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

Algorithm:

1. User initiates a token transfer on Chain A (Ethereum).
2. Smart contract locks the tokens in a "bridge" contract on Chain A.
3. A message is sent to Chain B (Polygon) to mint equivalent tokens.
4. User receives wrapped tokens on the destination chain.

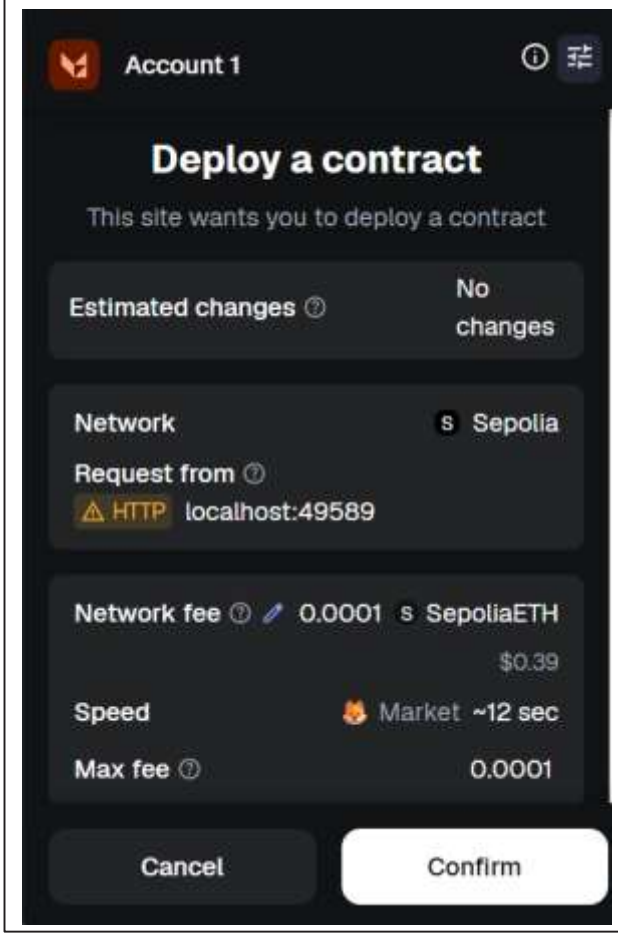
* Softwares used

1. Remix IDE
2. Solidity ^0.8.x
3. MetaMask Wallet (for multi-chain testing)
4. JavaScript VM for simulation.

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

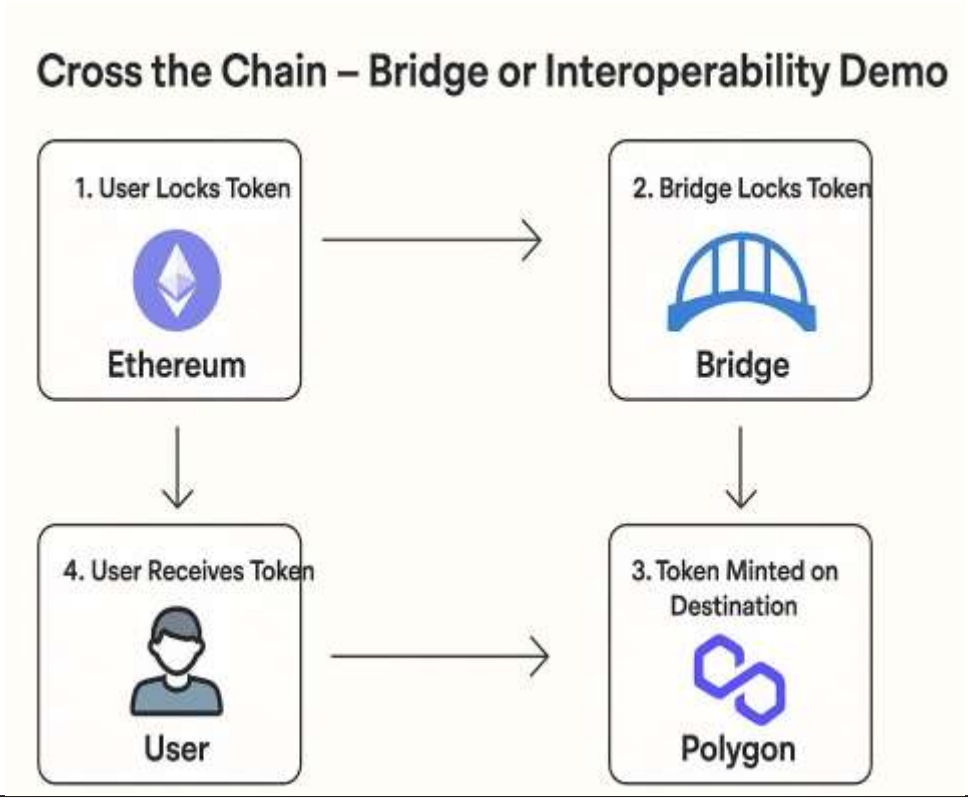
* Testing Phase: Compilation of Code (error detection)

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

contract TokenBridge {
    mapping(address => uint256) public lockedBalance;

    // Lock tokens on source chain
    function lockTokens(uint256 _amount) public {    infinite gas
        lockedBalance[msg.sender] += _amount;
    }

    // Unlock tokens on destination chain
    function unlockTokens(address _to, uint256 _amount) public {    infinite gas
        require(lockedBalance[_to] >= _amount, "Insufficient locked tokens");
        lockedBalance[_to] -= _amount;
        // Mint or transfer equivalent tokens on destination
    }

    // View locked balance
    function getLockedTokens(address _user) public view returns (uint256) {    2806 gas
        return lockedBalance[_user];
    }
}

```



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

* Implementation Phase: Final Output (no error)



* Observations

1. The bridge architecture uses a lock-and-mint mechanism to maintain token peg across chains.
2. Ideal for DeFi apps requiring cross-chain liquidity sharing.
3. Special relayer contracts or oracle networks (e.g., Chainlink CCIP) are used for message passing.
4. Cross-chain smart contracts must handle security issues like replay protection and double-spend.

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 :

Regn. No. :

Signature of the Faculty:

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