permanent, immutable links in transactions, timestamps, and secure content without having to put the data in the string.

Content creators: IPFS empowers content creators to build and share on the decentralized web, whether it's delivering content for free from the control of an intermediary or creating NFTs that stand the test of time.

Offline users: High latency networks pose a major obstacle for people with poor internet infrastructure. Peer-to-peer IPFS provides flexible access to data independent of latency or backbone connectivity.

Support for a resilient internet: If someone hacks Wikipedia's web servers or an engineer at Wikipedia makes a major mistake that sets their servers on fire, you can still get the same web pages from somewhere else .

Can speed up surfing when you're away or disconnected: If you can retrieve a file from someone nearby instead of hundreds or thousands of miles away, you can usually get it quickly than. This is especially valuable if your community is locally networked but doesn't have a good connection to the wider internet.

### Pros and Cons of IPFS

a) Pros of IPFS

* Completely decentralized storage system.
* The network is built to be highly scalable.
* The network is resistant to denial of service attacks among others because it is completely decentralized. In this way, timely access to information is guaranteed at all times.
* Its use is completely free and the source code is available under a free software license.
* IPFS is extensible, allowing anyone to adapt new functions without major problems. For example, privacy modules, connection to TOR, I2P, among others can be added.

b) Cons of IPFS

* IPFS is a technology in development, so its use in production is not yet widespread.
* IPFS is complex to use for inexperienced users of the system.
* IPFS has no privacy extensions by default.
* Unlike projects like SIA, IPFS was not designed with an incentive model at its core. Because of this, they had to develop separate projects like Filecoin that were limited in their integration.

### Potential applications of IPFS in Blockchain

The core technology of IPFS is based on a distributed hash table (Distributed Table Hash) on a decentralized peer-to-peer network (P2P Network), Blockchain uses distributed ledger technology, because of the same structure, the two platforms These technologies can be combined with each other.

With the advantages of high security, better data transmission thanks to a decentralized operating model, IPFS promises to become an indispensable component in Web 3.0 infrastructure, allowing storage Preventive decentralized data storage against manipulation and tampering.

Besides, IPFS will also become the preferred storage platform of many Dapps because of its censorship resistance.

## What is Django?

Django is a free, open source, Python-based web framework that follows the Model-View-Template (MVT) architectural pattern. It reduces the hassle of web development so that you can focus on writing your app instead of reinventing the wheel.

### Learn about Django's MVT Pattern

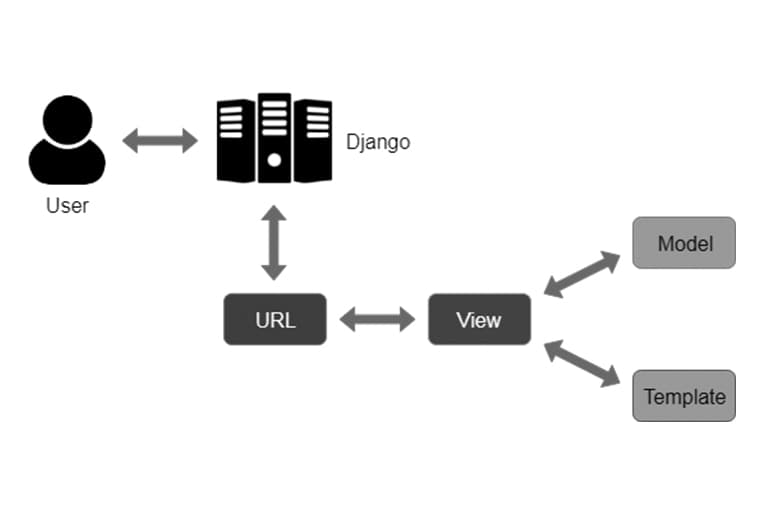


Figure 2. 8 Django's MVT Model

### Why use Django

With django you can get web applications from idea to launch in minutes. And to do this django has a few lightweight features as follows:

Fast: Django is designed to help developers take applications from concept to completion as fast as possible..

Has all the necessary libraries/modules: Django includes dozens of extras that you can used to handle common Web development tasks. Django takes care of user authentication, admin content, site maps, RSS feeds and more - instantly.

Guaranteed Security: Django takes security very seriously and helps developers avoid it many common security flaws, such as SQL injection, cross-site scripting, cross-site request forgery and click. Its user authentication system provides a secure way to manage accounts and user password.

Good Scalability: Some of the busiest places on the planet use fast scalability django's speed and flexibility to meet the heaviest traffic demands.

Flexibility: Companies, organizations, and governments have used Django to build everything - from systems to systems content management to social networks to scientific computing platforms.

### Introduction to Django REST framework

REST stands for REpresentational State Transfer (representative state transfer) is a type of programming architecture, it defines rules for designing web services that focus on system resources. In REST architecture everything is considered as a resource, they can be: text file, image, html page, video, or dynamic data… REST server provides access to resources, REST client accesses. and change those resources. Here resources are identified based on URI, REST uses some representation to represent resources like text, JSON, XML.

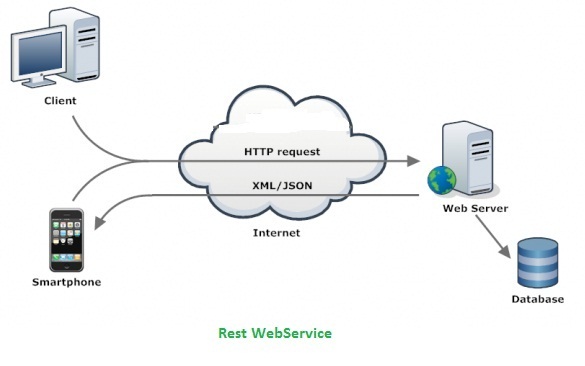


Figure 2. 9 Demo through REST web service

## Alchemy in blockchain

### What is Alchemy?

Alchemy is a blockchain scaling platform that allows developers to securely create, test, and monitor their decentralized apps (DApps). The platform provides dependable network connectivity and node management endpoints.

They simplify decentralized development and go beyond just providing remote nodes with features like Notify — which allows developers to send real-time push notifications to users for critical events based on blockchain activities and their NFT API — that provides a suite of services allowing you to instantly find, verify, and display any NFT across multiple blockchains.

### Supported networks

The Alchemy platform supports DApp development on the Ethereum Layer 1 mainnet as well as testnets such as Rinkeby, Goerli, Kovan, and the Ropsten network.

Furthermore, Alchemy supports Polygon, the Arbitrum network, and Optimism, all of which are Layer 2 networks. Layer 2 networks are distinct chains created on top of Ethereum (Layer 1) as smart contracts, allowing for faster transaction speeds and cheaper gas prices while also increasing the contract’s speed and scalability.

Alchemy supports the Polygon mainnet as well as its testnet (Polygon Mumbai) in addition to the Arbitrum and Optimism mainnets and testnets.

Recently, Alchemy also announced upcoming support for the Solana blockchain. This would Solana developers access to distributed and scalable infrastructure, allowing them to quickly deploy, optimize, and scale DApps on the Solana blockchain.

### Alchemy interface

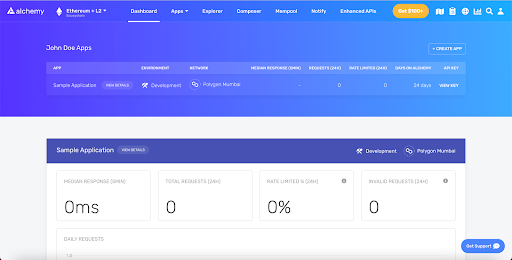


Figure 2. 10 Alchemy interface

You get instant access to all of your applications from its dashboard as well as quick access to concise metrics about each application, such as total requests and responses, the number of invalid requests, and much more.

Viewing additional details about each application gives you the option to properly configure each application, as well as a better analytical view that allows you to filter requests by date, type, and even country.

It’s also worth noting that you have access to Alchemy’s composer, a platform that allows you to configure and send blockchain requests directly using web forms.

### APIs, documentation, and DX

All of Alchemy’s services and integrations are available as top-level APIs that can be accessed through HTTP or WebSockets, and they provide extensive documentation as well as hands-on tutorials for all of their services and network connections. Furthermore, the team developed a wrapper around Web3.js for easier integration and improved access to various APIs while developing DApps.

Alchemy also has security features for your apps. Among the precautions offered are whitelisting addresses so that only these addresses can interact with your contract. Another security solution is to add whitelisted domains and IPs so that requests can only be made using your API keys from these mediums.

Finally, you can easily onboard team members to your projects. You can invite new members to your project straight from the User Settings page, as well as provide administrative privileges to preferred users.

### Pricing

Alchemy provides a generous free plan with no daily request limit for up to 300 million compute units per month. In addition, with this free subscription, you can construct up to five distinct applications and connect them to the platform’s mainnet and testnet. You also have free access to other alchemy solutions including Supernode, Build, Monitor, and Notify.

You can upgrade your application to the growth plan, which costs $49 per month. This increases the monthly compute units to 400 million and the number of applications to fifteen. For more information, you can check out their pricing page here.

# *SYSTEM ANALYSIS AND DESIGN*

## System analysis and design

### System requirements

This is a topic in which it is mainly dealing with the process of managing artists' music updates and postings. The application aims at perfection in both quality and reputation. The application is applied blockchain technology, creating a decentralized system without middle management. All management activities are handled by blockchain, to ensure the author's exclusivity.

System Operation:

* If you are a user, you can search, view a list of songs by topic, listen to each song, listen to albums, register, log in to the system, manage account information, create playlists, manage personal library, follow artists, bookmark favorite songs...
* If you are an artist, you will have the same functions as a user, in addition to the functions of uploading songs, creating albums, and sending private songs to other users.

**a) Identify actors and usecases**

|  |  |  |
| --- | --- | --- |
| **Ordinal number** | **Actor** | **Use case** |
| 1 | Client | * Log in   + Log in accout   + Register an account * Log out * Manage account information   + Update information * Search   + Search by song name   + Search by artist name * Discover   + View recommend songs   + View new song   + View genre   + View recent songs * Library   + View favorite songs   + View artist follow   + View playlist * Create playlist   + Public playlists   + private playlists |
| 2 | Artist | * Log in   + Log in accout   + Register account   + Register accout artist * Log out * Manage account information   + Update information * Discover   + View recommend songs   + View new song   + View genre   + View recent songs * Library   + View favorite songs   + View artist follow   + View playlist * Search   + Search by song name   + Search by artist name * Create playlist   + Public playlists   + Private playlists * Upload song   + Chose album   + Add song   + Add picture   + Add description * Add album   + Create album name * Send song |

