

Bhartiya Vidya Bhavan’s

Sardar Patel Institute of Technology, Mumbai-400058

Department of Computer Science and Engineering

**OEIT1:Blockchain Technology and Applications**

**Lab-6A: Create Non Fungible Tokens (NFTs)**

Student Name: Adwait Purao

UCID: 2021300101

Branch: COMPS B

**Objective:** **Create Nonfungible Tokens (NFTs)**

**Outcomes:** After successful completion of lab students should be able to

1. Create a new NFT project
2. Configure the network to deploy on Ganache
3. Configure the private key
4. Create the badge image
5. Add the badge to the local IPFS
6. Pin the badge to a remote IPFS
7. Create the badge metadata
8. Deploy the smart contract
9. Award the badge to your wallet
10. Check the badge on Etherscan
11. Add the badge to your mobile wallet

**System Requirements:**

PC (C2D, 4GB RAM, 100GB HDD space and NIC),Ubuntu Linux 14.04/20.04

Internet connectivity,Python Cryptography and Pycrypto,REST API,Go Language

Go Ethereum, Truffle Suite, Metamask, IPFS, OpenSea Market Place

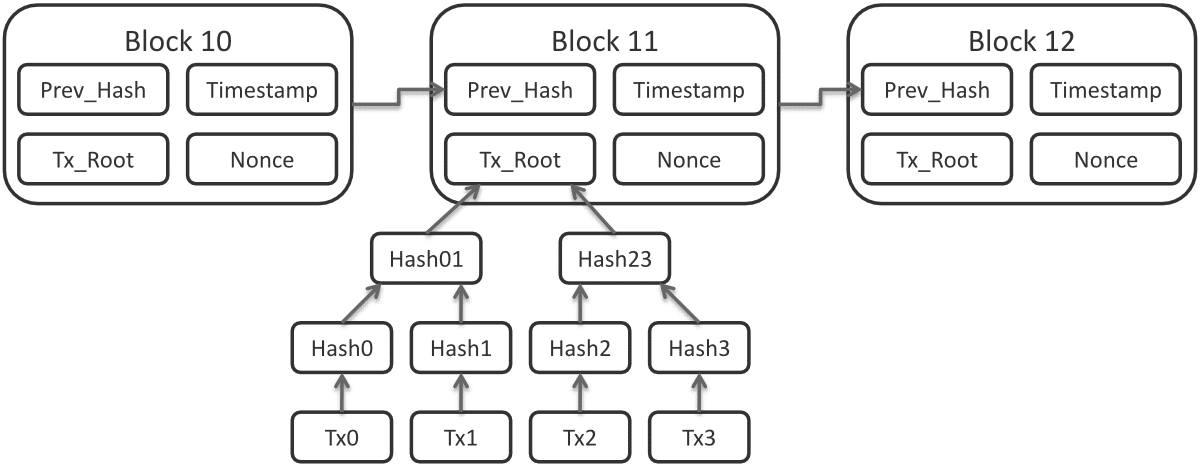


Figure-1: Blockchain Implementation

**Procedure:**

[1] Install Ethereum Blockchain

Clone it from git and compile it. <https://github.com/ethereum/go-ethereum>

$cd ~

$mkdir BTA

$cd BTA

$mkdir lab6a

$cd lab6a

Refer to [1]

Follow the steps and add screenshots for every step with captions.

**UniqueContract.sol**

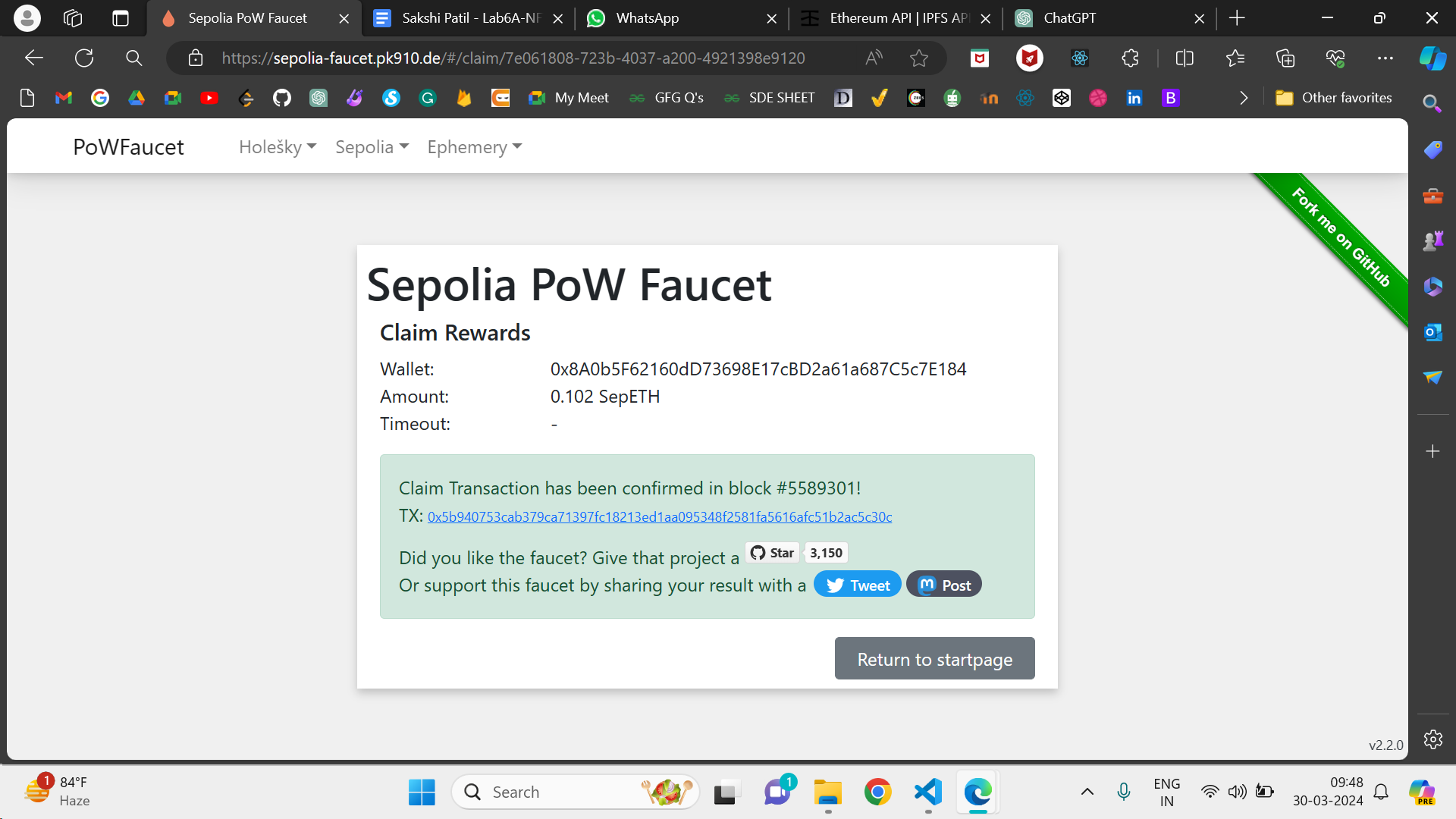
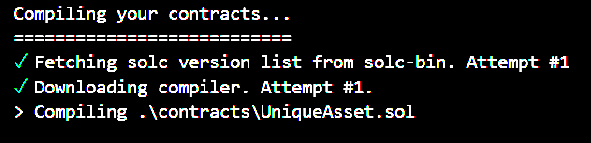
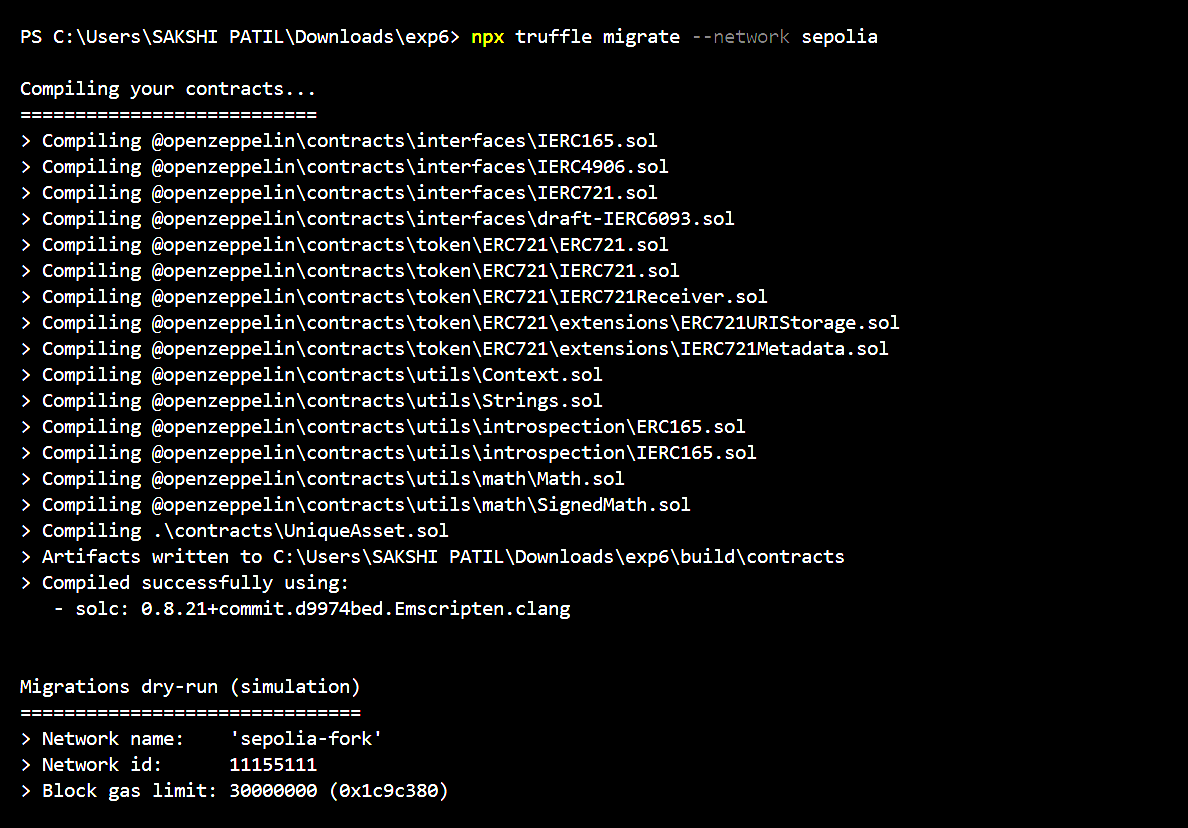
| // SPDX-License-Identifier: MIT pragma solidity ^0.8.19;  import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";  contract UniqueAsset is ERC721URIStorage {  uint private \_tokenIds;   constructor() ERC721("UniqueAsset", "UNA") {}   function awardItem(address recipient, string memory metadata) public returns (uint256) {  uint256 newItemId = \_tokenIds;  \_mint(recipient, newItemId);  \_tokenIds += 1;  \_setTokenURI(newItemId, metadata);  return newItemId;  } } |
| --- |

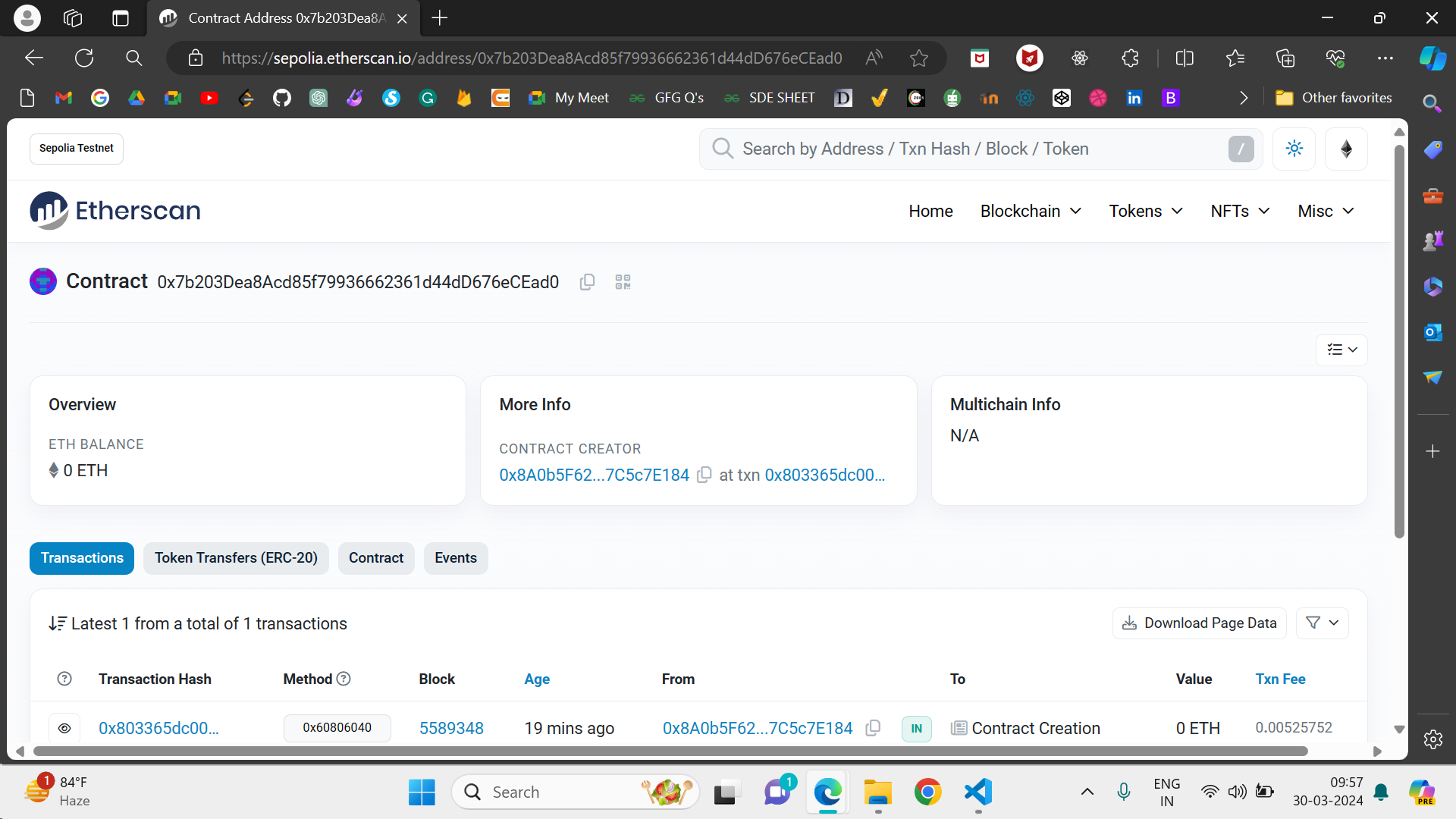
**migration file**

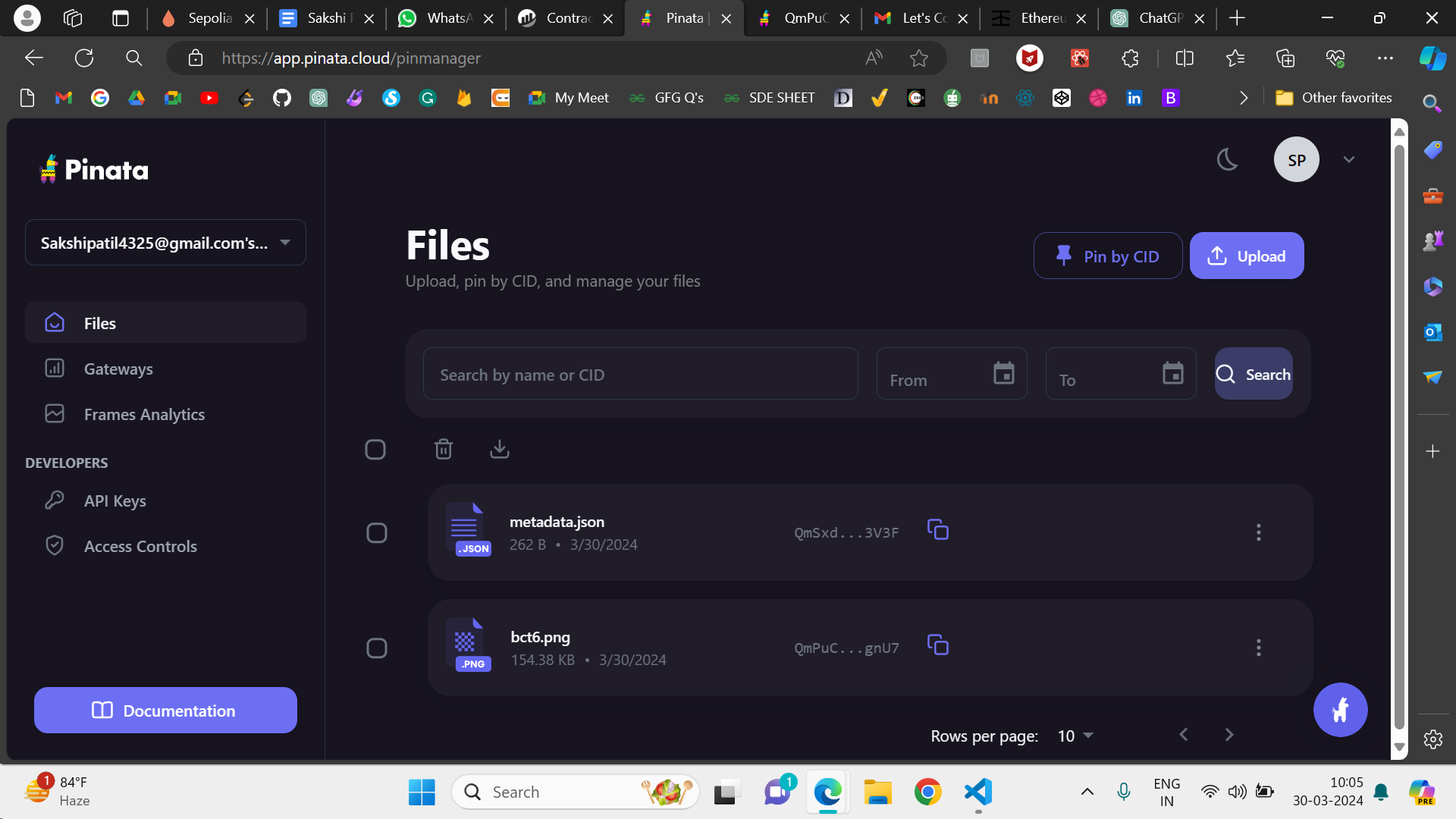
| const UniqueAsset = artifacts.require("UniqueAsset");  module.exports = function(deployer) {  deployer.deploy(UniqueAsset); }; |
| --- |

