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:


```
608060405234801561001057600080fd
5b50336000806101000a81548173ffffff
ffffffffffffffffffffffffffffffff021916908
373ffffffffffffffffffffffffffffffffffffff16
0217905550611c3806100606000396000
f3fe608060405234801561001057600080
fd5b506004361061006257600035600e
01c806308f52751146100675780632b4426c91461009a578063cfd26146100b65780638da5cb5b146100ea57806
```

391acc17214610108578063ca2f564714
610124575b600080fd5b610081600480
360381019061007c9190610c73565b61
0140565b604051610091949392919061
0dcb565b6040518091039f35b6100b4
600480360381019061100af9190610c73
565b610392565b005b6100d060048036
038101906100cb9190610c73565b6104
b8565b6040516100e195949392919061
0e1e565b6040518091039f35b6100f26
106a9565b6040516100ff9190610ec756
5b6040518091039f35b610122600480
360381019061011d9190610ee2565b61
06cd565b005b61013e60048036038101
906101399190610c73565b6108f8565b
005b606080600080600060018660405161
610158919061fc5565b908152602001
604051809103902060405180604a00160
405290816000820180546101819061100
0b565b80601f0160208091040260200l

6040519081016040528092919081815260200182805461 01ad9061100b565b80

b565b801561031e5780601f106102f357
61010080835404028352916020019161
031e565b820191906000526020600020
905b8154815290600101906020018083
1161030157829003601f168201915b50
50505050815260200160038201548152
6020016004820160009054906101000a
900460ff166002811115610356576103
55610d54565b5b600281111561036857
610367610d54565b5b81525050905080
60200151816040015182606001518360
80015194509450945094505091935091
93565b80600160028111156103a75761
03a6610d54565b5b6001826040516103
b79190610fc5565b9081526020016040
51809103902060040160009054906101
000a900460ff1660028111156103e8576
103e7610d54565b5b146104285760405
17f08c379a0000000000000000000000
00000000000000000000000000000000

000815260040161041f906110ae565b6
040518091039 0fd5b600260018360405
161043a9190610fc5565b90815260200
160405180910390206004016000610100
00a81548160ff021916908360028111156
1046d5761046c610d54565b5b021790
55508160405161048091906 10fc5565b
604051809103902 07f10903cdad98900
af2897d7e63e43696fd2ff20b6b781cc35
e32cc9ecada0c2ec6040516040518091039
0a25050565b600181805160208101820
1805184825260208301602085012081
835280955050505050506000915090508
060000180546104f19061100b565b80601
f0160208091040260200160405190810
160405280929190818152602001828
05461051d9061100b565b801561056a
5780601f1061053f576101008083540402
835291602001919161056a565b820191906
000526020600020905b8154815290

60010190602001808311610 54d578290
03601f168201915b505050505090806 0
0101805461057f9061100b565b80601f
0160208091040260200160405190810 1
6040528092919081815260200182805 4
6105ab9061100b565b80156105f85780
601f106105cd576101008083540402 83
529160200191610 5f8565b8201919060
0052602060002090 5b81548152906001
0190602001808311610 5db5782900360
1f168201915b505050505090806002 01
805461060d9061100b565b80601f0160
208091040260200160405190810160 40
528092919081815260200182805461 06
399061100b565b80156106865780601f
1061065b576101008083540402835291
602001916106865 65b82019190600052
602060002090 5b81548152906001019 0
602001808311610669578290 03601f16
8201915b505050505090806003015490

80600401600090549061010000a900460
ff16905085565b600080549061010000a9
00473ffffffffffffffffffffffffffffffffff
1681565b600080549061010000a900473
ffffffffffffffffffffffffffffffffffff1673ff
ffffffffffffffffffffffffffffffffff163373f
fffffffffffffffffffffffffffffffffff161461
075b576040517f08c379a00000000000
0000000000000000000000000000000000
0000000000008152600401610752906
11140565b604051809103 90fd5b6000
600184604051610 76d9190610fc5565b
90815260200160405180910390206000
018054610789906 1100b565b90501461
07cb576040517f08c379a00000000000
0000000000000000000000000000000000
0000000000008152600401610 7c2906
111ac565b604051809103 90fd5b6040
518060a00160405280848152602001 83
81526020018281526020014281526020

01600060028111156108025761080161
0d54565b5b8152506001846040516108
159190610fc5565b908152602001604 0
5180910390206000820151816000019 0
8161083891906113785 65b5060208201
5181600101908161084e919061137856
5b5060408201518160020190816108 64
919061137856 5b5060608201518160 03
0155608082015181600401600061010 0
0a81548160ff021916908360028111156
1089a57610899610d54565b5b0217905
5509050508260405161 08b09190610fc
5565b60405180910390207f1deb5b3f13
32885a208a3b03c4438c7daad01496d3
2418f03019bade53c0841e83834260405
16108eb939291906114 4a565b6040518
0910390a2505050565b600805490610
1000a900473ffffffffffffffffffffffffffffffff
fffffff1673ffffffffffffffffffffffffffffffff
ffffff163373ffffffffffffffffffffffffffffffff

ffffff16146109865760405 17f08c379a0
0000000000000000000000000000000
0000000000000000000081526004
0161097d90611140565b604051809103
90fd5b60006001826040516109891 90
610fc5565b90815260200160405180 91
039020600001805461 09b49061100b56
5b90501161 09f6576040517f08c379a00
0000000000000000000000000000000
0000000000000000000815260040
16109ed906114db565b604051809103 9
0fd5b60006002811115610a0a57610a0
9610d54565b5b60018260405161 0a1a9
190610fc5565b90815260200160405180 18
091039020600401600090549061010 00
a900460ff16600281111561 0a4b57610a
4a610d54565b5b14610a8b576040517f
08c379a0000000000000000000000
0000000000000000000000000000000
8152600401610a829061156d565b6040

5180910390fd5b6001808260405161 0a
9c9190610fc5565b9081526020016040
5180910390206004016000610 1000a81
548160ff0219169083600281111561 0ac
f57610ace610d54565b5b021790555 08
0604051610ae29190610fc5565b60405
180910390207f368cb808019f63c8cdd7
e7087e42db5b33818a846fdd80e952766
474915241586040516040518091 0390a
250565b6000604051905090565b60008
0fd5b600080fd5b600080fd5b600080fd
5b6000601f19601f8301169050919050
565b7f4e487b71000000000000000000
00000000000000000000000000000000
000000600052604160045260246000fd
5b610b8082610b37565b810181811067
ffffffffffffffff82111715610b9f57610b9e
610b48565b5b80604052505050565b60
00610bb2610b19565b9050610bbe8282
610b77565b919050565b600067ffffffff

fffffff821115610bde57610bdd610b485
65b5b610be782610b37565b905060208
1019050919050565b828183376000838
3015250505056 5b6000610c16610c118
4610bc3565b610ba8565b90508281526
0208101848484011115610c3257610c3
1610b32565b5b610c3d848285610bf45
65b5093925050565b600082601f830
112610c5a57610c59610b2d565b5b813
5610c6a848260208601610c03565b915
0509291505056 5b6000602082840312 1
5610c8957610c88610b23565b5b60008
2013567fffffffffffffff811115610ca757
610ca6610b28565b5b610cb384828501
610c45565b9150509291505056 5b6000
8151905091905056 5b6000828252602 0
8201905092915050565b60005b838110
15610cf657808201518401526020 81
019050610cdb565b60008484015250 50
5050565b6000610d0d82610cbc565b61

0d178185610cc7565b9350610d278185
60208601610cd8565b610d3081610b37
565b840191505092915050565b600081
9050919050565b610d4e81610d3b565b
82525050565b7f4e487b710000000000
0000000000000000000000000000000000
0000000000006000526021600452602
46000fd5b60038110610d9457610d93
610d54565b5b50565b6000819050610d
a582610d83565b919050565b6000610d
b582610d97565b9050919050565b610d
c581610daa565b82525050565b600060
8082019050818103600083015 2610de5
8187610d02565b90508181036020830 1
52610df98186610d02565b9050610e08
6040830185610d45565b610e15606083
0184610dbc565b95945050505050565b
600060a0820190508181036000830152
610e388188610d02565b905081810360
20830152610e4c8187610d02565b9050

818103604083015261 0e608186610d02
565b9050610e6f606083018561 0d4556
5b610e7c608083018461 0dbc565b9695
5050505050565b600073ffffffffffffff
ffffffffffffffffffffffff82169050919050
65b6000610eb182610e86565b9050919
050565b610ec181610ea6565b8252505
0565b60006020820190506 10edc60008
30184610eb8565b92915050565b60008
060006060848603121561 0efb57610efa
610b23565b5b600840135 67ffffffffff
ffff811115610f1957610f18610b28565b
5b610f2586828701610c45565b935050
602084013567ffffffffffffff811115610
f4657610f45610b28565b5b610f528682
8701610c45565b92505060408401356 7
ffffffffffffff811115610f7357610f7261
0b28565b5b610f7f86828701610c45565
b91505092509250925 65b60008190509
2915050565b6000610f9f82610cbc565b

610fa98185610f89565b9350610fb9818
560208601610cd8565b808401915050 9
2915050565b6000610fd18284610f945
65b9150819050915050565b7f4e487
b7100000000000000000000000000
0000000000000000000000060005
2602260045260246000fd5b600060028
2049050600182168061102357607f821
691505b602082108103611036576110 3
5610fdc565b5b50919050565b7f497465
6d206973206e6f74207665726966965
64206f7220616c726561647920600082
01527f636f6e73756d656400000000000
0000000000000000000000000000000
0000602082015250565b6000611098 6
02883610cc7565b91506110a38261103
c565b6040820190509050565b60006
020820190508181036000830152611 0c
78161108b565b905091905056 5b7f4f6e
6c7920636f6e747261637420 6f776e657

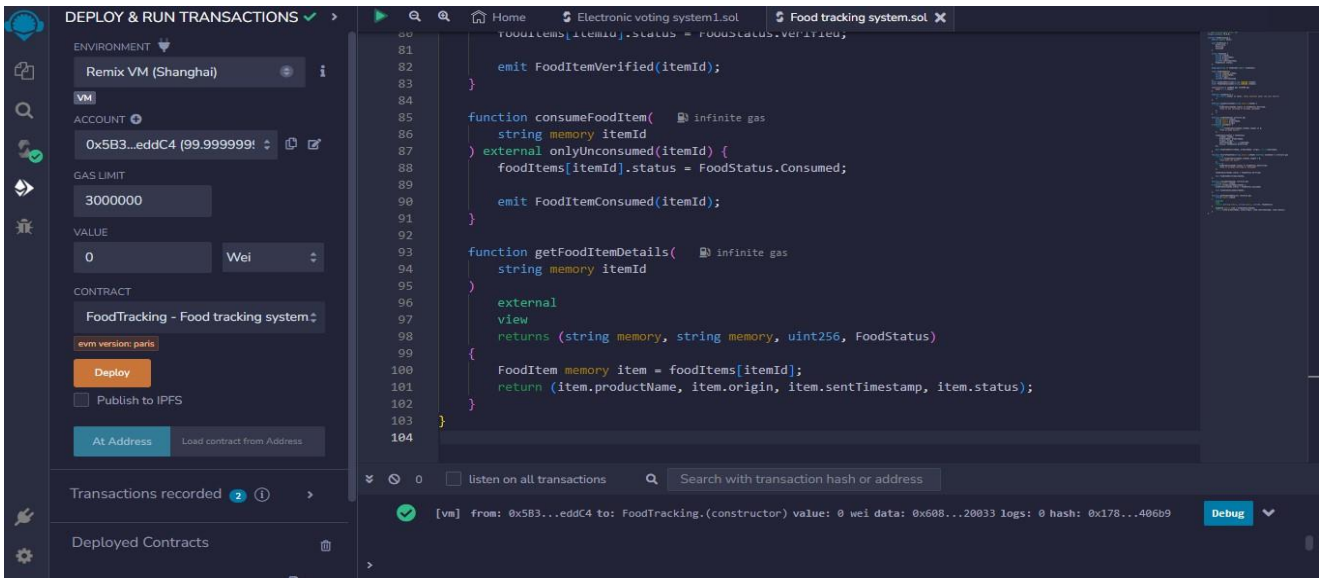
22063616e2063616c6c2074686960008
201527f730000000000000000000000
000000000000000000000000000000
000000602082015250565b600061112
a602183610cc7565b91506111358261 1
0ce565b6040820190509190565b600
060208201905081810360008301526111
1598161111d565b905091905056 5b7f4
974656d20616c72656164792065786973
374730000000000000000000000006
000820152505 65b600061119660138 36
10cc7565b91506111a18261116056 5b6
020820190509190505 65b60006020820
190508181036000830152611 1c581611
189565b905091905056 5b60008190508
160005260206000209050919050565b6
0006020601f830104905091905056 5b6
00082821b905092915050565b6000600
8830261122e7ffffffffffffffffffffffffff
ffffffffffffffffffffffffffffff826111f15

65b61123886836111f1565b955080198
41693508086168417925050509392505
050565b60008190509190505 65b60006
11275611270611 26b84610d3b565b611
250565b610d3b565b905091905 0565b6
000819050919050565b61128f8361125
a565b6112a361129b8261127c565b848
4546111fe565b82555050505 0565b600
090565b6112b86112ab565b6112c3818
484611286565b505050565b5b8181101
56112e7576112dc6000826112b0565b6
001810190506112c9565b5050565b601
f82111561132c576112fd816111cc565b
611306846111e1565b81016020851015
611315578190505b611329611321856 1
11e1565b8301826112c8565b50505b50
5050565b600082821c9050929150505 6
5b600061134f600198460080261133 1
565b19808316915050929150505 60b60
006113683836113 3e565b9150826002

0282179050929150505 65b6113818261
0cbc565b67ffffffffffffff81111561139
a57611399610b48565b5b6113a482546
1100b565b6113af8282856112eb565b6
000602090506 01f831160018114 6113e
25760008415 6113d057828701519 0505
b6113da858261135c565b86555061144
2565b601f1984166113f0866111cc565b
60005b828110156114185784890151 82
556001820191506020850194506020 81
019050611 3f3565b868310156114355 7
848901516114316 01f89168261133e56
5b8355505b6001600288020188555050
505b505050505050565b600606082 01
905081810360008301526114648186 61
0d02565b9050818103602083015261 14
788185610d02565b9050611487604083
0184610d45565b9493505050565b7f
4974656d20646f6573206e6f742065786
9737400000000000000000000006

0008201525056 5b60006114c56013836
10cc7565b91506114d08261148f565b6
0208201905091905056 5b60006020820
1905081810360008301526114f481611
4b8565b905091905056 5b7f4974656d2
0697320616c726561647920766572696
6696564206f7220636f6e736000820152
7f756d656400000000000000000000000
0000000000000000000000000000000000
006020820152505 65b600061155760248
3610cc7565b9150611562826114fb56
5b6040820190509190505 65b60006020
8201905081810360008301526115868161
61154a565b905091905056fea2646970
6673582212207293306a7cb31a50468fc
c254553e1fa266d31971102f817aa2c15
a2ea91ffb864736f6c63430008120033

OUTPUT:


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Remix VM (Shanghai)
VM
ACCOUNT
0x5B3...eddC4 (99.999999
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT
FoodTracking - Food tracking system
evm version: paris
Deploy
Publish to IPFS
At Address
Load contract from Address
Transactions recorded 2
Deployed Contracts
Home
Electronic voting system1.sol
Food tracking system.sol
emit FoodItemVerified(itemId);
function consumeFoodItem(infinite gas
string memory itemId
) external onlyUnconsumed(itemId) {
foodItems[itemId].status = FoodStatus.Consumed;
emit FoodItemConsumed(itemId);
function getFoodItemDetails(infinite gas
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);
listen on all transactions
Search with transaction hash or address
[vm] from: 0x5B3...eddC4 to: FoodTracking.(constructor) value: 0 wei data: 0x608...20033 logs: 0 hash: 0x178...406b9
Debug