```solidity
// 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];
    }
}
```