İstanbul
Bilgi Üniversitesi

Title: NFT MARKETPLACE PROJECT
Report 1 for Thesis

by

AHMAD YOUSSEF, 117200027
Sadiq Tijjani, 119200113
Baris Tonbul, 117200054

Supervised by

Dr. Hakan Ayral

Submitted to the
Faculty of Engineering and Natural Sciences
in partial fulfillment of the requirements for the

Bachelor of Science

in the
Department of Computer Engineering

2nd July, 2022

Abstract

This project concerns building a digital marketplace that will serve as a meeting point for buyers and sellers of NFTs to come together and trade, the Marketplace will display a gallery of collectibles with their respective prices and details, similar to everyday art auctions. We will be trying to facilitate an easy exchange of value-able assets on the ethereum network.

The idea is for any user to use their crypto-wallets to log-in to our website and then be able to easily Mint an NFT for sale or bid for an NFT in the marketplace gallery. Since the NFT has recently gotten alot of attention as more and more people realize the value of it, this has led to a boom in the NFT market and has shown glimpses of a future paradigm shift in the art world.

The following report will expand about the topics of blockchain and how it works, the NFT, what it is and how it is connected to blockchain, later in the report we get into the nitty-gritty of how we intend to build our marketplace and then finally in the latter part of the report we illustrate the goal and results expected for the project.

Table of Contents

List of Figures

List of Abbreviations

i.e.	Id est (Latin: this means)
e.g.	Exempli gratia (Latin: for example)

1 INTRODUCTION

The emergence of blockchain has allowed for recording transactions and tracking assets in any given network, the efficiency, security and transparency this blockchain ledger provides has given opportunities for applications in many areas in human life, most especially in the area of digital legal tender(digital money) as seen in the ever-growing crypto-currency boom. So what exactly is blockchain and how does it work?

Blockchain is a peer-to-peer immutable decentralised distributed ledger that allows for recording digital assets transparently without involving any third party intermediary. A decentralized network has a number of advantages over a centralized network including an increase in reliability, transparency and security, Moreover decentralized networks are much less difficult to scale and thus allow for better shared communication and distributed processing.

Using this technology, participants can confirm transactions without a need for a central clearing authority. A transaction in the block requires a wallet, a blockchain wallet is a program(or application) that allows users to hold and spend crypto-currencies like Ethereum(ETH) or Bitcoin(BTC), these wallets are secured using crytographic methods(public and private key). The user's private key allows for the user to spend the associated crypto-currencies, in addition, there is also a public key and there is a cryptographic link between the public key and the private key, theoretically, a user can create billions of public key addresses from his/her private key ,once a user creates a public key address, that address is publicly available to all users in the network as an address where they can send crypto-currencies like Bitcoin. Therefore a transaction works thusly;- If User A want to invoke a transaction with User B , A will specify transaction amount and the public key of B, this message is then signed using A's private key, the signed message is the encrypted using a secure hash algorithm(SHA-256), after these steps a block in the ledger will be created representing that transaction is created, once the block has been created the transaction is then broadcasted over the peer-to-peer network where User B will be listening, User B will then receive and decrypt the message using his/her private key, a concensus algorithm called Proof of work will then use cryptographic proofs to check if the public-private keys pairs are valid, once a transaction is verified, it is combined with other blocks to create a new block of data for the ledger.

The blockchain immutable ledger gives way for a number of applications not only in more traditional industrial sectors like healthcare, insurance and other financial services but also into the ownership of digital media assets and collectibles. This is where the Non-fungible token comes into play, the non-fungible token is a class of token that draws it's value from it's uniqueness, NFTs are usually associated with digital art or images with the license to use these assets for a specific reason. The NFT is a unit of data stored on the blockchain ledger that works like a crytographic token, but unlike other such tokens like Bitcoin, Ethereum and XRP, NFTs are not mutually interchangeable hence the name non-fungible tokens. NFTs are created when the blockchains merge together records(blocks) of cryptographic hash with other previously existing blocks creating a data chain of identifiable blocks. The blockchain transaction process ensures authentication of each digital media file by providing a digital signature that is used to track Non-Fungible Token ownership. The NFT has become very popular recently as artists, content creators and the general public realize the value in electronically owning a piece of art and/or other rare collectible items(for example, Jack Dorsey's first tweet on twitter was sold for 2.9 million dollars as an NFT), one can own a digital media asset as an NFT with the help of an immutable ledger which means that the NFT boom really is a move from the analog traditional way of art collection and ownership to a digital paradigm for art collection and ownership, but just like one can go to the louvre and take a picture of the mona lisa(or re-create it entirely), it is also possible for someone to take the screenshot(digital copy) of an NFT, simply because the ownership of an NFT does not inherently grant intellectual property rights of the asset to the owner. The hope of NFTs is that as we move further and further into the digital age and as the blockchain ledger increasingly finds more and more use in everyday human life(like in voting systems, healthcare systems and social media applications), one can not only own a media asset digitally, but one can also own the rights and intellectual property to that media asset as well.

The main purpose of this project is to build a digital marketplace that allows sellers who are interested in selling their NFTs and buyers who are interested in buying said NFTs to meet and trade.

2 Design

Our website will be a Decentralized application(Dapp), a Dapp is basically an application built on a decentralized network that combines a smart contract and a frontend user interface, Dapps are like other normal web applications except that they run on a peer-to-peer network, such as blockchain. Dapps are very beneficial in that they are resistant to censorship, they are open source and they are blockchain based(built on smart-contracts). Dapps work essentially according to the various logics of the algorithms contained in the smart contracts upon which they are built, smart contracts remove the need for a third party to handle transactions between peers. Since the middle man is replaced by code, all kinds of costs are reduced, including time and money. To design our Website, we will be using the following design tools:-

