

**NFTVerse - A NFT MarketPlace**  
**MINOR PROJECT - I**

Submitted by-  
**Geetali Agarwal (20103254)**  
**Kritarth Bansal (20103256)**  
**Saksham Gupta (20103268)**

Under the Supervision of-  
**Dr. Nitish Andola**



**Department of CSE/IT**  
**Jaypee Institute of Information Technology, Noida**  
**December- 2022**

## **ACKNOWLEDGEMENT**

We would like to place on record our deep sense of gratitude to Dr. NITISH ANDOLA, Assistant Professor (senior grade), Jaypee Institute of Information Technology, India for his generous guidance, help and useful suggestions. We express our sincere gratitude to him for his stimulating guidance, continuous encouragement, and supervision throughout the course of present work.

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

Student ID	Student Name	Student Signature
20103254	Geetali Agarwal	
20103256	Kritarth Bansal	
20103268	Saksham Gupta	

## **DECLARATION**

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

Place: Jaypee Institute of Information Technology, Noida

Date: 5<sup>th</sup> Dec 2022

Name- Geetali Agarwal

Enroll no.- 20103254

Batch- B9

Name- Kritarth Bansal

Enroll no.- 20103256

Batch- B9

Name- Saksham Gupta

Enroll no.- 20103268

Batch- B9

## **CERTIFICATE**

This is to certify that the work titled “NFT Verse” submitted by Geetali Agarwal, Kritarth Bansal, and Saksham Gupta of B.Tech of Jaypee Institute of Information Technology, Sec-62 Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Name: Dr. Nitish Andola

Designation: Assistant Professor (Sr. Grade)

Date: 5<sup>th</sup> December, 2022

Signature:

## **Abstract**

Before the widespread application of blockchain based technologies, the mechanisms in place for verifying ownership of digital assets and thus, means of securing them remained susceptible to tampering that translated into significant losses. Decades of research and advancements in blockchain led to the development of Non-Fungible Tokens (NFTs), which are tokens that represent digital assets and have proof of ownership embedded. The novel characteristic of each token being unique and distinctive from another has strengthened the security of assets and reinforced unique ownership. This cutting-edge technology continues to grow and capture the attention of the masses as more applications of NFTs are identified with time. This project aims to present a comprehensive NFT marketplace and its underlying core technologies, namely blockchain and Ethereum.

## **Table of Contents**

<b>Chapter No.</b>	<b>Topics</b>	<b>Page No.</b>
<b>Chapter-1</b>	<b>Introduction</b>	
	1.1 General Introduction	7
	1.2 Problem Statement	7
	1.3 Significance of NFT	8
	1.4 Brief Description of the Solution Approach	9
<b>Chapter-2</b>	<b>Literature Survey</b>	
	2.1 Summary of papers studied	11
<b>Chapter-3</b>	<b>Requirement Analysis and Solution Approach</b>	
	3.1 Requirement Analysis	13
<b>Chapter-4</b>	<b>Modelling and Implementation Details</b>	
	4.1 Detailed Design	14
	4.2 Implementation of Smart Contracts	15
	4.3 Integration of Pinata with Smart Contracts	20
	4.4 Integrating Front End with Smart Contracts:	20
<b>Chapter-5</b>	<b>Testing</b>	
	5.1 Testing of all Functionalities	21
<b>Chapter-6</b>	<b>Conclusion and Future Work</b>	
	6.1 Conclusion	27
	6.2 Future Work	27
	6.3 References	28

# **CHAPTER – 1: INTRODUCTION**

## **1.1 GENERAL INTRODUCTION**

Non-fungible token or NFT is a collectible digital asset. It holds value in the form of cryptocurrency as well as in the form of representatives for art or culture. Just like art is a promising investment with considerable value, NFTs also enjoy the benefit of value.

It is a digital token and can be considered a variant of cryptocurrency. On the other hand, it is different in the fact that it is unique and does not allow like-for-like exchanges. For example, you cannot exchange NFTs like paper-based currency. If you trade your NFT for another NFT, then you have a completely different digital asset in your possession.

Ethereum is the popular blockchain platform for holding NFTs alongside the role of other blockchains in supporting transactions on Ethereum. The process of “minting” or creating an NFT generally involves digital objects representing tangible as well as intangible items. The items could include art, GIFs, designer sneakers, collectibles, music, videos, and sports highlights, and virtual avatars alongside video game skins. Rare items such as tweets have also found recognition in the world of NFTs.

Therefore, it is easier to consider NFTs as physical collector’s items only in the digital space. Rather than obtaining an actual painting, the buyer can get a digital file with exclusive ownership rights. Hence we have created a NFT marketplace to cater to demands of both buyer and seller.

## **1.2 PROBLEM STATEMENT**

The information below will help you scrutinize this subject area from different angles: definition, typical workflow, and business model. Armed with these valuable insights, you will be able to determine how to create an NFT marketplace that will stand out from the competitors. In the simplest terms, an NFT marketplace can be seen as an e-commerce

platform where users can buy and sell digital assets such as digital art, collectibles, gaming items, audio records, etc. These assets are linked to blockchain technology, namely Ethereum, and are called non-fungible tokens. Each NFT is unique, meaning we cannot exchange it for the same digital asset simply because there is none. For that very reason, they are referred to as non-fungible tokens.

### **1.3 WHY IS NFT MARKETPLACE IMPORTANT?**

Big brands, tech enthusiasts, investors, and collectors are finding ways to adopt NFTs to match their pace with speedily advancing technologies, trends and opportunities in the market and expand their offerings. NFTs provide exclusive ownership rights, trading options and an online marketplace to promote extreme liquidity. On top of that, it helps establish a stronger relationship between brands and users.

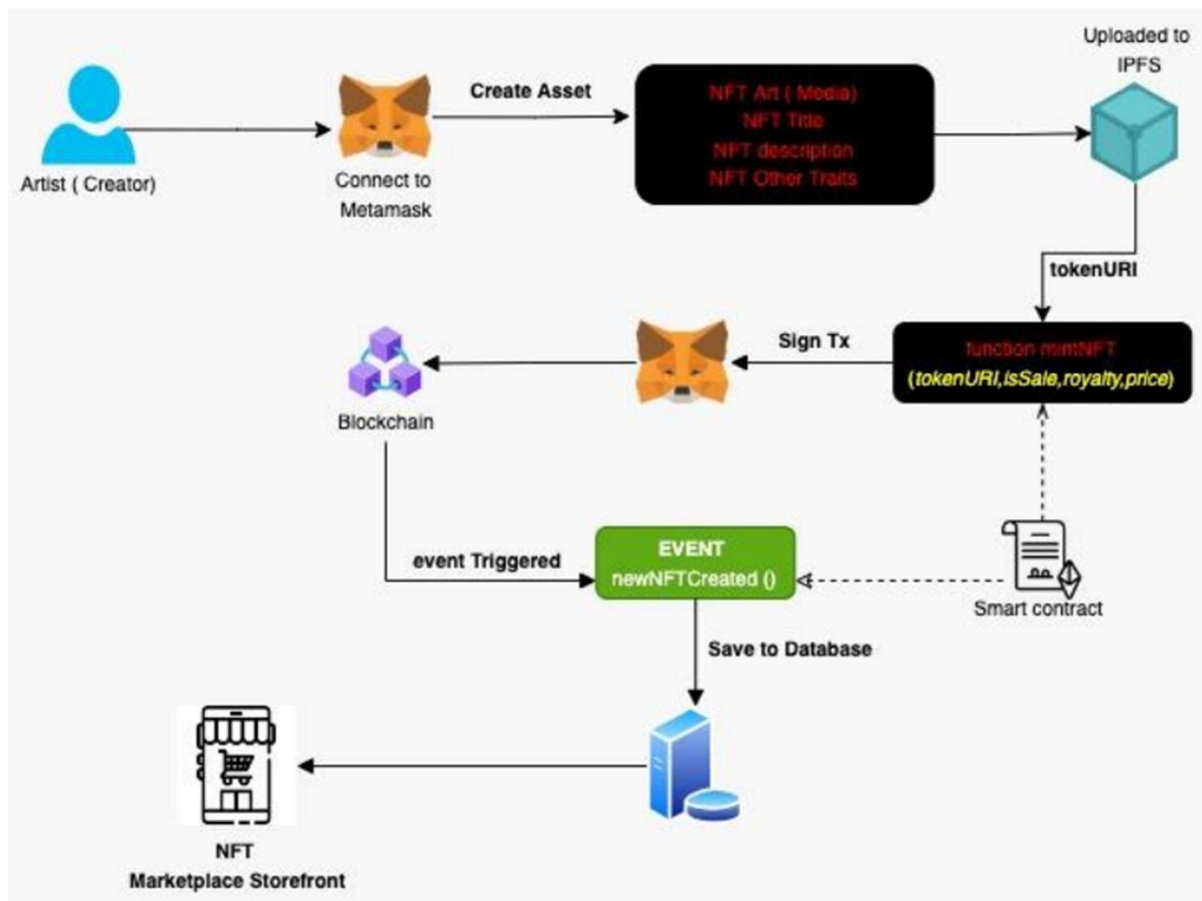
Thus, having a fully functional NFT marketplace that displays those digitally created NFT assets has become crucial. Here are some reasons why NFT marketplace development is considered important.

- NFT marketplaces are backed by blockchain technology. Thus, all the transactions related to NFTs are made visible through a publicly distributed ledger. Further, a huge volume of NFT transactions can also take place instantly with no risk involved.
- All NFT marketplaces are decentralized in nature. This means no third party is involved in any activities occurring on the platform, such as transactions, buying, selling, and trading of NFTs. So, there is no central authority overseeing such platforms.
- Anybody can easily access NFT assets as they are instantly available in NFT marketplaces. Each NFT displayed on an NFT marketplace platform is unique in nature, with special features and digital identities.
- Developing an NFT marketplace also increases user loyalty and encourages users from around the world to trust and participate. As NFTs are publicly accessible ownership records, they guarantee authenticity and exclusive ownership.



## 1.4 BRIEF DESCRIPTION OF THE SOLUTION APPROACH

Blockchain based NFT website includes setting up a blockchain network, integrating chosen digital wallets, creating smart contracts, building a customer-facing web app, establishing security.



While developing an NFT marketplace, it is important to maintain consistency and build an interface that is easy to use for both new and existing users. The architectural components of an NFT marketplace must include a blockchain on which the platform will work, wallets, smart contracts and IPFS.

### ***Blockchain integration***

Various types of blockchain networks can be utilized to build a full-functioning NFT marketplace. These include public, private, consortium and hybrid blockchain networks. The technology helps store information related to NFTs and transactions occurring on the platform.

### ***Token minting***

No matter if you allow NFTs with or without gas costs, minting NFTs is important. The process involves validating all the data gathered, creating a new block, and recording it into the blockchain.

### ***Digital wallet***

A digital wallet is an electronic wallet that helps users store digital funds. It further allows for buying and selling of NFT assets.

### ***NFT metadata***

It is the description of the NFT assets stored on the marketplace. The description may include its name, creation date and time, attributes, true owner, etc.

### ***IPFS (Interplanetary File System)***

IPFS is a data storage system used to avoid the power-consuming storage of digital assets with metadata on the blockchain.

### ***Smart contracts***

An NFT marketplace deploys smart contracts to create a unique identifier for each NFT. It employs the ERC-721 standard to create non-fungible tokens, differentiating them from fungible crypto tokens.

## CHAPTER – 2: LITERATURE SURVEY

### **[1] Year of Publication: 2021**

**Title:** Blockchain based Distributed Marketplace

**Authors:** Oliver R Kabi and Virginia N. L. Franqueira

Developments in Blockchain technology have enabled the creation of smart contracts; i.e., self-executing code that is stored and executed on the Blockchain. This has led to the creation of distributed, decentralized applications, along with frameworks for developing and deploying them easily. This paper described a proof-of-concept system that implements a distributed online marketplace using the Ethereum framework, where buyers and sellers can engage in e-commerce transactions without the need of a large central entity coordinating the process. The performance of the system was measured in terms of cost of use through the concept of 'gas usage'. It was determined that such costs are significantly less than that of Amazon and eBay for high volume users. The findings generally support the ability to use Ethereum to create a distributed on-chain market, however, there are still areas that require further research and development.

### **[2] Year of Publication: 2021**

**Title:** Non-Fungible Tokens (NFT's): The Future of Digital Collectibles

**Authors:** Yashika Nagpal

Non-Fungible Tokens (NFT's) depict a digital certificate of authentication being created on the blockchain technology which is similar to other virtual crypto assets and currencies. The popularity of the blockchain technology along with dealing in crypto assets has seen to grow manifold in the recent years. Having said that, the NFT market is also mushrooming as witnessed in recent years. The very concept of NFT originates from a token standard of Ethereum, aiming to differentiate and distinguish each token with its unique signature being bound with digital properties. The impressive return on its rapidly increasing market

worldwide has drawn massive attention, with India too having witnessed a heightened interest in this digital sector, especially from the upcoming new-age investors and digital creators. However, development of the NFT ecosystem being in an early stage has seen an absence of a regulatory legal framework to govern such pre-mature digital crypto assets in the country of India. The upcoming artists may tend to get lost in this frenetic evolution with lack of systematic summaries. This paper intends to explore the concept of NFT in contrast to cryptocurrency and copyright along with its working and technical components.

**[3] Year of Publication: 2021**

**Title:** NFTs: Applications and Challenges

**Authors:** Wajiha Rehman, Hijab e Zainab, Jaweria Imran, Narmeen Zakaria Bawany

Prior to the widespread application of blockchain-based technology, the mechanisms for verifying ownership of digital assets, the means to protect them, remained vulnerable to tampering, leading to significant losses. Decades of blockchain research and advancement have led to the development of Non-Fungible Tokens (NFTs). Non-Fungible Tokens (NFTs) are tokens that represent digital assets with embedded proof of ownership. The new property that each token is unique and distinguishable from other tokens enhances asset security and enhances unique ownership. This state-of-the-art technology continues to grow and gain public attention as more applications for NFTs are identified over time. The purpose of this study is to provide a comprehensive overview of NFTs and their underlying core technologies: blockchain and Ethereum. In addition, numerous platforms for buying and selling NFTs and applications of NFTs in various fields such as education, fashion, sports, and digital art are presented. Additionally, the whitepaper highlights key challenges in adapting NFT technology in terms of security, privacy, environmental impact, ownership, governance, and ownership.

## **CHAPTER – 3: REQUIREMENT ANALYSIS**

### **3.1 Hardware and Software dependency and prerequisites**

#### **3.1.1. Hardware Used**

- Processor: Intel i3 10gen 64 bit or Ryzen 3 or higher
- RAM: 4 GB
- Hard disk Space: 10 GB
- An Internet Connection

#### **3.1.2. Software Used**

- Operating System: Windows, Linux or Mac
- Code Editor: Vs code, Remix IDE

#### **3.1.3. Tools and Dependencies Used:**

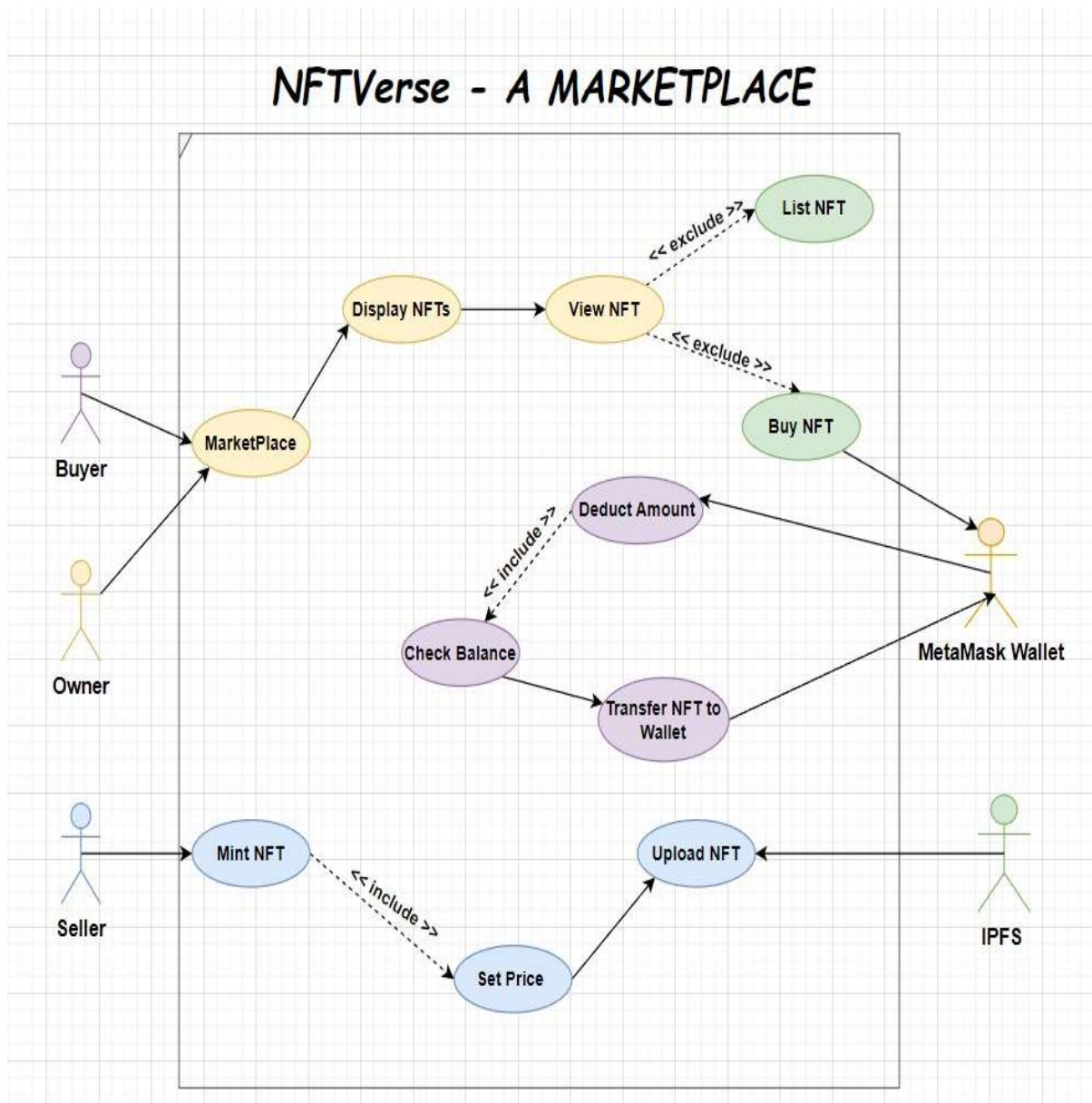
- . Web3.js
- Open Zeppelin
- . Hardhat
- . Tailwind CSS
- . React
- . File server module
- . Pinata
- . Metamask Wallet
- . Node package manager
- . IPFS
- . ERC- 721
- . Alchemy (Node service provider)

#### **3.1.4. Language Used**

- . Solidity
- . JavaScript
- . HTML
- . CSS

## Chapter 4: Modelling and Implementation Details

### Use Case Diagram of the NFT Marketplace:



## Smart Contracts:

Smart contracts are written in Solidity language in `NftMarketplace.sol` file which are based on the **OpenZeppelin** library's ERC-721 implementation.

At the top of our smart contract, we import three **OpenZeppelin** smart contract classes:

- `@openzeppelin/contracts/token/ERC721/ERC721.sol` contains the implementation of the ERC-721 standard, which our NFT smart contract will inherit.
- `@openzeppelin/contracts/utils/Counters.sol` provides counters that can only be incremented or decremented by one. Our smart contract uses a counter to keep track of the total number of NFTs minted and set the unique ID on our new NFT. (Each NFT minted using a smart contract must be assigned a unique ID—here our unique ID is just determined by the total number of NFTs in existence. For example, the first NFT we mint with our smart contract has an ID of "1," our second NFT has an ID of "2," etc.)
- `@openzeppelin/contracts/access/Ownable.sol` sets up access control on our smart contract, so only the owner of the smart contract (you) can mint NFTs.

After our import statements, we have our custom NFT smart contract, which is surprisingly short — it only contains a counter, a constructor, and single function! This is thanks to our inherited OpenZeppelin contracts, which implement most of the methods we need to create an NFT, such as `ownerOf` which returns the owner of the NFT, and `transferFrom`, which transfers ownership of the NFT from one account to another.

In our ERC-721 constructor, you'll notice we pass 2 strings, "MyNFT" and "NFT." The first variable is the smart contract's name, and the second is its symbol. You can name each of these variables whatever you wish!

Finally, we have our function `mintNFT(address recipient, string memory tokenURI)` that allows us to mint an NFT! You'll notice this function takes in two variables:

- `address recipient` specifies the address that will receive your freshly minted NFT
- `string memory tokenURI` is a string that should resolve to a JSON document that describes the NFT's metadata. An NFT's metadata is really what brings it to life,

allowing it to have configurable properties, such as a name, description, image, and other attributes.

- `_tokenId`s: This is the latest token ID that corresponds to an NFT minted with this smart contract. `tokenId`s map to `tokenURI` which is the URL that contains the metadata of the corresponding NFT
- `_itemsSold`: Is a count of the number of items sold on the marketplace
- `owner`: This is the owner of the smart contract. The only address that can issue a withdrawal request.
- `listPrice`: The price (in ETH) any user needs to pay to list their NFT on the marketplace
- `ListedToken`: A solidity struct (similar to Javascript object) dictating the format an NFT's data is stored in
- `TokenListedSuccess`: Event emitted when a token is successfully listed
- `idToLisitedToken`: It is the mapping of all existing `tokenId`'s to the corresponding NFT token

Following is the structure of NFT Token:

```
struct Nft {  
    uint256 Id;  
    address payable deployer;  
    address payable seller;  
    uint256 Rate;  
    bool isOccupied;  
}
```

This mapping maps `tokenId` to token info and is helpful when retrieving details about a `tokenId`:

```
mapping(uint256 => Nft) private idToNft;
```



Following functions are used in this NFT marketplace

**generateNft()** - top level function when creating a token for the first time

```
function generateNft(string memory tokenAddr, uint256 Rate)
    public
    payable
    returns (uint256)
{
    _tokenIds.increment();
    uint256 newNftId = _tokenIds.current();
    _safeMint(msg.sender, newNftId);
    _setTokenURI(newNftId, tokenAddr);
    generateListedNft(newNftId, Rate);

    return newNftId;
}
```

**generateListedNft()** - Helps create the object of type Listed Token for the NFT and update the id .

```
function generateListedNft(uint256 tokenId, uint256 Rate) private {
    require(msg.value == listRate, "Kindly send correct Rate");
    require(Rate > 0, "Kindly set the feasible rate");

    idToNft[tokenId] = Nft(
        tokenId,
        payable(address(this)),
        payable(msg.sender),
        Rate,
        true
    );

    _transfer(msg.sender, address(this), tokenId);
    emit NftSuccess(
        tokenId,
        address(this),
        msg.sender,
        Rate,
        true
    );
}
```

**ToListed Token retrieveAllNFTs()** - get all the NFTs currently listed for sale on the marketplace

```
function retrieveAllNFTs() public view returns (Nft[] memory) {
    uint256 nftCount = _tokenIds.current();
    Nft[] memory Nfts = new Nft[](nftCount);
    uint256 curInd = 0;
    for (uint256 i = 0; i < nftCount; i++) {
        uint256 curId = i + 1;
        Nft storage curItem = idToNft[curId];
        Nfts[curInd] = curItem;
        curInd += 1;
    }
    return Nfts;
}
```

**marketplace retrieveMyNFTs()** - get all the NFTs from the current user on the marketplace

```
function retrieveMyNFTs() public view returns (Nft[] memory) {
    uint256 totalItems = _tokenIds.current();
    uint256 itemCnt = 0;
    uint256 curInd = 0;
    for (uint256 i = 0; i < totalItems; i++) {
        if (
            idToNft[i + 1].deployer == msg.sender ||
            idToNft[i + 1].seller == msg.sender
        ) {
            itemCnt += 1;
        }
    }
    Nft[] memory items = new Nft[](itemCnt);
    for (uint256 i = 0; i < totalItems; i++) {
        if (
            idToNft[i + 1].deployer == msg.sender ||
            idToNft[i + 1].seller == msg.sender
        ) {
            uint256 curId = i + 1;
            Nft storage curItems = idToNft[curId];
            items[curInd] = curItems;
            curInd += 1;
        }
    }
    return items;
}
```

**Transaction()** - the function that executes the sale on the marketplace

```
function Transaction(uint256 tokenId) public payable {
    uint256 Rate = idToNft[tokenId].Rate;
    address seller = idToNft[tokenId].seller;
    require(
        msg.value == Rate,
        "Plz give the Asking amount"
    );
    idToNft[tokenId].isOccupied = true;
    idToNft[tokenId].seller = payable(msg.sender);
    _itemsSold.increment();
    _transfer(address(this), msg.sender, tokenId);
    approve(address(this), tokenId);
    payable(deployer).transfer(listRate);
    payable(seller).transfer(msg.value);
    return;
}
```

## Compilation:

For compilation we used node command

```
$ npx hardhat compile
Compiling...
Compiled 1 contract successfully
```

## Deployment:

We deployed our smart contracts on Goerli using Hardhat and metamask private key, with alchemy as node service provider

```
},
goerli: {
  url: "https://eth-goerli.g.alchemy.com/v2/-VTrp56mvQyse2g7auFZWukuHgtTwBHE",
  accounts: [
    "a376fe04f4778f5c615c0b6f53176744425d8f626fc806c393a8ac3b8ac47915",
  ],
},
},
},
```

For deployment we used the command:

As general rule, you can target any network from your Hardhat config using:

```
npx hardhat run --network <your-network> scripts/deploy.js
```

This is the standard code provided by the Hardhat for the deployment of our contracts.

```
const { ethers } = require("hardhat");
const hre = require("hardhat");
const fs = require("fs");

async function main() {
  const [deployer] = await ethers.getSigners();
  const balance = await deployer.getBalance();
  const Marketplace = await hre.ethers.getContractFactory("NFTMarketplace");
  const marketplace = await Marketplace.deploy();

  await marketplace.deployed();

  const data = {
    address: marketplace.address,
    abi: JSON.parse(marketplace.interface.format("json")),
  };

  //This writes the ABI and address to the mktplace.json
  fs.writeFileSync("./src/Marketplace.json", JSON.stringify(data));
}
```

## Issues in Smart Contracts:

- Minting of NFT's takes time, we have to wait for some time before we start minting another NFT. We have to add ES6 javascript features like asyn() and awaitSyntax() for decreasing the time for minting NFT's.

**Pinata:**

We are using Pinata Interplanetary File System (IPFS) to store NFT's, which provides us with a hash to access the image wherever required.

- 1) . uploadFileToIPFS()- This function uploads the NFT image file to IPFS and then returns an IPFS URL which can be queried to obtain the image.
- 2) uploadJSONToIPFS(JSON)- This function takes the entire JSON to be uploaded as input and uploads it to IPFS. The value returned by the function is an IPFS URI which can be queried to get the metadata. This URI is super helpful when we want to retrieve the NFT metadata info later.

**Integrating Front End with Smart Contracts:**

These JavaScript files provides us with the functionalities to merge front end with Smart Contracts.

**1. SellNFT.js-** This file provides us with 3 main functionalities:

1. Upload the image to IPFS
2. Upload the metadata with an image to IPFS
3. Send the metadata tokenURI and price to the smart contract

**2. Marketplace.js-** In this file we are pulling all the NFTs from the smart contract.

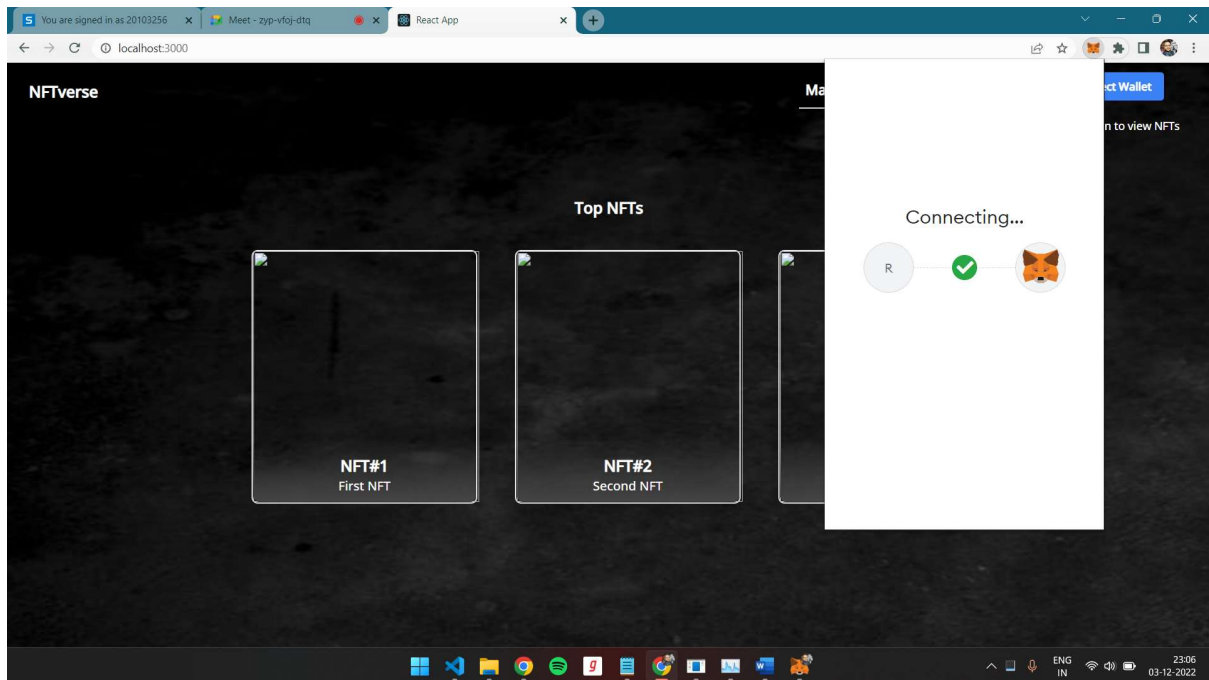
**3. Profile.js-** This file pulls all the NFTs that the logged in user owns

**4. NFTPage.js-** This is the individual page for every NFT, which serves two functionalities:

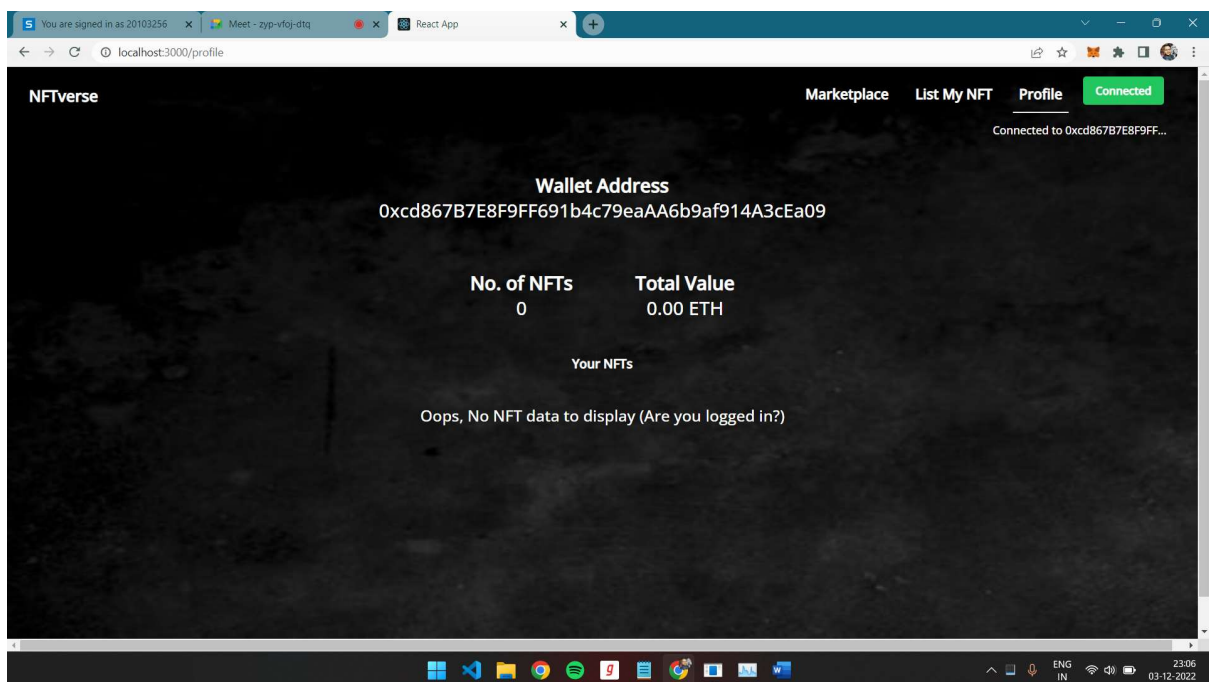
1. display all the data of a particular NFT
2. let any user buy it with a "Buy this NFT" button

## Chapter 5: Testing

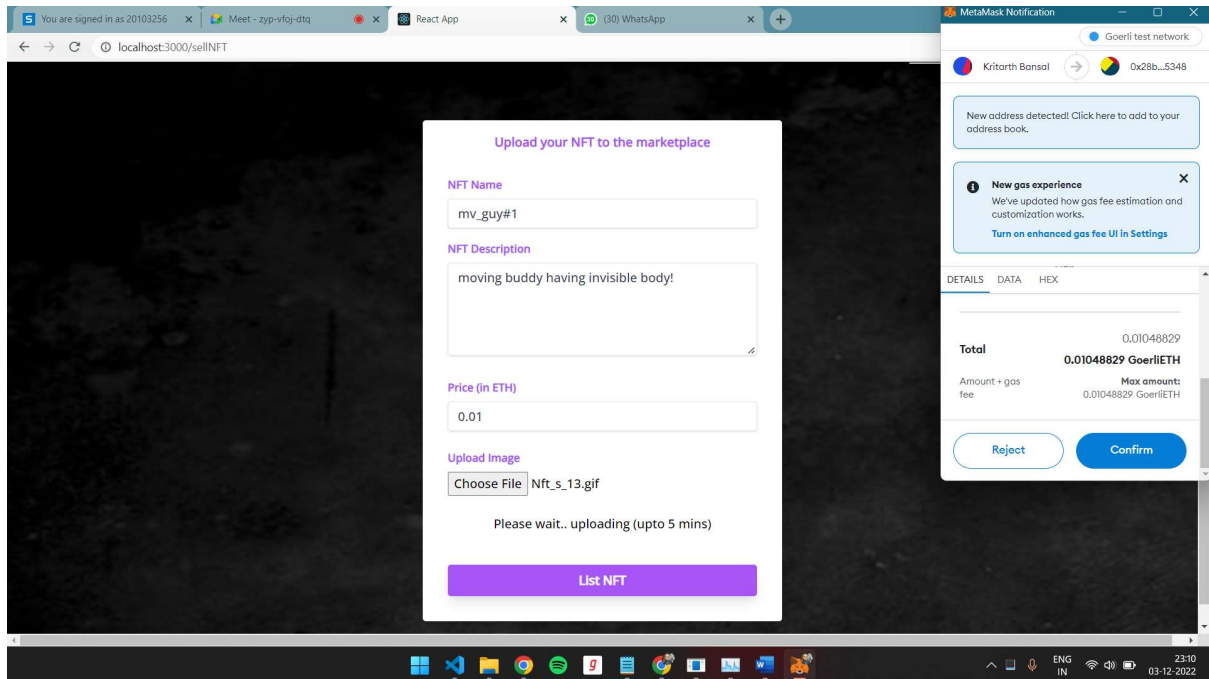
Establishing a connection to metamask wallet:



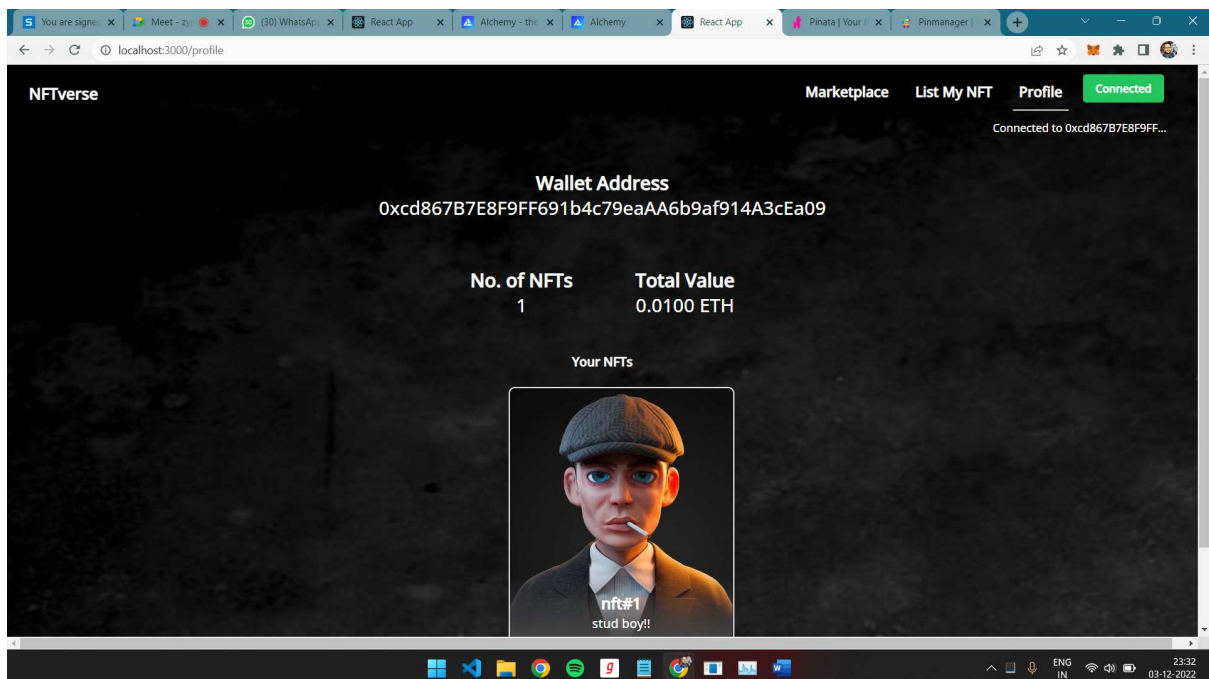
Successfully connected to metamask wallet (0xcd...). Initially there are no NFT's.



Minting NFT's and confirming transaction on metamask with the help of generateNft() and generateListedNft() function:

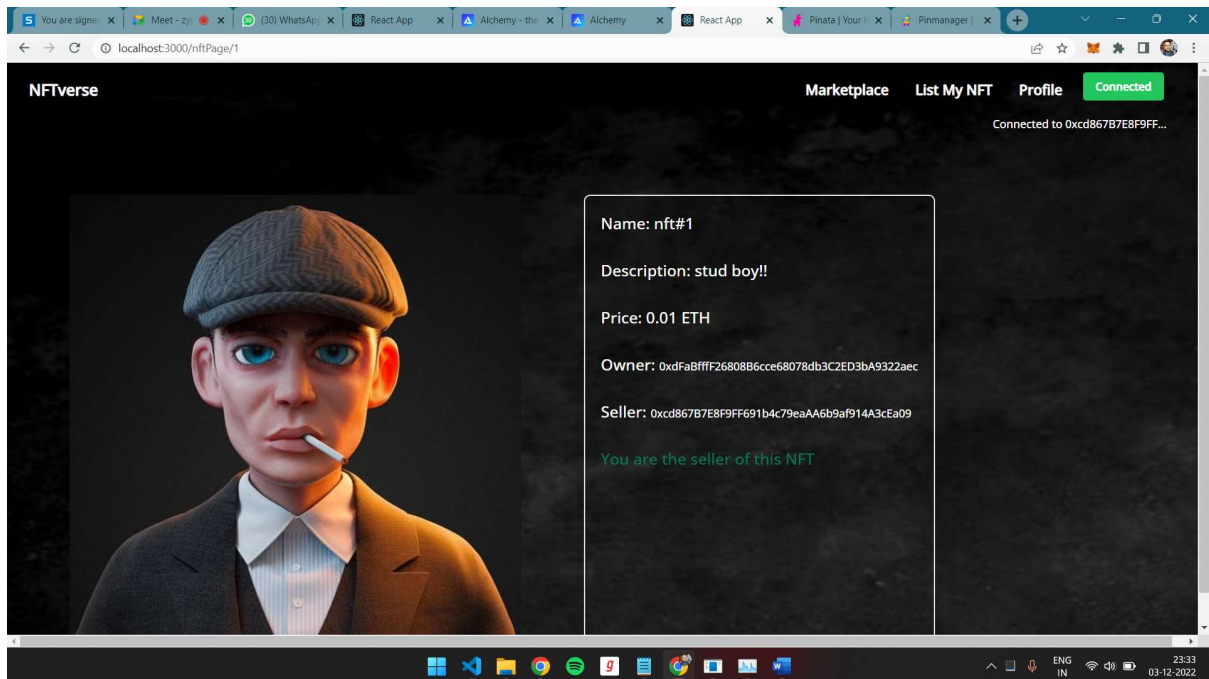


NFT has been successfully listed:

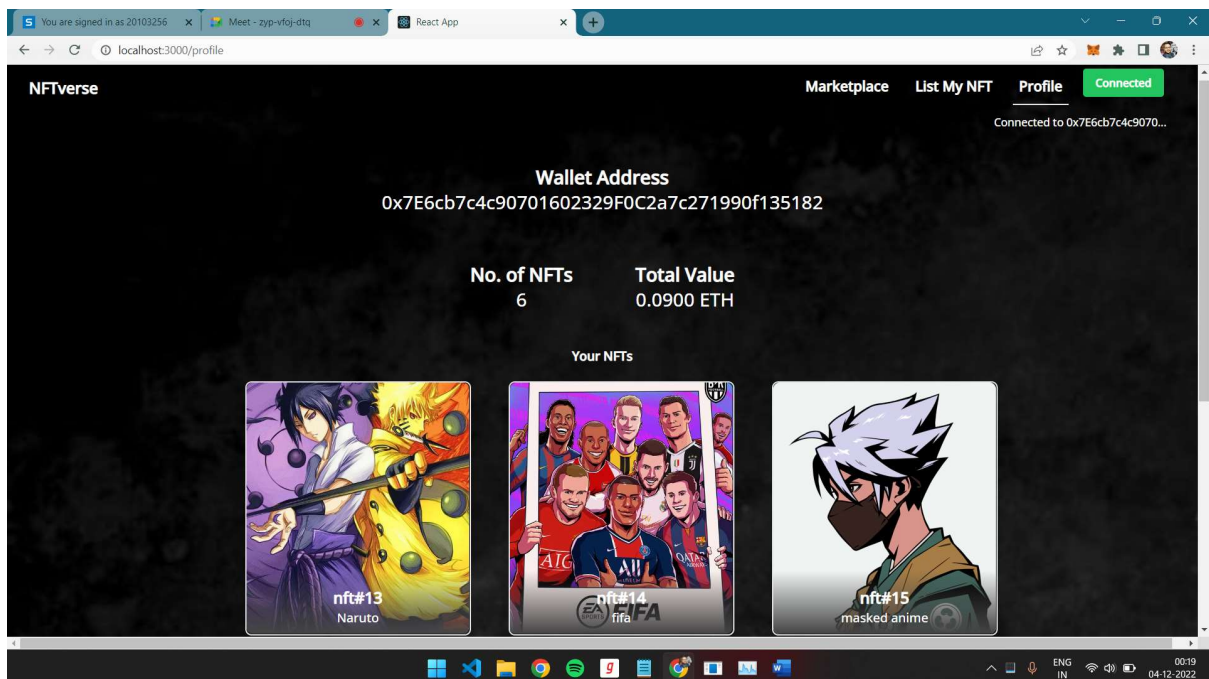




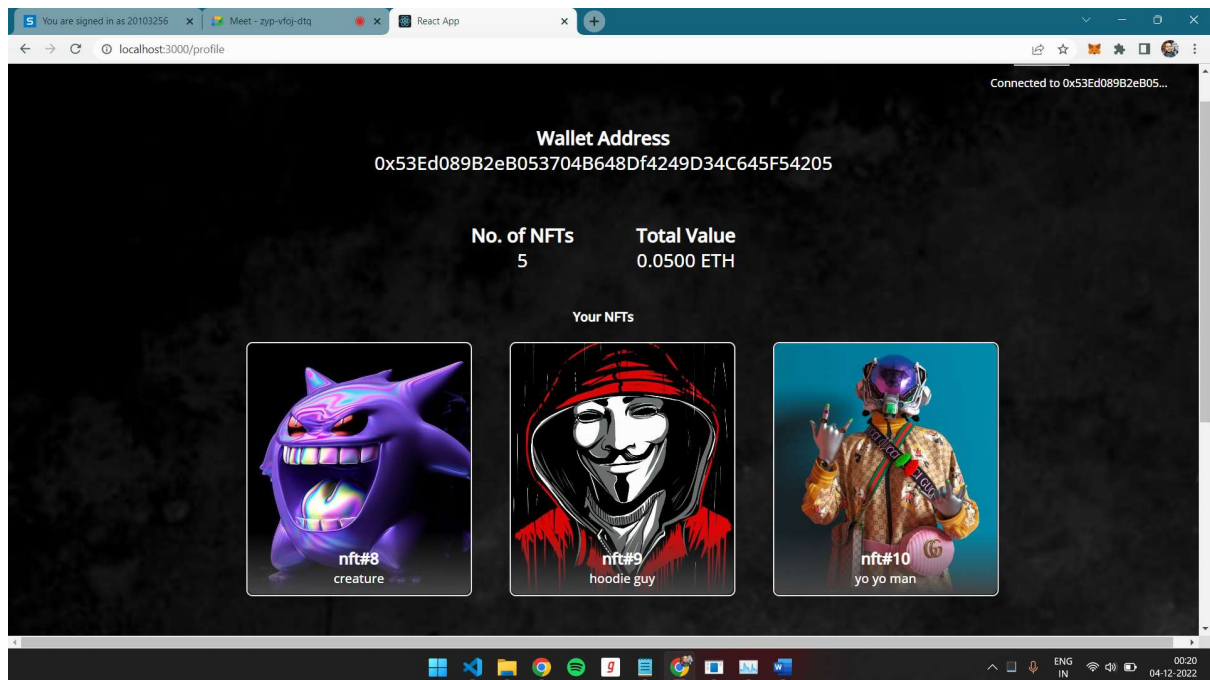
We can open the NFT to check its details. As this is the profile page of seller we are getting the message “**You are the seller**”.



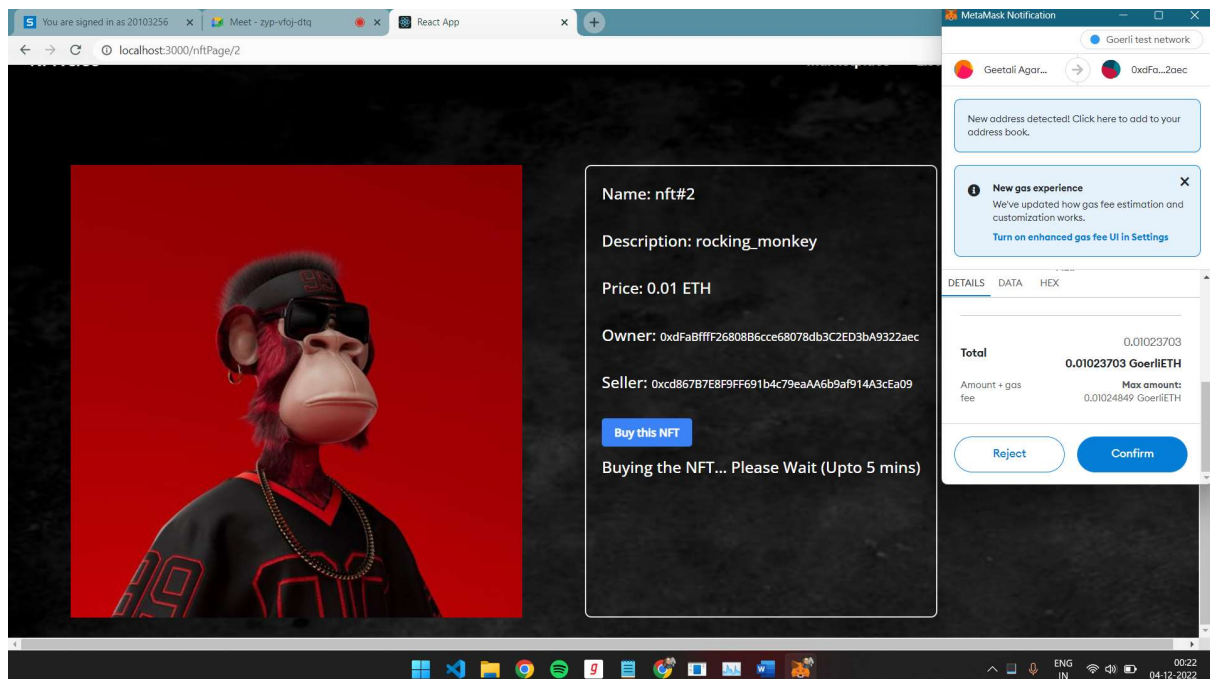
Profile page of another user which is being displayed through **retrieveMyNft()** function :



Profile page of another user which is being displayed through `retrieveMyNft()` function :

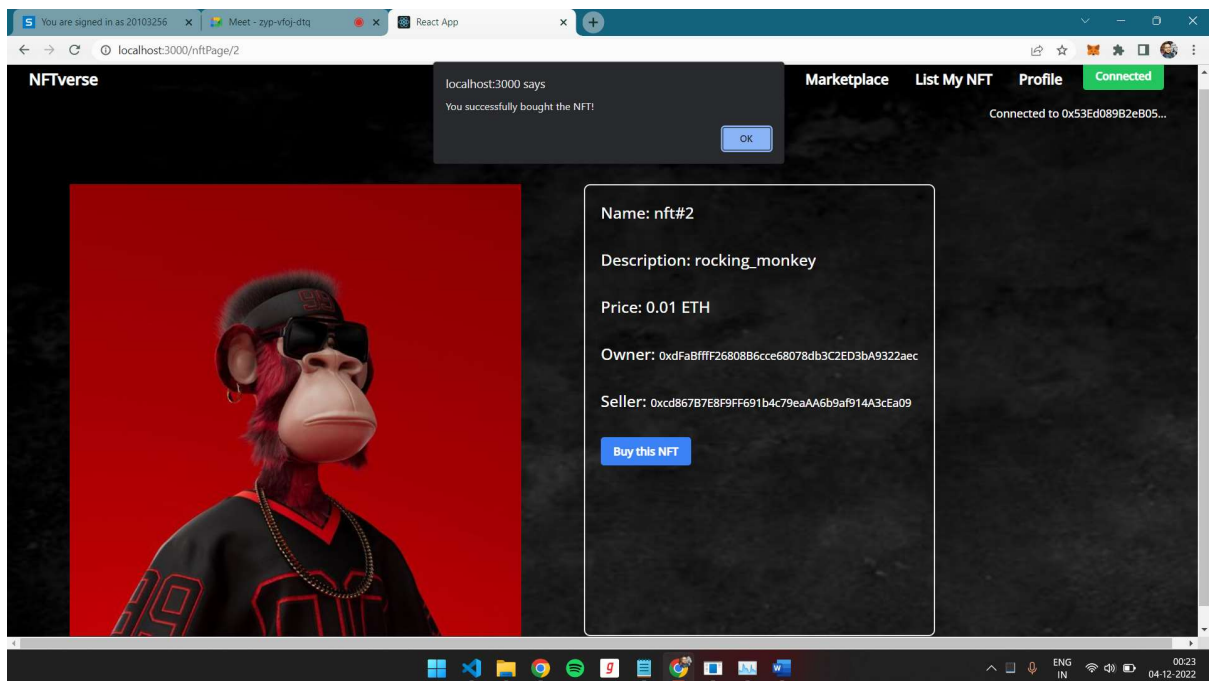


Buying Nft from another account using `executeSale()` function of the Smart Contract:

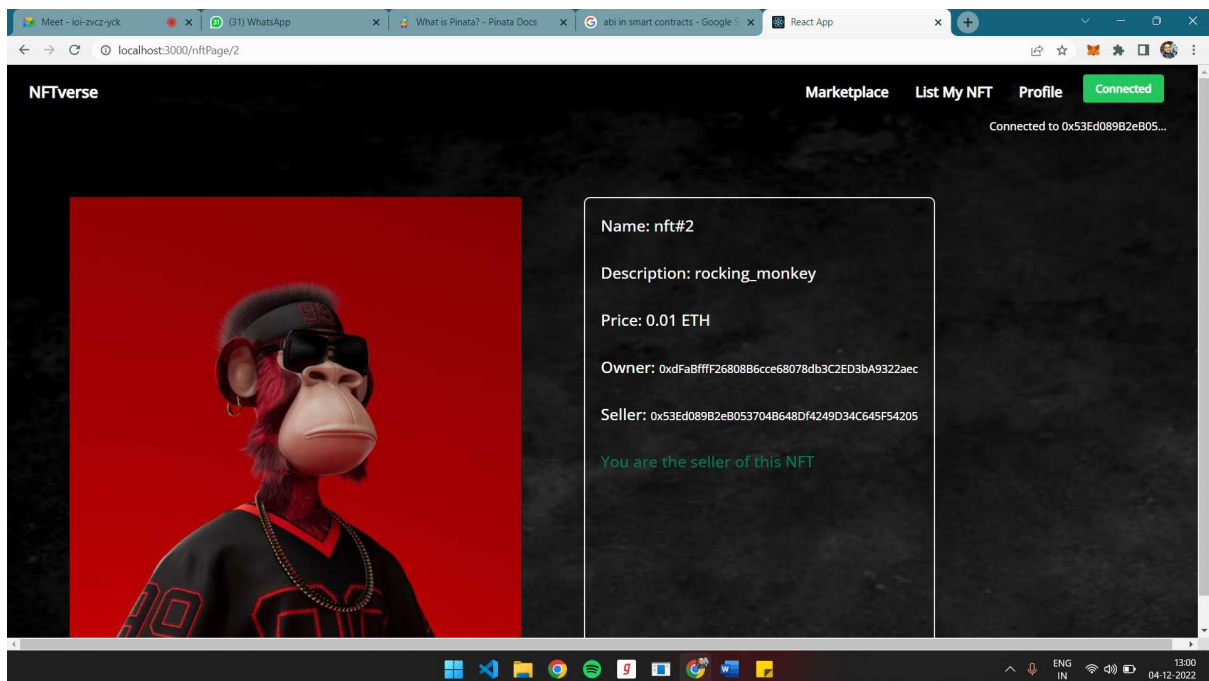




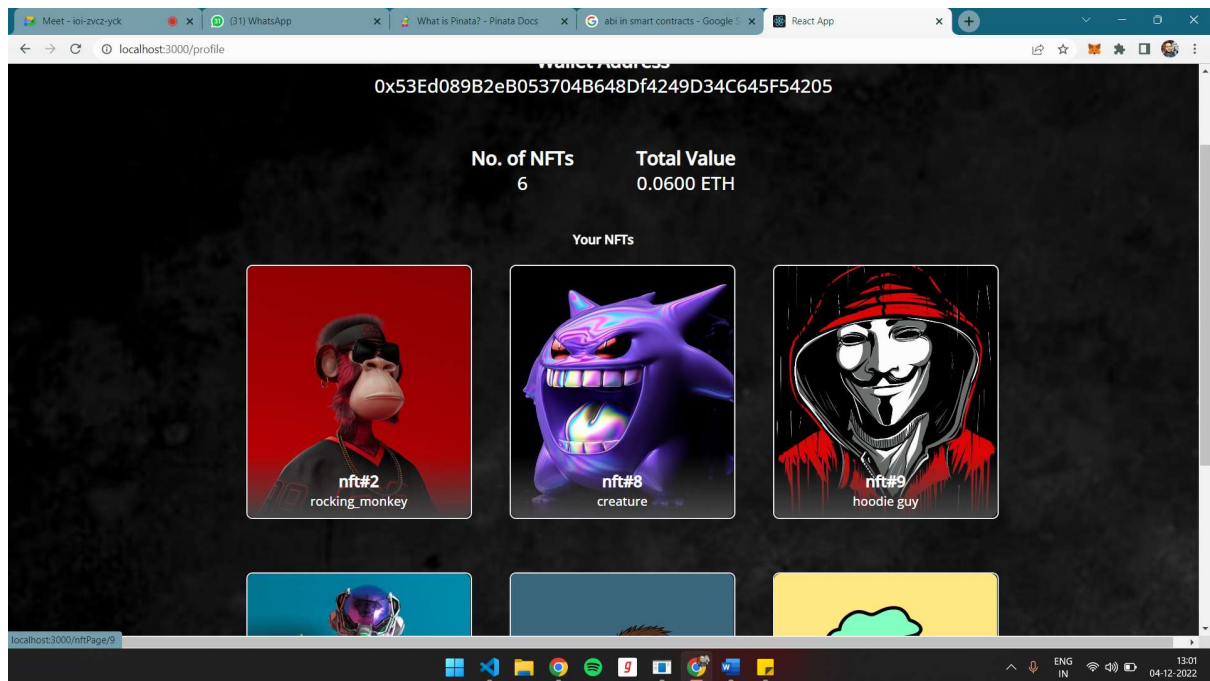
Nft Successfully bought:



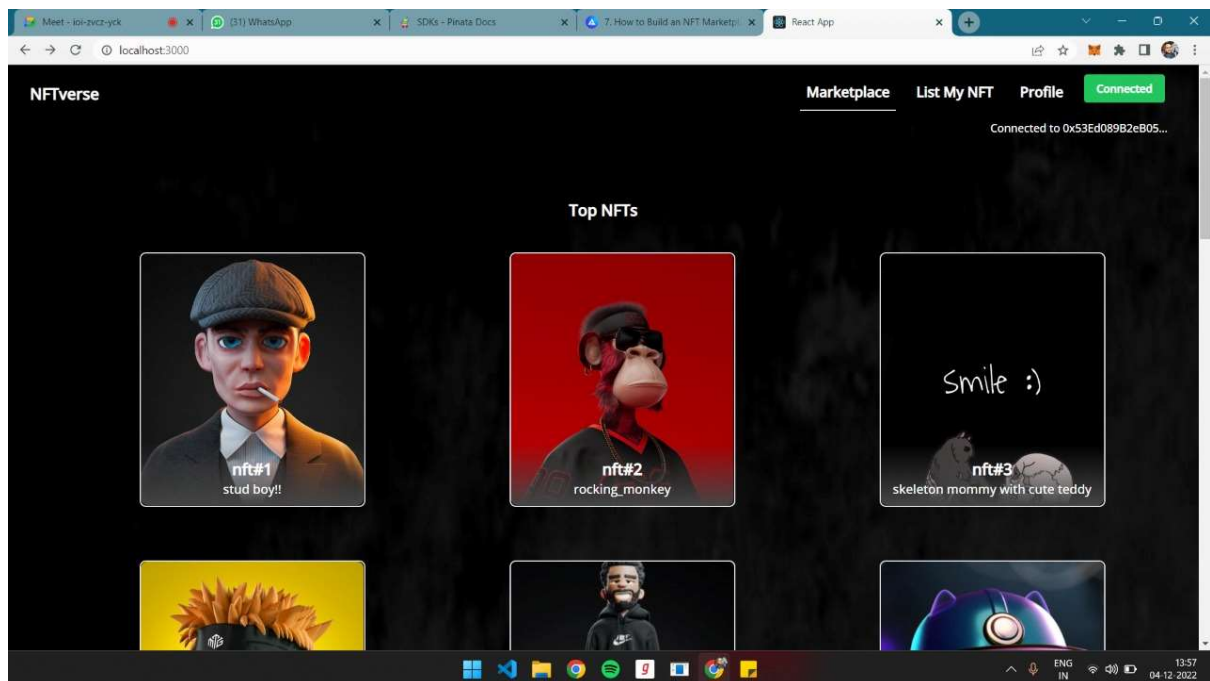
Ownership of the Nft has been successfully changed to the new buyer.



Nft has been successfully added to the profile page of the buyer.



All the Nft's which have been uploaded by the various sellers are being displayed on the Marketplace section through get marketplace retrieveAllNFT() function.



## **Chapter 6: Conclusion, Future Work and References**

### **Conclusion:**

So we have created a fully functional NFT marketplace where NFTs can be conveniently bought and sold. The ever-growing popularity of NFTs has sparked the need to have more NFT marketplaces. This is to allow storing of these digital assets and further make them accessible for easy buying and selling. In a way, it can be stated that NFT marketplace development would be a major driver of the blockchain industry's growth and success.

### **Future Work:**

We can add functionality to let users list pre-existing NFTs to the marketplace.

We can add Royalties such that the original NFT creator gets 10% of the proceeds every time that NFT gets sold.

## REFERENCES:

- [1]: [https://www.researchgate.net/publication/327069975\\_Blockchain-based\\_Distributed\\_Marketplace](https://www.researchgate.net/publication/327069975_Blockchain-based_Distributed_Marketplace)
- [2]: <https://www.ijlmh.com/paper/non-fungible-tokens-nfts-the-future-of-digital-collectibles/>
- [3]: [https://www.researchgate.net/publication/357900561\\_NFTs\\_Applications\\_and\\_Challenges](https://www.researchgate.net/publication/357900561_NFTs_Applications_and_Challenges)
- [4]: <https://www.leewayhertz.com/develop-nft-marketplace-platform/>
- [5]: <https://ethereum.org/en/>
- [6]: <https://trufflesuite.com/>
- [7]: <https://hardhat.org/>
- [8]: <https://www.pinata.cloud/>
- [9]: <https://tailwindcss.com/>
- [10]: <https://www.alchemy.com/>
- [11]: <https://goerli.etherscan.io/>
- [12]: <https://docs.openzeppelin.com/contracts/3.x/erc721>