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 : Hello Solidity – Writing First Smart Contract

Objective/Aim:

- To gain practical experience in writing a basic smart contract using the Solidity programming language.
- To compile, deploy, and interact with the contract on the Ethereum **Sepolia Testnet** using **Remix IDE** and **MetaMask**.
- To learn how smart contracts can store and retrieve simple data on the blockchain.

Apparatus/Software Used:

- Laptop/PC
- Word for documentation
- Internet for research
- Chrome Browser
- Remix – Ethereum IDE

Theory/Concept:

- **Solidity**: A high-level, contract-oriented programming language used for writing smart contracts that run on Ethereum.

- **Smart Contract**: A self-executing program that automatically enforces rules and agreements without the need for intermediaries.

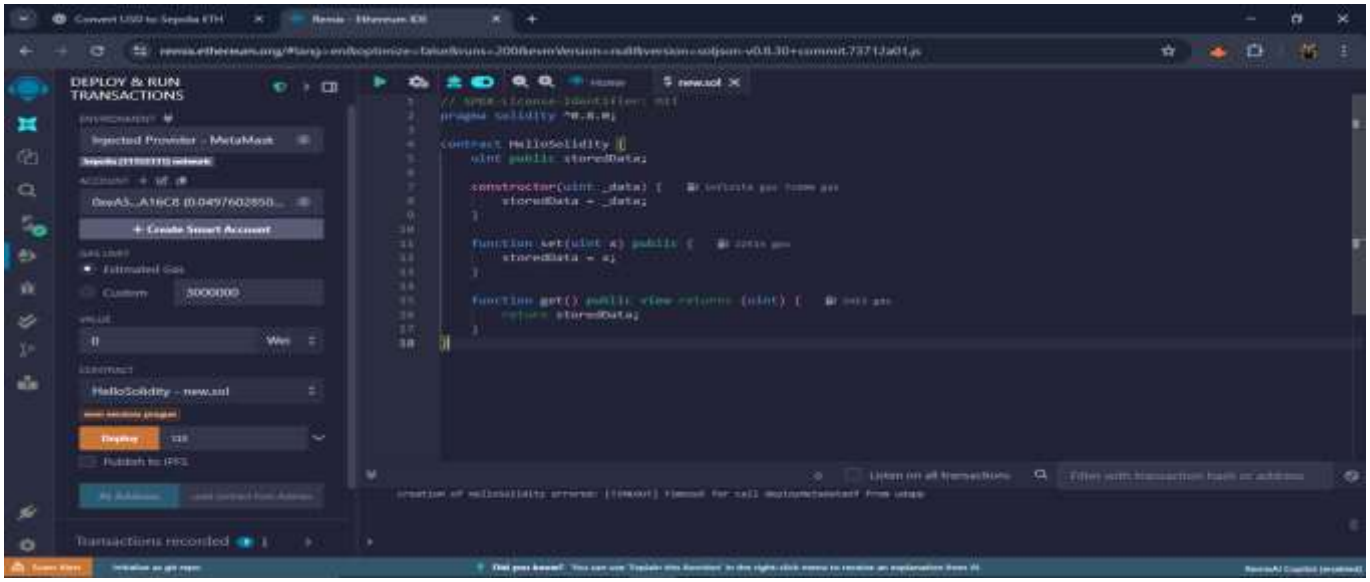
- **Remix IDE**: A web-based platform that provides tools to develop, test, and deploy Solidity contracts.

- **MetaMask**: A browser extension wallet that enables users to manage Ethereum accounts and connect to dApps and testnets like Sepolia.

- **Sepolia Testnet**: An Ethereum test network that mimics the mainnet environment for safe experimentation and deployment of smart contracts.

Procedure:

- **Access Remix IDE**
  - Open your web browser and navigate to: https://remix.ethereum.org
- **Create a New Solidity File**
  - In the Remix file explorer, create a new file (e.g., `new.sol`).
  - Write a smart contract named `HelloSolidity` with the following components:
    - A `public uint` variable named `storedData`
    - A constructor that accepts a parameter `_data` and sets it to `storedData`
    - A `set()` function to update the value of `storedData`
    - A `get()` function to retrieve the current value
- **Connect Remix to MetaMask**
  - In Remix, under the **Deploy & Run Transactions** tab, choose `Injected Provider - MetaMask` from the **Environment** dropdown.
  - Approve the connection request in your MetaMask extension.
- **Ensure Correct Network**
  - Make sure MetaMask is connected to the **Sepolia Testnet** and your wallet is unlocked.
- **Compile the Contract**
  - Use the Solidity compiler in Remix to compile the `HelloSolidity` contract. Ensure there are no errors.
- **Deploy the Contract**
  - Select the `HelloSolidity` contract from the dropdown under the **CONTRACT** section.
  - Enter an initial value (e.g., `123`) in the input box next to the **Deploy** button.
  - Click **Deploy** and confirm the transaction in MetaMask.



Observation Table

Observation Point	Details
Remix Environment	Remix IDE (browser-based IDE for Solidity)
Contract Name	`HelloSolidity`

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: