# INTRODUCTION TO BLOCKCHAIN

# PBL REPORT


**Department of CSE/IT**

**Jaypee Institute of Information Technology University, Noida**

**2021-2022**

## NFT MARKET PLACE

**Group Members:**

Hardik Agarwal 9919103007 (F1)

Suryashankar Das 9919103018 (F1)

Satyam Chaubey 9919103019 (F1)

Abhir Raj Shrivastava 9919103059 (F2)

**Submitted To:**

Himanshu Agarwal

# ABSTRACT

The immense popularity of the NFT marketplaces for trading non-fungible tokens is rising day by day. This is a perfect opportunity for business platforms to enter the digital space by having complete knowledge of the NFTs. Hence, the first and most important term for a business platform to be aware of what are NFTs? Therefore, NFT or Non-Fungible Tokens are cryptographic assets that are created on blockchain technology, and have unique identification codes and meta-data, which makes them distinguishable, distinct, and completely unique. In fact, even cryptocurrencies are not distinct, since they are identical to one another. This is the reason that they can be traded at equivalency for commercial usage, but not NFT, which are 100% unique. In our project we have created a NFT market place where user can create, buy or sell their digital art as NFT on blockchain system.



Keywords*: Blockchain, Non-Fungible, Crypto, Metadata, Market, Art*

# 1.0 Introduction

NFT marketplace is a platform that allows users to secure their NFTs and trade them. There is a wide range of methods to trade NFTs. The most common method is the auction method, where the NFTs are sold in an auction. And the other way is to buy the NFT at its fixed rate. For any marketplace to be put in use, a crypto wallet is necessary, and it should be synced with the marketplace effectively. This is a perfect platform for business entities to begin their careers in the NFT space. Thus, a business platform has to build an NFT marketplace from scratch in order to enjoy every benefit that was provided by the NFT space.

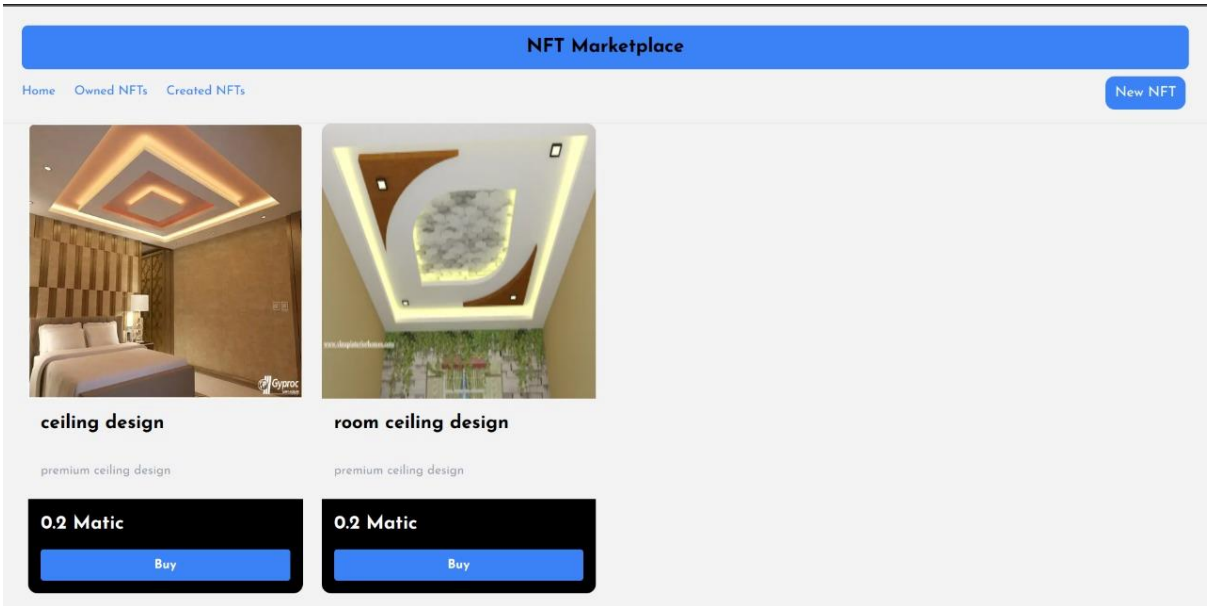


Fig: Landing page of our Project

## 1.1. Working of NFT Market Place

- The user should register into the platform and connect a crypto wallet.
- The creation of NFT takes place, and all the required parameters are defined.
- The next step would be to list the minted NFTs for sale and proceed to wait for the completion of moderation.
- The buyers who enter the platform will bid when the auction is running actively.
- When the desired NFT is bought, the marketplace transfers cryptocurrencies and NFTs to the connected crypto wallet.

### 1.2. Factor influencing NFT market Place

- Transparency
- Decentralization
- Security
- Monetization Mode
- Smart Contract

## 2.1 Software Requirements

### Frontend

- Next.js
- Ethers.js
- Tailwind

### Blockchain and Smart Contracts

- Alchemy: Polygon Mumbai Testnet
- Solidity
- Hardhat
- Infura

### Data Storage

- IPFS

## 2.2 Hardware Requirements

- 2GB ram
- 1.2 GHz processor
- Intel i5
- Windows 7/8/8.1/10/11

## 3.0 Design of the System

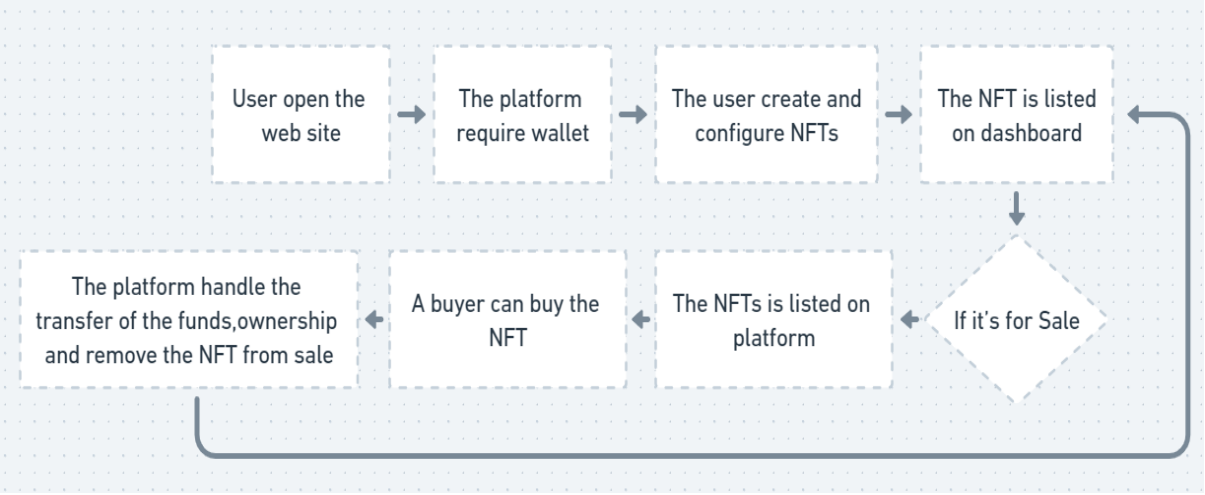


*Fig: Design of the system*

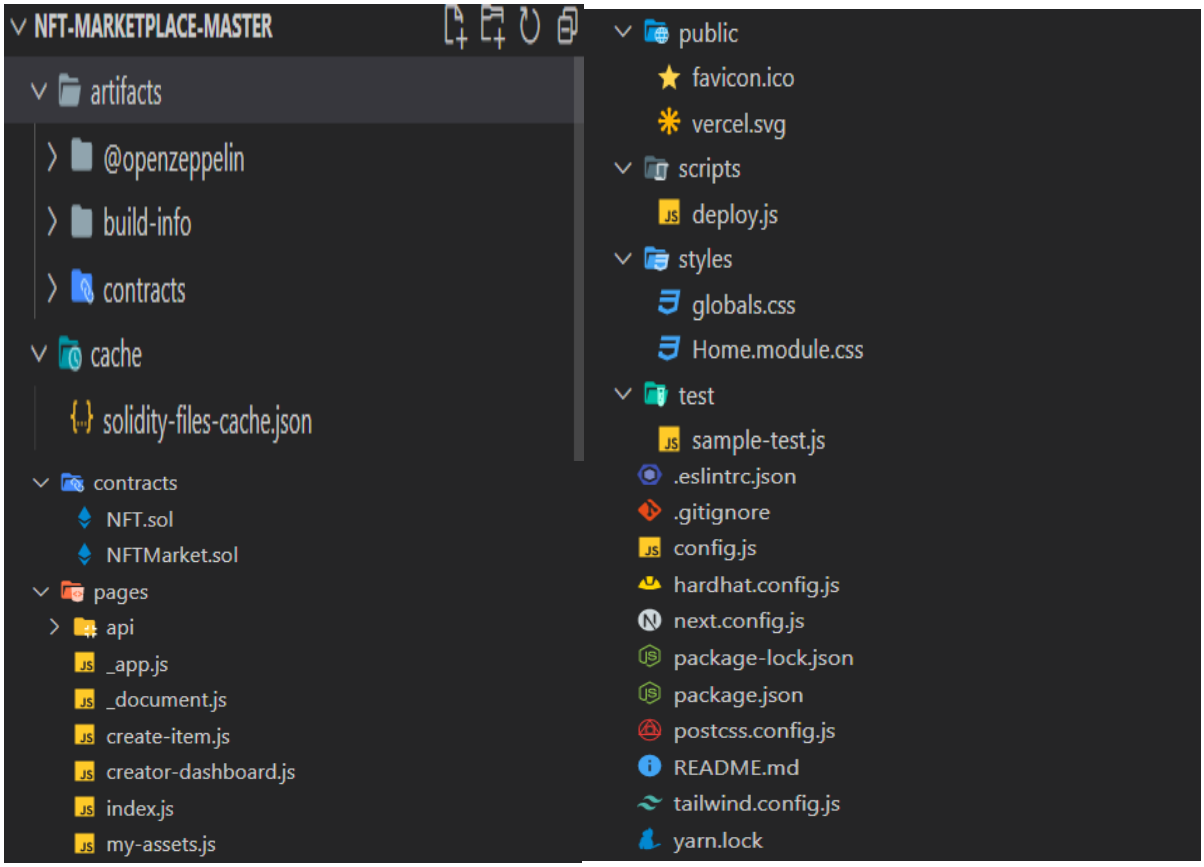


*Fig: Folders in vscode*

## 4.0 Implementation

When a user puts an NFT for sale, the ownership of the item will be transferred from the creator to the marketplace contract. When a user purchases an NFT, the purchase price will be transferred from the buyer to the seller and the item will be transferred from the marketplace to the buyer. The marketplace owner will be able to set a listing fee. This fee will be taken from the seller and transferred to the contract owner upon completion of any sale, enabling the owner of the marketplace to earn recurring revenue from any sale transacted in the marketplace. The marketplace logic will consist of just one smart contract which allows users to mint NFTs and list them in a marketplace. I believe this is a good project because the tools, techniques, and ideas we will be working with lay the foundation for many other types of applications on this stack – dealing with things like payments, commissions, and transfers of ownership on the contract level as well as how a client-side application would use this smart contract to build a performant and nice-looking user interface.

## Smart Contracts: -

### NFT.sol

```sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

// ERC721 Contract
contract NFT is ERC721URIStorage {
    using Counters for Counters.Counter;
    Counters.Counter private _tokenIds;
    address contractAddress;

    constructor(address marketplaceAddress) ERC721("NFT Token", "NFT") {
        contractAddress = marketplaceAddress;
    }

    // NFT mint function
    function mintToken(string memory tokenURI) public returns (uint256) {
        _tokenIds.increment();
        uint256 newItemId = _tokenIds.current();

        _mint(msg.sender, newItemId);
        _setTokenURI(newItemId, tokenURI);
        setApprovalForAll(contractAddress, true);
        return newItemId;
    }
}
```

**NFTMarket.sol**

```
// SPDX-License-Identifier: MIT
pragma solidity ^ 0.8.4;

import "@openzeppelin/contracts/utils/Counters.sol";
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";

contract NFTMarket is ReentrancyGuard
{
    using Counters for Counters.Counter;
    Counters.Counter private _nftIds;
    Counters.Counter private _nftsSold;

    address payable owner;
    uint256 listingPrice = 0.025 ether;

    constructor()
    {
     owner = payable(msg.sender);
    }

    struct NftItem
    {
     uint256 itemId;
     address nftContract;
     uint256 tokenId;
     address payable seller;
     address payable owner;
     uint256 price;
     bool sold;
    }

    mapping(uint256 = > NftItem) private idToNftItem;

    event NftCreated(
        uint256 indexed itemId,
        address indexed nftContract,
        uint256 indexed tokenId,
        address seller,
        address owner,
        uint256 price,
        bool sold);

```

## 5.0. Testing Details

For Testing purpose, we have used “chai”.

```
const { expect } = require("chai");
const { ethers } = require("hardhat");

describe("NFTMarket", function () {
 it("Should create and execute market sales", async function () {
   const Market = await ethers.getContractFactory("NFTMarket");
   const market = await Market.deploy()
   await market.deployed()
   const marketAddress = market.address

   const NFT = await ethers.getContractFactory("NFT");
   const nft = await NFT.deploy(marketAddress)
   await nft.deployed()
   const nftContractAddress = nft.address

   let listingPrice = await market.getListingPrice()
   listingPrice = listingPrice.toString()

   const auctionPrice = ethers.utils.parseUnits('100', 'ether')

   await nft.createToken("https://www.mytokenlocation.com")
   await nft.createToken("https://www.mytokenlocation2.com")

   await market.createMarketItem(nftContractAddress, 1, auctionPrice, { value: listingPrice })
   await market.createMarketItem(nftContractAddress, 2, auctionPrice, { value: listingPrice })
```

# References

[1] Kireyev, Pavel. "NFT Marketplace Design and Market Intelligence." (2022).

[2] Vasan, Kishore, Milán Janosov, and Albert-László Barabási. "Quantifying NFT-driven networks in crypto art." *Scientific reports* 12.1 (2022): 1-11.

[3] Gilbert, Sam. "Crypto, web3, and the Metaverse." (2022).

[4] https://webkul.com/blog/complete-guide-to-develop-nft-marketplace-from-scratch/

[5] https://www.leewayhertz.com/develop-nft-marketplace-platform/