Table 3. 1 Indentify actors and use case

### Use case diagram

**a) General use case diagram**

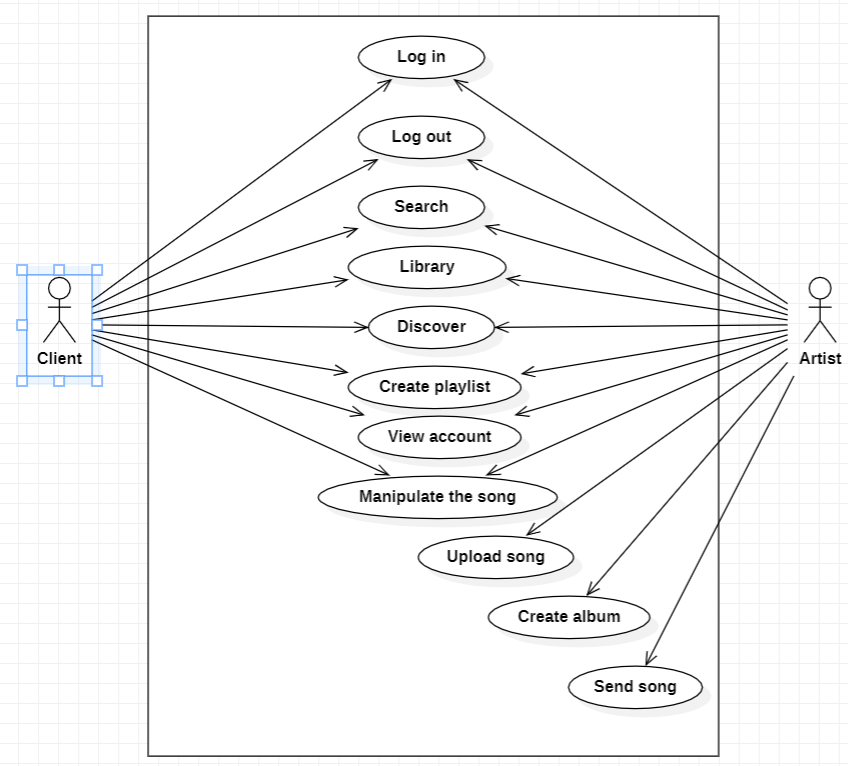


Figure 3. 1 General use case diagram

**b) Detailed use case diagrams**

* **Use case diagram for register**

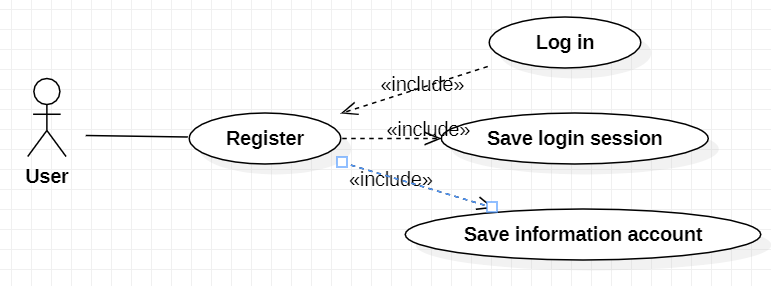


Figure 3. 2 Use case diagram for register

* **Use case diagram for login**

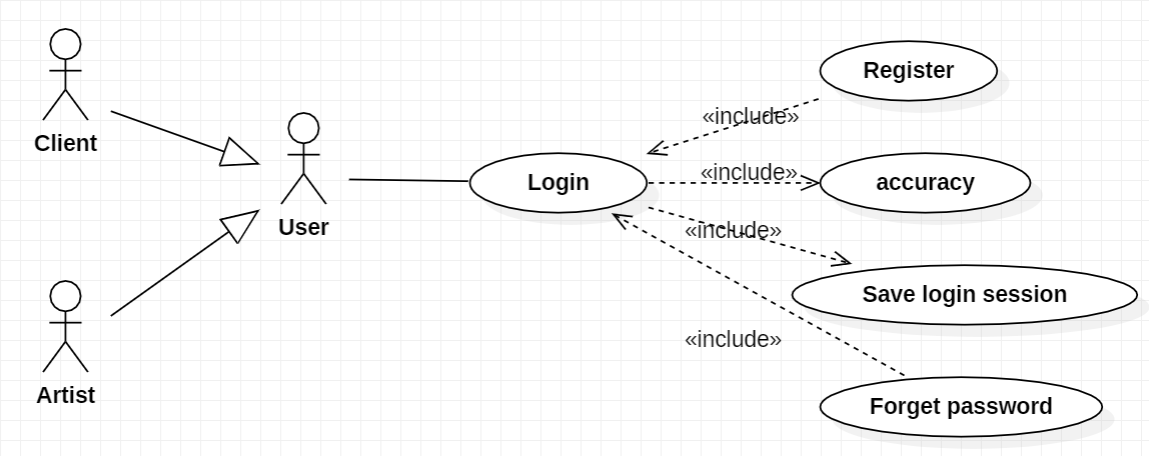


Figure 3. 3 Use case diagram for login

* **Use case diagram for search**

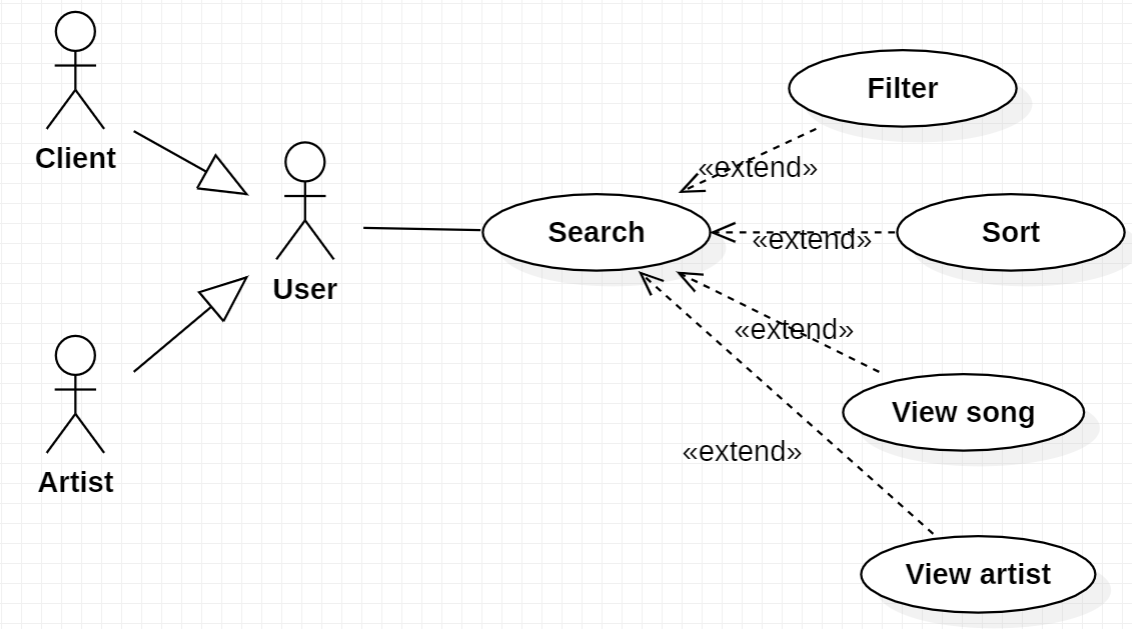


Figure 3. 4 Use case diagram for search

* **Use case diagram for personal library management**

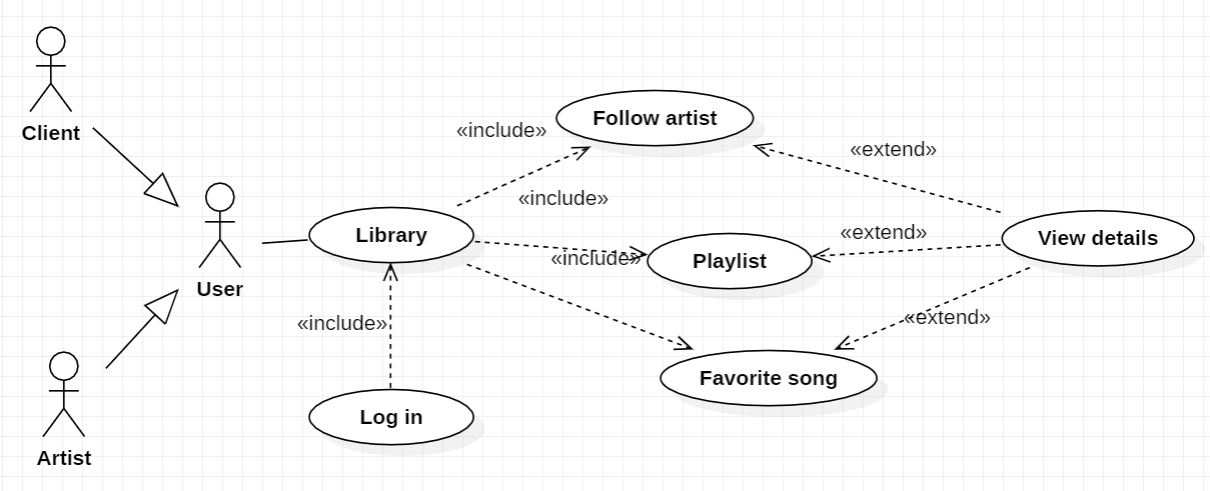


Figure 3. 5 Use case diagram for personal library management

* **Use case diagram for discovery**

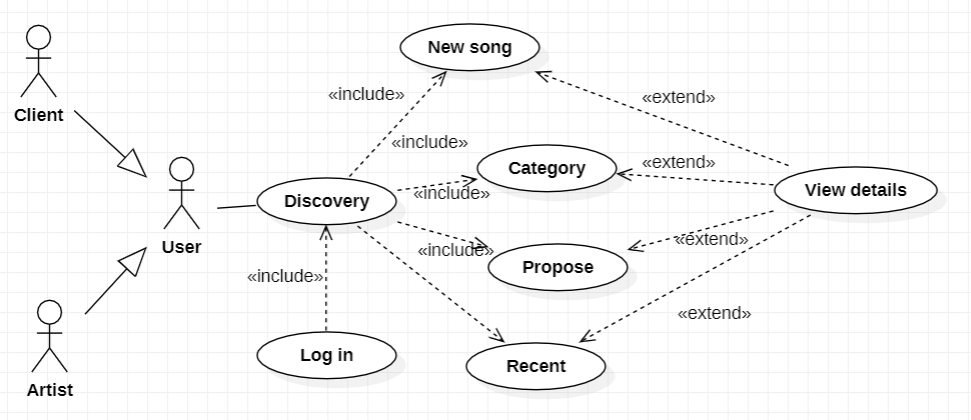


Figure 3. 6 Use case diagram for discovery

* **Use case diagram for create playlist**

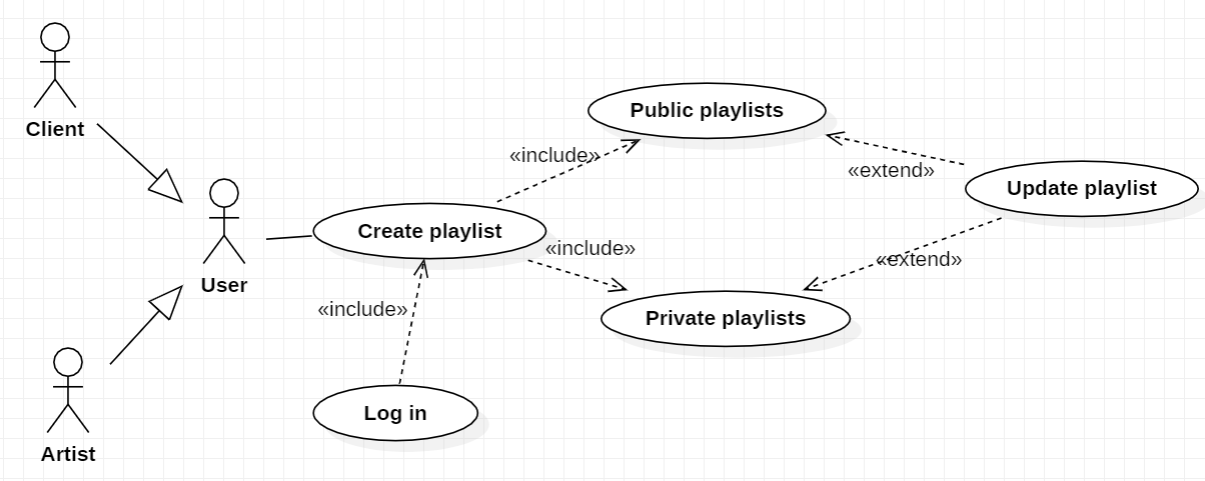


Figure 3. 7 Use case diagram for create playlist

* **Use case diagram for upload song**

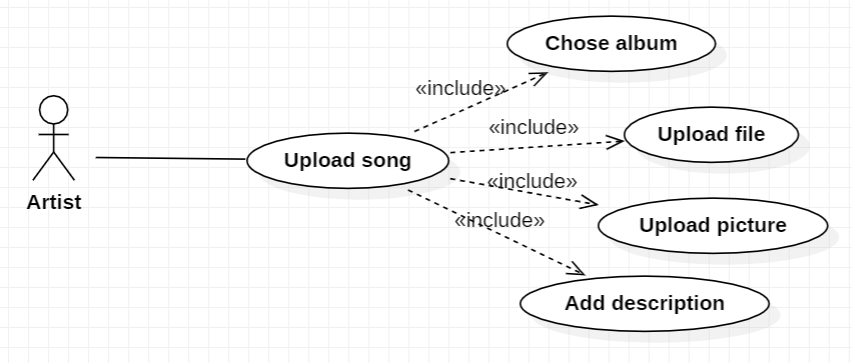


Figure 3. 8 Use case diagram for upload song

## System deployment

### Database Design

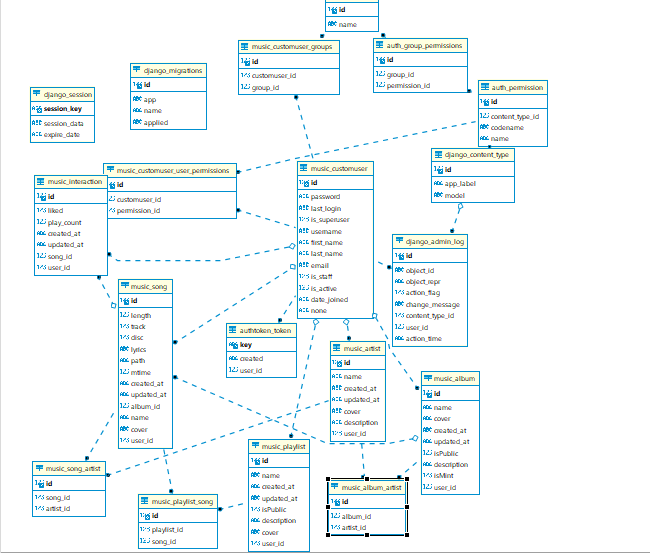


Figure 3. 9 Design database system

### System interface

**Discovery page interface**

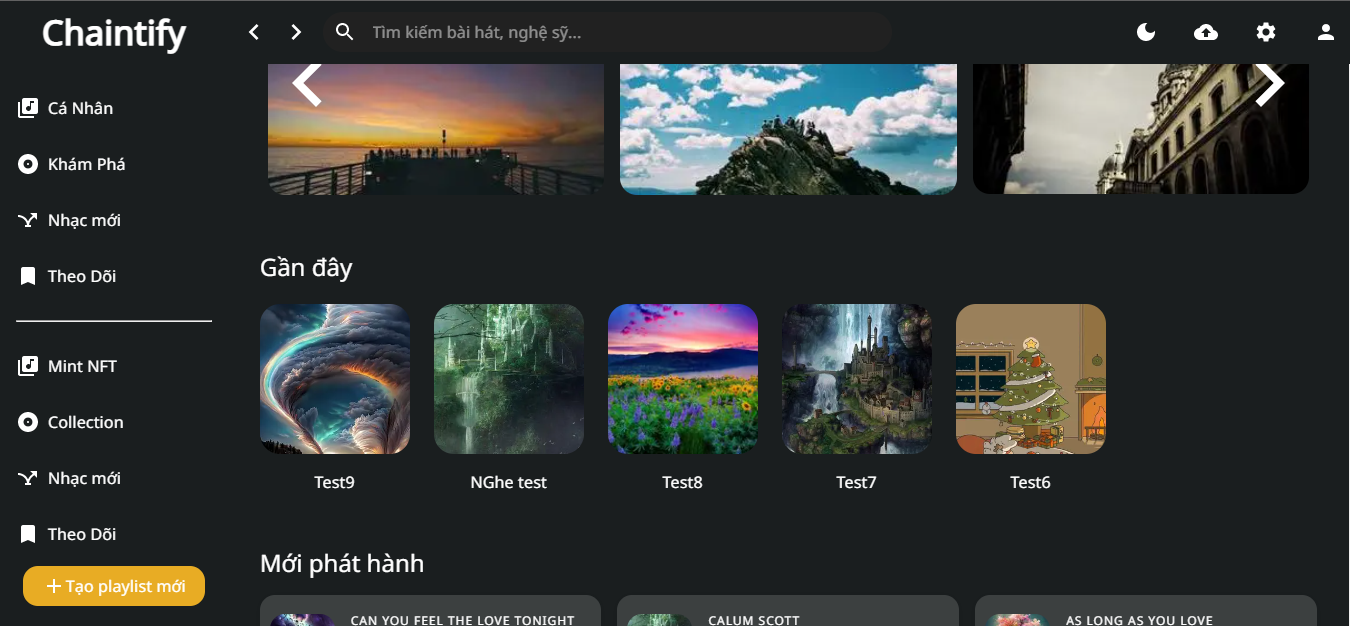


Figure 3. 10 Discovery page interface

**Library page interface**



Figure 3. 11 Library page interface

**Album page interface**

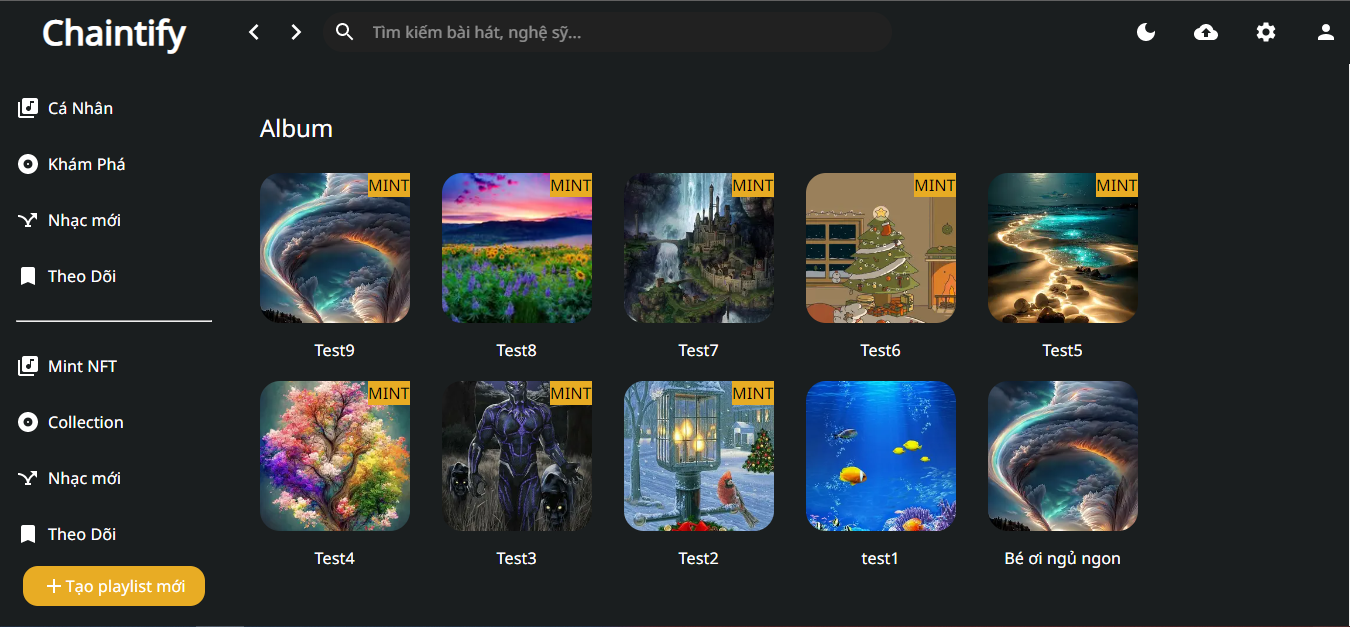


Figure 3. 12 Album page interface

**Mint NFT page interface**

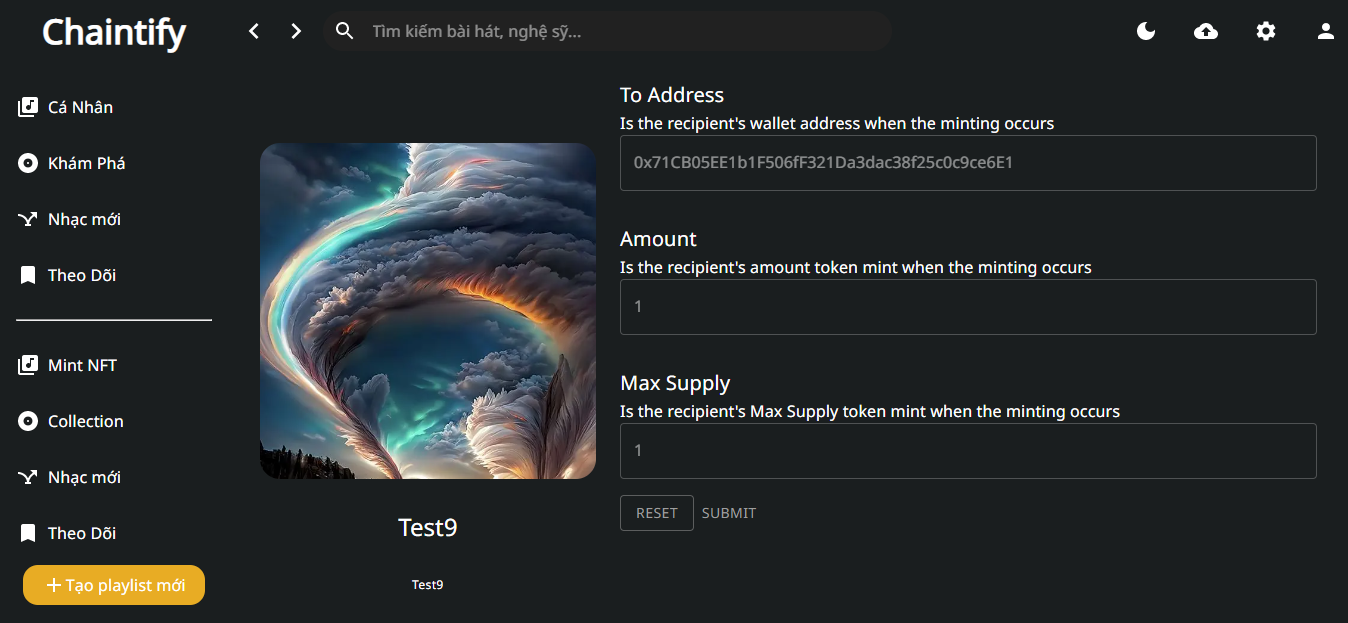


Figure 3. 13 Mint NFT page interface

**List song page interface**

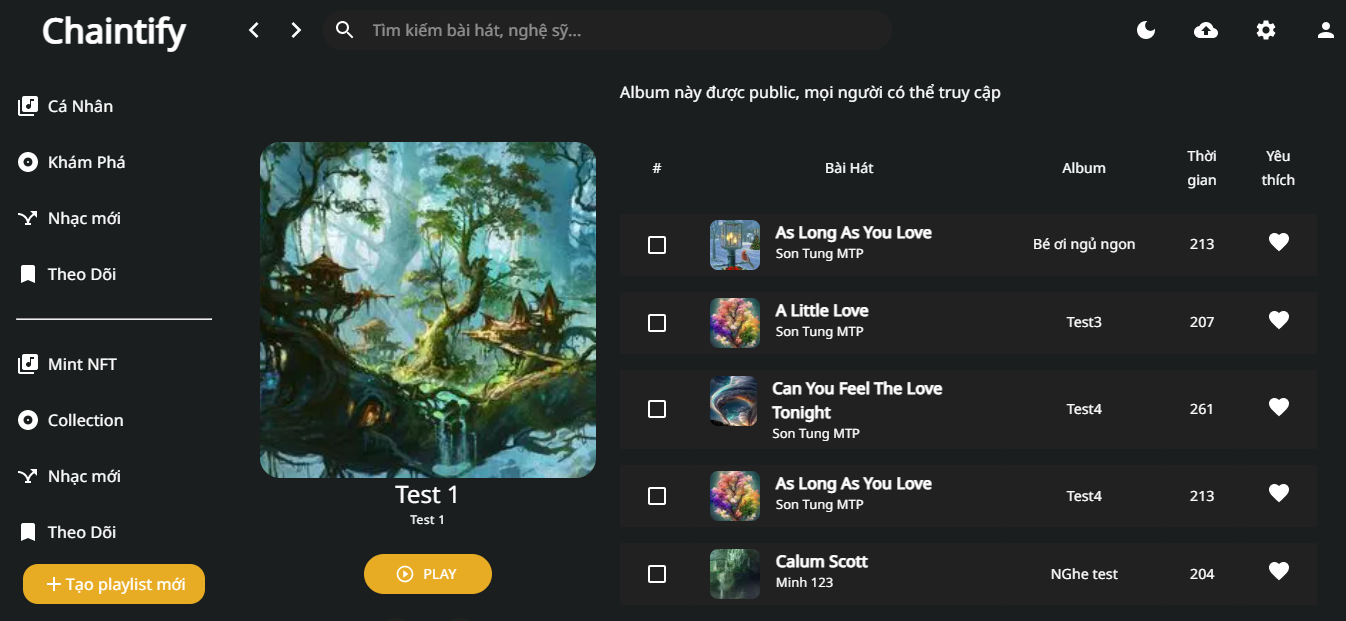


Figure 3. 14 List song page interface

**Player music interface**

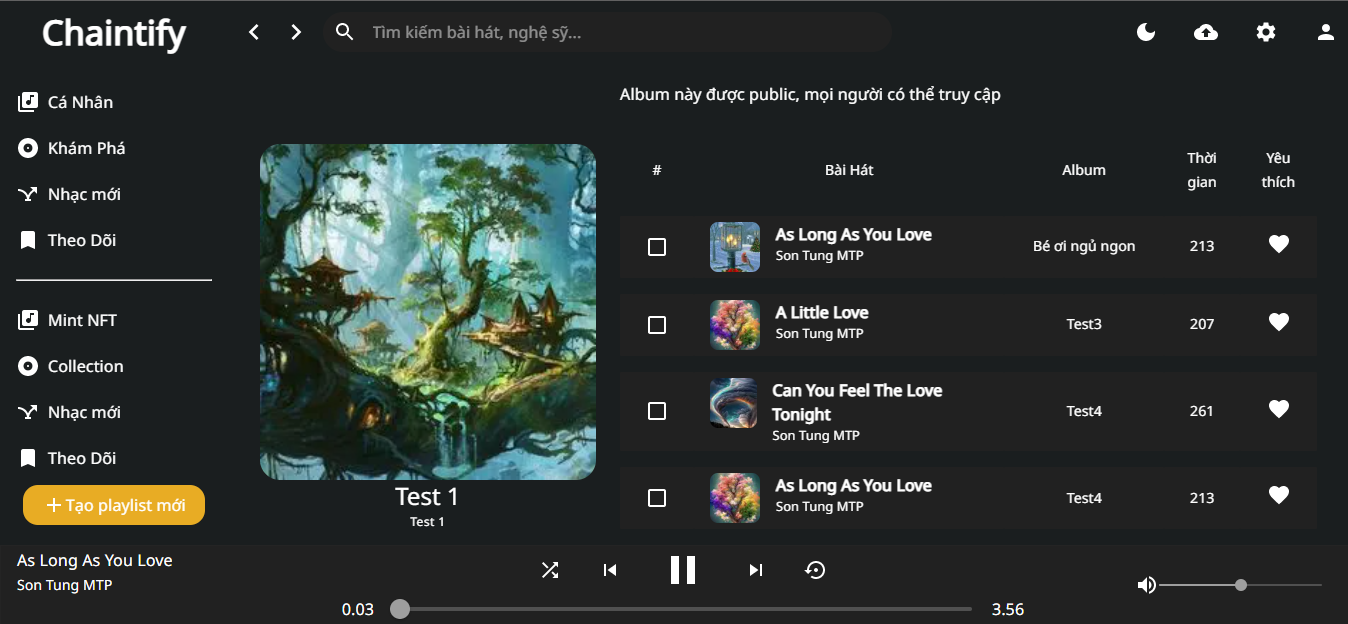


Figure 3. 15 Player music interface

## Summary

*In this chapter, we have ...............*

CONCLUSIONS AND SUGGESTIONS

##### Conclusions

Due to limited qualifications and not much time invested in the topic, the project achieved the following results:

* Present reports in a scientific and systematic way of their own knowledge and understanding, with reference to documents on issues related to the content of research and research.
* During the project, the topic has improved more knowledge learned at school during the semester.