2.1 React.js

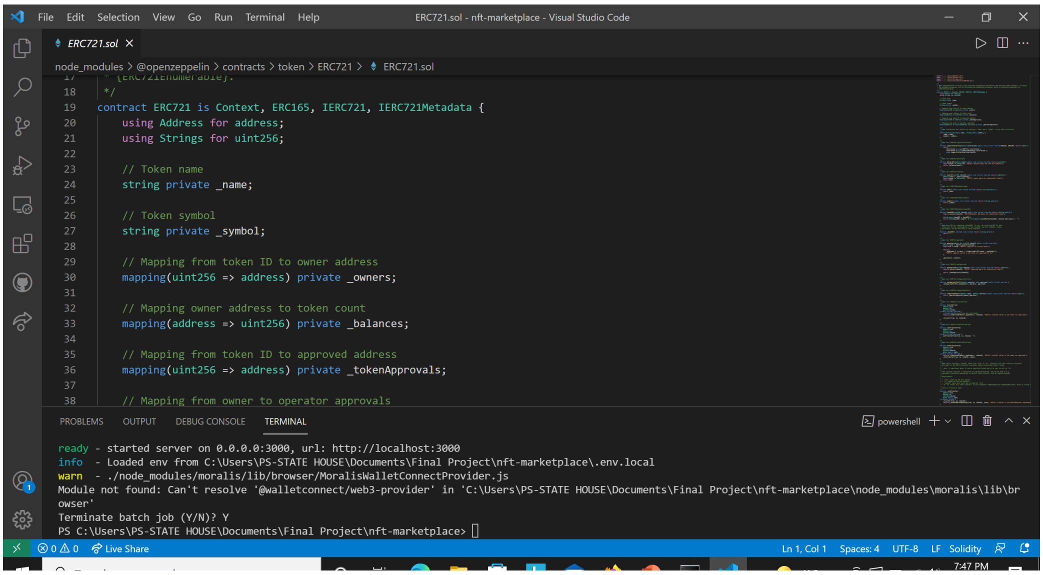
ReactJS is a flexible JavaScript library for building composable and reusable User Interfaces. It can be used on client and server side as well as with other frameworks.

2.2 NPM

Npm will make it simple for us to share and reuse code or even update it since it is a package manager for Node platform, and if a package references to another package with a git URL, npm depends on a pre-installed git.

2.3 Solidity

Solidity is an object-oriented programming language, meaning that it is organized by data or objects rather than functions or logic. Its main purpose is for developing smart contracts for the Ethereum blockchain. A smart contract is a program(collection of functions and states) that runs on the ethereum blockchain, they're not controlled by a user, instead they are deployed to the network and run as programmed. User accounts can then interact with a smart contract by submitting transactions that execute a function defined on the smart contract. Smart contracts for minting token usually implement the ERC 721 standard which defines a minimum interface a smart contract must implement to allow unique tokens to be managed, owned, and traded.



```
 * {ERC721Enumerable}.
 */
contract ERC721 is Context, ERC165, IERC721, IERC721Metadata {
    using Address for address;
    using Strings for uint256;

    // Token name
    string private _name;

    // Token symbol
    string private _symbol;

    // Mapping from token ID to owner address
    mapping(uint256 => address) private _owners;

    // Mapping owner address to token count
    mapping(address => uint256) private _balances;

    // Mapping from token ID to approved address
    mapping(uint256 => address) private _tokenApprovals;

    // Mapping from owner to operator approvals
```

```
ready - started server on 0.0.0.0:3000, url: http://localhost:3000
info  - Loaded env from C:\Users\PS-STATE HOUSE\Documents\Final Project\nft-marketplace\.env.local
warn  - ./node_modules/moralis/lib/browser/MoralisWalletConnectProvider.js
Module not found: Can't resolve '@walletconnect/web3-provider' in 'C:\Users\PS-STATE HOUSE\Documents\Final Project\nft-marketplace\node_modules\moralis\lib\br
owser'
Terminate batch job (Y/N)? Y
PS C:\Users\PS-STATE HOUSE\Documents\Final Project\nft-marketplace>
```

Figure 1: Snapshot Of Implementation of ERC 721 Standard In Solidity

2.4 Truffle

Truffle is a development environment that helps us to compile, deploy, test, and debug Ethereum software. It helps developers manage and automate the recurring tasks that are inherent to the process of building smart contracts and dApps, as well as easily introducing more functionality around this workflow.

2.5 Ant Design

Ant design is an enterprise class react user interface library that contains components that are easy to use for building interactive user interfaces.

2.6 Vercel

Vercel is a platform for frontend frameworks and static websites that is built to seamlessly integrate with it's user's headless content, commerce, or database.It provides a way for developers to easily take care of notoriously difficult tasks like deploying instantly, scaling automatically, and serving personalized content around the globe.

2.7 IPFS

IPFS is a file sharing system that can be leveraged to more efficiently store and share large files. It relies on cryptographic hashes that can easily be stored on a blockchain. Nonetheless, IPFS does not permit users to share files with selected parties. This is necessary, if sensitive or personal data needs to be shared.

IPFS services can be gotten from a number of outlets including Moralis.

2.8 MetaMask

MetaMask allows users to store and manage account keys, broadcast transactions, send and receive Ethereum-based cryptocurrencies and tokens, and securely connect to decentralized applications through a compatible web browser or the mobile app's built-in browser.

2.9 Binance Smart Chain

Binance Smart Chain (BSC) is a blockchain that runs in parallel with Binance Chain. Binance Smart Chain has Ethereum Virtual Machine compatible programmability and native cross-chain communication with Binance Chain by using a consensus of Proof of Staked Authority (PoSA). This dual-chain architecture will empower its users to build their decentralized apps (DApps) which retain the high-performance matching of Binance DEX and crypto-assets on one blockchain and take advantage of the fast trading to exchange on the other. We used BSC temporarily while Polygon was having network issues.

2.10 Polygon

Polygon Matic or Polygon is a framework for developing inter-operable blockchain networks. The framework primarily focuses on addressing some of the prominent setbacks associated with Ethereum. Polygon is able to leverage a new sidechain solution for addressing setbacks such as lack of community governance, throughput complications, transaction speed and poor user experience.The Polygon Ethereum layer 2 scaling solution will help us to facilitate and verify transactions in the marketplace.

2.11 Moralis

Moralis provides managed backend for blockchain projects. It will Automatically sync the balances of your users into the database, allowing for set up of on-chain alerts, observable smart contract events, index building, and more. Using Moralis is one of the fastest way to build and deploy dApps on Ethereum, BSC, Polygon, Solana, and Elrond. Building on Moralis ensures that our dApp is future-proof.

3 Website Overview

In the simplest terms, our marketplace(or any Dapp for that matter) is giving the user a way interact with the smart contracts on the blockchain. Basically, our website serves as a front-end user interface through which any user can interact with the back-end(i.e the smart contracts running on the ethereum network) by opting to sell an NFT, buy an NFT, list an NFT etc. Below is an explanaition of the core rudimentary features of our website:-

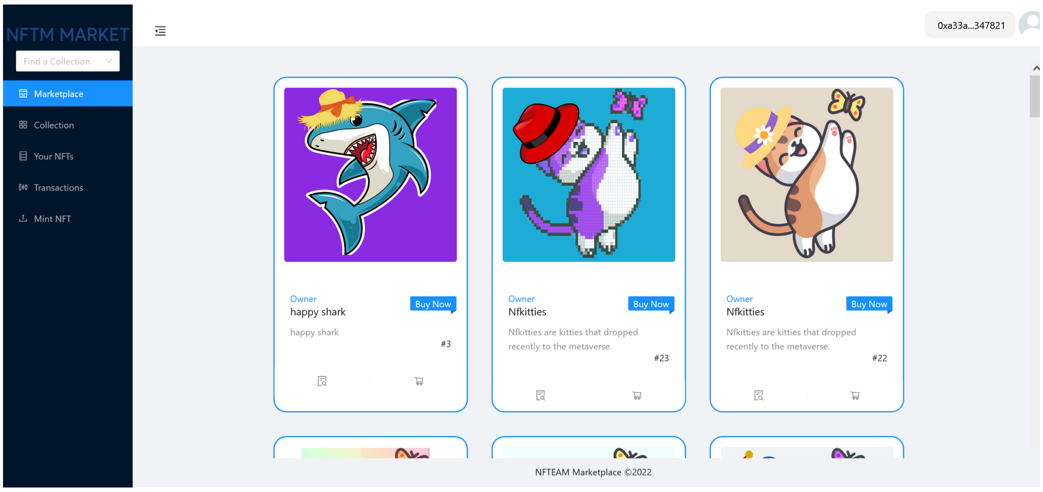
3.1 Market Contract

Before we get to the front end of the marketplace, let's first take a look at the finished contracts. The token contract handles the creation and minting of all tokens on our website, using ERC721URIStorage, counters and ERC721 from the open zeppelin library, we are able to handle the metadata of the tokens, keep track of the number of tokens, mint, set Token URI and set Approval.

The marketplace contract handles all transactions between parties on our website using the ERC721, Counters and Reentrancy Guard from open zeppelin. Take for example when a User wants to list an NFT, this is essentially a transaction between two parties, the message sender(user) and the market address(the zero address), for this transaction to occur our CreateMarketItem function takes in token contract, token ID, and price, it then makes sure that the price is greater than 0, an item mapping is then made for the user's token to our MarketItem struct and the NFT contract is transferred from the message sender(user) to the market(zero address). We also have a Transfer NFT function that allows users to send NFTs to one another, a Delist function that allows users to delist NFTs and a market sale function to handle buy/sell transactions on the marketplace. We have also hot coded a Royalty function into the contract, that would allow the creator of the NFT to receive a certain percentage of the funds from all consequent sales involving his/her token after it's been sold.

3.2 Main Marketplace

This will be a gallery through which any incoming user can view the digital art on sale and select which he/she would like to purchase .



happy shark

happy shark

#3

Nfkitties

Nfkitties are kitties that dropped recently to the metaverse.

#23

Nfkitties

Nfkitties are kitties that dropped recently to the metaverse.

#22

NFTEAM Marketplace ©2022

Figure 2: Marketplace Main Page

3.3 Collection

This page will display the various tokens of the NFKitties collection that are native to the NFTEAM Marketplace, users will also be able to select and purchase an NFKitty to add to their portfolio.

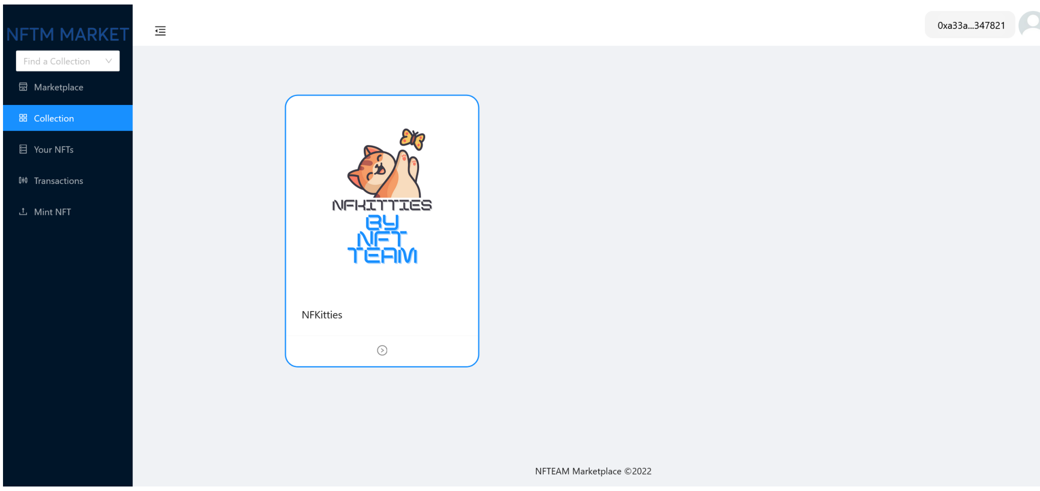


Figure 3: Marketplace Collection Page

3.4 Discovery

For all tokens listed for sale on the marketplace there will be a generated hyperlink on the top left of the token card which when clicked will the send the clicker to the discovery page showing all non-private information of the owner of the token, including Username, profile picture and all tokens owned.

