

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).len
```



```
gth == 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:

```
6080604052348015610010576000
80fd5b50336000806101000a81548
173fffffffffffffffffffffffffffff
021916908373ffffffffffffffffffff
ffffffffffffffff1602179055506115c380
6100606000396000f3fe608060405
```



234801561001057600080fd5b5060
0436106100625760003560e01c80
6308f52751146100675780632b442
6c91461009a578063820cfd261461
00b65780638da5cb5b146100ea57
806391acc17214610108578063ca2
f564714610124575b600080fd5b61
0081600480360381019061007c91
90610c73565b610140565b604051
6100919493929190610dcb565b60
405180910390f35b6100b46004803
6038101906100af9190610c73565b
610392565b005b6100d060048036
038101906100cb9190610c73565b
6104b8565b6040516100e1959493
929190610e1e565b604051809103
90f35b6100f26106a9565b6040516
100ff9190610ec7565b6040518091
0390f35b610122600480360381019
061011d9190610ee2565b6106cd5



65b005b61013e600480360381019
06101399190610c73565b6108f856
5b005b6060806000806000600186
6040516101589190610fc5565b908
1526020016040518091039020604
0518060a00160405290816000820
180546101819061100b565b80601f
0160208091040260200160405190
8101604052809291908181526020
018280546101ad9061100b565b80
156101fa5780601f106101cf576101
0080835404028352916020019161
01fa565b820191906000526020600
020905b815481529060010190602
0018083116101dd57829003601f16
8201915b50505050508152602001
6001820180546102139061100b56
5b80601f016020809104026020016
0405190810160405280929190818
15260200182805461023f9061100b



565b801561028c5780601f1061026
1576101008083540402835291602
0019161028c565b8201919060005
26020600020905b8154815290600
1019060200180831161026f578290
03601f168201915b5050505050815
26020016002820180546102a5906
1100b565b80601f01602080910402
6020016040519081016040528092
91908181526020018280546102d1
9061100b565b801561031e578060
1f106102f35761010080835404028
352916020019161031e565b82019
1906000526020600020905b81548
1529060010190602001808311610
30157829003601f168201915b5050
5050508152602001600382015481
5260200160048201600090549061
01000a900460ff166002811115610
35657610355610d54565b5b60028



1111561036857610367610d54565
b5b8152505090508060200151816
0400151826060015183608001519
4509450945094505091935091935
65b80600160028111156103a7576
103a6610d54565b5b60018260405
16103b79190610fc5565b90815260
2001604051809103902060040160
009054906101000a900460ff16600
28111156103e8576103e7610d545
65b5b14610428576040517f08c379
a000000000000000000000000000
0000000000000000000000000000
00815260040161041f906110ae565
b60405180910390fd5b6002600183
60405161043a9190610fc5565b908
1526020016040518091039020600
40160006101000a81548160ff0219
169083600281111561046d576104
6c610d54565b5b02179055508160



40516104809190610fc5565b60405
180910390207f10903cdad98900af
2897d7e63e43696fd2ff20b6b781cc
35e32cc9ecada0c2ec60405160405
180910390a25050565b600181805
1602081018201805184825260208
3016020850120818352809550505
0505050600091509050806000018
0546104f19061100b565b80601f01
6020809104026020016040519081
0160405280929190818152602001
82805461051d9061100b565b8015
61056a5780601f1061053f5761010
0808354040283529160200191610
56a565b820191906000526020600
020905b815481529060010190602
00180831161054d57829003601f16
8201915b50505050509080600101
805461057f9061100b565b80601f0
1602080910402602001604051908



1016040528092919081815260200
18280546105ab9061100b565b801
56105f85780601f106105cd576101
0080835404028352916020019161
05f8565b820191906000526020600
020905b815481529060010190602
0018083116105db57829003601f16
8201915b50505050509080600201
805461060d9061100b565b80601f0
1602080910402602001604051908
1016040528092919081815260200
18280546106399061100b565b801
56106865780601f1061065b576101
0080835404028352916020019161
0686565b82019190600052602060
0020905b81548152906001019060
200180831161066957829003601f1
68201915b5050505050908060030
1549080600401600090549061010
00a900460ff16905085565b600080



