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 the live data is fetched from off-chain and stored on-chain.

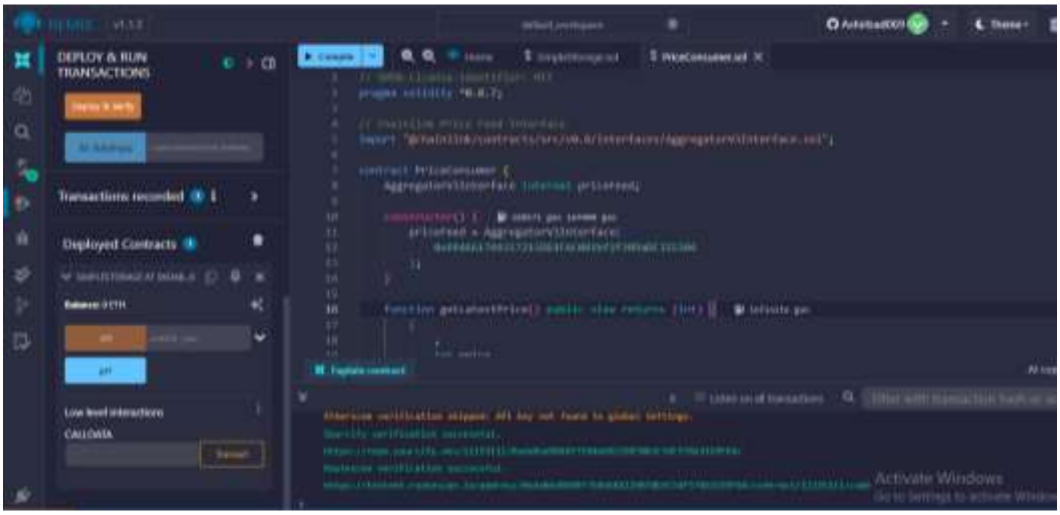
* Softwares used

1. Remix IDE
2. MetaMask Wallet (Sepolia testnet)
3. Chainlink Data Feeds
4. Solidity 0.8.x.

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

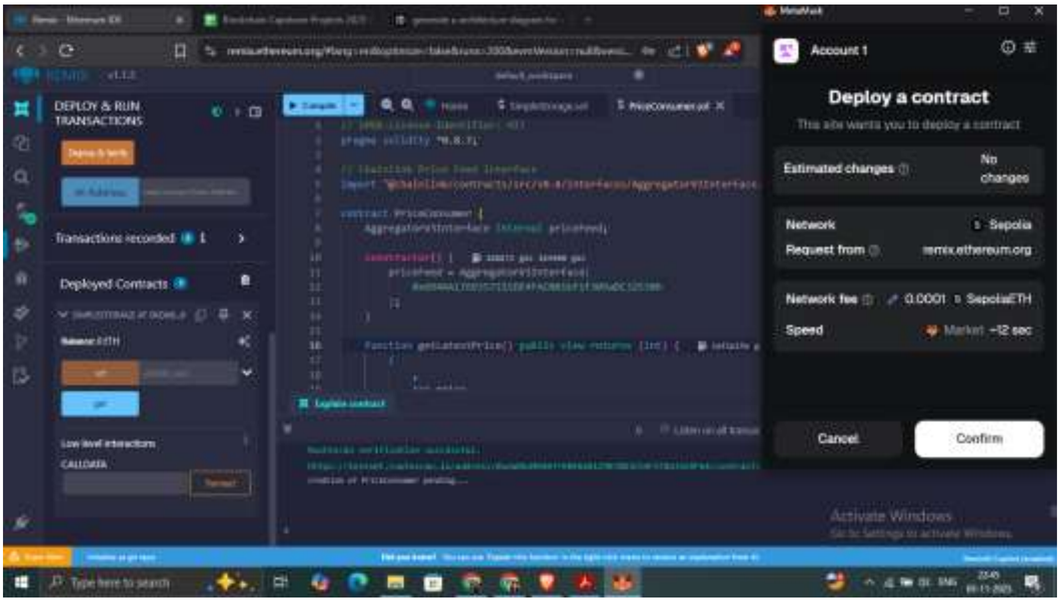
* Testing Phase: Compilation of Code (error detection)

1.Smart contract was compiled and deployed on Sepolia Testnet using Remix and MetaMask.
2.Chainlink price feed address for ETH/USD was correctly linked.
3.Contract function getLatestPrice() was called.
4.The returned integer value represented the current market price of Ethereum in USD.
5.Tool logs and Transaction Hash confirmed live oracle data interaction.



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

// Chainlink Price Feed Interface
import "@chainlink/contracts/src/v0.8/interfaces/AggregatorV3Interface.sol";

contract PriceConsumer {
    AggregatorV3Interface internal priceFeed;

    constructor() {
        priceFeed = AggregatorV3Interface(
            [illegible]
        );
    }

    function getLatestPrice() public view returns (int) {
```

```
Etherscan verification skipped: API key not found in global settings.
Sourcify verification successful.
[illegible]
Routescan verification successful.
[illegible]
```



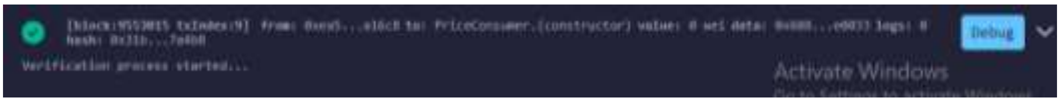
Deploy a contract

This site wants you to deploy a contract

Estimated changes: No changes

Network: Sepolia
Request from: remix.ethereum.org

Network fee: 0.0001 SepoliaETH
Speed: Market ~12 sec



```
[block:9553815 txIndex:9] from: 0xed5...a16c8 to: PriceConsumer.(constructor) value: 0 wei data: 0x608...e0033 logs: 0
hash: 0x31b...7a4b8
Verification process started...
```

1. Deployment successful on Sepolia testnet.
2. getLatestPrice() returned real-time ETH/USD price (e.g., 3310 USD).
3.Transaction details visible on-chain (Etherscan) and in Remix logs.
4.This proved data was fetched from the real world via Chainlink oracle to the smart contract.

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

* Observations

1. Smart contracts cannot access external data directly but can use oracles like Chainlink for real-time information.
2. Chainlink Data Feeds provide secure, tamper-proof prices for blockchain applications.
3. External data fetched becomes part of blockchain history, making it immutable once stored on-chain.
4. Oracle integration enables powerful decentralized applications in areas like DeFi, insurance, and gaming.

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.