

## TOOLS AND TECHNOLOGY REQUIREMENT

### Download VS Code

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AgricultureRegistry {
    struct foodProduct {
        string name;
        string description;
        uint256 quantity;
        address owner;
    }

    mapping(uint256 => foodProduct) public products;
    uint256 public productCount;

    event ProductAdded(uint256 productId, string name, string description,
uint256 quantity, address owner);
    event ProductUpdated(uint256 productId, string name, string description,
uint256 quantity);

    modifier onlyOwner(uint256 _productId) {
        require(products[_productId].owner == msg.sender, "Only the owner can
perform this action");
        _;
    }

    function addProduct(uint256 ProductId, string memory _name, string memory
_description, uint256 _quantity) external {

        products[ProductId] = foodProduct(_name, _description, _quantity,
msg.sender);
        productCount++;
        emit ProductAdded(productCount, _name, _description, _quantity,
msg.sender);
    }

    function updateProduct(uint256 _productId, string memory _name, string
memory _description, uint256 _quantity) external onlyOwner(_productId) {
        foodProduct storage product = products[_productId];
        product.name = _name;
        product.description = _description;
        product.quantity = _quantity;
    }
}
```

```
    emit ProductUpdated(_productId, _name, _description, _quantity);
}

function getProductDetails(uint256 _productId) external view returns
(string memory name, string memory description, uint256 quantity, address
owner) {
    foodProduct memory product = products[_productId];
    return (product.name, product.description, product.quantity,
product.owner);
}
}
```