



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 :** Build a Use Case - Tokenized Supply Chain Prototype

### Objective/Aim:

To develop and test a blockchain-based **tokenized supply chain model** where each physical product is represented by a digital token.  
To explore how tokenization can enhance **product tracking, data transparency, and process automation** in supply chain management.

### Apparatus/Software Used:

1. **Remix IDE** – for smart contract development and deployment
2. **Ganache** – for local Ethereum network simulation
3. **MetaMask Wallet** – to handle user accounts and sign blockchain transactions
4. **IPFS** – for decentralized storage of product-related data
5. **Ethereum Testnet (Goerli/Sepolia)** – for testing contract functionality
6. **Blockchain Explorer** – to monitor and verify blockchain transactions

### Theory/Concept:

A **tokenized supply chain** integrates blockchain technology with real-world logistics by converting every product or shipment into a unique digital token. These tokens represent ownership and movement of goods through various stages—manufacturing, transport, and retail.

Each token stores vital product details such as origin, timestamp, owner, and condition, ensuring transparency and immutability throughout the supply chain.

#### Key Concepts:

- **Tokenization:**  
Converts physical goods or batches into digital assets (tokens) recorded on the blockchain.
- **Traceability:**  
Enables every stakeholder to view the complete product journey from production to sale.
- **Security:**  
Blockchain's immutable nature ensures that data cannot be altered or falsified.
- **Smart Contracts:**  
Automate product transfers, ownership changes, and event updates based on predefined conditions such as delivery or payment confirmation.

## Procedure:

Set up the environment: -

- Install Node.js, Hardhat, and **OpenZeppelin libraries**.
- Launch a local blockchain network using **Ganache** or **Hardhat Node**..

Create the smart contract

- Write a Solidity smart contract named *TokenizedSupplyChain.sol*
- Define structures for product data, ownership, and current stage.

Compile & Deploy the contract

- Compile using Remix IDE or Hardhat
- Deploy to the test network and record the contract **address** and **ABI**..

Add product metadata:

- Create a JSON file containing details such as product ID, batch number, and origin.
- Upload it to **IPFS** or **Pinata** to get a unique IPFS URI.

***Mint tokens for each product:***

- The manufacturer mints a token using `mintProduct()` with the IPFS metadata.
- Each token gets a unique Token ID that represents a real-world product.

□

***Simulate supply chain flow:***

- **Manufacturer → Distributor:** Transfer token ownership and update status to *Dispatched*.
- **Distributor → Retailer:** Update token stage to *In Transit* or *Delivered*.
- **Retailer → Consumer:** Complete the cycle by transferring the token on purchase.

***Verify transactions:***

- Check transaction details, token history, and ownership changes using **Etherscan** or any blockchain explorer.

## Observation Table:

Action	Smart Contract Function	Stage/Status	Result/Output
Token Minting	<code>mintProduct()</code>	Manufactured	Token created with product metadata
Transfer to Distributor	<code>transferFrom()</code>	Dispatched	Ownership changed to Distributor
Update Transit Info	<code>updateStatus()</code>	In Transit	Status updated successfully
Delivery Confirmation	<code>updateStatus()</code>	Delivered	Final stage recorded on blockchain

## ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Interpretation Result and	10		
Record of Applied and Action Learning	10		
Viva	10		
<b>Total</b>	<b>50</b>		

**Signature of the Student:**

Name :

Regn. No.

**Signature of the Faculty:**