

Hello Solidity – Writing First Smart Contract

School: ............................................................................................................. Campus: ....................................................... Academic Year: ...................... Subject Name: ........................................................... Subject Code: ..........................

Semester: ............... Program: ........................................ Branch: ......................... Specialization: .......................... Date: .....................................

(Learning by Doing and Discovery)

**\* Coding Phase: Pseudo Code / Flow Chart / Algorithm**

ALGORITHM:

1. Start
2. Install the MetaMask extension in your browser and set up a wallet.
3. Connect to a test network (e.g., Sepolia) and collect test ETH from a faucet.
4. Open Remix IDE in your browser.
5. Create a new file named SimpleStorage.sol.
6. Write a simple contract in Solidity:
7. Write your code
8. Compile the contract using the Solidity Compiler in Remix.
9. In the Deploy tab, select Injected Provider - MetaMask and connect.
10. Click Deploy and confirm the transaction in MetaMask.
11. After deployment, click the greet() function to view the output. 12.End

# \* Software Used:

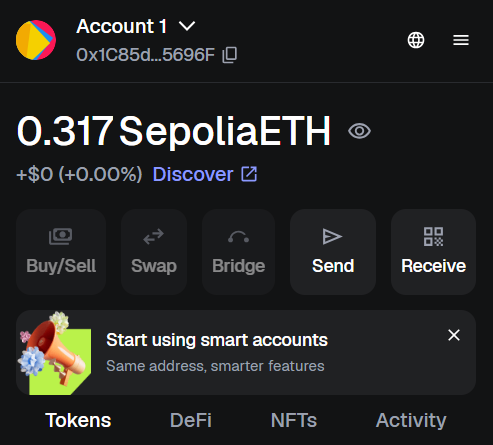
1. Metamask Wallet
2. Remix IDE
3. Ethereum sepolia Faucet

1. **Setting up MetaMask and Test Network**

* Download and install the MetaMask extension in the Brave browser.
* Create or import a wallet in MetaMask.
* Connect MetaMask with Remix IDE to enable contract deployment.
* Visit an Ethereum faucet (testnet faucet) to request free test Ether.
* Verify that the test Ether is credited to your MetaMask account.

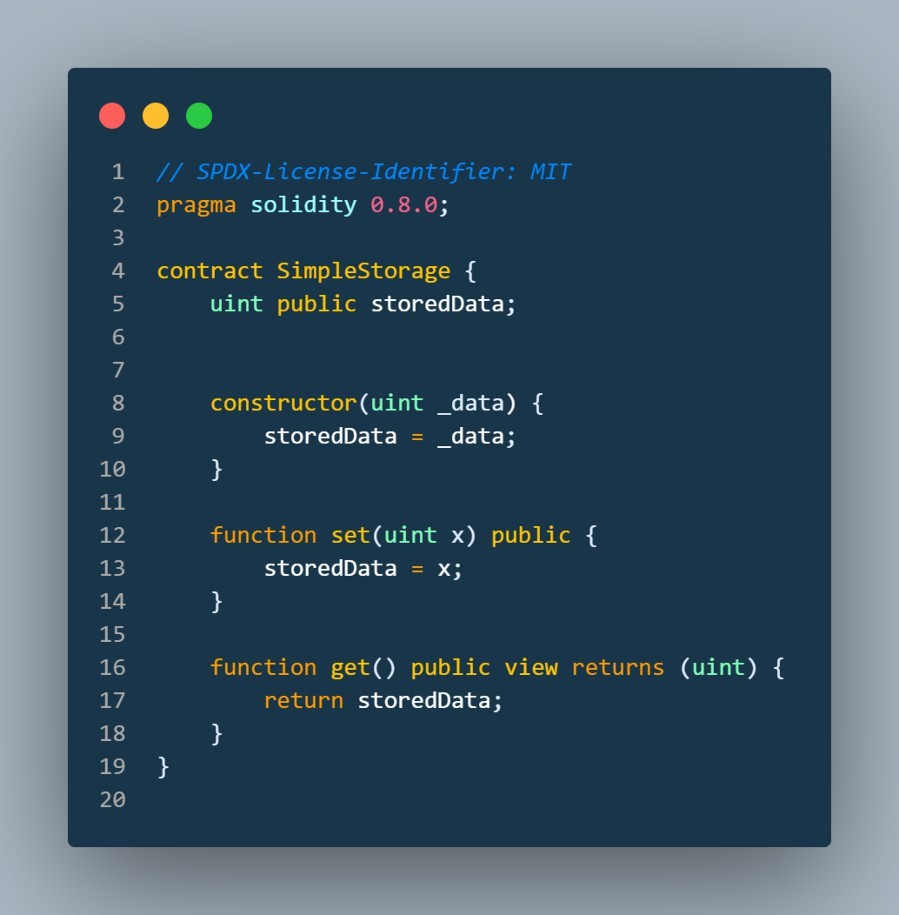


* Give your wallet address to receive 0.05 faucet to our Sepolia Account.

.

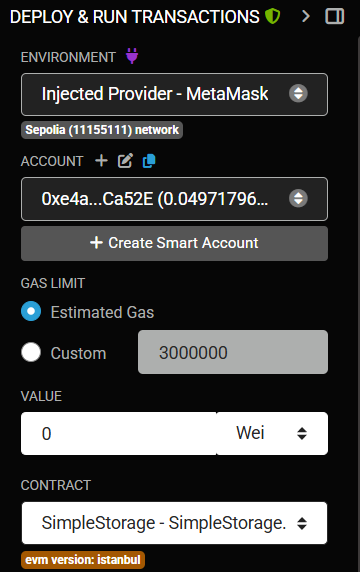
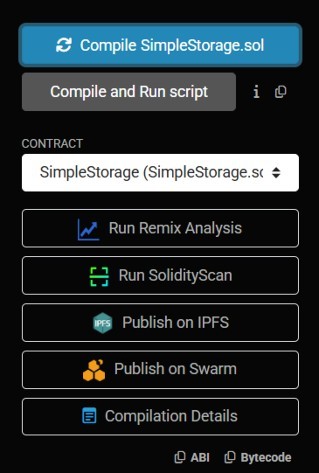
1. **Creating a Smart Contract File:**

* Open the Remix IDE in the Brave browser and create a new Solidity file named SimpleStorage.sol. Write the required smart contract code in this file, which will later be compiled and deployed.

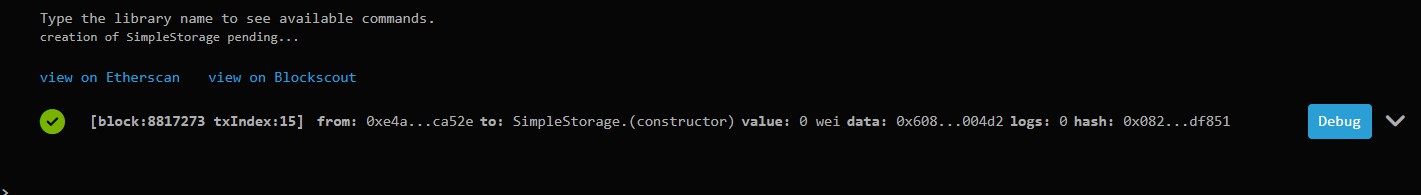


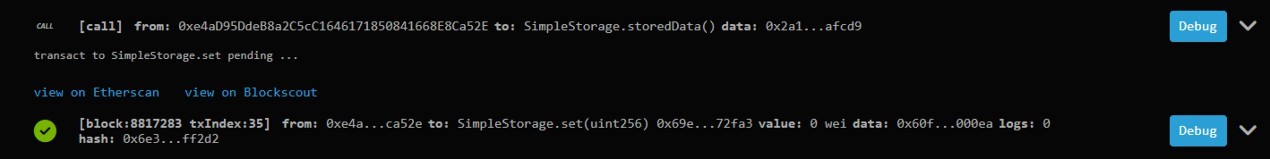
1. **Writing and Deploying the Smart Contract:**

* In the smart contract code, define two functions: **set()** to store data and **get()** to retrieve the stored data.
* After completing the code, navigate to the **Environment** settings in Remix IDE and select **Injected Provider – MetaMask**.
* The MetaMask wallet address, along with the available test balance, will be automatically detected and displayed.
* Then click on compile after clicking our file has successfully compiled

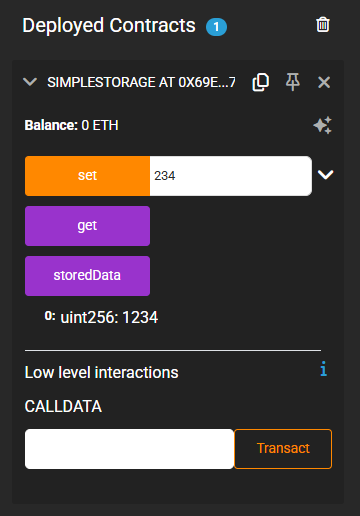


1. Then we click on the compile after clicking our fie has successfully Compiled in RemidIDE





1. Then Click on the deploy, then it will be deployed:



## \* Implementation Phase: Final Output (no error)

Applied and Action Learning



# Observation:

1. The smart contract was compiled successfully without any errors in Remix IDE.
2. Deployment was carried out on the **Ethereum test network** using MetaMask as the wallet interface.
3. MetaMask processed the transaction and confirmed it on the blockchain.
4. The deployed contract address was generated, which can be verified on **Etherscan** along with the transaction details.



|  |  |  |  |
| --- | --- | --- | --- |
| **Rubrics** |  |  |  |
| 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:***



***Signature of the Faculty:***