MINOR PROJECT I

A MUSIC WEB APPLICATION BASED ON BLOCKCHAIN

MUSX

ODD SEMESTER

Group No. : 51

Submitted by:

KHUSHI KALRA (9920103025)

VAISHALI RANJAN (9920103013)

VREETI AGGARWAL (9920103007)

Under the Guidance of:

HIMANSHU AGRAWAL

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA (U.P.)

DEPARTMENT OF CSE/IT

DECEMBER, 2022

CERTIFICATE

This is to certify that the minor project report entitled, **“A WEB APPLICATION BASED ON BLOCKCHAIN FOR MUSIC COPYRIGHT PROTECTION – MUSX”** submitted by **VAISHALI RANJAN, KHUSHI KALRA AND VREETI AGGARWAL** of Bachelor of Technology Degree in **Computer Science and Engineering** of the **Jaypee Institute of Information Technology, Noida** has been carried out by them under my supervision and guidance. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor:

Name of the Supervisor: HIMANSHU AGRAWAL

Department of Computer Science and Engineering

Jaypee Institute of Information Technology, Sector-128, Noida- 201304

Date:

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: NOIDA

Date:

Name: VREETI AGGARWAL

Enrollment No.: 9920103007

Name: VAISHALI RANJAN

Enrollment No.: 9920103013

Name: KHUSHI KALRA

Enrollment No: 9920103025

ACKNOWLEGEMENT

We would like to place on record my deep sense of gratitude to **HIMANSHU AGRAWAL**, Assistant Professor, Jaypee Institute of Information Technology, India for his/her generous guidance, help, inspiration and constructive suggestions. New ideas and direction from him made it possible for us to sail through various areas of our project which further helped us to make it better.

We would also thank our institution and faculty members without whom this project would have been a distant reality.

We also wish to extend our thanks to seniors and other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

Name: VAISHALI RANJAN

Enrollment No.: 9920103013

Name: KHUSHI KALRA

Enrollment No: 9920103025

Name: VREETI AGGARWAL

Enrollment No.: 9920103007

ABSTRACT

The popularity of the Internet has accelerated the circulation of digital resources, and the problem of copyright management of digital resources has become prominent. By investigating the industry's supply chain, we conclude that on-demand streaming platforms (e.g. Spotify and Apple Music) have allowed consumers to easily access music products but have introduced a level of intermediation between artists and customers leading to inefficiency of the royalty payments systems.

The introduction of blockchain technology has a breakthrough meaning for the solution of digital resource copyright management. Blockchain is a public ledger to which everyone has access but without a central authority having control. It is an enabling technology for individuals and companies to collaborate with trust and transparency. Blockchain technology is considered to be the driving force of the next fundamental revolution in information technology. Many implementations of blockchain technology are widely available today, each having its particular strength for a specific application domain. This report explores the implementation of blockchain on the music industry with a focus on the implications technology can have for artists.

Based on the development of blockchain technology, we propose **MusX -** a blockchain-based decentralized music streaming platform to connect music enthusiasts directly to independent music artists. The artists can use this platform to share their music with greater freedom while ensuring ownership and avoiding duplication of their music. Audience can also support small artist by donating some amount to them. The goal of this project is to identify blockchain applications that would enable the disintermediation of the music industry, allowing artists to create and capture more value from their own products.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

IPFS	Interplanetary File System
DApp	Decentralized Application
XML	Extensible Markup Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
UI	User Interface

LIST OF FIGURES

Figure title **Page no.**

CHAPTER-1

INTRODUCTION

1.1 Web Programming

Web programming refers to the writing, mark-up and coding involved in Web development, which includes Web content, Web client and server scripting and network security. The most common languages used for Web programming are XML, HTML, JavaScript, and CSS. Web programming is different from just programming, which requires interdisciplinary knowledge on the application area, client and server scripting, and database technology.

Web programming can be briefly categorized into client and server coding. The client side needs programming related to accessing data from users and providing information. It also needs to ensure there are enough plugins to enrich user experience in a graphic user interface, including security measures. To improve user experience and related functionalities on the client side, JavaScript is usually used. It is an excellent client-side platform for designing and implementing Web applications.

HTML5 and CSS3 supports most of the client-side functionality provided by other application frameworks.

1.1.1 Overview

There are two broad divisions of web development – front-end development (also called Client-side development) and back-end development (also called server-side development). Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it. This is done with three codes – HTML, CSS and JavaScript. HTML, stands for HYPER TEXT MARKUP LANGUAGE, is a special code for 'marking up' text in order to turn it into a web page. Every web page on the net is written in HTML, and it is the backbone of every web application. CSS, short for Cascading Style Sheet, is a code for setting style rules for the appearance of web pages. CSS handles the cosmetic side 10 of the web. Finally, JavaScript is a scripting language that's widely used to add functionality and interactivity to web pages.

1.1.2 Description

React.js is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components. React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application.

1.2 Blockchain

Blockchain is a distributed database or ledger that is shared among the nodes of a computer network. It is a database of record of transactions which is distributed, and which is validated and maintained by a network of computers around the world. Instead of a single central authority such as a bank, the records are supervised by a large community and no individual person has control over it and no one can go back and change or erase a transaction history. As compared to a conventional centralized database, the information cannot be manipulated due to blockchain's built in distributed nature of structure and confirmed guarantees by the peers. In another words, when a normal centralized database is located on an individual server, blockchain is distributed among the users of a software. Blockchain technology is based on decentralized network meaning it operates as a peer to peer network.

1.2.1 Blockchain components

- Web3.js: It is an extensive collection of libraries that will help our web app to communicate with an ethereum blockchain using JSON based Remote Procedure Calls.
- IPFS (InterPlanetary File System): It is a distributed storage system that utilizes Peer-to-Peer (P2P) networking. Unlike traditional "location-based" storage systems, IPFS is "content-based" as it maintains unique global hashes of all the files. This gives it many advantages like independence from a single controlling entity and preventing duplicate files, thus being perfect for implementing our DApp.
- Dapp: A decentralized application uses a decentralized system to perform different tasks. Dapps help the user interact with the smart contract programs stored on a blockchain.
- Solidity: It is the high-level language used to write smart contracts that can be stored on the blockchain. This will define the logic behind our blockchain, dictating the behaviour of all the accounts.
- Ganache: It is used to create a personal local Ethereum blockchain, whose dummy accounts can be accessed by a cryptocurrency wallet and used in a Dapp. This helps us run our Dapp like it is connected to the actual Ethereum network without long transaction times.
- MetaMask: It is a crypto-wallet that can be accessed through a browser extension. We can use this to interact with our local blockchain created using Ganache.

- Truffle: It is the development environment that will be used to develop smart contracts and Dapps. It can be used for the compilation, testing as well as deployment of smart contracts.
- Solidity: It is a high-level programming language designed for implementing smart contracts.

1.2.2 Global advantage

- One of the biggest advantages of blockchain is dissemination which allows a database to be shared without a central body or entity. Because of the decentralized nature of the blockchain, it is almost impossible to temper the data as compared to conventional database.
- Since blockchain does not have any central point of failure due to its decentralized network, it can withstand any security attack.
- Blockchains provide transparency and immutability to the transactions as all the transactions cannot be altered or deleted.
- Blockchain’s peer-to-peer connections help to identify fraud activities in the network and distributed consensus. It is almost impossible to invade a network as attacker can impact the network only when they get control of 51% of the nodes.
- By using blockchain, sensitive business data can be protected using end to end encryption.

CHAPTER – 2
LITERATURE SURVEY

2.1 Literature Survey

Research Paper	**Year**	**Methodology**	**Pros**	**Cons**
The Impact of Blockchain on the Music Industry	2018	This paper studies the supply chain model of the current music industry and find that the streaming services have brought an intermediate level between artists and listeners.	They conclude that although these problems can be overcome using blockchain technology	This leads to problems like lack of transparency, very little bargaining power for the artists and an inefficient royalty payment system.
Blockchain-based Distributed Marketplace	2018	This paper describes a 8ecentralized Ethereum framework that uses the proof-of- concept system to implement a distributed online marketplace.	They determined that their costs were significantly less than existing online marketplaces – Amazon and eBay for a large volume of users.	They claim that the distribution of revenue to artists is an ongoing issue.
The blockchain technology on the music industry	2018	This article aims to analyze how the blockchain technology might emerge as distributed force in the music industry.	It focuses on what technological changes mean for the increasing and distributing music industry revenue.	Challenges for wide-spreading adoption may depend on blockchain technology to become "invisible", affecting the backend of music transaction and not the user experience.

Fig 2.1 Literature Survey

CHAPTER – 3

REQUIREMENT ANALYSIS

3.1 Problem statement

- **Third Party:** After the many deductions, only a small amount of money remains for the creators who are the original owners.
- **Small Artists:** Small emerging artists may not have enough resources to produce or invest in music.
- **Copyright Infringement:** World is facing major scrutiny when it comes to online copyright infringement in the music industry and has been bombarded with lawsuits because of their inability to compensate artists appropriately.

3.2 Solution to problem statement

- **No Third Party:** By using blockchain transaction, we'll be able to remove third parties, such as banks and music companies involved in the payment process, right holders can help them receive almost all their royalties.
- **Small Artists:** Our platform will provide them the opportunity to showcase their talent without worrying about the production cost since it directly connects their contents to the listener.
- **Copyright Policy:** Blockchain has an excellent potential to be broadly applied in copyright protection and management applications. When an user will download a song, a hash wil be generated with the song, and if the user tries to re-upload the song with a creator account, then the copyright issue will be raised.

3.3 Software and hardware requirements

- **Software requirements:**
  - Visual Code Studio (VS Code)
  - Web browser
  - Metamask
  - Ganache
- **Hardware requirements**
  - PC/System

3.4 VS Code

3.4.1 VS Code UI (User Interface)

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders one have access to, and an editor on the right, showing the content of the files one has opened. The UI is divided into five areas that is Editors, Side bar, Status bar, activity bar and Panels. Below figure 3.2 shows the features that are included in the VS code editor.

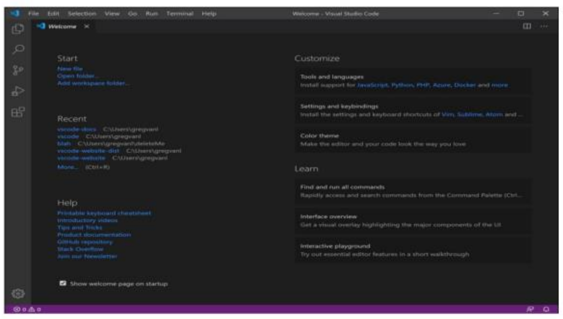


Fig 3.1 VS Code UI

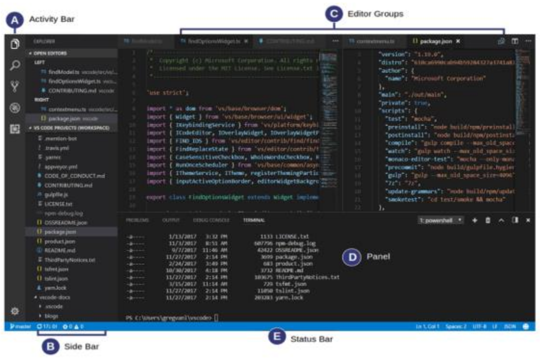


Fig 3.2 VS Code Features

CHAPTER – 4

DETAILED DESIGN

4.1 Architecture of the project:

We designed a tentative architecture of our Dapp. Here the user (artist or audience) publishes song/stream song trough front-end. Each user has their own metamask wallet which is connected through local ganache blockchain. Contracts are compiled and migrate with the help of truffle engine through local ganache blockchain. When the user publish a song the song hash is generated and stored in a decentralised storage and song is displayed on the artist's account as a song published by the artist as well as in the audience account to buy the song. The transaction is made through a local ganache blockchain and web3.js is used to connect the frontend with the blockchain.

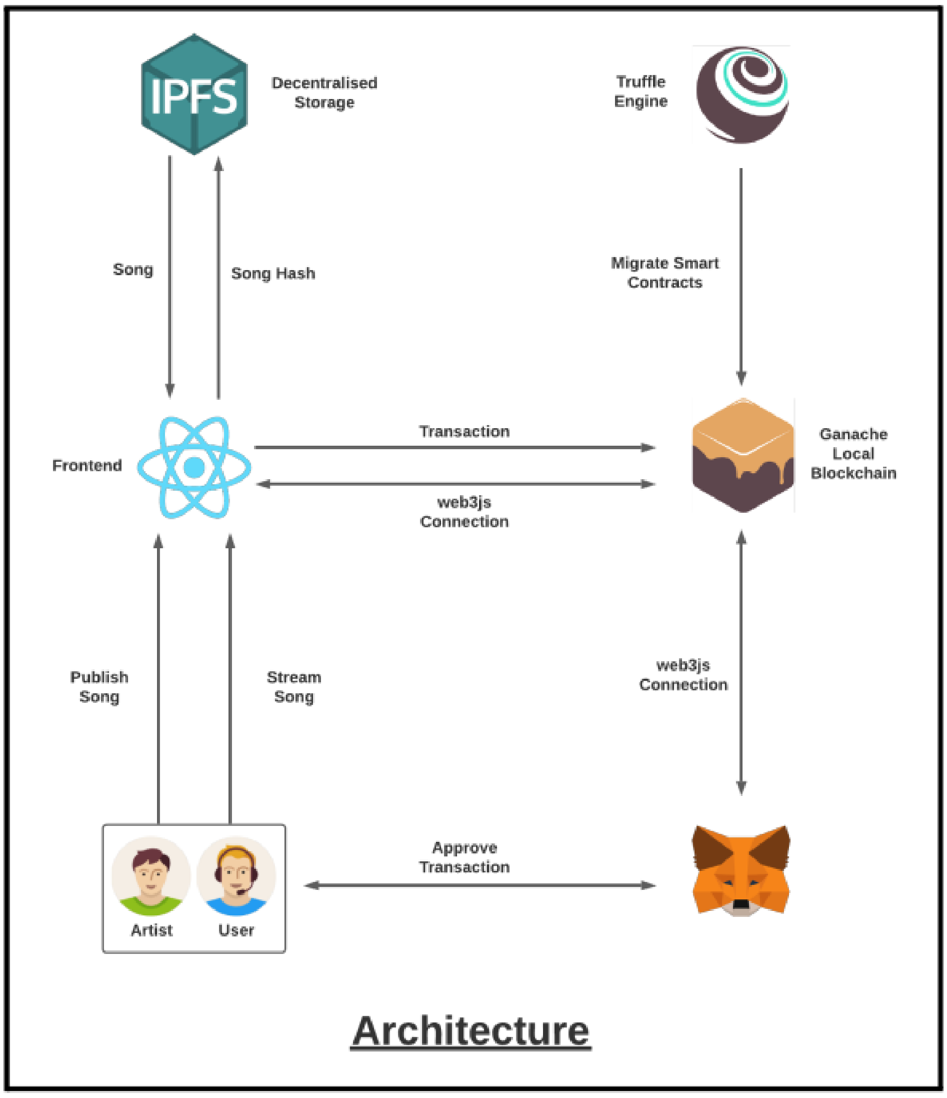


Fig 4.1 Architecture of the project.

CHAPTER – 5

IMPLEMENTATION

5.1 Frontend development

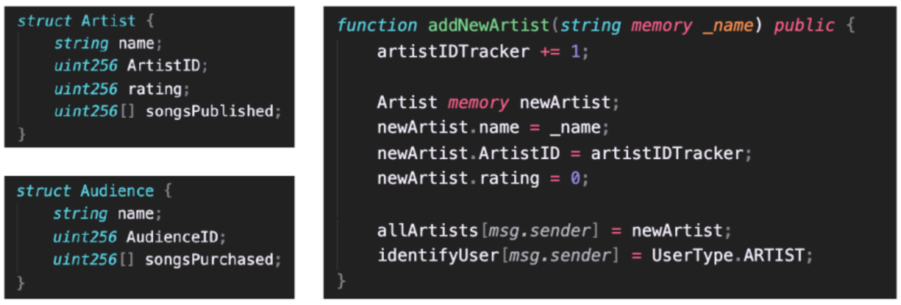
We designed a basic frontend to interact with our DApp and test the functionality. We worked on integrating our smart contract with our React frontend. The login page (as shown below) allows a new user to register as Artists or Audiences and a transaction is made for their registration on the local Ganache Blockchain.

Each artist can upload a song file and add its name, genre and cost. The artist page shows a list of all the songs added by him/her.

5.2 Blockchain

5.2.1 Smart contracts

We produce the first iteration of the smart contract for our application. The smart contract was tested on our local Ganache blockchain using Truffle. We created two structure- one for the artist and another for the audience. The artist has some properties like the name of the artist and the artist id (initially the artist id is initialized with 0, every time an artist register to our page the number of artist id is increased by 1 as shown in fig 5.1), rating (popularity, based of the number of purchases) and the songs published by the artist. Audience has some properties too like the name of the audience, audience id and the songs purchased by the audience.



```
struct Artist {
    string name;
    uint256 ArtistID;
    uint256 rating;
    uint256[] songsPublished;
}
```

```
struct Audience {
    string name;
    uint256 AudienceID;
    uint256[] songsPurchased;
}
```

```
function addNewArtist(string memory _name) public {
    artistIDTracker += 1;

    Artist memory newArtist;
    newArtist.name = _name;
    newArtist.ArtistID = artistIDTracker;
    newArtist.rating = 0;

    allArtists[msg.sender] = newArtist;
    identifyUser[msg.sender] = UserType.ARTIST;
}
```

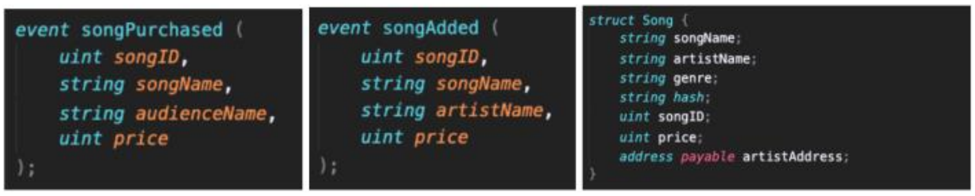
Fig 5.1 Smart Contracts.

After defining the structure for Song on our platform and creatiing solidity functions for the functionality of adding and buying songs in our smart contract we defined some appropriate checks such as:

1. addSong
   a. Only artists can add a song.
   b. The IPFS hash of the song must not be already in use (to avoid duplicates).

2. buySong
   a. Only Audience members can buy a song.
   b. Audience members cannot buy a song twice.

We defined two events songs purchased and the song added. Once the song is purchased it is added to the library of the audience. We defined a song structure too with various necessary attributes. The associated solidity code snippets are given below in fig 5.2.



```
event songPurchased (
    uint songID,
    string songName,
    string audienceName,
    uint price
);
```

```
event songAdded (
    uint songID,
    string songName,
    string artistName,
    uint price
);
```

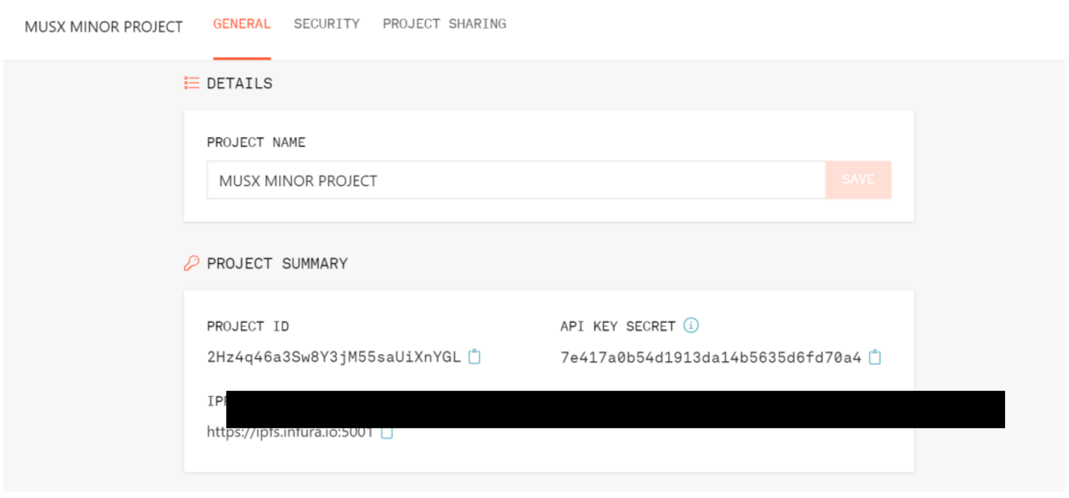
```
struct Song {
    string songName;
    string artistName;
    string genre;
    string hash;
    uint songID;
    uint price;
    address payable artistAddress;
}
```

Fig 5.2 Solidity code snippets.

5.2.2 IPFS

The IPFS storage system was integrated with the React frontend to enable storage of Artist and Song metadata. Artists will be allowed to upload new songs to the storage system, which then can be browsed by the audience.

The uploaded songs are stored on the IPFS Infura service. For each song a unique hash is generated which corresponds to its entry in the database and thus can be used to access the song. This hash is stored on the blockchain using the contract. A songID is also created to track the songs published by the artists and purchased by the audience. The buySong functionality is added to the audience along with the functionality to retrieve purchased songs from IPFS and play them.



DETAILS

PROJECT NAME
MUSX MINOR PROJECT

PROJECT SUMMARY

PROJECT ID
2Hz4q46a3Sw8Y3jM55saUiXnYGL

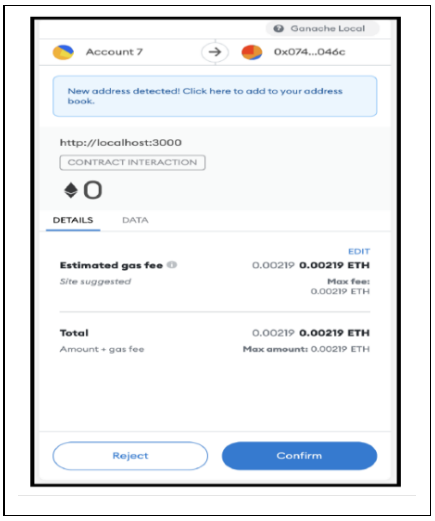
API KEY SECRET
7e417a0b54d1913da14b5635d6fd70a4

IP[redacted]
https://ipfs.infura.io:5001

Fig 5.3 Infura IPFS.

5.2.3 Transaction

While buying a song a transaction is made on the local Ganache Blockchain. The user has to import an account to metamask wallet from the dummy accounts provided by the ganache for developing purpose. The user has to connect the metamask to the current site and then register to an account. Once the user clicks on the register button the user has to pay some amount to create an account in MusX. The transaction is approved by the user through the metamask wallet as shown in fig 5.4.



Ganache Local

Account 7 → 0x074...046c

New address detected! Click here to add to your address book.

http://localhost:3000

CONTRACT INTERACTION

♦0

DETAILS DATA

EDIT

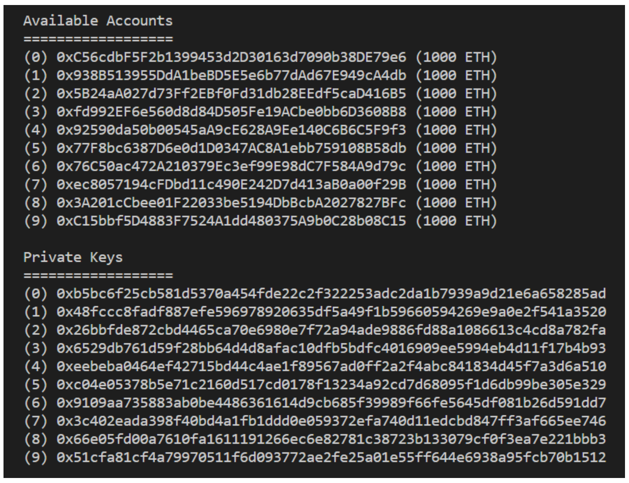
Estimated gas fee 0.00219 **0.00219 ETH**

Site suggested **Max fee:** 0.00219 ETH

Total 0.00219 **0.00219 ETH**

Amount + gas fee **Max amount:** 0.00219 ETH

Fig 5.4 Transaction using Metamask.



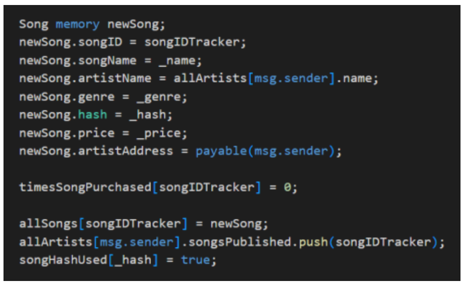
```
Available Accounts
==================
(0) 0xC56cdbF5F2b1399453d2D30163d7090b38DE79e6 (1000 ETH)
(1) 0x938B513955DdA1beBD5E5e6b77dAd67E949cA4db (1000 ETH)
(2) 0x5B24aA027d73Ff2EBf0Fd31db28EEdf5caD416B5 (1000 ETH)
(3) 0xfd992EF6e560d8d84D505Fe19ACbe0bb6D3608B8 (1000 ETH)
(4) 0x92590da50b00545aA9cE628A9Ee140C6B6C5F9f3 (1000 ETH)
(5) 0x77F8bc6387D6e0d1D0347AC8A1ebb759108B58db (1000 ETH)
(6) 0x76C50ac472A210379Ec3ef99E98dC7F584A9d79c (1000 ETH)
(7) 0xec8057194cFDbd11c490E242D7d413aB0a00f29B (1000 ETH)
(8) 0x3A201cCbee01F22033be5194DbBcbA2027827BFc (1000 ETH)
(9) 0xC15bbf5D4883F7524A1dd480375A9b0C28b08C15 (1000 ETH)

Private Keys
==================
(0) 0xb5bc6f25cb581d5370a454fde22c2f322253adc2da1b7939a9d21e6a658285ad
(1) 0x48fccc8fadf887efe596978920635df5a49f1b59660594269e9a0e2f541a3520
(2) 0x26bbfde872cbd4465ca70e6980e7f72a94ade9886fd88a1086613c4cd8a782fa
(3) 0x6529db761d59f28bb64d4d8afac10dfb5bdfc4016909ee5994eb4d11f17b4b93
(4) 0xeebeba0464ef42715bd44c4ae1f89567ad0ff2a2f4abc841834d45f7a3d6a510
(5) 0xc04e05378b5e71c2160d517cd0178f13234a92cd7d68095f1d6db99be305e329
(6) 0x9109aa735883ab0be4486361614d9cb685f39989f66fe5645df081b26d591dd7
(7) 0x3c402eada398f40bd4a1fb1ddd0e059372efa740d11edcbd847ff3af665ee746
(8) 0x66e05fd00a7610fa1611191266ec6e82781c38723b133079cf0f3ea7e221bbb3
(9) 0x51cfa81cf4a79970511f6d093772ae2fe25a01e55ff644e6938a95fcb70b1512
```

Fig 5.5 Ganache-cli command.

5.2.4 Ownership protection

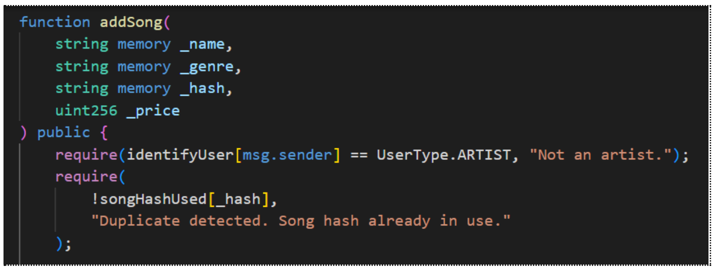
To prevent duplication of media files, we have stored hashes for the song files added by the artist. Once an artist uploads a song, we store the hash for the same song and if another user tries to upload the same song, first the hash is checked to file where the hashes are stored and if the hash match any of the existed hashes, then the transaction gets denied. In short, if someone tries to add a song file which already exists in the system, the transaction gets declined. The associate code snippets is shown below in fig 5.6.



```
Song memory newSong;
newSong.songID = songIDTracker;
newSong.songName = _name;
newSong.artistName = allArtists[msg.sender].name;
newSong.genre = _genre;
newSong.hash = _hash;
newSong.price = _price;
newSong.artistAddress = payable(msg.sender);

timesSongPurchased[songIDTracker] = 0;

allSongs[songIDTracker] = newSong;
allArtists[msg.sender].songsPublished.push(songIDTracker);
songHashUsed[_hash] = true;
```



```
function addSong(
    string memory _name,
    string memory _genre,
    string memory _hash,
    uint256 _price
) public {
    require(identifyUser[msg.sender] == UserType.ARTIST, "Not an artist.");
    require(
        !songHashUsed[_hash],
        "Duplicate detected. Song hash already in use."
    );
```

Fig 5.6 Duplicate song function.

CHAPTER 6

EXPERIMENTAL RESULTS AND ANALYSIS

6.1 Login as an artist/audience

When a user opens MusX site, a registration page is displayed to the user as given in fig 6.1. The registration page consists of our logo and a slider component with two tags i.e. artist and audience and a username text place where the user can enter a username of their choice and a register button. The user who is a singer and wants to publish a song then the user has to choose themselves as an artist by entering a username of their use. If a user only wants to listen and buy songs then he can choose themselves as an audience and register by entering a username of their choice.

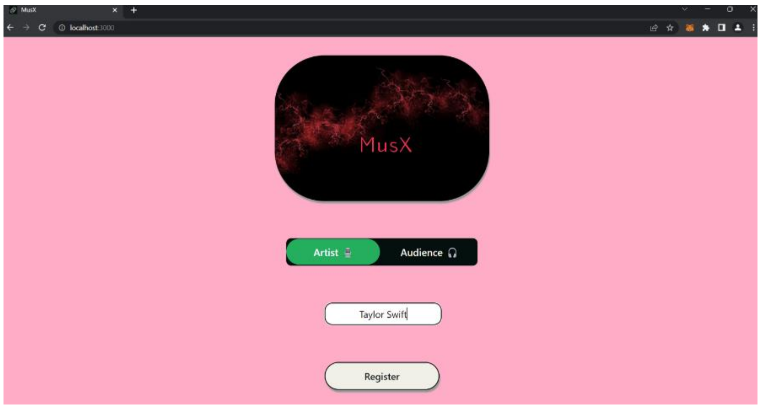
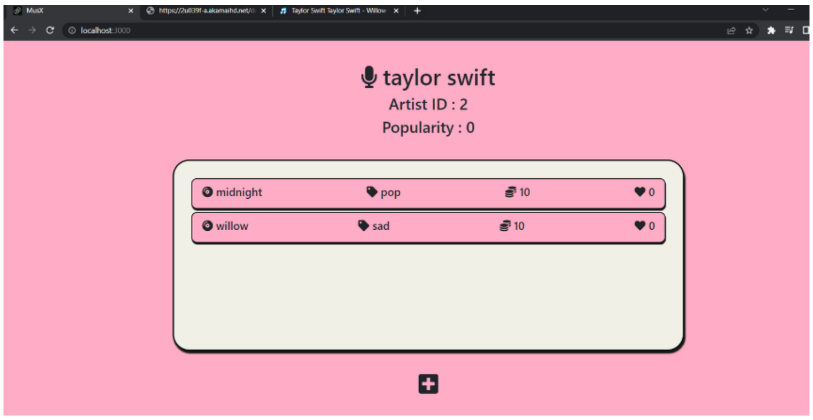


Fig 6.1 Login page.

6.1.1 Login as an artist

When the user register themselves as an artist with a username for example Taylor Swift. The username is highlighted on the top and an artist id is displayed for example Taylor Swift is the second artist to register on our site so it is displaying ARTIST ID: 2. The songs uploaded by the artist along with the name, genre, cost and number of likes is displayed as shown in fig 6.2.



taylor swift
Artist ID : 2
Popularity : 0

midnight	pop	10	0
willow	sad	10	0

Fig 6.2 Artist page.

A popularity measure for the artists is added which is calculated based on the total number of purchases made by the audience on their songs as shown in fig 6.3. For example if an audience buy his/her song the popularity measure is increased by one.

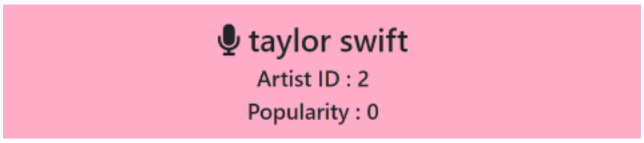
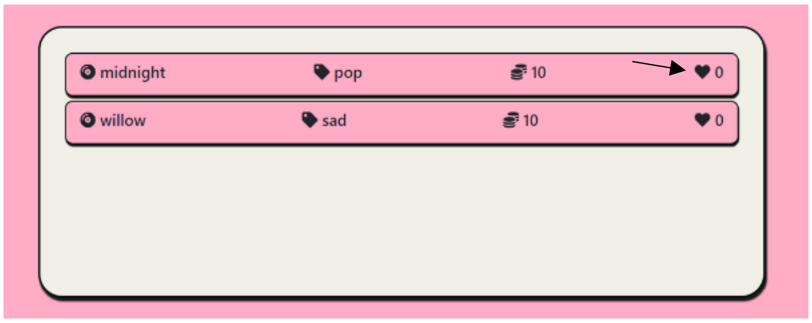


Fig 6.3 Popularity measure.

The artist can also view the number of purchases of individual songs to get an idea of how many people like his/her work as shown in fig 6.4. If an audience buys his “midnight” song then the number of likes will be increased by one in the midnight division.



midnight	pop	10	0
willow	sad	10	0

Fig 6.4 Purchase of individual songs.

The artist gets an option to upload the song by entering some details like name of the song, genre and the cost in milliwei as shown in fig 6.5. All the songs uploaded by the artist will be displayed on the profile with the name of the song, genre of the songs, cost of the song and the numbers of like/purchased.

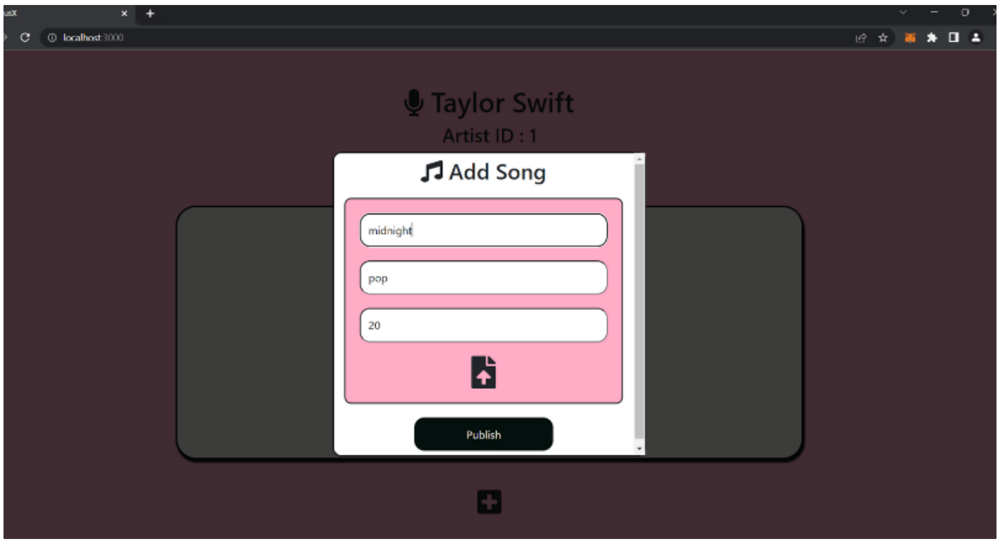
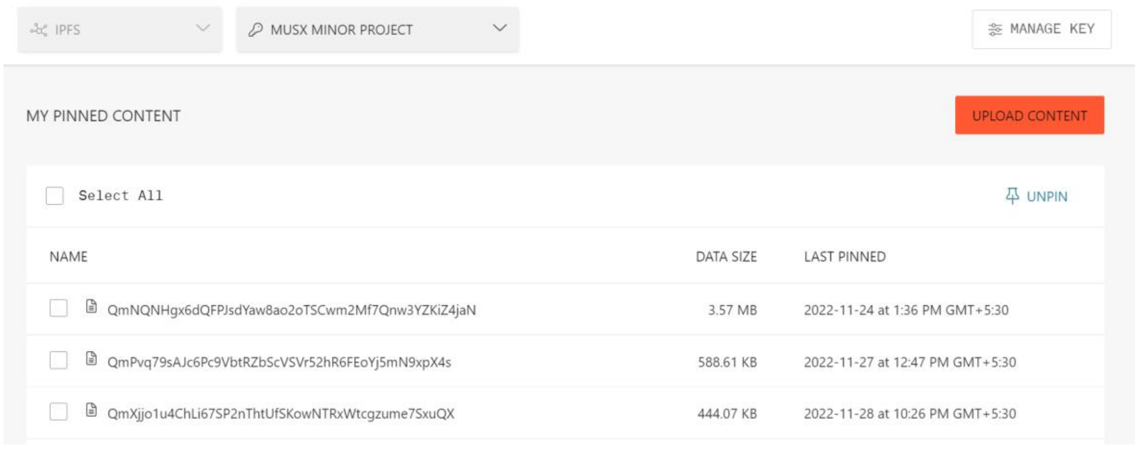


Fig 6.5 Upload song.

We created an account in Infura and created a project with name MUSX MINOR PROJECT for IPFS storage. The song once uploaded is stored in the form of hash in Infura-Ipfs MusX Minor Project as shown in fig 6.6.

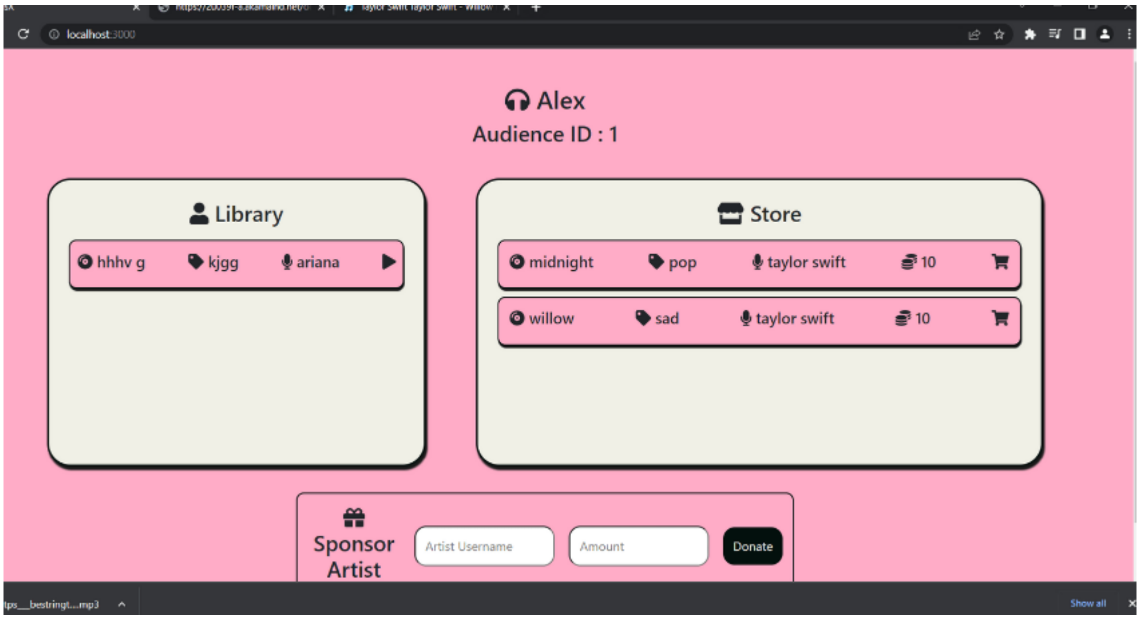


NAME	DATA SIZE	LAST PINNED
QmNQNHgx6dQFPJsdYaw8ao2oTSCwm2Mf7Qnw3YZKiZ4jaN	3.57 MB	2022-11-24 at 1:36 PM GMT+5:30
QmPvq79sAJc6Pc9VbtRZbScVSVr52hR6FEoYj5mN9xpX4s	588.61 KB	2022-11-27 at 12:47 PM GMT+5:30
QmXjjo1u4ChLi67SP2nThtUfSKowNTRxWtcgzume7SxuQX	444.07 KB	2022-11-28 at 10:26 PM GMT+5:30

Fig 6.6 Stored in Infura IPFS.

6.1.2 Login as an audience

When the user register as an audience the audience page opens up with the username at the top and an audience id. For example, Alex is the first audience to register in MusX so its audience id is one. Each time an audience register to our site the audience id in increased by one. The audience page consists of two divisions- library and store. All the songs uploaded by the artists are displayed on the store. The song consist of its name, genre, artist's name, cost and an option to buy song respectively as shown in fig 6.7.



hhhv g	kjgg	ariana	
midnight	pop	taylor swift	10
willow	sad	taylor swift	10

Fig 6.7 Audience page.

Once the audience desire to purchase a song, a transaction is made through local ganache blockchain with metamask wallet. Once the user confirms the payment as shown in fig 5.4, the song is added to his library with name of the song, genre, artist's name and play and stop button as shown in fig 6.8. Once the audience purchase a song, the popularity is artist's page is increased by one as well as the individual like on the song is also increased by one too. The audience can also play/stop the song. The code snippet of play/stop song is given below in fig 6.9.

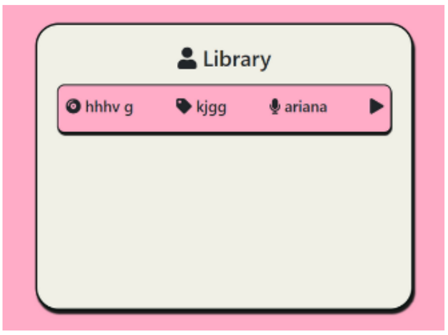
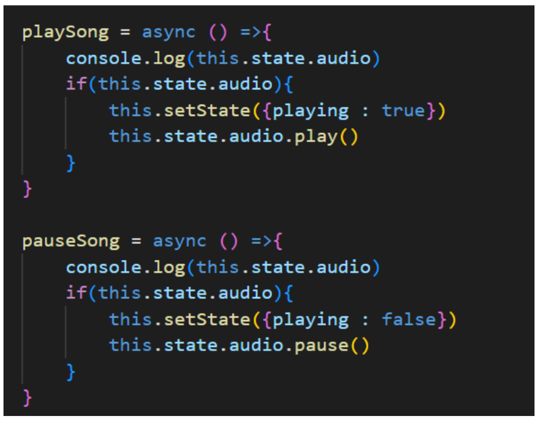


Fig 6.8 Library.



```
playSong = async () =>{
    console.log(this.state.audio)
    if(this.state.audio){
        this.setState({playing : true})
        this.state.audio.play()
    }
}

pauseSong = async () =>{
    console.log(this.state.audio)
    if(this.state.audio){
        this.setState({playing : false})
        this.state.audio.pause()
    }
}
```

Fig 6.9 Play/stop song code snippet.

Audiences can also make donations to their favourite artists to show their support. The audience needs to provide the artist username and donation amount in milliwei. The amount is directly transferred to the respective artist's wallet. This features helps the small artists who are new to the platform.

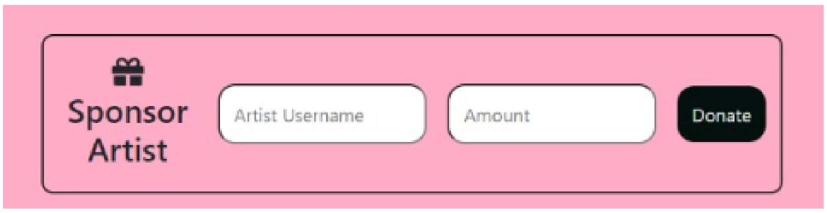


Fig 6.9 Donation to artist.

6.1.3 Copyright protection

The prevent duplication of file, we stored hashes for the song files added by the artist. Once an artist uploads a song, we store the hash for the same song and if another user tries to upload the same song, first the hash is checked to file where the hashes are stored and if the hash match any of the existed hashes, then the transaction gets denied. In short, if someone tries to add a song file which already exists in the system, the transaction gets failed as shown in fig 6.11.

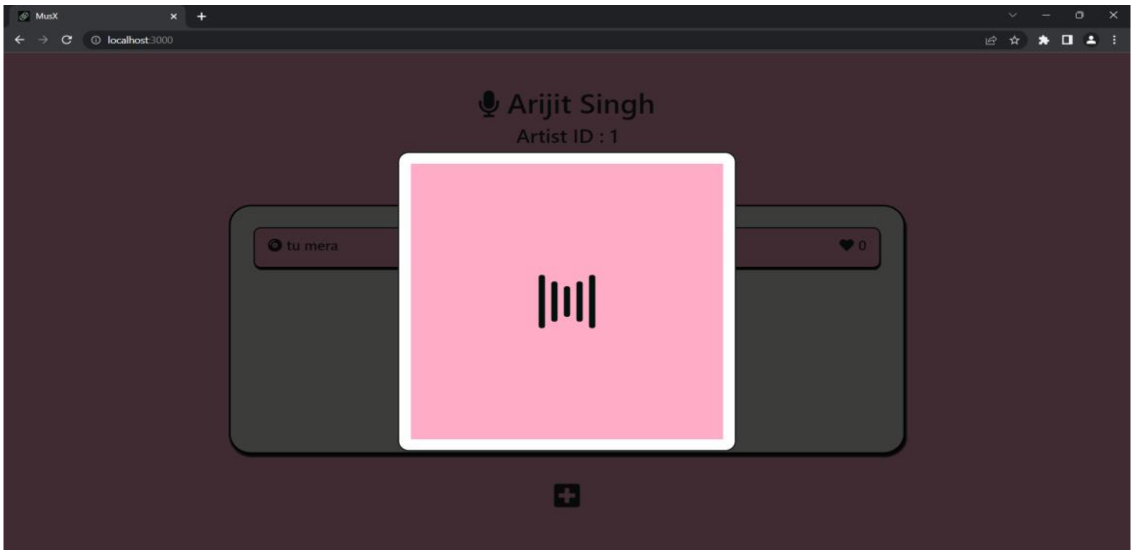
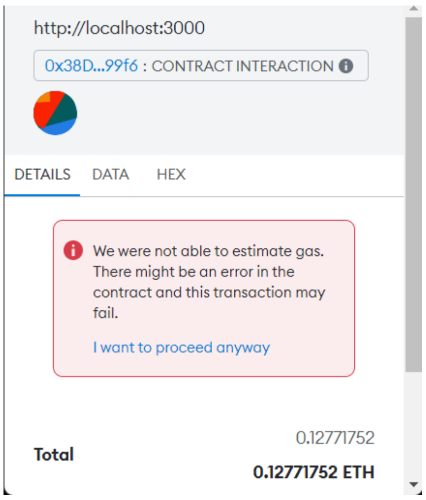


Fig 6.10 Re-uploading same song.


http://localhost:3000

0x38D...99f6 : CONTRACT INTERACTION

DETAILS DATA HEX

We were not able to estimate gas. There might be an error in the contract and this transaction may fail.

I want to proceed anyway

Total 0.12771752
0.12771752 ETH

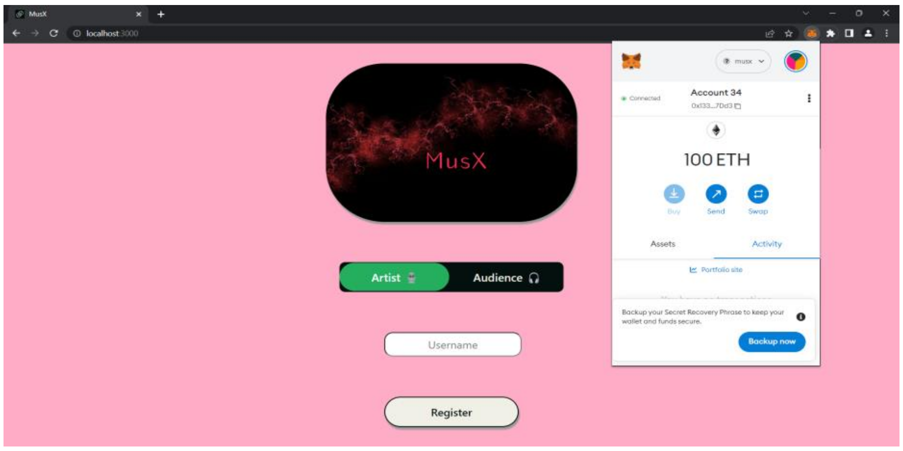
Fig 6.11 Transaction declined for re-uploading.

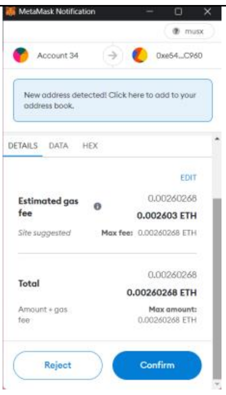
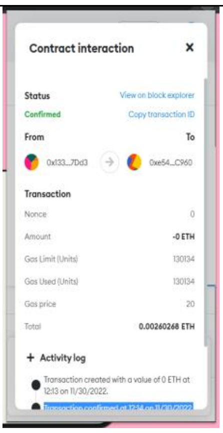
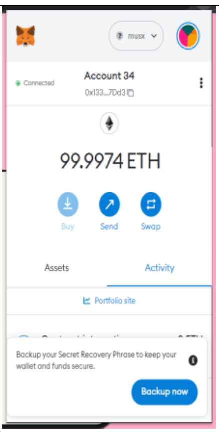
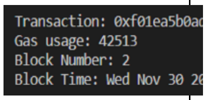
6.2 Transaction record:

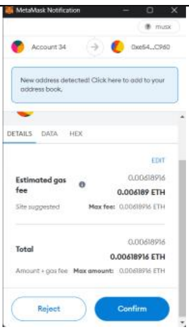
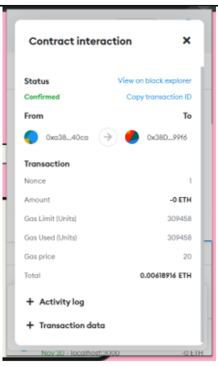
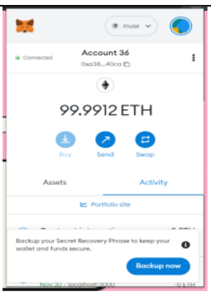
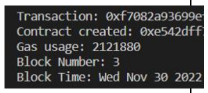
The user has to connect to metamask to run our Dapp. He/she has to import an account through local ganache blockchain.

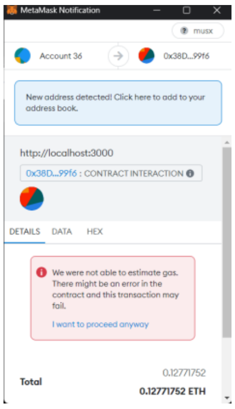
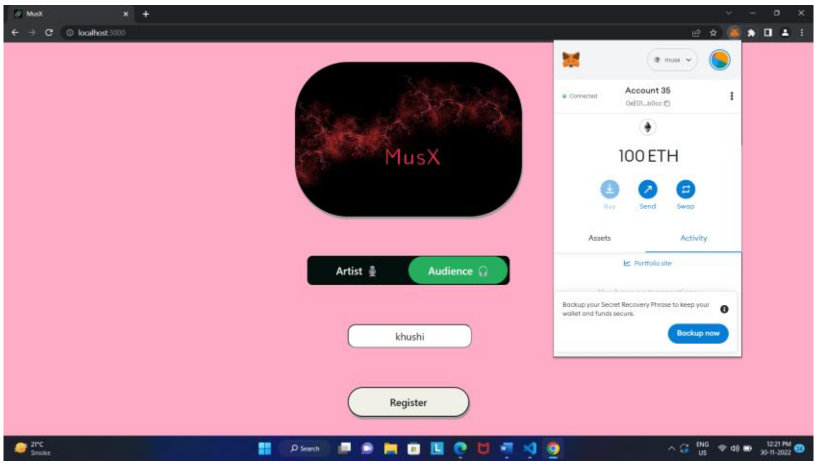
Importing accounts are accounts you import using a private key string.

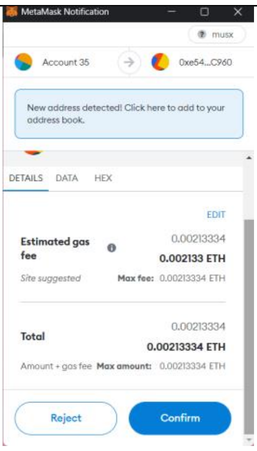
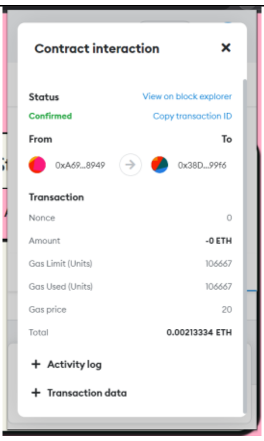
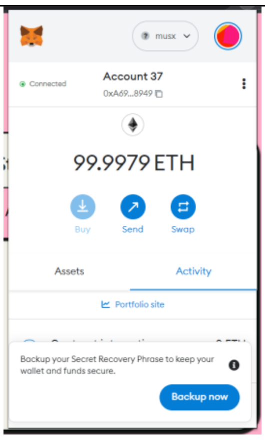
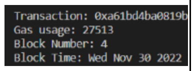
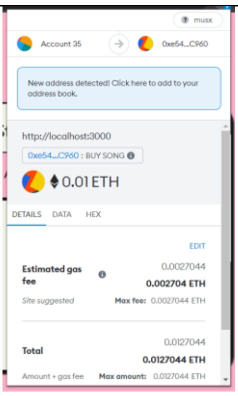
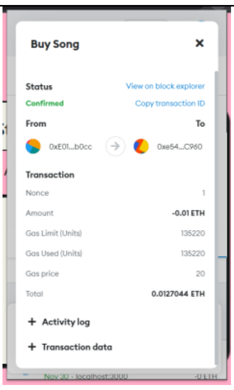
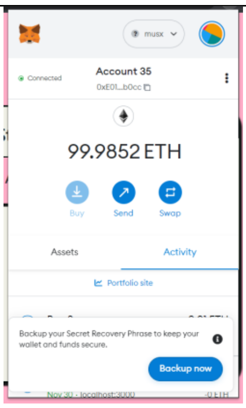
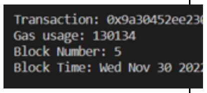
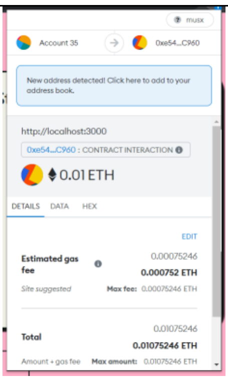
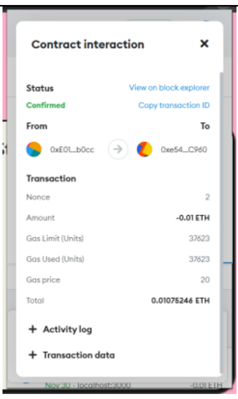
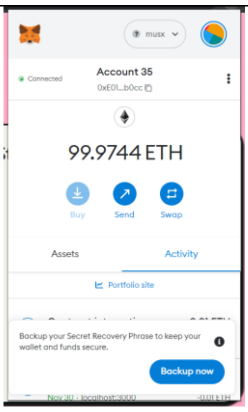
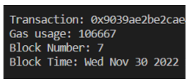
After importing the generated account fills up with the amount of 100 ETH to make transactions.

To create an artist account, we import a private key through local ganache blockchain.	 MusX localhost:3000 musx Connected Account 34 0x133...7Dd3 100 ETH Buy Send Swap Assets Activity Portfolio site Backup your Secret Recovery Phrase to keep your wallet and funds secure. Backup now MusX Artist Audience Username Register After importing an account from local ganache blockchain.			

To create an artist id. The user has to pay small amount of gas fee.	 MetaMask Notification
	musx
	Account 34
	0xe54...C960
	DETAILS
	DATA
	HEX
	EDIT
	Reject
	Confirm New address detected! Click here to add to your address book.
	Estimated gas fee 0.00260268 0.002603 ETH
	Site suggested Max fee: 0.00260268 ETH
	Total 0.00260268 0.00260268 ETH
	Amount + gas fee Max amount: 0.00260268 ETH
	
	Amount for creating an artist account.
		 Contract interaction
		View on block explorer
		Copy transaction ID
		Activity log Status Confirmed
		From 0x133...7Dd3 To 0xe54...C960
		Transaction
		Nonce 0
		Amount -0 ETH
		Gas Limit (Units) 130134
		Gas Used (Units) 130134
		Gas price 20
		Total 0.00260268 ETH
		Transaction created with a value of 0 ETH at 12:13 on 11/30/2022.
		Transaction confirmed at 12:14 on 11/30/2022
		
		Confirming the transaction.
			 musx
			Connected
			Account 34
			0x133...7Dd3
			99.9974 ETH
			Buy
			Send
			Swap
			Assets
			Activity
			Portfolio site
			Backup your Secret Recovery Phrase to keep your wallet and funds secure.
			Backup now Remaining amount after the transaction.
				 `Transaction: 0xf01ea5b0ad`
				`Gas usage: 42513`
				`Block Number: 2`
				`Block Time: Wed Nov 30 20`
				
				Confirmation of transaction.

To upload a song, user has to pay a small amount.	 MetaMask Notification
	musx
	Account 34
	0x64...C960
	DETAILS
	DATA
	HEX
	EDIT
	Reject
	Confirm
	New address detected! Click here to add to your address book.
	Estimated gas fee 0.00618916 0.006189 ETH
	Site suggested Max fee: 0.00618916 ETH
	Total 0.00618916 0.00618916 ETH
	Amount + gas fee Max amount: 0.00618916 ETH
	Amount for uploading a song.
		 Contract interaction
		View on block explorer
		Copy transaction ID
		Activity log
		Transaction data
		Status Confirmed
		From 0xa38...40ca To 0x38D...99f6
		Transaction
		Nonce 1
		Amount -0 ETH
		Gas Limit (Units) 309458
		Gas Used (Units) 309458
		Gas price 20
		Total 0.00618916 ETH
		Confirming the transaction.
			 musx
			Connected
			Account 36
			0xa38...40ca
			Buy
			Send
			Swap
			Assets
			Activity
			Portfolio site
			Backup now
			99.9912 ETH
			Backup your Secret Recovery Phrase to keep your wallet and funds secure.
			Remaining amount after the transaction.
				
				`Transaction: 0xf7082a93699e`
				`Contract created: 0xe542dff`
				`Gas usage: 2121880`
				`Block Number: 3`
				`Block Time: Wed Nov 30 2022`
				Confirmation of transaction.

When someone tries to upload a song which already exist on the system, then the transaction gets failed.	 MetaMask Notification musx Account 36 0x38D...99f6 DETAILS DATA HEX I want to proceed anyway New address detected! Click here to add to your address book. http://localhost:3000 0x38D...99f6 : CONTRACT INTERACTION We were not able to estimate gas. There might be an error in the contract and this transaction may fail. Total 0.12771752 0.12771752 ETH Transaction failed.			
To create an artist account, we import a private key through local ganache blockchain.	 MusX localhost:3000 MusX Artist Audience khushi Register musx Connected Account 35 0xE01...b0cc Buy Send Swap Assets Activity Portfolio site Backup now 100 ETH Backup your Secret Recovery Phrase to keep your wallet and funds secure. After importing an account from local ganache blockchain.			

To create an audience id. The user has to pay small amount of gas fee.	 MetaMask Notification musx Account 35 → 0xe54...C960 New address detected! Click here to add to your address book. DETAILS DATA HEX EDIT Estimated gas fee 0.00213334 0.002133 ETH Site suggested Max fee: 0.00213334 ETH Total 0.00213334 0.00213334 ETH Amount + gas fee Max amount: 0.00213334 ETH Reject Confirm Amount for creating an audience account.	 Contract interaction Status View on block explorer Confirmed Copy transaction ID From To 0xA69...8949 → 0x38D...99f6 Transaction Nonce 0 Amount -0 ETH Gas Limit (Units) 106667 Gas Used (Units) 106667 Gas price 20 Total 0.00213334 ETH + Activity log + Transaction data Confirming the transaction.	 musx Connected Account 37 0xA69...8949 99.9979 ETH Buy Send Swap Assets Activity Portfolio site Backup your Secret Recovery Phrase to keep your wallet and funds secure. Backup now Remaining amount after the transaction.	 Transaction: 0xa61bd4ba0819b Gas usage: 27513 Block Number: 4 Block Time: Wed Nov 30 2022 Confirmation of transaction.
To purchase a song.	 musx Account 35 → 0xe54...C960 New address detected! Click here to add to your address book. http://localhost:3000 0xe54...C960 : BUY SONG 0.01 ETH DETAILS DATA HEX EDIT Estimated gas fee 0.0027044 0.002704 ETH Site suggested Max fee: 0.0027044 ETH Total 0.0127044 0.0127044 ETH Amount + gas fee Max amount: 0.0127044 ETH Amount for purchasing a song.	 Buy Song Status View on block explorer Confirmed Copy transaction ID From To 0xE01...b0cc → 0xe54...C960 Transaction Nonce 1 Amount -0.01 ETH Gas Limit (Units) 135220 Gas Used (Units) 135220 Gas price 20 Total 0.0127044 ETH + Activity log + Transaction data Confirming the transaction.	 musx Connected Account 35 0xE01...b0cc 99.9852 ETH Buy Send Swap Assets Activity Portfolio site Backup your Secret Recovery Phrase to keep your wallet and funds secure. Backup now Remaining amount after the transaction.	 Transaction: 0x9a30452ee23 Gas usage: 130134 Block Number: 5 Block Time: Wed Nov 30 202 Confirmation of transaction.
Donating to an artist.	 musx Account 35 → 0xe54...C960 New address detected! Click here to add to your address book. http://localhost:3000 0xe54...C960 : CONTRACT INTERACTION 0.01 ETH DETAILS DATA HEX EDIT Estimated gas fee 0.00075246 0.000752 ETH Site suggested Max fee: 0.00075246 ETH Total 0.01075246 0.01075246 ETH Amount + gas fee Max amount: 0.01075246 ETH Donation of amount.	 Contract interaction Status View on block explorer Confirmed Copy transaction ID From To 0xE01...b0cc → 0xe54...C960 Transaction Nonce 2 Amount -0.01 ETH Gas Limit (Units) 37623 Gas Used (Units) 37623 Gas price 20 Total 0.01075246 ETH + Activity log + Transaction data Confirming the transaction.	 musx Connected Account 35 0xE01...b0cc 99.9744 ETH Buy Send Swap Assets Activity Portfolio site Backup your Secret Recovery Phrase to keep your wallet and funds secure. Backup now Remaining amount after the transaction.	 Transaction: 0x9039ae2be2cae Gas usage: 106667 Block Number: 7 Block Time: Wed Nov 30 2022 Confirmation of transaction.

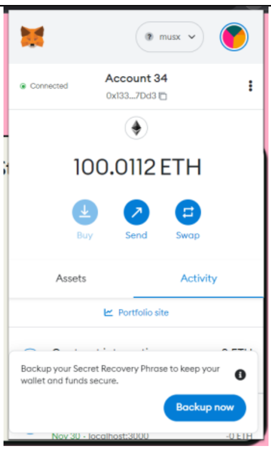
Artist account after one audience purchased a song and provide donation.	 musx Connected Account 34 0x133...7Dd3 100.0112 ETH Buy Send Swap Assets Activity Portfolio site Backup your Secret Recovery Phrase to keep your wallet and funds secure. Backup now Nov 30 · localhost:3000 -0 ETH Artist account after donation and purchasing by an sudience.

Fig 6.12 Transaction record.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

MusX helps to connect music enthusiasts directly to independent music artists. The artists can use this platform to share their music with greater freedom while ensuring ownership and avoiding duplication of their music. People can listen to their favourite songs and support the artists by making micropayments.

Blockchain offers the opportunity for the music industry to free itself from the current bureaucracy which is crippling both profitability and creative throughput. Blockchain platforms provide an empowering solution for an open, fair and transparent music industry ecosystem with an emphasis on value-based earnings for creators.

We thoroughly tested each and every functionality and accordingly fixed the bugs encountered. The code was refactored and unnecessary code snippets were removed. We revisited our smart contract to make improvements wherever possible.

7.2 Future scope

- Scale the system.
- Upcomping concert updates.
- Integrate a recommender system.
- Build a marketplace for the artists to sell their NFTs.

REFERENCES

Book

[1] Melanie Swan, "*Blockchain 1st edition*". O'REILLY.

[2] Tiana Laurence, *"Blockchain for dummies"*,Willey India Pvt. Ltd.

[3] Kapil Jain, *"Bitcoin Blockchain"*, BPC Publications, India.

Journal Article

[4] Camila Sitonia and Alberto Nucciarelli, "*The Impact of Blockchain on th Music Industry"*, R&D Management Conferences, 2018.

[5] Oliver Kabi and Virginia N. L. Franqueira, *"Blockchain-based Distributed Marketplace"*, Workshop on Blockchain and Smart Contract Technologies, 2018.

[6] Kim, Kenneth Chi Ho *"The Impact of Blockchain on the Music Industry"*, International journal of advanced smart convergence, 2019.

[7] Luis Claudio Arcos, *"The Blockchain technology on the Music Industry"*, Open Music Summer Lab, 2017.

Online

[8] "Introduction to Smart Contracts", https://ethereum.org/en/developers/docs/smart-contracts/

[9] "IPFS powers the Distributed Web", https://ipfs.io

[10] "React – A JavaScript library for building user interfaces", https://reactjs.org/

[11] "Solidity", https://docs.soliditylang.org/en/v0.8.9/

[12] "IPFS Infura Docs", https://docs.infura.io/infura/networks/ipfs

Online Courses

[13] "How to setup a Dapp project with React & Truffle", https://www.youtube.com/watch?v=nU0uqk0jLdw&ab_channel=EatTheBlocks

[14] Angela Yu, *"The Complete 2023 Web Development Bootcamp"*, Udemy.

[15] Filip Jerga, *"Solidity & Ethereum in React(Next JS)": The Complete Guide.* Udemy.

[16] Code Eater, https://www.youtube.com/@CodeEater21/featured

Work Distribution

1. All of us explored the basics of the technologies mentioned.
2. Khushi explored the resound project, Vreeti explored the musicDAPP project and Vaishali explored the Jelly-Beats project.
3. All of us brainstormed together to come up with potential features. Vreeti proposed we should use a custom token for our system to prevent the transaction of Eth in decimal values. Khushi proposed we can add a crowdfunding feature where users can support their artists. Vaishali proposed that we should also display some stats about songs
4. All of us collectively worked on the architecture diagram.
5. All of us discussed the possible structure of the smart contract. Vreeti came up with a basic contract in solidity which was further tested and improved by Khushi & Vaishali.
6. Khushi worked on designing the user interface and linking the app with web3, while Vreeti & Vaishali focussed on deploying and integrating the smart contract.
7. Vaishali worked on integrating the IPFS, while Khushi and Vreeti continued with developing smart contract and frontend.
8. All of us brainstormed the required attributes and functions in react to support the DApp and developed the Frontend.
9. Vaishali focused on file handling, IPFS functionality and its integration while Khushi and Vreeti integrated the react code with the smart contract.
10. Khushi focused on adding the support artist feature while Vreeti looked at hash generation and Vaishali focused on the popularity calculation.
11. All of us tested the DApp, refactored the code and prepared the final report and presentation for final evaluation and submission.