**Blockchain Experiment 3**

Shashwat Tripathi

Div: D15A INFT  
Roll No: 64

**AIM: Implement the embedding wallet (Metamask) and transaction using Solidity**

**TASKS PERFORMED:**

Set Up MetaMask: -- Install MetaMask -- Create or Import an Account: -- Fund Your Wallet : Sepolia Testnet (0.5 ETH per day) / RSK Testnet (0.05 RBTC per day)

Connect the Sepolia Testnet / RSK Testnet to Remix IDE

Create a Simple Solidity Smart Contract based on the MiniPoject chosen

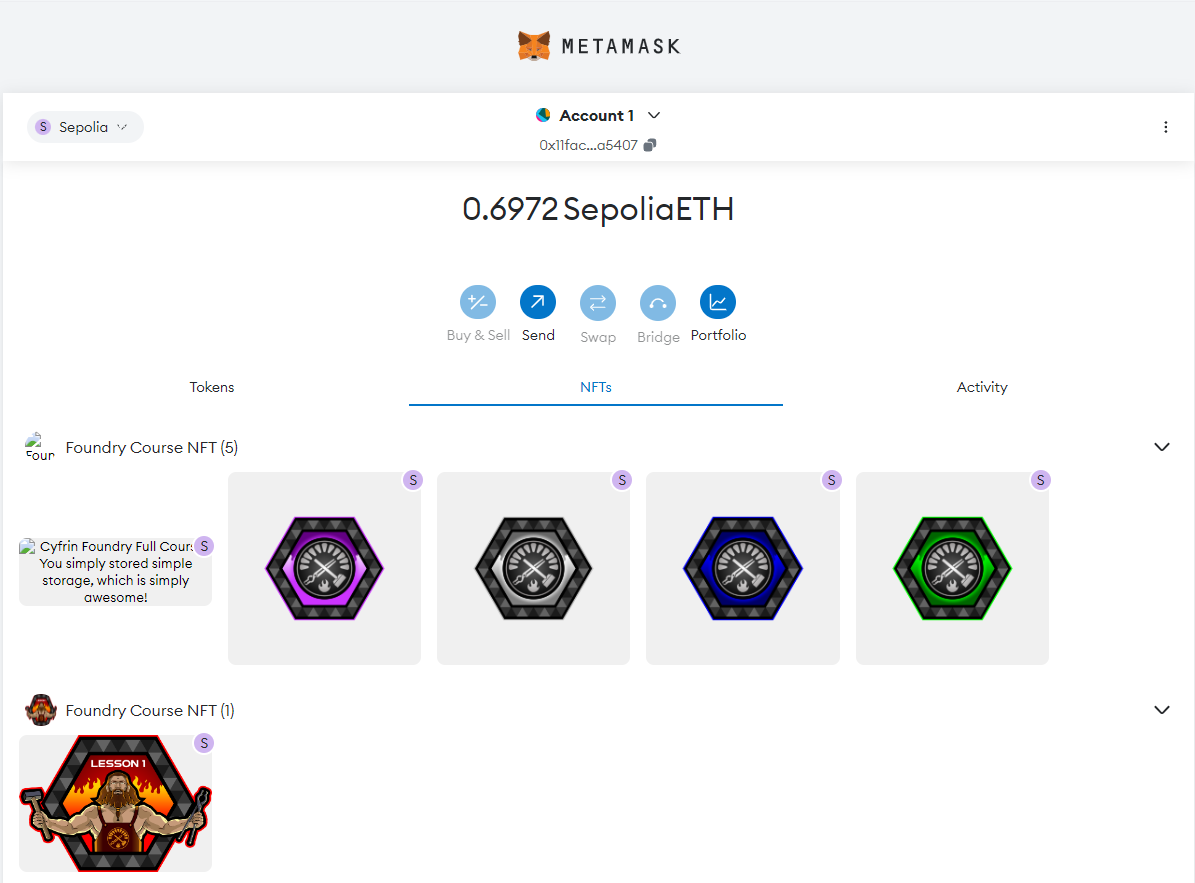
Compile and Deploy the Smart Contract.

Check the transaction details on the RSK Explorer

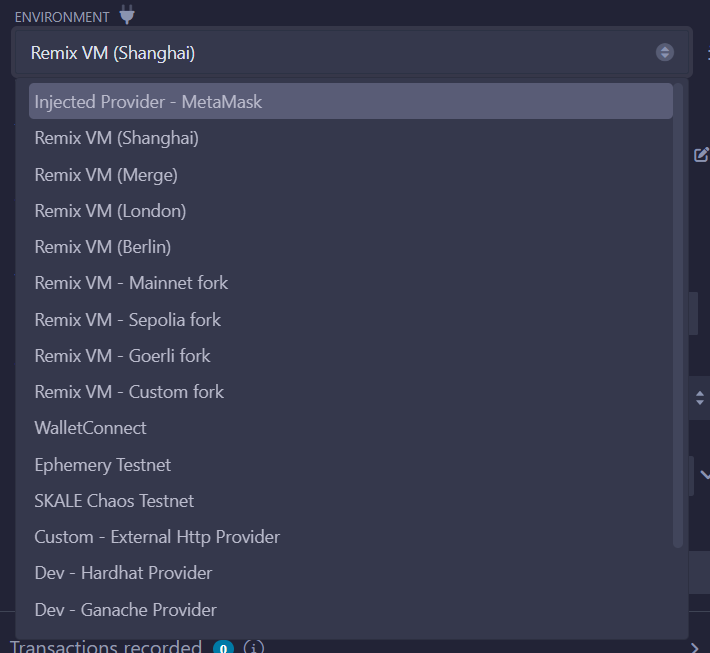
Interact with the smart contract

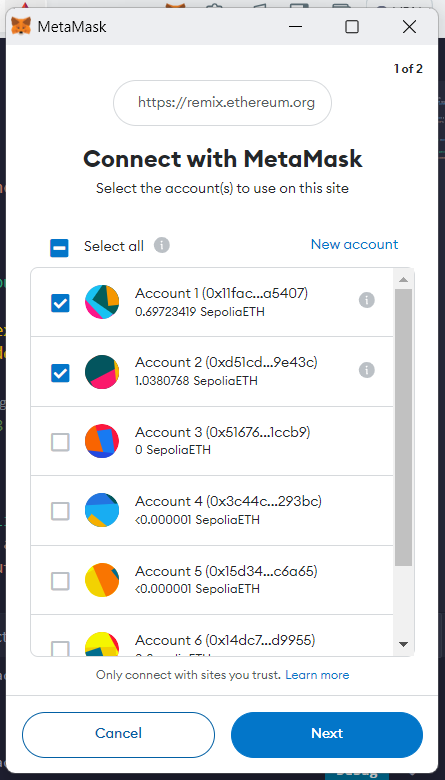
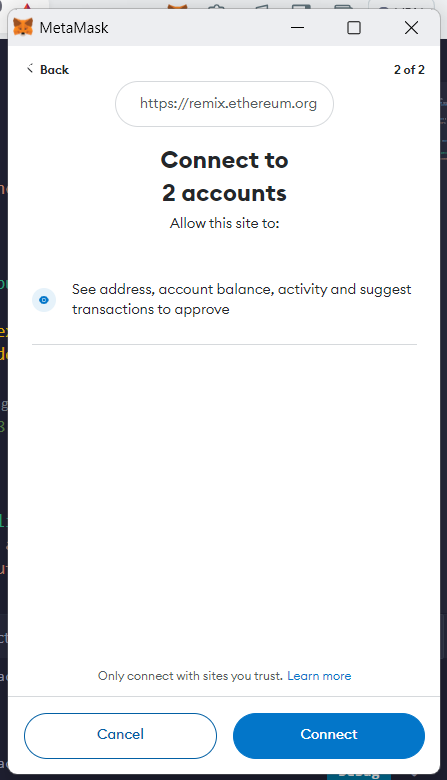
**PROGRAM & OUTPUT :**

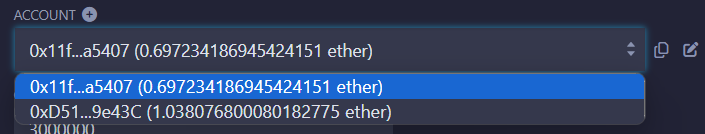
**1. Metamask Account Setup**



**2. Connecting the metamask account to Remix IDE**

****

** **

****

**3. Creating the Solidity Smart Contract**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract CBDC {

string public name = "Central Bank Digital Currency";

string public symbol = "CBDC";

uint256 public totalSupply;

mapping(address => uint256) public balanceOf;

mapping(address => mapping(address => uint256)) public allowance;

event Transfer(address indexed from, address indexed to, uint256 value);

event Approval(address indexed owner, address indexed spender, uint256 value);

constructor(uint256 initialSupply) {

totalSupply = initialSupply \* 10 \*\* 18; // 18 decimals

balanceOf[msg.sender] = totalSupply;

}

function transfer(address to, uint256 value) public returns (bool) {

require(to != address(0), "Invalid recipient address");

require(balanceOf[msg.sender] >= value, "Insufficient balance");

balanceOf[msg.sender] -= value;

balanceOf[to] += value;

emit Transfer(msg.sender, to, value);

return true;

}

function approve(address spender, uint256 value) public returns (bool) {

allowance[msg.sender][spender] = value;

emit Approval(msg.sender, spender, value);

return true;

}

function transferFrom(address from, address to, uint256 value) public returns (bool) {

require(to != address(0), "Invalid recipient address");

require(balanceOf[from] >= value, "Insufficient balance");

require(allowance[from][msg.sender] >= value, "Allowance exceeded");

balanceOf[from] -= value;

balanceOf[to] += value;

allowance[from][msg.sender] -= value;

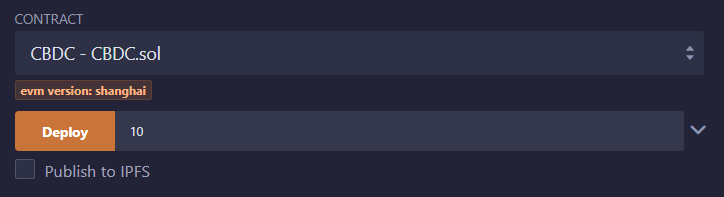
emit Transfer(from, to, value);

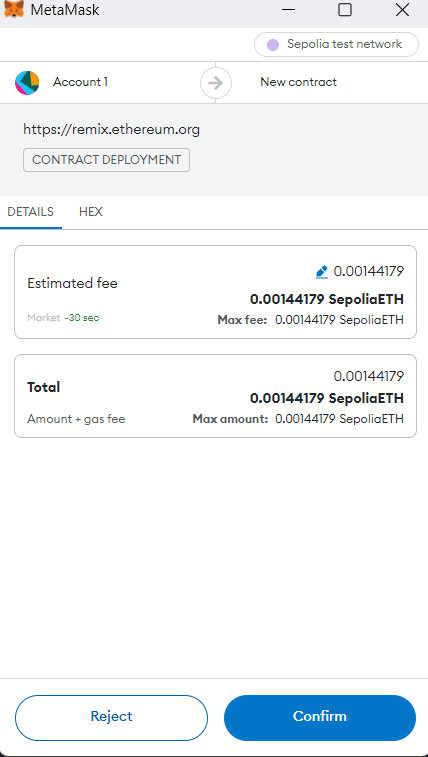
return true;

}

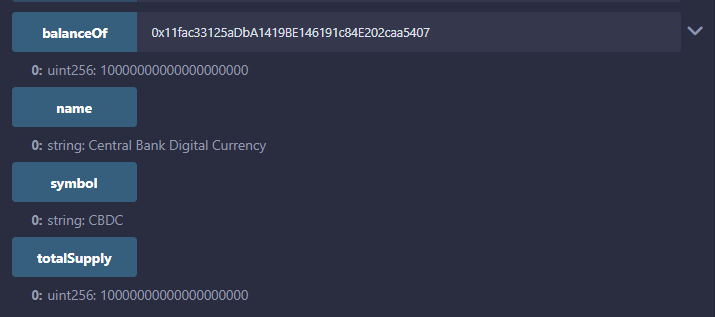
}

**4. Compile and Deploy the smart contract**

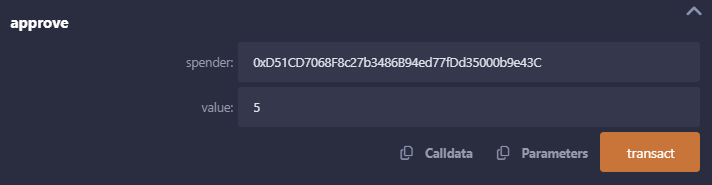




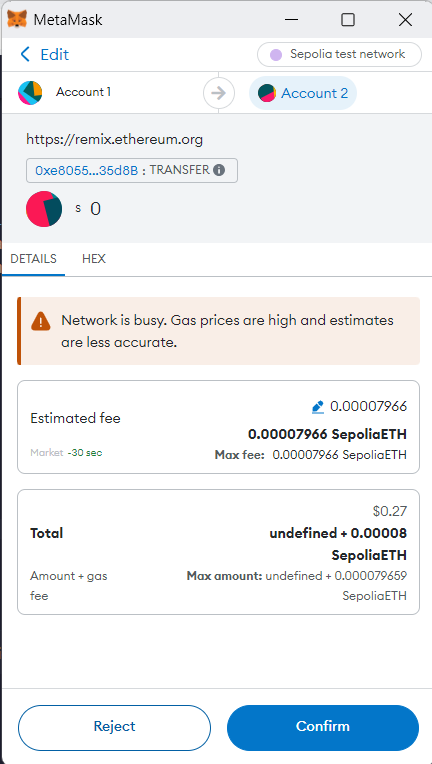
**5. Interacting with the contract**

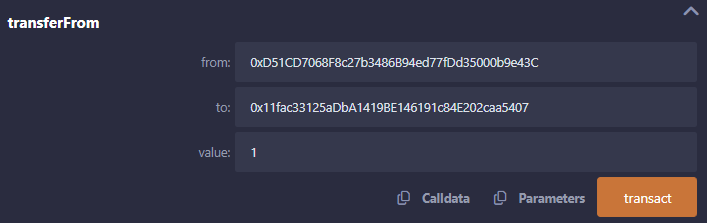


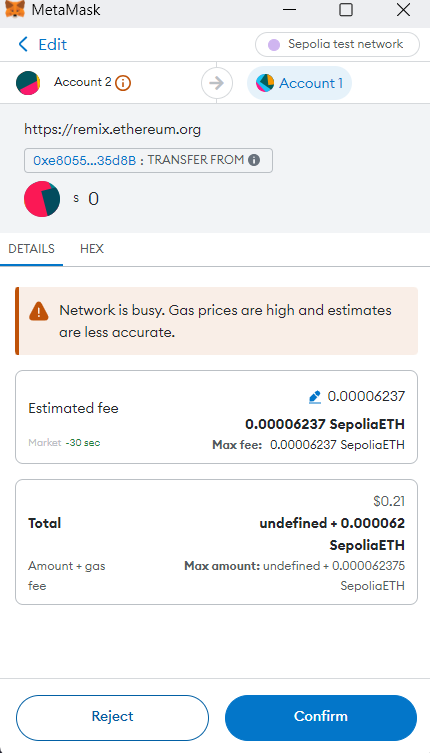
**Approving the below address to transact with tokens**



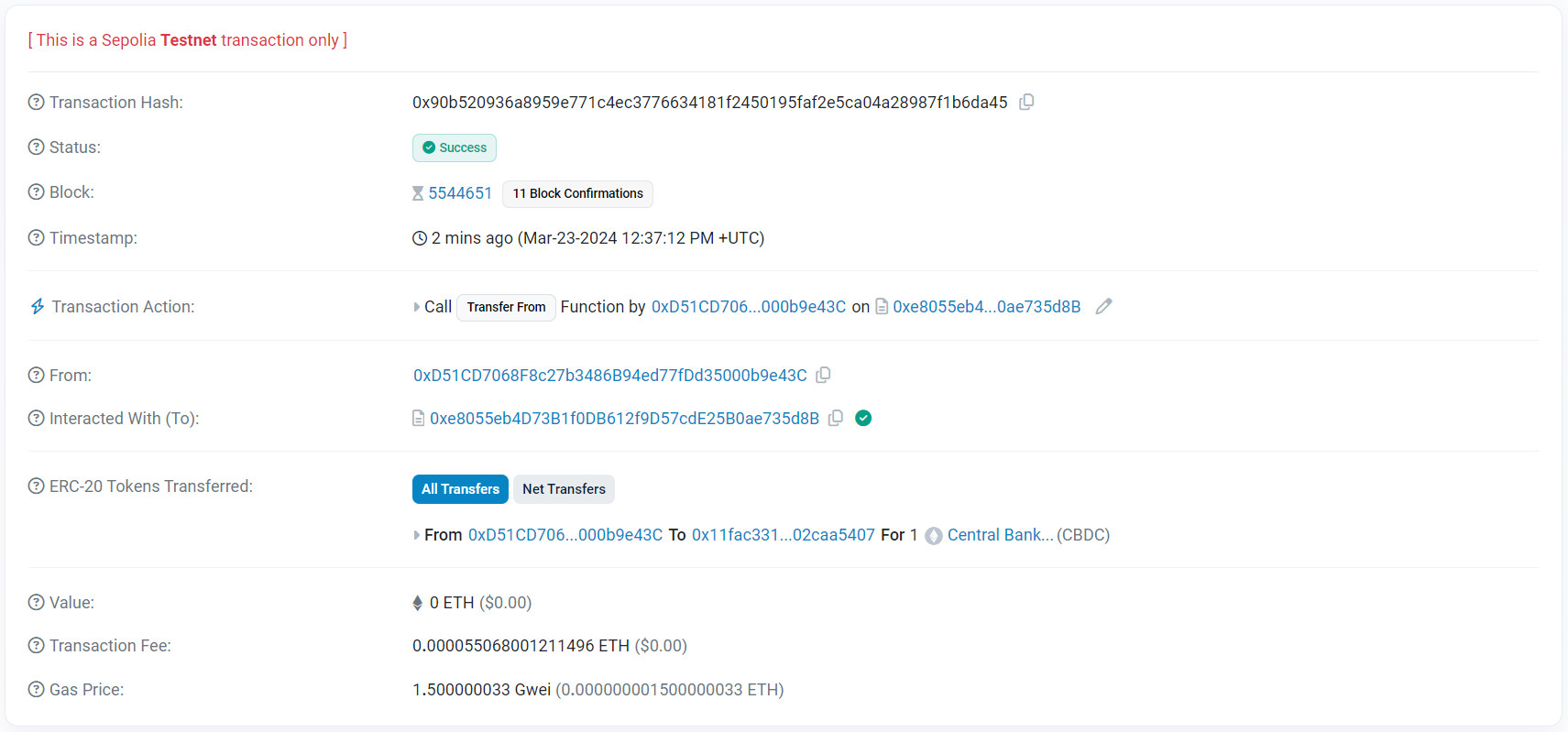
**Signing the txn**







**6. Confirmation on Etherscan Block Explorer**

****

**The transactions can be viewed on etherscan website as well (Sepolia Testnet)**

[**Link to this**](https://sepolia.etherscan.io/tx/0x90b520936a8959e771c4ec3776634181f2450195faf2e5ca04a28987f1b6da45)

**CONCLUSION:**

Understood the steps for embedding the Metamask wallet with Remix IDE and perform transactions

Successfully performed the transactions on the Remix IDE via the account from Metamask Wallet