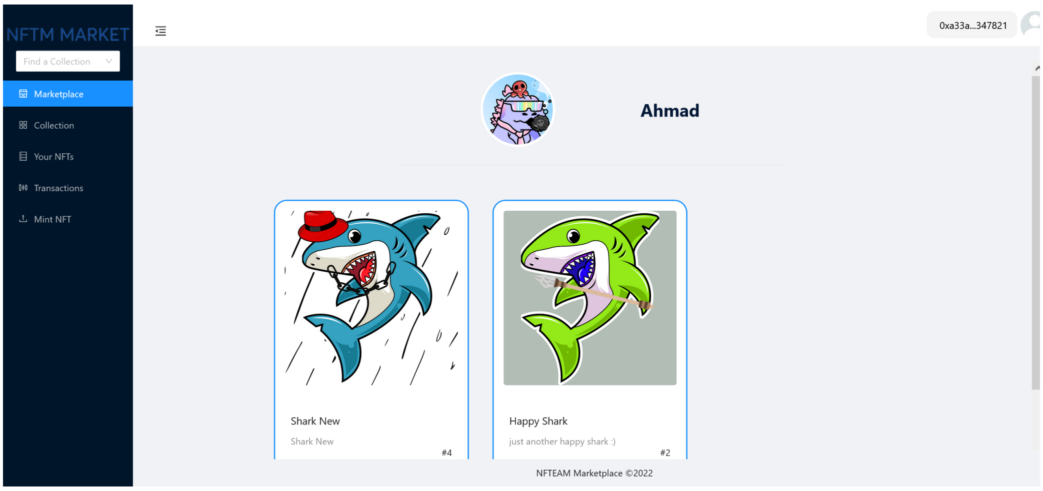
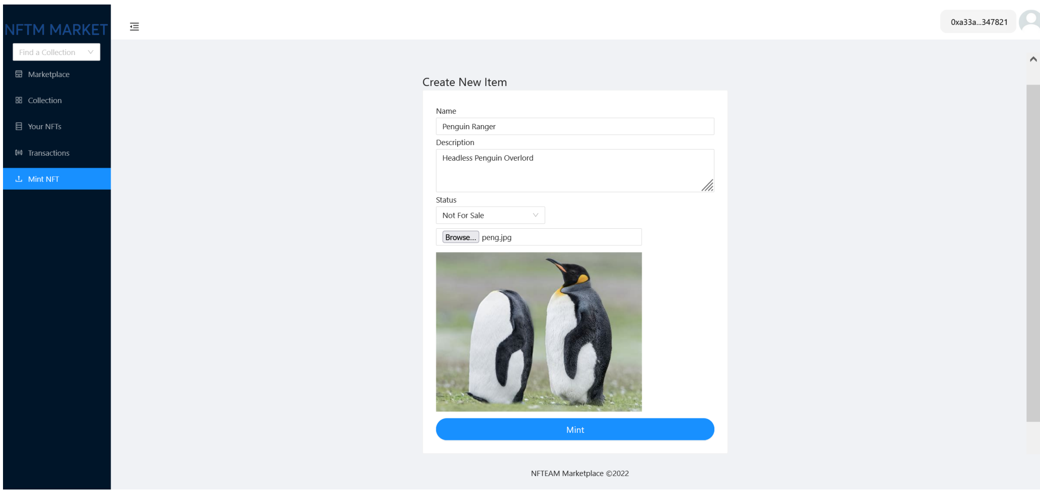


Figure 4: Marketplace Discovery Page

3.5 Mint NFT

This will be a button in the marketplace sidebar that when clicked, will lead the user to a page where he/she is prompted to pick the NFT he/she wants to mint, give it a name, description and price, then mint it for sale.



Create New Item

Name
Penguin Ranger

Description
Headless Penguin Overlord

Status
Not For Sale

peng.jpg

Figure 5: Marketplace Mint Page

3.6 Your NFTs

This will be a button that when clicked, will lead the user(if the user has registered his crypto wallet with the website and has purchased NFTs) to a page that is a gallery of all the NFTs he/she owns. In this page the user will not only also be able to transfer tokens to another user but will also be able to de-list tokens that they have listed for sale.

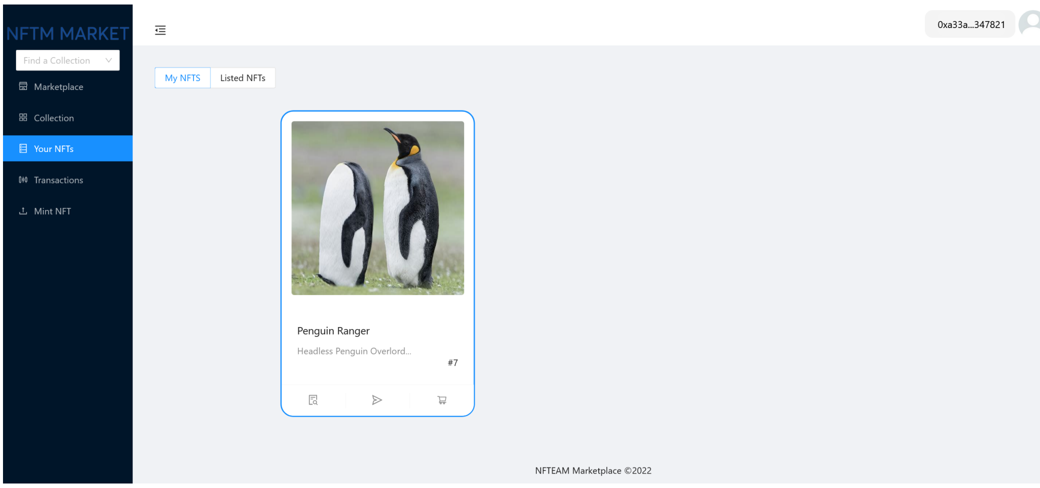
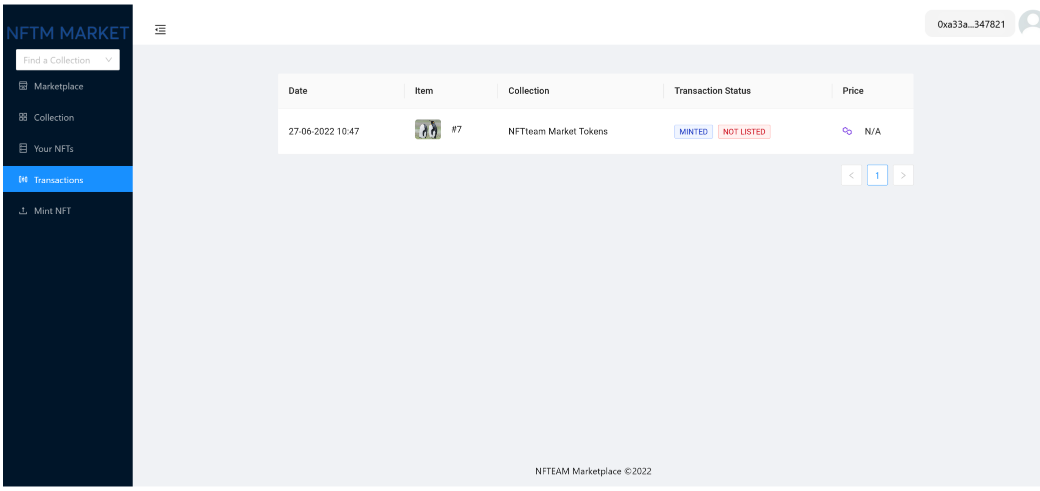


Figure 6: Marketplace YourNFTs Page

3.7 Transactions Overview

This will be a button that when clicked will load up the user's marketplace transaction details i.e of NFTs sold, Bought, Listed etc.



Date	Item	Collection	Transaction Status	Price
27-06-2022 10:47	#7	NFTteam Market Tokens	MINTED NOT LISTED	N/A

Figure 7: Marketplace Transactions Page

4 Problems encountered

- Early in the project, it was difficult to find core rudimentary references or documentation on how to setup, execute and verify(test) a decentralized web application.

- Later in the project, we decided to switch over to Moralis for IPFS and server hosting because it had more functionality and was easier to understand use(loads of documentation).
- Further into the project we encountered a problem with running transaction across the Polygon testnet because the faucet was not supplying any funds, so we switched over to the binance smart chain.
- Further into the project we switched over to the truffle IDE mainly because we found it easier to use.
- We found it a little difficult to map user data to transaction data for the transaction page, mainly because the data came in concatenated and had to be filtered according to transaction date.
- The metamask accounts for one of the team members kept having all fauceted funds removed after a certain period of idle time, we suspect the wallet might have been compromised, so they switched to a new wallet.
- We encountered alot of difficulty in writing functions of our contract at first because of syntax differences from what we were used and different rules of inheritance, so it took some getting used to.
- Sometimes contract deployment would fail unexpectedly and with no specific explanation(error number) given on the terminal because of problems with the testnet networks.
- We encountered a few issues with the moralis service during the course of making the project, occasionally the NFTs would refuse to load or the User would not be allowed to login because of some network issues on the moralis service.

5 Methodology

Agile methodology is the best methodology for our project as it continues as a cycle:

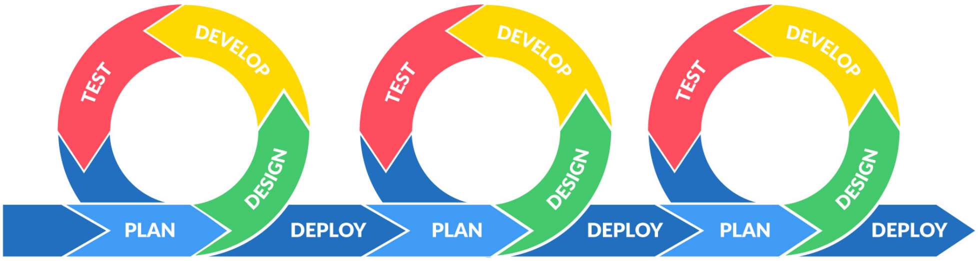


Figure 8: Agile Methodology

When comparing the agile methodology to other methods, agile method has (39 per-cent success rate, 52 per-cent challenge rate and only 9 per-cent failure rate).The agile methodology allows for continuous production cycles, small collaborative teams(as is ours) and most importantly for us, it is adaptive to change.The processes of plan-design-develop-test provide a straightforward, yet also flexible path way for the completion of this project.

As is encouraged in the agile methodology, at each iteration of working on our project, we started with defining the initial requirements(timelines,errors to be corrected, features to be added,removed or updated), then we planned the delivery method and design strategy according to the set expectations. After that, we started working on the set goals accordingly.

6 Previous Works About Project

The NFT market is still new and and is still in it's infancy but there are a few notable projects that have made a strong impression on the NFT community:-

6.1 OpenSea

OpenSea,founded in 2017, is an online decentralized marketplace for NFTs, it is based on the Ethereum ERC-721 standard and Polygon . In August of 2021, OpenSea recorded well over 3.5 billion dollars in NFT trading volume, rising from about 25 million in 2020.

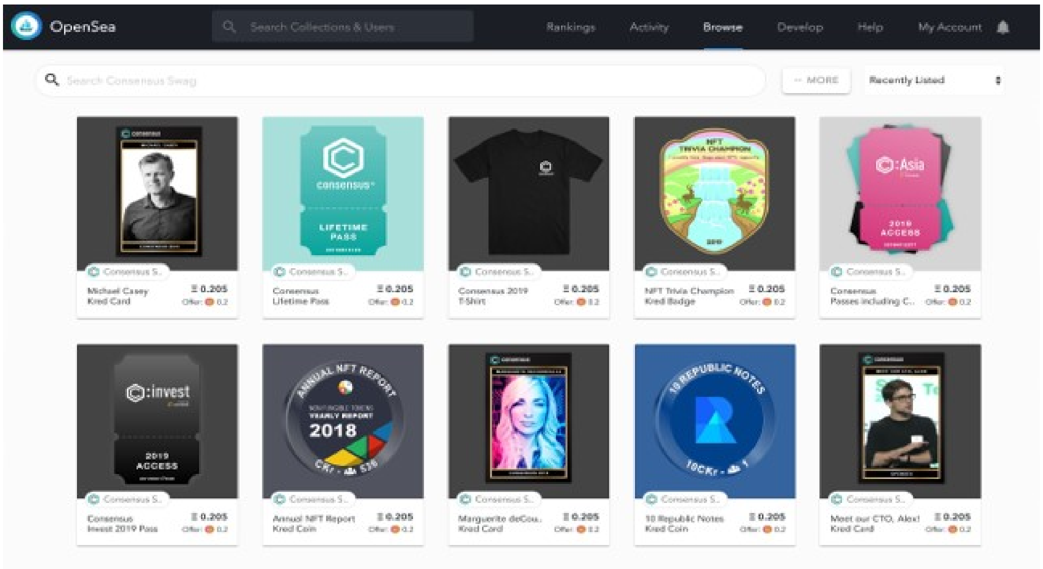
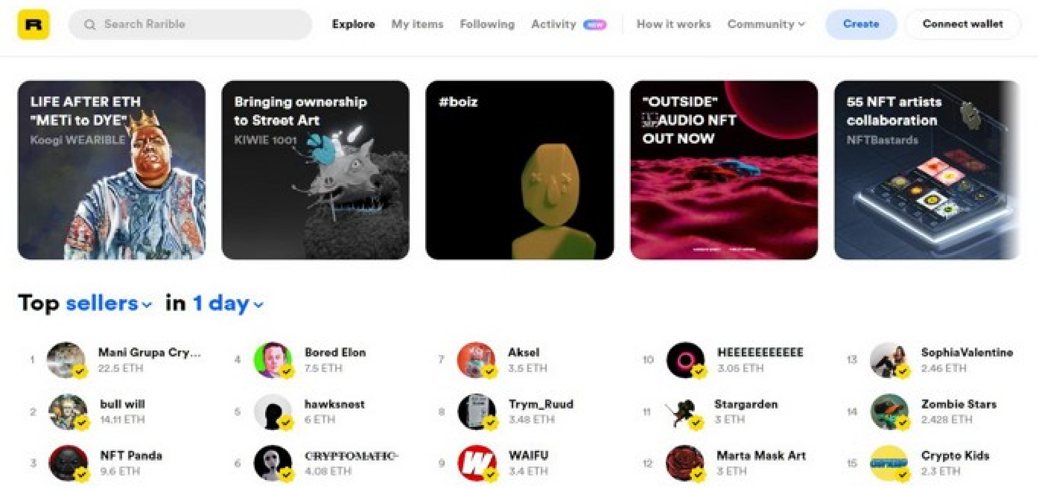


Figure 9: OpenSea Marketplace

6.2 Rarible

Rarible, launched in early 2020, is a platform for creating, buying and trading digital collectibles(otherwise known as NFTs), one of Rarible's key features is the ability to give previews of content in the forms of previews, trailers, or snippets while locking away the full content for a buyer.



Top sellers in 1 day

1	Mani Grupa Cry...	22.5 ETH
2	bull will	14.11 ETH
3	NFT Panda	9.6 ETH
4	Bored Elon	7.5 ETH
5	hawksnest	6 ETH
6	~~CRYPTOMATIC~~	4.08 ETH
7	Aksel	3.5 ETH
8	Trym_Ruud	3.48 ETH
9	WAIFU	3.4 ETH
10	HEEEEEEEEEEE	3.05 ETH
11	Stargarden	3 ETH
12	Marta Mask Art	3 ETH
13	SophiaValentine	2.46 ETH
14	Zombie Stars	2.428 ETH
15	Crypto Kids	2.3 ETH

Figure 10: Rarible Marketplace

7 Timeline

We have undertaken our project in roughly the fashion that is depicted below:-

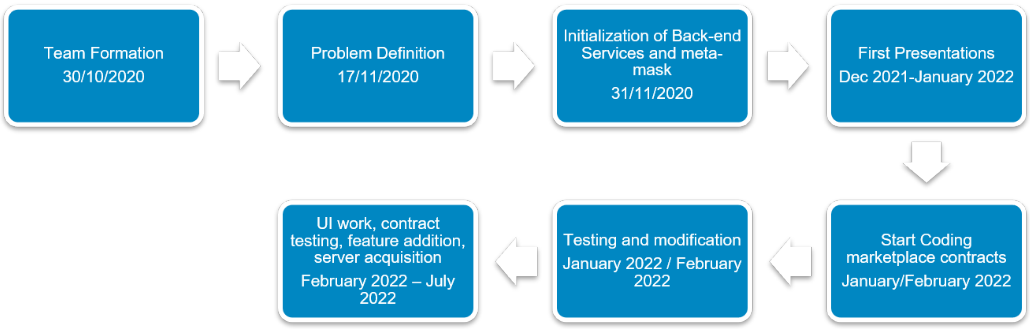

Team Formation
30/10/2020
Problem Definition
17/11/2020
Initialization of Back-end Services and meta-mask
31/11/2020
First Presentations
Dec 2021-January 2022
Start Coding marketplace contracts
January/February 2022
Testing and modification
January 2022 / February 2022
UI work, contract testing, feature addition, server acquisition
February 2022 – July 2022

Figure 11: Project Timeline

8 Project setup

The setup of our project is as follows:-

- Deploying smart contracts involves getting a smart contract address and an application binary interface(ABI) from the source solidity files being deployed into an ethereum network.

- Truffle compilation involves compiling the solidity source code and getting the ABI of the smart contract.

- Then on to the Truffle migration which involves deploying the compiled the byte-code of the smart contract to the ethereum network via the moralis RPC(Remote Procedural Call) provider. An RPC server is an interface provided by an application or service that allows remote clients to connect, pass commands and transfer data. Migrations are JavaScript files that aid in deploying contracts to the Ethereum network. Smart contracts can also be deployed without a migration, the use of migration varies with the development environment and particular features the developer wants for the project.

- The Moralis RPC node will then provide a connection between our Dapp and the polygon blockchain.After the RPC is executed, a request-reply process is initiated between different address spaces. One address will call the procedure while the other address collects and responds, these two address parties are the client and server respectively, meaning that the client sends the request to execute a process with the specified parameters on a separate remote server, consequently, a response is then sent back from the server to the client, allowing the user to interact with the marketplace contract. Polygon will then facilitate and validate all marketplace transactions over it's testnet, this however will be done after the aforementioned tests validate the contracts contain no errors.

- The Moralis Web3 is a collection of libraries which allow you to interact with a local ethereum node, using a HTTP or IPC connection. It aids in the interaction between contracts and the user using the UI. The truffle contract is a better ethereum contract construct for the node and the browser, it helps to directly call a smart contract method in a similar way to calling any java or JS object method instead of sending raw transactions over network, a developer doesn't have to use this method, the processes work the same with or without it, just a better way to call contract methods.

- Finally the User UI with metamask extension serve as a way for the user to invoke transaction on the blockchain.

