



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 : Mint it Yourself – NFT Creation and Deployment

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

1. Write Smart Contract

- Define an ERC-721 contract (NFT standard).
- Import OpenZeppelin libraries (ERC721URIStorage, Ownable).
- Add functions:
 - constructor() → set name, symbol, owner.
 - mintTo(address, metadataURI) → mint NFT with metadata.
- Compile contract.

2. Deploy Contract

- Connect Remix with MetaMask (Injected Provider).
- Deploy contract to Sepolia testnet with required parameters.
- Confirm transaction in MetaMask.

3. Mint NFT

- Call the mintTo() function with:
 - Recipient wallet address.
 - Metadata URI (IPFS link from Pinata).
- Confirm minting transaction in MetaMask.

4. Verify in Wallet

- Open MetaMask → NFT section → check NFT minted.
- Optionally, verify contract and view token on **Etherscan** / **Blockscout**.

Software used

1. MetaMask Wallet
2. Remix IDE.
3. MS Word.
4. Brave for researching.

* Implementation Phase: Final Output (no error)

- Write ERC-721 smart contract in **Remix**.
- Compile the contract without errors.
- Deploy on **Sepolia testnet** using MetaMask.
- Upload NFT metadata (image + JSON) to **IPFS (Pinata/Infura)**.
- Copy IPFS CID link → use it as tokenURI.
- Call mintTo() function to mint NFT.
- Confirm transaction in MetaMask.
- NFT appears in **MetaMask** → **NFTs tab**.

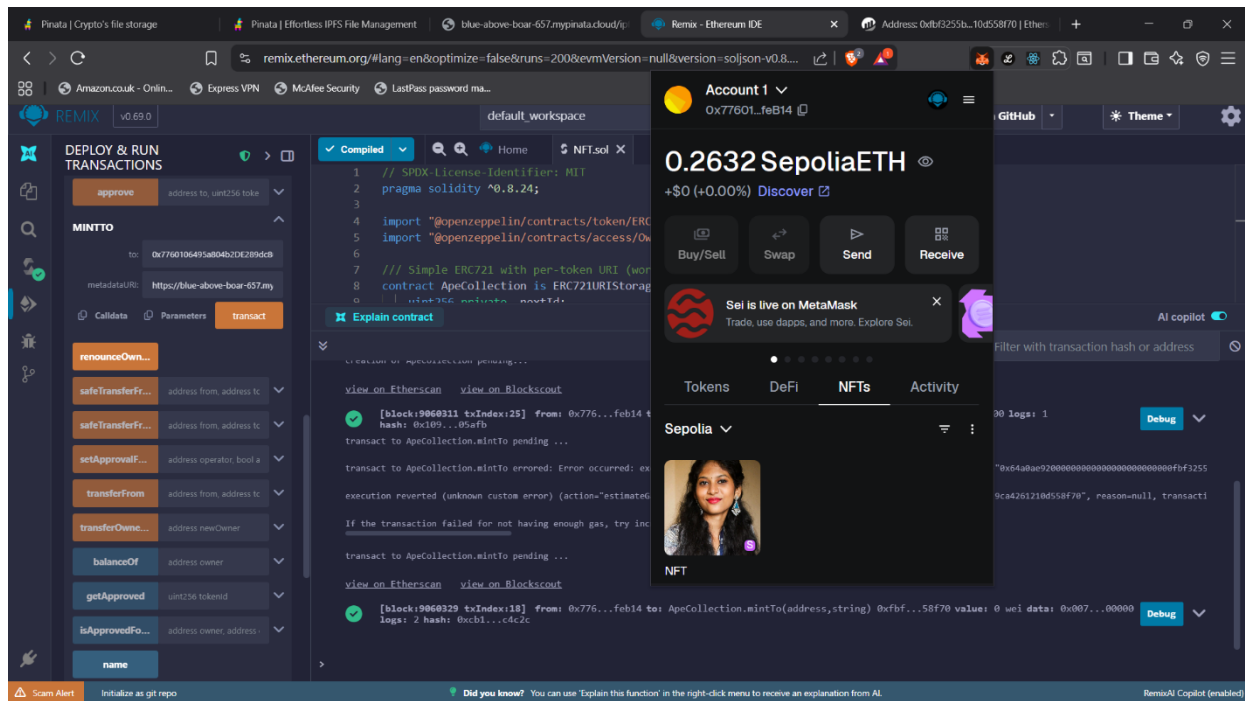
The image displays two screenshots of the Remix IDE and MetaMask interface, illustrating the final steps of the NFT implementation phase.

Top Screenshot: The Remix IDE shows the deployment of the ApeCollection contract. The "DEPLOY & RUN TRANSACTIONS" panel on the left indicates the contract is compiled and ready for deployment. The "DEPLOY" section shows the contract name as "ApeCollection" and the initial owner as "0x776...feb14". The "DEPLOY" button is highlighted. The "MINTTO" section shows the "mintTo" function being called with the address "0x776...feb14" and the token URI "https://blue-above-boar-657.mypinata.cloud/ipfs/0x776...feb14". The "MINTTO" button is highlighted. The "DEPLOY" and "MINTTO" buttons are both highlighted.

Bottom Screenshot: The MetaMask interface shows the "Deploy a contract" transaction request. The "Estimated changes" section shows "No changes". The "Request from" section shows "remix.ethereum.org". The "Network fee" section shows "0.0061 SepoliaETH". The "Speed" section shows "Market ~12 sec". The "Confirm" button is highlighted.

The bottom screenshot also shows the "Transaction request" panel in MetaMask, displaying the "Estimated changes" section with "You receive +1 #1 0xf32...58f70". The "Request from" section shows "remix.ethereum.org". The "Interacting with" section shows "Alert" and "0xf32...58f70". The "Network fee" section shows "0.0006 SepoliaETH". The "Speed" section shows "Market ~12 sec". The "Confirm" button is highlighted.

* Implementation Phase: Final Output (no error)



* Observations:

- ☐ **MetaMask Network** must be set to **Sepolia** before deploying/minting.
- ☐ Each action (deploy/mint) requires **gas fees (SepoliaETH)**.
- ☐ Metadata must follow JSON schema (image, name, description).
- ☐ The NFT becomes visible in MetaMask only if:
 - Metadata URI is valid.
 - NFT contract is ERC-721 compliant.
- ☐ Etherscan shows transaction history and confirms NFT ownership.

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 :

Regn. No. :

Signature of the Faculty: