



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 Experiment :

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

ALGORITHM:

- 1.Start
- 2.Identify all participants in the supply chain (Producer, Transporter, Retailer, Customer).
- 3.Assign each participant a unique blockchain address.
- 4.For every new product batch:
 - Generate a unique Product ID.
 - Record details (name, batch no., source, timestamp)
 - Store transaction on blockchain.
- 5.During transport:
 - Update location data and handling status on blockchain.
- 6.At retailer:
 - Verify authenticity by checking the blockchain ledger.
- 7.Customer scans QR code to view product origin and journey details.
- 8.Smart contract triggers payment once delivery is confirmed.
- 9.Stop

* Software used

- 1.Remix IDE
- 2.MetaMask Wallet
- 3.Solidity Smart Contract
- 4.Ethereum Test Network (Sepolia)

* Testing Phase: Compilation of Code (error detection)

1. Smart Contract Creation:

Open Remix IDE → create a new Solidity file chain.sol.

Write contract to register products, track transfer, and verify authenticity.

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract SupplyChain {
5     struct Product {
6         uint256 id;
7         string name;
8         address currentOwner;
9         string status;
10        bool isRegistered;
11    }
12    mapping(uint256 => Product) public products;
13    event ProductRegistered(uint256 productId, string name, address owner);
14    event ProductTransferred(uint256 productId, address from, address to);
15    event ProductStatusUpdated(uint256 productId, string newStatus);
16    function registerProduct(uint256 _id, string memory _name) public {
17        require(!products[_id].isRegistered, "Product already registered");
18        products[_id] = Product(_id, _name, msg.sender, "Created", true);
19        emit ProductRegistered(_id, _name, msg.sender);
20    }
21    function transferProduct(uint256 _id, address _newOwner) public {
22        require(products[_id].isRegistered, "Product not registered");
23        require(msg.sender == products[_id].currentOwner, "Only current owner can transfer");
24        address oldOwner = products[_id].currentOwner;
25        products[_id].currentOwner = _newOwner;
26        products[_id].status = "Transferred";
27        emit ProductTransferred(_id, oldOwner, _newOwner);
28    }
29
30    function updateStatus(uint256 _id, string memory _newStatus) public {
31        require(products[_id].isRegistered, "Product not registered");
32        require(msg.sender == products[_id].currentOwner, "Only current owner can update");
33        products[_id].status = _newStatus;
34        emit ProductStatusUpdated(_id, _newStatus);
35    }
36
37    function verifyProduct(uint256 _id) public view returns (
38        uint256,
39        string memory,
40        address,
41        string memory
42    ) {
43        require(products[_id].isRegistered, "Product not found");
44        Product memory p = products[_id];
45        return (p.id, p.name, p.currentOwner, p.status);
46    }

```

2. Deployment:

Deploy contract using the “Deploy & Run Transactions” tab in Remix.

Connect MetaMask to Sepolia Testnet for fake ETH gas usage.

```

creation of Chain pending...

view on Etherscan  view on Blockscout

[block:9537868 txIndex:-] from: 0xe4a...ca52e to: Chain.(constructor)
value: 0 wei data: 0x608...e0033 logs: 0 hash: 0x435...375fd

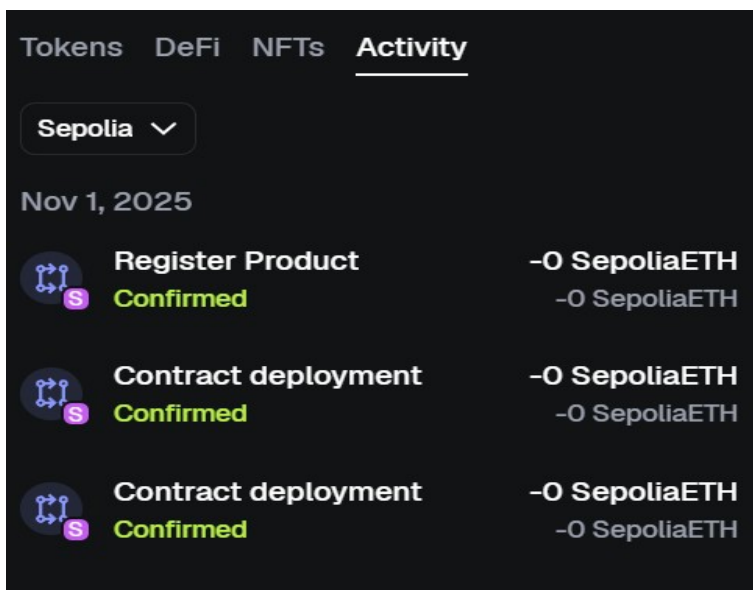
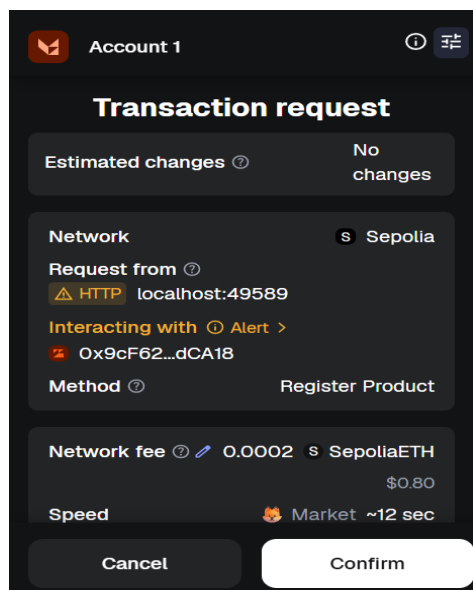
```

* Implementation Phase: Final Output (no error)

Applied and Action Learning

Testing Transactions:

Call functions such as registerProduct(), transferProduct(), and verifyProduct()
Observe transaction hash and confirmation on blockchain.



0x9cF62AbF0dbB580A4f1776F646B28b41B2ddCA18

* Observations

1. Blockchain provides transparent and tamper-proof product tracking throughout the supply chain.
2. Smart contracts automate verification and payment processes efficiently.

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:

Page No.....

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