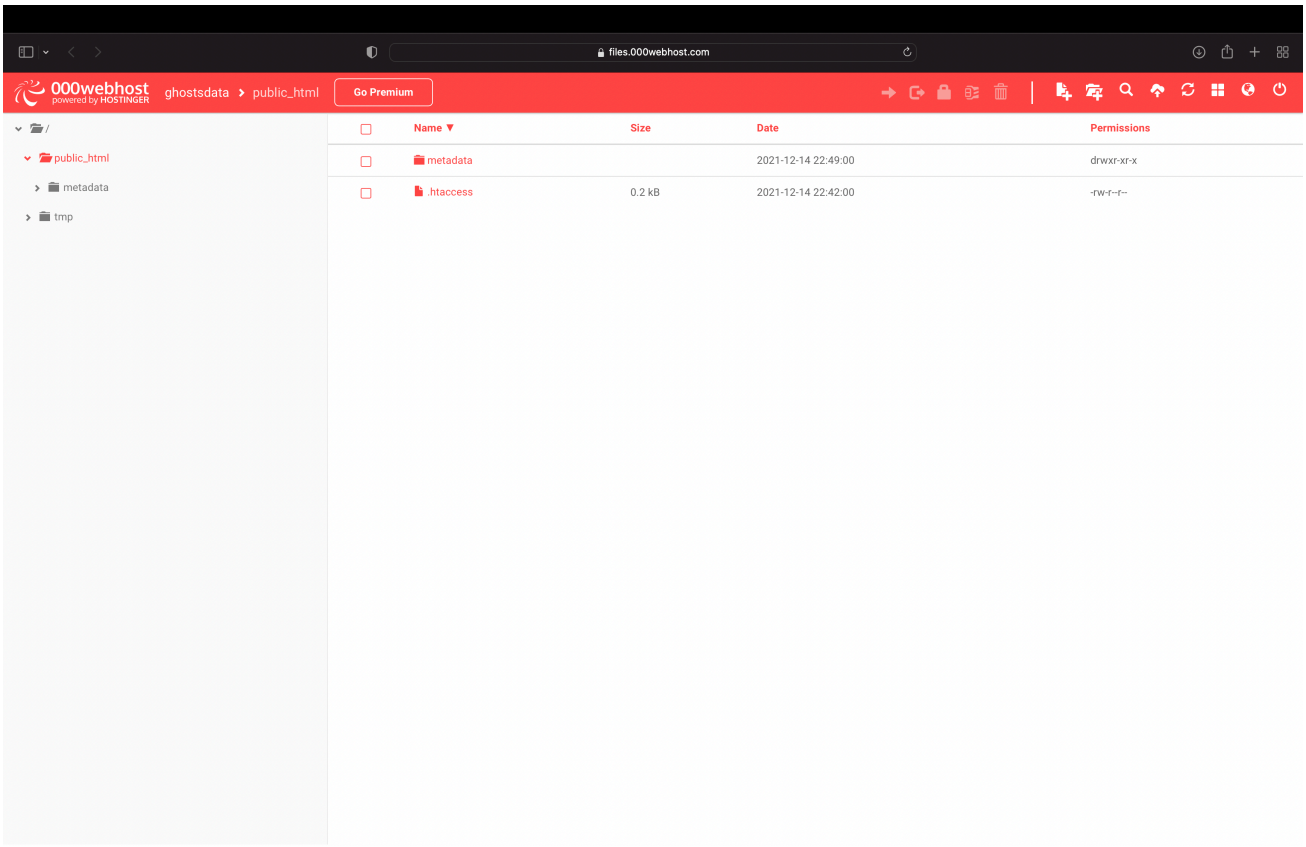
DOCUMENTATION FOR

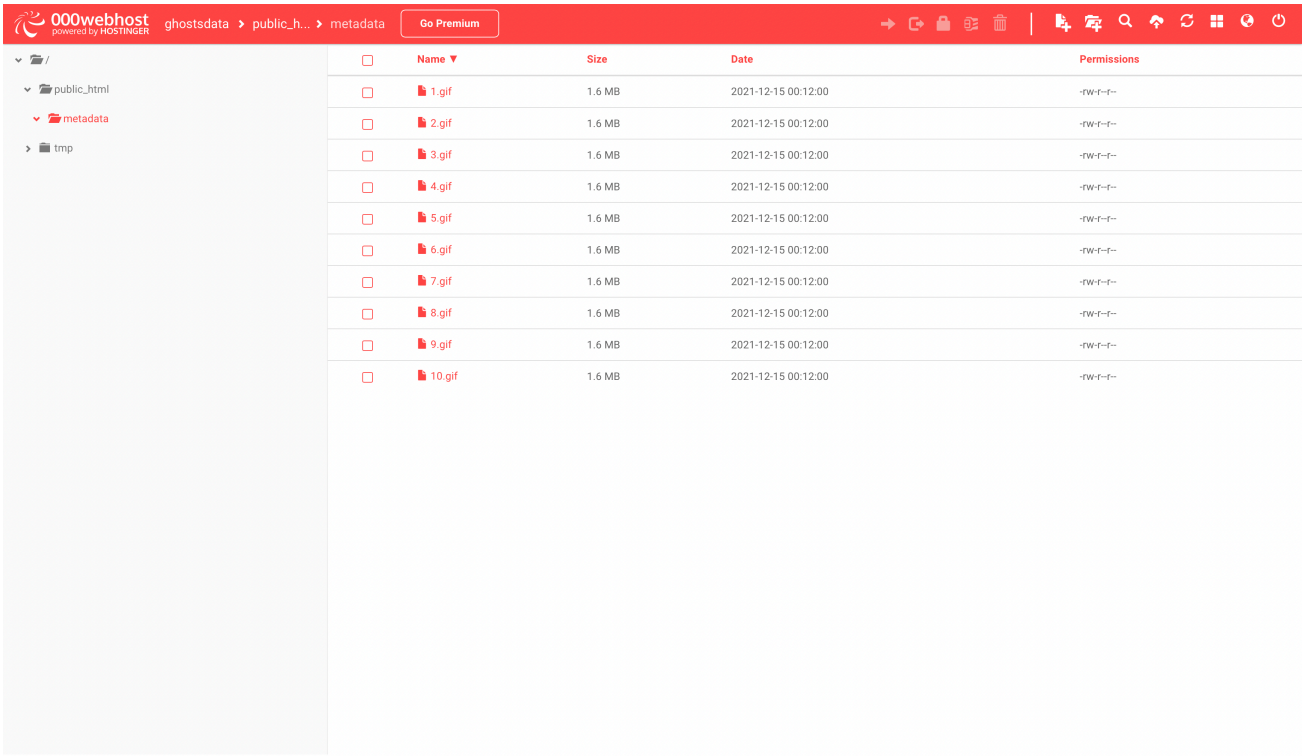
APE CLUB NFT DROP

PART ONE

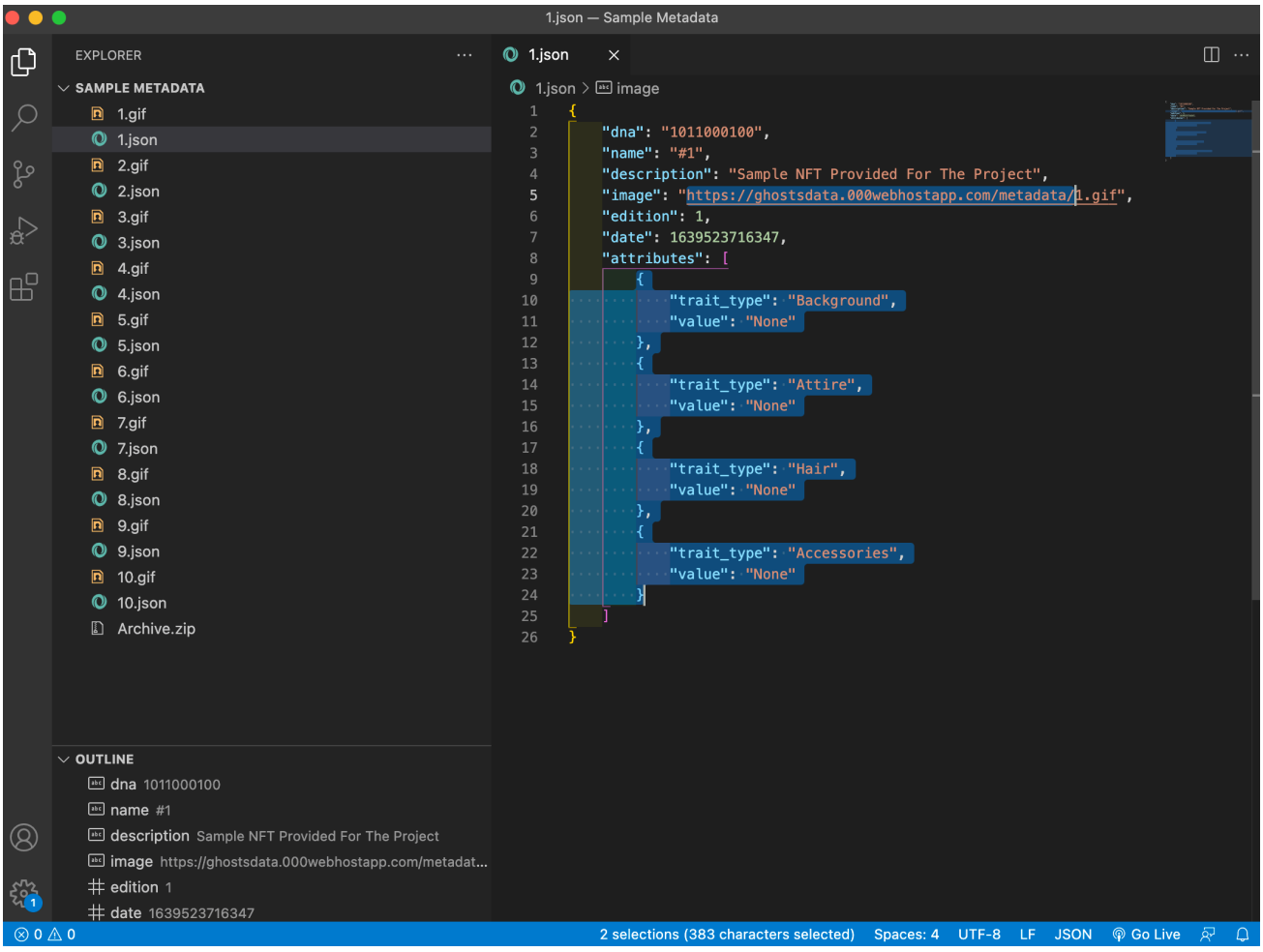
1.) Create a ‘metadata’ folder wherever you plan on hosting your data. It can either be a normal server with secure socket layer or IPFS. A normal server in this case



2.) Upload your collection images into this ‘metadata’ folder. I have provided ten example images that I will be using here. You can find them in ‘Sample Metadata’



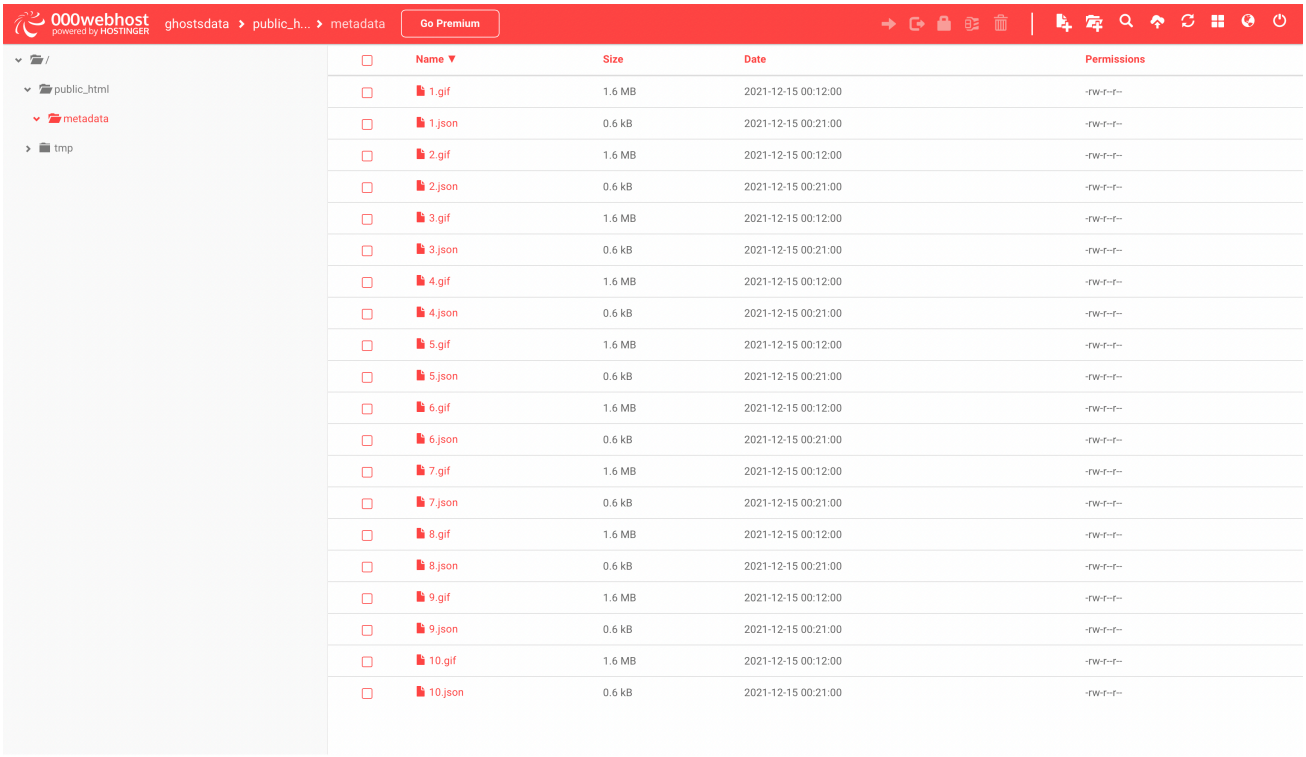
3.) Now pop open the ‘Sample Metadata’ folder in your favourite editor. In this case I’m using VS code.



