Experiment No-8

Title: Development of a Blockchain-Based Supply Chain Traceability System.

Aim: To design and implement a blockchain-based system that ensures transparency and traceability in a supply chain using smart contracts.

Theory:

- **Supply Chain Traceability** means tracking the journey of a product from manufacturer → distributor → retailer → customer.

- **Problem in Traditional Systems:** Data can be changed, lacks transparency, and is hard to verify.

- **Blockchain Solution:**

  - Every product movement is recorded as a **transaction**.
  - Data is **tamper-proof** and **transparent**.
  - Smart contracts ensure that each stage follows predefined rules.

- **Smart Contract:** A blockchain program that stores and verifies product information automatically.

- **Ethereum Blockchain:** Used to deploy supply chain contracts.

- **Tools Used:**

  - **Solidity** (Smart contract programming).
  - **Remix IDE** (Write and deploy contract).
  - **MetaMask** (Wallet to connect blockchain).
  - **Ethereum Testnet** (Goerli/Sepolia).

Key Characteristics of dApps :

1. **Transparency** – Everyone can see the product journey.
2. **Immutability** – Records cannot be changed once added.
3. **Trustless System** – No middleman needed.
4. **Security** – Data stored with cryptography.
5. **Traceability** – Full history of product can be tracked.

Steps of Execution:

1. Install **MetaMask** and connect to Testnet.
2. Open **Remix IDE** and write a supply chain smart contract.
3. Compile and deploy the contract on Testnet.
4. Add product details (ID, name, manufacturer).
5. Update product status at each stage (shipped, received, delivered).
6. Retrieve full history of a product.
7. Verify transparency of records.

Stepwise Procedure :

- **Setup Environment**
  - Install **MetaMask**.
  - Get **test ETH** from faucet.
  - Open **Remix IDE**.
- **Write Smart Contract**
  - Create a new Solidity file `SupplyChain.sol`.
  - Define product structure (ID, name, currentOwner, status).
- **Compile Smart Contract**
  - Use Solidity compiler (0.8.x).
  - Fix any errors and compile successfully.
- **Deploy Contract**
  - Use **Injected Web3** (MetaMask) environment.
  - Confirm transaction in MetaMask.
- **Add Products**
  - Call `addProduct()` with details.
- **Update Product State**
  - Call `updateProduct()` to change status (e.g., Manufactured → Shipped → Delivered).
- **Verify Traceability**
  - Call `getProduct()` to check full history.

Program (Smart Contract in Solidity)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract SupplyChain {
    struct Product {
        uint id;
        string name;
        string status;
        address currentOwner;
    }
    mapping(uint => Product) public products;
    uint public productCount;

    event ProductAdded(uint id, string name, address owner);
    event ProductUpdated(uint id, string status, address owner);

    // Add new product
    function addProduct(string memory _name) public {
        productCount++;
        products[productCount] = Product(productCount, _name, "Created", msg.sender);
        emit ProductAdded(productCount, _name, msg.sender);
    }

    // Update product status
    function updateProduct(uint _id, string memory _status) public {
        require(_id > 0 && _id <= productCount, "Invalid product ID");
        Product storage p = products[_id];
        p.status = _status;
        p.currentOwner = msg.sender;
        emit ProductUpdated(_id, _status, msg.sender);
    }

    // Get product details
    function getProduct(uint _id) public view returns (uint, string memory, string memory,
address) {
        require(_id > 0 && _id <= productCount, "Invalid product ID");
        Product memory p = products[_id];
        return (p.id, p.name, p.status, p.currentOwner);
    }
}
```

Key Points :

- Blockchain ensures **tamper-proof supply chain records**.
- Every product is uniquely identified by an **ID**.
- Each update is recorded permanently on the blockchain.
- Transparency improves **trust between manufacturer, distributor, and customer**.

Conclusion :

In this experiment, we developed a blockchain-based supply chain traceability system using Ethereum smart contracts. The system securely recorded product details, updated status at different stages, and allowed verification of the complete product history. This demonstrates how blockchain can improve transparency and trust in supply chain management.

Viva Questions:

1. What is supply chain traceability?
2. How does blockchain improve supply chain management?
3. What is immutability in blockchain?
4. Why is Ethereum suitable for building supply chain dApps?
5. What functions are needed in a supply chain contract?
6. What are the main benefits of recording product data on blockchain?
7. How is trust improved by blockchain in supply chain systems?