Badger NFT Gallery and Marketplace

With confidence in traditional banking being at a low since the 2008 financial crisis, cryptocurrency value has surged to reflect people's growing distrust in the latter. Bitcoin and Ethereum transactions are at an all-time high, and NFT sales represent a large percentage of that increase. Trading surpassed $2.04 Bn USD in May of this year, a 117% increase from January of this year.

To reflect the growing trend for people investing in decentralized assets, we have developed an NFT Gallery and Marketplace web app for our final project.

Project Setup

Our project is on GitHub. Clone the repository and run ***npm install*** in the backend (/**backend*)*** and frontend **(/frontend)** folders to install all package dependencies required.

Install IPFS Desktop, and open it. Go to **Settings** > **IPFS config** (Scroll to the bottom) and add `"Access-Control-Allow-Origin": ["*"]` inside `"HTTPHeaders"` in line 3. Restart the IPFS daemon.

Put the **.env.local** file in the frontend folder and the **.env** file in the backend folder, and set your IPFS gateway variable to the one in your IPFS daemon (**Status > Advanced**)

Then, navigate to the backend folder in your terminal and run ***npx hardhat compile*** to compile the backend and generate the JSON for the smart contract.

Navigate to the frontend folder and run ***npm run dev*** to start a local server in localhost, and open a web browser. You can now use the application!

Architecture

Our Web app is mainly powered by Alchemy, a web3 development platform that provides APIs for developing dapps (transacting, analytics, network APIs, etc.). Think of them as Amazon Web Services for Web3. Notable clients include OpenSea, GameStop, Meta, and AAVE.

Alchemy is mainly powered by its supernode architecture, which means that users interact with dedicated infrastructure for each type of node functionality (.i.e each service, like analytics, or transaction executions, etc.) are separated. Moreover, requests are processed through a proprietary coordinator service which ensures that nodes are always coordinated and hence have the same state. This allows alchemy to achieve high correctness and scalability at the same time. [1] This is similar to the architecture used by AWS for their web2 services.

For our own project, we make extensive use of Alchemy's NFT API, which provides us with the analytical tools we needed to build our NFT Gallery. The API reduces things like getting the NFTs in a particular wallet to a GET request, which returns an array of the owned NFTs along with their associated metadata.

Our Smart contract itself is built with Solidity and uses the OpenZepplin implementation of ERC-721 with the ERC721Mintable interface extension for supporting minting new tokens and ERC721URIStorage to store token metadata to IPFS links.

The contract itself is designed so that once minted, the NFT belongs to the owner, who has to list it on the marketplace (transferring ownership to the contract address) in order to sell it.

For minting a token, we use the createToken function, which auto increments token Ids and calls ERC721Mintable _safeMint function. The market owner pays for the gas fees of minting a token.

(Smart contract address: 0xfd0CaF2575dFA7a5d1c6ae0c52F67e7ff2111e30)

For listing a token, we call the listToken function. The user must pay 0.1 ETH as a listing fee. This transfers ownership of the token to the marketplace.

For buying a token, we call the executeSale function. The user pays for the price of the token. This function transfers ownership of the token from the marketplace to the buyer, and pays the seller the price of the token and the owner of the contract this listing fee.

[1] Introducing Alchemy Supernode: The Industry Leading Ethereum API

Every time the above 2 functions are called, a new IPFS URL is sent because the following metadata changes for every exchange of hands.

We use IPFS to store NFT metadata. The important metadata stored tokenId, user owner of the token (does not change if transferred to the market), listing boolean (which changes if the token is listed and transferred to the market or not), and price (which changes if the token is relisted).

IPFS (Inter Planetary File System), is a peer-to-peer file-sharing service that distributes content to users in a manner similar to services such as BitTorrent. This is a natural choice for an NFT database, given its decentralized nature.

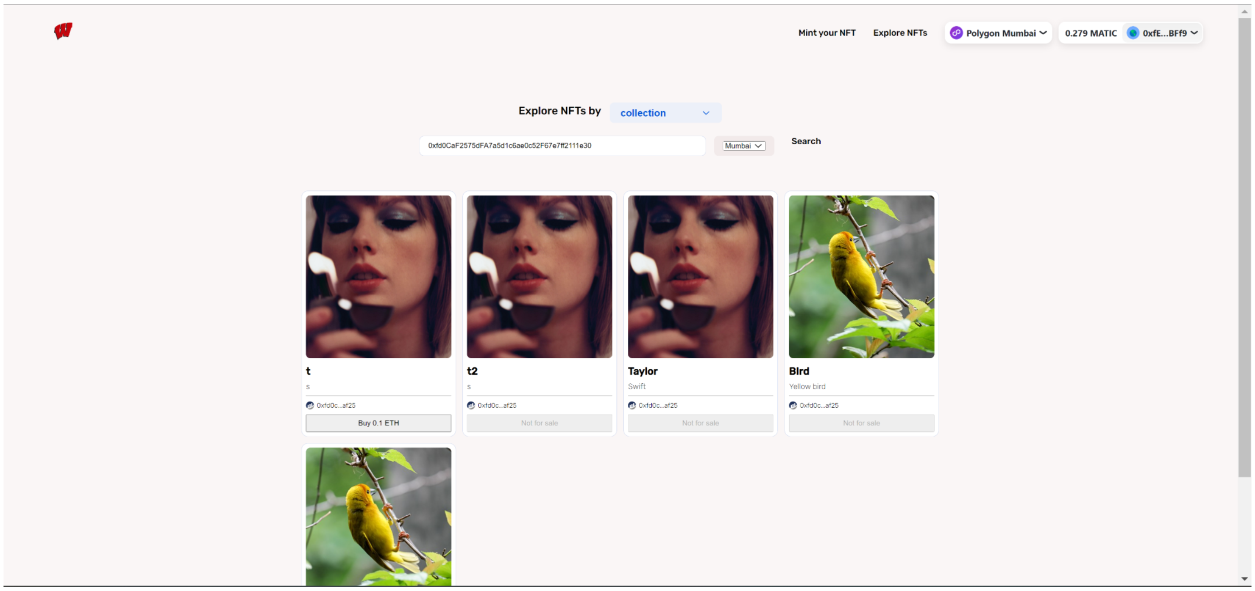
IPFS does have its limitations due to the low bandwidth and the fact that you have to run an IPFS node locally in order to use our application. Some solutions are using a service like Pinata, which allows you to interact with a faster IPFS node. However, we felt that relying on a third party for our database partially defeats the purpose of decentralization, so we decided to use IPFS directly.

Our Frontend is built with NextJS and React, which ensures server-side rendering.

Functionality

The app is designed to be an NFT explorer and marketplace.

The NFT Explorer aspect of the app allows you to view the NFTs associated with any particular NFT collection (by way of its contract address) or individual wallet in the Ethereum Mainnet or Polygon Mumbai testnet. Simply paste the address and click search to view all of the NFTs (along with any associated metadata). This also includes a link to the block explorer for the respective chain. We only support two chains in our current implementation since each Alchemy API key has to be associated with a particular chain (specified at the time of generating the key). We hope to add more chains in future versions (which will not be difficult since all we need to do is ask Alchemy for more API keys).



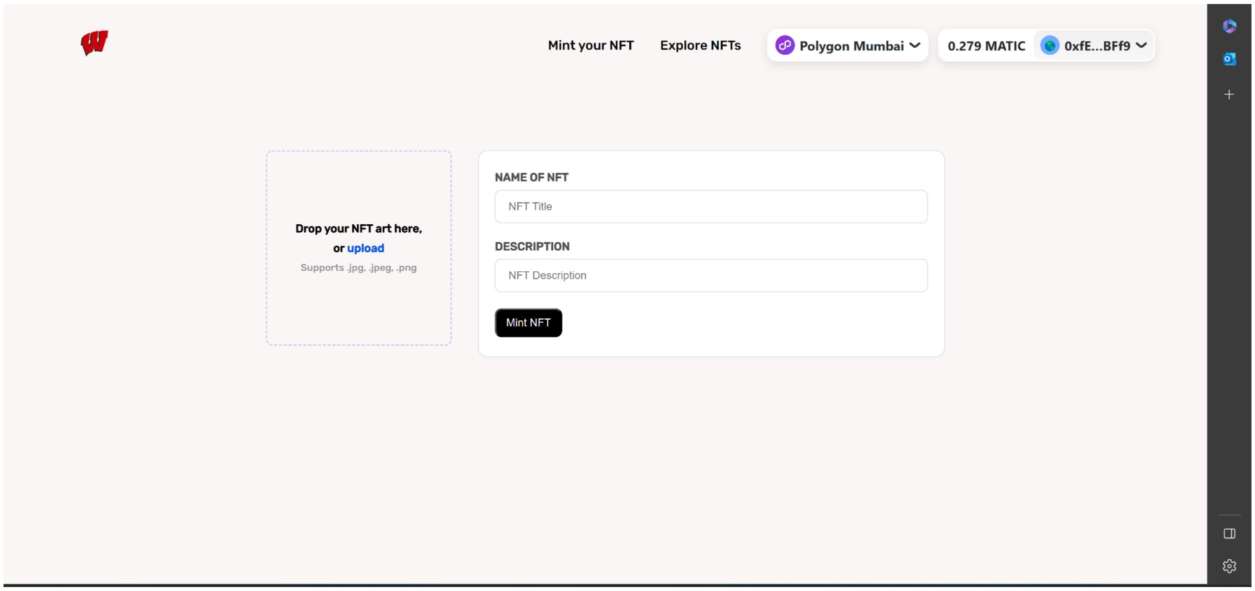
Explore NFTs by collection

0xfd0CaF2575dFA7a5d1c6ae0c52F67e7ff2111e30

Mumbai

We also have our own Smart Contract associated with the NFT, which is deployed on the Polygon Mumbai Testnet. This allows you to create NFT tokens associated with our contract and trade them with other account holders on the Mumbai testnet. This functionality is integrated into our explorer page so that if the NFT is associated with our contract, a button to buy or list the NFT will render on the card (depending on if it is listed or not). If you are the owner of the NFT, then there will be an option to list the NFT on the marketplace (transferring ownership from yourself to the marketplace). Note**: It takes 5-10 min to list the**

NFT on the marketplace due to the bandwidth limitations of IPFS**.**



You can also mint your own NFTs on our contract with the provided interface (as shown above). Simply upload your picture, title the NFT, and describe it.