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 Experiment: Hello Solidity – Writing First Smart Contract

*Coding Phase: Pseudo Code / Flow Chart / Algorithm

Step 1: First we need to start the development environment by opening a web browser, then go to the Remix IDE :https://remix.ethereum.org and select the "Solidity" environment.

Step 2: Then we need to create a new solidity file by clicking on the file explorer and click "create new file" and name the file as SimpleStorage.sol

Step 3: Then we need to write the smart contract

Step 4: After writing the contract we need to compile the contract

Step 5: After successful compilation we need to deploy the contract by setting up the environment to Injected Provider-MetaMask

Step 6: After deploying the transaction, MetaMask will show a popup.

Click **"Confirm"** to broadcast the transaction to the Sepolia testnet.

Step 7 **:** Open MetaMask → Click the "Activity" tab.

Find the latest transaction where we will find all the information regarding transactions along with id and we can check by clicking on "Blockchain Explorer"

Step 8 : In the deployed contract section we can use set and get function to enable the effect in the metamask wallet.

Step 9: End

*Software Used

- Laptop
- Remix IDE
- MetaMask browser extension
- Sepolia Etherscan explorer

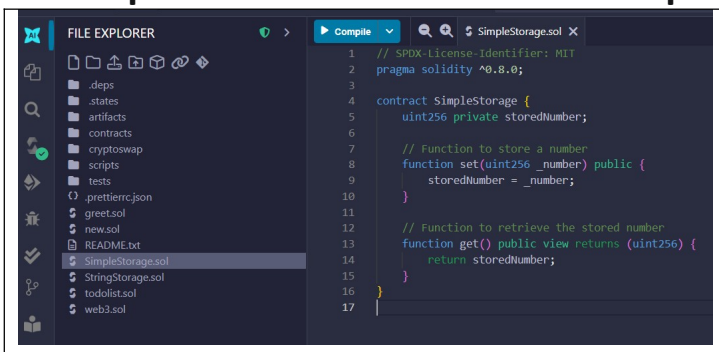
**** As applicable according to the experiment.***

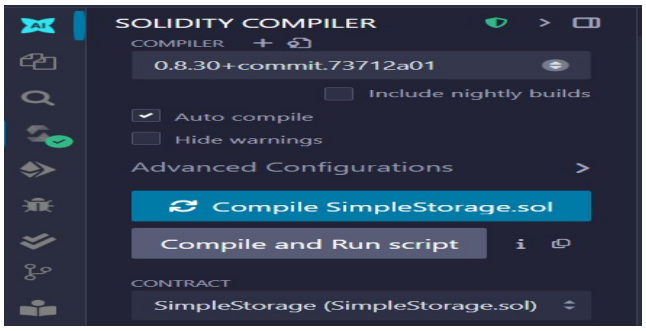
Two sheets per experiment (10-20) to be used.

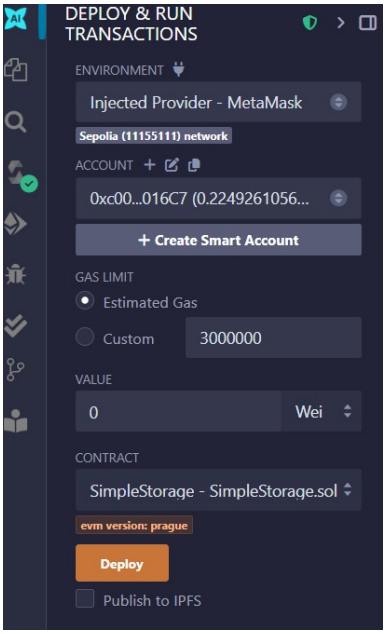
* Testing Phase: Compilation of Code (error detection)

