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 : ERC-20 Basics – Tokenization Concepts

Objective/Aim:

To explore the concept of digital tokenization using Ethereum's ERC-20 standard and to understand how fungible tokens are created, deployed, and interacted with on the Ethereum blockchain using smart contracts.

Apparatus/Software Used:

1. Remix IDE
2. MetaMask Wallet
3. Etherscan
4. OpenZeppelin Library
5. Web Browser/Google Chorme

Theory/Concept:

Tokenization is the process of representing digital or physical assets as blockchain-based tokens. These tokens can symbolize items such as currency, property, shares, or digital collectibles and can be securely transferred on the blockchain.

The **ERC-20 standard (Ethereum Request for Comment 20)** defines a set of functions and rules that ensure interoperability between tokens, wallets, and decentralized applications. This standard simplifies token creation and guarantees consistent behavior across platforms.

Essential Characteristics of ERC-20 Tokens:

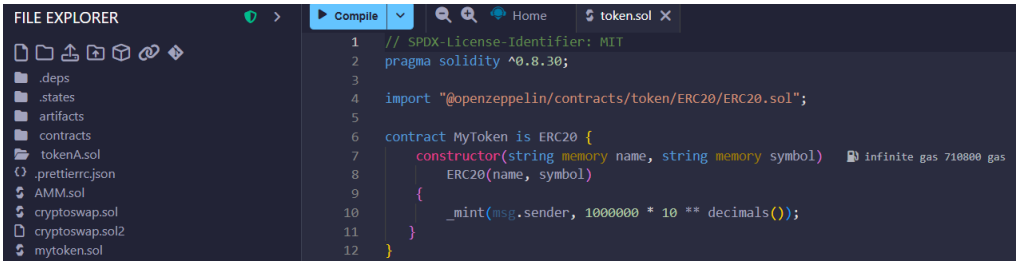
- **Uniformity:** Each token is equal and interchangeable with others.
- **Compatibility:** ERC-20 tokens can be used across wallets and exchanges without modification.
- **Transparency:** All token transactions are permanently recorded on the Ethereum blockchain.
- **Automation:** Smart contracts define and manage the logic of token operations.

Basic ERC-20 Functions:

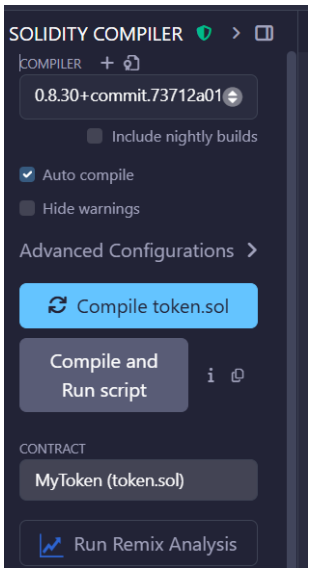
- totalSupply() – Provides total token supply in circulation.
- balanceOf(address) – Displays token balance of a wallet address.
- transfer(address, uint256) – Moves tokens from sender to recipient.
- approve(address, uint256) – Grants spending permission to another address.
- transferFrom(address, address, uint256) – Enables token transfer through an approved spender.
- allowance(address, address) – Shows the amount one address can spend from another's balance.

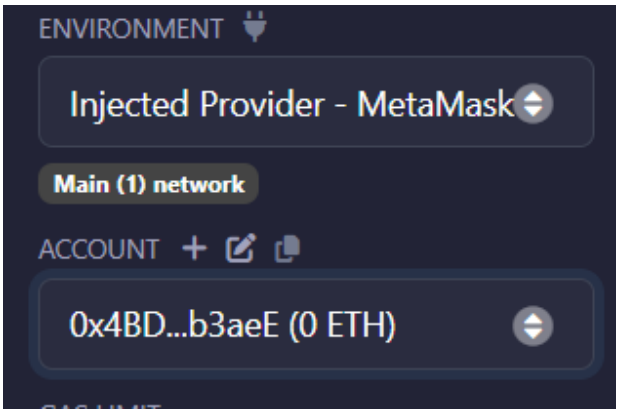
Procedure:

Step 1: Launch **Remix IDE** in a web browser.
Step 2: Create a new Solidity file (e.g., `MyToken.sol`).
Step 3: Import **OpenZeppelin's ERC20** contract template..
Step 4: Write the smart contract by defining the token name, symbol, and initial supply.
Step 5: Compile the code and ensure there are no errors..
Step 6: Connect Remix with **MetaMask** using the **Injected Provider** environment.
Step 7: Deploy the contract on the selected Ethereum test networ
Step 8: Confirm the transaction in MetaMask.
Step 9: Note down the deployed contract address..
Step 10: Note down the deployed contract address.
Step 11: Import the contract address into MetaMask to view your custom token.
Step 12: ***Test token transfer between two wallet addresses.***

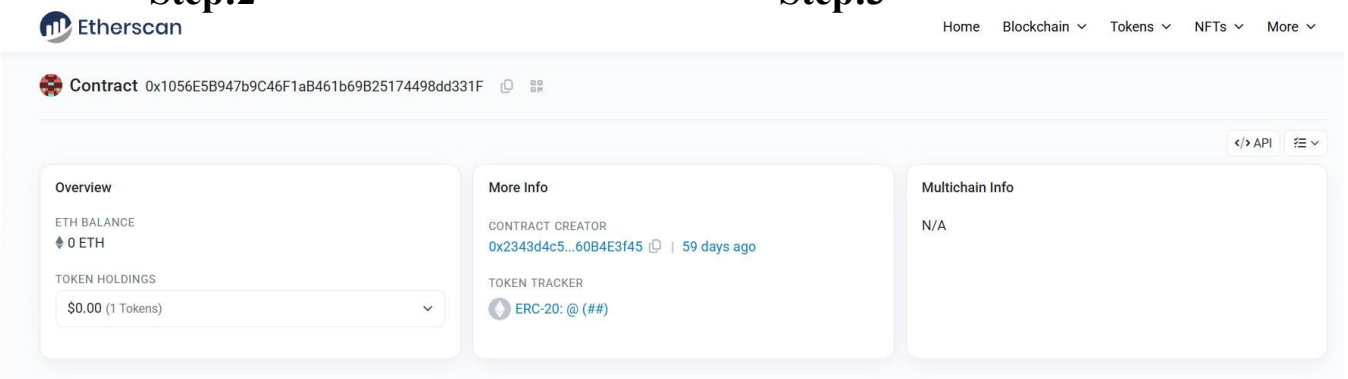


Step:1

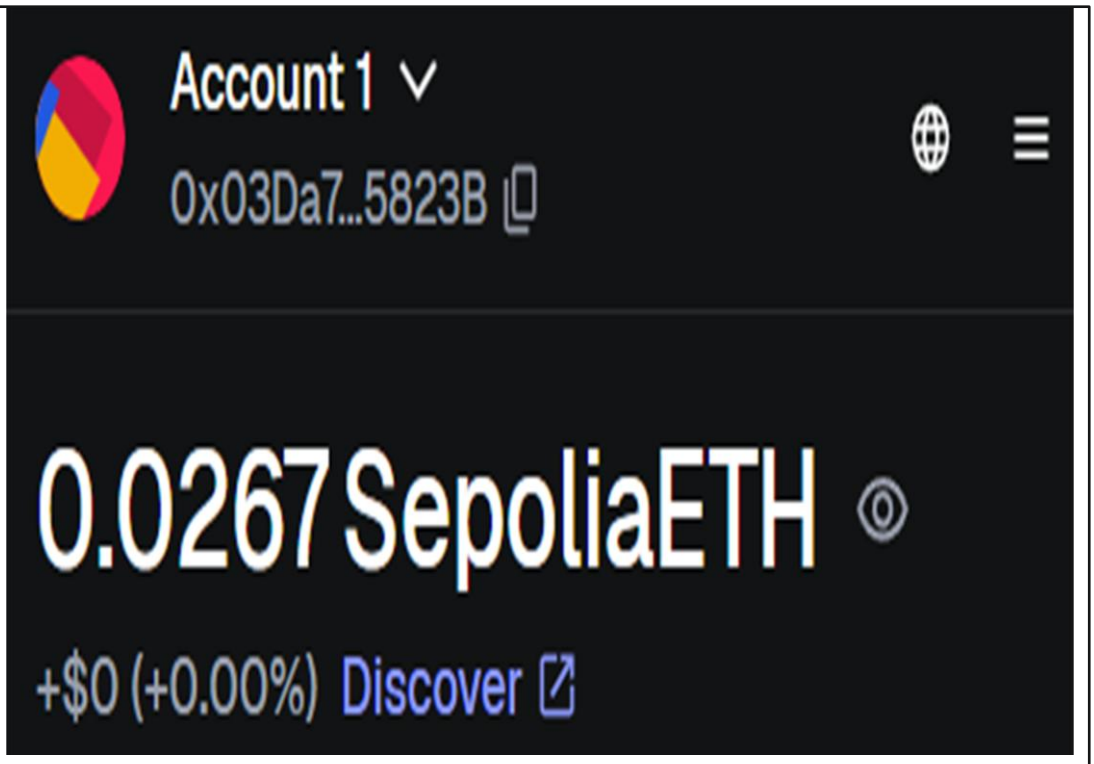




Step:2 **Step:3**



Step:4



Observation Table:

1. The ERC-20 smart contract was successfully compiled and deployed on the Ethereum test network.
2. The created token appeared in the MetaMask wallet after importing the contract address.
3. Token transfer between wallets was successful and visible on Etherscan, confirming proper execution of ERC-20 functions.

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 Faculty:

Signature of the Student:

Name :

Regn. No.