// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract Bank {

    address public owner;

    mapping(address => uint) public balances;

    // Constructor to set the contract deployer as the owner

    constructor() {

        owner = msg.sender;}

    // Function to deposit Ether into the bank account

    function deposit() public payable {

        require(msg.value > 0, "Deposit value must be greater than zero");

        balances[msg.sender] += msg.value;  }

    // Function to withdraw Ether from the bank account

    function withdraw(uint \_amount) public {

        require(balances[msg.sender] >= \_amount, "Insufficient balance");

        payable(msg.sender).transfer(\_amount);

        balances[msg.sender] -= \_amount;  }

    // Function to check the balance of the caller

    function checkBalance() public view returns (uint) {

        return balances[msg.sender];   }

    // Function to check contract's total Ether balance (only for owner)

    function getContractBalance() public view returns (uint) {

        require(msg.sender == owner, "Only owner can check contract balance");

        return address(this).balance; }}



