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 DeFi – AMM or Lending Prototype

Objective/Aim:

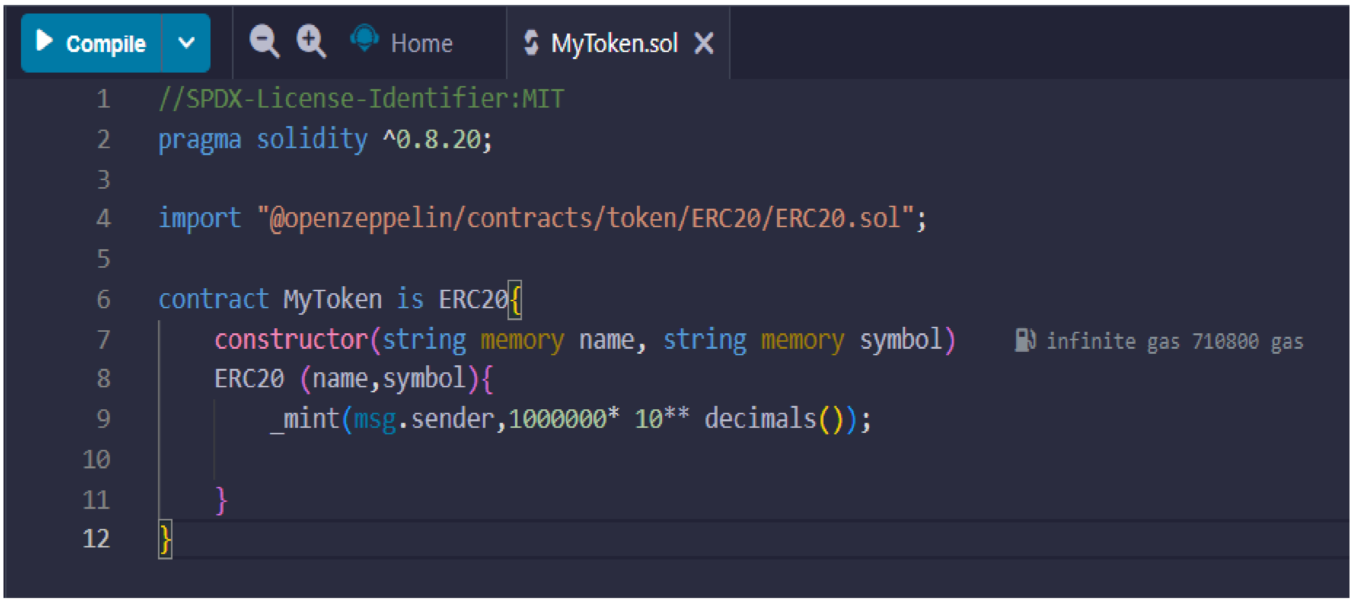
- To understand the working of Decentralized Finance (DeFi) protocols.
- To build and test a simple Automated Market Maker (AMM) or Lending/Borrowing prototype using smart contracts on a blockchain testnet.
- To observe how users can swap tokens.

Apparatus/Software Used:

- Remix IDE (for writing and deploying Solidity smart contracts).
- MetaMask Wallet (for connecting to Ethereum testnet like Sepolia).
- Solidity programming language.
- DeFi protocol code (AMM smart contract or lending smart contract).

Procedure:

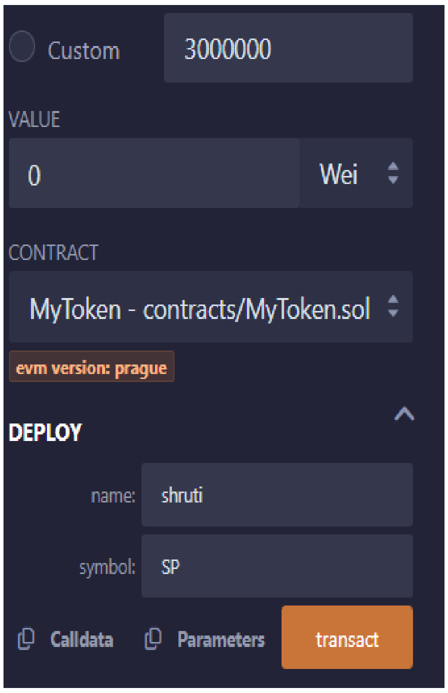
1. Open Remix IDE then write the solidity code to create two tokens and compile it.



```
//SPDX-License-Identifier:MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20{
    constructor(string memory name, string memory symbol)
    ERC20 (name,symbol){
        _mint(msg.sender,1000000* 10** decimals());

    }
}
```

Procedure:

2. After compilation go to the deploye section and write token name and symbol and deploye it and import that deployed token in MetaMask.



Custom 3000000

VALUE

0 Wei

CONTRACT

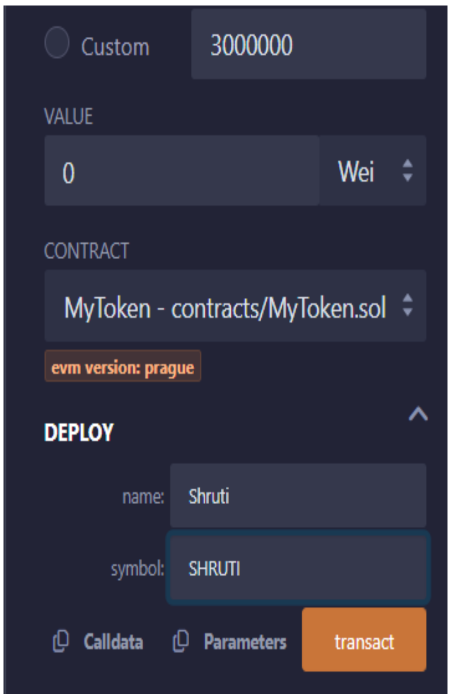
MyToken - contracts/MyToken.sol

evm version: prague

DEPLOY

name: shruti

symbol: SP



Custom 3000000

VALUE

0 Wei

CONTRACT

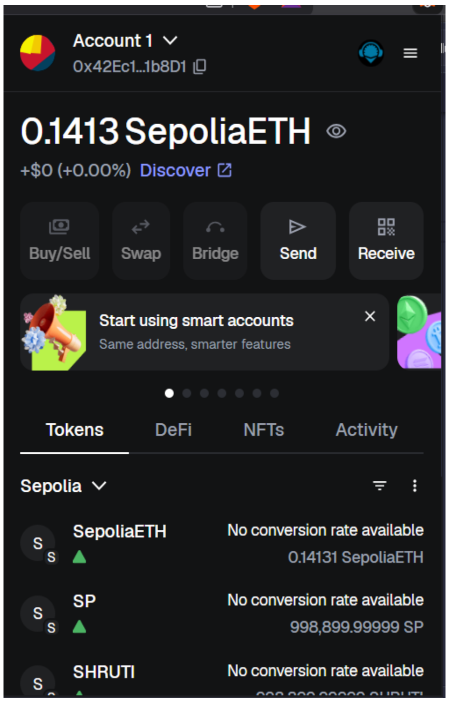
MyToken - contracts/MyToken.sol

evm version: prague

DEPLOY

name: Shruti

symbol: SHRUTI



Account 1
0x42Ec1...1b8D1

0.1413 SepoliaETH

+$0 (+0.00%) Discover

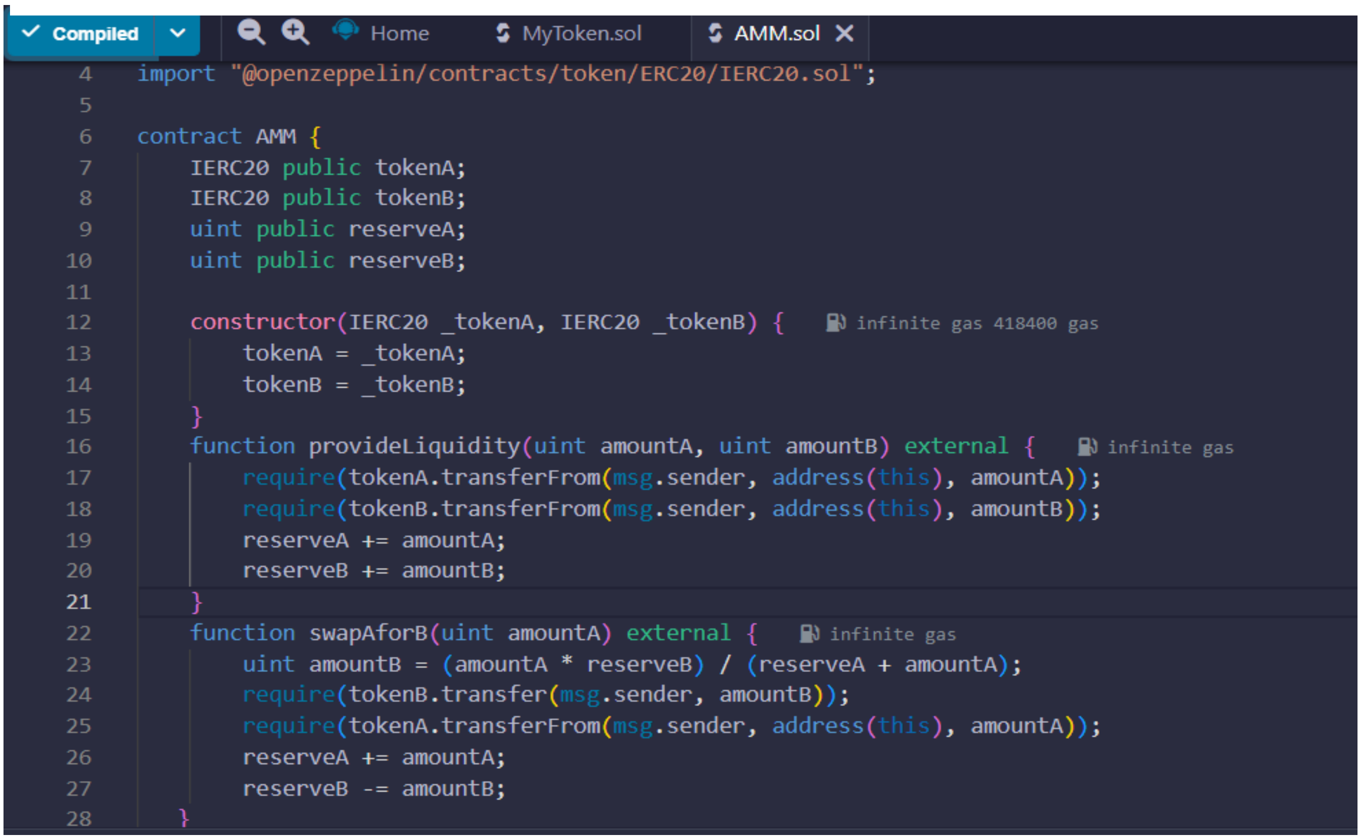
Start using smart accounts
Same address, smarter features

SepoliaETH — No conversion rate available — 0.14131 SepoliaETH

SP — No conversion rate available — 998,899.99999 SP

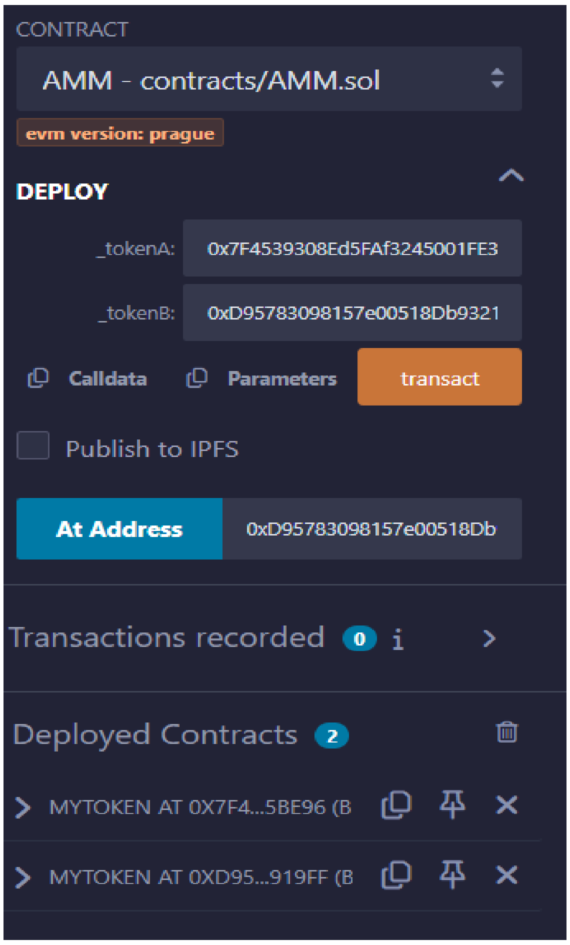
SHRUTI — No conversion rate available

3. Now write the smart contract for AMM and swap token then compile it .

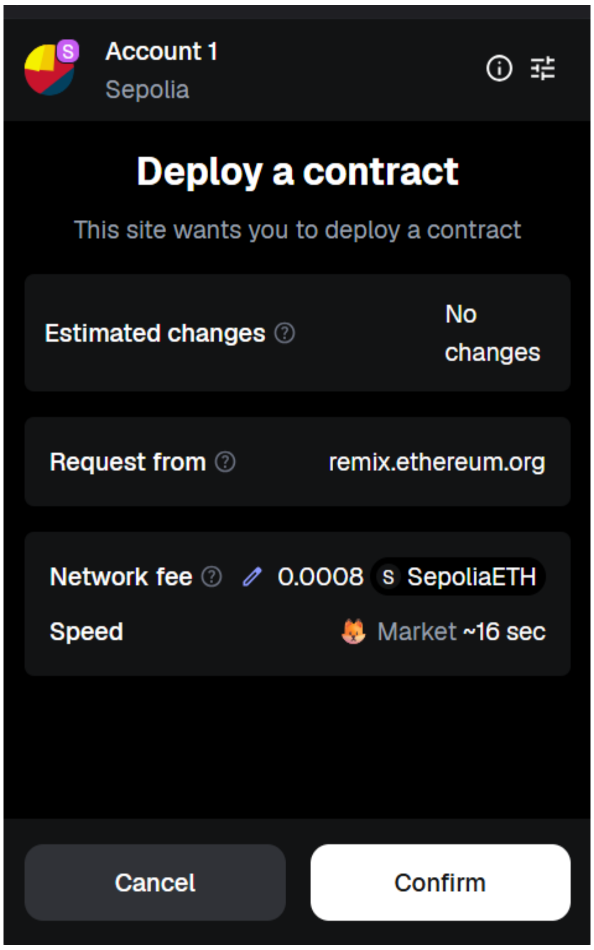


```
import "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract AMM {
    IERC20 public tokenA;
    IERC20 public tokenB;
    uint public reserveA;
    uint public reserveB;

    constructor(IERC20 _tokenA, IERC20 _tokenB) {    infinite gas 418400 gas
        tokenA = _tokenA;
        tokenB = _tokenB;
    }
    function provideLiquidity(uint amountA, uint amountB) external {    infinite gas
        require(tokenA.transferFrom(msg.sender, address(this), amountA));
        require(tokenB.transferFrom(msg.sender, address(this), amountB));
        reserveA += amountA;
        reserveB += amountB;
    }
    function swapAforB(uint amountA) external {    infinite gas
        uint amountB = (amountA * reserveB) / (reserveA + amountA);
        require(tokenB.transfer(msg.sender, amountB));
        require(tokenA.transferFrom(msg.sender, address(this), amountA));
        reserveA += amountA;
        reserveB -= amountB;
    }
```

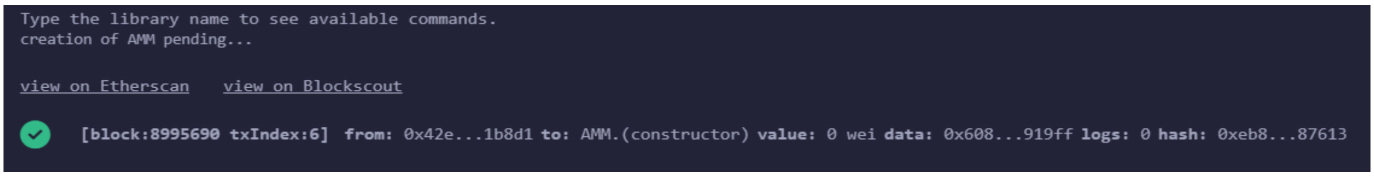
4. After compilation go to the deploye section and and give the contract address of that two token we previously created and click on transact and confirm transaction on MetaMask .Now our smart contract is successfully deployed.



CONTRACT
AMM - contracts/AMM.sol
evm version: prague
DEPLOY
_tokenA: 0x7F4539308Ed5FAf3245001FE3
_tokenB: 0xD95783098157e00518Db9321
Calldata Parameters transact
Publish to IPFS
At Address 0xD95783098157e00518Db
Transactions recorded 0
Deployed Contracts 2
MYTOKEN AT 0X7F4...5BE96 (B
MYTOKEN AT 0XD95...919FF (B



Account 1
Sepolia
Deploy a contract
This site wants you to deploy a contract
Estimated changes: No changes
Request from: remix.ethereum.org
Network fee: 0.0008 SepoliaETH
Speed: Market ~16 sec
Cancel Confirm



```
Type the library name to see available commands.
creation of AMM pending...

view on Etherscan   view on Blockscout

✓  [block:8995690 txIndex:6]  from: 0x42e...1b8d1 to: AMM.(constructor) value: 0 wei data: 0x608...919ff logs: 0 hash: 0xeb8...87613
```

5. The AMM smart contract correctly handled liquidity addition for both tokens and Swap transactions were executed successfully, and token balances updated as expected. All transactions were confirmed on the Ethereum test network without errors.

Observation :

1.The ERC-20 tokens (TokenA and TokenB) were successfully deployed and visible in MetaMask.
2.The AMM smart contract correctly handled liquidity addition for both tokens. 3.Swap transactions were executed successfully, and token balances updated as expected.
4.All transactions were confirmed on the Ethereum test network without errors

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 :

Signature of the Faculty:

Regn. No. :

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