Program:

//SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

//created a smart contract that allows a user to deposit, withdraw and save ETH!!

contract p2{

//mapped the address of the caller balance in the contract

mapping(address => uint) public balances;

// whatever the user deposit is added to msg.value of the sender address we mapped above

function deposit() public payable{

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

}

//we create the fucntion of witdraw

function withdraw(uint \_amount) public{

//we create a require arg to make sure the balance of the sender is >= \_amount if not ERR

require(balances[msg.sender]>= \_amount, "Not enough ether");

//if the amount is availabe we subtract it from the sender

balances[msg.sender] -= \_amount;

//True bool is called to confirm the amount

(bool sent,) = msg.sender.call{value: \_amount}("Sent");

require(sent, "failed to send ETH");

}

function getBal() public view returns(uint){

return address(this).balance;

}

}

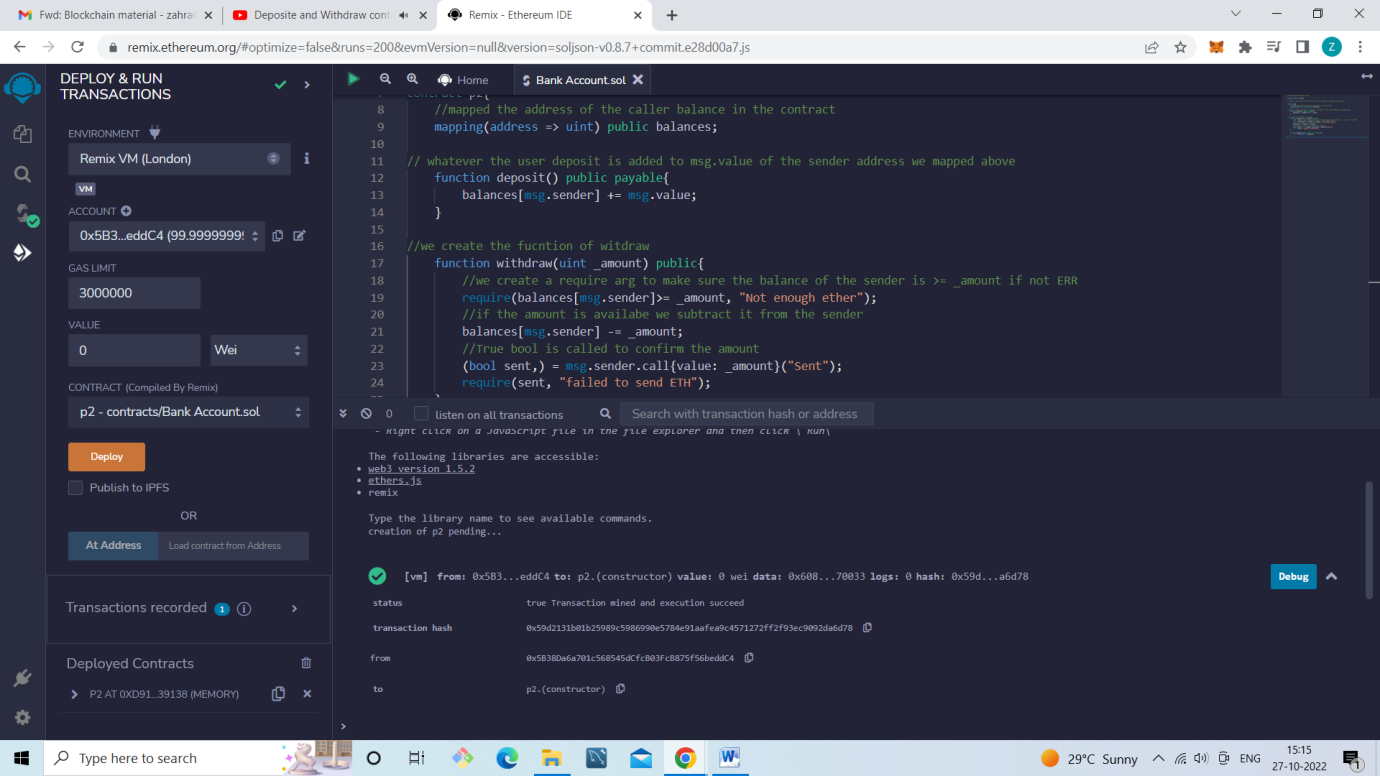
Steps:

1. Go to <https://remix.ethereum.org/>
2. In contracts folder, create a new file, name it, type the code and save it.
3. Compile and deploy.

