ASSIGNMENT – 13

INPUT –

// SPDX-License-Identifier: Unlicensed

pragma solidity ^0.8.27;

contract MyBank

{

    mapping(address=> uint ) private \_balances;

    address public owner;

    event LogDepositeMade(address accountHoder, uint amount );

    constructor ()

     {

         owner=msg.sender;

         emit LogDepositeMade(msg.sender, 1000);

     }

        function deposite() public payable  returns (uint)

        {

        require ((\_balances[msg.sender] + msg.value) >  \_balances[msg.sender] && msg.sender!=address(0));

        \_balances[msg.sender] += msg.value;

        emit LogDepositeMade(msg.sender , msg.value);

        return \_balances[msg.sender];

        }

        function withdraw (uint withdrawAmount) public  returns (uint)

        {

                require (\_balances[msg.sender] >= withdrawAmount);

                require(msg.sender!=address(0));

                require (\_balances[msg.sender] > 0);

                \_balances[msg.sender]-= withdrawAmount;

                payable(msg.sender).transfer(withdrawAmount);

                emit LogDepositeMade(msg.sender , withdrawAmount);

                return \_balances[msg.sender];

        }

        function viewBalance() public view returns (uint)

        {

            return \_balances[msg.sender];

        }

}

OUTPUT –



