Music NFTs Player using BLOCKCHAIN

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Vaibhav Mishra (191505)

Under the supervision of

Dr. Pardeep Kumar

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

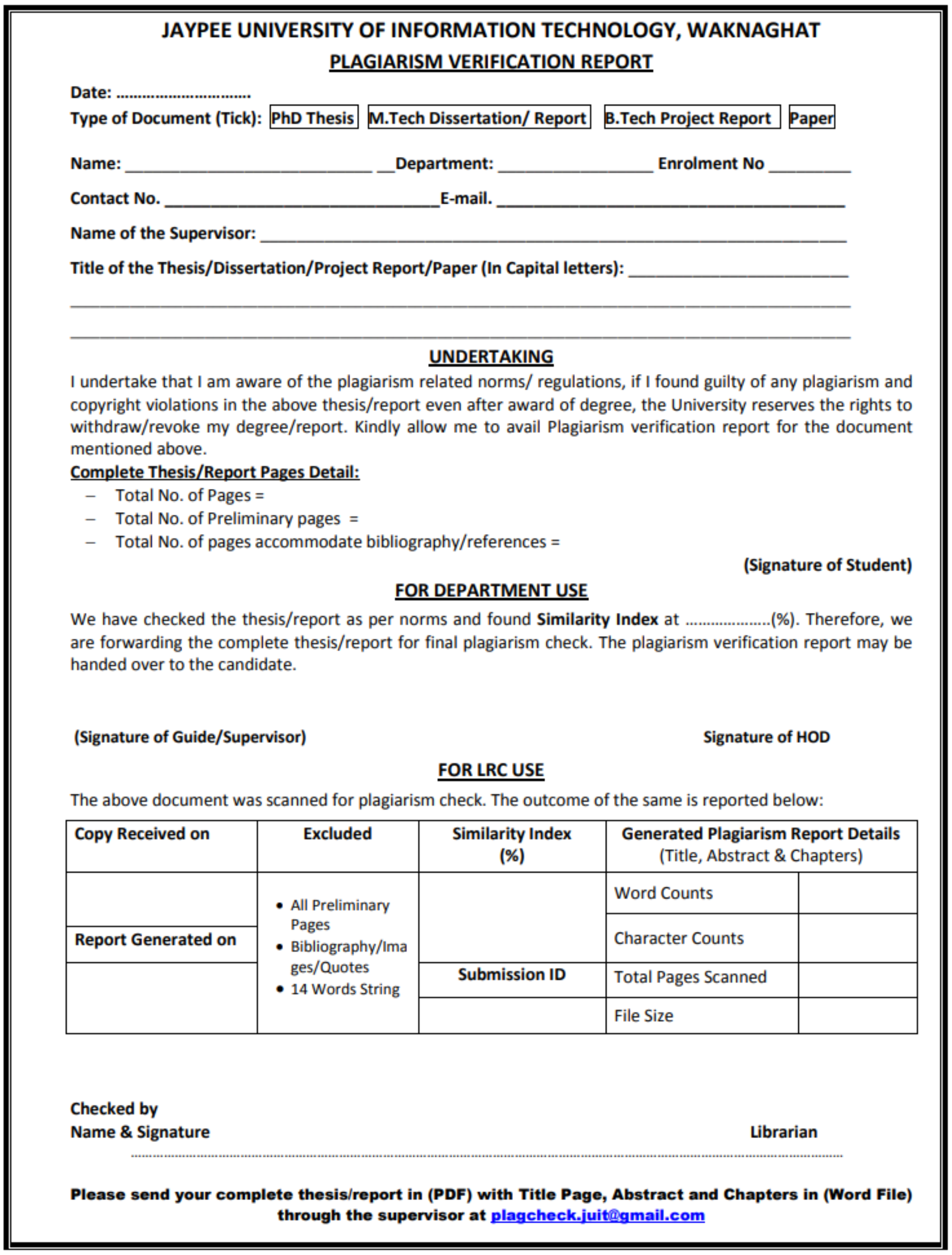
I hereby declare that the work presented in this report entitled **"Music NFTs Player using BLOCKCHAIN"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2023 to May 2023 under the supervision of **Dr. Pardeep Kumar** (Associate Professor,SM-ACM).
I also authenticate that I have carried out the above mentioned project work under the company name NONCELABS Pvt. Ltd**.**
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Vaibhav Mishra,
191505

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pardeep Kumar
Associate Professor,SM-ACM
Department of Computer Science and Engineering and Information Technology
Dated: 10/05/2023

PLAGIARISM CERTIFICATE



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: ____________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. _______________________________**E-mail.** _______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at **plagcheck.juit@gmail.com**

ACKNOWLEDGEMENTS

We are thankful to all the people who joined as part of making this journey of fulfilling this project into a working model. I am grateful to NONCELABS Pvt. Ltd. for giving me a wonderful platform for exploring new software developing skills during the making of this project.

Additionally, I would like to extend my sincere appreciation to Dr. Vivek Sehgal, Head of the Department, for providing all the assistance and support necessary for the project's conception, execution, and presentation.

I am also thankful to my mentor Dr. Pardeep Kumar as well as other staff members of the Computer Science and Engineering department, Jaypee University of Information Technology for their constructive and helpful inputs.

Vaibhav Mishra,
191505

Table of Contents

List of Abbreviations

AML	**Anti-Money Laundering**
API	**Application Programming Interface**
BFT	**Byzantine Fault Tolerance**
DAO	**Decentralised Autonomous Organization**
DAG	**Directed Acyclic Graph**
DeFi	**Decentralised Finance**
DLT	**Distributed Ledger Technology**
ERC	**Ethereum Request for Comments**
ICO	**Initial Coin Offering**
IPFS	**Inter Planetary File System**
KYC	**Know Your Customer**
MVP	**Minimum Viable Product**
NFT	**Non-Fungible Token**
P2P	**Peer-to-Peer**
PBFT	**Practical Byzantine Fault Tolerance**
PoW	**Proof of Work**
PoS	**Proof of Stake**

<u>**List of Figures**</u>

List of Tables

Abstract

The goal of the D-App is to allow the users to buy music made by different artists as NFTs and also resell them if needed, another feature is that the artist will also receive some payment in the form of NFTs. This project is basically an NFT Marketplace and was assigned to me by the company in order to help me get a better understanding of the skills that are needed in the area of BlockChain design and development. This webApp was supposed to be the implementation of the knowledge that we have acquired while learning blockchain designing and development in the last four to five months. This webApp follows a two-tier architecture. The presentation tier or the front-end is kept simple and user friendly. The main purpose of this tier is to display information and to collect information from the user i.e It is implemented for the users to interact with the smart contracts built using Ethereum solidity. The application tier is the heart of web applications. It is also where the solidity contracts are used to interact with the users and process the information received. All the information about the user collected from the presentation tier is processed and shown in the temporary backend which is the EVM Testnet interaction generated by Hardhat tool . It is mainly implemented using nodejs.

The two-tier architecture ensured faster development as both tiers can be developed simultaneously. It also ensures improved scalability, reliability and security.The most important part is the integration of the smart contracts and the frontend followed by the temporary backend created using EVM Testnet interactions.

Chapter-1
INTRODUCTION

1.1 Introduction

It is a basic decentralised application that allows users to buy listed songs as NFTs and also resell them whenever needed. The main part is the interaction with the smart contracts and ensure secured transaction of the NFTs from the users respective accounts. Front-end of the website is made using html, css, javascript. Bootstrap templates are used extensively. For the temporary back-end we are using nodejs and Hardhat tool which is used to create a testing environment for our code's testing by generating fake accounts on EVM Testnet interaction using Hardhat tool.
The D-App allows users to buy NFTs of music by first connecting their accounts to D-App and only then transactions are carried out in the testing environment.

1.2 Problem Statement

To apply industrial best practices and create a fast, scalable and secure decentralised application. To learn and apply the knowledge of front-end development in real life projects and to understand the in-depth working of BLOCKCHAIN development and smart contracts.

1.3 Objectives

To create testable, structured, clean and maintainable decentralised applications by using industrial best practices. To apply the knowledge about the technologies taught to us thus far and gain practical experience.

1.4 Methodology

The front-end of this application is developed exclusively using Reactjs. Plain CSS is used for styling since the main objective of the project is to integrate smart contracts with the front end for a user-friendly NFT marketplace and not the front-end. This combination allows faster development and scalability. Each component can be developed simultaneously and error in one component won't affect other components.
For the temporary backend and to create a safe testing environment we have used Hardhat tool which is quite popular and easy to use in the world of Blockchain Development.

Transactions will be carried out as well as verified using Metamask which is an inbuilt software wallet that communicates with the Ethereum Blockchain.

1.5 Organisation

Dedicated to building scalable blockchain products, Nonceblox is a group of talented blockchain architects, consultants, SMBs and cryptocurrency advisors. Hyperledger, Polygon, Solana, BSC and Polkadot solutions were developed by Nonceblox. Our talented team of blockchain designers, experts, business SMEs and cryptocurrency advisors are dedicated to developing blockchain technology that is efficient and beneficial for businesses.

Fig 1: Company logo

Chapter-2
LITERATURE SURVEY

1. React documentation

React is a front-end framework that allows users to write reusable code for every component of a website. Each component can be combined to develop a complete UI (user interface)[1].

2. Javascript documentation - Mozilla

Write simple to advanced asynchronous javascript to build fully scripted amd scalable web applications[2].

3. Git and Github

Official document that tells you about the concept of version control, Git, and how it works with GitHub[3].

4. Blockchain

A report by Satpal Singh Kushwaha recognizes and states that Blockchain is a revolutionary technology that allows users to communicate reliably[4]. It changes the business model of organisations without the assistance of a reliable third party.

5. Ethereum

A report by Dr. Gavin Wood talks about the blockchain paradigm which combined with cryptographically secure transactions has seen its usefulness in many projects, notably Bitcoin[5]. [6] All these operations can be viewed as a simple application of classification , but as a computational application. We can call this application as a single process with shared states.

6. WEB3.0 Auth. documentation

Blockchain development requires authentication and security which can be provided by the web3.0 authorization to connect your wallet along with the number of stages of authorization for a user-satisfactory transaction [7].

7. Introducing JSX - React

Official React website that guides beginners about how to write JSX code. Also explains how the Babel compiler works.

8. HardHat

Hardhat is a development tool for Ethereum software.[8] It has different components for editing, compiling, debugging and deploying smart contracts and dApps, and all together form a complete development environment.

9. Mocha

Mocha is a JavaScript testing framework that offers browser support, asynchronous testing, test reporting, and validation libraries for Node.js programs[9].

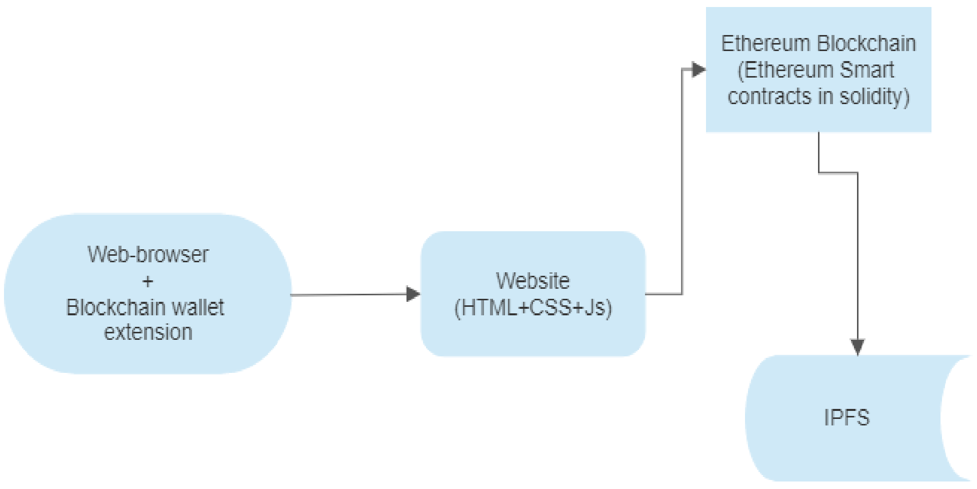
10. IPFS

It is a decentralised file storage system This storage system allows direct interaction through a secure global P2P network. Combining these two strengths, IPFS enables the web to constantly innovate and improve the way we use Internet protocols such as HTTP.

Chapter-3
SYSTEM DESIGN AND DEVELOPMENT

3. SYSTEM DEVELOPMENT

3.1 System design diagram



Ethereum Blockchain (Ethereum Smart contracts in solidity)
Web-browser + Blockchain wallet extension
Website (HTML+CSS+Js)
IPFS

Fig 2: System design diagram

3.2 System design implementation

How it Works : A react web application usually has two sub-folders for client-side and server-side applications but here the web browser and Blockchain wallet will directly interact with the website (front-end) which we have built and the website will directly read and write data in to the Ethereum blockchain which has ethereum smart contracts written in solidity. After this all the music files will get stored in the decentralised file storage i.e IPFS. This makes it faster than other sites and secure along with it.

3.2.1 Identification of features

The features of web application includes :

- Connecting to the website using the browser and blockchain wallet extension for that browser.
- Purchasing the song from the website.
- The purchased song will be removed from the available songs for that account and will be shown under purchased songs..
- The user can sell the purchased song NFT with the new amount , the song will be removed from purchased songs and will be shown in available songs.
- Once the NFT on sale is purchased by someone that NFT will be shown in my-reasle tab.
- Users can disconnect accounts after use.

3.2.2 Libraries and frameworks used

JS is a lightweight, compiled or interpreted script with many features. Many non-browser themes like Node.js, Apache CouchDB, and Adobe Acrobat use it as it is one of the best server-side scripting languages.

Reactjs : React is a free open source front-end library based on javascript for building user-based components. It is maintained by the meta and a group of freelancers and businesses. With a framework like Next.js, React can be used to build single page, mobile, or server-generated apps. Because React deals only with the UI and renders objects to the DOM, libraries in React applications often provide instructions and other user functions.

Nodejs : [10] The Node.js, an open source server environment, is cross-platform and works with many operating systems, including Windows, Linux, Unix, macOS, and more. JavaScript code can be executed outside the web using Node.js, a backend engine that uses the V8 JavaScript engine. JavaScript is a scripting language that developers can use to create server-side scripts and command-line tools using Node.js.

The server usually needs to be able to run JavaScript code to generate before the page is provided to the user, a dynamic web content website.

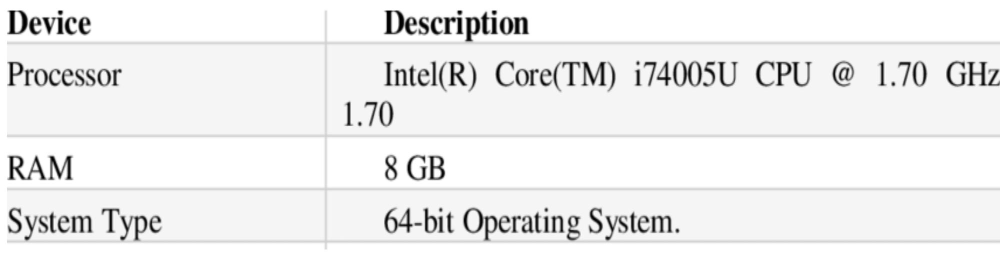
Blockchain : Blockchain is a shared, immutable database that streamlines the process of recording transactions and tracking resources across businesses. Assets can be tangible (houses, cars, cash, land) or intangible (intellectual property, patents, copyrights, brands). Almost anything of value on the blockchain network can be traced and traded, reducing risk and lowering costs for all parties involved. Business depends on knowledge. The sooner and more taken, the better.
Blockchain is ideal for transferring this information as it provides instant, shared and transparent information stored in a transferable ledger that only network members cannot access. Blockchain networks can track orders, payments, money, production and more. As members share a single view of reality, you see all the details of the end-to-end deal, giving you confidence and unlocking the experience of new benefits and opportunities**.**

Nodemon : If we change and save the file, nodemon will automatically start the server. Without a nodemon, the server must be restarted after each change. It saves a lot of time and effort. Once tested, you can run the site on your localhost using the "nodemon app" command.

3.2.3 Technical Requirements

- **VS Code**(preferred IDE) / sublime text
- **Hardhat** for creating EVM Testnet interactions

3.2.4 Hardware Configuration



Device	Description
Processor	Intel(R) Core(TM) i74005U CPU @ 1.70 GHz 1.70
RAM	8 GB
System Type	64-bit Operating System.

Table 1: Hardware requirements

3.2.5 Software Configuration

Operating System	**Windows**
Language	**Javascript/JSX**
Package manager	**Node package manager (NPM)**
Runtime environment	**Node.js + Hardhat**

Table 2: Software requirements

3.3 Website Design

Header containing the name of the website along with a navigation bar. The navigation bar has links to "Home" , "purchased" and "my-resales" pages.

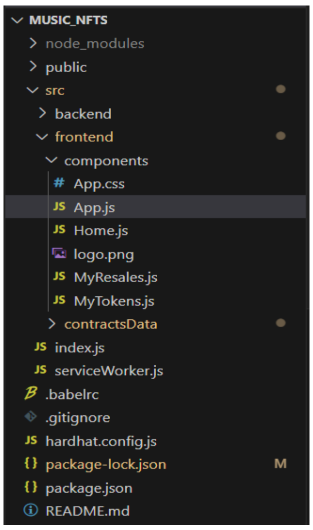
Body which has different contents for each page. Eg:- In the "purchased" page body contains the purchased NFTs while in the "my-resales" page it shows what NFTs have been sold.

3.4 Front-end implementation

3.4.1 Bootstrapping a basic react application

We need to run npx create-react-app appname and it will automatically create a folder with everything needed to create a React app [1].

3.4.2 React Workspace and folder hierarchy



```
v MUSIC_NFTS
  > node_modules
  > public
  v src
    > backend
    v frontend
      v components
          App.css
          App.js
          Home.js
          logo.png
          MyResales.js
          MyTokens.js
      > contractsData
    index.js
    serviceWorker.js
  .babelrc
  .gitignore
  hardhat.config.js
  package-lock.json                M
  package.json
  README.md
```

Fig 3: Bootstrapped react app made using npm create-react-app command

3.4.3 Installing packages and dependencies

In React, many packages and dependencies can be installed using the following command:
npm install package name - to install regular dependencies.
npm i - Install all dependencies at once.

3.4.4 Importing and exporting components

In React, we can export a particular component and then import it into the parent component to reuse it many times.

Importing :
Importing a component - import Home from './Home.js'
Importing dependencies - import { ethers } from "ethers"
Importing hooks - import {useState} from ‘react’;

Exporting :
We can export a component as follows :

```
export default function App() {
  return (
    <Home />
  );
}
```

3.4.5 How react works (JSX and Babel) :

JSX is short for JavaScript XML. JSX lets you write html in javascript. This is what makes React powerful and clean.

An expression in JSX : `const myMatter = <h1>Install Node.js </h1>;`

A block of html in react can be written as :

```
const myMatter = (
  <div>
    <p>First para</p>
    <p>Second para</p>
  </div>
);
```

What is Babel ?

Created by Sebastian Mckenzie, BabelJS is a JavaScript transpiler that converts new features into regular expressions. With this it is easy to use features of both old and modern browsers. It is used to write the JSX used by React.

The language that browsers understand is JavaScript. We use a variety of browsers to run our apps, including Chrome, Firefox, Internet Explorer, Microsoft Edge, Opera and UC Browser. The language of JavaScript is called ECMAScript; The latest stable version, ECMAScript 2015 ES6, is compatible with old and new browsers.

Since ES5 we have ES6, ES7 and ES8. There are many new features in ES6, but not all browsers support them. Same goes for ES7, ES8 and ESNext (upcoming ECMAScript release). It's not clear when all browsers can use each version of ES release.

If we want to use the new ECMAScript features and run it in all existing browsers, we need a program to write our final code in ES5. Babel was designed to solve this problem.

3.4.6 States in React

In React, every change made by the user is considered a state change. A state contains information about the objects it is in. Every time we change the state of the object, it is reset with the new state. The SetState() constructor is used to change the state of an object. For example, if we type in search, the state changes for each letter and the object has to be repeated.Example :

```
Class MyClass extends React.Component {
   constructor(props) {
      super(props);
      this.state = { attribute : "value" };
   }
}
```

3.4.7 Props in react

Props is short for things. It works like an HTML character. Objects in React are similar to state, but the main difference between state and object is that objects can be passed from parent object to child object. This process is called "propeller drilling".

Eg:-

Adding an attribute called ‘brand’ to ‘Vehicle’ component :

```
const Ele = <Vehicle brand="Toyota" />;
```

Passing the prop to the component :

```
function Truck(data) {

  return <h1>The price is : { data.price }</h1>;

}
```

3.4.8 React Hooks

React version 16.8 introduces hooks. This is done to modify the class object. Hooks allow us to access partial state and other React features. The state can be changed via hooks.It also helps us to ‘hook’ into the lifecycle methods. The programmer needs to import the hook before using it. Hooks can be imported by the following line of code :

- import {useState} from ‘react’;

Below is the list of the most frequently used react hooks :

1. **useState hook :** It is used to set and update the state of a component.

   Used to toggle/change the state of a component. Basically, if something changes in the React app, its state changes. If we type something in the search field, its state changes and every time the state changes the page returns.

2. **useEffect hook :** It is used to perform a specific task after an event has occurred.

   The useEffect hook takes two parameters: an anonymous function and an array. When the parameters in the array are triggered, the code in the anonymous function is executed. When a state changes, we increment the value of the count variable.

   Syntax of useEffect hook :

   ```
   useEffect( ( ) => {

           },   [dependency_if_any]

            ) }
   ```

3. **useContext hook :** Used to prevent column drilling. An element has a state that can be accessed by all objects, regardless of how many parent objects are on it.The status should be held by the highest parent in the group who needs to access the status. We have many nested components. Components at the top and bottom of the stack must access state.To do this without context, we need to pass the state to "props" for each link. This is called "propeller drilling".

4. **useRef hook :** Explain the concept of unregulated features. Uncontrolled components are controlled by the DOM itself, not by the reactive state. References are used to interact directly with the real DOM (Data Model) rather than the virtual DOM. References do not result in rework. Let's say we want to use the "focus" feature that exists in javascript but not in JSX.Using the references we can get the first DOM node and use the focus property on it. This is a very useful reference to useRef when the button is clicked and you want

the input to focus. To do this we first need to access the DOM entry and then call its focus() function. To do this in JavaScript, select the input using querySelector or by id/class, then call the focus() function. But React has no way to do this.

5. **useMemo hook :** It is used to stop unnecessary repetition. When the parent's state changes, the child object needs to be rebuilt, even if its state has not changed. This makes memory redundant. The useMemo hook prevents this behaviour. Consider a static greeting card that will appear in the application. Other states such as counters are also available in the app. The welcome card is a sub-application of the main parent application, so when the counter increases, the static card will be updated with the changes in the internal state of the form, please.

6. **useCallback hook :** It's similar to using a memo. Returns the noted function, while UseMemo returns the noted value. UseMemo does not work when accessories are passed from parent to child. In this case, callback hook is preferred.

   Pass an array of dependencies with an inline callback. What implementCallback returns is a memory version that changes only when one of the callback's parameters changes. This will help kids get back to being the best, who don't have to do everything because they trust equality.

React hooks lifecycle :

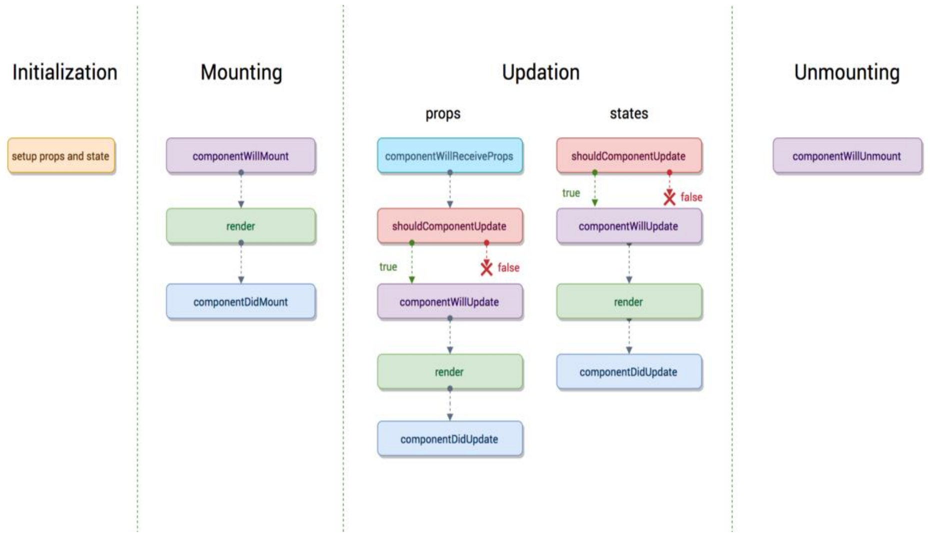


Fig 4 : React hooks lifecycle

Mounting, Updating, and Unmounting are the three phases that make up the lifecycle of a React component. Each level has a special path that is responsible for a certain level in the physical component. Technically, this method does not work with functional objects, only with class objects.

But thanks to the introduction of the Hooks concept in React, you can now use this living system's changes when interacting with provincial functionality. In a nutshell, React Hooks are functions that can be "pinned" to React's state and lifecycle of functional objects.

Phases of React component lifecycle :

During the Mounting step, a new component is created and added to the DOM, which also marks the beginning of the component lifecycle. This is often called "start processing" and can only happen once.

Components are updated or redone during updates. This reaction is triggered when the status or equipment is updated. This phase will happen over and over as part of React's purpose.

The unmounting step, which removes a fragment from the DOM, is the final stage of a fragment's lifecycle.

For each stage of its lifecycle, the object class can be called in different ways (more on that below). Only this lifecycle method created by or included in the class is not explicitly used by the functional object. Functional objects, on the other hand, can take advantage of the situation when using React hooks.

3.4.9 Ethers.js

Ethers.js is a web 3.0 library that can be used to interact with smart contracts on the Ethereum blockchain and other Ethereum Virtual Machine (EVM) over blockchains.Ethers.js and React create a front-end web application that can:

1. Connect to Metamask wallet in a browser
2. Read wallet balances
3. Access existing blocks on the Ethereum network
4. Listen for blockchain updates on the Ethereum network
5. Read data from smart contracts
6. From Metamask write operations for smart contracts

Installation : npm install --save ethers

To connect Metamask wallet to React app, we will first create a new React component to connect Metamask to React app using Ethers.js.

```
import React, { Component } from 'react';

class Metamask extends Component {

   constructor(props) {

       super(props);

    }
render() {
   return(
      <div>
      </div>
  )
 }
}
export default Metamask;
```

3.5 State Management using React-redux :

Redux is one of the most important tools in React. It is an open source javascript library to store all state in one central place. It is used for state administration only. When there is more than one state, redux stores all states in a central place called the Redux Store. From there, "actions" are sent to modify the data.

Redux provides a simple and convenient way to manage state. It works on the principle of "one-way data flow". It helps to evaluate and better manage the application.

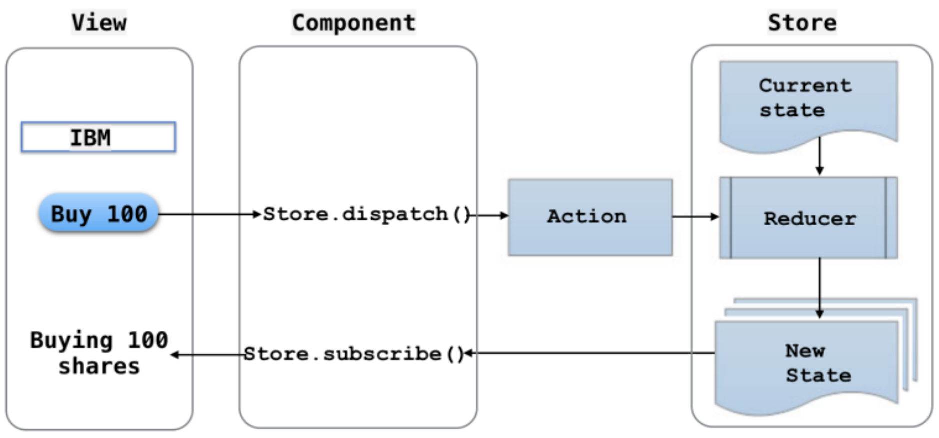


Fig 5 : The flow control of redux

The three main components of Redux are :

Redux Store : Maintain the current state of all objects. When we need information, we need to go to the Store. The reducers then update the data and send an action. Then store the new file in the Redux Store.

Creating a Redux Store :

```
const store = createStore(reducers)

ReactDOM.render(

   <Provider store={store}>

      <App />

   </Provider>,

 document.getElementById('root')

);
```

Action : Operations are simple objects with two properties, type and payload. An attribute payload is an optional item that contains some information required to complete a task, while an attribute type is usually a description of the action. Sending data from the app to the Redux store is the main purpose of the process.

Reducers : Reducers are pure functions that adjust the state of the application based on user input. Reducers perform an action and change state input and output state. Due to state invariance, reducers always create a new state that represents a new version of the previous state. It is mainly used to perform some action and update existing data/state in the Redux Store.

Data flow in a Redux application :

- Data flows happen when the user interacts with the product. As a result of this interaction, it sends a constructive "action".
- When an order is sent, it is received by the central reducer of the application and sent to all other reducers. Therefore, depending on the order shipped, it is the responsibility to determine whether to update the case or not.
- This is defined using a simple keyword to filter related jobs. Every (small) reducer in the app accepts the submission and returns the current state when the submission type matches.
- It is important to remember that with Redux the situation never changes. Instead, the reducer always creates a new state that is the same as the original state but with some changes.
- The store then notifies the item of the state change, which allows it to recover the state and reuse the item.
- Another important point in this context is that the data flow is unidirectional or flowing in one direction in a React-Redux application.

3.6 Routing

One of the tools used to customise React is the React Router. It allows users to switch between multiple pages made of various React components, change the URL of the browser and keep the UI in sync with the URL.

Installation : npm install react-router-dom

Importing :

```
 import {

        BrowserRouter as Router,

        Routes,

        Route,
    Link
} from 'react-router-dom';
```

Usage : This application has three main routes. Home, Purchased and My-Resales.

3.7 Connecting Front-end and Smart Contract :

An important part of this process is the use of a DApp address contract, which acts as a unique identifier for smart contracts on the blockchain.Integrating smart contracts with the front-end can facilitate users' interaction with smart contracts and facilitate the adoption and use of smart contracts.

Using the unique DApp contract address as a port, we can use various tools to forward our smart contracts.Tools used to connect smart contract to frontend are as follows:

1. Web3.js : Web3.js is a JavaScript library that allows you to interact with the Ethereum blockchain from within your DApp. It provides a way to read and write blockchain data using a contract address, and a mechanism to listen for events broadcast by the contract.

2. Hardhat : It is a development framework that provides a framework for creating, testing and deploying smart contracts. It includes a console that lets you use contract interfaces to interact with your smart contracts and run various commands, and a test library that you can use to write tests for your contracts.

3. Metamask : [11]MetaMask is a browser extension that allows you to interact with the Ethereum blockchain through your web browser. It provides a way to sign transactions and access DApps in the browser without the need for an Ethereum base. The DApp contract is used to identify the specific contract you want to interact with.

Steps to connect your smart contract to a frontend :

1. Deploy your contract to the blockchain :

First, you need to send the smart contract to the blockchain network. You can do this with a tool like Hardhat, which creates an address in the contract that you can use to interact with your contract. Make sure you create a wallet and have cryptocurrency capital to cover the transfer. It is important to note that the address in the mainnet's contract is not the same as the address in the blockchain test's contract.

3.7.1 Blockchain Technology : Blockchain is an ever-growing list of data blocks (blocks) that are securely interconnected by cryptographic hashes. [11][12][13][14] Each block comprises of transaction data (often represented as a Merkle tree with data nodes represented by leaves), a timestamp, and a cryptographic hash of the preceding block. Timestamps prove that the changed data existed when the block was created. Since each block contains information about the previous block, they form a chain (similar to a data link structure) in which each additional block is linked to the previous block. Therefore, blockchain transactions cannot be undone because once they are recorded, the information in a particular block cannot be changed retrospectively without modifying all subsequent blocks.

A peer-to-peer (P2P) computer network often oversees a blockchain, which functions as a public distributed ledger where nodes collectively act in accordance with the additional accepted algorithm protocol and legitimate new business blocks. Blockchains can be regarded as safe in terms of architecture and distributed system instances, including strong Byzantine fault tolerance,[15] despite the fact that blockchain data cannot be changed. This is because blockchain forks are a possibility.

W. Scott Stornetta, Dave Bayer, and Stuart Haber. [16] Bitcoin is the first digital currency to leverage blockchain technology to address the issue of double spending without the aid of a trust or incorporation server. The person who created Bitcoin has been mentioned in various apps and blockchains that are commonly utilised by cryptocurrencies. Blockchain may be compared to a payment rail. [17].

It is advised to utilise private blockchains for business purposes. The business of such private blockchains without basic security has been referred to as "snake oil" by Computerworld;

[18] Others, however, have suggested that permissioned blockchains, when well built, can be more disruptive and safe than permissionless blockchains in actual use [19].

Structure and design : A blockchain is a distributed, distributed and mostly public ledger made up of data called blocks used to record transactions across multiple computers, so not all transactions can be changed without changing all blocks after that [20]. Participants may now more easily and independently detect and evaluate changes [21]. Blockchain databases are self-managed using a peer-to-peer network and distributed time servers. They are recognized by a grand coalition driven by self-interest. This design supports stable transactions where participants are not fully aware of the security information [22].

The use of blockchain eliminates the endless transfer of digital assets. It eliminates duplicate spending by allowing each unit of value to be traded just once. Blockchain is described as a value exchange. A blockchain can manage ownership because, when designed appropriately to extend the transaction agreement, it reveals what needs to be given and received.

In essence, blockchain may be thought of as having many layers:

1. Infrastructure (hardware)

2. Network (finding node, knowledge dissemination and authentication)

3. Consensus (PoW, PoS)

4. Data (transaction,blocks)

5. App (Dapp if applicable/smart contract)

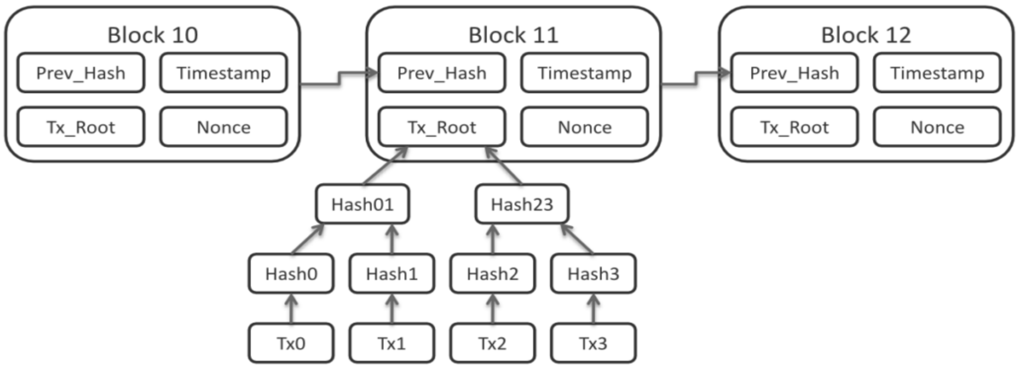


Fig 6 : Structure of a Blockchain

Blocks : Valid transactions that have been hashed and encoded into Merkle trees are stored in blocks. The cryptographic hash of the block before it on the blockchain is contained in every block and connects them. For a chain, join the blocks together. The integrity of each subsequent block up to the genesis block (block 0), the initial block, is checked iteratively in this procedure. To guarantee the accuracy of the blocks and the information they contain, digital signatures are frequently used on blocks.

In some cases, forking temporarily allows for the simultaneous creation of several blocks. Each blockchain includes a unique mechanism for scoring various historical scores so that the version with the higher score may be chosen, in addition to hash-based historical security. Orphan blocks are those that are not chosen to be part of the chain. Different iterations of the history sequence are used by competitors of the backup file. They only receive the top marks for the knowledge they possess.

When a buddy receives a better score—typically an older score with new blocks added—they may continue or gather further data and relay the upgrades to their friend. There is no assurance that a certain submission will continue to be the greatest it has ever been. Blockchains are frequently made so that some new blocks can be added to existing ones, and rewards are offered for adding new blocks rather than erasing existing ones. Therefore, when additional blocks are added to it, the likelihood of altering access will ultimately drop to a very low level.

In the proof-of-work mechanism used by Bitcoin, for instance, the chain with the most proofs of work is regarded as genuine by the network. To prove the accuracy of the computation, a variety of techniques can be utilised. In a blockchain, calculations are rerun rather than using the standard separate and equal method.

Block Time : Block time is the average time it takes for the network to create an additional block on the blockchain. When the time block is complete, the information available becomes evidence. In cryptocurrencies, this is the time when transactions occur, so a shorter time means faster transactions. Ethereum's block time is set between 14 and 15 seconds, while Bitcoin's average block time is 10 minutes.

Hard Forks : A hard fork is a change in the blockchain technology, which does not support backwards compatibility and necessitates software updates from all users in order to join the network. In a hard fork, the network splits into two groups: those who adhere to the new rules and those who adhere to the old rules.

For instance, Ethereum hard forked in 2016 for "all" DAO investors who were defrauded by a hacker who used a bug in the code. The fork in this instance causes the chains of Ethereum and Ethereum Classic to divide. To counter the loss of 50 million NXTs from significant cryptocurrency exchanges in 2014, the Nxt community was requested to propose a hard fork that would enable the blockchain data to be restored.

After negotiations and the payment of the ransom, the challenging plan was rejected and some of the money was returned. Or, similar to the Bitcoin split that occurred on March 12, 2013, most nodes using the new software would return to the previous set of rules in order to prevent a permanent split.

A recent example of a fork is Bitcoin, which led to the split of Bitcoin Cash in 2017. Network separation often results from disagreement over ways to boost transaction rates to keep up with demand.

Decantralization : Blockchain reduces some of the hazards related to centralised data storage by storing data on a peer-to-peer network. Distributed networks and ad hoc communications are both possible with decentralised blockchains.

When a central authority seizes control of more than half of the network, they may then control specific blockchain data, enabling double-spending. This is known as a "51% attack."

Public key cryptography is one of the security measures used in blockchain technology. The blockchain's address is included in the public key, which is a lengthy string of seemingly random digits.

According to the address there, the value of the tokens sent via the network is recorded. The owner may access their digital assets or engage with the numerous services currently allowed by blockchains thanks to the private key, which operates similarly to a password. On the blockchain, data is widely thought to be unbreakable.

Every part of the central system has a copy of the blockchain. Data quality is maintained by big data replication [40] and reliability.

No user is more "trusted" than another when using software to broadcast transactions to the network, and there is no such thing as a "legal" copy. The utmost effort is made while sending messages. The initial blockchains were reliant on power-hungry mining nodes to authenticate transactions, add the blocks they produced, and broadcast the finished blocks to other nodes.

08 Blockchains utilise a variety of tools, such proof-of-work, to validate transactions. Agreements that follow contain evidence of stake. As the cost of the computer programmes needed to analyse huge data rises, the emergence of decentralised blockchains raises the possibility of centralization.

Finality : Last but not least, believe that a solid block that was just added to the blockchain won't be taken out ("firmable") in the future. The majority of distributed blockchain systems, regardless of proof-of-work or proof-of-sight, rely only on "certainty of probability": a new agreement is unlikely to alter or return.

Proof-of-stake, based on Byzantine fault tolerance, is designed to provide the so-called "final end": one validator requests a random block, and the remaining validators vote for it. Blocks are irreversibly linked to the blockchain. A modification of this approach, "end-to-end transactions", is used in operating systems such as Ethereum's Casper protocol, validators that sign two variables simultaneously on the blockchain will be "hacked" to. Leveraged stock fell.

Openness : Public blockchains are easier to use than some traditional ownership records, which, although being accessible to the public, call for physical access to examine. Since every early blockchain was permissionless, there was debate about what exactly constituted a blockchain. Whether a user's privacy policy is allowed (permitted) by the central government is one of the problems in this on-going discussion. The word "blockchain" may be used to refer to any data architecture that distributes information into time-stamped blocks, claim opponents of permissioned or private blockchains. The Multiple Version Concurrency Control (MVCC) distribution system is used by these blockchains in the database.

Just as MVCC prevents two transactions from modifying the same item in the database, a blockchain prevents two transactions from using the same item in the blockchain. 30–31 Opponents say authorization systems are similar to traditional business documents, do not support authentication, and such systems are weak in defence. Computerworld's Nikolai Hampton noted that "many blockchain solutions are little more than arbitrary data" and that "Blockchain owners should be treated with mistrust in the absence of a defined security standard.

Public Blockchain (Permissionless) : The absence of the requirement for regulation and protection from unscrupulous actors is one advantage of open, permissionless, or public blockchain networks. By leveraging blockchain as a transport layer, apps may be introduced to the network without the consent or confidence of others.

Blockchains are currently secured by Bitcoin and other cryptocurrencies by requiring proof of work for new entries. Bitcoin employs the Hashcash puzzle to grow the blockchain. Although Adam Back invented Hashcash in 1997, Cynthia Dwork, Moni Naor, and Eli Ponyatovski initially proposed the concept in their 1992 essay "Pricing by Processing or Combating Spam."

2016 saw a decline in venture capital funding for blockchain-related startups in the US but a rise in China. Blockchains that are open (public) are used by Bitcoin and many other cryptocurrencies. Bitcoin holds the biggest market share as of April 2018.

Private Blockchain (Permissioned) : Blockchains with permissions utilise an access control layer to regulate who has access to the network. It has been proposed that permissioned blockchains, rather than the prohibited blockchains that are frequently used in practice, can ensure decision-making provided they are correctly structured.

Distributed Ledger : A distributed ledger (also known as shared ledger or distributed ledger technology or DLT) is an agreement to reproduce, share, and consolidate digital information distributed across geographically (eg) different locations, countries, or institutions. Unlike centralised databases, distributed ledgers do not need a central controller, As a result, there is no single point of failure.

For a distributed ledger to be replicated over remote computer nodes (servers, people, etc.), it often requires a peer-to-peer (P2P) computer and consensus mechanism. Blockchain, which is frequently linked to the digital currency Bitcoin, is the most widely used type of distributed ledger technology. It can be either a public network or a private network.

Implementing DLT is hampered in part by data management. Digital transmission can also be used as a diary on occasion; this is referred to as replica log technology (RJT).

Disadvantages of Private Blockchain : There is no need for a "51%" block on a private blockchain because the private blockchain (likely) already controls 100% of all block production, according to Nikolai Hampton's statement in Computerworld. Control 100% of their network if you can kill or destroy the blockchain. You can use and make any changes to the equipment on private corporate servers. Data security requires Gigawatts of computer power, which is time-consuming and expensive. Furthermore, "Private blockchains lack 'competition'; there is no incentive to consume more capacity or locate blocks quicker than rivals.

Blockchain analysis : Analysis of public blockchains is now crucial due to the rise in popularity of cryptocurrencies like Bitcoin, Ethereum, Litecoin, and others. Anyone with the necessary knowledge may gain access to a public blockchain in order to examine and evaluate information. It has proven difficult for many cryptocurrencies, exchanges, and institutions to comprehend and access cryptocurrency movements. This is due to the usage of blockchain-enabled cryptocurrencies in the criminal shadow economy to fund the purchase of narcotics, firearms, and other goods. Cryptocurrencies are thought to be secret and unchangeable, which leads to numerous crimes. That has changed, and now specialised IT businesses are providing blockchain services, increasing awareness of cryptocurrencies and fiat-to-crypto transfers among banks, law enforcement, and financial institutions. Some people think that this change has caused crooks to favour using new cryptocurrencies like Monero. The issue is the privacy of the same data and public access to blockchain data. The major argument surrounding cryptocurrency and blockchain is this.

Standardisation : Standards Australia asked the International Organisation for Standardisation (ISO) in April 2016 to take into consideration developing standards to accommodate blockchain technology. The concept sparked the development of Blockchain, distributed ledger technology, and ISO technical committee 307. The working group covers vocabulary, application architectures, security and privacy, identification, smart contracts, governance and interoperability of blockchain and DLT, and standards for corporate and governmental regulations in particular. Along with external organisations like the International Telecommunication Union (ITU), the European Commission, the International Federation of Surveyors, the International Monetary Fund Organisation (SWIFT), and the United Nations European Organisation, more than 50 countries took part in the design process.

Working on blockchain standards are several national and open standards organisations.These organisations include the National Institute of Standards and Technology (NIST), the Advanced of Structured Information Standards (OASIS), the Institute of Electrical and Electronics Engineers (IEEE), the European Committee for Electrotechnical Standardisation (CENELEC), and some members of the Internet Engineering Task Force (IETF).

Centralise BlockChain : Despite the fact that the majority of blockchain applications are dispersed and decentralised, Oracle has highlighted the value of the blockchain language in its Oracle 21c database. The primary blockchain with immutable features in Oracle 21c Database is the blockchain table. Centralised blockchains are frequently more transparent and have lower transaction latency than consensus-based distributed blockchains, in comparison to decentralised blockchains.

USES : The use of blockchain technology is widespread. The original application of blockchain was as a distributed ledger for digital currency like bitcoin; by the end of 2016, a few more products were emerging from proofs of concept. In order to assess the effect of blockchain on back office processes, numerous businesses are testing the technology as of 2016.

A total of $2.9 billion, an increase of 89% from the previous year, was invested in blockchain technology in 2019.

Furthermore, according to International Data Corporation, market spending on blockchain technology will total $12.4 billion by 2022. The two biggest network service providers in the world, PricewaterhouseCoopers (PwC), also believe that blockchain technology has promise. by 2030, creating jobs worth more than $3 trillion yearly.

Blockchain was recognised as a technology that will have an economic influence and significance in 2019 by the BBC World Service radio programme "Fifty Things Shaping the Modern Economy" and podcast.

Tim Harford, an economist and publisher of the Financial Times, discusses below why the technology will be used more frequently and what obstacles need to be addressed. June 29, 2019.

The number of blockchain wallets doubled to 40 million between 2016 and 2020. The usage of blockchain technology in sustainable management is covered in a 2022 article.

Cryptocurrencies : The majority of cryptocurrencies record transactions using blockchain technology. For instance, blockchains constitute the foundation of both the Ethereum network and the Bitcoin network.

Silk Road, a criminal company operating on Tor, used cryptocurrencies for payments, and some of them were confiscated by the US government through blockchain exploration and seizure.

The government has a mixed policy of legalising its citizens or banks to own cryptocurrencies. China is using blockchain technology in many industries, including the country's digital currency, which will be launched in 2020.

Western governments, including the European Union and the United States, have created similar programs to bolster their interests.

Financial services : Many banks have reportedly indicated interest in adopting distributed ledgers for transactions and collaborating with businesses to construct private blockchains, which is faster than anticipated, according to Reason, which cites a September 2016 IBM study. is taking place.

This technology has a special appeal to the bank since it can provide back office solutions. Additionally, if the institutional use of blockchain technology grows and the sector develops early, it is effectively the foundation for a brand-new financial sector, with all the advantages it entails. [96] Banks such as :

UBS is opening new research centres dedicated to blockchain technology to explore how blockchain can be used in financial services to increase efficiency and reduce costs.

Deutsche Bank Berenberg believes that blockchain is an "overrated technology" that already has a lot of "proof of concept" but still faces major challenges and a few myths of successful people.

Initial Coin Offerings (ICOs) and Security Token Offerings (STOs), also known as Digital Security Offerings (DSOs), are a new class of digital assets that have emerged as a result of the Blockchain. STO/DSO can be utilised as new assets like property, real estate, artwork, or personal property, as well as assets like joint ventures. They can be listed on a private or regulated public market. In this industry, a large number of businesses provide services for implementing tokenization, private STOs, and public STOs.

Other applications : Blockchain technology may be used to build a safe, open, and transparent ledger system to track digital consumption, record sales data, and compensate content creators like musicians or cellular customers. In the next 24 months, 2 percent of higher education participants have launched a blockchain project, and 18 percent plan to create a course, according to Gartner's 2019 CIO Survey. In order to use blockchain technology for music distribution, IBM teamed up with ASCAP and PRS for Music in 2017. To "give artists control over how their music and related information is shared between fans and other musicians," Imogen Heap's Mycelia service has been suggested as a blockchain substitute.

The novel distribution approach may be used to peer-to-peer insurance, parametric insurance, and microinsurance in the post-blockchain insurance sector.

Because they include several partners and are supported by the American Museum and Library Services Association, the sharing economy and the Internet of Things will also profit from blockchain.

Other blockchain models include Hyperledger, a project of the Linux Foundation that supports distributed ledgers based on the blockchain. Hyperledger Burrow was created by Monax, while Hyperledger Fabric was led by IBM.

The Quorum private blockchain, which has space for JP Morgan-approved contracts, is a further option. The Oracle 21c database now features blockchain tables from Oracle.Energy peer-to-peer exchanges have also leveraged blockchain technology.

By linking distinctive identifiers to items, papers, and products and storing false or unchangeable business-related information, blockchain may be used to detect counterfeit goods. However, it has been stated that blockchain technology should be supplemented by tools that show how actual assets relate to the blockchain system and content production guidelines that demonstrate Ala KYC requirements.

To "define, pilot, and implement" an anti-fraud system at the European level, EUIPO established the Anti-Fraud Blockchain Marathon Forum. The Dutch standardisation organisation NEN verifies certificates using QR codes and blockchain. Beijing and Shanghai become the nation's blockchain-focused cities on January 30, 2022.

Blockchain Interoperability : Blockchain interoperability has become a prominent problem as more and more blockchain systems, even ones that just support cryptocurrencies, appear. The goal is to facilitate asset transfers across blockchain systems. According to Wegner, "interoperability is the ability of two or more softwares to collaborate despite differences in languages, connections, and functions." Therefore, despite these differences, the goal of blockchain interoperability is to promote this kind of cooperation amongst blockchain systems.There are currently several blockchain interoperability options available.

They fall into three categories: blockchain connections, blockchain engines, and cryptocurrency interactions. A draught of a blockchain interoperability architecture has been created by many IETF participants.

Energy Consumptions Concerns : Blockchain mining, which involves computations made by peer-to-peer computers, is a technique used by several cryptocurrencies to confirm and validate transactions. This has to be really powerful. The BIS questioned the widespread use of public proof of work in blockchains in June 2018.

Initial worries regarding further blockchain adoption, including Cardano (2017), Solana (2020), and Polkadot (2020), were emphasised. models. According to researchers, a proof-of-stake network requires 100,000 times less energy than Bitcoin.

According to study from Cambridge University, in 2021, Argentina (121 TWh) and the Netherlands (109 TWh) both consumed less power than Bitcoin (121 TWh). A Bitcoin transaction, according to Digiconomist, needs 708 kilowatt-hours of power, or about what a typical American home uses in a day.

Bitcoin was referred to be a "inefficient way of doing business" by US Treasury Secretary Janet Yellen un February 2021, who also noted that "the power used to make these transactions is incredible." Bill Gates stated that "Bitcoin as a business consumes more energy than any other human experience," and that "that's not a good thing" in March 2021.

In his research on the effectiveness of working on public proof-of-work blockchains and the online security of blockchains, Nicholas Weaver of the International Institute of Computer Science at the University of California, Berkeley, discovered that these factors alone are insufficient. 17–23 million tonnes of CO2 were created in 2018 by the 31–45 TWh of power needed for bitcoin.

The two biggest proof-of-work blockchains, Bitcoin and Ethereum, are expected to use twice as much power as 120 million tonnes of carbon dioxide annually by 2022, according to estimates from Cambridge University and Digiconomist.

A proof-of-stake paradigm is being considered by several cryptocurrency developers in place of the current proof-of-work methodology.

3.7.2 Smart Contracts :

The conditions and acts of a promise or agreement are executed, controlled, or recorded by a smart contract, which is a computer programme or transaction protocol. Smart contracts are designed to lessen dispute and violence as well as the need for trusted agents, arbitration costs, and fraud. Cryptocurrencies and smart contracts are frequently linked, and Ethereum's smart contracts are frequently seen as the foundation for decentralised finance (DeFi) and NFT application forms.

The most technologically comparable device to the usage of smart contracts is a vending machine. The Bitcoin protocol was referred to in Vitalik Buterin's original Ethereum whitepaper from 2014 as a "weak" implementation of Nick Szabo's "smart contracts," and a "strong" implementation based on the Solidity programming language was presented as a supplement to Turing.

Numerous cryptocurrencies have enabled programming languages that enable smarter contracts between untrusted parties ever since Bitcoin [clarification needed].

Smart contracts, as opposed to regular, well-defined, legal contracts that choose words that are stated and acted upon in a machine-readable fashion, should not be confused with smart contracts.

Smart contracts are intended to be both mechanical and legal, even though they are typically not considered to be legal contracts.

The performance of contractual duties that result from voluntary application through a computer or contractual exchange, methods of payment for obligations, or automation of obligations on the exchange of tokens or cryptocurrencies are all examples of smart contracts. A smart contract is not a formal contract. According to some experts, the significance or acceptance of programming languages may have an impact on the legality of smart contracts.

Working of Smart Contract : Smart contracts are added to the blockchain in a manner similar to how transaction fees are done on it. Wallet transactions are sent to the blockchain. The exchange consists of a smart contract programmed with the recipient's specific address. The transaction must then be included in a block that is uploaded to the blockchain at the time the smart contract code is run to establish the smart contract's initial state. Byzantine fault-tolerant algorithms guard against the compromising of smart contracts in a distributed system. A smart contract cannot be modified once it is sent.

Arbitrary states and calculations can be stored and carried out using blockchain smart contracts. Smart contracts are used by end users in transactions. These smart contract transactions have the ability to trigger further smart contracts. The situation will alter as a result of these modifications, and tokens will be transferred from one smart contract or account to another or between other smart contracts.

Ethereum is the most widely used blockchain for executing smart contracts. Bamboo, IELE, Simplicity, Michelson (compiles with Coq), Liquidity (compiles for Michelson), Scilla, DAML, and Pact are a few languages that emphasise proof-of-concept.

Most people agree that the blockchain protocol offers Byzantine fault tolerance. Secure randomness is necessary for smart contract use in the real world, nevertheless, such as in lotteries and casinos. In actuality, blockchain technology decreases the draw's cost, which is advantageous to the participants. Block hashes or timestamps, oracles, contract designs, unique contracts like RANDAO and Quanta, and systems from strategic hash Nash equilibria may all be used to provide the unpredictability that characterises blockchains.

Transaction data is accessible to all blockchain users via smart contracts built on top of the blockchain. The data reveals the changes, yet this creates a situation where the incorrect (including unlawful) is clearly obvious to the public but won't be corrected anytime soon.

In June 2016, this difficult-to-fix hack was successfully launched on The DAO. The developers spent over $50 million worth of ether trying to find a solution that would result in successful acceptance. Before the DAO programme, it was challenging for Ethereum software to recover money from attackers. The DAO programme has extended the time it takes for hackers to remove assets. The difference between multi-wallet signature assaults and the quantity of missing/cross-attacks (2018), totalling more than $184 million, are two more prominent attacks.

The issues with Ethereum's smart contracts, particularly the confusing and insecure contract architecture, Collaboration, compiler problems, issues with the Ethereum Virtual Machine, blockchain network assaults, flawed immutability, and a lack of a centralised data centre. Find issue patterns, exploits, and vulnerabilities.

2. Connect your Front-end to the contract :

Next, we need to connect your frontend to the blockchain network. You can do this with a library like Web3.js, which then allows you to communicate with the Ethereum blockchain from your web browser and helps to connect with the blockchain and store addresses in the contract.

3.7.3 Web3.js :

Centralization has helped millions of people join the World Wide Web and create a stable, secure environment where it exists. Meanwhile, the few centralised organisations that unilaterally decide what is and are not allowed are firmly rooted in the vastness of the World Wide Web.

The solution to this query is Web3. Web3 is a user-created, user-operated, and user-owned platform that accepts dissemination as opposed to the web being monopolised by large tech businesses. Web3 gives individuals, not corporations, the ability to make decisions.

Gavin Wood, an Ethereum co-founder, first used the term "Web 3.0" immediately after the cryptocurrency's 2014 introduction. Gavin provided a solution to an issue that many ex-crypto users had: the web required a lot of confidence. However, a large portion of the internet that people today are familiar with and use depends on a private firm to act in the interests of the general public. For a fresh and improved view of the internet, Web3 has expanded to include everything. At its foundation, Web3 leverages blockchain, cryptocurrencies, and NFTs to give consumers ownership authority. The Twitter tweet from 2020 offers the finest presentation: Web1 will be read-only, Web2 read-write, and Web3 read-write.

Principles of Web3 :

Web3 is decentralised : Ownership is not a large part of the internet that the site manages and owns, but is shared between its creators and users.

Web3 is permissionless : Everyone has an equal opportunity to participate in Web3 and no one will be left behind.

Web3 has native payments : it uses cryptocurrency for spending and sending money online instead of relying on the outdated infrastructure of banks and payment processors.

Web3 is trustless : It works using incentives and business processes rather than relying on trusted third parties. Web3 sends money directly in the browser using tokens like ETH without the need for third-party trust.

Besides having your data on Web3, you can have an aggregation based platform using tokens similar to company shares. DAOs allow you to control ownership of the platform and decide its future.

DAOs are technically defined as smart contract-based agreements that automate decentralised decisions in the pool (tokens). Users with tokens vote to decide the use of resources, and the rules decide whether the results of the vote are applied.

However, many Web3 communities are defined as DAOs (Decentralised autonomous organisations). These communities have achieved the separation of distribution and automation by law.

3. Specify the address of the smart contract you deployed :

In your front-end code, you must specify the address of the smart contract to be used. You can use this address to read and write information to your contract or to listen to events from the contract. You can find out by looking at the transaction hash of the transaction in a blockchain explorer like Etherscan.

Once you have the contract address, you can use Web3.js to create an instance of your smart contract. This will allow you to call their functions and get their data from the frontend.

4. Create UI elements in your front-end :

Then you can create a UI (buttons, forms, etc.) on the front-end so users can interact with smart contracts. For example, you can create a form that allows users to submit data to a smart contract, or a button that allows them to retrieve data from a smart contract.

To call a smart contract function, you can use the contract instance's .call() or .send()

method, depending on whether the function is a read-only or mutable state binding. You must pass all required arguments to the parameter.

DApp Front-end integration things to ponder :

1. Use secure key management : It is important to manage the private keys used to sign transactions and interact with smart contracts. Tools like MetaMask can help with this.

2. Keep your contract code simple : Complex contract code can be difficult to debug and maintain, so try to keep your contract code as simple and clear as possible.

3. Test your DApp : carefully As mentioned above, it is very important to thoroughly test your DApp before sending it to the mainnet. This will help ensure your DApps are working as expected and catch potential issues early.

4. Keep up to date with the latest security practices : It is important to stay up to date with the latest security best practices in the blockchain and smart contract space. This will help ensure your DApp is secure and hack resistant.

5. Monitor your DApp : While your DApp is live, it is important to monitor it regularly to make sure it is working as expected and to identify potential problems. You can use the location in the contract to monitor the activity of smart contracts and identify potential issues.

3.8 The Temporary Back-end implementation

Frameworks required : Nodejs and Hardhat

Temporary Backend setup:

Initialising Nodejs and setting up Hardhat :

First we have to install the latest version of Node Js because older versions may not support hardhat tools.

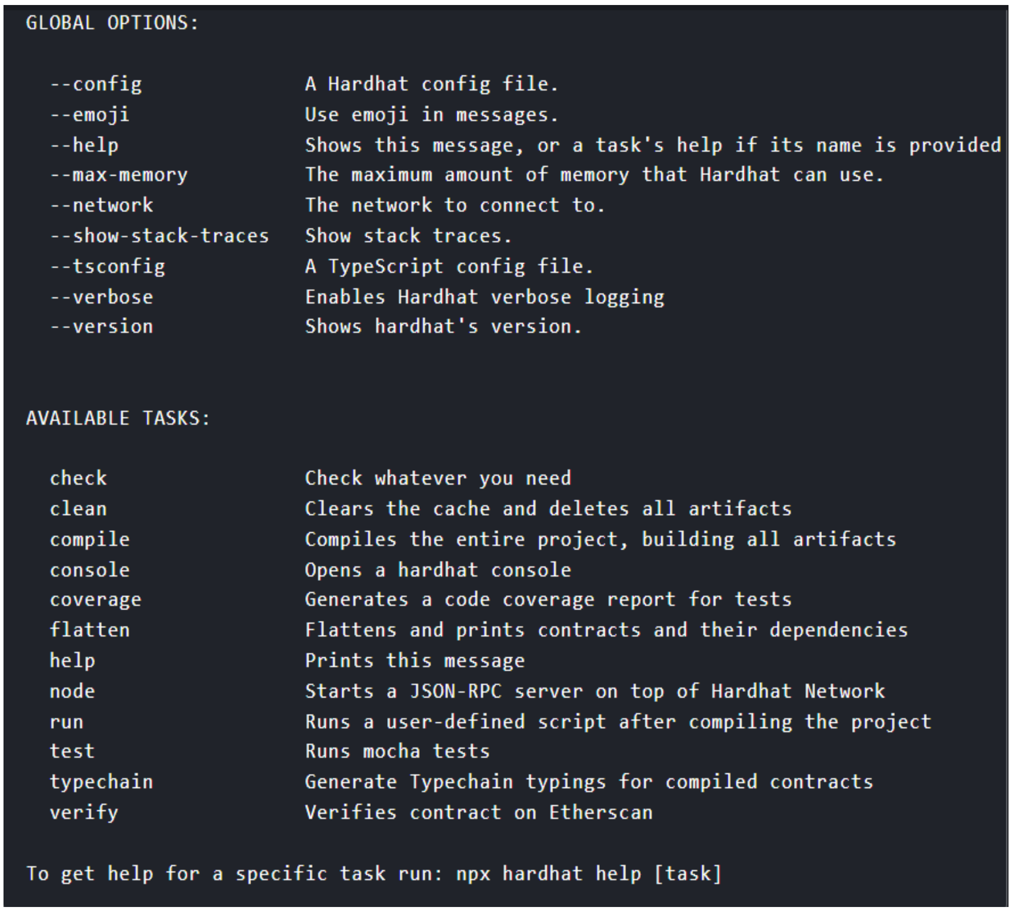
After the installation of node js we can run commands: Npm -v to check the version of node js installed and to confirm if we have Node js or not .

To install Hardhat, go to an empty folder, initialise an npm project (i.e. npm init), and run npm install --save-dev hardhat

Once it's installed, just run this command and follow its instructions:

npx hardhat

This command shows what options and tasks are available in hardhat to run.



```
GLOBAL OPTIONS:

  --config              A Hardhat config file.
  --emoji               Use emoji in messages.
  --help                Shows this message, or a task's help if its name is provided
  --max-memory          The maximum amount of memory that Hardhat can use.
  --network             The network to connect to.
  --show-stack-traces   Show stack traces.
  --tsconfig            A TypeScript config file.
  --verbose             Enables Hardhat verbose logging
  --version             Shows hardhat's version.


AVAILABLE TASKS:

  check                 Check whatever you need
  clean                 Clears the cache and deletes all artifacts
  compile               Compiles the entire project, building all artifacts
  console               Opens a hardhat console
  coverage              Generates a code coverage report for tests
  flatten               Flattens and prints contracts and their dependencies
  help                  Prints this message
  node                  Starts a JSON-RPC server on top of Hardhat Network
  run                   Runs a user-defined script after compiling the project
  test                  Runs mocha tests
  typechain             Generate Typechain typings for compiled contracts
  verify                Verifies contract on Etherscan

To get help for a specific task run: npx hardhat help [task]
```

Fig 7 : Available tasks and options in Hardhat

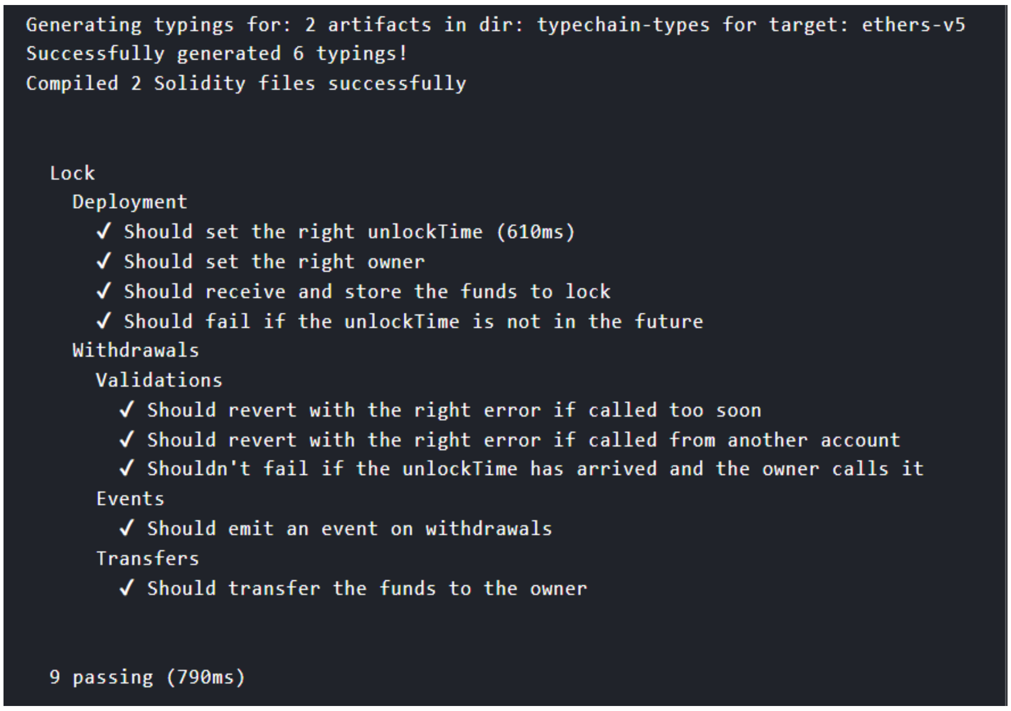
Compiling the smart contracts :

Next, if you take a look in the contracts/ folder, you'll see Lock.sol, to compile it we simply type the command :

npx hardhat compile

Next, we will find a test folder to test the compiled contract , To test we simply type the command :

npx hardhat test

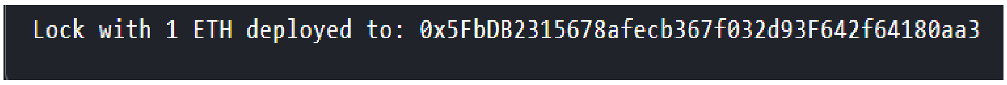


```
Generating typings for: 2 artifacts in dir: typechain-types for target: ethers-v5
Successfully generated 6 typings!
Compiled 2 Solidity files successfully


  Lock
    Deployment
      ✓ Should set the right unlockTime (610ms)
      ✓ Should set the right owner
      ✓ Should receive and store the funds to lock
      ✓ Should fail if the unlockTime is not in the future
    Withdrawals
      Validations
        ✓ Should revert with the right error if called too soon
        ✓ Should revert with the right error if called from another account
        ✓ Shouldn't fail if the unlockTime has arrived and the owner calls it
      Events
        ✓ Should emit an event on withdrawals
      Transfers
        ✓ Should transfer the funds to the owner


  9 passing (790ms)
```

Fig 8 : Testing the Contracts

Next we have to deploy the scripts for our contract by typing the command :

npx hardhat run

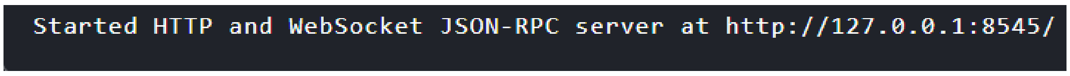


```
Lock with 1 ETH deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3
```

Fig 9 : Deployment of contracts

Finally, to connect a wallet or DApp to the hardhat network we type the following command:

npx hardhat node



```
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/
```

Fig 10 : Starting the Hardhat testnet

3.9 Running the DApp on Localhost :

Firstly, we have to connect to the Hardhat network by running the command in a separate terminal since this acts as a temporary backend for our testnet environment with the user authentication data for 20 accounts by default :

npx hardhat node

Then, we have to deploy the scripts for smart contracts by executing the command in a separate terminal:

npm run deploy

Finally , when our temporary backend is running we just have to start our website on a localhost by using command :

Npm run start

This will start the frontend of the website.

The website will ask for users to connect their metamask wallet. To connect metamask firstly we have to select the Hardhat test network in the networks tab.After which we have to use private keys from the terminal on which our temporary backend is running with the help of the command : npx hardhat node , to import the testing accounts with testing balance of 1000 ETH in every account of the testnet.

Once the wallet is connected to the metamask and the accounts have been imported the users can buy, sell the music NFTs and can also play the music on the website. The metamask will ensure safe and secure transactions and also show notifications of the transactions made. History of transactions can also be seen using the metamask itself.

If a user wants to check the information about the block on which the data is being written then the user can copy the hash of the testnet account through which the transaction is being made and check the testnet block information on the online site of "https://app.tryethernal.com/blocks" .

Chapter-4

EXPERIMENTS AND RESULT ANALYSIS

4.1 Connecting metamask :

For buying an NFT from the website the users have to first connect through the crypto wallet to the browser and only then can they see the contents of the website.

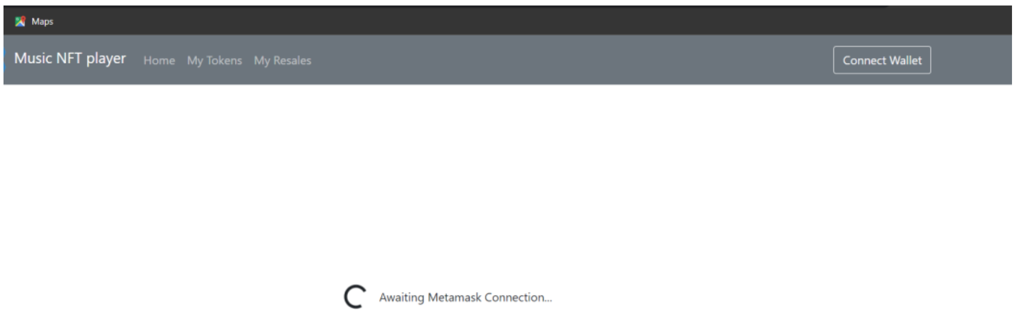


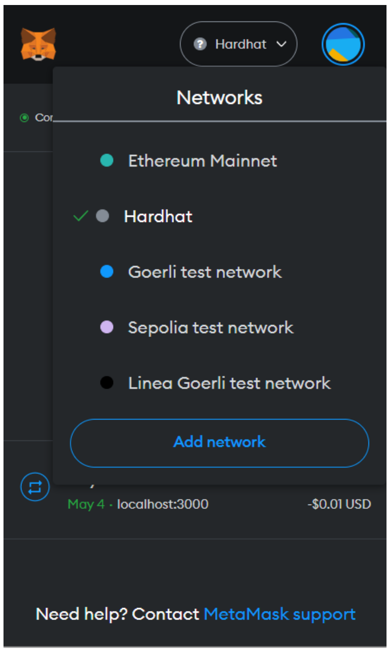
Fig 11 : Metamask connection request

The connect wallet button sends a request to the browser to open metamask or any other crypto wallet to be connected.In the metamask go to the network section and connect with hardhat network only otherwise you will be dealing with real money.

4.2 connecting to hardhat testing network :

If your metamask does not show Hardhat in the available networks then click on the Add network button and from there click on Add a network manually button.

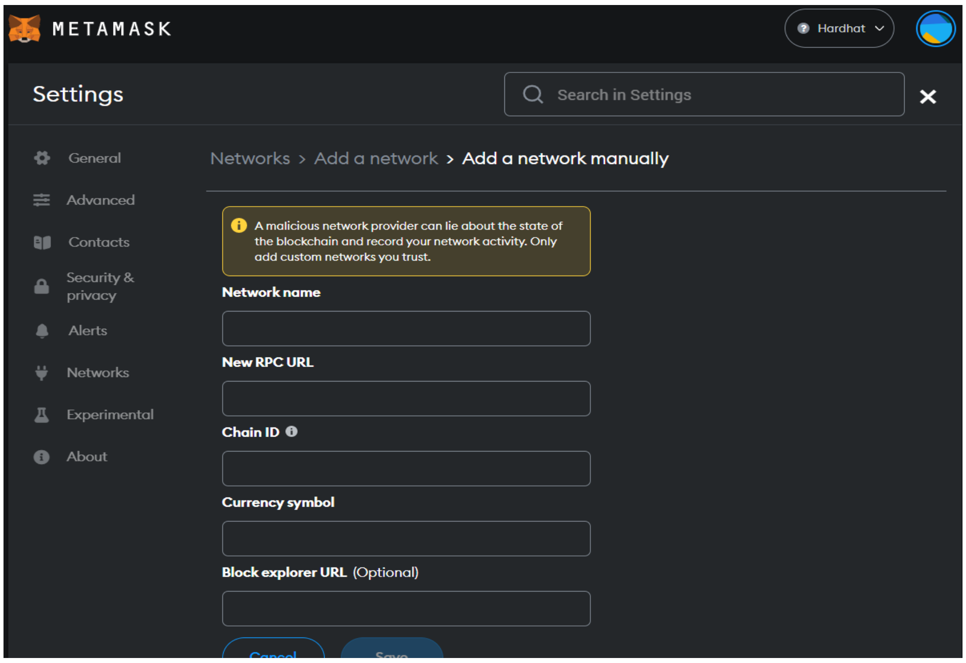
You will be directed to the metamask page for adding networks and after filling the required field with correct network information users will be able to connect to the Hardhat testing network.



May 4 · localhost:3000 -$0.01 USD

Need help? Contact MetaMask support

Fig 12 : Adding network in metamask



Networks > Add a network > Add a network manually

A malicious network provider can lie about the state of the blockchain and record your network activity. Only add custom networks you trust.

Network name

New RPC URL

Chain ID

Currency symbol

Block explorer URL (Optional)

Fig 13 : Adding Network

4.3 User interaction with the website :

Once the website is connected to the metamask the users can see and play the music NFTs and also buy them if they have sufficient balance in their testing accounts created by EVM testnet interaction of Hardhat tool using Node.js

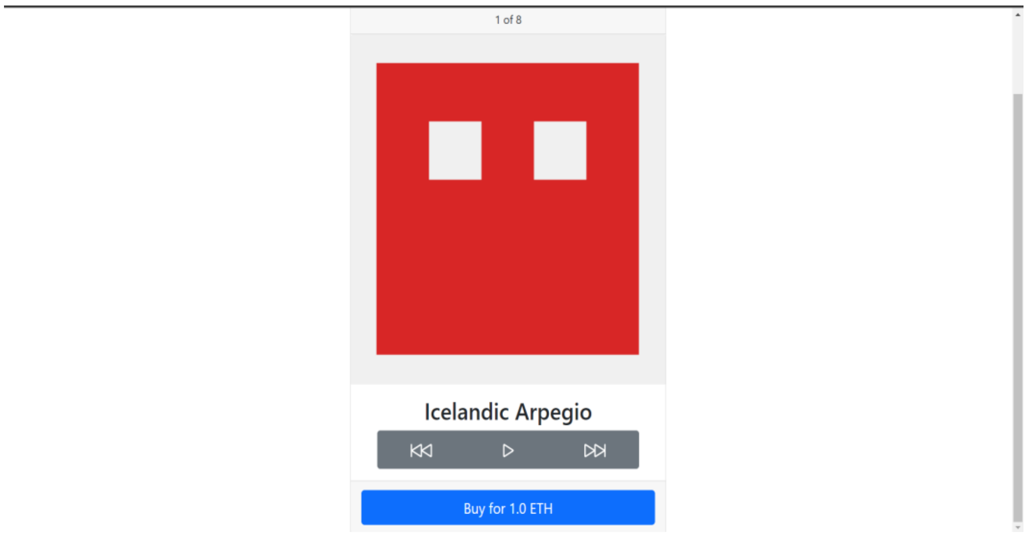


Fig 14 : Connected the crypto wallet

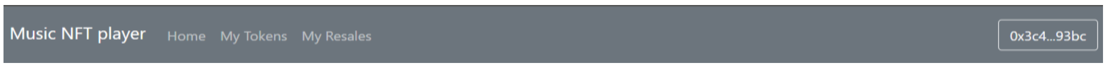


Fig 15 : The Navigation Bar

On clicking the website logo, the user will be redirected to the Juit.ac.in home page. Other than that, if a user wants to close the website they simply have to disconnect their crypto wallet .Initially the home page is visible and users can traverse to different pages through the navigation bar.

4.4 Importing the testing accounts for making transaction :

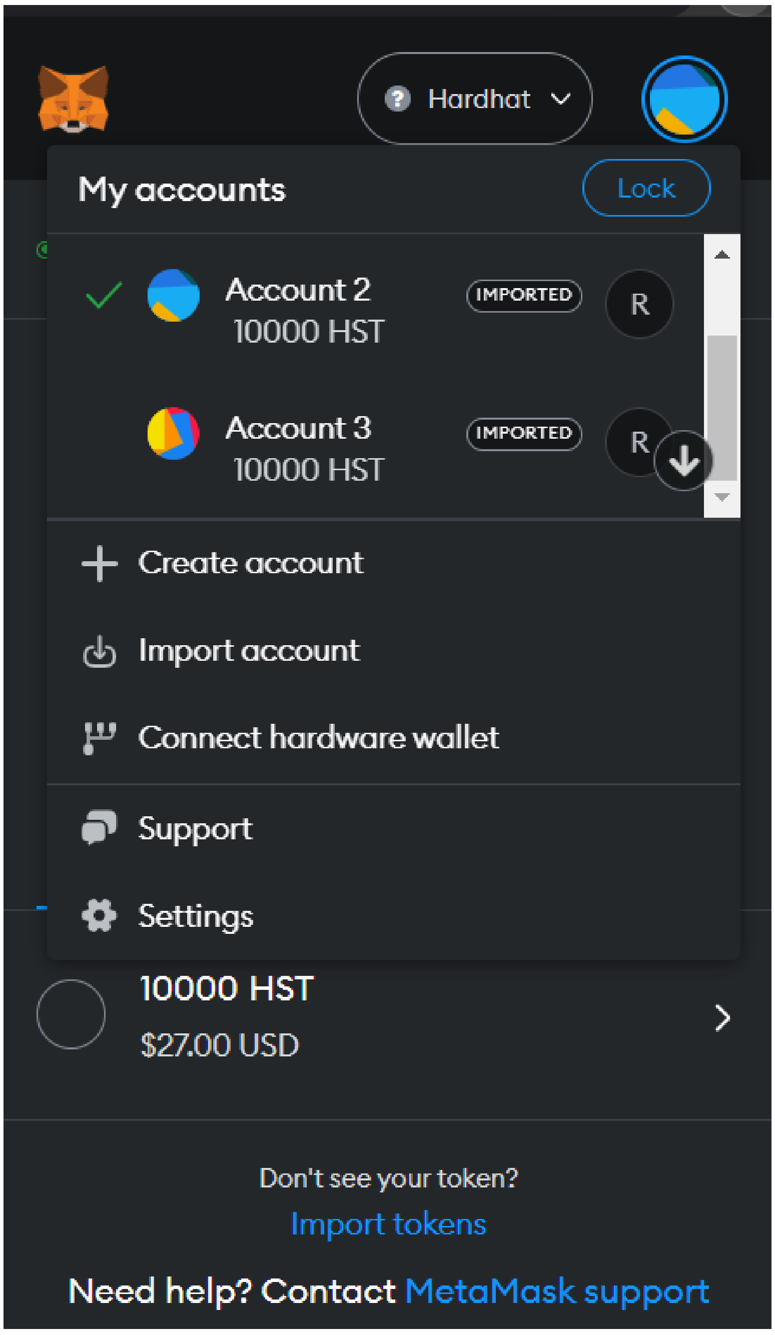
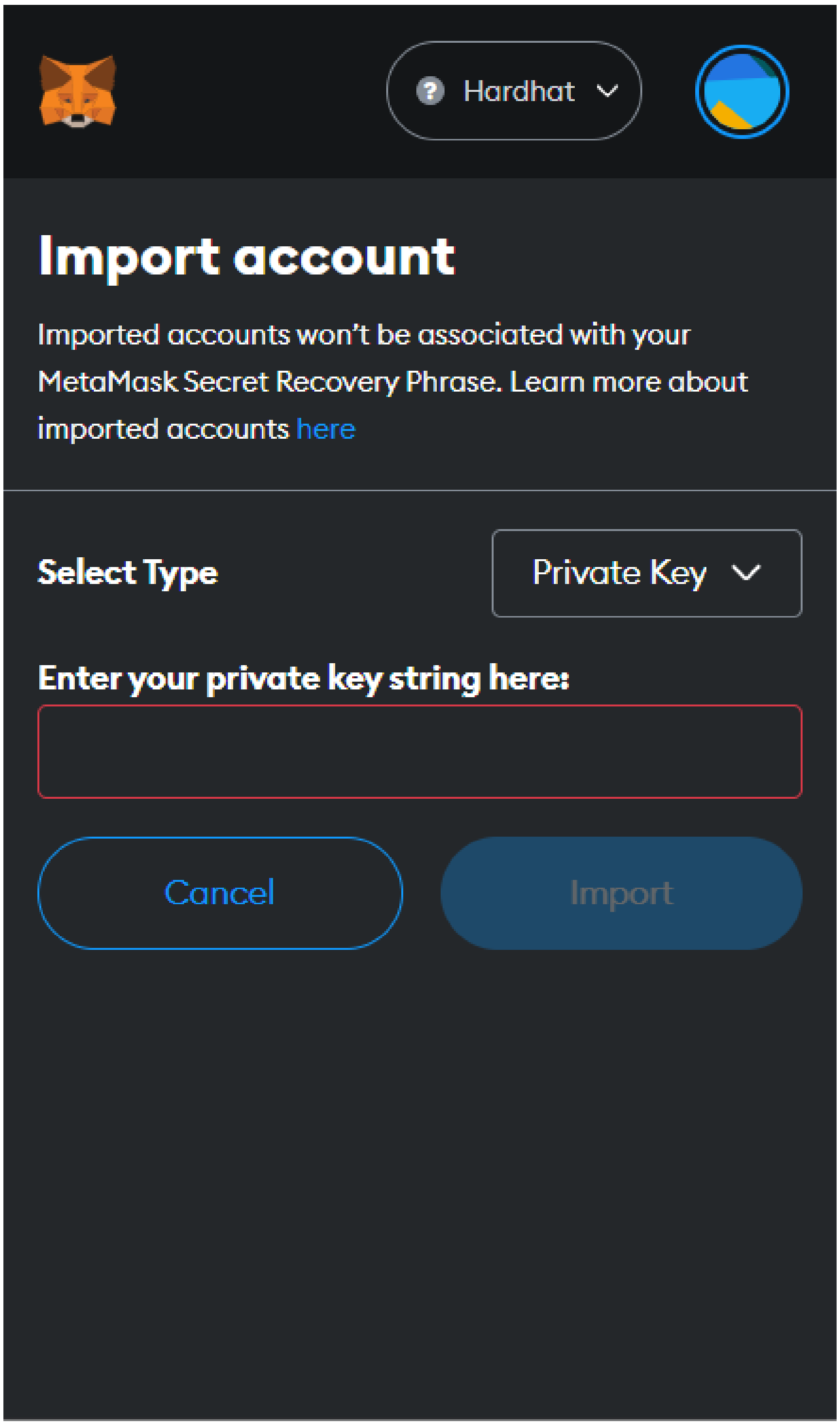


Fig 16 : Importing accounts in metamask

The users have to import the accounts provided by the hardhat testing environment by clicking on the import account option available in the metamask.



Hardhat

Import account

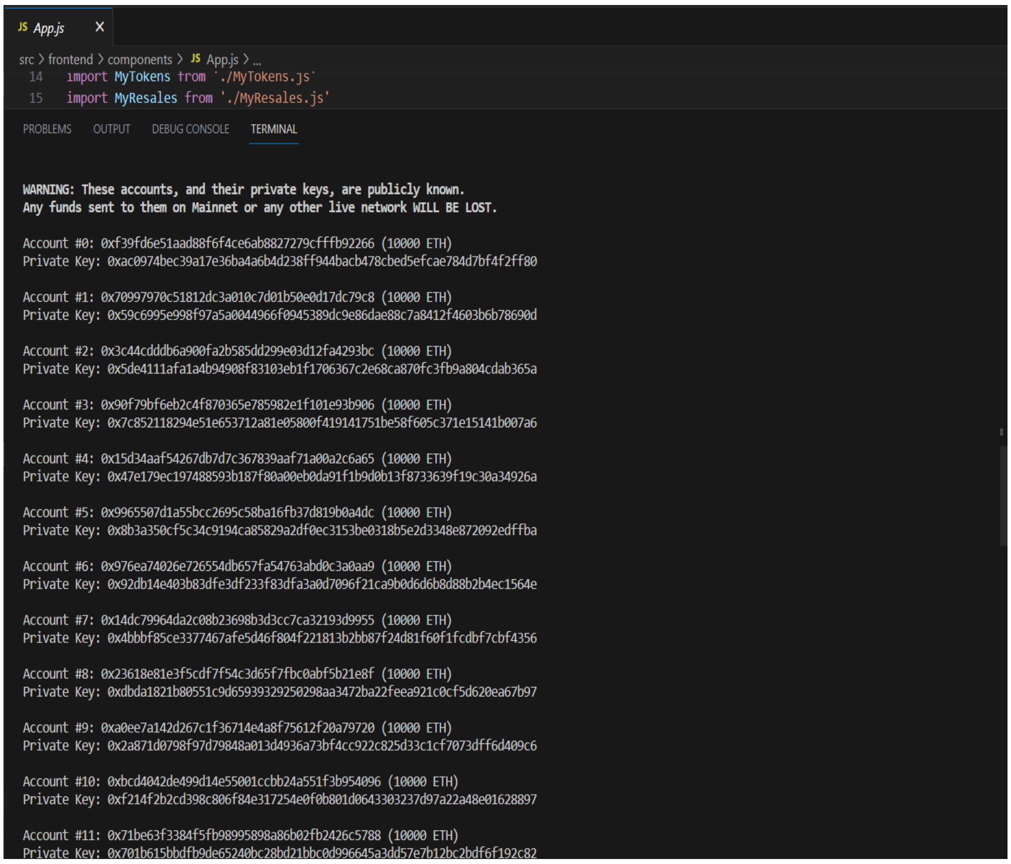
Imported accounts won't be associated with your MetaMask Secret Recovery Phrase. Learn more about imported accounts here

Select Type: Private Key

Enter your private key string here:

Fig 17 : Importing the testnet accounts

To import a testnet account we have to enter the private key of the account. This is already generated by the hardhat tool and is running in the backend. We just have to copy paste the key and the account will be imported for making transactions with a balance of 1000ETH.



```
import MyTokens from './MyTokens.js'
import MyResales from './MyResales.js'
```

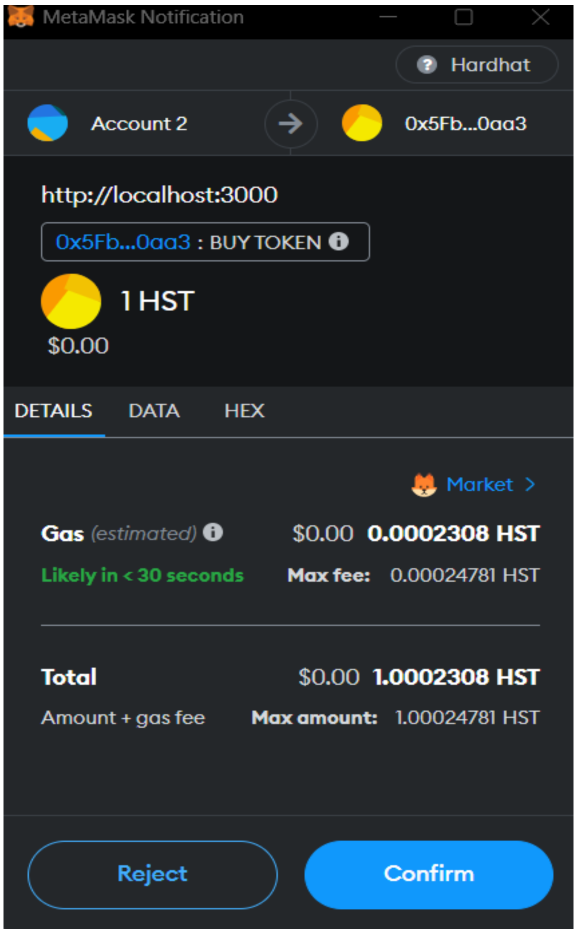
```
WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3c44cdddb6a900fa2b585dd299e03d12fa4293bc (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90f79bf6eb2c4f870365e785982e1f101e93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34aaf54267db7d7c367839aaf71a00a2c6a65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507d1a55bcc2695c58ba16fb37d819b0a4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976ea74026e726554db657fa54763abd0c3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dc79964da2c08b23698b3d3cc7ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81e3f5cdf7f54c3d65f7fbc0abf5b21e8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22feea921c0cf5d620ea67b97

Account #9: 0xa0ee7a142d267c1f36714e4a8f75612f20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

Account #10: 0xbcd4042de499d14e55001ccbb24a551f3b954096 (10000 ETH)
Private Key: 0xf214f2b2cd398c806f84e317254e0f0b801d0643303237d97a22a48e01628897

Account #11: 0x71be63f3384f5fb98995898a86b02fb2426c5788 (10000 ETH)
Private Key: 0x701b615bbdfb9de65240bc28bd21bbc0d996645a3dd57e7b12bc2bdf6f192c82
```

Fig 18 : Testnet accounts running in backend

4.5 Making transaction :

Now the users can easily buy and sell the NFTs using the metamask. The metamask is secure and always shows notification and transaction details when the user tries to buy or sell any music NFT.

When a user buys an NFT that particular music NFT is removed from the list of available music NFTs for everyone using that website since that particular NFT has been bought by someone and will only be shown in the My tokens section of that particular account holder. Once that user resells that NFT then again that particular NFT which was removed earlier from the available NFTs list will now again join that list. Hence the list updation is dynamic.



MetaMask Notification

Hardhat

Account 2 → 0x5Fb...0aa3

http://localhost:3000

0x5Fb...0aa3 : BUY TOKEN

1 HST

$0.00

DETAILS DATA HEX

Market >

Gas *(estimated)* $0.00 **0.0002308 HST**

Likely in < 30 seconds **Max fee:** 0.00024781 HST

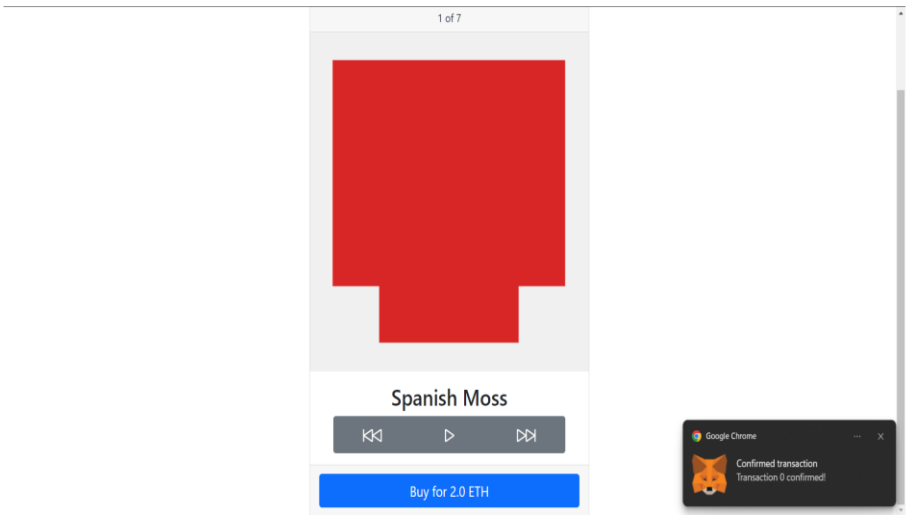
Total $0.00 **1.0002308 HST**

Amount + gas fee **Max amount:** 1.00024781 HST

Reject Confirm

Fig 19 : Transaction details

Users can accept or reject the transaction through metamask. Once the Transaction is made a notification will be sent by metamask regarding the status of the transaction.



Confirmed transaction

Transaction 0 confirmed!

Fig 20 : Notification of transaction

Since the first music NFT was bought it has been removed from the list of available NFTs and will now be shown in the My-Tokens section.

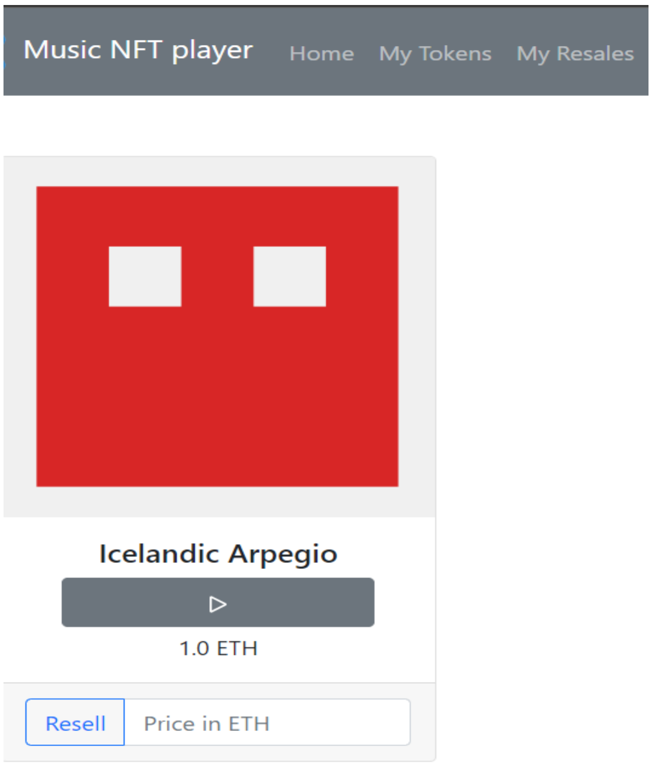


Fig 21 : Bought NFTs

From this section it now shows an option for the user if the user wants to resell it and for what price as well . Once the user sets the reselling price and resells it for that price the music NFT will again be visible in available NFTs on the homepage of other users as well and if the music NFT is bought by some other account holder or some other user then the sold NFT will be shown in the Myreales section with the selling price.

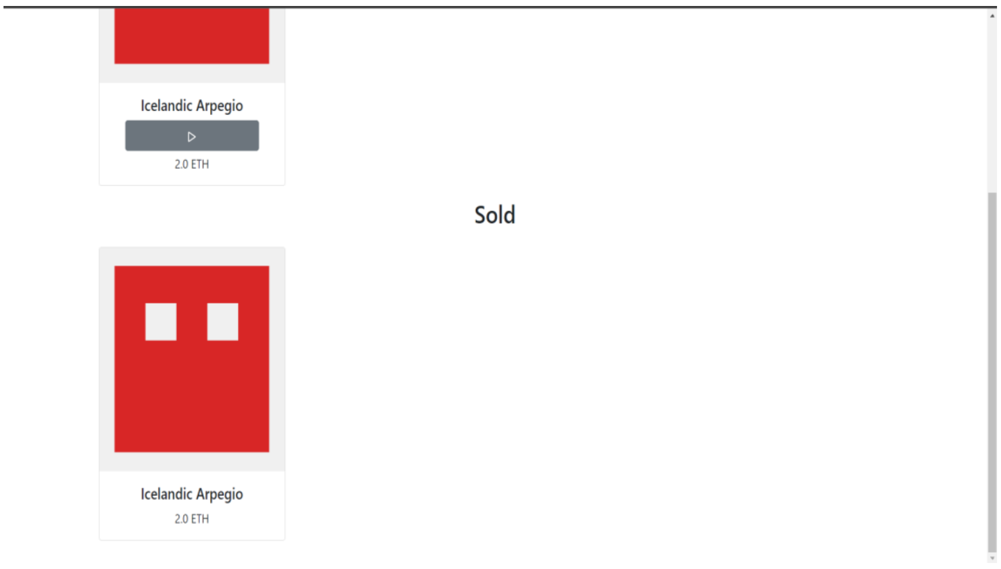


Fig 22 : Completed Transactions

Now, the Myresales tab shows the sold Music NFT and also the price for which it was sold.

The UI of the website is quite simple but it is still responsive and has been created using hooks which makes it compatible for any device which has the ability to connect to a crypto wallet and the website will work perfectly.

Chapter-5
CONCLUSIONS

5.1 Conclusions

The main aim of the training was to be able to understand and implement the concepts of basic Reactjs, Ethereum Blockchain smart contracts development in solidity, expressjs and to be able to create a web application which could perform transaction from the crypto wallets and can be tested using Hardhat tool using the EVM testnet interactions. Thanks to this work, I have achieved these goals. The internship here taught me that developers should not only monitor the code, but also remember that users can improve their experience. The interface should be stable and smooth, making it easy for users to navigate the website. Also, it's important to write clean and readable code so other developers can easily understand it and detect bugs (if any) easily. We should try to write code that can be reused in the future by other developers with minimum hassle.

As an intern and even as a fully fledged developer there is always room for improvement and we should try to learn from the feedback we receive.

5.2 Future Work

1. The frontend of the website was not the main objective so improving the front-end can make the website look more interesting and tempting to use.
2. Showing the gas fee which is transferred to the artist of the music NFT as a royalty.
3. After using the testnet tool (hardhat) we have to check the website on an actual database for making transactions.
4. Providing a user login page rather than directly logging in with the crypto wallet will increase the security by one level .
5. Providing mobile notifications rather than just browser notifications will also increase security since an OTP will be required and hence security will be increased by one more level.

References

[1] "React Documentation ",[online] available : "https://legacy.reactjs.org/docs/getting-started.html"

[2]"JavaScript Documentation" ,[online] available: "https://developer.mozilla.org/en-US/docs/Web/JavaScript"

[3] "Github Documentation ",[online] available : "https://docs.github.com/en"

[4] Satpal Singh Kushwaha, Sandeep Joshi, Amit Kumar Bairwa, Sandeep Chaurasia", publisher: CRC Press, , doi : 27.03.2023.

[5] Gavin Wood "A secure decentralised generalised transaction ledger",publisher: IJRASET, vol. 1, doi : 30.09.2017.

[6] https://bitcoin.org/bitcoin.pdf

[7] "Web3.0 Documentation ",[online] available : "https://web3js.readthedocs.io/en/v1.8.2/"

[8] "Hardhat Documentation ",[online] available : "https://hardhat.org/docs"

[9] "Mocha Documentation ",[online] available : "https://mochajs.org/"

[10] "Node.js Documentation ",[online] available : "https://nodejs.org/en/docs"

[11] Morris, David Z. (15 May 2016). "Leaderless, Blockchain-Based Venture Capital Fund Raises $100 Million, And Counting". *Fortune*. Archived from the original on 21 May 2016. Retrieved 23 May 2016.

[12] Popper, Nathan (21 May 2016). "A Venture Fund With Plenty of Virtual Capital, but No Capitalist". *The New York Times*. Archived from the original on 22 May 2016. Retrieved 23 May 2016

[13] "Blockchains: The great chain of being sure about things". *The Economist*. 31 October 2015. Archived from the original on 3 July 2016. Retrieved 18 June 2016. The technology behind bitcoin lets people who do not know or trust each other build a dependable ledger. This has implications far beyond the crypto currency".

[14] Narayanan, Arvind; Bonneau, Joseph; Felten, Edward; Miller, Andrew; Goldfeder, Steven (2016). *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton, New Jersey: Princeton University Press. ISBN 978-0-691-17169-2.

[15] Iansiti, Marco; Lakhani, Karim R. (January 2017). "The Truth About Blockchain". *Harvard Business Review*. Cambridge, Massachusetts: Harvard University. Archived from the original on 18 January 2017. Retrieved 17 January 2017. The technology at the heart of bitcoin and other virtual currencies, blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way.

[16] Oberhaus, Daniel (27 August 2018). "The World's Oldest Blockchain Has Been Hiding in the New York Times Since 1995". *Vice*. Retrieved 9 October 2021.

[17] Lunn, Bernard (10 February 2018). "Blockchain may finally disrupt payments from Micropayments to credit cards to SWIFT". *dailyfintech.com*. Archived from the original on 27 September 2018. Retrieved 18 November 2018.

[18] Hampton, Nikolai (5 September 2016). "Understanding the blockchain hype: Why much of it is nothing more than snake oil and spin". *Computerworld*. Archived from the original on 6 September 2016. Retrieved 5 September 2016.

[19] Bakos, Yannis; Halaburda, Hanna; Mueller-Bloch, Christoph (February 2021).

"When Permissioned Blockchains Deliver More Decentralization Than Permissionless". *Communications of the ACM*. **64** (2): 20–22. doi:10.1145/3442371. S2CID 231704491.

[20] Armstrong, Stephen (7 November 2016). "Move over Bitcoin, the blockchain is only just getting started". *Wired*. Archived from the original on 8 November 2016. Retrieved 9 November 2016.

[21] Catalini, Christian; Gans, Joshua S. (23 November 2016). "Some Simple Economics of the Blockchain" (PDF). *SSRN*. doi:10.2139/ssrn.2874598. hdl:1721.1/130500. S2CID 46904163. SSRN 2874598. Archived (PDF) from the original on 6 March 2020. Retrieved 16 September 2019.

[22] Tapscott, Don; Tapscott, Alex (8 May 2016). "Here's Why Blockchains Will Change the World". *Fortune*. Archived from the original on 13 November 2016. Retrieved 16 November 2016.