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 :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Deploying Your First Smart Contract in Solidity on Remix IDE

1. **Compiling the Code**
   - Remix automatically converts your Solidity contract into bytecode and ABI, so you don't need extra tools.
   - It also checks that the code has no syntax errors and that you're using the correct compiler version.
   - If the code has mistakes or the version doesn't match, deployment won't go through.
2. **Connecting for Deployment**
   - MetaMask works as the link between Remix and the Ethereum blockchain (testnet or mainnet).
   - When you deploy, MetaMask asks for confirmation and shows:
     - The wallet address that's deploying the contract
     - Estimated gas fees
     - The network being used
   - This step shows that deployment happens via a blockchain transaction, not just by uploading the file.
3. **Processing the Transaction**
   - You receive a transaction hash that you can follow on a block explorer like Etherscan or Sepolia Explorer.
   - After confirmation, your contract is assigned a permanent address on the blockchain.
   - Gas fees are deducted from your MetaMask account, proving that deploying contracts has a cost (even on testnets, it simulates real value).
4. **Using the Contract**
   - Once live, Remix provides a built-in interface to interact with the contract functions.
   - Calling functions that change data again needs MetaMask approval.
   - You can see updates to the data or events directly in the Remix console.

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

Coding Phase: Pseudo Code / Flow Chart / Algorithm

Steps to Write and Deploy Your First Solidity Contract in Remix

Step 1: Open Remix IDE

- Go to https://remix.ethereum.org in your browser.

Step 2: Create a New File

- On the left panel, open the File Explorer.
- Inside the contracts folder, click the + (new file) button.
- Give your file a name, for example: SimpleStorage.sol.

Step 3: Write the Smart Contract

```
pragma solidity ^0.8.0;
contract SimpleStorage{
    uint public storedData;
    constructor(uint data){
        storedData = data;
    }
    function set(uint x)public {
        storedData = x;
    }
    function get() public view returns (uint) {
        return storedData;
    }
}
```

Step 4: Compile the Contract

- Click on the Solidity Compiler tab (the icon with a “tick mark”).
- Hit the Compile button.

Step 5: Deploy the Contract

- Switch to the Deploy & Run Transactions tab (Ethereum logo with play button).
- Under Environment, choose Injected Provider – MetaMask to connect your wallet. Click Deploy, and MetaMask will pop up to confirm the transaction.

Step 6: Interact with the Contract

Once deployed, you’ll see your contract under Deployed Contracts

Softwares used

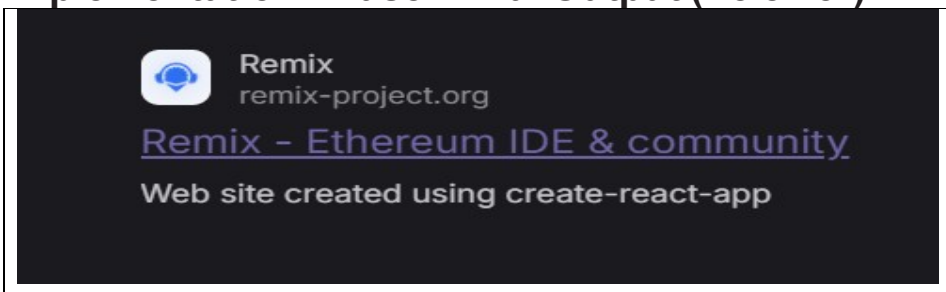
- *Remix IDE*
- MetaMask Wallet

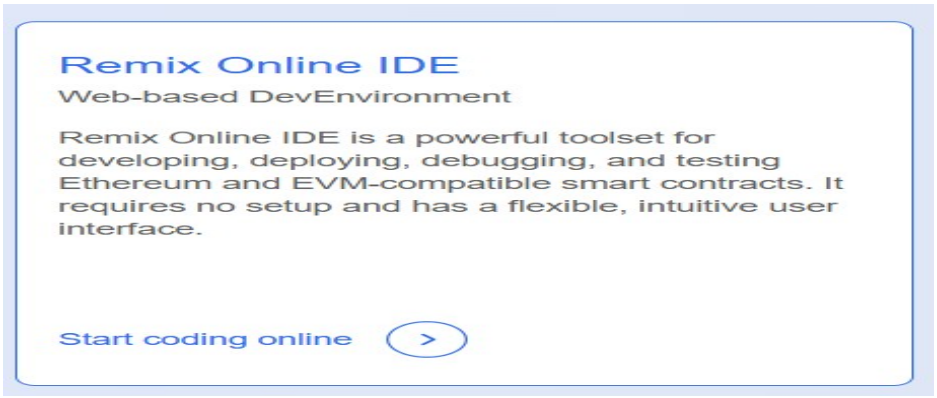
* Testing Phase: Compilation of Code (error detection)

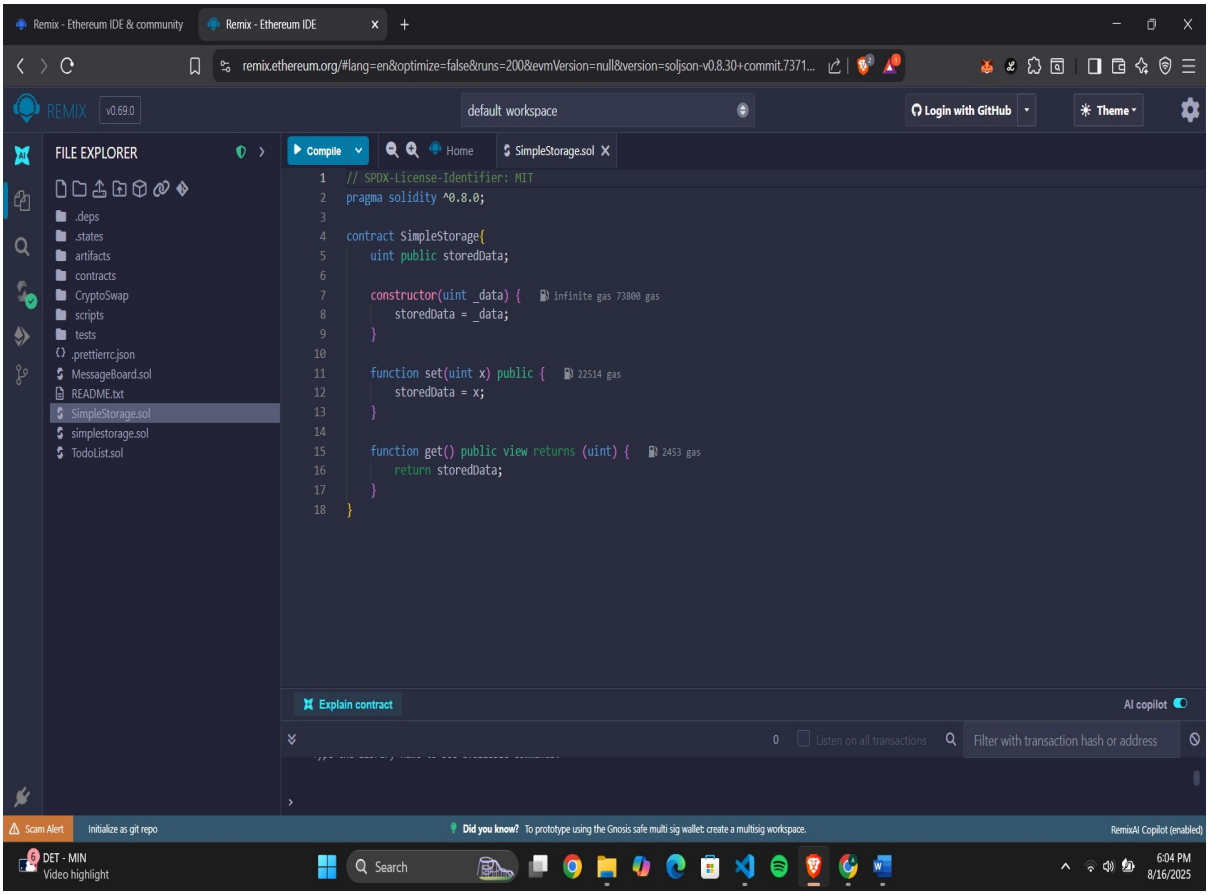
No Error

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

* Implementation Phase: Final Output (no error)

