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 : Build a Market – Basic NFT Marketplace Logic

Objective/Aim:

Implement a basic NFT marketplace on a testnet where users can list NFTs for sale and buy listed NFTs using test ETH. Perform marketplace interactions using MetaMask and record observations.

Apparatus/Software Used:

- MetaMask Wallet
- Brave Web Browser
- Remix IDE – https://remix.ethereum.org
- Ethereum Sepolia Testnet
- OpenZeppelin Contracts

Theory/Concept:

NFT (ERC-721):
A Non-Fungible Token standard representing unique digital items.

NFT Marketplace:
A smart-contract system allowing users to:

- List their NFTs for sale
- Set a price
- Purchase NFTs from sellers
- Transfer ownership securely on-chain

Key Functions:

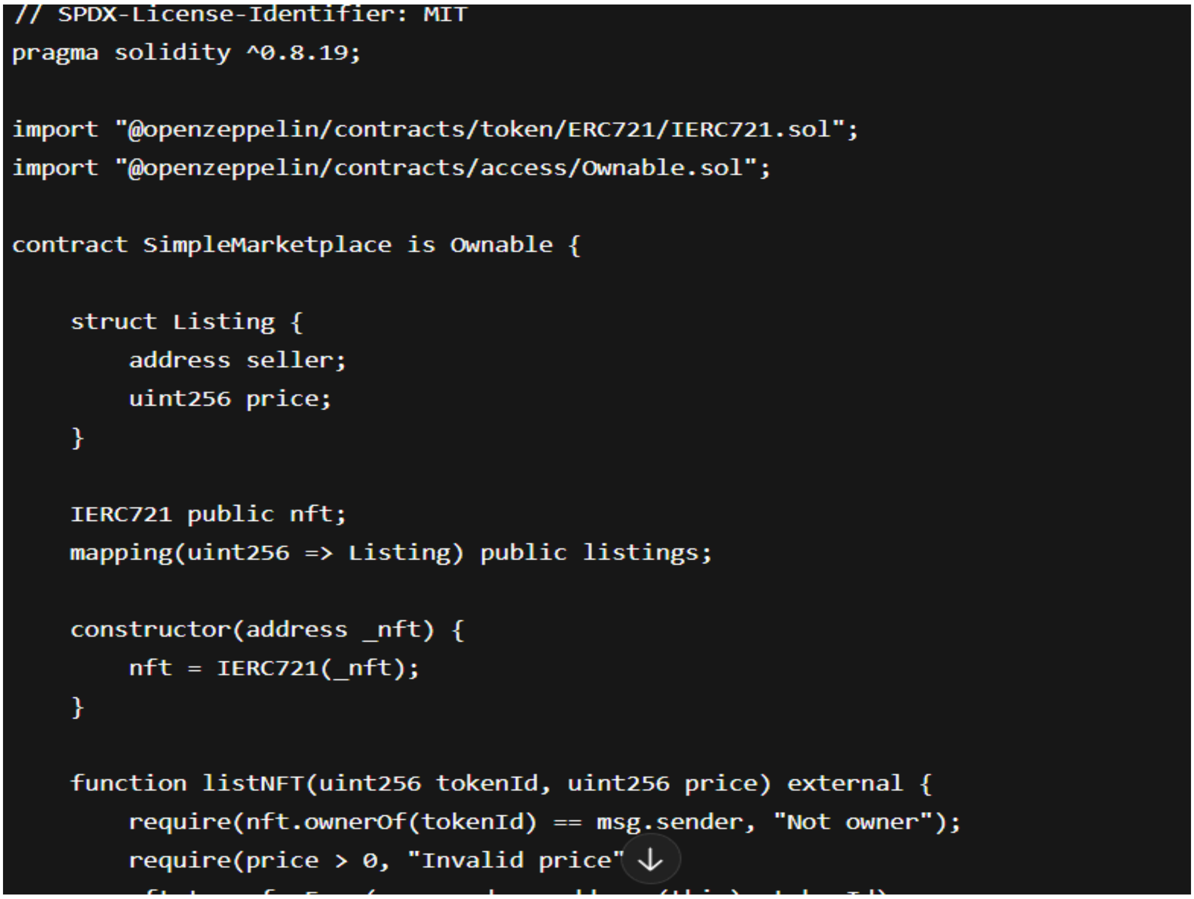
- listNFT(tokenId, price)
- buyNFT(tokenId)
- withdraw() (optional for seller revenue)

Like DeFi protocols in the original file, this lab uses **testnets only** to avoid real-fund exposure.

