**Name:** Swapnil Thorat

**PRN:** 121B1C055

**Assignment No. 3**

**Code:**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract DecentralizedBank {

    // Mapping of user addresses to their Ether balances

    mapping(address => uint) private balances;

    // Event to log successful deposits

    event Deposit(address indexed user, uint amount);

    // Event to log successful withdrawals

    event Withdrawal(address indexed user, uint amount);

    // Deposit function (payable): Allows users to deposit Ether into the bank

    function deposit() public payable {

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

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

        emit Deposit(msg.sender, msg.value);  // Emit deposit event

    }

    // Withdraw function: Allows users to withdraw Ether from their balance

    function withdraw(uint \_amount) public {

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

        // Deduct the amount from the user's balance

        balances[msg.sender] -= \_amount;

        // Transfer Ether to the user

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

        emit Withdrawal(msg.sender, \_amount);  // Emit withdrawal event

    }

    // Function to check the user's balance

    function getBalance() public view returns (uint) {

        return balances[msg.sender];

    }

}

**Output:**



