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 : Talk to the World – Backend and Oracle Integration

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

1.Write a smart contract that imports Chainlink's price feed interface.
2.Initialize a reference to the desired data feed (ETH/USD).
3.Deploy the contract on the Sepolia testnet using Remix + MetaMask.
4.Call the smart contract function to get the latest ETH/USD price from the Chainlink oracle.
5.Observe and verify that the live data is fetched from off-chain and stored on-chain.

* Software used

1.Remix IDE
2.MetaMask Wallet (Sepolia Testnet)
3.Chainlink Data Feeds

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

* Testing Phase: Compilation of Code (error detection)

1.Write a Smart Contract

Open Remix IDE and create a new Solidity file (e.g., OracleFeed.sol).
Import the Chainlink Price Feed Interface from the official Chainlink documentation.

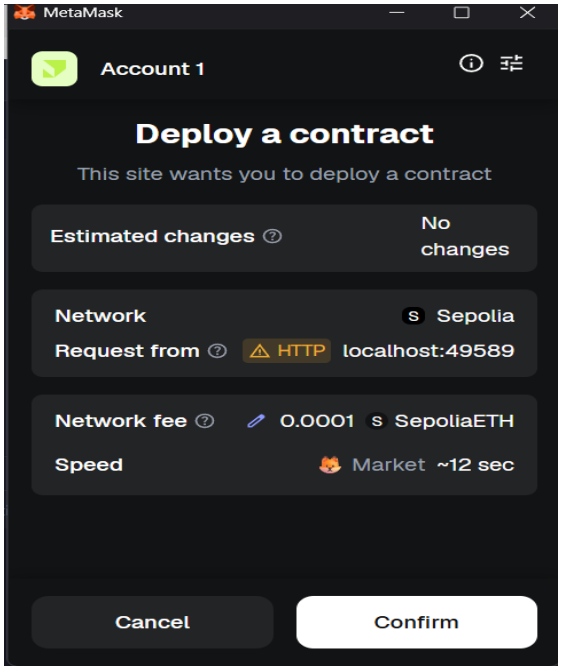
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

// Import Chainlink price feed interface
import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";

contract ETHUSDPriceFeed {
    AggregatorV3Interface internal priceFeed;

    // Constructor to initialize the Chainlink price feed address
    constructor() {    147034 gas 122600 gas
        // ETH/USD price feed address on Sepolia Testnet
        priceFeed = AggregatorV3Interface(
            0x694AA1769357215DE4FAC081bf1f309aDC325306
        );
    }

    // Function to get the latest ETH/USD price
    function getLatestPrice() public view returns (int) {    infinite gas
        (
            ,
            int price,
            ,
            ,
        ) = priceFeed.latestRoundData();
        return price; // returns ETH price in USD (8 decimal places)
    }
```

2.Initialize Chainlink Data Feed

Inside the contract, declare an AggregatorV3Interface variable.
Initialize it with the ETH/USD feed address for the Sepolia testnet

3.Deploy the Contract

Connect MetaMask to Sepolia Testnet.
Compile the contract using Solidity compiler version 0.8.x.
Deploy it using Injected Provider – MetaMask option in Remix.

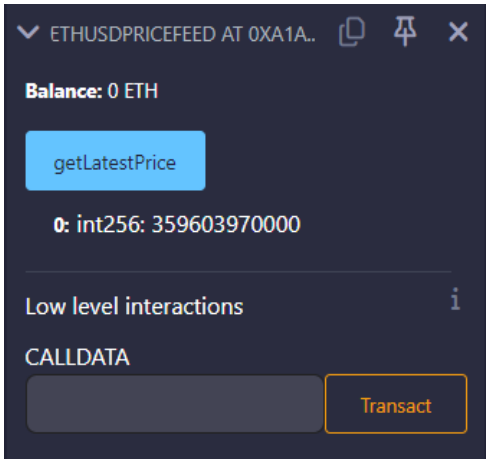


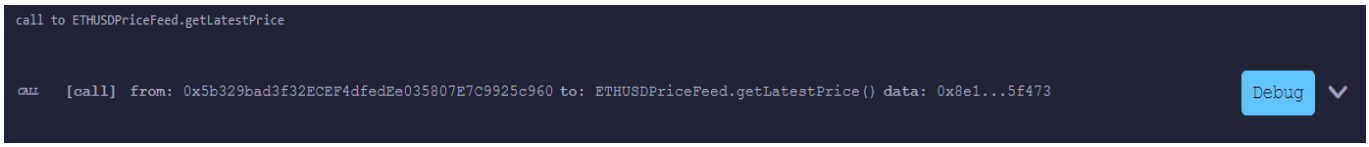
*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

Calling the Function getLatestPrice()

After deployment, expand the deployed contract in Remix.
Click on getLatestPrice() to execute the function.
The function fetches the latest ETH/USD price directly from Chainlink's off-chain oracle and displays it in Remix output.





* Observations

1.Deployment successful on Sepolia Testnet.
2.getLatestPrice() returned real-time ETH/USD price (e.g., 3310 USD).
3.Data and transaction details were verifiable on Etherscan and Remix logs.
4.Demonstrated how off-chain real-world data can be fetched on-chain using Chainlink oracle.

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.