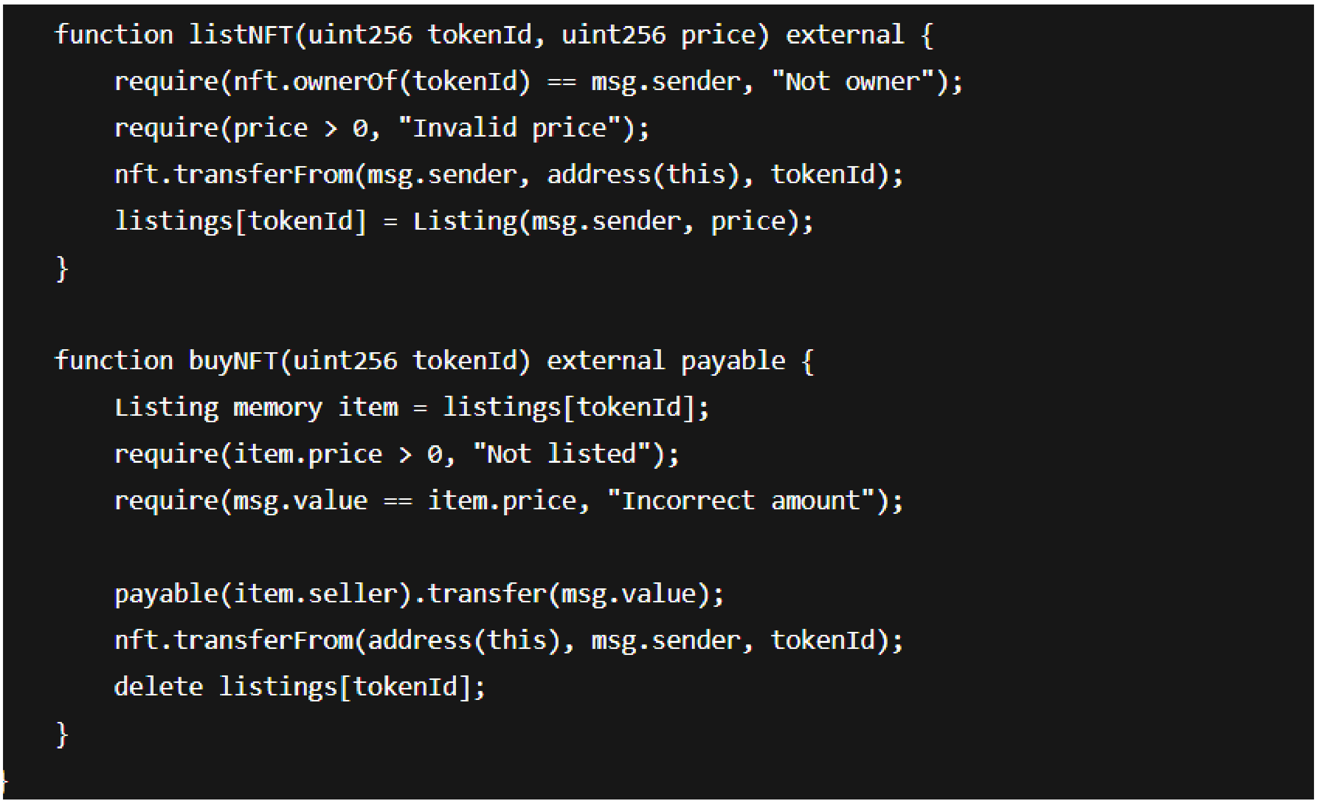
Procedure:

1. **Open MetaMask** and switch to the **Sepolia testnet**.
   (Matches the pattern of Step 1 in the file.)
2. Open **Remix IDE** and create a new file named Marketplace.sol.
3. Copy the following **basic marketplace contract**:

Solidity contract



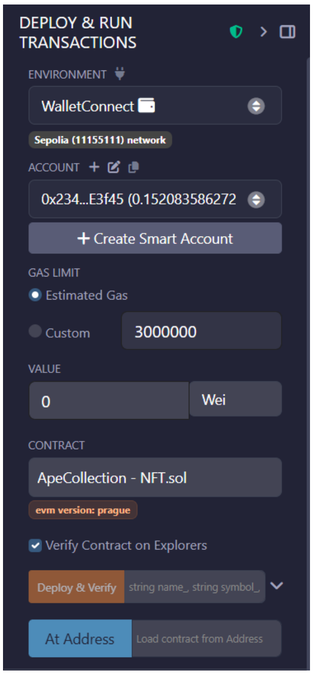
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
import "@openzeppelin/contracts/access/Ownable.sol";

contract SimpleMarketplace is Ownable {

    struct Listing {
        address seller;
        uint256 price;
    }

    IERC721 public nft;
    mapping(uint256 => Listing) public listings;

    constructor(address _nft) {
        nft = IERC721(_nft);
    }

    function listNFT(uint256 tokenId, uint256 price) external {
        require(nft.ownerOf(tokenId) == msg.sender, "Not owner");
        require(price > 0, "Invalid price"
```



```
    function listNFT(uint256 tokenId, uint256 price) external {
        require(nft.ownerOf(tokenId) == msg.sender, "Not owner");
        require(price > 0, "Invalid price");
        nft.transferFrom(msg.sender, address(this), tokenId);
        listings[tokenId] = Listing(msg.sender, price);
    }

    function buyNFT(uint256 tokenId) external payable {
        Listing memory item = listings[tokenId];
        require(item.price > 0, "Not listed");
        require(msg.value == item.price, "Incorrect amount");

        payable(item.seller).transfer(msg.value);
        nft.transferFrom(address(this), msg.sender, tokenId);
        delete listings[tokenId];
    }
}
```

□

- Compile the contract using Solidity compiler version 0.8.x.
- Deploy the contract by providing the NFT contract address as the constructor parameter.
- Once deployed, expand the Deployed Contracts section.
- Call listNFT(tokenId, price) by entering:
- Your NFT's token ID
- A price in wei
- Confirm the transaction in MetaMask.
- To simulate a buyer account, switch accounts in MetaMask.
- Call buyNFT(tokenId) and send the exact test ETH amount.
- Verify that:
- Ownership of NFT transferred to the buyer
- Marketplace contract no longer holds the NFT



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: WalletConnect

Sepolia (11155111) network

ACCOUNT: 0x234...E3f45 (0.152083586272

+ Create Smart Account

GAS LIMIT

● Estimated Gas

○ Custom 3000000

VALUE: 0 Wei

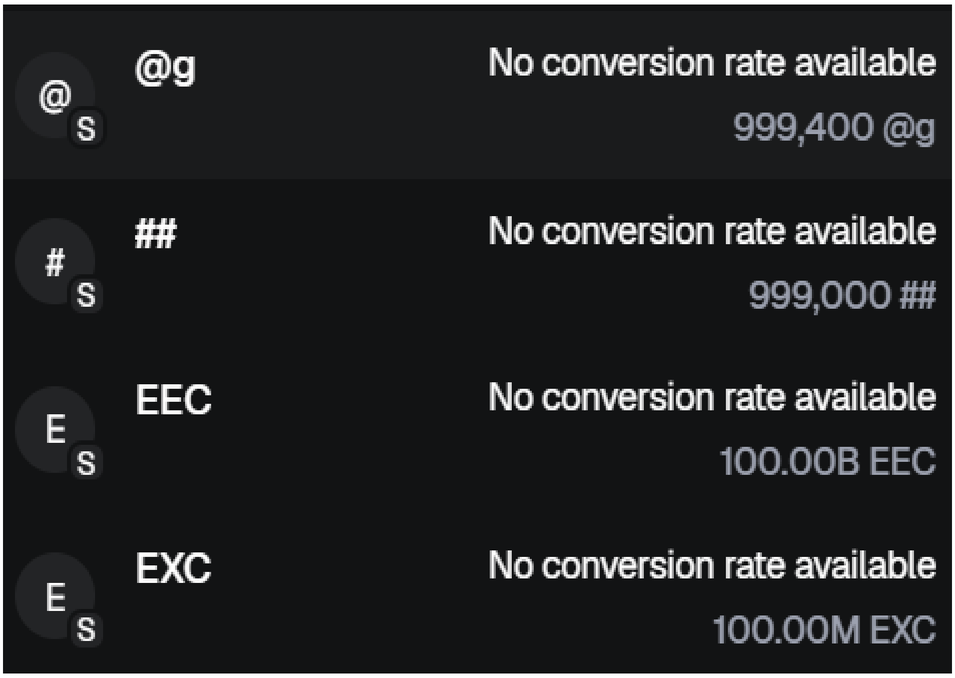
CONTRACT: ApeCollection - NFT.sol

evm version: prague

☑ Verify Contract on Explorers

Deploy & Verify string name_, string symbol_,

At Address Load contract from Address



@g No conversion rate available 999,400 @g

No conversion rate available 999,000

EEC No conversion rate available 100.00B EEC

EXC No conversion rate available 100.00M EXC

Deploy of contract **Token**

Observation

1. MetaMask successfully connected to Sepolia and interacted with the marketplace contract.
2. NFT was listed with the given price and held in the marketplace contract.
3. Buyer account was able to purchase the NFT by sending the exact test ETH amount.
4. Ownership transferred correctly from seller → marketplace → buyer.

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: