Names: TUYISENGE Jackson  
Registration Number: 23RP00810  
Class: BTech-IT  
Year: Four  
Module: Blockchain Technology

**SUMMATIVE ASSESSMENT – PRACTICAL**

**Project Title: Blockchain-Based Farm Produce Sale Log**

Date of Submission: 30 April 2025

Table of Contents

[1. Project Abstract: 3](#_Toc197689933)

[2. Problem Statement: 3](#_Toc197689934)

[3. Clear Objective: 3](#_Toc197689935)

[4. Description of the Problem or Need: 3](#_Toc197689936)

[5. Proposed Solution: 3](#_Toc197689937)

[6. Key Features: 4](#_Toc197689938)

[7. Technical Stack 4](#_Toc197689939)

[8. Security Measures 4](#_Toc197689940)

[9. Implementation Plan 5](#_Toc197689941)

# 1. Project Abstract:

This project aims to develop a decentralized system that allows farmers to record and manage the sales of their farm produce using blockchain technology. By offering a transparent and tamper-proof logging mechanism, the system enhances accountability and trust in local produce markets. Farmers can manually enter sale records, track stock levels, view sales history, and generate reports all from a single dashboard.

# 2. Problem Statement:

Many smallholder farmers lack efficient and secure ways to record their produce sales, which leads to poor financial tracking, disputes over transactions, and limited access to formal markets or loans due to missing verifiable data.

# 3. Clear Objective:

To build a blockchain-based sale logging platform that enables individual farmers to securely record, manage, and trace all farm produce sales, thus promoting transparency and better financial management.

# 4. Description of the Problem or Need:

Traditional methods of record-keeping in agriculture are manual and prone to errors or manipulation. Farmers often use notebooks or rely on memory to track their sales. This results in lost data, disputes with buyers, and challenges when applying for loans or joining cooperatives. There is a need for a secure, simple, and farmer-centered digital solution to log all produce sales accurately and reliably.

# 5. Proposed Solution:

The proposed system is a web-based platform where a farmer can:

* Log every sale transaction.
* Track produce stock levels.
* Generate blockchain-verified digital receipts.
* Monitor income trends over time.

The blockchain integration ensures that each record is immutable and timestamped, preventing tampering and enhancing trustworthiness.

# 6. Key Features:

* **Farmer Authentication:** Secure login and profile management.
* **Produce Management:** Add items for sale.
* **Sale Logging:** Record each transaction with produce details, quantity, buyer info, and price.
* **Blockchain Integration:** Immutable record-keeping of all transactions.
* **Sales History:** Easily view and export previous sales.
* **Reports & Analytics:** Visual summaries of income, most sold items, and performance trends.
* **Low Stock Alerts:** Notify farmer when an item is running out.

# 7. Technical Stack

* **Frontend**: React.js or Vue.js (for dashboard and user interaction)
* **Blockchain**: Ethereum using Solidity for smart contracts
* **Smart Contract Tools**: Hardhat or Truffle (for development, testing, and deployment)
* **Storage**: IPFS (optional, for storing any receipts or documents)
* **Wallet Integration**: MetaMask or WalletConnect

# 8. Security Measures

* Smart contract auditing using MythX or Slither
* Role-based access control (RBAC) for farmer authentication
* Encrypted data transmission using HTTPS
* Rate limiting and input validation to prevent common attacks (e.g., SQL injection, XSS)
* Secure wallet login and transaction confirmation via MetaMask
* Regular backups and failover strategies

# 9. Implementation Plan

|  |  |
| --- | --- |
| **Timeline** | **Deliverables** |
| **Day 1:** Setup & Planning | - Initialized GitHub repository - Defined project folder structure - Installed necessary dev tools (Node.js, Hardhat/Truffle, React, etc.) - Chose blockchain platform (e.g., Ethereum testnet) - Drafted system architecture (frontend ↔ backend ↔ blockchain) |
| **Day 2:** Smart Contract Development | - Smart contract for produce sales written in Solidity - Handles: adding produce, logging sales, generating receipts - Deployed to local/testnet using Hardhat or Truffle - ABI and contract address exported for frontend/backend use |
| **Day 3:** Frontend Development | - React/Vue components for: login, dashboard, produce list, add/edit forms, sale logging form, sales history, and reports - UI integrated with dummy data - Wallet (MetaMask) login implemented |
| **Day 4:** Blockchain Integration | - Connected frontend and backend to deployed smart contract - Verified on-chain sale logging works |
| **Day 5:** Blockchain Testing | - Test cases written for common user interactions - Blockchain transaction feedback shown in UI |
| **Day 6:** Security & Penetration Testing | - Smart contract tested with tools like Slither - API vulnerability checks (rate limiting, validation) - Secured authentication flow - Wallet transaction validation and fallback handling |
| **Day 7:** Deployment & Final Testing | - Frontend deployed on Netlify/Vercel - Backend deployed on Render/Railway - Final testing (manual + automated) - Documentation created for system usage and setup - Demo video or walkthrough recorded (optional) |