Project 3: Lucky_Day_Auction_NFT

Lucky Day NFT Auction House!

Bid For Your Favourite NFT!

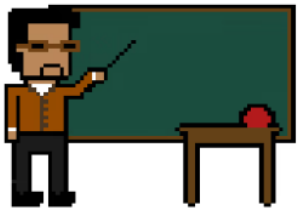


Table of Contents

Description

Our DAPP is a basic auction app. The user is able to connect from their own wallet. In this scenario we are using Ganache. The addresses are the first ten addresses that ahave 100 ETH in each making it able to place the bid and send the transactions with unique attributes corresponding to the NFT. The user can see what image is being sold and will

automatically withdraw from the account of choosing while show a transaction receipt. A secure hash that can be decoded using the [JSON debugger](https://jwt.io/#debugger-io). Here you can review all the details that in English.

Goals

---

1. A platform that connects artists and collectors through blockchain technology with complete transparency. It holds asset/token/deed that is to be auctioned using ERC721 standards.
2. Ability to place bids in auctions, over a decentralized network with following functions and features:

```
    - ability to participate in an English auction whereby bid prices keep increasing
over the duration of the auction.
    - ability to monitor the auction process (start bid, bid price, highest bidder
etc.)
    - ability to view the frequency of each bidder
    - safe and secure transfer of NFT ownership upon auction completion
    - safe and secure transfer of funds upon auction completion
    - refund of funds to bidders that did not get not lucky
```

3. Use of Polygon Network provides the users with a method that is lower in cost and more efficient (faster) as compared to transacting directly over the Ethereum Network (Polygon is essentially a "scaling solution" that aims to address the inefficiences of transacting with the high demand ethereum network - it groups multiple transactions into a single block before comitting to the main ethereum network)
4. Works with digital assests stored over an established and secure file storage system (IPFS - Pinata)
5. Lucky Day does not charge any fees or retain any of the profits from the NFT sales hence providing a free of cost platform for the artists. As opposed to OpenSea, who charge a chunky one-time registration fee to list each NFT as well as recurring fees.

Data Collection and Preparation

In order to test and demo our application we need to have an inventory of digital artwork. We created some custom ones using Photoshop. In order to generate the IPFS links for the custom artwork, we utilized Pinata.

Development and Technologies

Our NFT marketplace is build using the following technologies:

- Solidity (smart contracts)
- Remix IDE
- Streamlit (frontend)
- MetaMask (wallet)
- Decentralized Blockchain Network (Polygon TestNet/Ganache)
- Xbox GameBar/Quicktime Player (Demo Video)
- ChainLink (new technology/library - not covered in class)
- Pinata
- Photoshop
- Python

Libraries Used

- os
- json
- requests
- eth_account
- eth_typing
- web3
- pathlib
- dotenv
- streamlit
- dataclasses
- typing
- chainlink (AggregatorV3Interface)

- openzeppelin (ERC721, ERC721URIStorage, Ownable, Counters)

Instructions - Environment Preparation

Files:

Download the following file to help you get started:

1. Auction.sol

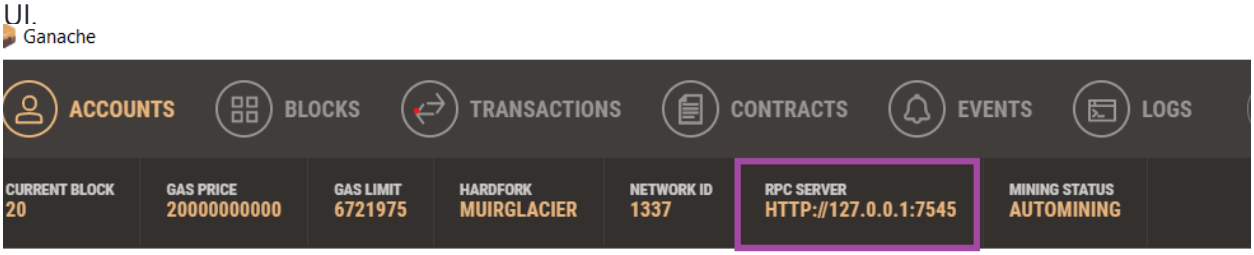
Add Polygon Mumbai Testnet to MetaMask steps:

1. Open MetaMask and select `Settings`
2. Select `Networks`
3. Select `Add Network`
4. Enter Network Name `Matic-Mumbai`
5. Enter New RPC URL `https://rpc-mumbai.maticvigil.com/`
6. Enter Chain ID `80001`
7. Enter Currency Symbol `MATIC`
8. Enter Block Explorer URL `https://mumbai.polygonscan.com/`
9. Add MATIC to accounts via https://faucet.polygon.technology/

Obtain RPC Server Address

1. Option 1 - Polygon Mumbai Test - Intended Project Blockchain - Create account with https://rpc.maticvigil.com/ and create dapp RPC link for Mumbai Testnet.

2. Option 2 - Ganache - Backup Project Blockchain - Simply copy RPC Server from Ganache UI.



Load Keys In .env File

1. Load `PINATA_API_KEY` and `PINATA_SECRET_API_KEY` to .env file for IPFS Hashing and Storage
2. Load `WEB3_PROVIDER_URI` with RPC Server address.
3. Load `SMART_CONTRACT_ADDRESS` according to streamlit dapp. Auction dapp requires `auction.sol` contract address when deployed from Remix.
4. Load wallet's `MNEMONIC` seed phrase.

Remix Steps:

To run the application, clone the code from this GitHub repository.

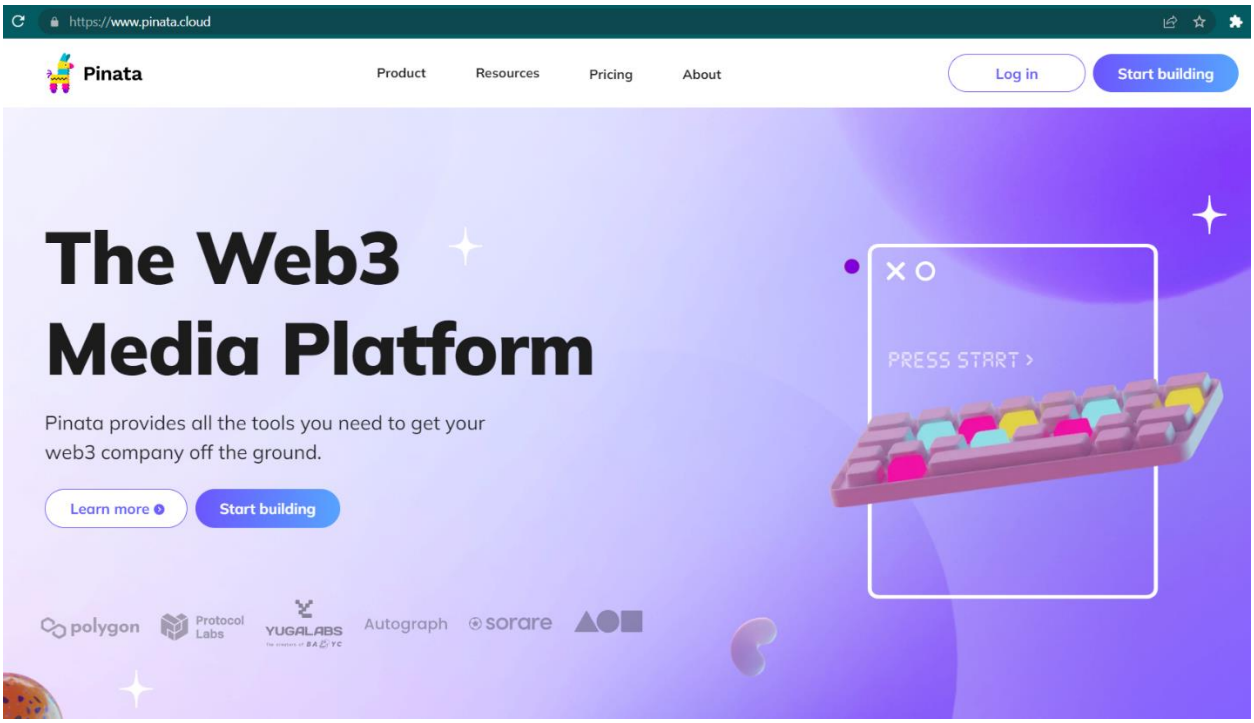
1. Compile the `auction.sol` to ensure it compiles without any errors.

2. Prior to deployment, ensure your MetaMask/wallet is connected and the corresponding item (Injected Web3 for Remix IDE) is selected.

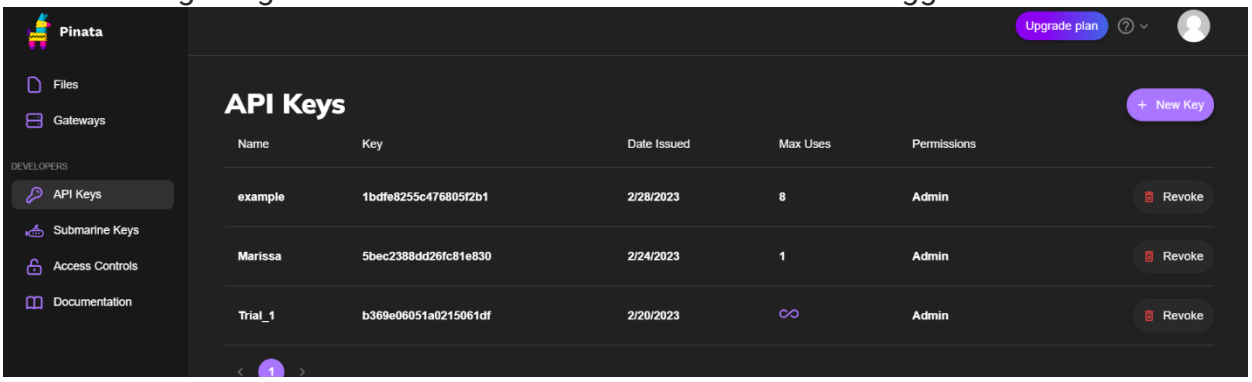
3. Deploy the `auction.sol` and check the deployed contracts to ensure it is there. Copy the address as it would be required for the next step.
4. Add the `auction.sol` contract address to the Deploy the AuctionRegistry.sol and proceed to deploy the AuctionRegistry.sol
5. In `AuctionRegistry.sol` deployed contract, use the address of the Auction contract in the SetApprovalForAll input field and a value of true to ensure the NFT that will be registered can participate in the Auction.
6. In `AuctionRegistry.sol` deployed contract, use the registerNFT fields to provide an address ownner and key NFT details and register it for the Auction.

7. To proceed with the auction process on the registered NFT, please follow the steps demonstrated in the Auction Demo (see Videos Demos section).

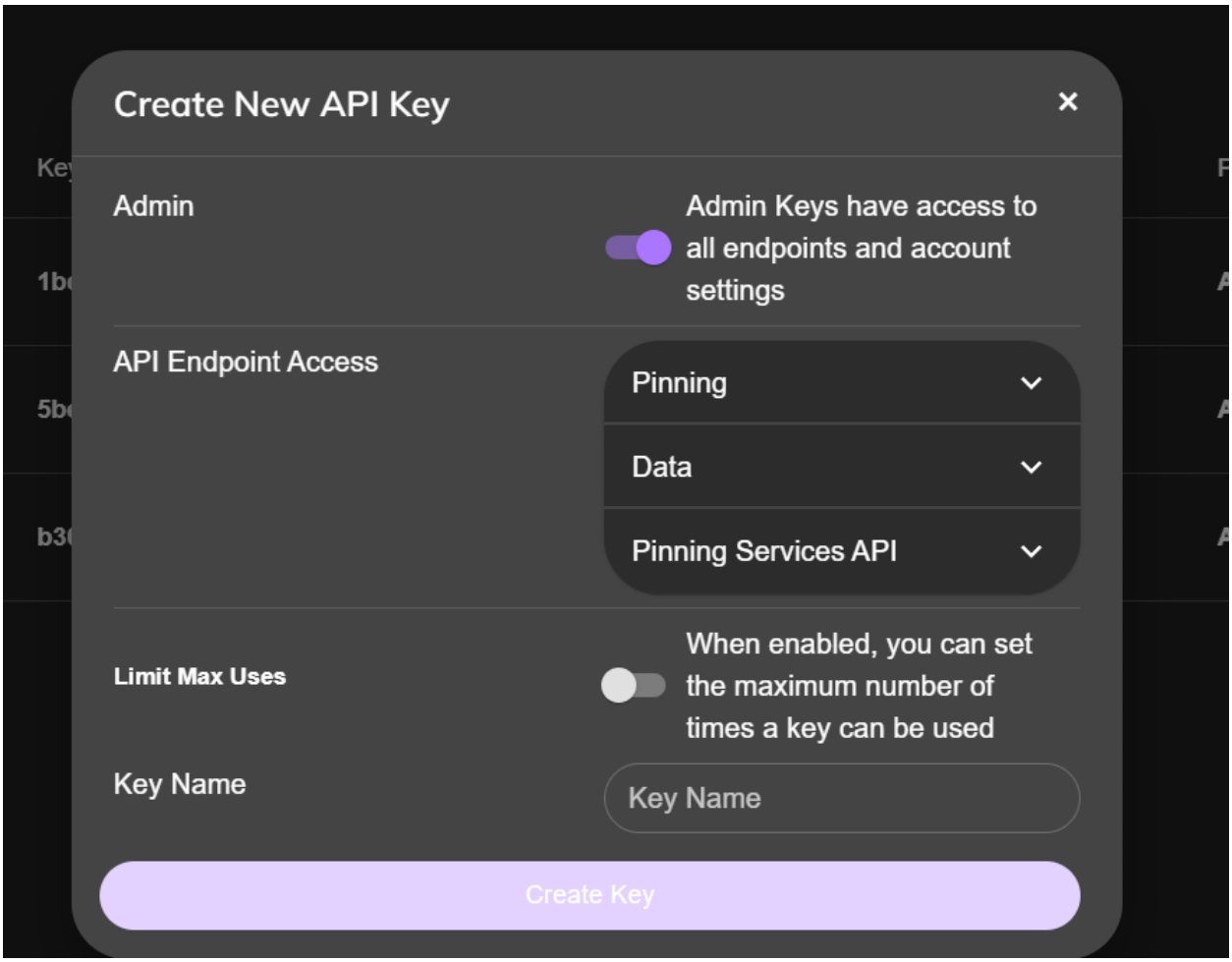
Pinata Steps:

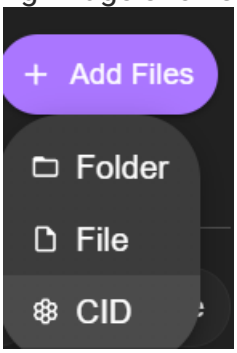
1. To begin, we go to the [Pinata website](), as the following image shows:
2. 

:
3. We then click the Login button and log in to the Pinata dashboard. Note: If signing up for the first time you will be greated with two keys and a JWT, copy and paste into the env file
4. The following image shows the Pinata dashboard once we've logged in:

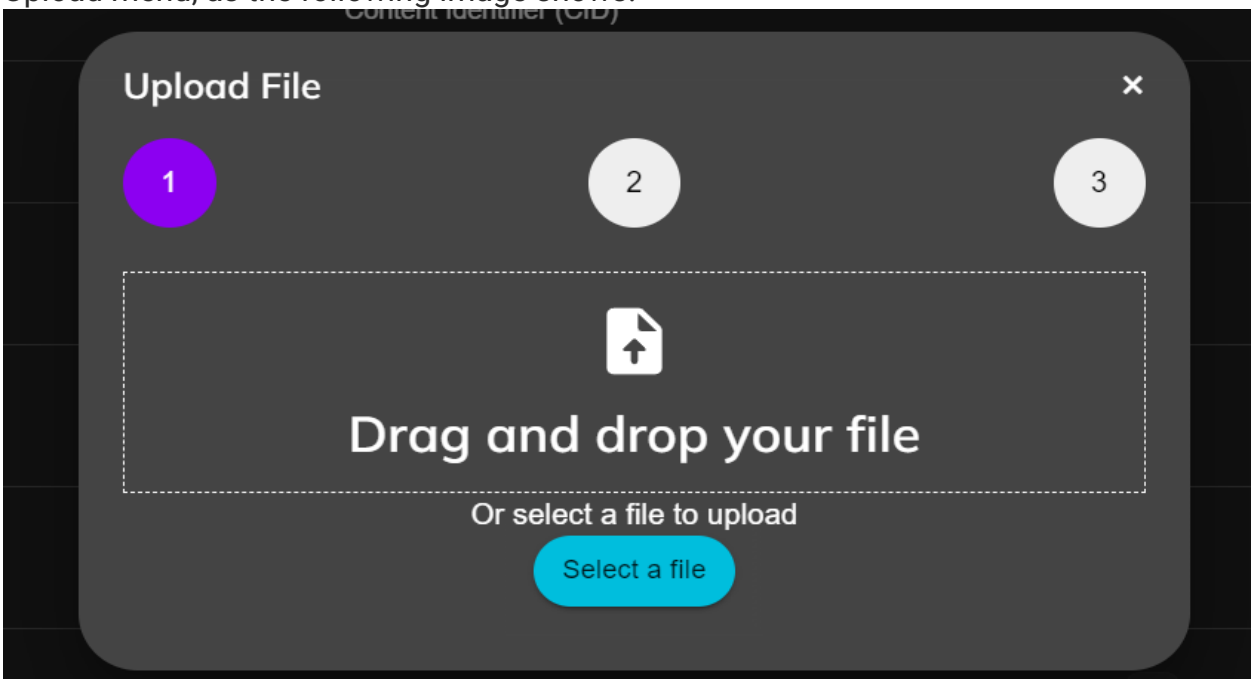


5. Create an new API key, Click on “New Key”.
6. Turn on all features except the Limit Max Uses button as it limits the tracing to the image we would like to upload later.

7. 

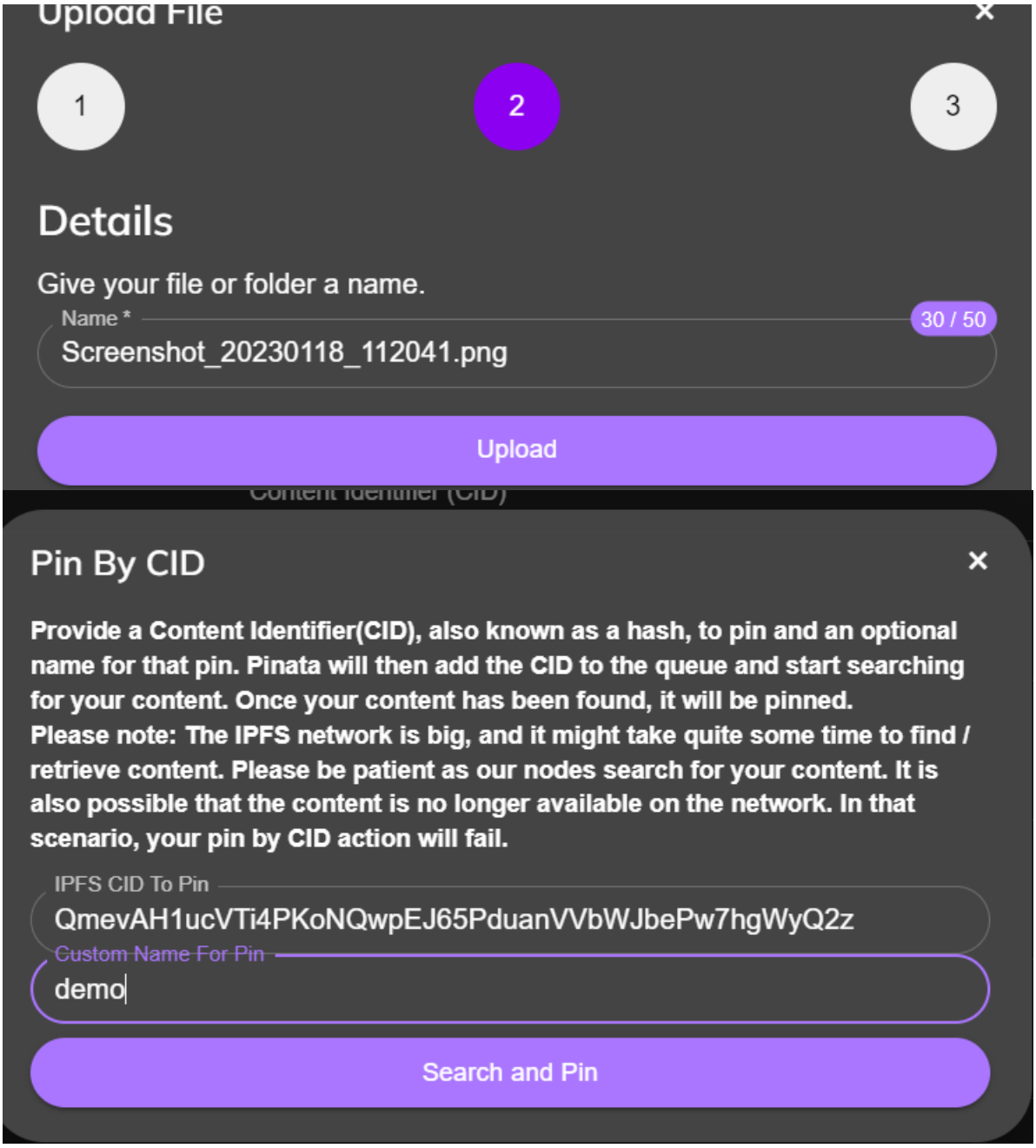
8. Enter a special name for the environment you would like to use to pin the images.
9. Now click on the files tab and once the page is loaded you should see purple box similar to the one from step 4.
10. The Upload menu expands, listing the Folder, File, and CID menu items, as the following image shows:
11. 

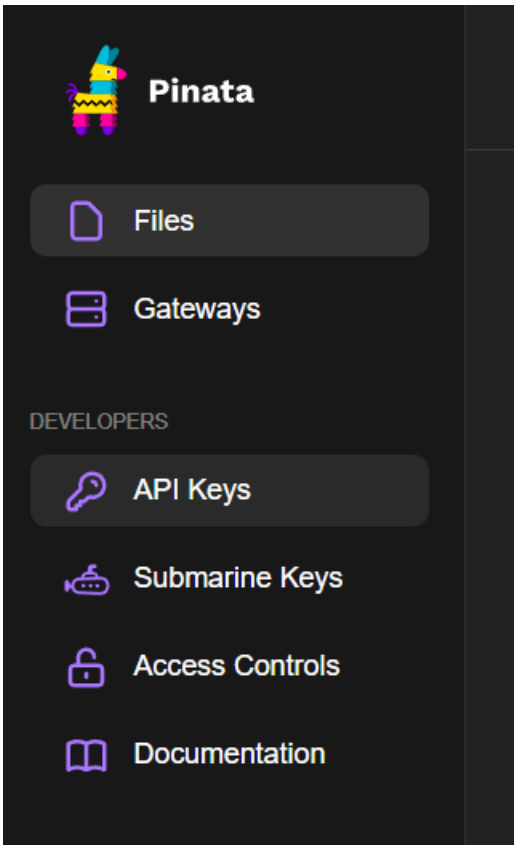
12. We'll now upload this file to Pinata. First, on the files page, we click the Upload menu, as the following image shows:



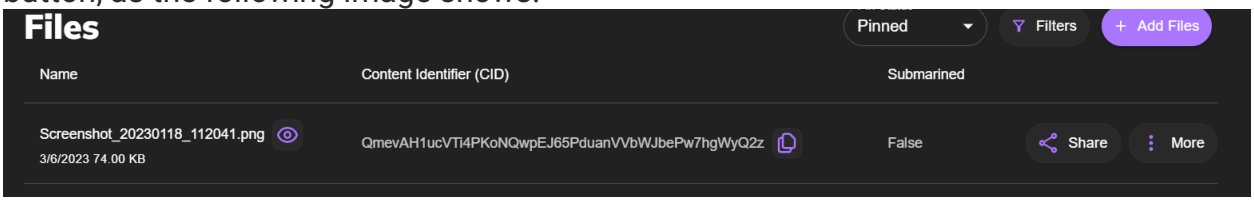
13. We can now click Folder or File to navigate to an entire folder or a single file, respectively, to upload from our local computer. Or, we can click CID to provide a hash for a file that's already pinned to IPFS.

14. In the Upload File dialog box, we click "Click to upload." Our browser then presents us with the opportunity to navigate to and select the file on our computer that we want to upload.

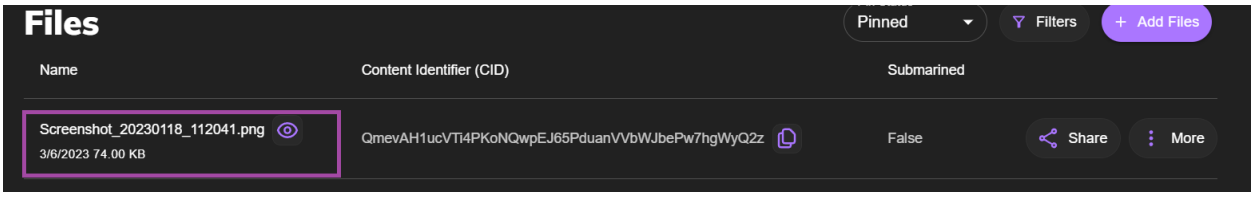

Upload File
1
2
3
Details
Give your file or folder a name.
Name *
30 / 50
Screenshot_20230118_112041.png
Upload
Pin By CID
Provide a Content Identifier(CID), also known as a hash, to pin and an optional name for that pin. Pinata will then add the CID to the queue and start searching for your content. Once your content has been found, it will be pinned.
Please note: The IPFS network is big, and it might take quite some time to find / retrieve content. Please be patient as our nodes search for your content. It is also possible that the content is no longer available on the network. In that scenario, your pin by CID action will fail.
IPFS CID To Pin
QmevAH1ucVTi4PKoNQwpEJ65PduanVVbWJbePw7hgWyQ2z
Custom Name For Pin
demo
Search and Pin



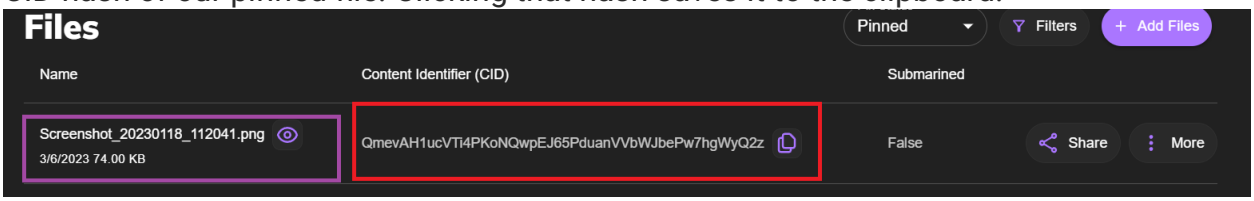
15. Next, in the Custom Name For Pin box, we type a descriptive name for our pinned file that will display later in the Pinata dashboard. Finally, we click the Upload button, as the following image shows:



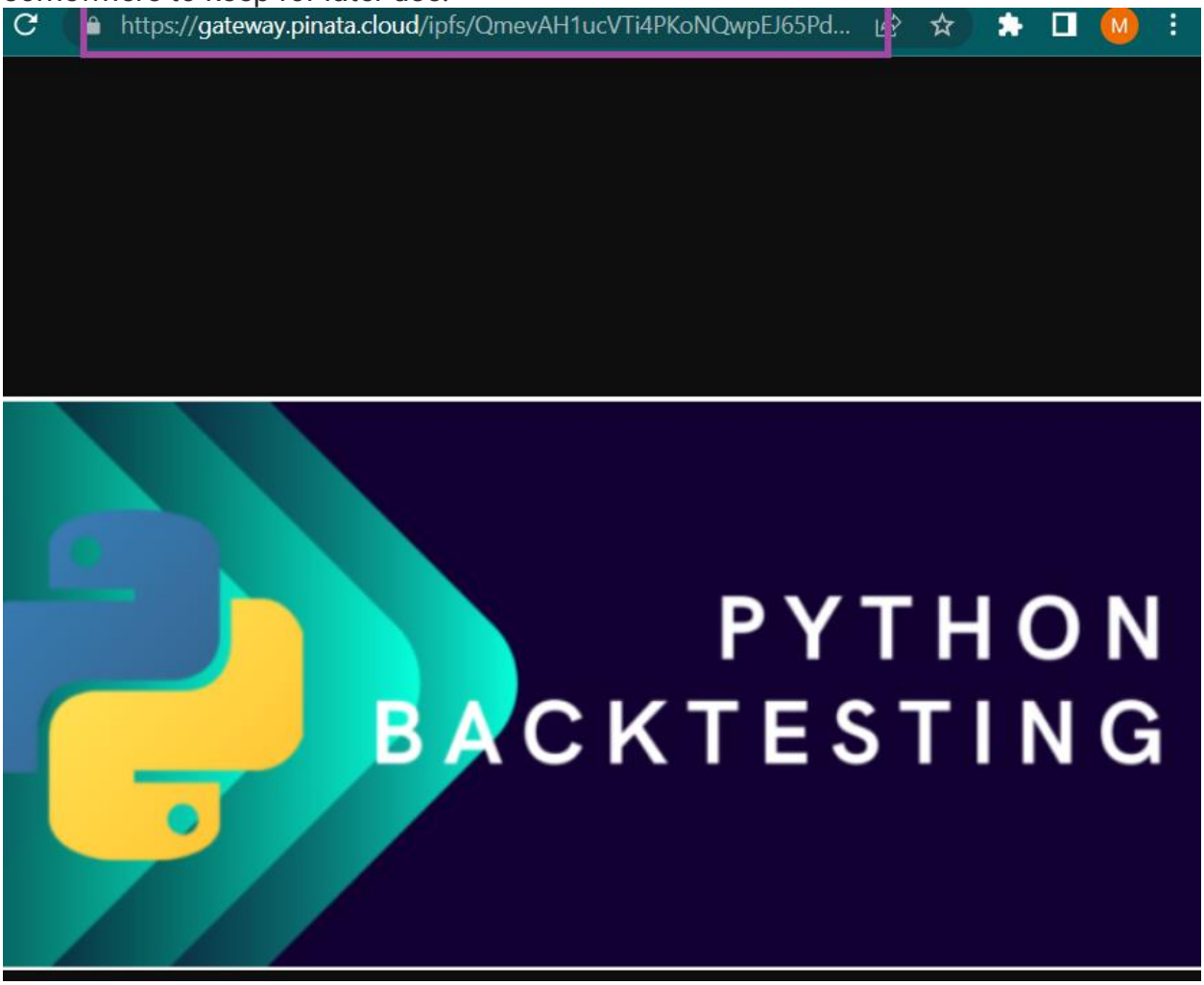
16. Notice that the name of our pinned file is a clickable link, as the following image shows:



17. Note that in the preceding image, the IPFS CID column displays the IPFS CID hash of our pinned file. Clicking that hash saves it to the clipboard.

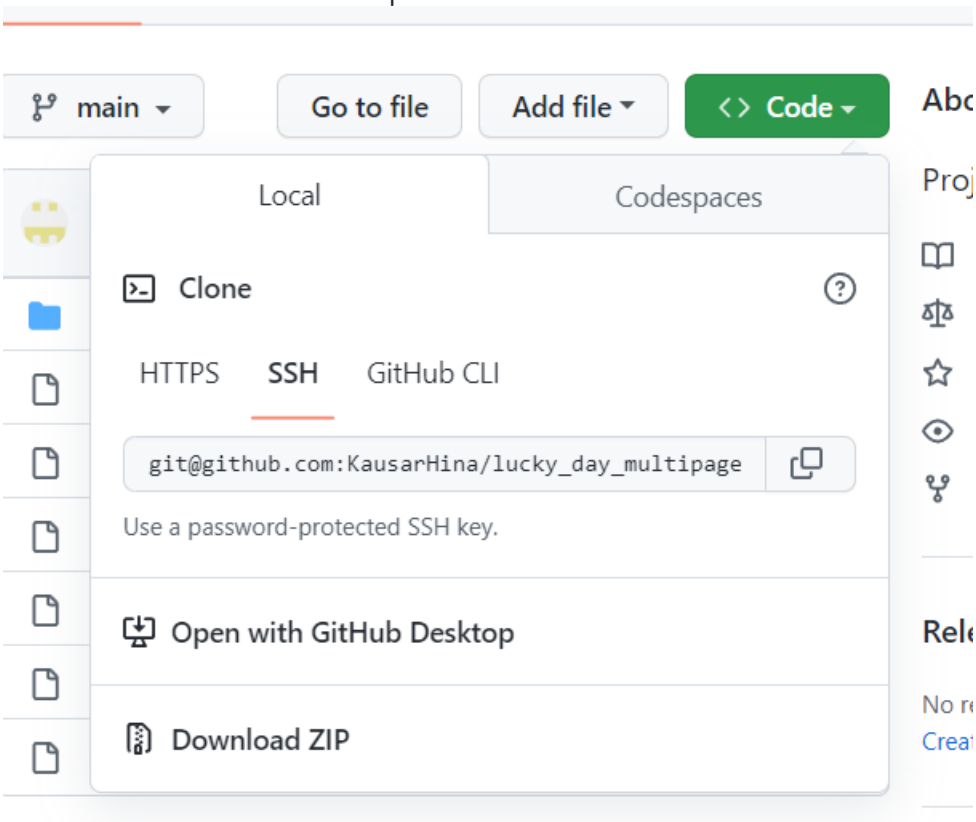


18. . In the preceding image, notice that to display the file in our web browser, the Pinata gateway created a standard web URL. This URL is always gateway.pinata.cloud/ipfs/ followed by the hash and we can then paste it somewhere to keep for later use.



Streamlit dapp:

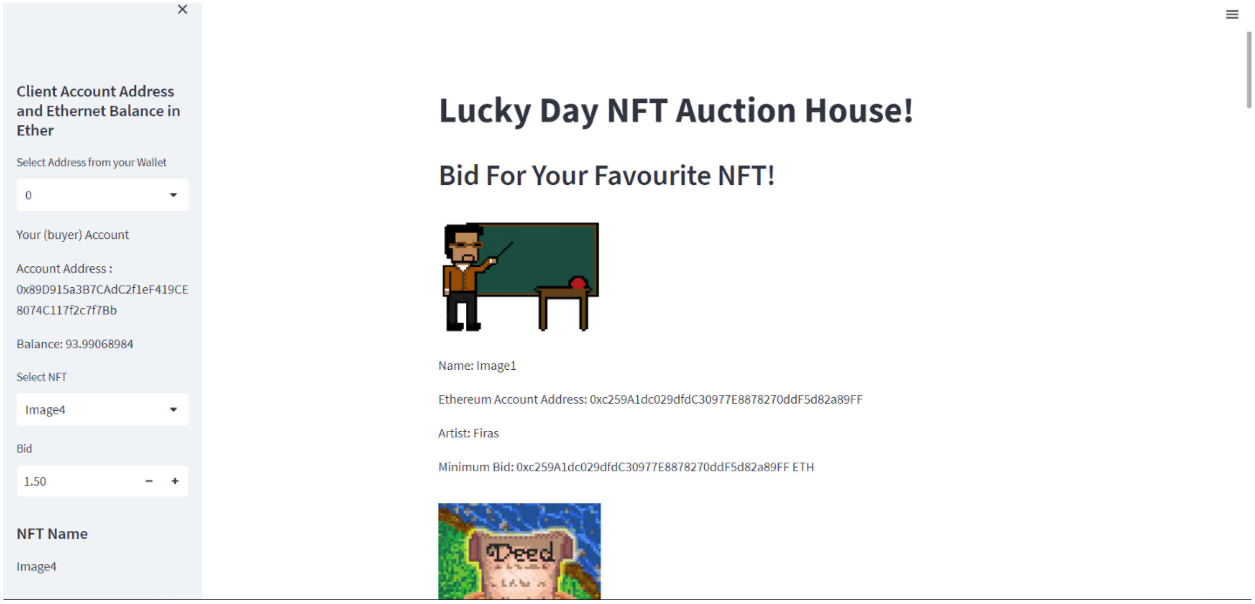
1. Copy deployed `Auction.sol` contract address to SMART_CONTRACT_ADDRESS key in .env file in location of the pages folder underneath the 2_Lucky_Day_Multipage . . To clone the code. Follow the steps bellow:

