

MUSIC MART – A SOLUTION WITH BLOCKCHAIN

BY

MD TAHSIN IFTEKHAR CHOWDHURY
ID: 191-15-12370

This Report Presented in Partial Fulfilment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering.

Supervised By

MD AZHARUL ISLAM TAZIB
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

MR. MD. SADEKUR RAHMAN
Assistant Professor
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
FEBRUARY 2023

APPROVAL

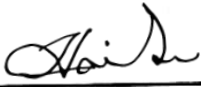
This project titled “**Music Mart-A Solution with Blockchain,**” submitted by Md Tahsin Iftekhar Chowdhury, ID No: 191-15-12370, to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering (BSc) and approved as to its style and contents. The presentation has been held on 02/02/2023.

BOARD OF EXAMINERS



Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Dr. Sheak Rashed Haider Noori
Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Md. Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



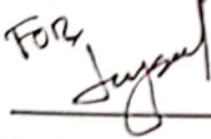
Dr. Md. Sazzadur Rahman
Associate Professor
Institute of Information Technology
Jahangirnagar University

External Examiner

DECLARATION

I hereby declare that I have done this project under the supervision of **Md Azharul Islam Tazib, Lecturer, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Md Azharul Islam Tazib

Lecturer

Department of CSE

Daffodil International University

Co-Supervised by:



Mr. Md. Sadekur Rahman

Assistant Professor

Department of CSE

Daffodil International University

Submitted by:



Md Tahsin Iftekhar Chowdhury

ID: 191-15-12370

Department of CSE

Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartfelt gratitude to Almighty ALLAH for His divine blessing in making me possible to complete the final year project/internship successfully.

I am grateful and wish my profound indebtedness to **Md Azharul Islam Tazib, Lecturer, Department of CSE** Daffodil International University, Dhaka. My supervisor's thorough expertise and strong interest in the field of Web Applications, Blockchain Technologies for his tireless patience, expert guidance, unwavering encouragement, dedicated supervision, insightful criticism, and invaluable advice in completing this project and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I want to express my sincerest gratitude to **Professor Dr. Touhid Bhuiyan, Head Department of CSE**, for his kind help in finishing this project and to other faculty members and the staff of the CSE department of Daffodil International University.

I want to acknowledge our entire coursemates at Daffodil International University, who participated in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

The NFT trend is projected to continue its growth in the upcoming years. The popularity of NFTs is rising rapidly and it has now become possible to build a digital world known as the Metaverse. The Metaverse enables users to create their own avatars and immerse themselves in a virtual environment. This groundbreaking development combines several innovative trends into a unified experience. Avatars, social networks, workout routines, virtual conferences, and work meetings can all be created within the Metaverse. The recent rebranding of Facebook to Meta marks its arrival on the global stage. So, I developed a web application, “**MUSIC MART – A Solution with Blockchain**”, to help music artists find suitable platforms to create, sell, and buy music. It will help them in their growth in this modern era.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgments	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTRODUCTION	1-4
1.1 Introduction	1
1.2 Objective	1
1.3 Motivation	2
1.4 Expected Outcome	3
1.5 Report Layout	3
CHAPTER 2: BACKGROUND	5-8
2.1 Introduction	5
2.2 Related Works	5
2.3 Comparative Studies	6
2.4 Scope of the Problems	6
2.5 Challenges	8
CHAPTER 3: REQUIREMENT SPECIFICATION	9-24
3.1 Introduction	9
3.2 Business Process Model	9
3.3 Hardware & Software Requirements for System	10

3.4 Use Case Model	10
3.5 System Use Case	10
3.6 Data flow diagram	11
3.7 Implementation Requirements	12
CHAPTER 4: DESIGN SPECIFICATION	24-29
4.1 Front-end Design	24
4.2 Back-end Design	28
CHAPTER 5: IMPLEMENTATION AND TESTING	30-33
5.1 Implementation of Database	30
5.2 Implementation of Blockchain Network	30
5.3 Implementation of Front-end Design	31
5.4 Implementation of Interaction	32
5.5 Testing Implementation	32
5.6 Test Results and Reports	33
CHAPTER 6: CONCLUSION & FUTURE SCOPE	34
6.1 Conclusion	34
6.2 Future Scope	34
REFERENCES	35-36

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.5.1: System Use Case for User	11
Figure 3.5.2: Data Flow for Seller	11
Figure 3.5.3: Data Flow for Seller and Buyer	11
Figure 3.5.4: Data Flow Diagram	12
Figure 3.7.1: Ethereum Logo	13
Figure 3.7.2: Web Concepts	17
Figure 4.1.1: Home Page Top	25
Figure 4.1.2: Home Page Bottom	25
Figure 4.1.3: Individual NFT Page Sample-1	26
Figure 4.1.3: Individual NFT Page Sample-2	26
Figure 4.1.5: Buying/ Selling NFT Page	26
Figure 4.1.6: Checkout Page	27
Figure 4.1.7: In Progress of Buying an NFT	27
Figure 4.1.8: Completion of Buying an NFT	27
Figure 4.1.9: NFT Creation Form	28
Figure 4.1.10: Uploading File in NFT Page	28
Figure 4.1.11: Payment Page	28
Figure 4.1.12: Transaction Details Page	29
Figure 4.1.13: After Creating an NFT Page	29
Figure 4.2.1: Web3 Storage	30
Figure 4.2.2: Ethereum (Goerli) Test Network	30

CHAPTER 1

INTRODUCTION

1.1 Introduction

“Music Mart” is a web application using blockchain technology for music artists to find out a suitable platform for creating, selling, and buying NFTs. An NFT marketplace is a digital web platform that utilizes blockchain technology to facilitate the buying and selling non-fungible tokens (NFTs).

NFTs, or non-fungible tokens, are digital assets that represent ownership of a unique item or piece of content [1], such as a digital art piece, collectible, or virtual real estate. Unlike fungible assets like cryptocurrencies, which are interchangeable, NFTs are one-of-a-kind and cannot be replicated. They are created and stored on a blockchain, which ensures their authenticity and ownership, and they can be sold, bought and traded just like physical assets. NFTs have gained popularity in recent years as a new way for creators and collectors to monetize and authenticate digital content.

Despite the short-term fluctuations in the quickly developing NFT field, the global NFT market is expected to grow from \$3.0 billion in 2022 to \$13.6 billion by 2027. An NFT represents a certificate of authenticity and ownership of a unique digital or physical asset. Though often compared to cryptocurrencies that also run on blockchain technology, NFTs are unique and cannot be exchanged for an equal value, different from fungible assets such as cryptocurrencies.

NFTs are stored on a public blockchain and contain the information about:

- Who and when created an asset.
- Who and when bought the asset.
- At what price the asset was bought.
- Who owns the asset at the moment.

1.2 Objectives

Blockchains are designed to create authentic and unique digital assets, known as non-fungible tokens (NFTs). These assets are distinct and cannot be replicated, making them

valuable and desirable. While Ethereum is currently the most popular blockchain for creating NFTs, other blockchain platforms can also support the creation and issuance of NFTs. This allows for greater flexibility and choice in the types of digital assets that can be created and traded on different blockchain networks.

Non-Fungible Tokens can be anything from games to digital stuff. Indeed, tickets for concerts or other events can be represented as NFTs. Using blockchain technology, event organizers can create unique, tamper-proof digital tickets that cannot be replicated. This eliminates the need for physical ticket printing and makes it much easier to manage the distribution and sale of tickets. Additionally, using NFTs for concert tickets also enables a more secure and efficient way to transfer and resell tickets by ensuring the ticket's authenticity and eliminating the risk of fraud and forgery. The minute you get one, it is like a contract stating that the specific item or collectible belongs to you. Through the blockchain, you can verify your ownership of the item.

Here are our application goals:

- Buying and selling platform for music/ sound
- Transferring ownership as NFT
- Making opportunities for music artists to sell their music, song, sound, tune, etc.
- Helping content creators

1.3 Motivation

NFTs are becoming the most popular investment opportunity increasingly right now. Everyone is very interested in buying NFTs. The future of the bought NFT is 50-50 in terms of owning entirely and selling them for a higher price. These NFTs are traded on a platform called the NFT marketplace. The NFT marketplace is the medium for purchasing and selling NFTs. This platform is developed on blockchain technology and smart contracts.

The motivation behind building an NFT marketplace is to provide a platform for creators and collectors to monetize and authenticate digital content in a unique and valuable way. Using blockchain technology, NFT marketplaces can ensure the authenticity and ownership of digital assets and provide a secure and transparent way for transactions. Additionally, NFT marketplaces can provide a new revenue stream for

the music industry and offer new opportunities for creators and collectors to discover and appreciate digital art and collectibles.

Hence, the complete exchange of the NFT from the seller to the buyer is done with the help of a smart contract. These smart contracts are programmed with various terms and conditions. If the transaction complies with all the terms and conditions, the transaction takes place. These smart contracts are developed on the blockchain network. Based on the blockchain, the transaction and minting costs are decided. And these cost factors are programmed on the smart contract.

NFT marketplace development is right now the best business idea. A wide range of business platforms is planning to jump a business in the digital space with NFT. Therefore, now is the time to create an NFT marketplace. With the help of NFT marketplace development, business platforms can earn a wide range of money for their business. On the other hand,

- No such platform like this.
- Current situation of the music industry.

1.4 Expected Outcome

“Music Mart” is a web app that is easy to use and user-friendly for finding suitable music artists for NFT buying and selling. It will be a platform for collectors and creators. By using only, a smartphone or computer, someone can create, sell and buy NFTs for their music. Besides, by launching my own NFT Marketplace, I can gain more visibility, revenue, and user density than any other crypto business. The marketplace has a core set of functionalities:

1. Minting and listing an NFT
2. Buying and selling an NFT
3. Viewing listed NFTs, NFTs can be owned, and NFTs can be sold

1.5 Report Layout

This report consists of 6 chapters. These are:

Introduction

- Introduction
- Objectives
- Motivation
- Expected Outcome
- Report Layout

Background

- Introduction
- Related Works
- Comparative Studies
- Scope of the Problem
- Challenges

Software Requirement Specification

- Introduction
- Business Process Model
- Hardware & Software Requirements for the System
- Use Case Model
- System Use Case
- Data Flow Diagram
- Implementation Requirements

Design Specification

- Front-end Design
- Back-end Design

Implementation and Testing

- Implementation of Database
- Implementation of Blockchain Network
- Implementation of Front-end Design
- Implementation of Interaction
- Testing Implementation
- Test Results and Reports

Conclusion and Future Scope

- Conclusion
- Future Scope

CHAPTER 2

BACKGROUND

2.1 Introduction

NFTs (Non-Fungible Tokens) are unique digital assets that are stored on a blockchain network. They are different from cryptocurrencies such as Bitcoin or Ethereum as they cannot be exchanged for an equal value. NFTs are used to represent digital ownership of various assets such as artwork, music, videos, and collectibles. They are verified by the blockchain network, which ensures the authenticity and scarcity of these assets. NFTs are becoming gradually popular as they provide a new way to own and trade digital assets [2], creating a new market and industry. The popularity of NFTs is likely to continue to grow, as the concept of digital ownership becomes more widely accepted and integrated into our digital lives.