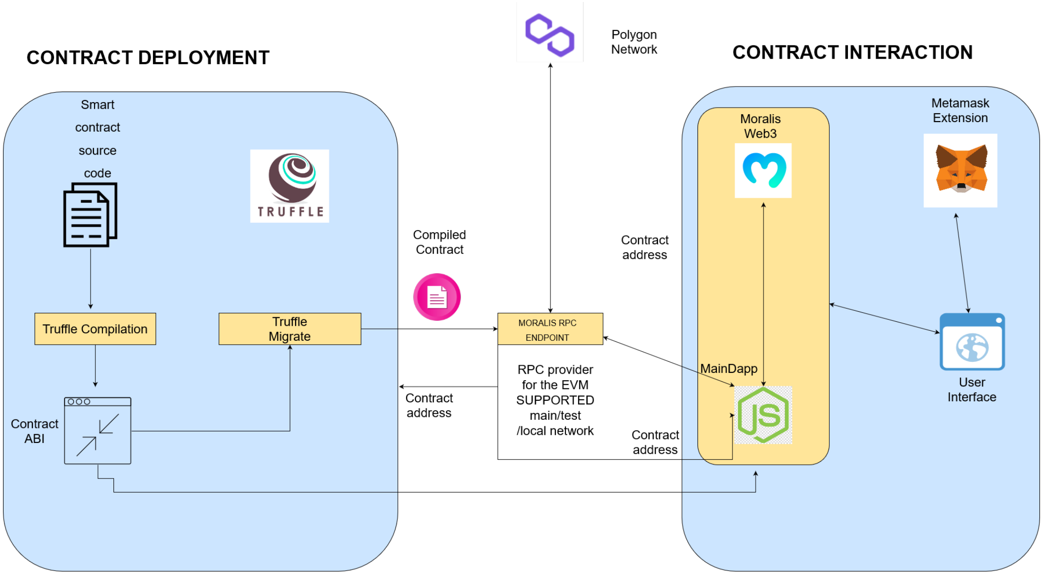
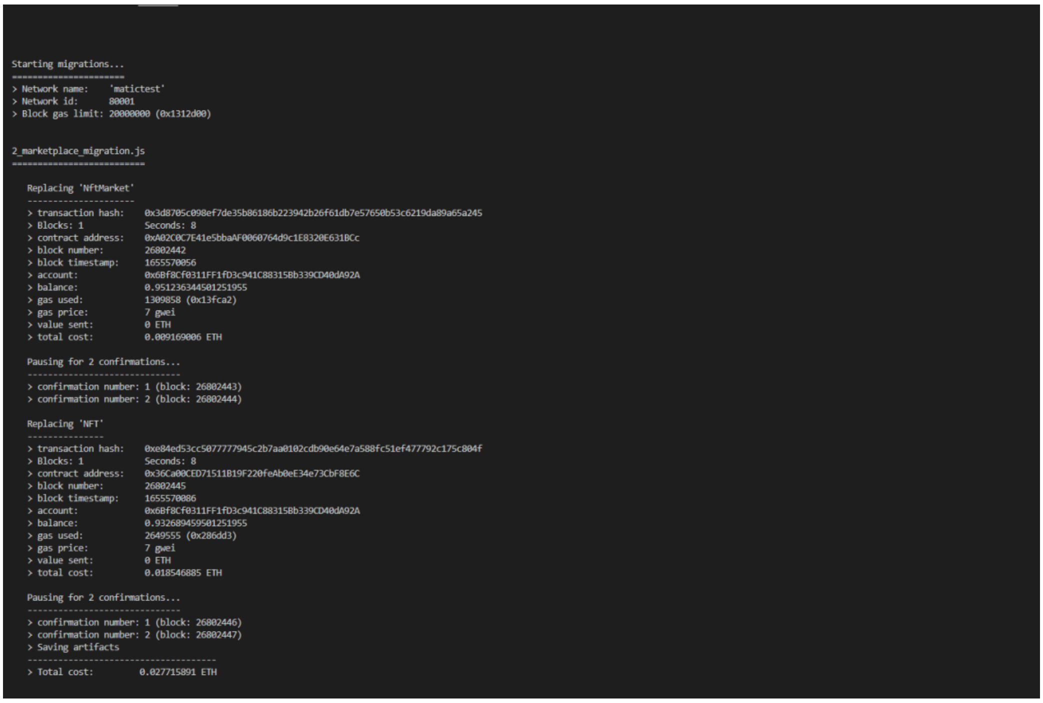


Figure 12: Experimental Setup

9 Experiments and Discussion

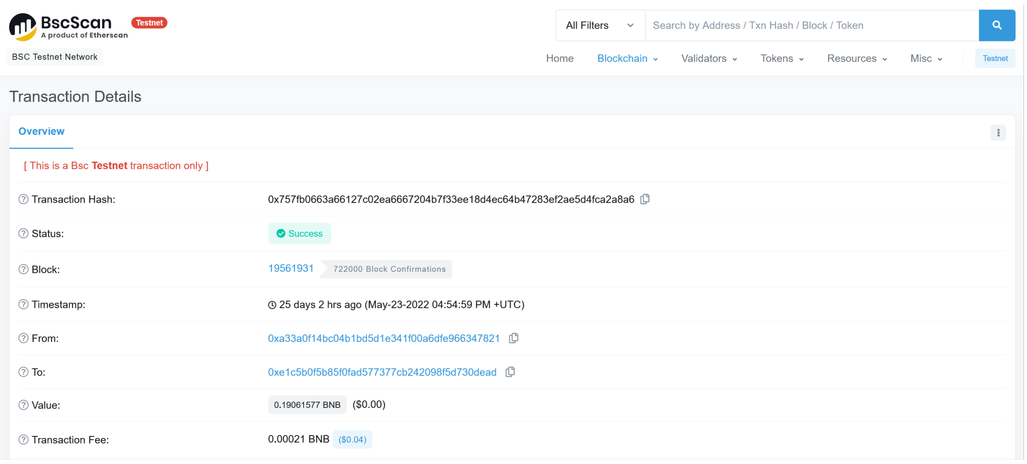
Our marketplace is built and based on smart contracts which facilitate transactions on the market between parties, therefore the project involved a lot of deploying of contracts to the Polygon testnet using truffle. After the contract's functions are finished, we compile using the truffle compile command in the terminal, after which we then use truffle deploy –network maticmumbai, these process also will take a little gas to in order to run:-



```
Starting migrations...
======================
> Network name:    'matictest'
> Network id:      80001
> Block gas limit: 20000000 (0x1312d00)


2_marketplace_migration.js
==========================

   Replacing 'NftMarket'
   ---------------------
   > transaction hash:    0x3d8705c098ef7de35b86186b223942b26f61db7e57650b53c6219da89a65a245
   > Blocks: 1            Seconds: 8
   > contract address:    0xA02C0C7E41e5bbaAF0060764d9c1E8320E631BCc
   > block number:        26802442
   > block timestamp:     1655570056
   > account:             0x6Bf8Cf0311FF1fD3c941C88315Bb339CD40dA92A
   > balance:             0.951236344501251955
   > gas used:            1309858 (0x13fca2)
   > gas price:           7 gwei
   > value sent:          0 ETH
   > total cost:          0.009169006 ETH

   Pausing for 2 confirmations...
   ------------------------------
   > confirmation number: 1 (block: 26802443)
   > confirmation number: 2 (block: 26802444)

   Replacing 'NFT'
   ---------------
   > transaction hash:    0xe84ed53cc5077777945c2b7aa0102cdb90e64e7a588fc51ef477792c175c804f
   > Blocks: 1            Seconds: 8
   > contract address:    0x36Ca00CED71511B19F220feAb0eE34e73CbF8E6C
   > block number:        26802445
   > block timestamp:     1655570086
   > account:             0x6Bf8Cf0311FF1fD3c941C88315Bb339CD40dA92A
   > balance:             0.932689459501251955
   > gas used:            2649555 (0x286dd3)
   > gas price:           7 gwei
   > value sent:          0 ETH
   > total cost:          0.018546885 ETH

   Pausing for 2 confirmations...
   ------------------------------
   > confirmation number: 1 (block: 26802446)
   > confirmation number: 2 (block: 26802447)
   > Saving artifacts
   -------------------------------------
   > Total cost:         0.027715891 ETH
```

