Centurion University
Shaping Lives... Empowering Communities...

School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiement : Mint it Yourself – NFT Creation and Deployment

Objective/Aim:

- Understand ERC-721 (NFT) concepts: token ID, metadata, tokenURI, owner, transfer.
- Create NFT metadata (JSON + image) and pin to IPFS.
- Deploy an ERC-721 contract on Sepolia.
- Mint NFTs (owner mint via script + user mint via a minimal web dApp).

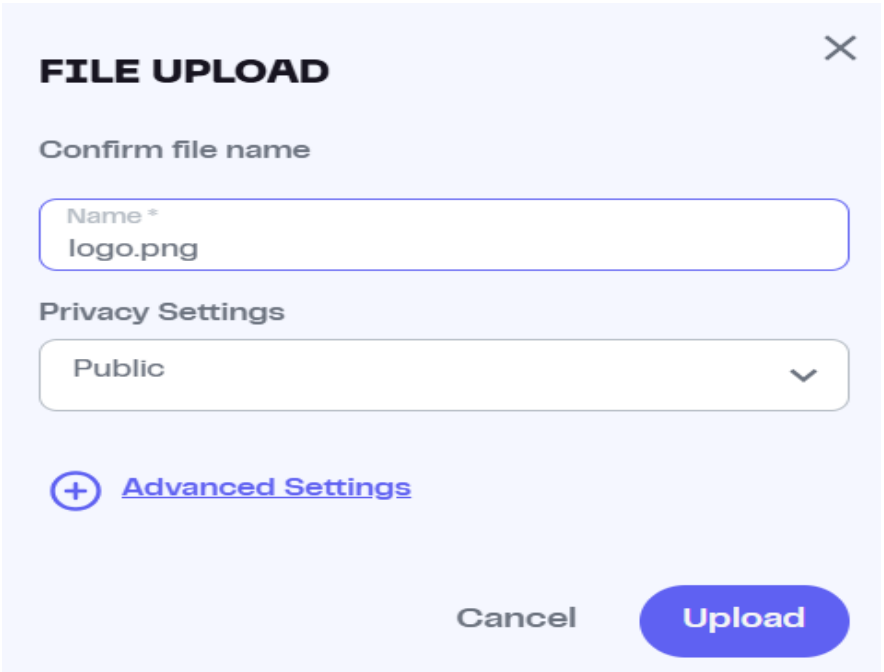
Apparatus/Software Used:

- MetaMask
- Etherscan **(https://goerli.etherscan.io/) for Ethereum testnets.**

.

Procedure:

1.First open pinata and uplode a photo and copy the url of the photo.

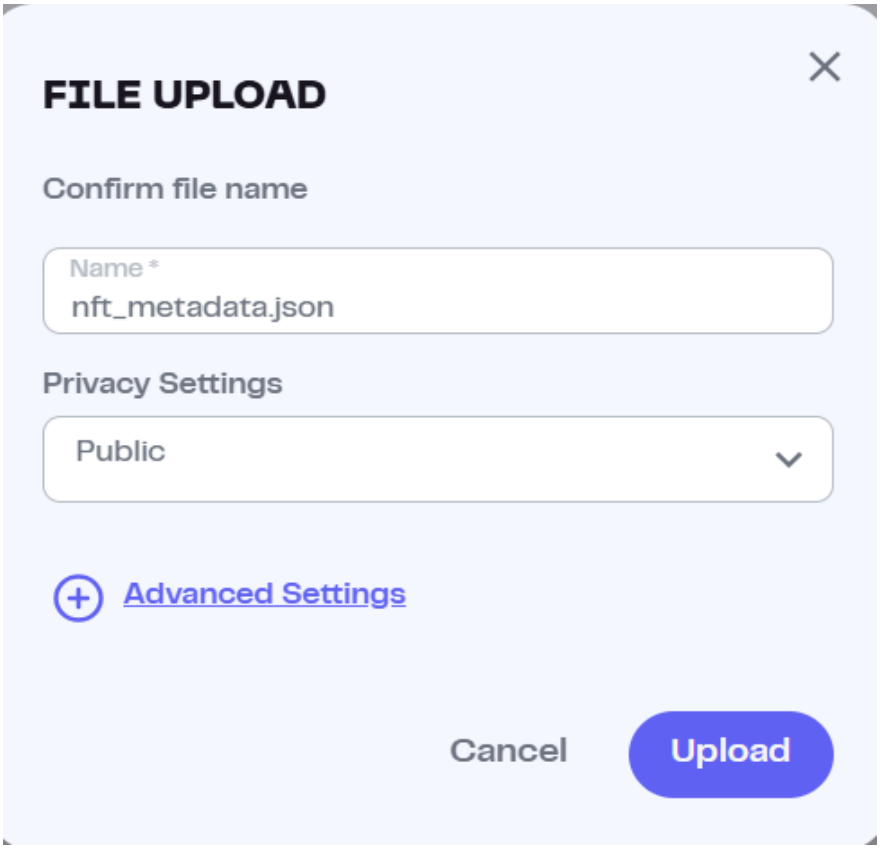


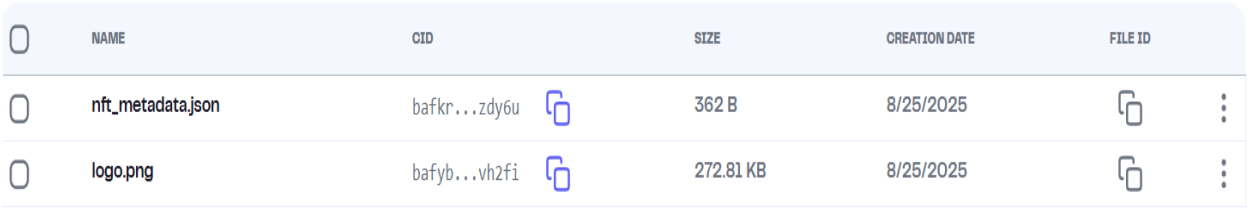
Procedure:

2.Then go to vs code and write code in nft_metadata.json file and past the copied url of the photo.

```
{
  "name": "Shruti",
  "description": "NFT demo for Blockchain Studnets on Sepolia.",
  "image": "https://tomato-fancy-krill-754.mypinata.cloud/ipfs/bafybeig5nj74lnjg7gnkckpjqhxsgjqabqh55j4wcuuuuphf5wdgjvh2fi",
  "attributes": [
    {
      "trait_type": "photo",
      "value": "s"
    },
    {
      "trait_type": "logo",
      "value": "ss"}
]
}
```

3. Now go to pinata and uplode this nft_metadata.json file.



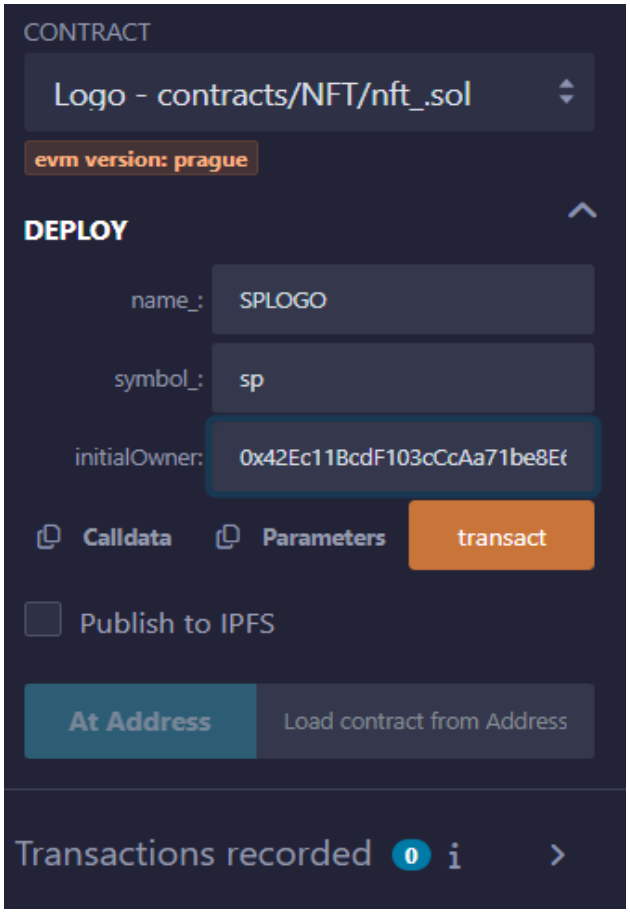


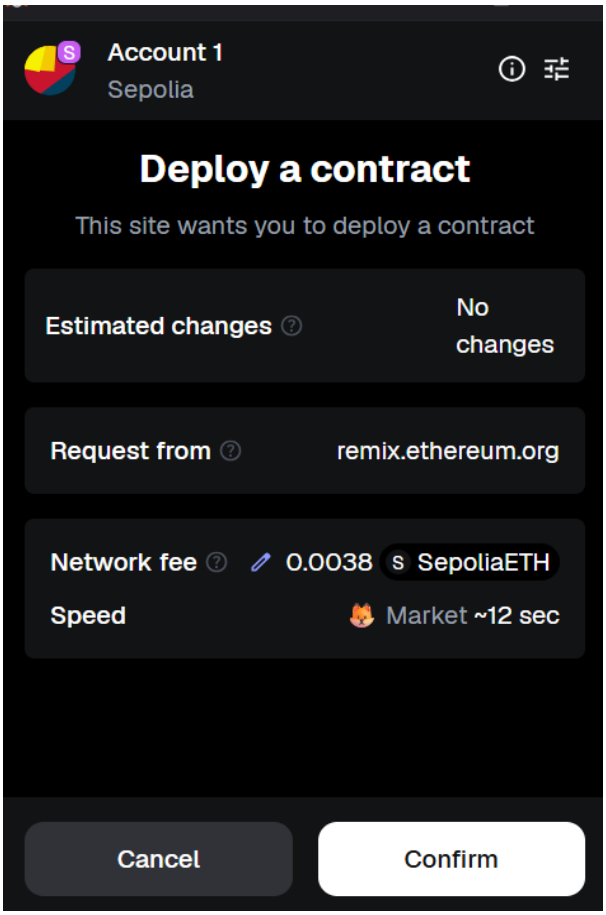
Procedure:

4. After uploding file go to remix and write smart contact for creating Nft and deploye the smart contract.

```
pragma solidity ^0.8.24;

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

/// Simple ERC721 with per-token URI (works great with IPFS + OpenSea)
contract Logo is ERC721URIStorage, Ownable {
    uint256 private _nextId;

    // Pass the owner address on deploy (OpenZeppelin v5 pattern)
    constructor(string memory name_, string memory symbol_, address initialOwner)   infinite gas 1930600 gas
        ERC721(name_, symbol_)
        Ownable(initialOwner)
    {}

    /// Mint to to with a full metadata URI like ipfs://<CID>/metadata.json
    function mintTo(address to, string memory metadataURI) external onlyOwner returns (uint256) {   infinite gas
        _nextId += 1;
        uint256 tokenId = _nextId;
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, metadataURI);
        return tokenId;
    }

    function totalMinted() external view returns (uint256) {   2454 gas
        return _nextId;
}
```

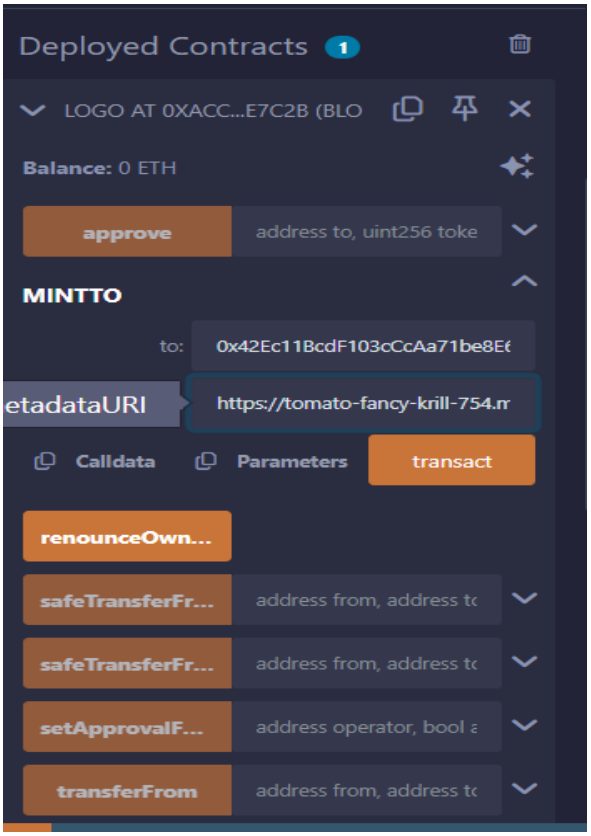
5. After compailation go to the deploye section and give the sring name,symbol and initial owner(wallet address) and deploye it by cofirming transaction on MetaMask.

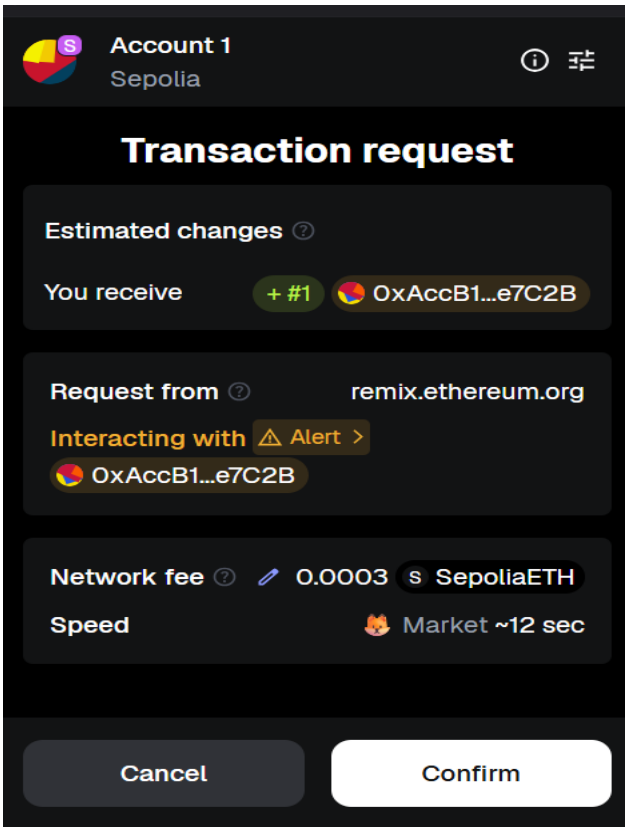




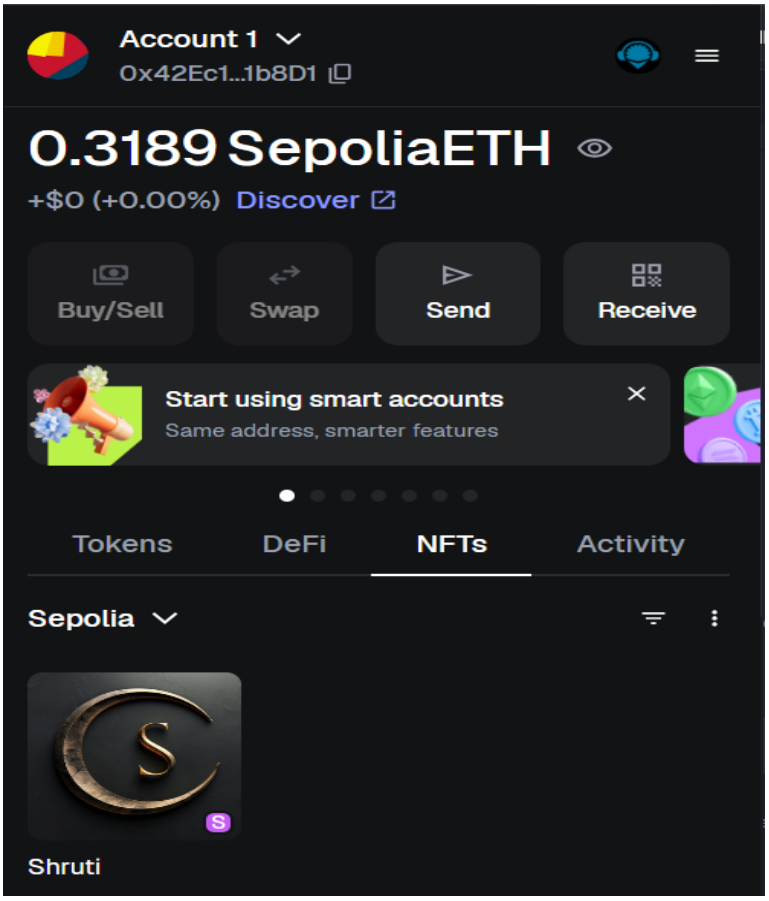
Procedure:

6. Now go to the deployed contract and in the this contract section there is MINTTO button , here in this button gibe the wallet address and metadataURI and mint it.





7. After transaction confirmed, check your MetaMask wallet the NFT is display.



Observation:

- The wallet connected to Sepolia with test ETH.
- Image and metadata were uploaded to IPFS successfully.
- The contract deployed without errors and minted Token ID 1.
- tokenURI returned the correct IPFS link showing the NFT metadata.
- Public mint worked and the NFT was visible on the block explorer and wallet.

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	**50**		

Signature of the Student:

Name :

Signature of the Faculty:

Regn. No. :

**As applicable according to the experiment.*
Two sheets per experiment (10-20) to be used.