// SPDX-License-Identifier: UNLICENSE

pragma solidity >=0.7.0;

contract Bank

{

    mapping(address => uint) public user\_account;

    mapping(address => bool) public user\_exist;

    function create\_account() public payable returns(string memory)

    {

        require(user\_exist[msg.sender] == false, "Account Already Created");

        user\_account[msg.sender] = msg.value;

        user\_exist[msg.sender] = true;

        return "Account Created Successfully ";

    }

    function deposit(uint amount) public payable returns(string memory)

    {

        require(user\_exist[msg.sender] == true, "Account not created!");

        require(amount > 0, "Amount should be greater than 0");

        user\_account[msg.sender] += amount;

        return "Amount deposisted sucessfully";

    }

    function withdraw(uint amount) public payable returns(string memory)

    {

        require(user\_exist[msg.sender] == true, "Account Not Created");

        require(amount > 0, "Amount Must Be Greater Then 0");

        require(user\_account[msg.sender] >= amount, "Amount Must Be Less Or Equal Then Diposited Amount");

        user\_account[msg.sender] -= amount;

        return "Amount Withdraw Successfully ";

    }

    function account\_balance() public view returns(uint)

    {

        return user\_account[msg.sender];

    }

    function account\_exist() public view returns(bool)

    {

        return user\_exist[msg.sender];

    }

}