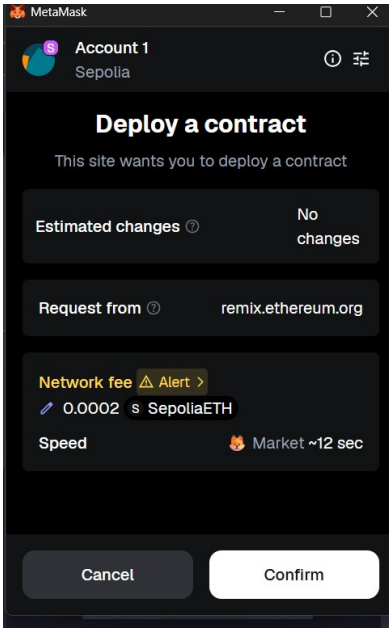
No error

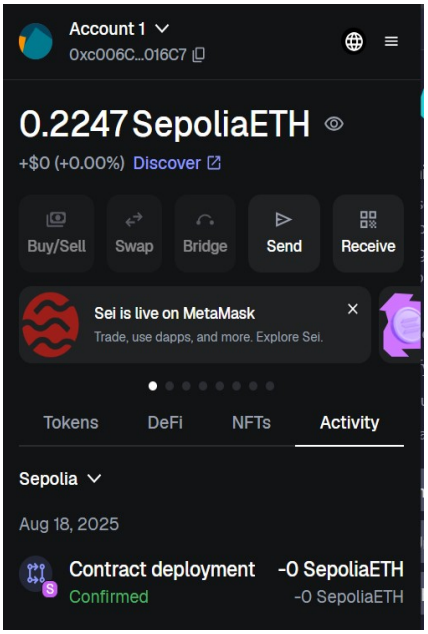
* Implementation Phase: Final Output (no error)

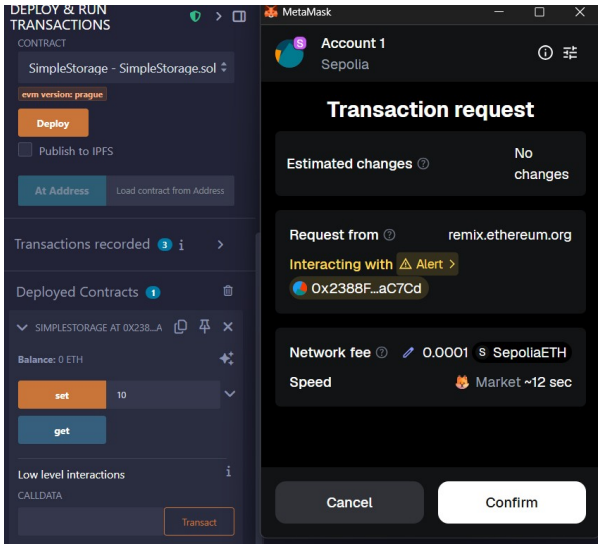


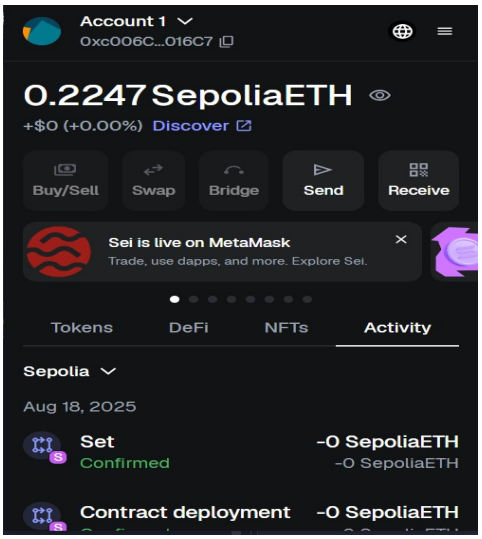












**** As applicable according to the experiment.***
Two sheets per experiment (10-20) to be used.

* Observations

The contract successfully demonstrates the core functionality of a smart contract — storing and retrieving a single unsigned integer (uint256) on the Ethereum blockchain.

The set() function updates the stored value, showing how state changes are handled in Solidity and how these changes require gas when executed.

The get() function allows users to retrieve the stored value, illustrating the use of view functions that do not cost gas and do not modify contract state.

Testing in Remix IDE was effective for understanding deployment and function interaction.

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