54906101000a900473fffffffffffffffff
fffffffffffffffffffffffff1681565b600080
54906101000a900473fffffffffffffffff
fffffffffffffffffffffffff1673ffffffffffffff
fffffffffffffffffffffffff163373ffffffffffff
fffffffffffffffffffffffff161461075b5
76040517f08c379a000000000000000
00000000000000000000000000000000
000000000000000000000815260040161
075290611140565b604051809103
90fd5b600060018460405161076d9
190610fc5565b9081526020016040
5180910390206000018054610789
9061100b565b9050146107cb5760
40517f08c379a0000000000000000000
00000000000000000000000000000000
0000000000000000081526004016107c
2906111ac565b60405180910390fd
5b6040518060a001604052808481
5260200183815260200182815260



2001428152602001600060028111
1561080257610801610d54565b5b
8152506001846040516108159190
610fc5565b9081526020016040518
0910390206000820151816000019
0816108389190611378565b50602
082015181600101908161084e919
0611378565b50604082015181600
20190816108649190611378565b5
0606082015181600301556080820
1518160040160006101000a81548
160ff021916908360028111156108
9a57610899610d54565b5b021790
5550905050826040516108b09190
610fc5565b60405180910390207f1
deb5b3f1332885a208a3b03c4438c
7daad01496d32418f03019bade53c
0841e8383426040516108eb93929
19061144a565b60405180910390a
2505050565b60008054906101000



a900473ffffffffffffffffffffffffffffffffffff
ffffff1673ffffffffffffffffffffffffffffffffffff
ffffff163373ffffffffffffffffffffffffffffffffffff
ffffffffffff1614610986576040517f08
c379a0000000000000000000000000000000
000000000000000000000000000000000000
000000815260040161097d906111
40565b60405180910390fd5b60006
001826040516109989190610fc556
5b90815260200160405180910390
2060000180546109b49061100b56
5b9050116109f6576040517f08c37
9a0000000000000000000000000000000000
000000000000000000000000000000000000
00081526004016109ed906114db5
65b60405180910390fd5b60006002
811115610a0a57610a09610d5456
5b5b600182604051610a1a919061
0fc5565b908152602001604051809
1039020600401600090549061010



00a900460ff166002811115610a4b
57610a4a610d54565b5b14610a8b
576040517f08c379a0000000000000
0000000000000000000000000000
00000000000000000000081526004016
10a829061156d565b60405180910
390fd5b60018082604051610a9c91
90610fc5565b90815260200160405
1809103902060040160006101000
a81548160ff021916908360028111
15610acf57610ace610d54565b5b0
21790555080604051610ae291906
10fc5565b60405180910390207f36
8cb808019f63c8cdd7e7087e42db5
b33818a846fdd80e9527664749152
415860405160405180910390a250
565b6000604051905090565b6000
80fd5b600080fd5b600080fd5b600
080fd5b6000601f19601f83011690
50919050565b7f4e487b7100000000



000000000000000000000000000000000000
0000000000000000000000000000000006000526
04160045260246000fd5b610b8082
610b37565b810181811067ffffffffff
ffffff82111715610b9f57610b9e610
b48565b5b80604052505050565b6
000610bb2610b19565b9050610bb
e8282610b77565b919050565b600
067fffffffffffffffffff821115610bde576
10bdd610b48565b5b610be782610
b37565b905060208101905091905
0565b82818337600083830152505
050565b6000610c16610c1184610
bc3565b610ba8565b90508281526
0208101848484011115610c32576
10c31610b32565b5b610c3d84828
5610bf4565b509392505050565b60
0082601f830112610c5a57610c596
10b2d565b5b8135610c6a8482602
08601610c03565b9150509291505



0565b600060208284031215610c8
957610c88610b23565b5b6000820
13567fffffffffffffffff811115610ca75
7610ca6610b28565b5b610cb3848
28501610c45565b9150509291505
0565b600081519050919050565b6
0008282526020820190509291505
0565b60005b83811015610cf65780
8201518184015260208101905061
0cdb565b60008484015250505050
565b6000610d0d82610cbc565b61
0d178185610cc7565b9350610d27
818560208601610cd8565b610d30
81610b37565b8401915050929150
50565b6000819050919050565b61
0d4e81610d3b565b82525050565b
7f4e487b7100000000000000000000
000000000000000000000000000000
00000000006000526021600452602
46000fd5b60038110610d9457610d



93610d54565b5b50565b60008190
50610da582610d83565b91905056
5b6000610db582610d97565b9050
919050565b610dc581610daa565b
82525050565b6000608082019050
8181036000830152610de5818761
0d02565b90508181036020830152
610df98186610d02565b9050610e0
86040830185610d45565b610e156
060830184610dbc565b959450505
05050565b600060a082019050818
1036000830152610e388188610d0
2565b90508181036020830152610
e4c8187610d02565b90508181036
040830152610e608186610d02565
b9050610e6f6060830185610d4556
5b610e7c6080830184610dbc565b
9695505050505050565b600073ffff
ffffffffffffffffffffffffffffffffffff821690
50919050565b6000610eb182610e



86565b9050919050565b610ec181
610ea6565b82525050565b600060
2082019050610edc600083018461
0eb8565b92915050565b60008060
0060608486031215610efb57610ef
a610b23565b5b600084013567ffffff
fffffffffff811115610f1957610f18610
b28565b5b610f2586828701610c45
565b935050602084013567ffffffffff
ffffff811115610f4657610f45610b2
8565b5b610f5286828701610c4556
5b925050604084013567fffffffffffff
ff811115610f7357610f72610b2856
5b5b610f7f86828701610c45565b9
150509250925092565b600081905
092915050565b6000610f9f82610c
bc565b610fa98185610f89565b935
0610fb9818560208601610cd8565b
80840191505092915050565b6000
610fd18284610f94565b915081905



092915050565b7f4e487b71000000
000000000000000000000000000000
0000000000000000000000000600052
602260045260246000fd5b6000600
2820490506001821680611023576
07f821691505b6020821081036110
3657611035610fdc565b5b5091905
0565b7f4974656d206973206e6f74
207665726966696564206f7220616
c72656164792060008201527f636f
6e73756d6564000000000000000000
000000000000000000000000000000
0000602082015250565b60006110
98602883610cc7565b91506110a3
8261103c565b6040820190509190
50565b6000602082019050818103
60008301526110c78161108b565b
9050919050565b7f4f6e6c7920636f
6e7472616374206f776e657220636
16e2063616c6c207468696000820



1527f73000000000000000000000000
0000000000000000000000000000000
000000000000602082015250565b
600061112a602183610cc7565b91
50611135826110ce565b60408201
9050919050565b60006020820190
5081810360008301526111598161
111d565b9050919050565b7f49746
56d20616c7265616479206578697
37473000000000000000000000000
000600082015250565b600061119
6601383610cc7565b91506111a18
2611160565b60208201905091905
0565b60006020820190508181036
0008301526111c581611189565b9
050919050565b600081905081600
05260206000209050919050565b6
0006020601f830104905091905056
5b600082821b905092915050565b
60006008830261122e7ffffffffffffff

ff
ffff826111f1565b61123886836111f
1565b95508019841693508086168
417925050509392505050565b600
0819050919050565b60006112756
1127061126b84610d3b565b61125
0565b610d3b565b9050919050565
b6000819050919050565b61128f83
61125a565b6112a361129b826112
7c565b8484546111fe565b8255505
05050565b600090565b6112b8611
2ab565b6112c3818484611286565
b505050565b5b818110156112e75
76112dc6000826112b0565b60018
10190506112c9565b5050565b601f
82111561132c576112fd816111cc5
65b611306846111e1565b8101602
0851015611315578190505b61132
9611321856111e1565b830182611
2c8565b50505b505050565b60008



2821c905092915050565b6000611
34f60001984600802611331565b19
80831691505092915050565b6000
611368838361133e565b91508260
02028217905092915050565b6113
8182610cbc565b67ffffffffffffffff81
111561139a57611399610b48565b
5b6113a4825461100b565b6113af8
282856112eb565b6000602090506
01f8311600181146113e257600084
156113d0578287015190505b6113
da858261135c565b865550611442
565b601f1984166113f0866111cc5
65b60005b8281101561141857848
9015182556001820191506020850
194506020810190506113f3565b86
8310156114355784890151611431
601f89168261133e565b8355505b6
001600288020188555050505b505
050505050565b600060608201905



0818103600083015261146481866
10d02565b9050818103602083015
26114788185610d02565b9050611
4876040830184610d45565b94935
0505050565b7f4974656d20646f65
73206e6f742065786973740000000
0000000000000000000000600082015
250565b60006114c5601383610cc
7565b91506114d08261148f565b60
2082019050919050565b60006020
8201905081810360008301526114f
4816114b8565b9050919050565b7f
4974656d20697320616c72656164
79207665726966696564206f72206
36f6e7360008201527f756d656400
00000000000000000000000000000
000000000000000000000000000060
2082015250565b60006115576024
83610cc7565b9150611562826114f
b565b604082019050919050565b6



0006020820190508181036000830
1526115868161154a565b9050919
05056fea264697066735822122072
93306a7cb31a50468fcc254553e1f
a266d31971102f817aa2c15a2ea91
ffb864736f6c63430008120033

OUTPUT:

