



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 Experiment :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

NFT Creation and Deployment

Step 1: Prepare NFT Metadata

1. Open VS Code (Visual Studio Code).
2. Create a JSON file named nft_metadata.json with your NFT metadata:

```
{  
  "name": "CUTM Badge #2",  
  "description": "NFT demo for Blockchain Studnets on Sepolia.",  
  "image": "https://brown-important-alpaca-  
785.mypinata.cloud/ipfs/bafkreih42pxk2yk6jzxi24nzhn34ycfxislehun6mgkws4ulsn3vg  
h5mb4",  
  "attributes": [  
    {  
      "trait_type": "Department",  
      "value": "CSE"  
    },  
    {  
      "trait_type": "Campus",  
      "value": "BBSR"  
    }  
  ]  
}
```

Step 2: Upload NFT Image to IPFS via Pinata

1. Go to Pinata and log in.
2. Navigate to the API Keys section and generate a new key with appropriate access.
3. Upload your NFT image (e.g., ganesh.png) to IPFS using the Pinata upload tool.
4. Copy the CID or the IPFS URL of the uploaded image.
5. Paste the IPFS image link (e.g., ipfs://<CID>/ganesh.png) into the image field of your nft_metadata.json.

Step 3: Upload Metadata JSON to IPFS

1. After updating your metadata file with the correct image IPFS link, upload nft_metadata.json to Pinata.

Coding Phase: Pseudo Code / Flow Chart / Algorithm

1. Copy the CID or IPFS URL of this metadata file.
Example: ipfs://<CID>/nft_metadata.json

Step 4: Write the Smart Contract in Solidity

1. Open Remix IDE.
2. Create a new file named NFT.sol.
3. Paste the following contract code:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.24;
```

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

```
contract Ganesha is ERC721URIStorage, Ownable {
    uint256 private _nextId;

    constructor(string memory name_, string memory symbol_, address initialOwner)
        ERC721(name_, symbol_)
        Ownable(initialOwner)
    {}

    function mintTo(address to, string memory metadataURI) external onlyOwner returns (uint256) {
        _nextId += 1;
        uint256 tokenId = _nextId;
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, metadataURI);
        return tokenId;
    }

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

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Step 5: Compile the Contract

1. In Remix, navigate to the Solidity Compiler tab.
2. Select the appropriate compiler version (^0.8.24).
3. Click Compile NFT.sol.

Step 6: Deploy the Contract

1. Go to the Deploy & Run Transactions tab.
2. Select the Injected Provider - MetaMask environment to connect your wallet.
3. Provide the constructor parameters:
 - name_: e.g., Ganesha
 - symbol_: e.g., GNSH
 - initialOwner: Your MetaMask address
4. Click Deploy and approve the transaction in MetaMask

Step 7: Mint the NFT

1. After deployment, use the contract's mintTo function:
 - to: Your MetaMask address
 - metadataURI: The IPFS URL of your nft_metadata.json file (e.g., ipfs://<CID>/nft_metadata.json)
2. Click Transact and approve the minting transaction.

Step 8: View Your NFT

- Once the NFT is minted, it should be visible in your MetaMask wallet under NFTs (if supported).

* Softwares used

- Pinata
- Remix IDE
- VScode

* Implementation Phase: Final Output (no error)

The screenshot displays the final output of the implementation phase, showing the code, the file storage, and the deployment process.

Code Editor (nft_metadata.json):

```

1 {
2   "name": "CUTM Badge #2",
3   "description": "NFT demo for Blockchain Studnets on Sepolia.",
4   "image": "https://brown-important-alpaca-785.mypinata.cloud/ipfs/bafkreih42pxk2yk6jzxi24nzhn34ycfxislehur",
5   "attributes": [
6     {
7       "trait_type": "Department",
8       "value": "CSE"
9     },
10    {
11      "trait_type": "Campus",
12      "value": "BBSR"
13    }
14  ]
15 }

```

Pinata Cloud File Storage:

File Name	Hash	Size	Uploaded
Earth	QmWao...utFTq	9.41 KB	8/27/2025
nft_metadata.json	bafkr...qazca	392 B	8/25/2025
Food	bafkr...ddfze	135.21 KB	8/25/2025

Remix - Ethereum IDE:

The Remix IDE interface shows the deployment of the `nft.sol` contract. The contract is named `Ganesha` and is deployed to the `prague` EVM version. The initial owner is `0DcB61E3808Dc351d1cB77Aa4fF`. The contract is deployed to the `default_workspace`.

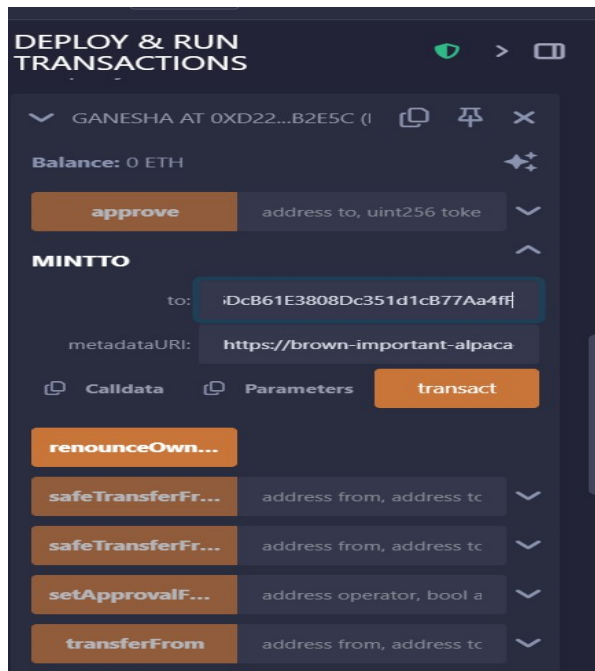
Contract Code (nft.sol):

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.24;
3
4 import "@openzeppelin/contracts/token/ERC721/extension"
5 import "@openzeppelin/contracts/access/Ownable.sol";
6
7 /// Simple ERC721 with per-token URI (works great with
8 contract Ganesha is ERC721URIStorage, Ownable {
9   uint256 private _nextId;
10
11   // Pass the owner address on deploy (OpenZeppelin
12   constructor(string memory name_, string memory sym
13     ERC721(name_, symbol_)
14     Ownable(initialOwner)
15   {}
16
17   /// Mint to `to` with a full metadata URI like ipf

```

Implementation Phase: Final Output (no error)



0.5719 SepoliaETH

+\$0 (+0.00%) [Discover](#)

Buy/Sell

Swap

Send

Receive

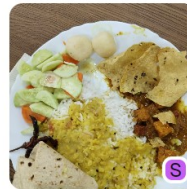
Tokens

DeFi

NFTs

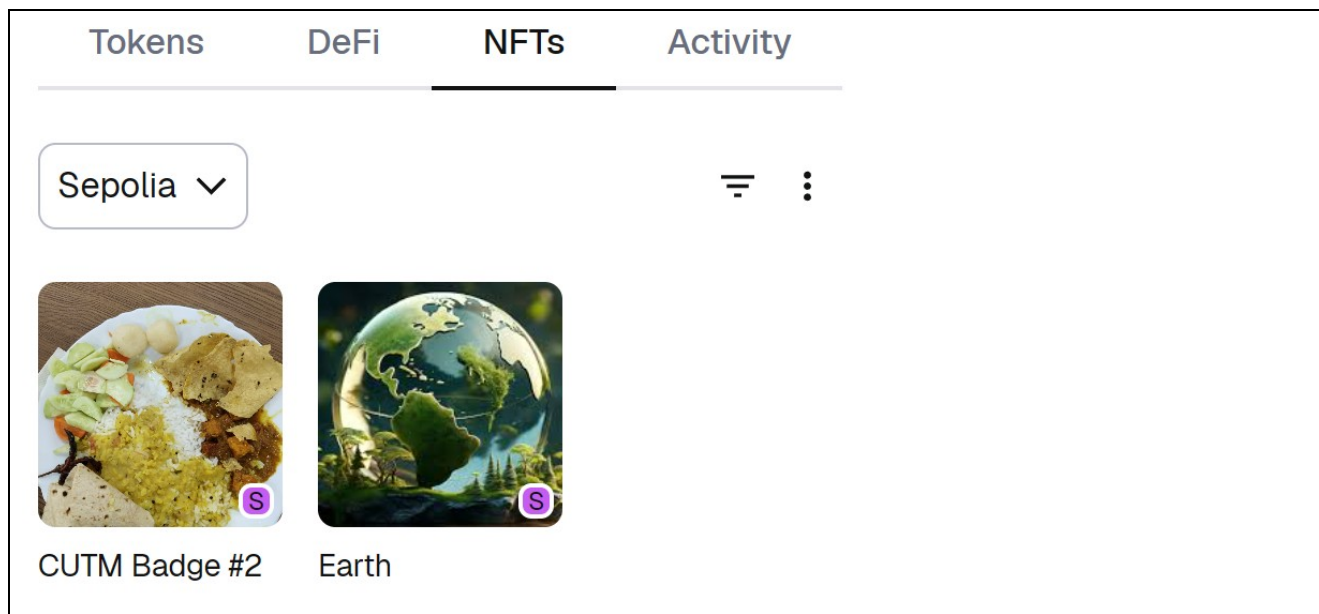
Activity

Sepolia



* Implementation Phase: Final Output (no error)

Applied and Action Learning



* Observations

- Metadata was properly structured and uploaded to IPFS using Pinata.
- Smart contract used OpenZeppelin's ERC721URIStorage for NFT functionality.
- Image and metadata were stored on IPFS, ensuring decentralization.
- Contract deployed successfully using Remix and MetaMask.
- NFT was minted correctly and appeared in the MetaMask wallet.
- The process followed best practices for secure and efficient NFT deployment.

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 student:

Name :

Regn. No.

:

Signature of the Faculty:

Page No.....

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