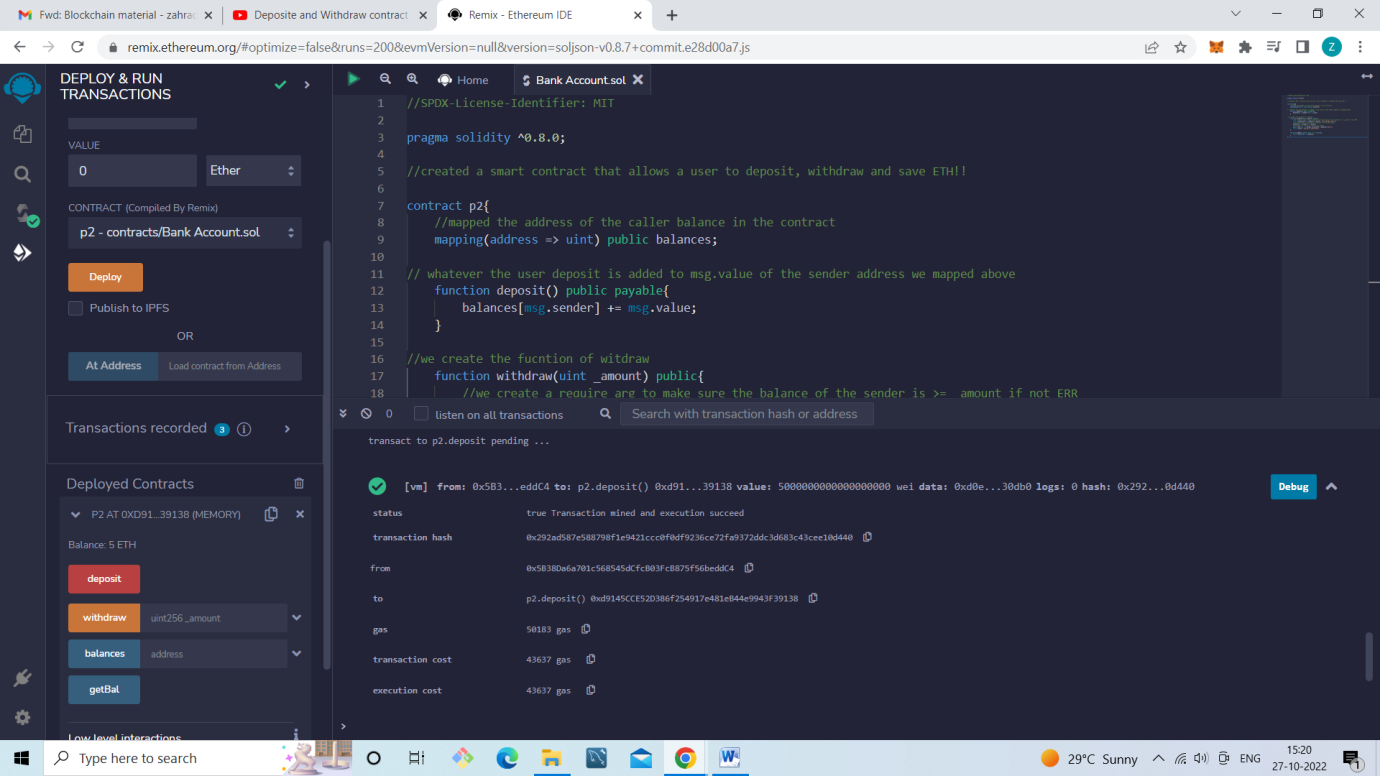
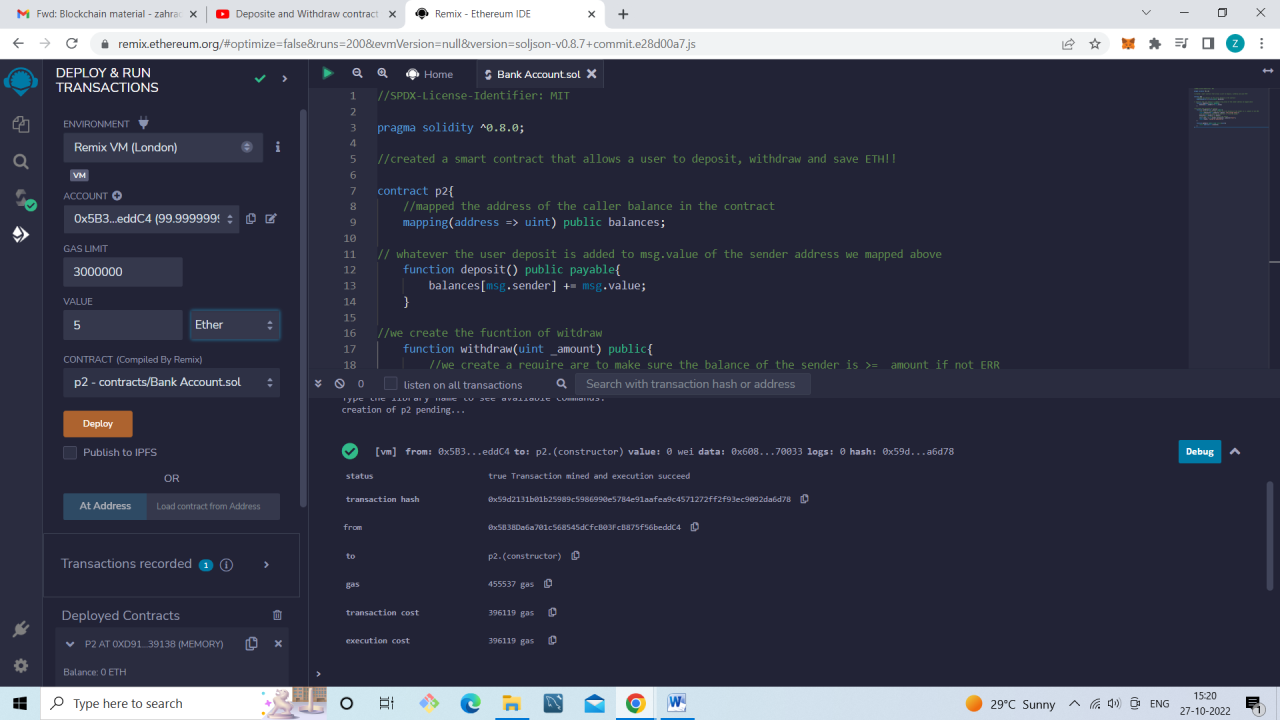
Owner: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