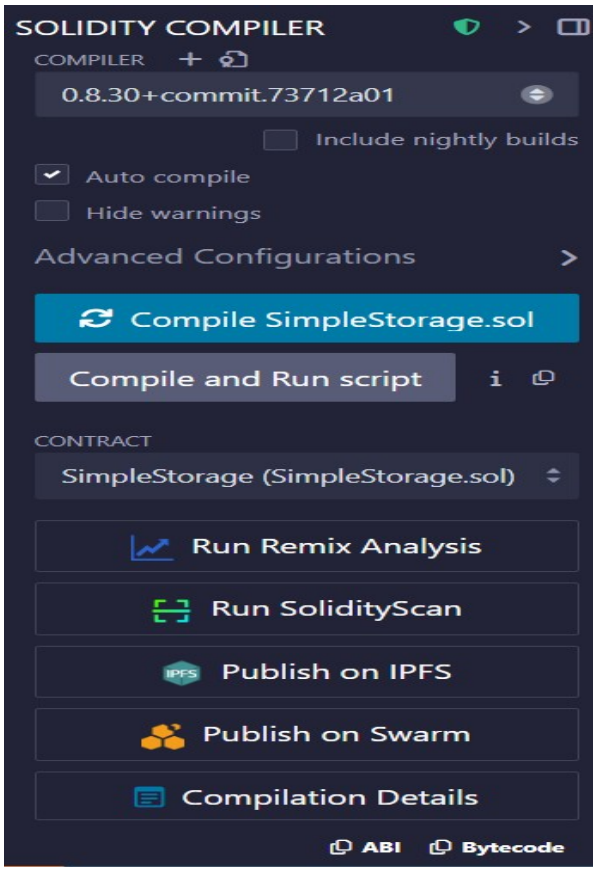


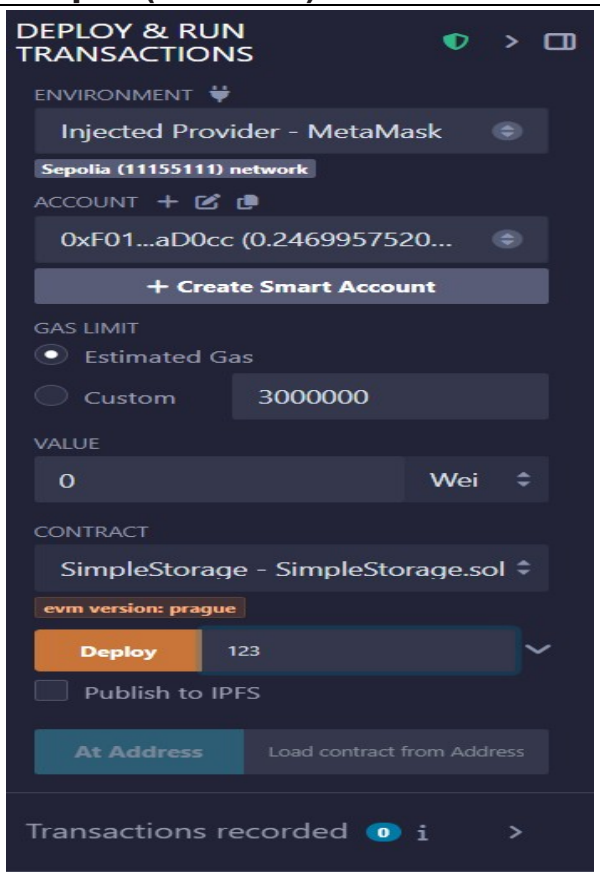


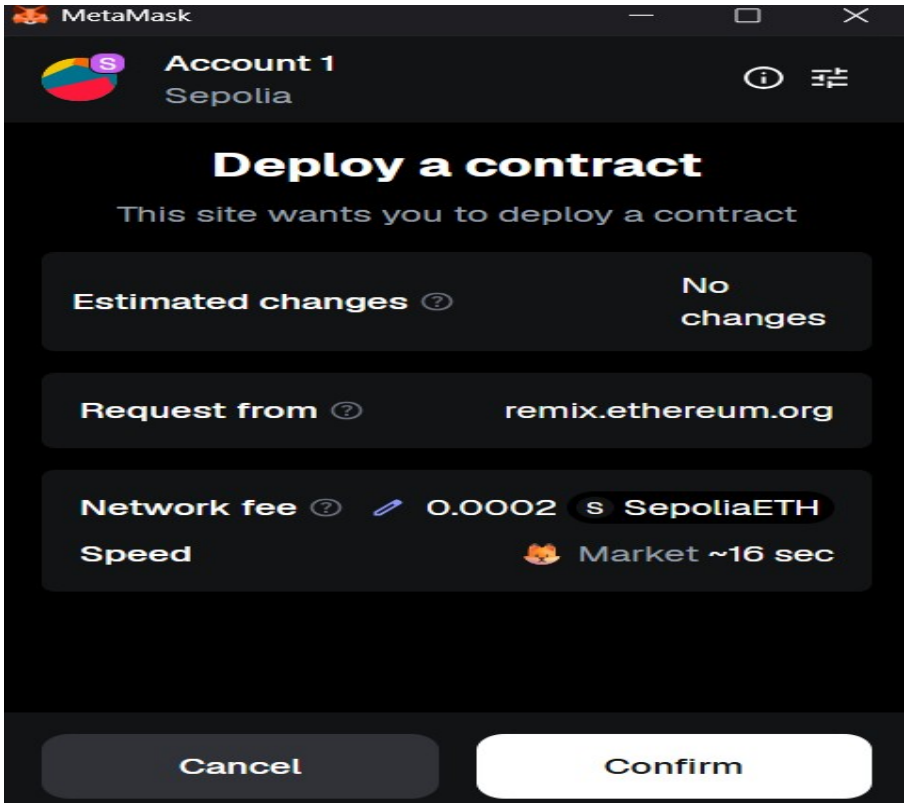


*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

Implementation Phase: Final Output (no error)

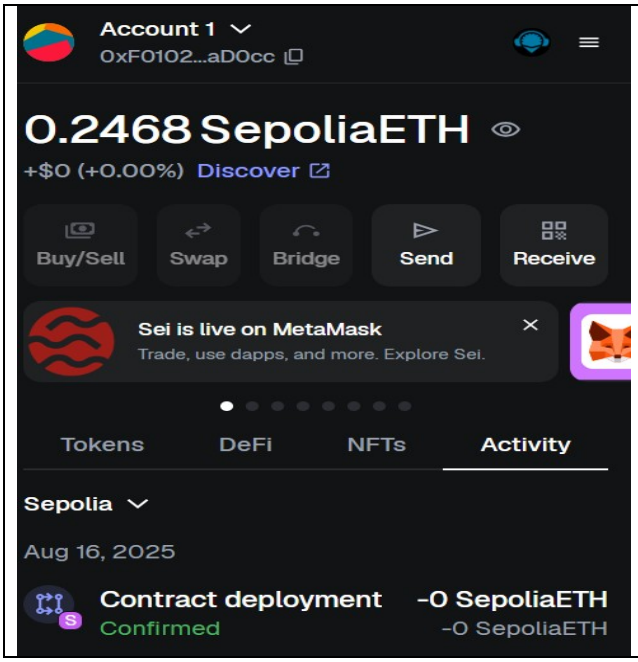






*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)



* Observations

- *After deployment, the smart contract cannot be changed — if there's a bug or error in the code, you must deploy a new version.*
- *The amount of gas spent varies with how complex the constructor is and how much data is stored or modified.*
- *Even the simplest contract will leave a permanent record on the blockchain and require gas to execute.*
- *MetaMask functions not only as a crypto wallet but also as the tool that signs your transactions and sends them to the blockchain network*

ASSESMENT

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 :

Regn. No. :

Signature of the Faculty:

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