Figure 13: Truffle Terminal after deploy command to Polygon mumbai.

Further along in the project, in order to be able to see use and see if the contract functions have been successfully interacted with, we use something called a testnet explorer, this is an interface which helps users of any blockchain network to keep track of successful(and unsuccessful) transactions with their smart contract on the network. Take for example the transfer NFT capability of our marketplace, this requires the owner of a token to invoke a transfer by calling upon the TrasnferNFT function in our smart contract, this transaction over the network can either succeed or fail for multiple reasons(Lack of approval, network failures, incorrectly written contract functions etc), the testnet explorer provides a good interface to keep track of these things, here's a couple screenshots from when we were temporarily switched over to the BSC network:



Transaction Details

[This is a Bsc **Testnet** transaction only]

Transaction Hash: 0x757fb0663a66127c02ea6667204b7f33ee18d4ec64b47283ef2ae5d4fca2a8a6

Status: Success

Block: 19561931 722000 Block Confirmations

Timestamp: 25 days 2 hrs ago (May-23-2022 04:54:59 PM +UTC)

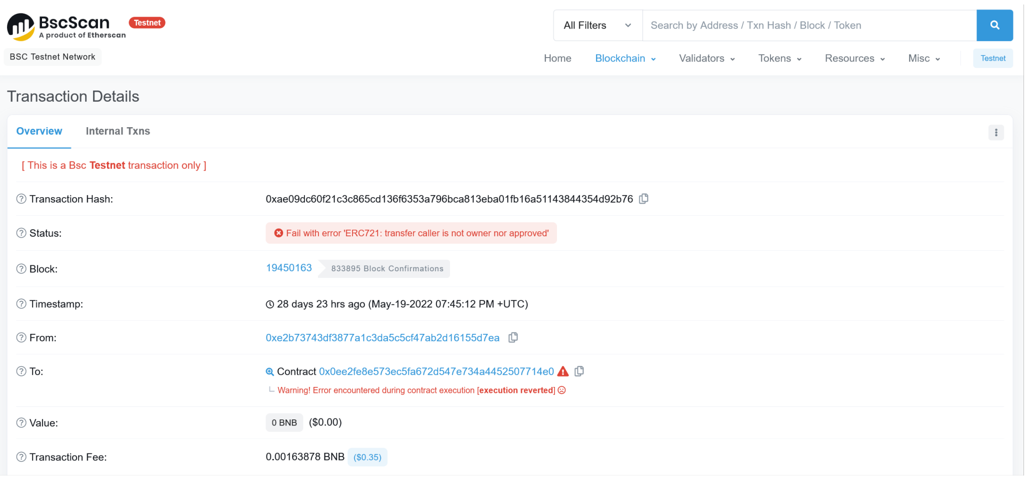
From: 0xa33a0f14bc04b1bd5d1e341f00a6dfe966347821

To: 0xe1c5b0f5b85f0fad577377cb242098f5d730dead

Value: 0.19061577 BNB ($0.00)

Transaction Fee: 0.00021 BNB ($0.04)

(a) A successfully executed market transaction



Transaction Details

[This is a Bsc **Testnet** transaction only]

Transaction Hash: 0xae09dc60f21c3c865cd136f6353a796bca813eba01fb16a51143844354d92b76

Status: Fail with error 'ERC721: transfer caller is not owner nor approved'

Block: 19450163 833895 Block Confirmations

Timestamp: 28 days 23 hrs ago (May-19-2022 07:45:12 PM +UTC)

From: 0xe2b73743df3877a1c3da5c5cf47ab2d16155d7ea

To: Contract 0x0ee2fe8e573ec5fa672d547e734a4452507714e0

Warning! Error encountered during contract execution **[execution reverted]**

Value: 0 BNB ($0.00)

Transaction Fee: 0.00163878 BNB ($0.35)

(b) A failed market transaction due to lack of approval

10 Conclusions and Marketplace Optimization

The explosion in art based NFTs is a sign of a subtle paradigm shift in ways art is consumed. It directly affects both art and the artist. Working on and affecting key problems of the traditional setup such as accessibility to art, compensation of the artist, Ownership and licensing of media, the role of the middleman(art dealer) etc. It is with this in mind that we believe it was worthwhile to pursue this project.
A few improvements could be made on the marketplace:

1. It is possible to implement a bidding system where a creator can open up a bid for his/her NFT and have potential buyers bid for the NFT for period of time, after which the highers bidder will get the NFT.

2. It may also be improved by making it cross-chain, meaning the marketplace can run on multiple blockchain networks, it can handle transactions on Polygon should the user choose so and it can also handle transactions on the Binance smart chain should the user choose to.

3. The user interface could be greatly enhanced by adding features such as dark mode, popularity rankings , rarity meters and so on.

References

[1] https://www.npmjs.com/
[2] https://docs.npmjs.com/cli/v6/commands/npm
[3] https://www.ibm.com/topics/what-is-blockchain
[4] https://www.nbcnews.com/tech/tech-news/nft-boom-digital-collectibles-rcna430
[5] https://www.securities.io/non-fungible-tokens-nfts-continue-to-reshape-the-blockchain-market/
[6] https://pagepro.co/blog/what-is-nextjs/
[7] http://satoshinakamoto.me/bitcoin.pdf
[8] https://trufflesuite.com/docs/truffle/
[9] https://docs.moralis.io/moralis-dapp/files/ipfs
[10] https://docs.moralis.io/speedy-nodes/what-are-speedy-nodes