NFT has already become a mainstream phenomenon. This tendency can be explained by the growing number of influencers involved, the emergence of new gaming communities, and the increasing demand for digital art.

There are already many NFT marketplaces, so that this niche can be quite challenging. However, with an experienced team on your side, it is possible to build a unique, cost-effective platform to attract new users. So, I inspire to make this platform specifically for music artists.

2.2 Related Works

Think of NFTs as digital products and the marketplace as an online store. Just like eCommerce platforms trade clothing or, for example, home decor items, NFT marketplaces can target different niches.

In the field of NFT marketplaces, a significant amount of work has been done to develop new technologies, platforms, and applications that allow for the creation, exchange, and collection of NFTs. This work has involved collaboration between artists, developers, and blockchain experts to create user-friendly and secure platforms that cater to the needs of NFT enthusiasts.

Some notable NFT marketplaces include OpenSea, Rarible, SuperRare, and Nifty Gateway, developed to allow artists and creators to monetize their digital creations and offer unique experiences to collectors. Additionally, there has also been a growing interest in creating NFT marketplaces for music and gaming, providing new revenue streams for artists and content creators.

There has also been a growing amount of research into the economics and governance of NFT marketplaces, exploring this new asset class's potential benefits and challenges. This work has involved developing new methods for pricing and valuing NFTs, exploring the potential impact on the music and gaming industries, and examining the social and ethical implications of NFTs.

There are some NFT marketplaces for digital arts, but not for music priority.

2.3 Comparative Studies

For this application, one needs to be connected with an internet connection and a digital wallet. NFT marketplace includes research comparing various NFT marketplaces and platforms in terms of their features, functionality, user experience, and overall performance. These studies aim to identify the strengths and weaknesses of different NFT marketplaces and platforms and to highlight the areas where they excel or need improvement. The focus of these studies is often on areas such as security, scalability, usability, overall user experience, and the level of support and resources available to NFT creators and sellers. These comparative studies may also consider factors such as transaction fees, liquidity, the size of the NFT market, and the level of community engagement and user participation. These studies aim to provide insights and recommendations that can help stakeholders make informed decisions when choosing an NFT marketplace.

2.4 Scope of the Problems

The scope of the problem related to the NFT marketplace includes the limitations and challenges in creating a secure and efficient platform for buying and selling unique digital assets. This includes issues with transaction processing, storing and distributing NFTs, and the need for user-friendly interfaces for buying and selling NFTs.

Additionally, there are concerns about the authenticity and ownership of NFTs, which need to be addressed to build a reliable NFT marketplace.

- 1) It has been over 24 hours now, and even though the transaction shows success, it still has not fully completed the transaction/swap. I gather that the transaction fee (measured in Gwei) can be a factor; however, my GWEI measurement shows 103, which is well in line with current Gas Tracker stats. Most importantly, I do not know how to check how long my transaction will take to complete, whether it is 2 hours from now or another week. As I mentioned, it has already been over 24 hours. An Ethereum (ETH) can take anywhere from 15 seconds to multiple days, depending on network congestion and the gas fees paid by the sender.
- 2) Ethereum gas is a term that describes the fee required for executing transactions and smart contracts on the Ethereum blockchain. It is used to pay for the computational power needed to perform these operations on the Ethereum Virtual Machine (EVM). Gas is necessary for the smooth functioning of the Ethereum network, as it ensures that all transactions and smart contracts are executed fairly and efficiently. It also serves as a mechanism to prevent spamming the network with unnecessary or malicious transactions by making it more expensive to perform these actions. In short, Ethereum gas is an essential component that enables the Ethereum blockchain to function securely and efficiently. They are your way of paying stakers who validate your transactions. Each unit of gas has a price denoted in gigawei (*gwei*) or *nanoeth*. A gwei equals 1,000,000,000 wei, which is the smallest Ether unit base. To clarify the connection between Ethereum and gas, one wei is equal to 0.000000001 ETH. Ethereum gas calculator can be used to determine the amount of gas required for a specific transaction or smart contract operation. This will provide you with an estimate of the Ethereum (ETH) cost required to perform the operation. The cost of gas can fluctuate based on the network congestion and the complexity of the operation. It's important to note that gas prices are dynamic and can fluctuate over time. By using a gas calculator or

monitoring gas prices on the Ethereum network, you can better understand the cost of executing transactions and smart contracts on the Ethereum blockchain.

2.5 Challenges

Lack of Medium

It would cause us to run into further difficulties. The fact that this application is online based that if someone wants to use this application, they will require a smart device. Additionally, we know that many of our music artists and collectors are the app's primary target users. This particular challenge is notable. They lack immediate access to learning about innovation. Second, without a sharp tool, they will lose their will. Some will use it, while others will not. So, our app is mainly can be used on smart devices. The user needs to have a smartphone or computer with a digital wallet to use our app and get the best results. If they don't have a smartphone or computer, they will not be able to use the application. One of the main features of this app is buying and selling NFTs. So, this could be one of the biggest challenges for this project.

Publicity

This project has a lot of publicity challenges. I have to publicize my project more and more to ensure everyone knows about it. Most of the farmers in our country can not read or write, so I must first be aware of the project and teach them how to use it. Therefore, I need to print a leaflet or have a program that teaches them how to use it. It is not easy to get information about this project to the people. Besides these, to spread out about this app to the masses, I have to make more publicity through social media, which is a very powerful tool nowadays. Making TVC and social media advertisements for the mass people will be a way of publicity. Like this, many more steps have to be taken to do proper publicity for this app. Otherwise, people will not get the proper facility for using it.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Introduction

In any phase of the development cycle, requirement specification is essential in software engineering. In general, it is a description of any system or a detailed description of the system that is being proposed. It also goes by the name “stakeholder requirement”. It may include various types of diagrams, including Entity Relation Diagrams (ER Diagrams), Use Case Diagrams, Business Process Models (BPM), and Unified Modeling Language Diagrams (UML).

Requirement analysis is a set of techniques used to determine precise options, desires, and expectations through dialogue with system users. Obtaining raw data first, then interpreting it to meet needs. Finally, the data on consumer demands are revealed. It requires both hardware and software to work together.

I will use Use Case Diagrams, Logic Diagrams, and Data Flow diagrams to illustrate the required definition for our proposed system in this chapter.

3.2 Business Process Model

Business Processing Model, usually referred to as BPM, is a standard and based on some conventional flowchart approaches that give notations mostly graphical for expressing business processes in any development life cycle. The primary goal of this BPM is to assist process modeling for both technical users and business users by giving business users clear notations. All company stakeholders may easily comprehend the business process modeling.

Basically, I want to provide users with the knowledge they need to choose the right tree for their environment. That is the real reason I want to make our software unprofitable initially. However, we will use its commercial rationale afterward. A user under this framework is not required to pay anything. This application is regularly used by the user, who finds useful recommendations on the kinds of trees they should plant.

3.3 Hardware & Software Requirements for System

This application is basically a web-based as well as a smartphone-based system. So to use this system user needs to have a smartphone or computer with a digital wallet. A smartphone as well as an android phone with versions upper than 10, this system will run smoothly, and any user can find their expected results using this app. There are no persuasive justifications for any hardware or software requirements to use this system. So, this is the requirement regarding Hardware and Software for this system.

3.4 Use Case Model

The relationship between the user and the system must be evident in every software development system. A use case diagram is an easy way to illustrate this user's interaction with the system. It has the capacity to distinguish between various system user subtypes and usage scenarios. To put it another way, it is a Unified Modeling Language (UML) that generally describes the collection of actions that a system or systems should or can carry out in coordination with one or multiple external users of the system. In this case, the steps are referred to as "use cases" and the system is referred to as an "actor."

So, basically, it shows how different kinds of users will use the system to get their expected results or suggestion about trees. Moreover, it describes the user's target for using this application, the way users and the system interact, and the actions the system must take to achieve these objectives. In this chapter, I will use Use Case Diagram to illustrate this. In other words, it is a Unified Modeling Language (UML) that generally represents the group of tasks that a system or systems should or should complete in conjunction with one or more external users of the system. In this instance, the system is referred to as an "actor" and the activities are referred to as "use cases".

3.5 System Use Case:

It contains the following list of use cases in my system. Through this approach, people can select trees by offering some inputs that are primarily dependent on their immediate surroundings. Before using this app, users must first register. This app can only be used by the one registered user that logs in with their credentials. Without registering, a user cannot view anything. Users must provide details when registering or logging in. Some of these fields must be filled out in order to continue. Other fields marked as empty will

show as null. Information in apps will also be updated and deleted by the app administrator.

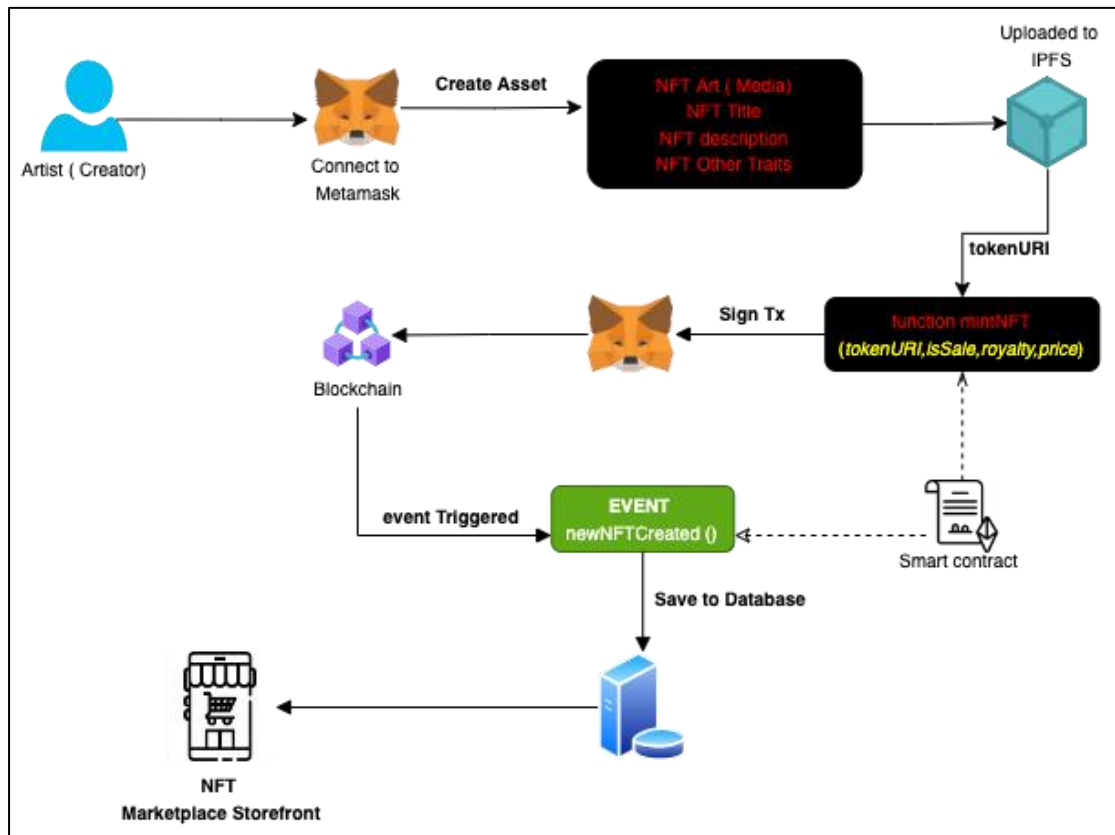


Figure 3.5.1: System Use Case for User

3.6 Data Flow Diagram

Here are our data flow diagram user requests from the device to the database, and the user gets the result

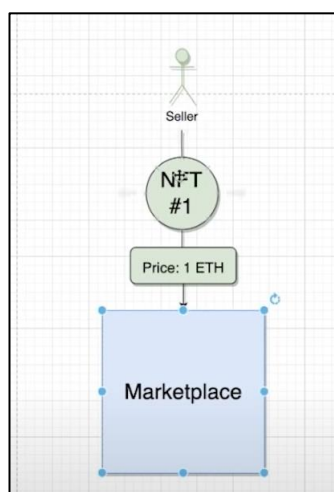


Figure 3.5.2: Data Flow for Seller

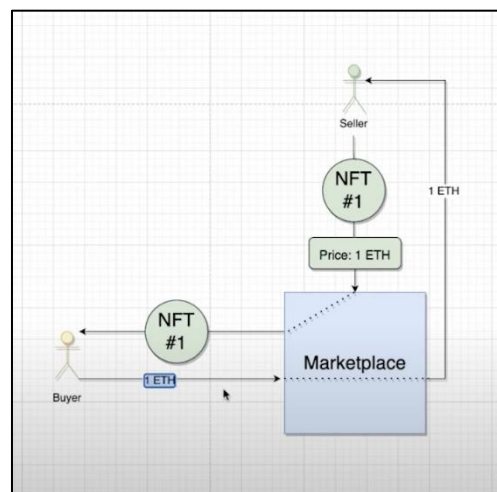


Figure 3.5.3: Data Flow for Seller and Buyer

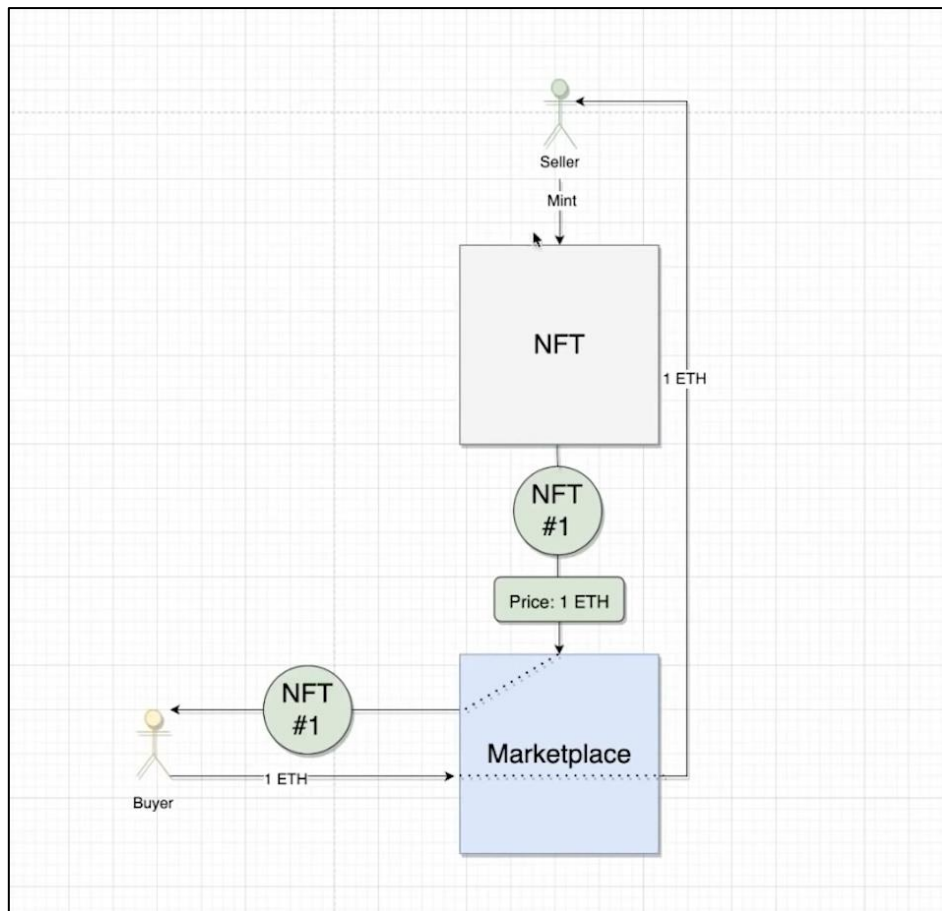


Figure 3.5.4: Data Flow Diagram

3.7 Implementation Requirements

The first step for starting the coding phase of this project is to choose the language in which the whole project will be done. In this project, I used Javascript [18] and Solidity [7] as the programming languages. I also used Next Js [20] for framework support. I used Web3 Storage for the IPFS and Goerli Testnet for blockchain purpose. Besides these, I have to use some libraries for some features to enhance the efficiency of my application. All of this coding is handled by an IDE called Visual Studio Code [16], which gives the developer excellent support for coding.

Ethereum:

Ethereum is a decentralized, open-source platform that utilizes blockchain technology to facilitate smart contracts and cryptocurrency transactions without the need for intermediaries. The cryptocurrency industry, being a relatively new sector, was kickstarted by the introduction of Bitcoin in 2009. Bitcoin served as a trial run for a

digital asset backed by a blockchain infrastructure. This covered the way for the development of other projects and digital assets, including Ethereum.

Ethereum's blockchain offers a wealth of opportunities for developers seeking to build solutions using the Ethereum platform. It has its own native cryptocurrency, known as Ether (ETH) is used to pay for various activities on the Ethereum network. ETH is traded on various cryptocurrency exchanges and its value fluctuates in the market. Other digital assets built on the Ethereum blockchain, such as ERC-20 tokens, also require ETH for transaction fees. The Ethereum blockchain was developed using the Solidity [7] programming language. The Ethereum Foundation, a non-profit organization, oversees the Ethereum project.



Figure 3.7.1: Ethereum Logo

While Bitcoin's origin remains shrouded in mystery, the creation of Ethereum has a clearer history. Ethereum was co-founded by Vitalik Buterin and a group of individuals, but the backstory of this massive blockchain deserves a more in-depth explanation.

The Ethereum blockchain operates through a consensus mechanism based on proof-of-stake. To append new blocks to the Ethereum blockchain, one must have a minimum of 32 ETH staked in the deposit contract and run the validator software. They can then be chosen randomly to present blocks that are verified by other validators and added to the Ethereum blockchain. In this model, the occurrence of only one chain is common, however, the existence of multiple blocks at the same position near the head of the chain can result from network latency or dishonest behavior. To resolve this issue, a fork-choice algorithm selects the set of blocks that are considered to be the official ones. This selection process is based on the blocks that form the most substantial possible chain, taking into account the number of validators that have endorsed the blocks and

the amount of ETH they have staked. A system of rewards and penalties strongly incentivizes participants, to be honest, and online as much as possible.

Ethereum is a blockchain that incorporates a computer within it. It serves as a foundation for the development of decentralized, uncensored, and permission-free applications and organizations.

In the Ethereum ecosystem, there is a shared, standard computer called the Ethereum Virtual Machine (EVM) that all participants in the Ethereum network have a common understanding of. Every node in the Ethereum network holds a copy of the EVM's state. Furthermore, any participant can broadcast a request for the EVM to perform a specific computation. Upon receiving such a request, other participants on the network verify, validate, and execute the computation, resulting in a change in the EVM's state that is committed and broadcasted to the entire network [3].

Requests for computation are referred to as transaction requests. The records of all transactions and the current state of the EVM are stored on the blockchain, which is maintained and agreed upon by all nodes in the network [3].

Cryptographic techniques ensure the immutability of transactions once they have been verified and added to the blockchain. These techniques also guarantee that all transactions are signed and executed with the proper authorization, ensuring that only the rightful owner can initiate transactions with their assets [3].

Testnet:

Using a blockchain to complete transactions involves paying gas fees and can take a while. Hence, building decentralized applications (dapps) and Web3 projects on a mainnet right away isn't practical due to the high cost. Testnets come into play to offer a smoother experience for the development.

A testnet is a duplicate of an original blockchain network that developers can use to experiment with and test network configurations prior to deployment and release of Web3 projects. It offers a convenient way for developers to design, develop, and assess the performance of their projects before making them available to the public. In effect,

a testnet serves as a "sandbox" environment where projects can be pre-released on-chain, allowing developers to evaluate public interest and engagement. [4].

It's necessary to understand transactions on testnets still require payment. But, unlike mainnets, testnet tokens don't hold any significant value. Therefore, you can obtain tokens for transactions through faucets, which are usually web apps where you input your wallet address and receive free testnet tokens.

Many blockchains have their own testnets, as mentioned earlier. Some networks even have multiple testnets. For example, here are some well-known Ethereum testnets:

- Kovan
- Goerli
- Sepolia
- Rinkeby
- Ropsten

However, Kovan, Rinkeby, and Ropsten have been inactive for some time, meaning that the Ethereum Foundation no longer supports these networks.

Goerli Testnet:

Testnets are one of the most valuable features in the Web3 industry, offering developers a safe and secure platform to test their projects before launching on a mainnet. The Ethereum ecosystem has witnessed the rise and fall of multiple networks, but currently, two major public testnets are maintained by the community, one of which is Goerli (Görli). It is a cross-client proof-of-stake (PoS) network and one of the most widely used testnets in Ethereum.

Goerli, also referred to as Görli, was launched in September 2018 and is one of the most prominent testnets of Ethereum. Additionally, it is one of two main testnets - the other being Sepolia - that have not been discontinued. The Goerli testnet originated as a project from the ETHBerlin hackathon, but it wasn't until early 2019 that it officially debuted at GörliCon in Berlin.

The Merge, a seminal event in Ethereum, likely did not escape your notice. It involved the integration of the original execution layer with the newer proof-of-stake (PoS)

consensus layer, Beacon Chain, marking a transition for Ethereum from its previous proof-of-work (PoW) consensus mechanism to PoS. The Merge also involved Goerli's transition from PoA to PoS, which took place in two stages in August of the same year. The first phase involved the implementation of the Bellatrix upgrade, followed by the second phase, where Goerli was merged with the Prater Beacon Chain's testnet, a proof-of-stake network operating in parallel with the Ethereum mainnet [4].

Goerli's integration and switch to proof-of-stake provided a final chance for developers to verify the correct configuration of their PoS validators prior to the mainnet's transition. It served as the last trial run before the official merging of the Ethereum mainnet with the Beacon Chain [4].

Goerli was established in 2019 and is one of two public Ethereum PoS testnets that are supported by the community, the other being Sepolia. It offers cross-client compatibility, enabling developers to use different client options for various programming languages in their Ethereum development projects.

