FOOD TRACKING SYSTEM:

INPUT:

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract FoodTracking {

address public owner;

enum FoodStatus {

Unverified,

Verified,

Consumed

}

struct FoodItem {

string itemId;

string productName;

string origin;

uint256 sentTimestamp;

FoodStatus status;

}

mapping(string => FoodItem) public foodItems;

event FoodItemSent(

string indexed itemId,

string productName,

string origin,

uint256 sentTimestamp

);

event FoodItemVerified(string indexed itemId);

event FoodItemConsumed(string indexed itemId);

constructor() {

owner = msg.sender;

}

modifier onlyOwner() {

require(msg.sender == owner, "Only contract owner can call this");

\_;

}

modifier onlyUnconsumed(string memory itemId) {

require(

foodItems[itemId].status == FoodStatus.Verified,

"Item is not verified or already consumed"

);

\_;

}

function sendFoodItem(

string memory itemId,

string memory productName,

string memory origin

) external onlyOwner {

require(

bytes(foodItems[itemId].itemId).length == 0,

"Item already exists"

);

foodItems[itemId] = FoodItem({

itemId: itemId,

productName: productName,

origin: origin,

sentTimestamp: block.timestamp,

status: FoodStatus.Unverified

});

emit FoodItemSent(itemId, productName, origin, block.timestamp);

}

function verifyFoodItem(string memory itemId) external onlyOwner {

require(

bytes(foodItems[itemId].itemId).length > 0,

"Item does not exist"

);

require(

foodItems[itemId].status == FoodStatus.Unverified,

"Item is already verified or consumed"

);

foodItems[itemId].status = FoodStatus.Verified;

emit FoodItemVerified(itemId);

}

function consumeFoodItem(

string memory itemId

) external onlyUnconsumed(itemId) {

foodItems[itemId].status = FoodStatus.Consumed;

emit FoodItemConsumed(itemId);

}

function getFoodItemDetails(

string memory itemId

)

external

view

returns (string memory, string memory, uint256, FoodStatus)

{

FoodItem memory item = foodItems[itemId];

return (item.productName, item.origin, item.sentTimestamp, item.status);

}

}

ABI CODE:

[

{

"inputs": [],

"stateMutability": "nonpayable",

"type": "constructor"

},

{

"anonymous": false,

"inputs": [

{

"indexed": true,

"internalType": "string",

"name": "itemId",

"type": "string"

}

],

"name": "FoodItemConsumed",

"type": "event"

},

{

"anonymous": false,

"inputs": [

{

"indexed": true,

"internalType": "string",

"name": "itemId",

"type": "string"

},

{

"indexed": false,

"internalType": "string",

"name": "productName",

"type": "string"

},

{

"indexed": false,

"internalType": "string",

"name": "origin",

"type": "string"

},

{

"indexed": false,

"internalType": "uint256",

"name": "sentTimestamp",

"type": "uint256"

}

],

"name": "FoodItemSent",

"type": "event"

},

{

"anonymous": false,

"inputs": [

{

"indexed": true,

"internalType": "string",

"name": "itemId",

"type": "string"

}

],

"name": "FoodItemVerified",

"type": "event"

},

{

"inputs": [

{

"internalType": "string",

"name": "itemId",

"type": "string"

}

],

"name": "consumeFoodItem",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "",

"type": "string"

}

],

"name": "foodItems",

"outputs": [

{

"internalType": "string",

"name": "itemId",

"type": "string"

},

{

"internalType": "string",

"name": "productName",

"type": "string"

},

{

"internalType": "string",

"name": "origin",

"type": "string"

},

{

"internalType": "uint256",

"name": "sentTimestamp",

"type": "uint256"

},

{

"internalType": "enum FoodTracking.FoodStatus",

"name": "status",

"type": "uint8"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "itemId",

"type": "string"

}

],

"name": "getFoodItemDetails",

"outputs": [

{

"internalType": "string",

"name": "",

"type": "string"

},

{

"internalType": "string",

"name": "",

"type": "string"

},

{

"internalType": "uint256",

"name": "",

"type": "uint256"

},

{

"internalType": "enum FoodTracking.FoodStatus",

"name": "",

"type": "uint8"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [],

"name": "owner",

"outputs": [

{

"internalType": "address",

"name": "",

"type": "address"

}

],

"stateMutability": "view",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "itemId",

"type": "string"

},

{

"internalType": "string",

"name": "productName",

"type": "string"

},

{

"internalType": "string",

"name": "origin",

"type": "string"

}

],

"name": "sendFoodItem",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

},

{

"inputs": [

{

"internalType": "string",

"name": "itemId",

"type": "string"

}

],

"name": "verifyFoodItem",

"outputs": [],

"stateMutability": "nonpayable",

"type": "function"

}

]

BYTE CODE:



OUTPUT:

