

# NFT Marketplace for Luxury Goods Authentication

*Using Hardhat for development of framework, JavaScript for testing/react, IPFS (Metadata storage), Solidity for writing smart contract, Ethers/blockchain interaction and React Routers (navigational components)*



# Executive Summary

- Presently counterfeit items for luxury/designer goods is becoming an increasingly accessible process. Currently, this sector has fallen behind in using innovative technology for the purpose of authenticating genuine, luxury/designer goods. This is precisely where the Fintech space can provide better security and authentication processes, bringing the designer industry into the 21<sup>st</sup> century.
- To accomplish this task, a unique NFT marketplace presents the opportunity for a platform exclusively dedicated to NFTS created for the purpose of authentication. Luxury items requiring the purchase of its own NFT can be used to identify authentic, original items.



## Selected Models

- Luxury Product Purchase Using NFTs
  - Main tools and technology created for platform:
    - Using Hardhat for development of framework, JavaScript for testing/react, IPFS (Metadata storage), Solidity for writing smart contract, Ethers/blockchain interaction and React Routers (navigational components)
    - This project required a variety of technology and coding languages as we were looking to not only create NFTs for the purpose of authenticating luxury items, but also to create a platform specifically dedicated to this need
- Features of Platform
  - Essentially the same as the already popular marketplace apps within the NFT space (OpenSea for example), making it simple to navigate
    - The unique feature here is the purpose and representation of this marketplace, a space exclusively for designer items as nothing currently exists for such a purpose
    - A platform with the same features of other NFT marketplace apps, but unique in its purpose, making it more desirable to use along with its authentication benefits
    - Platform has tabs linked to pages featuring your purchases, your listed items as well as, and most importantly, tools for creating your own NFT

# Tools and Technology



## Ether

- Buy and sell luxury items attached to their own NFT using Ether
- Digital data (Non Fungible Tokens) used to identify and verify a physical luxury/designer item on a blockchain
- Through Ether blockchain process, the items can be tracked (therefore authenticated) from the manufacturing used by the brand name all the way to the retailers and buyers.
- This process continues as the item is resold by other buyers tracked using the public ledger and the power of blockchain

## Solidity

- Ethereum contracts are built using Solidity (statically typed programming language)
- For this marketplace we created an NFT as well as Marketplace contract, Solidity is the language used to create smart contracts on Ethereum blockchain
- NFT contract is used to mint (publish on the blockchain) an NFT
- Marketplace contract manages the NFT items by providing a space for buying and selling

# Tools and Technology



## Hardhat

- JavaScript (JS) based framework that aids developers when testing, debugging, deploying or compiling dApps
- We used it for testing and deploying the contracts before publishing to the real Ethereum network
- Upon starting your node (the moderators that build the infrastructure of a decentralized network), process continues as the item is resold by other buyers tracked using the public ledger and the power of blockchain
- Fake wallets created in Hardhat are connected to Metamask (used to connect to blockchains for buying and selling)



## Website

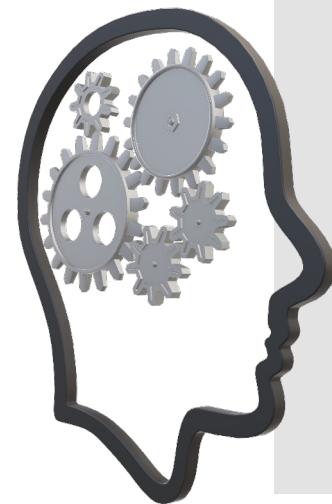
- JS, HTML and CSS coding used for creating the webpage to manage the platform's appearance and its dynamic functionality – frontend
- React-Router-Dom dependency is used to help navigate across different tabs located in the navigation bar
- React, JS and Solana is used in the backend. React and JS is used for testing and deployment. Solana is used to mint smart contracts on the Ethereum blockchain

