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 : Audit 101 – Smart Contract Vulnerabilities

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Algorithm:

```
1. Start
2. Open Remix IDE in a browser.
3. Create a new Solidity file named VulnerableContract.sol.
4. Write a smart contract with intentional vulnerabilities.
5. Compile the contract and check for warnings or compiler errors.
6. Deploy the vulnerable contract using MetaMask on a test network.
7. Analyze its behavior by performing function calls that exploit the weakness.
8. Identify and record the cause of vulnerability.
9. Modify the contract to fix the issue and redeploy it.
10. Re-test the contract to ensure the vulnerability no longer exists.
11. End
```

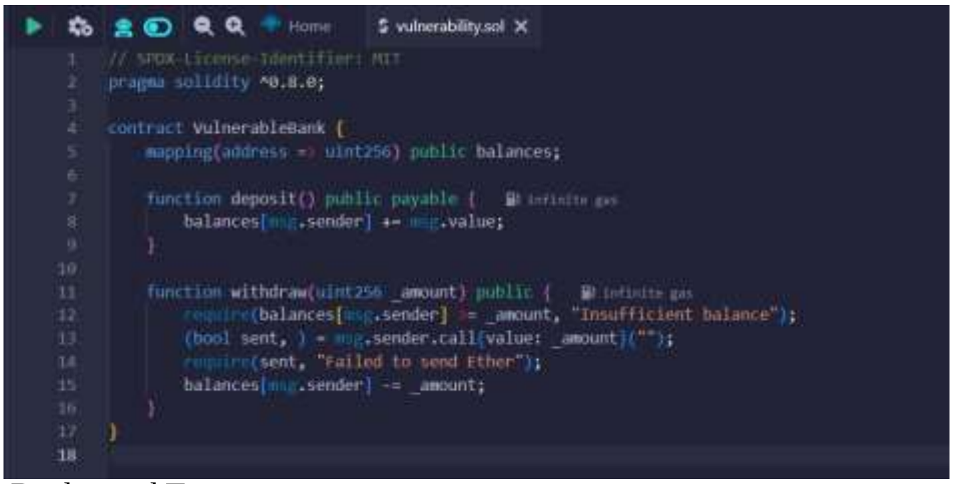
* Software used

1. Remix IDE
2. Solidity
3. MetaMask
4. Test Network (Sepolia)

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

* Testing Phase: Compilation of Code (error detection)

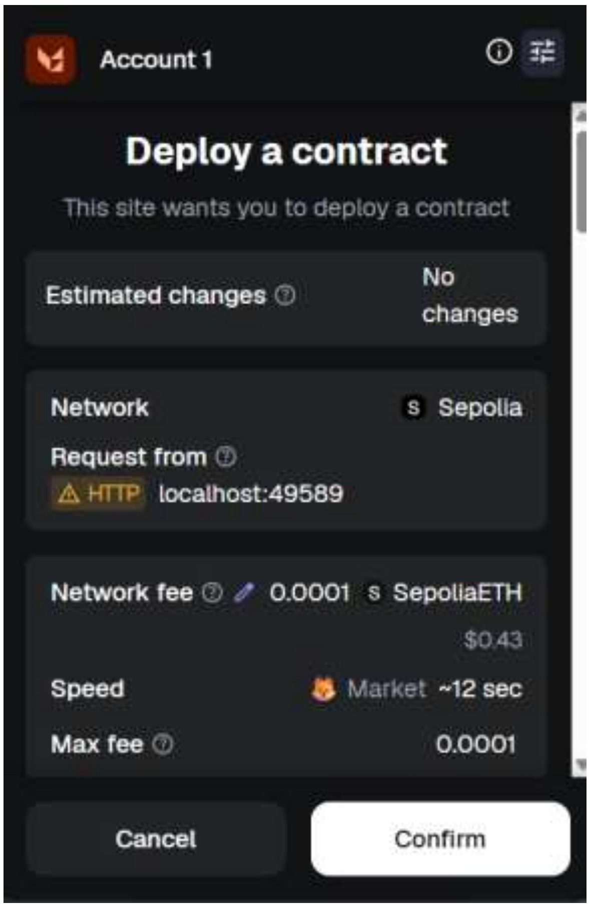
Open Remix IDE. Create a new file VulnerableContract.sol. Write Vulnerable Contract Code



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract VulnerableBank {
    mapping(address => uint256) public balances;

    function deposit() public payable {    infinite gas
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 _amount) public {    infinite gas
        require(balances[msg.sender] >= _amount, "Insufficient balance");
        (bool sent, ) = msg.sender.call{value: _amount}("");
        require(sent, "Failed to send Ether");
        balances[msg.sender] -= _amount;
    }
}
```

Deploy and Test:

Go to Deploy & Run Transactions.
Select Injected Provider - MetaMask and deploy the contract.
Deposit some ETH and attempt multiple withdrawals quickly.
Observe reentrancy behavior (if using test attacker contract).



Deploy a contract

This site wants you to deploy a contract

Estimated changes: No changes

Network: Sepolia

Request from: HTTP localhost:49589

Network fee: 0.0001 SepoliaETH $0.43

Speed: Market ~12 sec

Max fee: 0.0001

0xE4a5139CE5b039b4B11a62b54a715b8108742E40

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

* Implementation Phase: Final Output (no error)

Successfully analyzed a vulnerable smart contract.
Detected and mitigated a Reentrancy Attack vulnerability.
The corrected contract follows best security practices.

* Observations

1.Identified and fixed a reentrancy vulnerability in the smart contract.
2.Understood the importance of auditing and secure coding before deployment.

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.