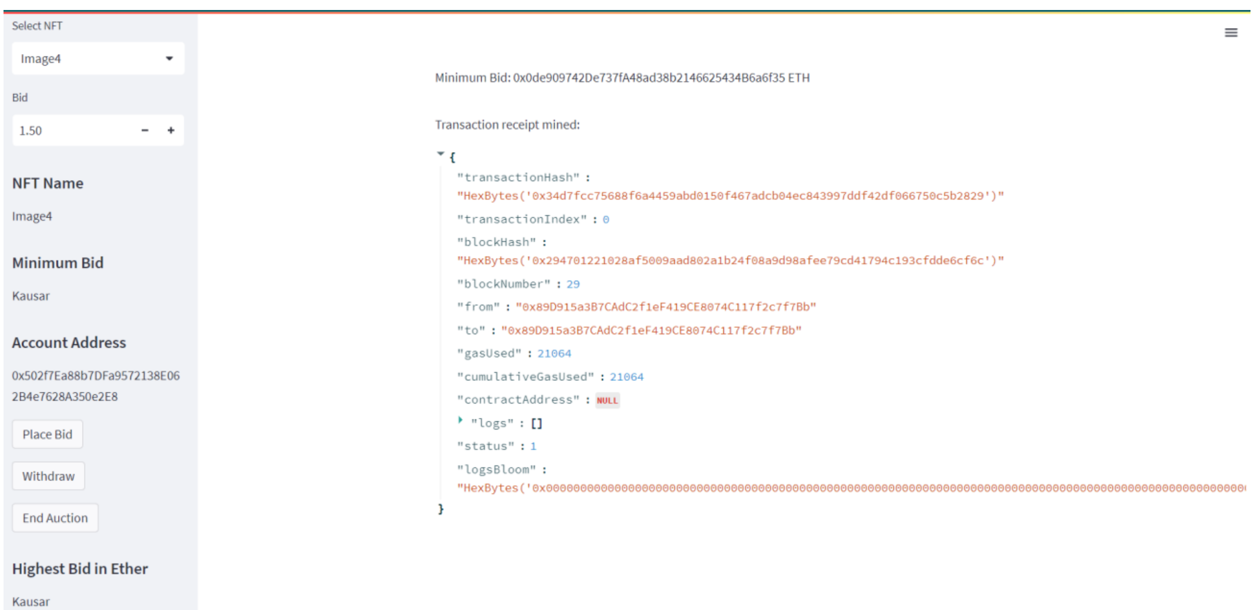


2.
3. Git clone SSH key
4. Click onto the App Folder
5. In the terminal select where that path is and - Make sure the base says something like C:\Users<your name>\Downloads\lucky_day_multipage\ app\ pages\
   - If all the .py files are downloaded, enter streamlit run '2_Lucky_Day_Auction_NFT.py'

Outcome and Summary

Our DAPP is a basic auction app. The user is able to connect from their own wallet. In this scenario we are using Ganache. The addresses are the first ten addresses that ahave 100 ETH in each making it able to place the bid and send the transactions with unique attributes corresponding to the NFT. The user can see what image is being sold and will automatically withdraw from the account of choosing while show a transaction receipt. A secure hash that can be decoded using the [JSON debugger](https://jwt.io/#debugger-io). Here you can review all the details that in English.





Optimization and Debugging Opportunities

1. **Smart Contract Bug** - `End` Function in `auction.sol` contract does not execute when the auction time has run out and thus leaving the auction open without a means to complete the transfer of NFT ownership and the withdrawal of bid funds to their respective accounts.

2. **Streamlit Bugs** - `Bid` function does not execute correctly for `app3.py`, i.e. our auction dapp. `End` function bug still applies.
3. **Inoperable Contract**- Our contract was able to compile itself but, as deploying goes there were many internal errors.

Contributors

Marissa Gonzales

Jodi Artman

Edith Chou

Kausar Hina

References and Resources

NFT Sales
NFT Sale Volumes
Gas-Free NFT IPFS
OpenSea Fees
dAPP Auction
What is Polygon?
NFT Auction
NFT Marketplace
NFT Marketplace
ChainLink Price Feed

OpenZeppelin Contracts Wizard
OpenZeppelin ERC721 Docs