4.) Open up every .json and replace the image url on line 5 with your url as can be seen above.

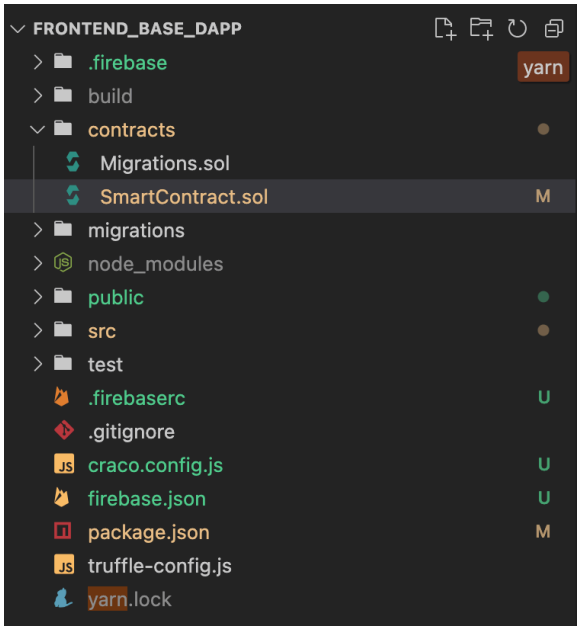
You can then modify the traits from line 8 to 24.

5.) Now upload the new '.json' files to the same metadata folder where your images are stored. Your metadata folder should look something like this :

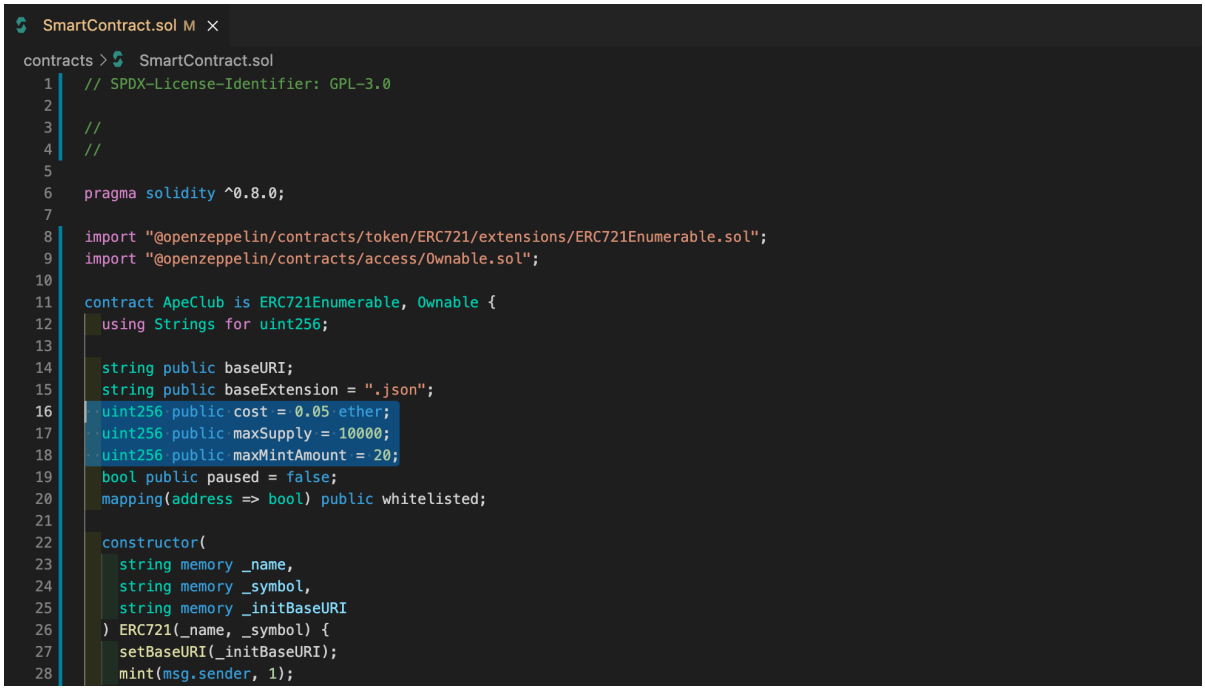


PART TWO

1.) Time to move onto the smart contract! So if you unzip the project and pop into the ‘frontend_base_dapp’ directory , you will find a smart contracts folder. In this folder is the smart contract for your drop , its called ‘SmartContract.sol’

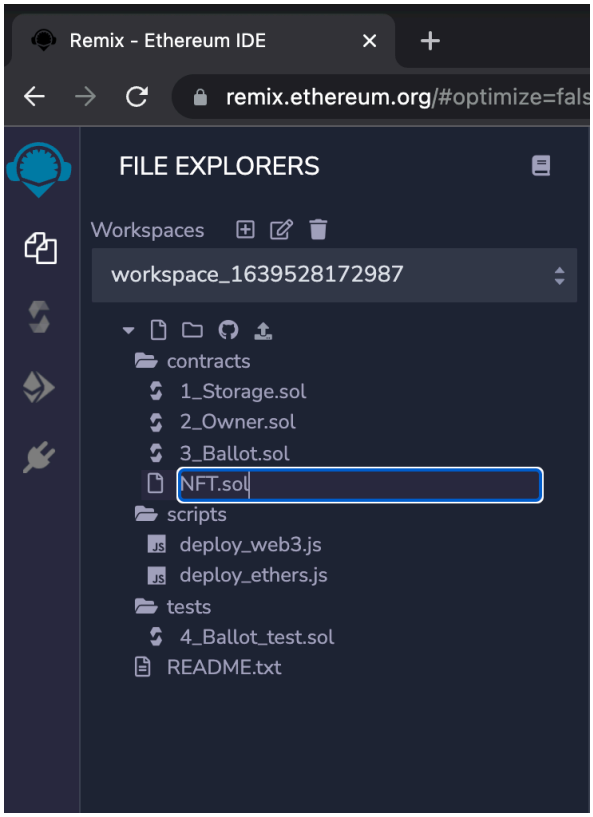


2.) If you open up the ‘SmartContract.sol’ file you will see the cost (amount charged per item for the mint) on line 16 , the max supply (total amount of NFTs to be minted) on line 17 and the max mint amount (the max amount someone can mint at for their wallet) on line 18.

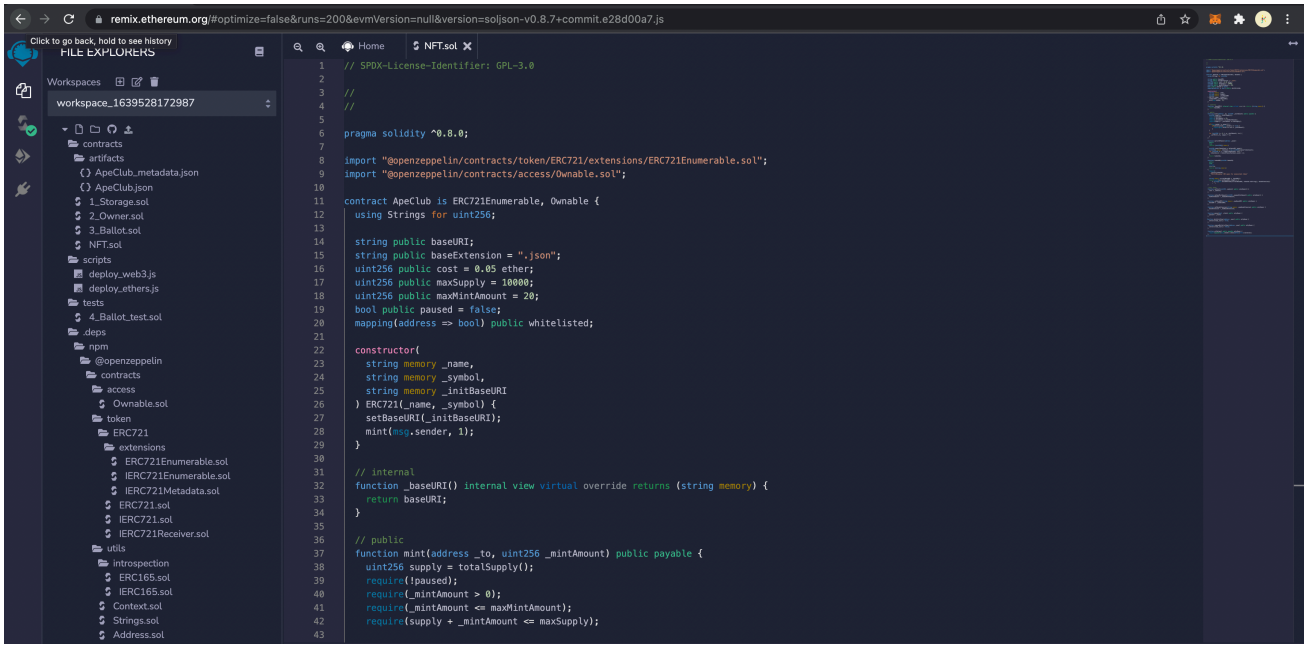


3.) Make the changes you want to and then copy the entire file and head over to https://remix.ethereum.org/

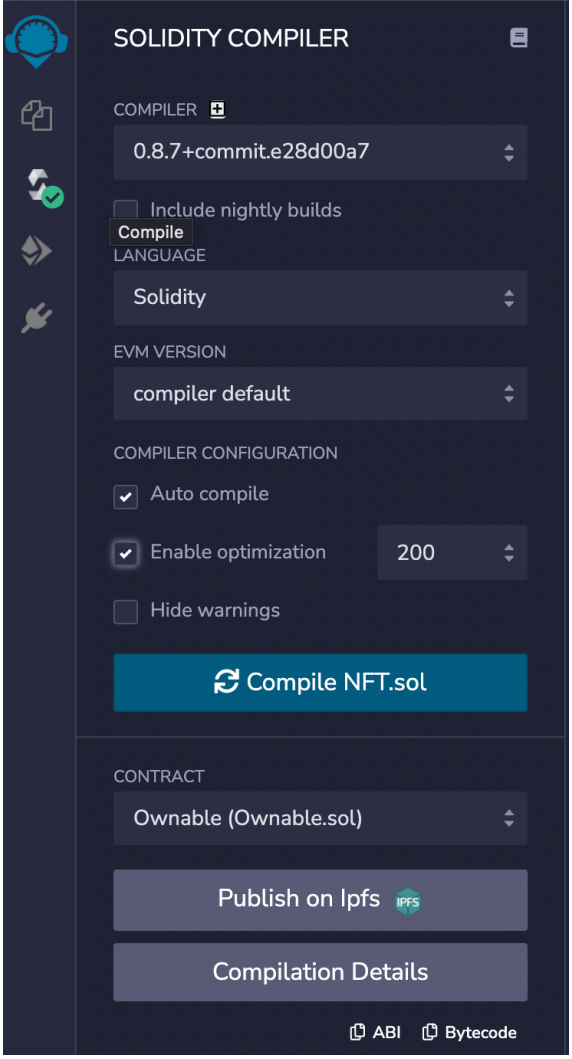
4.) Right click on the ‘contracts’ folder and create a new file and name it ‘NFT.sol’



5.) Paste the contents of ‘SmartContract.sol’ into the ‘NFT.sol’ file in Remix like so :

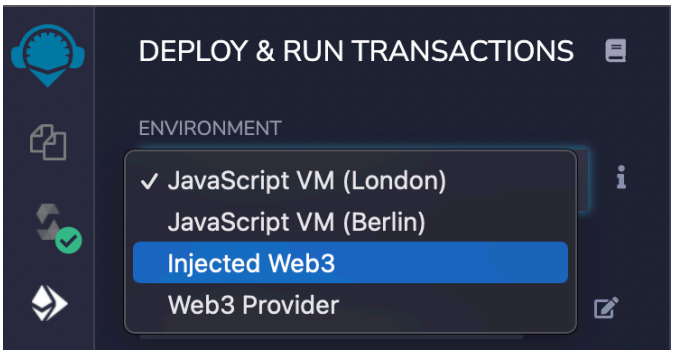


6.) Click on the second tab to the left of the Remix IDE. The Compiler tab. Check auto compile and enable optimisation. Then press the large ‘Compile NFT.sol’ button

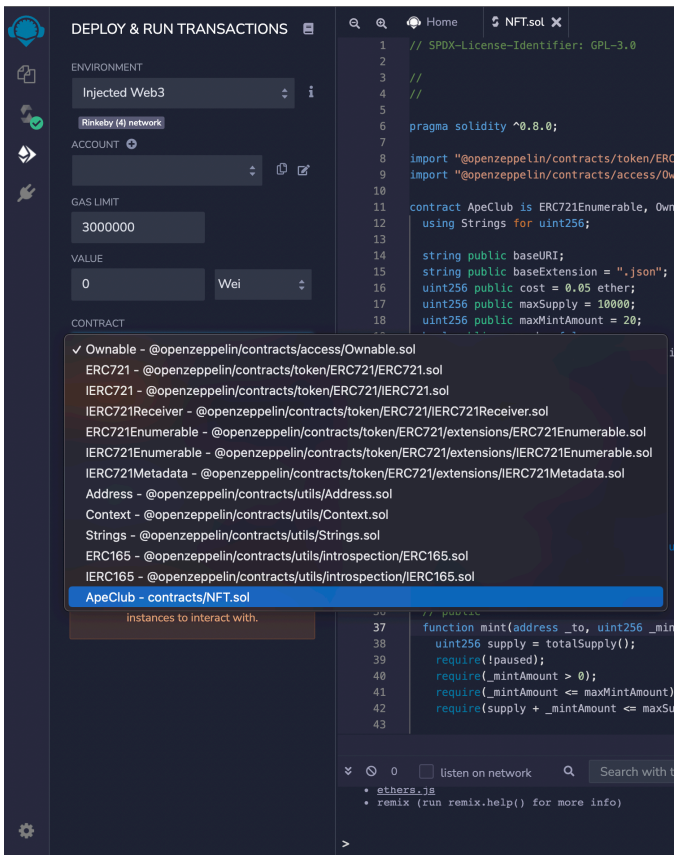


7.) Now hop over to the third tab to the left of the IDE. The Deploy & Run transactions tab.

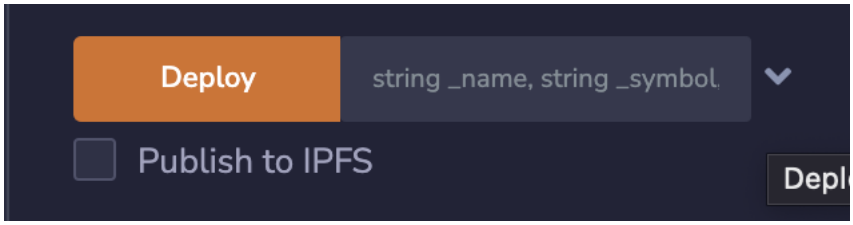
7.1) Under Environment , select ‘injected web3’ and connect MetaMask

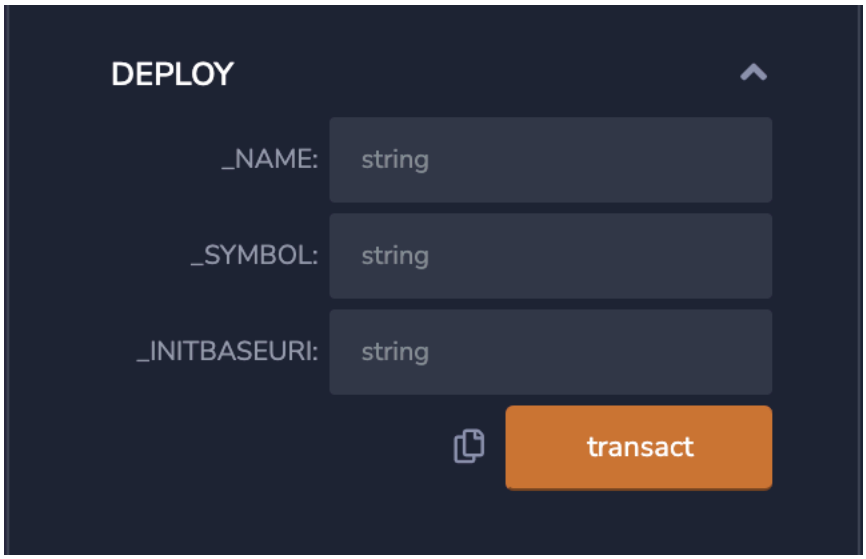


7.2) Then under Contract , select ‘NFT.sol’



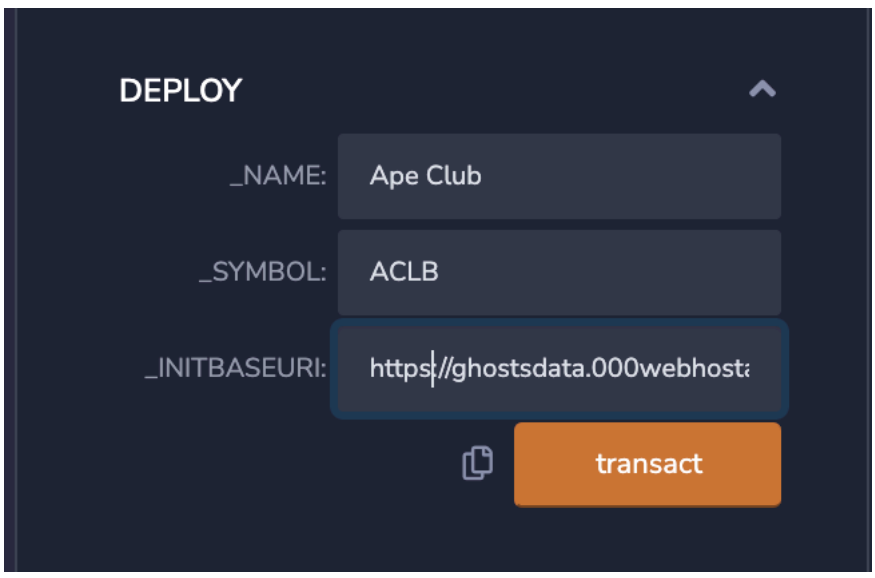
7.3) Press the arrow to the right to expose the options.





7.4 Here you will find the name , symbol and baseURI for the collection. The baseURI refers to the URL of the metadata. So for this collection it is https://ghostsdata.000webhostapp.com/metadata/

Here is the example :

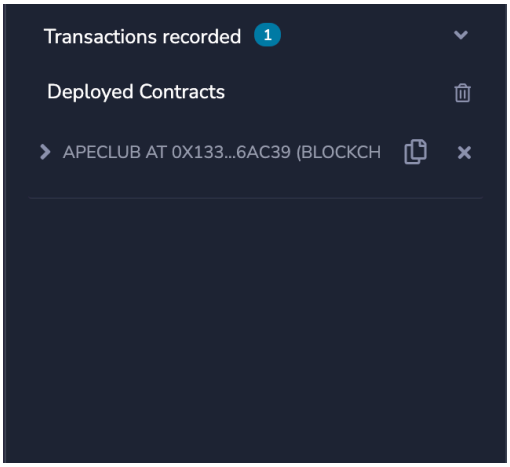


7.5) Press the transact button to deploy the contract. You will then be asked to sign the transaction in MetaMask and pay the gas fees.

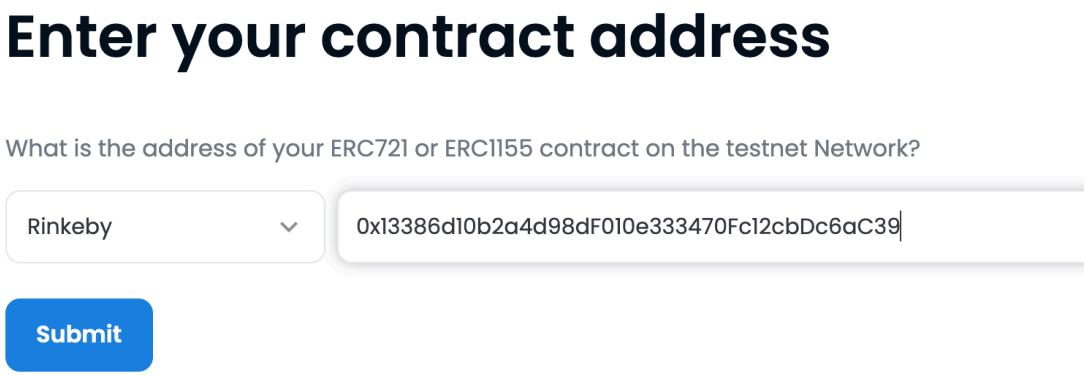
In this case I am deploying on the Rinkeby test net.

7.6) Once the transaction is confirmed , your contract will pop up in the bottom left corner of the deploy page.

Copy the contract address



7.7) Head over to https://opensea.io/get-listed , select the correct network and paste in the contract address that was received after deploying in Remix. Then press submit.



7.8 So now the collection is working because we can see it on OpenSea and all our metadata and images are correct. Great , time to move on!

testnets.opensea.io/collection/ape-club-v2

OpenSea Testnets

Search items, collections, and accounts

Explore Stats Resources Create

1	1	---	0.00
item	owner	floor price	volume traded

Welcome to the home of Ape Club V2 on OpenSea. Discover the best items in this collection.

Items Activity

Filter

Status

Buy Now On Auction

New Has Offers

Price

United States Dollar (USD)

Min to Max

Apply

Chains

On Sale In

Search

Single Items

Price: Low to High

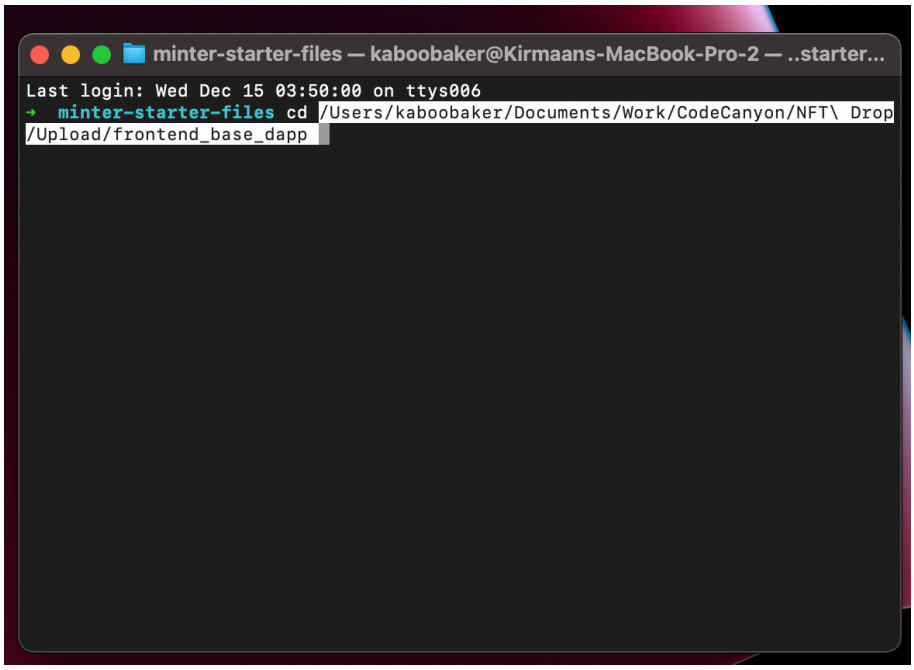
1 result

Ape Club V2
#1

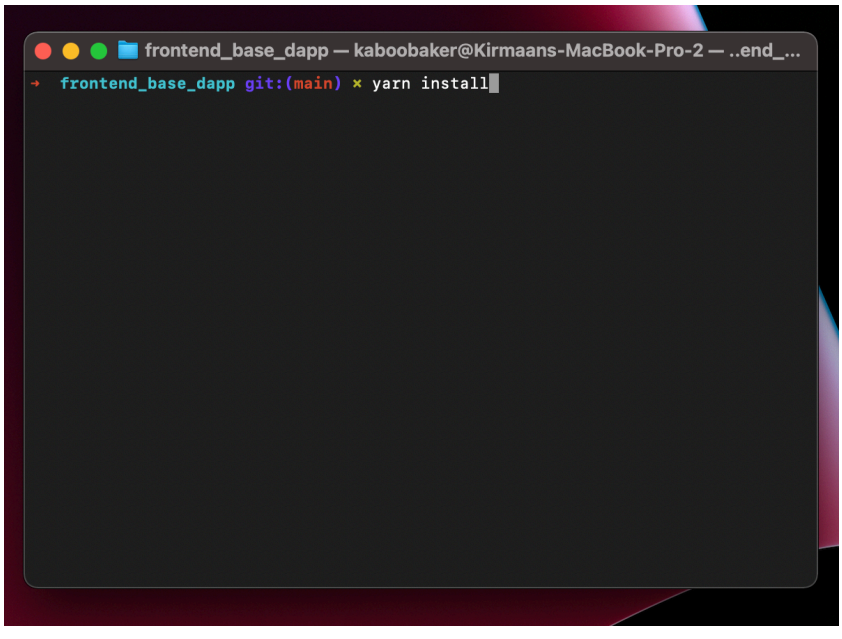
0

PART THREE

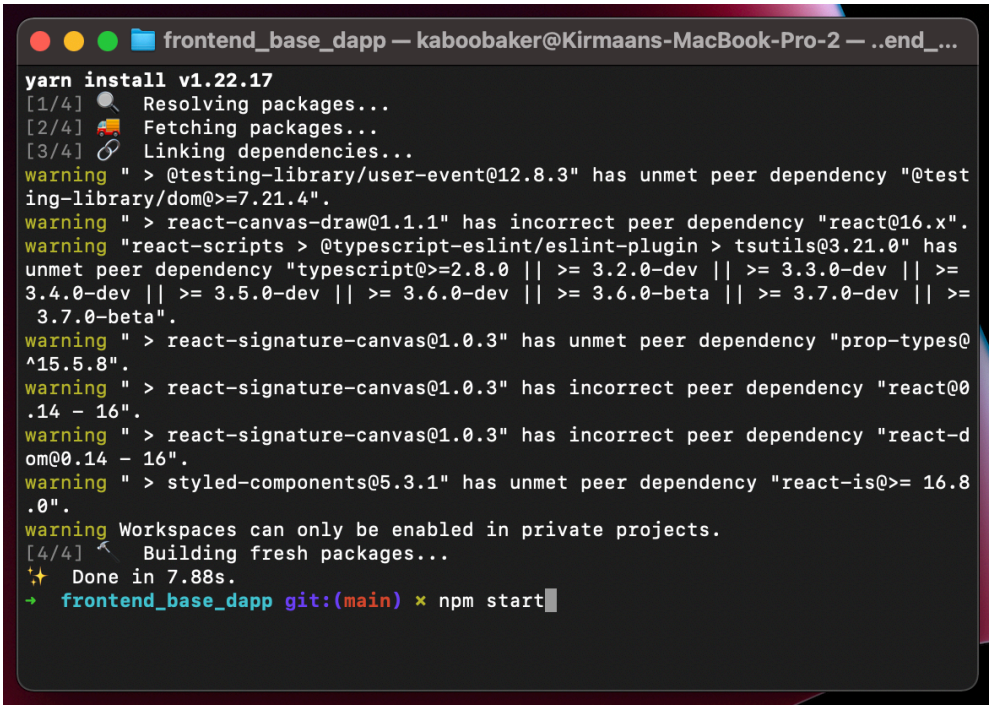
1.) Unzip the project file and navigate into the ‘frontend_base_dapp’ folder using terminal



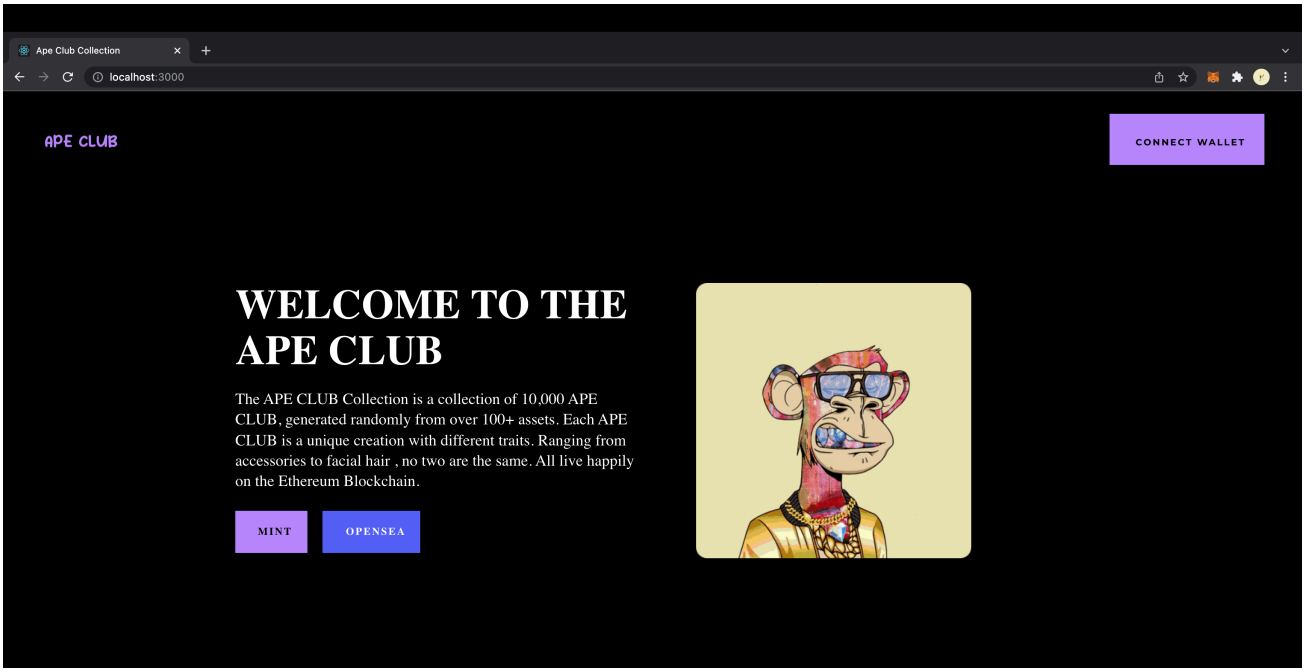
2.) Initialise the node modules with the following command ‘yarn install’



3.) Once completed , run the command ‘npm start’ so we can make sure the project is working correctly.



3.1) The project will launch in your default browser and you should have the working Dapp running!



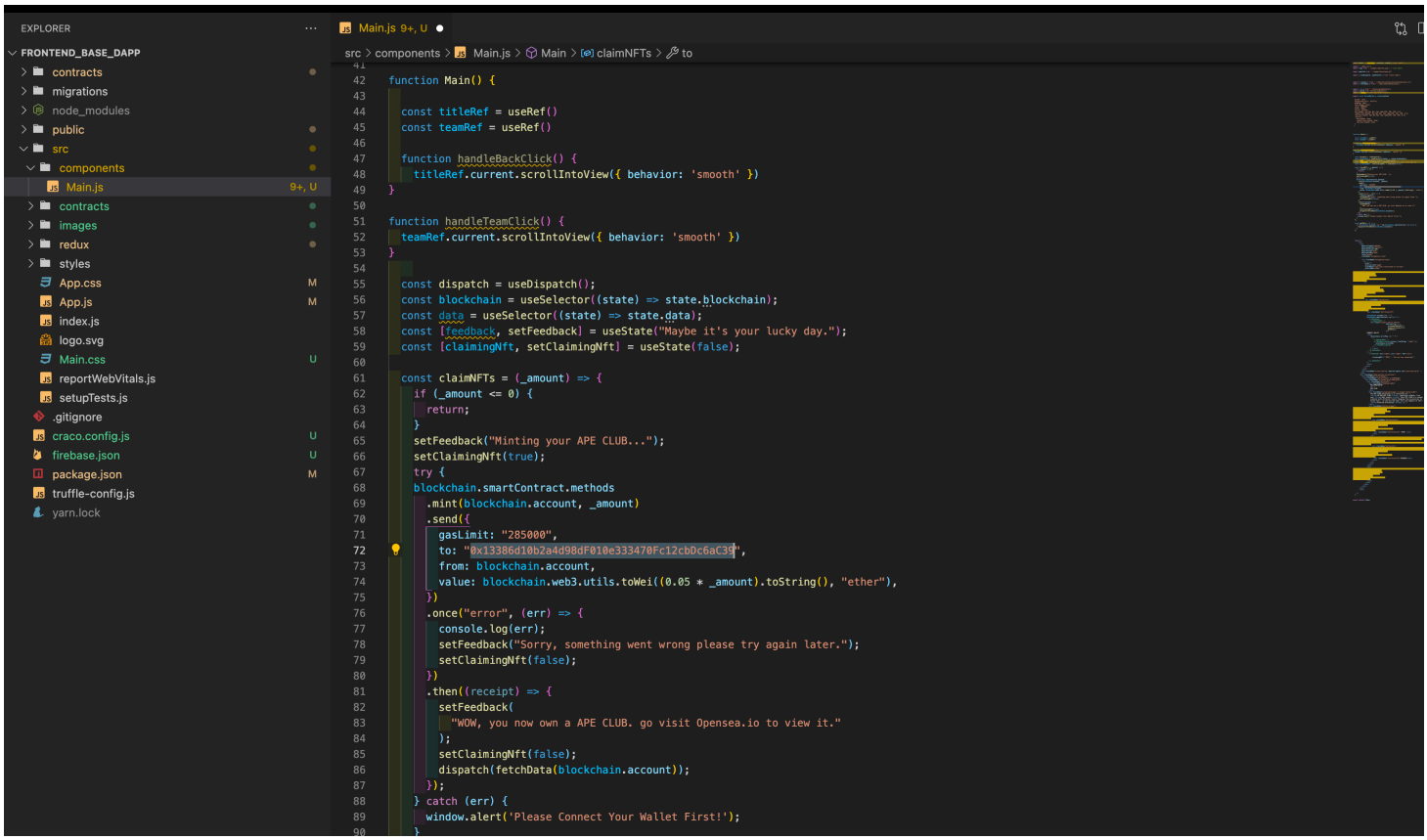
PART FOUR

1.) So now its time to swap in your contract address so that it mints an NFT from your collection.

Open the ‘frontend_base_dapp’ directory in your favourite text editor

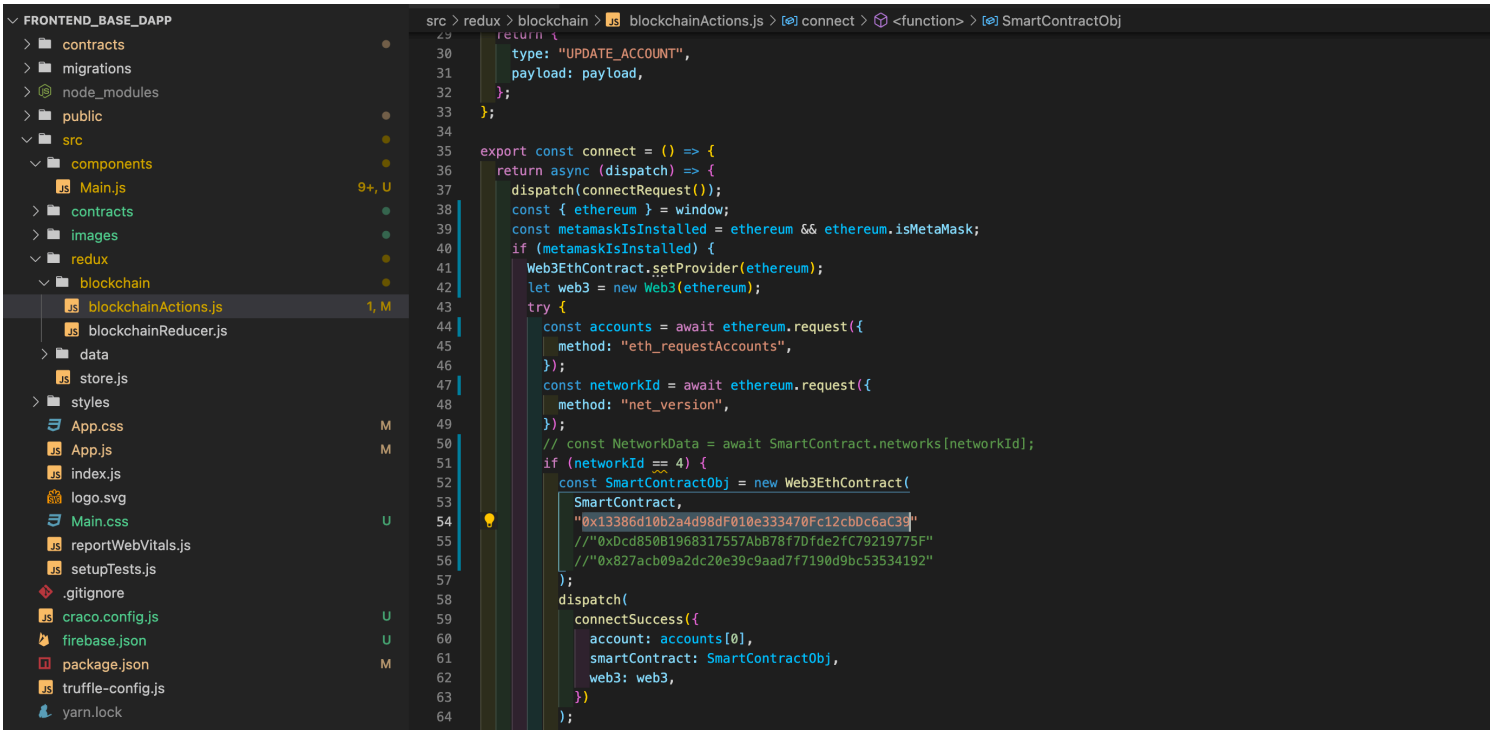
1.1) And navigate to ‘Main.js’ in the components folder located in the src folder in the root directory.

Replace the contract address on line 72 with the one we received from Remix.



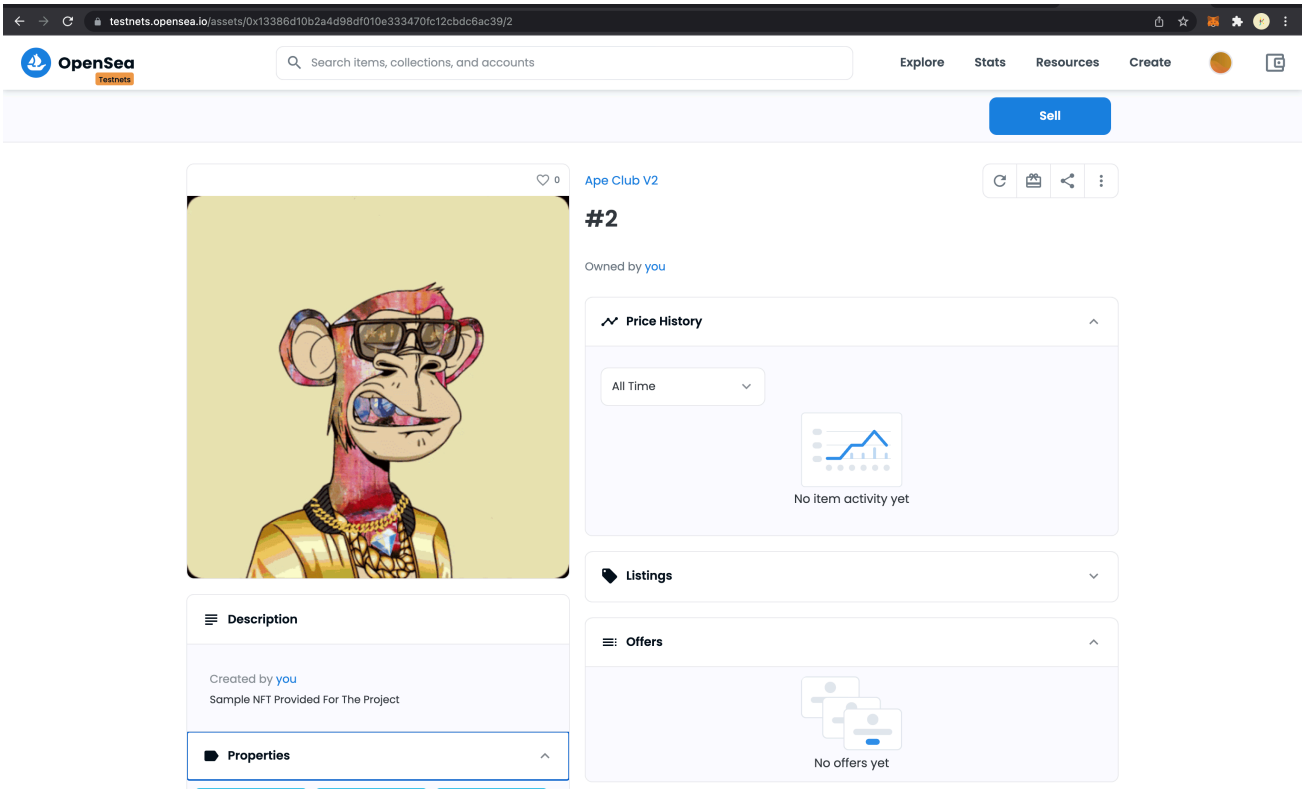
1.2) Now head over to the blockchainActions.js file located in the blockchain folder , which is located in the redux folder located in the src folder in the root of the project.

Replace the contract on line 54 with the one we received from Remix.



2.) Run the project again with ‘npm start’.

Connect your wallet and mint an item , it should work absolutely perfectly!



WE ARE NOW COMPLETE!