# Tools and Technology

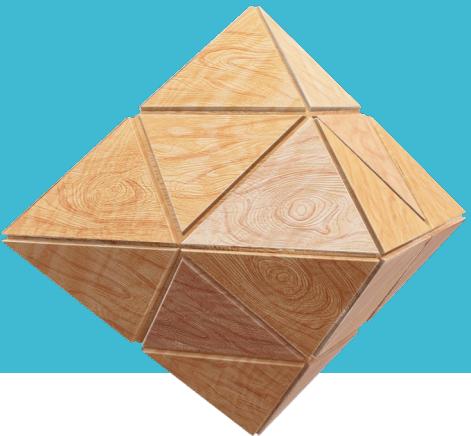


## Inter-Planetary File System (IPFS) for Metadata Storage

- IPFS is a file storage system where NFT metadata can be stored
- A large peer-to-peer network where computers across the globe can store and relay files
- The hash (NFT metadata) created and files from the NFT is stored in IPFS. IPFS can then provide a URL that can be shared for validation purposes
- IPFS provides extra level of security as well as ease of access when needed



# Platform Setup



- **Install NodeJS packages and dependencies**

- Create a package.json file in project main folder, which contains all packages to create and deploy NFT marketplace
- These packages along with their versions are chosen based on researching similar projects in blockchain and bootstrap that leverage react and solidity
- *I chose to use a process different from what we were learning in class for the purpose of gaining more exposure in other languages being used to create blockchain.*

```
(dev) ➜ UoFT node -v  
v17.1.0  
(dev) ➜ UoFT cd Project3  
(dev) ➜ Project3 git:(main) npm install
```

1. Check node version (should be <= 17.2.0)
2. Install dependencies – {package.json}

```
(dev) ➜ Project3 git:(main) npm install  
npm WARN deprecated multiaddr-to-uri@8.0.0: This module is deprecated, please upgrade to @multiformats/multiaddr-to-uri  
npm WARN deprecated multiaddr@10.0.1: This module is deprecated, please upgrade to @multiformats/multiaddr  
  
added 2389 packages, and audited 2393 packages in 22s  
  
217 packages are looking for funding  
  run `npm fund` for details  
  
69 vulnerabilities (1 low, 9 moderate, 32 high, 27 critical)  
  
To address issues that do not require attention, run:  
  npm audit fix  
  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
  
Run `npm audit` for details.
```

# Platform Setup



- **Start a local blockchain instance using hardhat**

- 20 wallets are created at instantiation with 1000ETH for testing
- We will use Account #0 as default for testing

```
(dev) → Project3 git:(main) npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
=====

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: 0x3c44cddb6a900fa2b585dd299e03d12fa4293bc (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

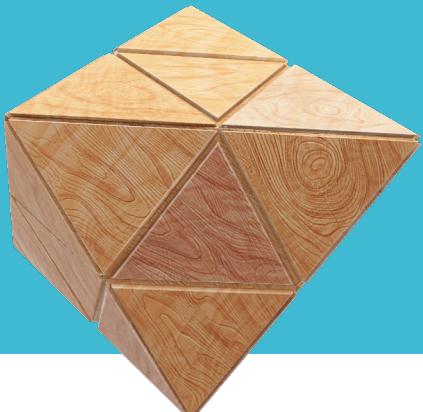
Account #6: 0x976ea74026e726554db657fa54763abd0c3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e
```

# Platform Testing



- **Run 'Test' script to check code validity**
    - 7 tests were written to continuously and effectively test during development
      - Contract Deployment (x2) – NFT and Marketplace
      - Minting an NFT
      - Transfer NFT to Marketplace
      - Check price is not zero
      - Purchasing an NFT
      - Check item id

# Platform – Contracts



- **NFT Contract**

- NFT Minting using [openzeppelin](#) **ERC721URIStorage** library

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

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

contract NFT is ERC721URIStorage {
    uint public tokenCount;
    constructor() ERC721("DApp NFT", "DAPP"){}
    function mint(string memory _tokenURI) external returns(uint) {
        tokenCount++;
        _safeMint(msg.sender, tokenCount);
        _setTokenURI(tokenCount, _tokenURI);
        return(tokenCount);
    }
}

fallback () external {}
```

- **Marketplace Contract**

- Provides functionality to
  - Make Item (NFT)
  - Purchase NFT
  - Get total price including marketplace fees

```
contract Marketplace is ReentrancyGuard {

    // Variables
    address payable public immutable fee_account; // the account that receives fees
    uint public immutable feePercent; // the fee percentage on sales
    uint public itemCount;

    struct Item { ... }

    // itemId -> Item
    mapping(uint => Item) public items;

    event Offered(...);
    event Bought(...);

    constructor(uint _feePercent) {...}

    // Make item to offer on the marketplace
    function makeItem(IERC721 _nft, uint _tokenId, uint _price) external nonReentrant {...}
    function purchase_nft(uint _itemId) external payable nonReentrant {...}
    function getTotalPrice(uint _itemId) view public returns(uint){
        return(items[_itemId].price*(100 + feePercent))/100;
    }
}
```

# Platform - Contract Deployment



- **Deploy NFT and marketplace contracts**

- Deploy NFT and Marketplace contracts created using solidity
  - Note: Account Balance is shown in Wei (Smallest fraction of ETH)

```
(dev) ➔ Project3 git:(main) npx hardhat run src/backend/scripts/deploy.js --network localhost
Compiling 14 files with 0.8.4
Solidity compilation finished successfully
Deploying contracts with the account: 0xf39Fd6e51aad88f6F4ce6aB8827279cffFb92266
Account balance: 10000000000000000000000000000000
```

- Contracts registered on the block chain

#### eth\_sendTransaction

```
Contract deployment: Marketplace
Contract address: 0xe7f1725e7734ce288f8367e1bb143e90bb3f0512
Transaction: 0x76d5056d3de03b479c5f586b8047f2a79a848297c5c30d1dd598aca10de6b6f8
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
Value: 0 ETH
Gas used: 988487 of 988487
Block #2: 0x89b4f8f445e6f0b57687b68c7730b1319529e069ec08e0fba737530392065cfa
```

#### eth\_sendTransaction

```
Contract deployment: NFT
Contract address: 0x5fbdb2315678afecb367f032d93f642f64180aa3
Transaction: 0xfaa9ee95061d5b0b30fc35e83a2ba9056ee5594db54d4de6ee3ccebafa842bd3
From: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
Value: 0 ETH
Gas used: 2526534 of 2526534
Block #1: 0x3c8e05707c51d93900386b066e440c5597923ce90fada51070f3534c09ce44e4
```

# Platform - MetaMask Connection



- **Connect Wallet to Metamask**
  - Connect hardhat network to MetaMask
    - Add Manual Network to MetaMask
    - Add the required information to connect to hardhat

Settings

Search in Settings

Networks

Add a network

Network name: HardHat Node

New RPC URL: http://localhost:8545

Chain ID: 31337

Currency symbol: ETH

Block explorer URL (Optional)

Cancel Save

HardHat Node

Account 3  
0xf39...2266

9999.9954 ETH  
\$16,956,792.21 USD

Buy Send Swap

Unavailable on this network

Activity

Portfolio site

Backup your Secret Recovery Phrase to keep your wallet and funds secure.

Backup now

Approve token with n...

# Platform – IPFS (Infura)

## Data Collection

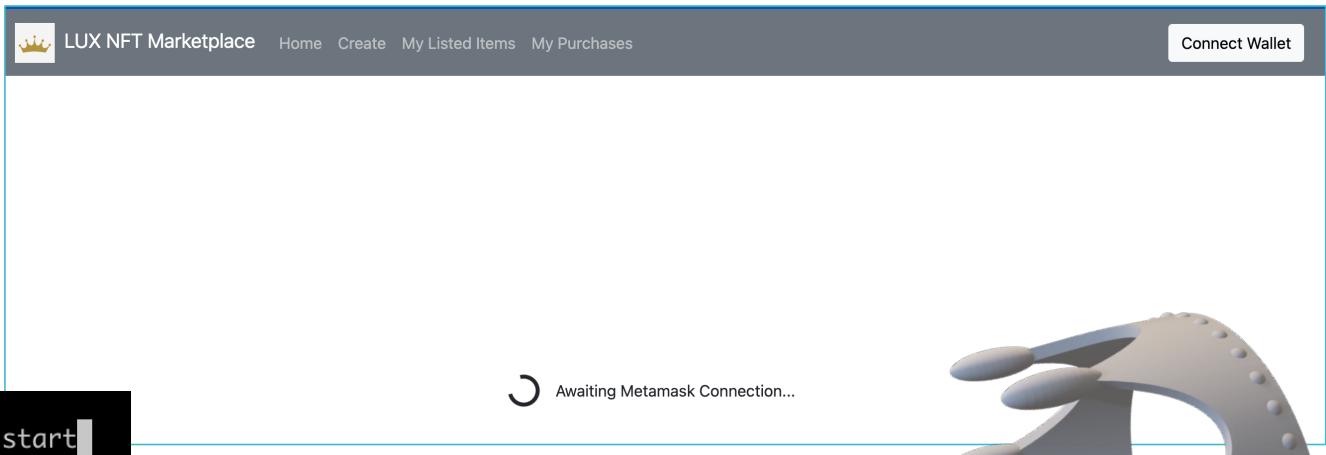


- **Create IPFS project on Infura**
  - Infura.io is used to store NFT files and metadata

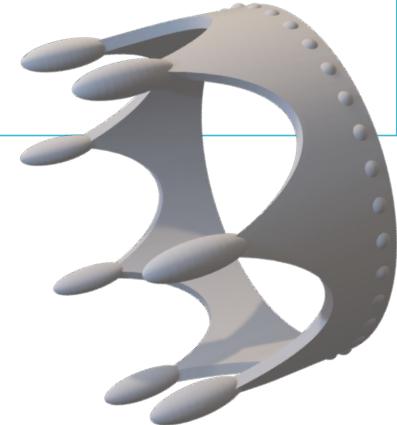
The screenshot shows the Infura web interface. At the top, there's a navigation bar with the Infura logo, 'INFURA' text, and links for 'Keys', 'Stats', 'TXNs', 'IPFS', 'Support', and 'Settings'. Below the navigation, there's a breadcrumb menu with 'DataThread' and tabs for 'GENERAL', 'SECURITY', and 'KEY SHARING'. The 'GENERAL' tab is active, indicated by an orange underline. The main content area is titled 'Details' and contains a 'PROJECT NAME' field with 'DataThread' and a 'SAVE' button. Below this is a 'Project Summary' section with fields for 'PROJECT ID' (2LTW8u7SSrQ9UtvTH682i05nGd6), 'API Key Secret' (redacted), and 'IPFS API ENDPOINT' (<https://ipfs.infura.io:5001>). The entire screenshot is framed by a light gray border.

# Platform - Frontend

- Start marketplace front
  - Click on Connect Wallet to connect to MetaMask Wallet



```
(dev) ➔ ~ cd UofT/Project3
(dev) ➔ Project3 git:(main) ✘ npm run start
```



Compiled successfully!

You can now view `starter_kit` in the browser.

Local: http://localhost:3001  
On Your Network: http://10.0.0.200:3001

Note that the development build is not optimized.  
To create a production build, use `npm run build`.

```
assets by path static/ 4.46 MiB
asset static/js/bundle.js 4.36 MiB [emitted] (name: main) 1 related asset
asset static/media/market.3c97c27021892978b385.png 98.6 KiB [emitted] [immutable] [from: src/frontend/components/market.png] (auxiliary name: main)
asset index.html 1.67 KiB [emitted]
asset asset-manifest.json 270 bytes [emitted]
cached modules 4.06 MiB (javascript) 98.6 KiB (asset) [cached] 847 modules
runtime modules 29.5 KiB 16 modules
./node_modules/webpack-dev-server/client/index.js?protocol=ws%3A&hostname=0.0.0.0&port=3001&pathname=%2Fws&logging=none&reconnect=10 6.59 KiB [built] [code generated]
webpack 5.69.1 compiled successfully in 3141 ms
```

# Platform - Create NFT

## • Create NFT

- Upload digital object (i.e. image etc.) to IPFS
- Assign Name, Description and ETH to the NFT
- Mint NFT of the uploaded object
- Publish NFT on the marketplace



LUX NFT Marketplace

Home Create My Listed Items My Purchases



Choose File

2CF3A877-63AB-44B4-A642-BAF9F2798B9B\_1\_201\_a.HEIC

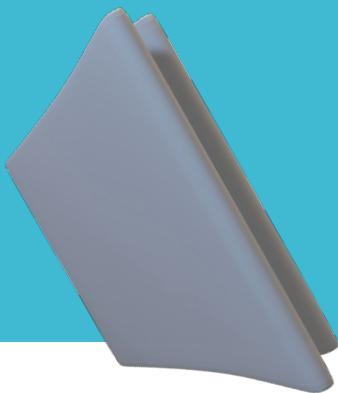
Lux Product 1

Vintage Handbag

2.12

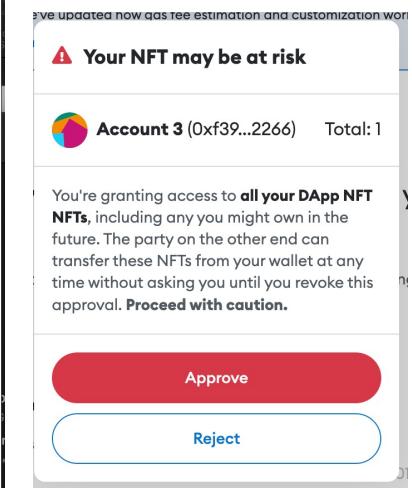
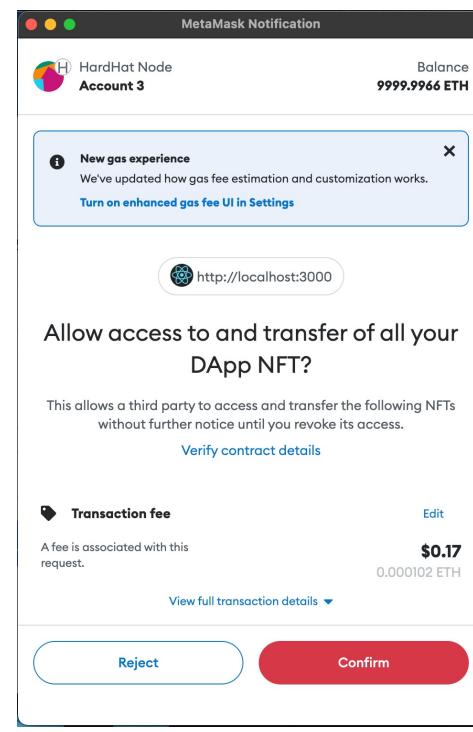
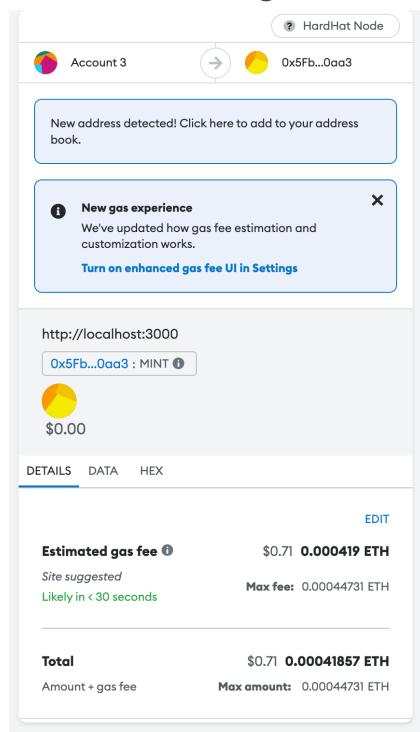
Create & List Luxury NFT!

# Platform - NFT Creation MetaMask Approval



- **Create NFT (2)**

- Confirm Connection with MetaMask and dApp
- Confirm NFT amount
- Confirm gas fees



# Platform - Check NFT

IPFS



- Check uploaded NFTs on 'Home' & 'My Listed Items' tabs
  - Encountered errors due to missing CORS header – troubleshooting in progress
  - Upon research, it was found that CORS error is common with IPFS Infura. It is being resolved by their team as it is caused randomly post project authentication to read and write files to IPFS

The screenshot shows a browser developer tools Network tab with a red border around the error message. The title of the error message is "Ensure CORS response header values are valid". The message states: "A cross-origin resource sharing (CORS) request was blocked because of invalid or missing response headers of the request or the associated preflight request. To fix this issue, ensure the response to the CORS request and/or the associated preflight request are not missing headers and use valid header values. Note that if an opaque response is sufficient, the request's mode can be set to no-cors to fetch the resource with CORS disabled; that way CORS headers are not required but the response content is inaccessible (opaque)." Below this, there is a table titled "AFFECTED RESOURCES" with one row. The table columns are: Request, Status, Preflight Request (if problematic), Header, Problem, and Invalid Value (if available). The single row shows a request with status "blocked", header "Access-Control-Allow-Origin", problem "Missing Header", and no invalid value listed.

Request	Status	Preflight Request (if problematic)	Header	Problem	Invalid Value (if available)
QmUhcAy5mTAEzNLjkHp2gyB4C28gQ7riaExHFmz1dQnF	blocked		Access-Control-Allow-Origin	Missing Header	

[Learn more: Cross-Origin Resource Sharing \(CORS\)](#)

# Platform - Check NFT

IPFS



- **Check IPFS file metadata hash to confirm upload from NFTMarketplace**
  - Currently having issues with reading data back from IPFS - resolution in progress
  - Currently NFT files and metadata can be stored on IPFS using NFTMarketplace. However, CORS header failure occurs when our dApp tries to access IPFS URL from infura

LUX NFT Marketplace

Home

Create

My Listed Items

My Purchases

0x39...2266

Loading...

Audit usage of navigator.userAgent, navigator.appVersion, and navigator.platform

Ensure CORS response header values are valid

A cross-origin resource sharing (CORS) request was blocked because of invalid or missing response headers of the request or the associated preflight request.

To fix this issue, ensure the response to the CORS request and/or the associated preflight request are not missing headers and use valid header values.

Note that if an opaque response is sufficient, the request's mode can be set to no-cors to fetch the resource with CORS disabled; that way CORS headers are not required but the response content is inaccessible (opaque).

4 requests

Request	Status	Preflight Request (if problematic)	Header	Problem	Invalid Value (if available)
QmUhcAy5mTAEnZxNLjkHp2gyB4C28gQ7riaExHFmz1dQnF	blocked		Access-Control-Allow-Origin	Missing Header	
QmUhcAy5mTAEnZxNLjkHp2gyB4C28gQ7riaExHFmz1dQnF	blocked		Access-Control-Allow-Origin	Missing Header	
QmUhcAy5mTAEnZxNLjkHp2gyB4C28gQ7riaExHFmz1dQnF	blocked		Access-Control-Allow-Origin	Missing Header	
QmUhcAy5mTAEnZxNLjkHp2gyB4C28gQ7riaExHFmz1dQnF	blocked		Access-Control-Allow-Origin	Missing Header	

QmZE92G8Rft6W2PXYFm4EdLmbjsMeu6zyeKk3Q998M7qNr 104.75 KB 2023-02-14 at 12:14 PM EST

QmXhnaqMDov2iHhgCJjPWyVzsTuiJgJcvUzL5qVneqbnx 182.00 B 2023-02-13 at 4:17 PM EST

QmUi3jZucCv6ourVMWQkJAVCDQ7HZujQhgGtLU6945YoX 179.00 B 2023-02-14 at 12:14 PM EST

QmUhcAy5mTAEnZxNLjkHp2gyB4C28gQ7riaExHFmz1dQnF 167.00 B 2023-02-17 at 5:26 PM EST

# Insights & 99 Problems



- Insights and Positives:
  - This kind of platform would also be needed in the fair trade sector to track the entire supply chain logistics to monitor unethical labour standards
  - Currently there is not a particular NFT marketplace completely catered towards the authentication of designer items, whereby the process is tracked and added to blockchain from inception. This platform provides a good start
  - Creating a digital solution for actual contracts between users (easier and more accessible than current practices that involve lawyers and more money)
- Problems:
  - Python would have been a better choice for coding this project where there is solid proof of concept
  - Currently the platform is in its bare bones and only one product can be added, this needs expansion
  - IPFS integration was partly a failure as the authentication fails using HTTP-IPFS-client in react

# Conclusions & Next Steps



- Expand capabilities of platform where multiple items are added
- Fix CORS header issues
- Integrate aspect of fair-trade sector to accompany platform as there are currently no platforms doing this specifically nor effectively