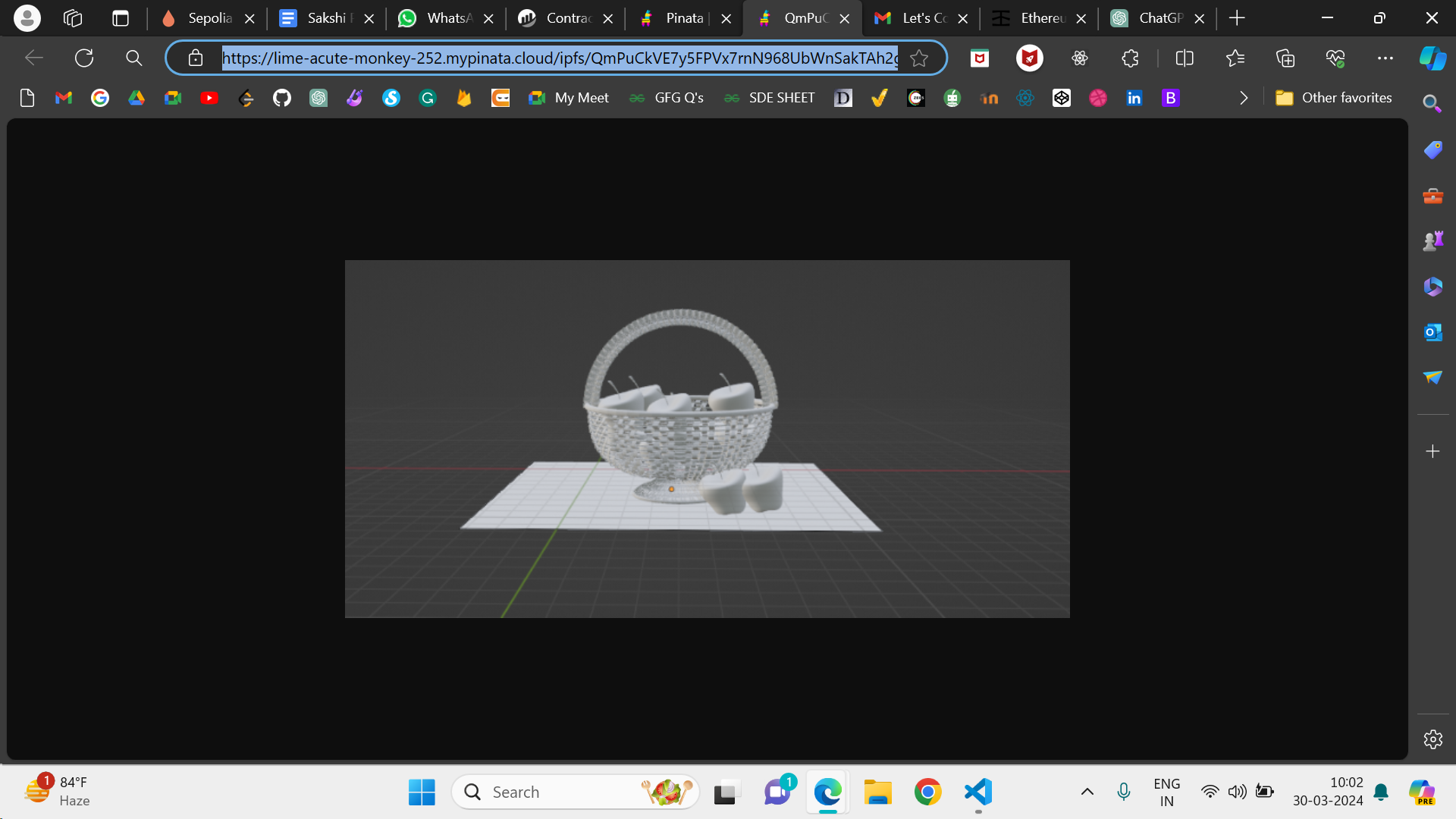
**truffle-config.js**

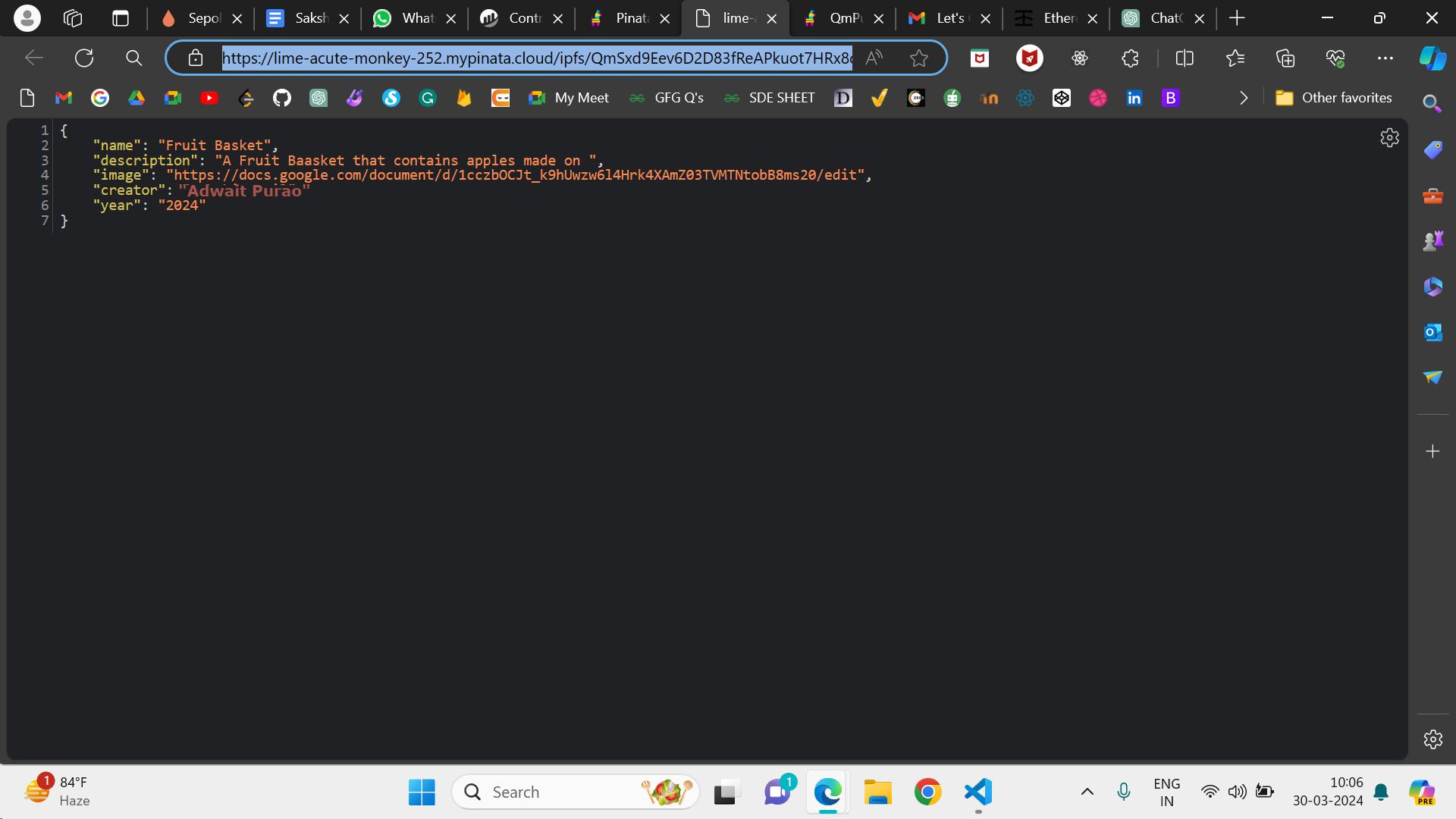
| **const HDWalletProvider = require('@truffle/hdwallet-provider');  module.exports = {  networks: {  sepolia: {  provider: () => new HDWalletProvider(  "drastic taste win among vapor best thrive uncover vibrant index quarter margin",  "https://sepolia.infura.io/v3/fcb58e5accf94b498717e664cd2a3c2e"  ),  network\_id: 11155111, // Sepolia's network id  chain\_id: 5, // Sepolia's chain id  gas: 5500000, // Gas limit used for deploys  confirmations: 2, // # of confirmations to wait between deployments (default: 0)  timeoutBlocks: 2000, // # of blocks before a deployment times out (minimum/default: 50)  skipDryRun: false // Skip dry run before migrations? (default: false for public nets)  },  },  compilers: {  solc: {  version: "0.8.21", // Fetch exact version from solc-bin (default: truffle's version)  settings: { // See the solidity docs for advice about optimization and evmVersion  optimizer: {  enabled: false,  runs: 200  },  evmVersion: "byzantium"  }  }  } };** |
| --- |

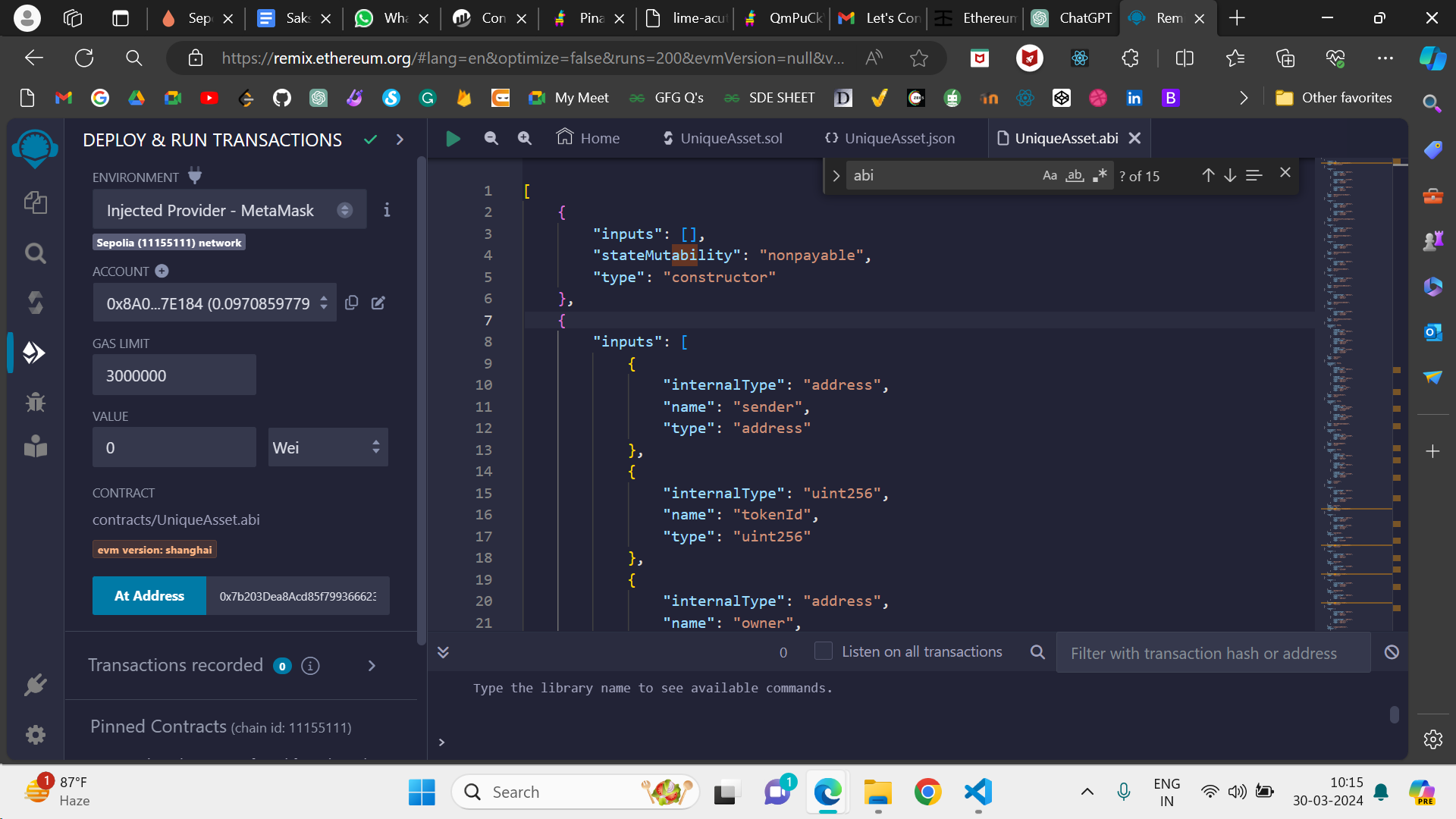
****

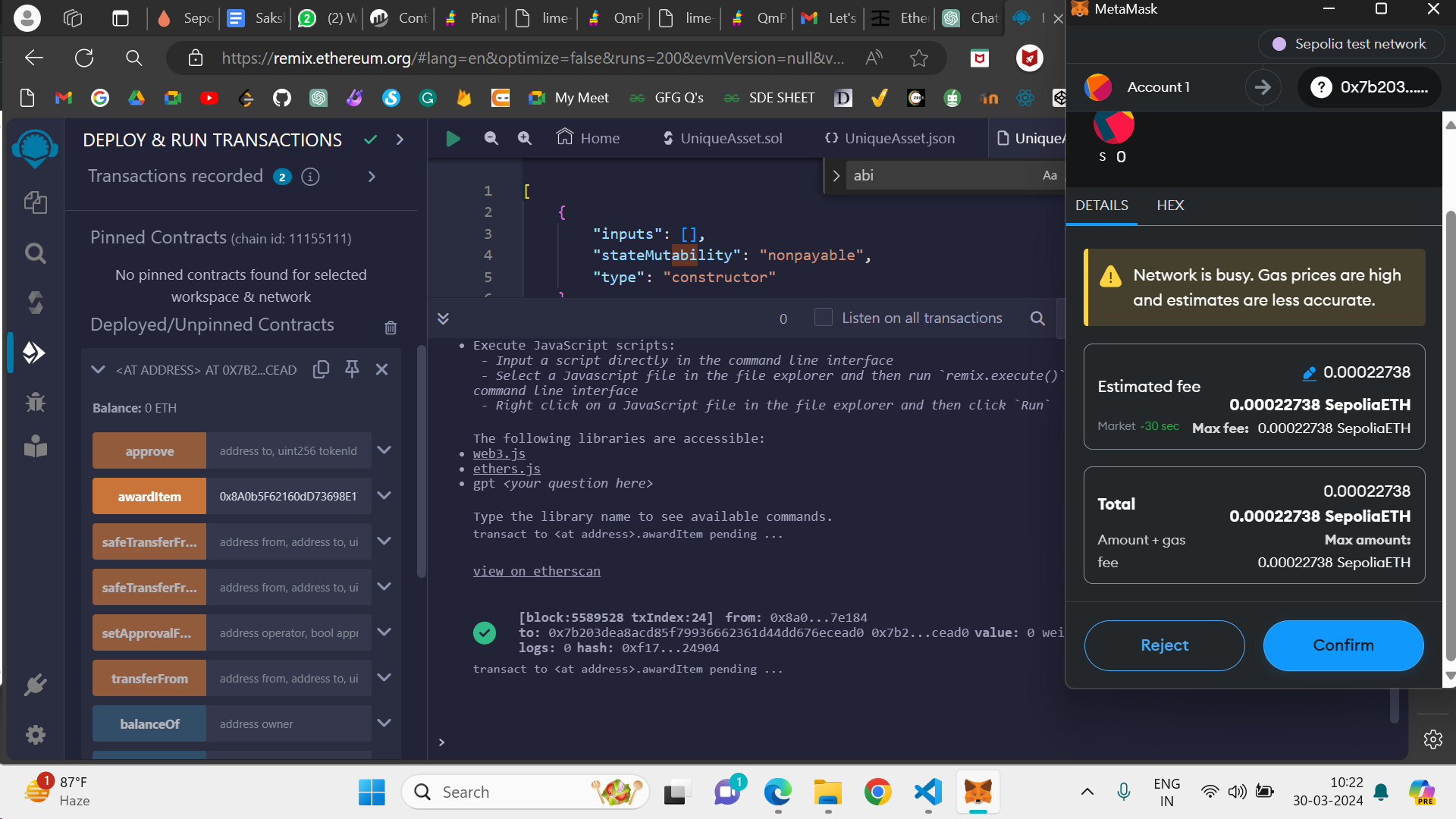
****

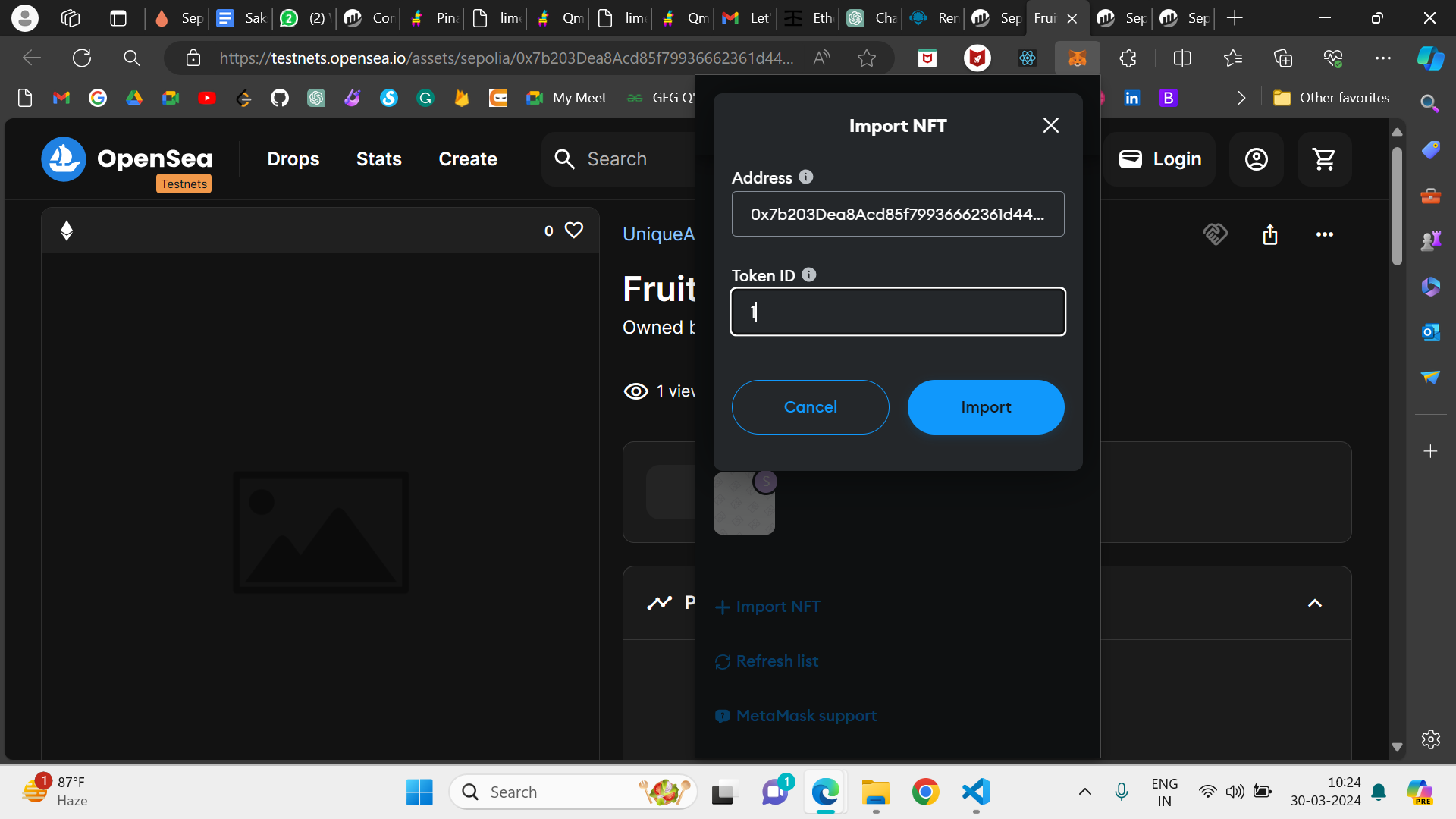
****

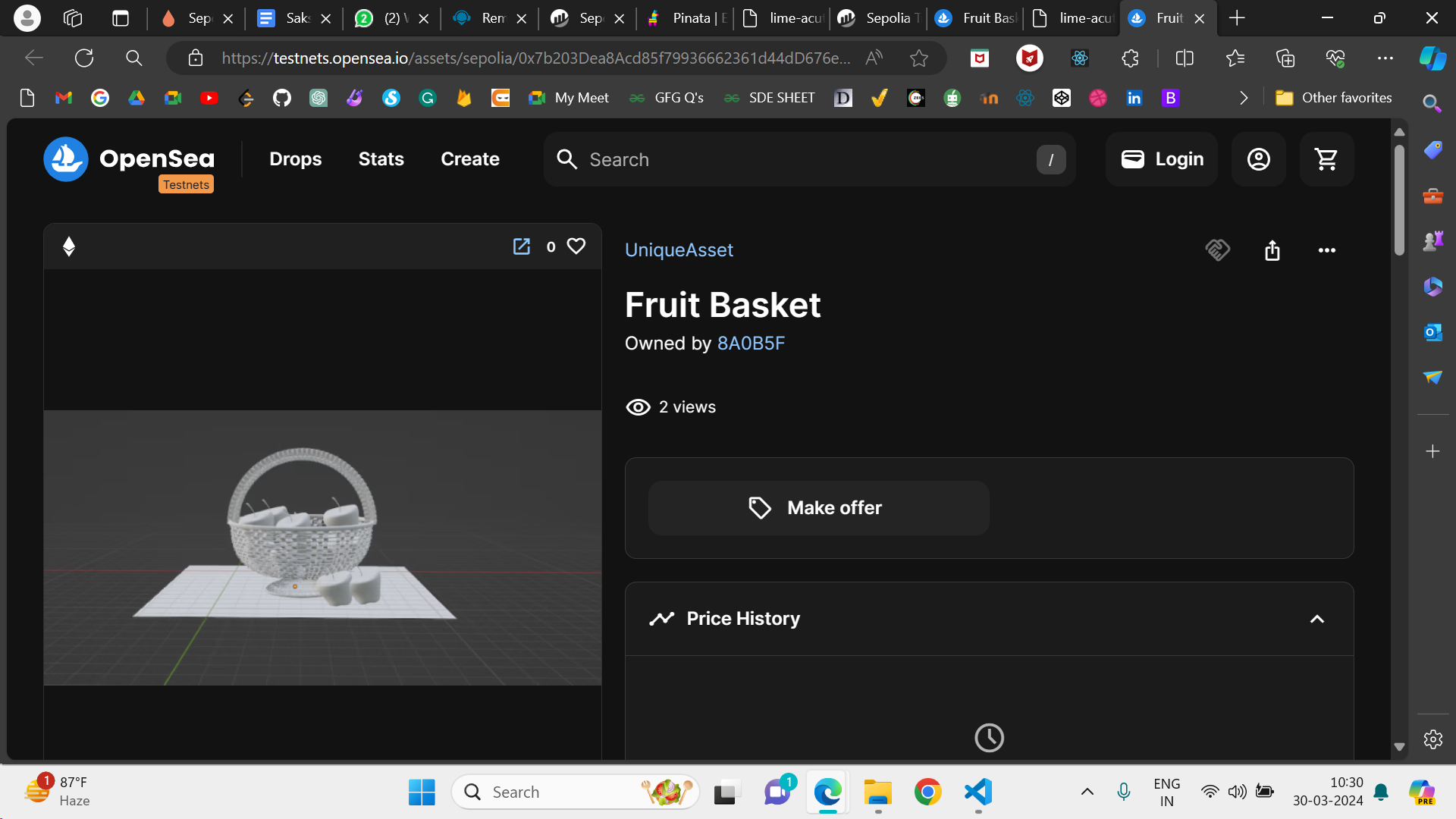
****

****

****

****

****

****

[**Fruit Basket - Asset Opensea**](https://testnets.opensea.io/assets/sepolia/0x7b203Dea8Acd85f79936662361d44dD676eCEad0/2)

**Conclusion:**

In this lab, we delved into the process of crafting Nonfungible Tokens (NFTs) on the Ethereum blockchain through the Sepolia network. Our journey began with setting up the development environment, initializing a new NFT project, and fine-tuning the network for deployment on Sepolia. Leveraging a Solidity contract, we sculpted a token that extends the ERC721URIStorage contract, thus enabling the minting of distinct tokens paired with metadata. With Truffle as our tool, we orchestrated the deployment, landing our contract on the Sepolia network, and allocated tokens to designated addresses. Our exploration culminated in scrutinizing the tokens on Etherscan and mastering the inclusion of these tokens in mobile wallets and marketplaces like OpenSea, equipping us with practical prowess in NFT conception and deployment.

**References:**

[1] Chapter 5 ERC-721 Non Fungible Tokens

https://www.oreilly.com/library/view/getting-started-with/9781484280454/html/521550\_1\_En\_5\_Chapter.xhtml

[2] Mastering Blockchain Technology by Imran Bashir 4th Edition Chapter 11,12 and 13,Packt Publications