* Increase the ability to think logically, can independently research an issue that we were not interested in before.
* Gain valuable experience in the design, familiarization and use of network models.
* Create a decentralized music player application applying Blockchain technology
* Learn and apply libraries, techniques, frameworks such as: next.js, IPFS, Django REST framework

**Comment on the report.**

**Advantages:**

* Tried to present the report in a scientific and systematic manner with his knowledge and understanding, with reference to documents on issues related to the content of research and research.
* Try to stick to the outline and follow the instructions of Dr. Tran Van Dai, but the report was made in a short time, so errors cannot be avoided. We look forward to the comments of the teachers to make the report more complete.

**Disadvantages:**

* Not tested on wide area network.
* Not tested on different operating systems
* The report has not fully resolved the problems arising in the processing process.
* The report is not aesthetically pleasing, the writing style is still confusing, there are still many inaccuracies that need to be overcome in the process of developing and upgrading the software at a later stage.

##### Suggestions

This is a problem with great potential in the process of integration, in order to develop into a complete system that can be widely applied in practice, the program needs:

* Upgrade the system to be able to apply on a wide area network on many different operating systems.
* Design program interface more professional.
* Development and expansion of application suitable for multinationals

REFERENCES

1. A. Dittmar. “More precise descriptions of temporal relations within task models”. In Interactive Systems: Design, Specification, and Verification, 7th International Workshop DSV-IS, Proceedings, pages 151–168, Limerick,Ireland, 5-6 June 2000.
2. E. G. Sirer and B. N. Bershad, "Using Production Grammars in Software Testing", in Proceedings of the Second Conference on Domain Specific Languages, Austin, Texas, October 3-5,1999.
3. Fewster M., Graham D., (1999), *Software Test Automation – Effective Use of Test Execution Tools, Reading,* MA: Addison – Wesley.
4. Fujiwara S, Bochman G, Khendek F, Amalou M, Ghedasmi A. “Test selection based on finite state models”. IEEE Transactions on Software Engineering 1991; 17(6):591–603