Furthermore, testnets play a crucial role in blockchain development as they offer a secure platform for developers to experiment with their dapps and Web3 projects without risking their mainnet. With access to the Goerli testnet, developers can streamline their Ethereum development process, saving valuable time and resources [4].

Aside from testnets, these networks often include blockchain explorers, enhancing the ease of Web3 development. Blockchain explorers like Goerli Etherscan offer user-friendly interfaces to retrieve information about transactions, blocks, addresses, fees, and more. Hence, if you're interested in Ethereum development, consider Goerli Etherscan. It streamlines the development process by offering a smoother experience.

Web3 Storage:

Protocol Labs has introduced Web3 Storage, a convenient interface for developers to access and retrieve data from Filecoin's decentralized storage network. Web3.Storage, which will be available for free permanently, provides developers with a simple solution for creating applications that use decentralized and redundant storage, as well as secure content addressed data. By reducing the complexity of direct interaction with

decentralized storage, Web3 Storage makes it easy for all types of developers, including both those new to Web3 and experienced IPFS users, to integrate with Filecoin storage [5].

Web3 Storage contains mainly two components:

- A data storage solution that spreads data across multiple Filecoin miners and IPFS network, ensuring data availability and retrieval using Content Identifier (CID) [5]
- An HTTP endpoint, JavaScript [18] client library, and web UI for interacting with the service [5]

Decentralized storage plays a crucial role in the Web3 era, although it doesn't necessarily require the use of blockchain technology. Despite the central role that blockchain plays in the creation of Web3, the goal of Web3 storage solutions is to achieve the highest level of decentralization possible. This eliminates the need for a central authority and single point of failure, ensuring reliable options for data storage [6].

To understand Web3 storage, it's important to have an understanding of what Web3 is. Web3 refers to the latest evolution of the internet, which is built upon blockchain technology and cryptocurrency principles. This evolution has progressed from the initial read-only experience of Web1 to the read-write experience of Web2, which is the current internet generation widely used today, to the read-write-own experience of Web3. People in the crypto industry already interact with various Web3 features on a daily basis [6].

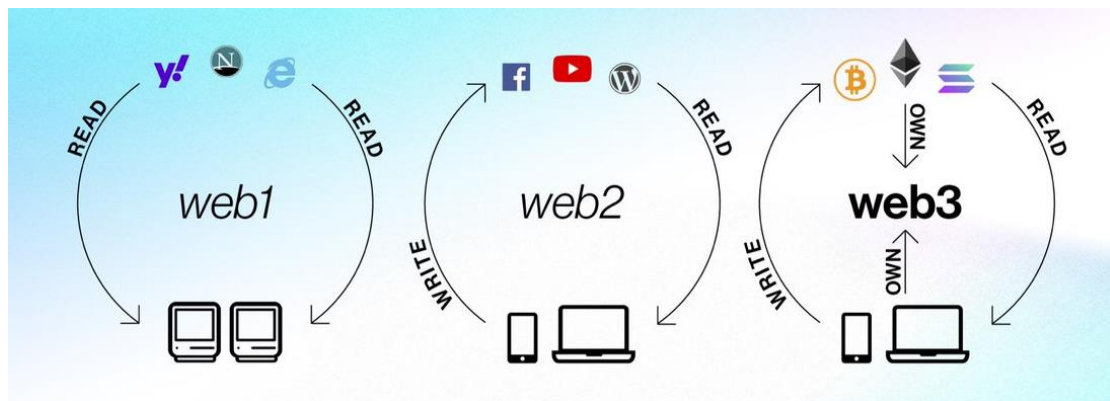


Figure 3.7.2: Web Concepts

The "own" aspect of Web3 highlights the limitations of centralized Web2 storage solutions like Google Cloud, OneDrive, Dropbox, etc. It is uncertain whether third parties are protecting or misusing your data, requiring a significant amount of trust from users. However, trustless operation is a core principle of Web3, making decentralized storage solutions necessary for true data ownership [6].

Additionally, there's a concern of a single point of failure in centralized cloud storage solutions. This means that users may be unable to retrieve their data during network disruptions and can even lose their data if the company's servers fail. Moreover, even personal storage devices like computer hard drives, external disks, and phone memories carry the risk of a single point of failure.

So, Web3 storage is a type of decentralized storage that ensures users have full ownership of their data and eliminates the risk of a single point of failure.

Ethers:

The cryptocurrency of Ethereum, Ether (ETH), serves as a means to incentivize participants to verify and process transactions and contribute computational resources to the network by creating a market for computation [3].

In Ethereum, Ether (ETH) is the native cryptocurrency that serves as a reward for participants who verify and execute transaction requests by providing computational resources to the network. To broadcast a transaction request, a participant must offer a bounty of ETH to incentivize network participants to perform the necessary work, such as verifying the transaction, executing it, and committing it to the blockchain. The

amount of ETH paid as a bounty reflects the computational resources required. This system also deters malicious participants from congesting the network with resource-intensive scripts, as they must pay for the computation resources they consume. [3].

The native cryptocurrency of Ethereum, ETH, serves three main purposes in ensuring the security of the network through crypto-economics:

- 1) ETH serves as a reward for validators who propose blocks or identify dishonest behavior among other validators;
- 2) ETH is used as collateral by validators to ensure honest behavior, as attempting to misbehave may result in the destruction of the staked ETH;
- 3) ETH is utilized to give weight to 'votes' for newly suggested blocks, influencing the consensus mechanism's fork-choice decision.

Smart Contracts:

Instead of writing new code each time, participants in the Ethereum Virtual Machine (EVM) request computations by calling upon pre-existing programs, referred to as smart contracts, which have been uploaded into the EVM state by application developers. These smart contracts are executed with specific parameters as requested by the users [3].

At its core, a smart contract can be thought of as a vending machine - a script that performs actions or computations based on specific parameters when called, as long as certain conditions are met. For instance, a basic smart contract for vending could create and transfer ownership of a digital asset when the caller sends ETH to a designated receiver.

Developers can create a smart contract and make it accessible to the network by using the blockchain as its data storage, for which a fee is paid to the network. Any user can then initiate the execution of the smart contract's code by paying a fee to the network [3].

Smart contracts allow developers to build and launch complex applications and services such as marketplaces, financial instruments, games, etc. by using the blockchain as its data layer.

Solidity:

Solidity [7] is an object-oriented programming language used for developing smart contracts on blockchain platforms, specifically Ethereum. It was created by Christian Reitwiessner, Alex Beregszaszi, and former Ethereum core contributors. Solidity-based programs run on the Ethereum Virtual Machine (EVM) [7].

Solidity [7] is a statically typed language specifically created for creating smart contracts that operate on the Ethereum Virtual Machine (EVM).

Solidity [7], an object-oriented programming language, was developed by Christian Reitwiessner, Alex Beregszaszi, and former Ethereum core contributors, and is used for implementing smart contracts on various blockchain platforms, with a focus on Ethereum. Its syntax is similar to ECMAScript, which will be familiar to web developers. Unlike ECMAScript, Solidity features static typing and variadic return types. It also includes advanced contract features, such as hierarchical mappings, structs, inheritance (including multiple inheritance with C3 linearization), and a documentation system for specifying user-centric descriptions of method-calls. The ABI (application binary interface) in Solidity enables multiple type-safe functions within a single contract.

MetaMask:

MetaMask is a digital cryptocurrency wallet utilized for interacting with the Ethereum blockchain. It provides users with access to their Ethereum wallet through either a browser extension or a mobile app, enabling them to engage with decentralized applications.

MetaMask was developed to fulfill the requirement for secure and user-friendly Ethereum-based websites. It primarily focuses on managing accounts and linking the user to the blockchain.

MetaMask provides users with multiple options for managing their accounts and keys, including the use of hardware wallets, while keeping the information separate from the website environment. This enhances security compared to the traditional method of storing user keys on a central server or local storage, which can be susceptible to large-scale security breaches.

The security feature of Ethereum API comes with developer convenience, making it easier for developers to interact with the globally available API. This API identifies the users of web3-compatible browsers such as MetaMask and whenever a transaction signature is requested (such as `eth_sendTransaction`, `eth_signTypedData`, or others), MetaMask will prompt the user in a clear and comprehensible manner. This helps to keep users informed and reduces the possibility of mass hacks by attackers, who can only attempt to phish individual users [8].

MetaMask has built-in fast connections to the Ethereum blockchain and several test networks through Infura, providing a seamless start without the need to synchronize a full node. However, users have the option to upgrade their security by selecting their preferred blockchain provider [8].

MetaMask currently supports any blockchain that has an Ethereum-compatible JSON RPC API, including private and custom blockchains. For development purposes, it's advisable to use a test blockchain like Ganache.

We acknowledge that new private blockchains are constantly emerging. If you're interested in integrating MetaMask with your own custom network, we offer an API that has already helped several networks achieve this integration.

JavaScript:

JavaScript [18], often referred to as JS, is a crucial technology of the World Wide Web, along with HTML [17] and CSS [19]. In 2022, it was utilized on 98% of websites for client-side webpage functionality and often integrated with third-party libraries. As the world's most widely used programming language, JavaScript is considered the programming language of the Web and is simple to learn. This tutorial intends to provide comprehensive knowledge of JavaScript, starting from fundamental concepts and proceeding to more advanced topics [9].

JavaScript [18] is a versatile, interpreted, or compiled-on-the-fly language with top-notch functions. Although it is widely recognized for being the scripting language of websites, it is also utilized in various non-web settings like Node.js, Apache CouchDB, and Adobe Acrobat. It is a multi-approach, single-threaded, and dynamic language that

encompasses object-oriented, imperative, and declarative (such as functional programming) programming styles, and utilizes a prototype-based approach. [10]

JavaScript [18] offers dynamic features such as constructing objects during runtime, accepting variables in function parameters, creating scripts dynamically, examining objects, and retrieving source code (JavaScript [18] functions maintain their source text which can be accessed using `toString()`).

Next JS:

Next.js [20] is a freely open-source for web development, developed by Vercel, which enables the creation of web applications using React and featuring server-side rendering and generating static websites. The React documentation lists Next.js [20] as one of its "Recommended Toolchains" and recommends it to developers when building a server-rendered website with Node.js. Unlike traditional React apps which only render content in the client-side browser, Next.js [20] expands this capability by allowing for applications to also be rendered on the server-side.

The copyright and trademarks for Next.js are owned by Vercel which also maintains and leads its open-source development.

Next.js [20] is a popular React framework that offers extra features like server-side rendering and static website generation. Unlike traditional React apps, which only render content in the client-side browser, Next.js [20] extends this capability to include server-side rendering. This solves several problems associated with client-side rendering, such as poor accessibility for users without JavaScript or with it disabled, potential security issues, slow page loading times, and reduced search engine optimization. Next.js is recommended as a "toolchain" for building server-rendered websites with Node.js and can be easily set up using Node Package Manager.

Google has made significant contributions to the Next.js project, submitting 43 pull requests in 2019. These contributions helped to optimize the framework by removing unused JavaScript, reducing loading times, and improving metrics. By March 2022, many well-known websites, such as Walmart, Apple, Nike, Netflix, TikTok, Uber, Lyft, and Starbucks, have adopted the framework. In early 2020, Vercel, the company behind Next.js, received a Series A funding of twenty-one million dollars to further

improve the software. Guillermo Rauch, the framework's original creator, is now the CEO of Vercel and the project is led by Tim Neutkens as its main maintainer [12].

Tailwind CSS:

Tailwind CSS is an open-source CSS framework that offers a different approach compared to other CSS frameworks such as Bootstrap. It doesn't provide predefined classes for elements like buttons or tables. Instead, it provides a collection of utility CSS classes that can be combined to style elements in a unique way.

For instance, in conventional systems, a class of "message-warning" would result in a yellow background color and bold text. However, to attain the same effect using Tailwind, you'd need to apply a combination of its specific classes, namely "bg-yellow-200" and "font-bold". [13]

It is crucial to understand the underlying philosophy behind Tailwind, as it differs from traditional CSS frameworks like Bootstrap in terms of basic concepts and its basic usage.

Tailwind operates in a mode where it generates all possible CSS combinations based on the project's settings. To optimize this output, another utility such as PurgeCSS is used to eliminate any unused classes from the final CSS.

Tailwind CSS, an open-source CSS framework, used to have the drawback of generating long waiting times and large CSS file sizes due to the vast number of classes generated from combining the various variants [13]. However, this mode of operation is no longer available in Tailwind CSS version 3.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

Front-end design is the design of client sites which will be visible to the clients of our project. In front-end design, we need to think about both user-friendly user interfaces and logically correct operations. In our project, API do the main work and transfers the data from the backend to the frontend. Through Next Js framework we have designed this overall easy-to-use and simple and nice user interface with the addition of several essential elements like buttons, icons, and text area.

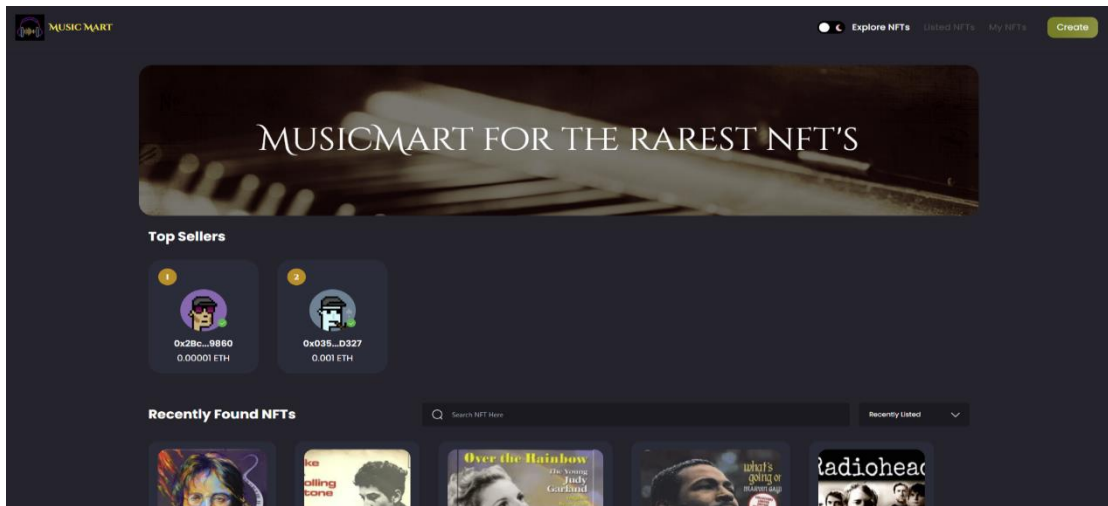


Figure 4.1.1: Home Page Top

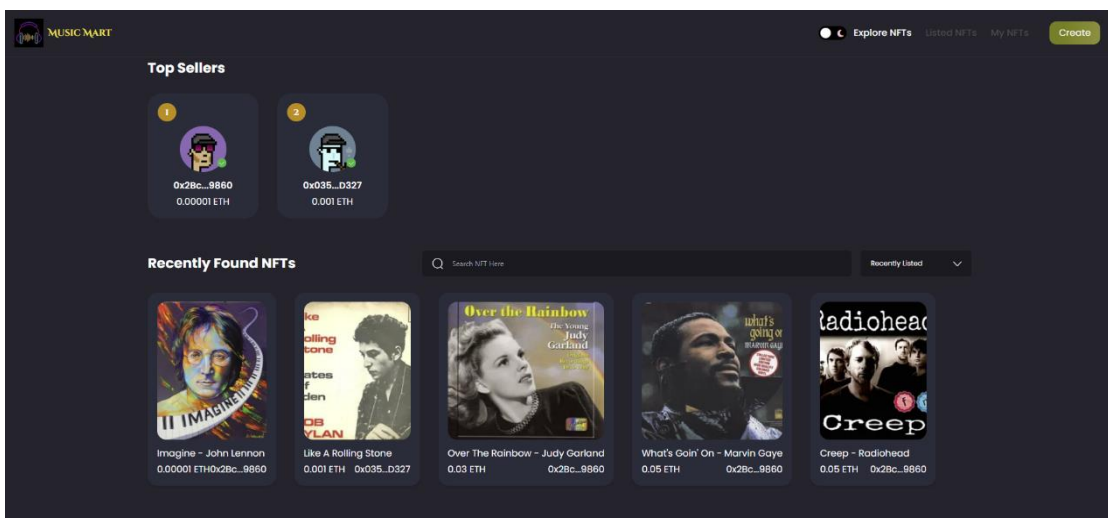


Figure 4.1.2: Home Page Bottom

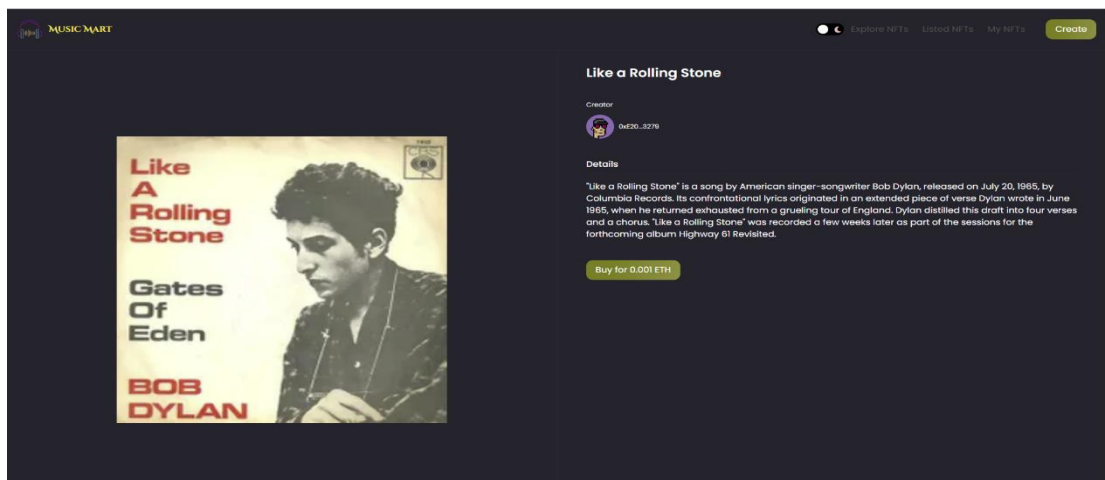


Figure 4.1.3: Individual NFT Page Sample-1

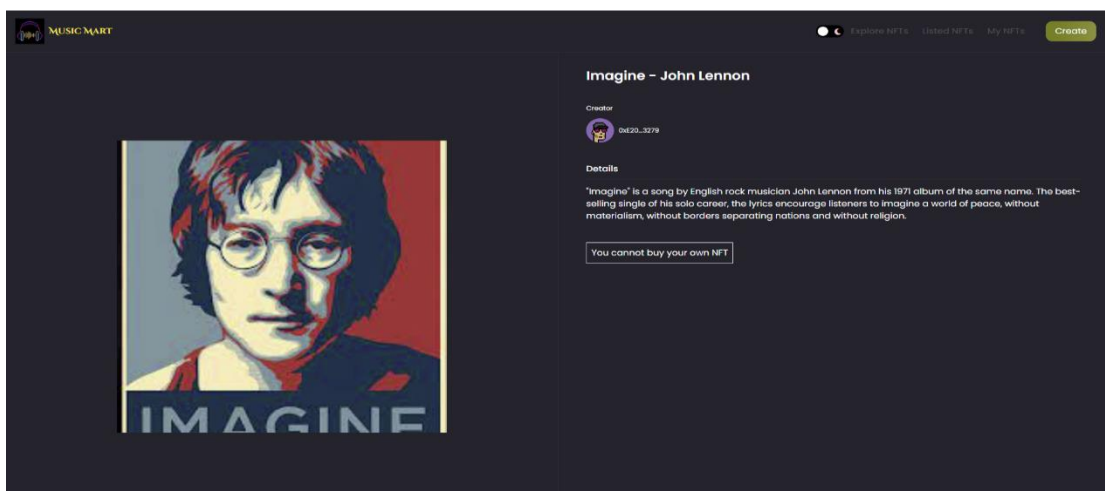


Figure 4.1.4: Individual NFT Page Sample-2

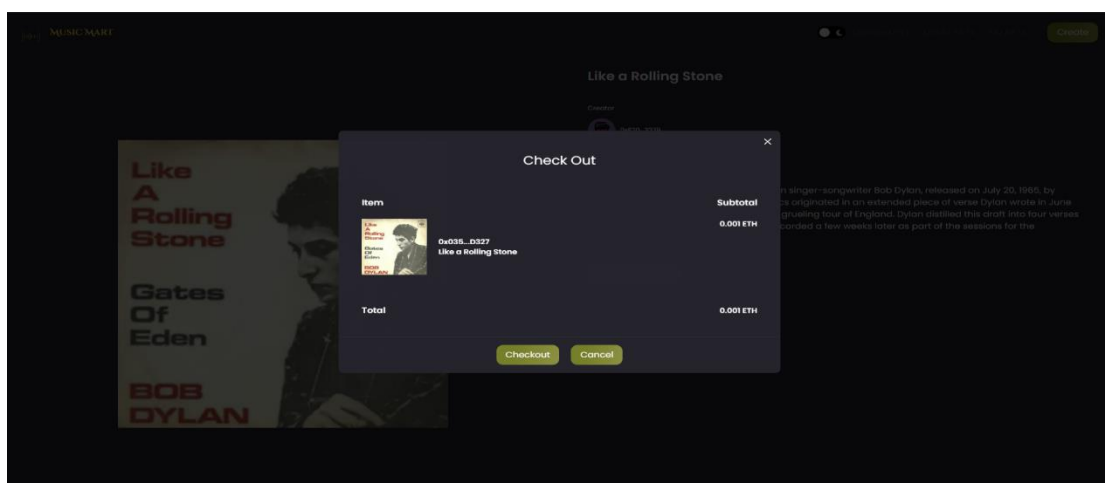


Figure 4.1.5: Buying/ Selling NFT Page

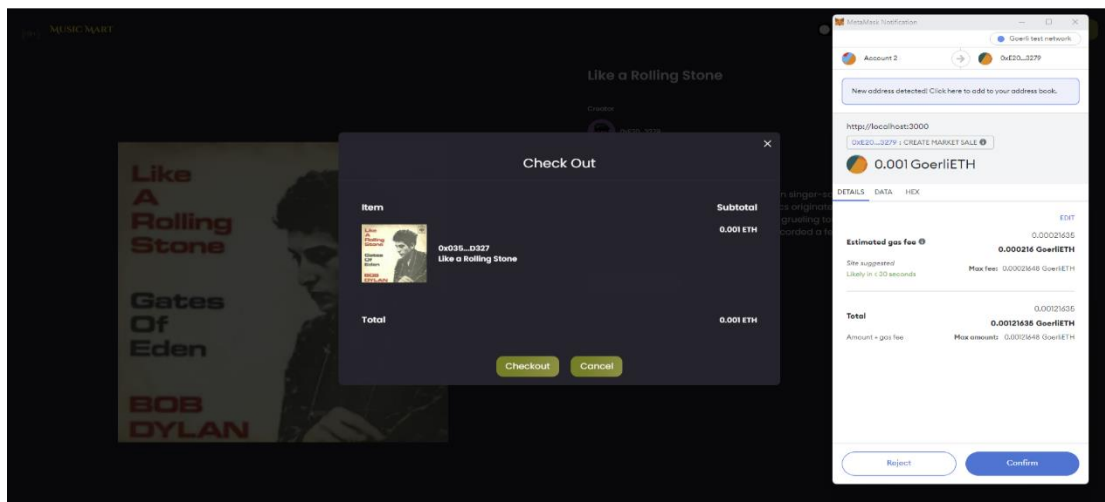


Figure 4.1.6: Checkout Page

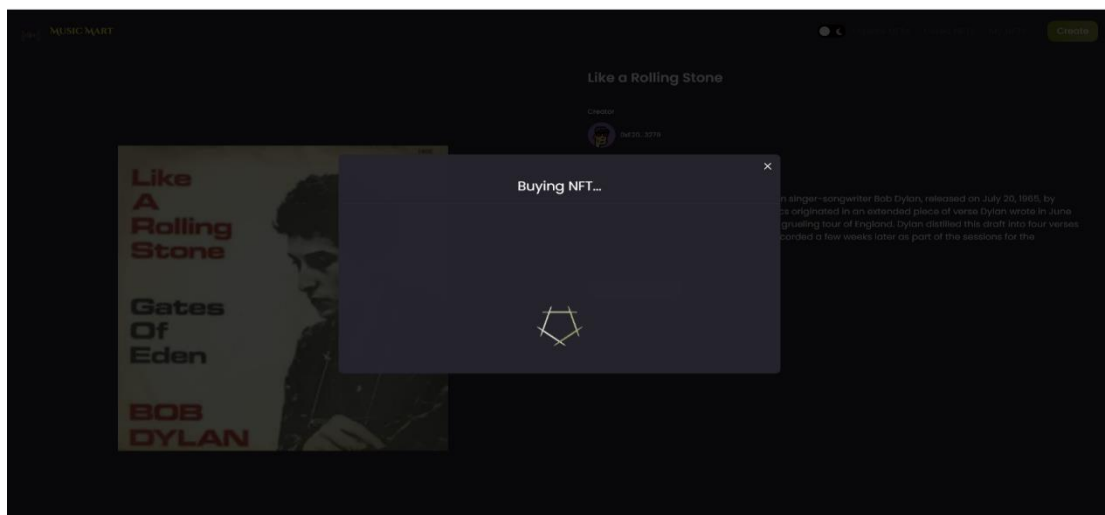


Figure 4.1.7: In Progress of Buying an NFT

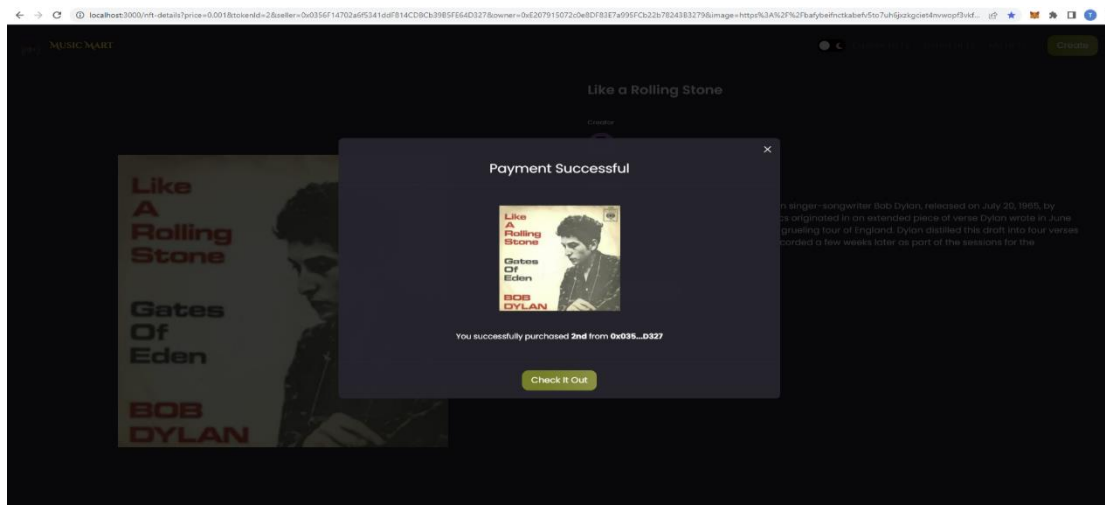


Figure 4.1.8: Completion of Buying an NFT

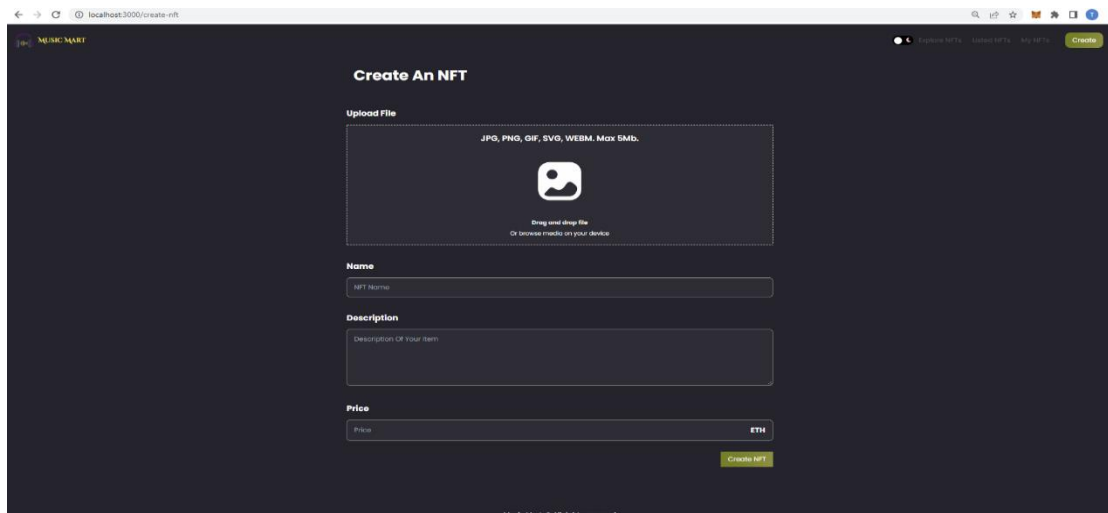


Figure 4.1.9: NFT Creation Form

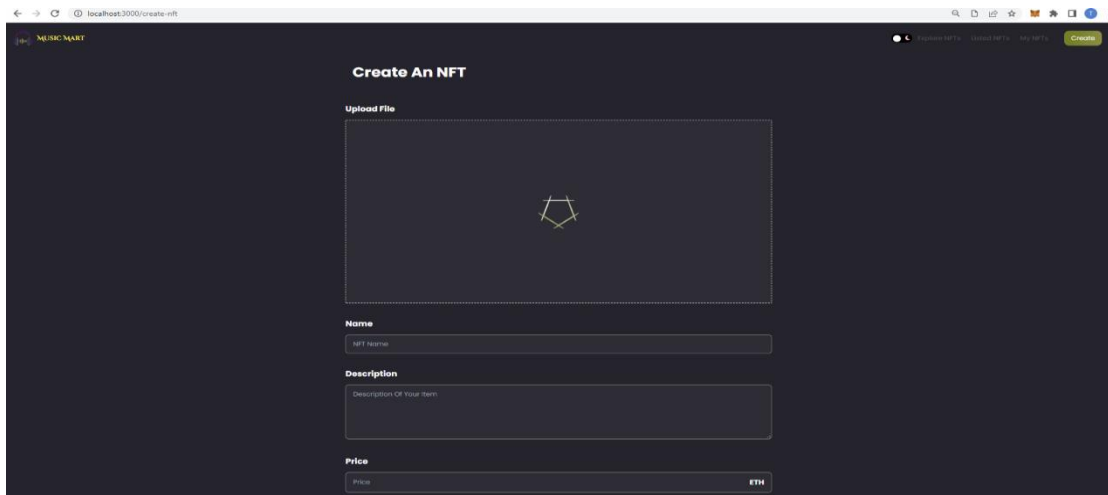


Figure 4.1.10: Uploading File in NFT Page

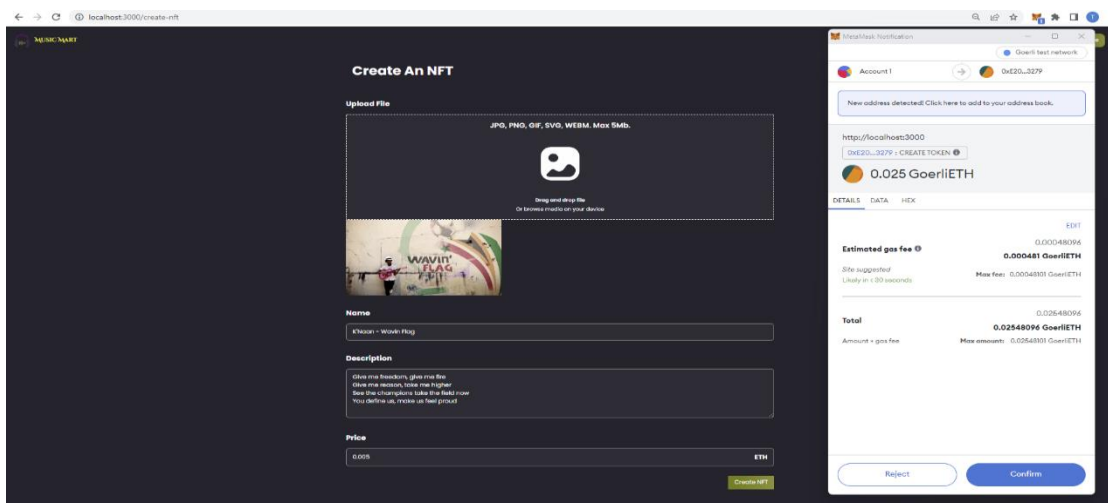


Figure 4.1.11: Payment Page

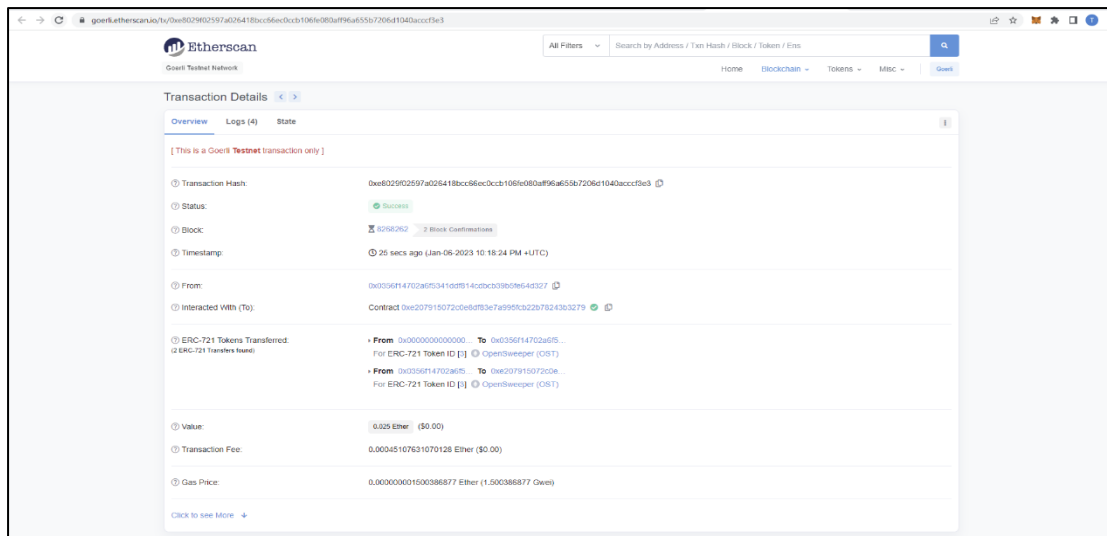


Figure 4.1.12: Transaction Details Page

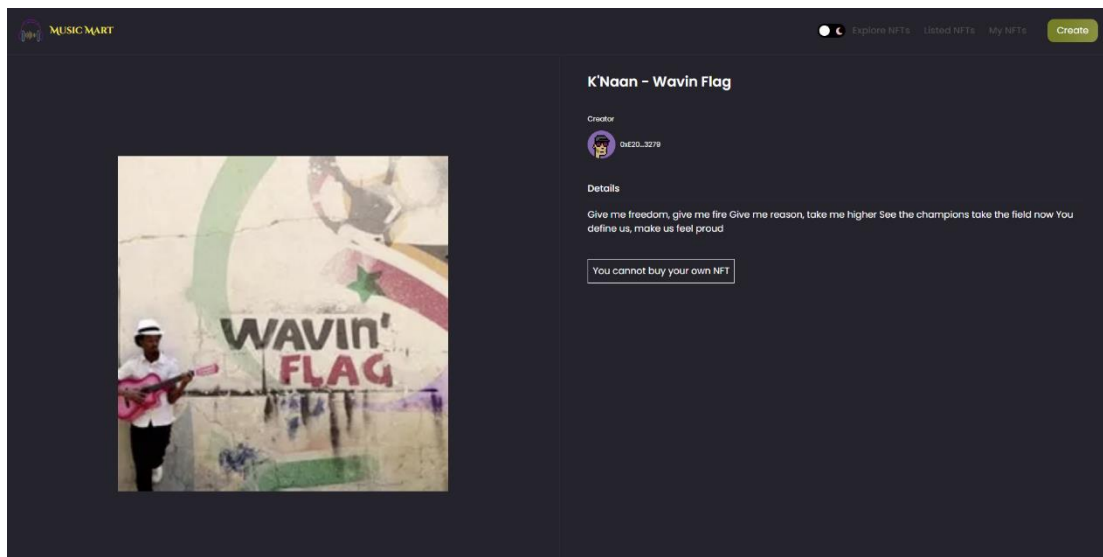


Figure 4.1.13: After Creating an NFT Page

4.2 Back-end Design

Music Mart is a Web app; hence we need a robust backend for operations. The database is the most important component since all tasks are conducted in accordance with the database. For the Music Mart, we utilized web3 storage for storing the data of files. Web3 storage was chosen because of its speed, dependability, availability, scalability, and security. I have used Ethereum (Goerli) test network for blockchain purposes. For the authentication and transactions, I have used MetaMask wallet.

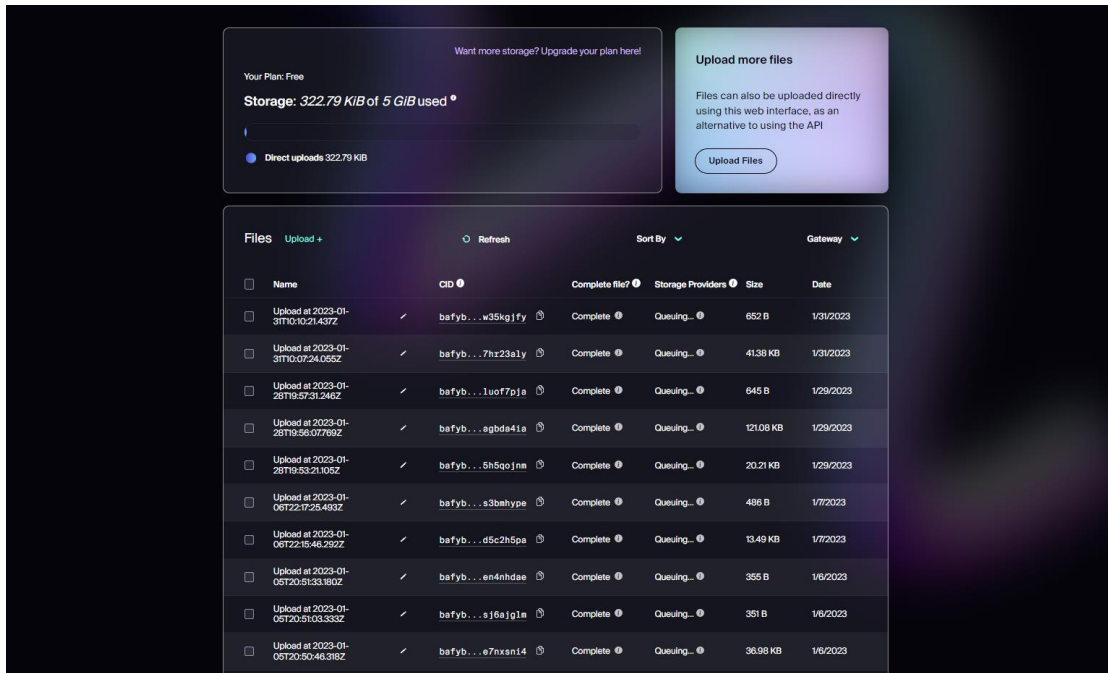


Figure 4.2.1: Web3 Storage

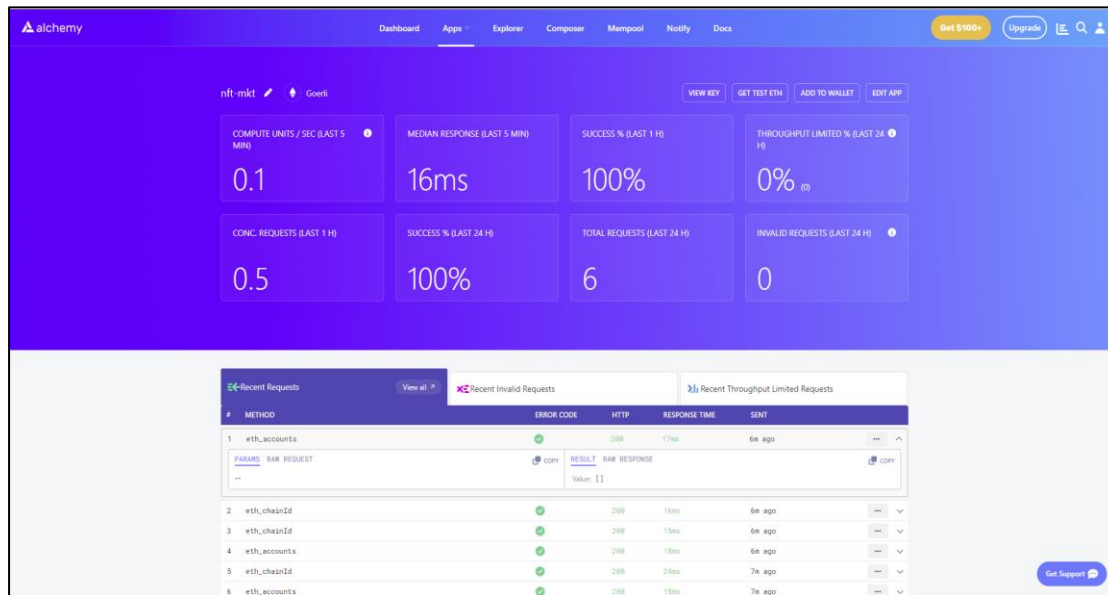


Figure 4.2.2: Ethereum (Goerli) Test Network

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database:

In this project, I have used IPFS to control and store data. Web3 Storage is a great platform for IPFS persistence.

IPFS

The InterPlanetary File System or IPFS is a protocol, hypermedia and file-sharing which allows for faster and more efficient file transfer by breaking a file into smaller pieces and distributing them across multiple nodes. This allows for faster download speeds as well as the ability to retrieve files even if a specific node is unavailable. IPFS utilizes a distributed hash table for indexing and locating files, making it more resilient to network outages and censorship. Overall, IPFS is a decentralized and distributed system that provides a more efficient and reliable way of storing and sharing data. In contrast to HTTP, which downloads files from a single server, the peer-to-peer IPFS protocol retrieves files by utilizing multiple nodes simultaneously, resulting in significant savings in terms of bandwidth usage.

5.2 Implementation of Blockchain Network:

Goerli, also known as Görli, is one of two primary testnets – along with Sepolia – that have not been deprecated.

Clients play a crucial role in the Ethereum ecosystem, providing developers with the ability to communicate with the Ethereum network through various programming languages. In essence, a client acts as a bridge between nodes and the Ethereum network [4].

The Ethereum Foundation maintains various clients for languages, including C#, Java, Rust, Go, etc. Moreover, several third-party clients fill that role if the foundation does not fully support a programming language. Nevertheless, you can find three prominent Ethereum client examples below:

- Geth (Go)
- Parity (Rust)

- Nethermind (C#, .NET)

While the Ethereum mainnet has multiple client support, the same can't be said for other major testnets. Historically, Rinkeby and Kovan were two of the largest testnets, with Rinkeby supporting Geth and Kovan supporting Parity, leading to a lack of inter-client compatibility. However, with the advent of Goerli, this issue has been addressed, providing a unified client experience.

Goerli stands out for its cross-client compatibility, allowing for a wider range of node software to be supported on the testnet. This provides a unified testing environment for different clients, including Geth, Parity Hyperledger, Nethermind, and others, to function together seamlessly.

Goerli's cross-client compatibility enhances the ease and simplicity of Ethereum development by enabling developers to utilize the same tools across different programming languages, including for the testnet [4].

5.3 Implementation of Front-end Design

Front-end design is very important for any application. There can be multiple techniques to do it. Here in our project, we have used HTML [17], CSS [19], JavaScript [18] and Next JS framework for design. We have designed our user interface through the HTML [17] and JavaScript [18] file with required icons, buttons, text edit section, navigation bar, different shapes of image view like circular, square, different backgrounds, and layout. As per the user's request, data comes from back-end to front-end through API. Besides, our front-end is connected with web3 storage for database purposes and the Goerli test network for blockchain purposes.

5.4 Implementation of Interaction

For a better user experience, I implemented the entire interactive UI in our suggested system. These interfaces' screenshots have already been provided. I have occasionally utilized icons in place of text, links, or buttons. The entire interface will be easy for the user to utilize.

5.5 Testing Implementation

Before releasing any system, it needs to go through a testing phase. It is a good practice as well as a welfare for the system. Because after release, if the system faces some difficulty that could not be handled before or if the system does not provide the expected output that a user expects, this will become useless. System testing is a kind of programming testing where a whole incorporates programming is taking place. So the biggest inspiration after this test is to access the framework consistent with the predefined necessities.

Testing Approaches:

Depending on the two methodologies test can be directed in two ways -

- Functional testing
- Implementation testing

Discovery testing is the process of testing usefulness at this point without considering actual usage. The reverse is known as white-box testing, where usefulness is tested as well as how it is carried out is also looked into.

Unit testing

Unit testing is that kind of testing process where every unit of a system is tested through the testing process to ensure whether the system is working well or not, whether the user of the system is getting their expected output or not, or whether the system serves its making purpose properly or not. Basically, it encourages us to choose individual units of a system and test them to check if it is functioning according to prerequisite and are without blunder. Individual software units or components are tested under this sort of software testing. [15] The objective of unit testing is to verify that every code segment functions as intended. During the coding stage of application development, developers conduct unit testing. This process involves isolating individual pieces of code and evaluating their correctness. A unit can be a single function, method, procedure, module, or object.

Unit testing is a form of WhiteBox testing that takes place prior to integrated testing in software development life cycle (SDLC), software testing life cycle (STLC), and V Models. Typically, it is performed by the developer, but in practicality, QA engineers may also conduct unit testing due to time constraints or developer reluctance to do so.

Framework Testing

Any efficient automated testing process must incorporate testing frameworks. [14] They can cut maintenance expenses and testing time and offer QA teams aiming to improve their agile procedures a higher return on investment (ROI). It serves as a blueprint for the test development process, providing a structured approach to testing and ensuring that all necessary tests are performed. This includes best practices for test case design, test data management, and test results reporting. It also includes integration with testing tools and automation frameworks, which can further streamline the testing process. QA professionals can use a testing framework to ensure that the testing process is thorough and consistent, leading to a higher quality product. Additionally, a testing framework can provide guidelines for handling test dependencies, such as identifying and managing external systems or services that are required for testing. It can also provide error and exception handling guidelines, ensuring that test cases are robust and can handle unexpected scenarios. Furthermore, a testing framework can provide guidance on test documentation, such as test case specifications, test plans and test reports, to ensure that all the necessary information is captured and easily accessible for future reference. Overall, a testing framework is a comprehensive tool that helps ensure the testing process's quality and efficiency.

Although following a structured framework for scripting or recording tests is not mandatory, utilizing an organized approach typically brings additional benefits that would otherwise go unachieved.

5.6 Test Results and Reports

Numerous testing methods are used in the real-time software development life cycle. But I used two testing approaches to evaluate our system. Unit testing and integrated testing were those two systems. Unit testing comes first in the coding section. I tested my various modules using unit testing where I made them. I tested the entire system using an integration testing system after finishing it. I encountered an issue in each test part, which I then resolved. Then I finish the system I proposed.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 Conclusion

Choosing and completing this project as my final year project is an excellent choice for me. Because it boosted me in many aspects of my development life, it is the first complete project I have ever made, so it boosted my confidence to do something big. On the other hand, I learned a lot of new things while doing this project. At first, I made the UI design then I dived into the coding part of this project. So, to code the system following the UI design, I had to learn many new things I didn't know. I also faced a lot of challenges while doing this project, but I overcame this. For this, I would like to thank the Almighty ALLAH, without his mercy, I would not be able to do such a project alone.

Then special thanks to my project supervisor, Md Azharul Islam Tazib, who guided me on the right track throughout this project. Without his consultation and supervision, it would be quite difficult for me to complete this project.

As I have mentioned before, I have gone through many difficulties and challenges while making this project, and from these, I have learned a lot of new things - both technical and ethical, and professional. When I started working on this project three months ago, there was much knowledge I recovered and gained while working on this project.

Though my application has some limitations, I can assure you that it has some pretty good and efficient features that will give a user inner peace. Maybe this project is complete for this final year project, but it is not ending here. I will provide some improvements regarding features and updates to this project. So, it can be said that this is the first step toward my great vision of what I am working for.

6.2 Future Scope

- Direct upload music, sound, track
- Speed up the transaction time
- More customization
- User feedback

References

- [1] "NFTs in the art world: how they are revolutionizing the way we buy and sell art" available at <https://www.linkedin.com/pulse/nfts-art-world-how-revolutionizing-way-we-buy-sell-adesuyi-victor> [Last accessed: 26-10-2022 3.00 AM]
- [2] "Static vs Dynamic NFTs: Complete Guide." available at <https://www.coinspeaker.com/guides/static-vs-dynamic-nfts-complete-guide/> [Last accessed: 26-10-2022 3.00 AM]
- [3] "Intro to Ethereum" available at <https://ethereum.org/en/developers/docs/intro-to-ethereum/> [Last accessed: 16-11-2022 2.00 AM]
- [4] "Goerli ETH - What is the Goerli Testnet?" available at <https://moralis.io/goerli-eth-what-is-the-goerli-testnet/> [Last accessed: 16-11-2022 2.00 AM]
- [5] "Introducing Web3.Storage: A simple interface for Filecoin storage." available at <https://filecoin.io/blog/posts/introducing-web3-storage/> [Last accessed: 18-11-2022 3.00 AM]
- [6] "Web3 Storage - How Web3 Data Storage Works | Moralis Web3." available at <https://moralis.io/web3-storage-how-web3-data-storage-works/> [Last accessed: 18-11-2022 3.00 AM]
- [7] "Solidity - Wikipedia." available at <https://en.wikipedia.org/wiki/Solidity> [Last accessed: 20-11-2022 3.00 AM]
- [8] "Introduction | MetaMask Docs." available at <https://docs.metamask.io/guide/> [Last accessed: 20-11-2022 3.00 AM]
- [9] "JavaScript Tutorial - W3Schools." available at <https://www.w3schools.com/Js/> [Last accessed: 26-11-2022 2.00 AM]
- [10] "Parallel and Asynchronous Community workgroup - GitHub." available at <https://github.com/Parallel-and-Asynchronous-Community/workgroup> [Last accessed: 26-11-2022 2.00 AM]
- [11] "Do I need SSR? Explained by Sharing Culture." available at <https://traey.afphila.com/do-i-need-ssr> [Last accessed: 27-11-2022 1.00 AM]
- [12] "Web Development Tools | knfrmd - Tips on how to build a better web." available at <https://www.knfrmd.com/posts/web-dev-tools> [Last accessed: 28-11-2022 3.00 AM]
- [13] "Tailwind CSS - Wikipedia." available at https://en.wikipedia.org/wiki/Tailwind_CSS [Last accessed: 30-11-2022 2.00 AM]
- [14] Framework Testing [Online]. Available <https://smartbear.com/learn/automated-testing/test-automation-frameworks/> [Last accessed: 30-11-2022 2.00 AM]
- [15] Unit Testing [Online]. Available <https://www.guru99.com/unit-testing-guide.html> [Last accessed: 30-11-2022 2.00 AM]
- [16] "Visual Studio Code." available at https://en.wikipedia.org/wiki/Visual_Studio_Code [Last accessed: 06-12-2022 3.00 AM]
- [17] "HTML" available at <https://en.wikipedia.org/wiki/HTML5/> [Last accessed: 06-12-2022 3.00 AM]

[18] “JavaScript” available at <https://en.wikipedia.org/wiki/JavaScript/> [Last accessed: 06-12-2022 3.00 AM]

[19] “CSS” available at <https://en.wikipedia.org/wiki/CSS/> [Last accessed: 06-12-2022 3.00 AM]

[20] “Next.js” available at <https://en.wikipedia.org/wiki/Next.js> [Last accessed: 08-12-2022 2.00 AM]

Plagiarism Report

Turnitin Originality Report Document Viewer

Processed on: 07-Feb-2023 2:38 AM EST
ID: 2008373251
Word Count: 9556
Submitted: 2
191-15-12370 By Tahsin Iftekhar

Similarity Index	Similarity by Source
25%	Internet Sources: 22% Publications: 4% Student Papers: 13%

include quoted | include bibliography | exclude small matches | mode: quickview (classic) report | print | download

- 4% match (Internet from 14-Dec-2022)
<https://moralis.io/goerli-eth-what-is-the-goerli-testnet/>
- 2% match (student papers from 23-Jan-2021)
[Submitted to Daffodil International University on 2021-01-23](#)
- 1% match (Internet from 26-Oct-2022)
<http://dspace.daffodilvarsity.edu.bd:8080>
- 1% match (Internet from 26-Oct-2022)
<http://dspace.daffodilvarsity.edu.bd:8080>
- 1% match (Internet from 26-Oct-2022)
<http://dspace.daffodilvarsity.edu.bd:8080>
- 1% match (Internet from 06-Feb-2023)
<https://moralis.io/web3-storage-how-web3-data-storage-works/>
- 1% match (student papers from 05-Nov-2019)
[Submitted to Daffodil International University on 2019-11-05](#)
- 1% match (Internet from 01-Jan-2022)
["Next.js", Wikipedia, en, 2022](#)
- 1% match (Internet from 01-Jan-2022)
["Solidity", Wikipedia, en, 2022](#)
- 1% match (Internet from 30-Jan-2023)
<https://www.softermii.com/blog/nft-marketplace-development-guide>
- 1% match (Internet from 30-Jan-2023)
<https://www.scnsoft.com/blockchain/create-nft-marketplace>
- 1% match (Xun Yi, Xuechao Yang, Andrei Kelarev, Kwok Yan Lam, Zahir Tari. "Chapter 2 Bitcoin, Ethereum, Smart Contracts andBlockchain Types", Springer Science and Business Media LLC, 2022)
[Xun Yi, Xuechao Yang, Andrei Kelarev, Kwok Yan Lam, Zahir Tari. "Chapter 2 Bitcoin, Ethereum, Smart Contracts andBlockchain Types". Springer Science and Business Media LLC, 2022](#)
- 1% match (student papers from 20-Dec-2022)
[Submitted to Chandigarh University on 2022-12-20](#)
- <1% match (Internet from 02-Apr-2021)
<http://dspace.daffodilvarsity.edu.bd:8080>