// Role: Blockchain Developer

// Module: Smart Contract for VeriHarvest Food Traceability

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.19;

contract VeriHarvest {

struct FoodBatch {

uint256 batchId;

string productName;

string supplier;

uint256 freshnessScore;

bool blockchainVerified;

string status; // "Safe", "Warning", "Failed"

address addedBy;

}

mapping(uint256 => FoodBatch) public batches;

uint256 public batchCount;

address public admin;

event BatchAdded(uint256 batchId, string productName, string supplier);

event BatchUpdated(uint256 batchId, string status);

constructor() {

admin = msg.sender;

}

modifier onlyAdmin() {

require(msg.sender == admin, "Only admin can perform this action");

\_;

}

function addBatch(string memory \_productName, string memory \_supplier, uint256 \_freshnessScore) public onlyAdmin {

batchCount++;

batches[batchCount] = FoodBatch(batchCount, \_productName, \_supplier, \_freshnessScore, true, "Safe", msg.sender);

emit BatchAdded(batchCount, \_productName, \_supplier);

}

function updateBatchStatus(uint256 \_batchId, string memory \_status) public onlyAdmin {

require(\_batchId <= batchCount, "Batch does not exist");

batches[\_batchId].status = \_status;

emit BatchUpdated(\_batchId, \_status);

}

function getBatchDetails(uint256 \_batchId) public view returns (FoodBatch memory) {

require(\_batchId <= batchCount, "Batch does not exist");

return batches[\_batchId];

}

}