Contract Address: 0xd9145CCE52D386f254917e481eB44e9943F39138

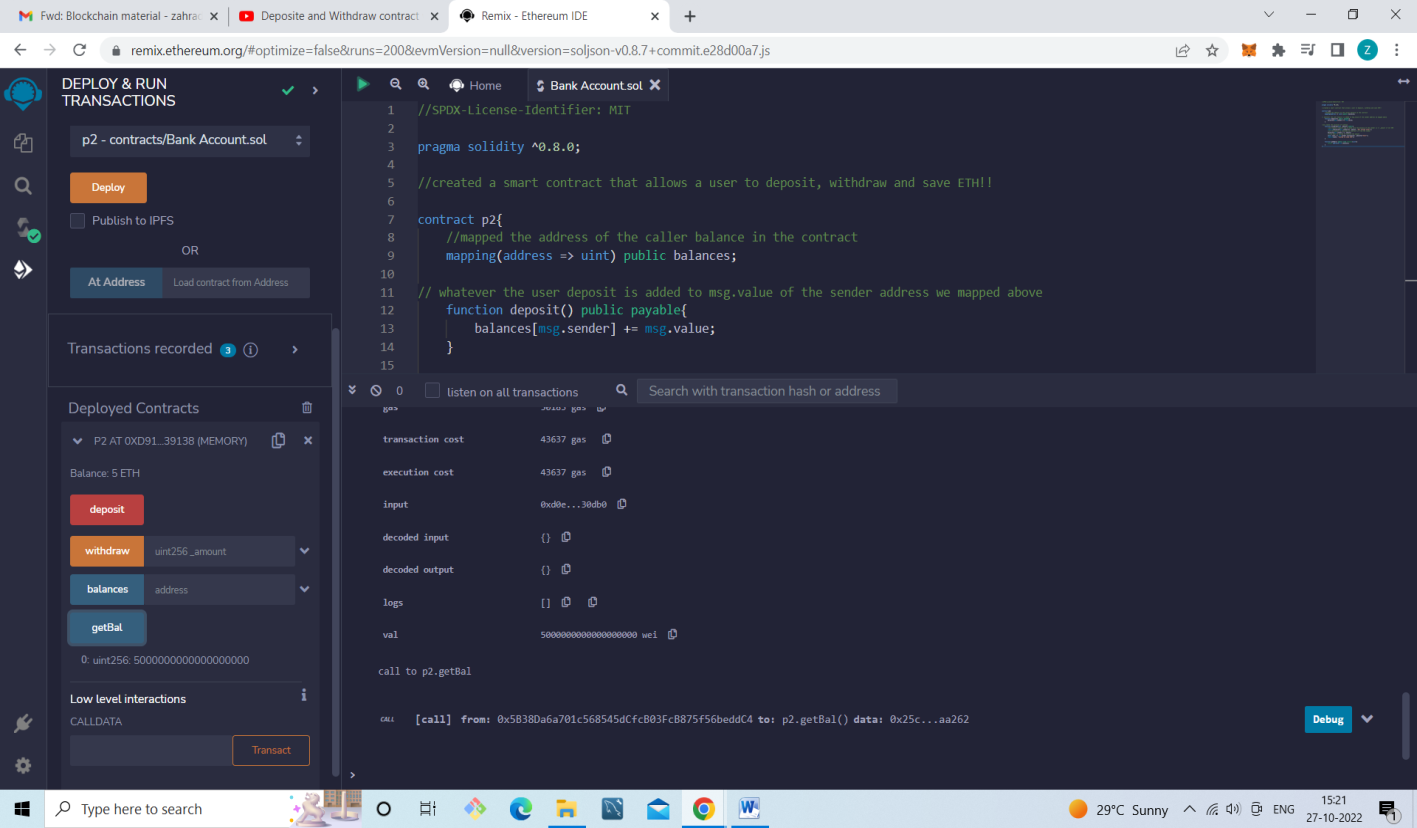
One ether = **1,000,000,000,000,000,000 wei** (1018)

Output Screenshots: 

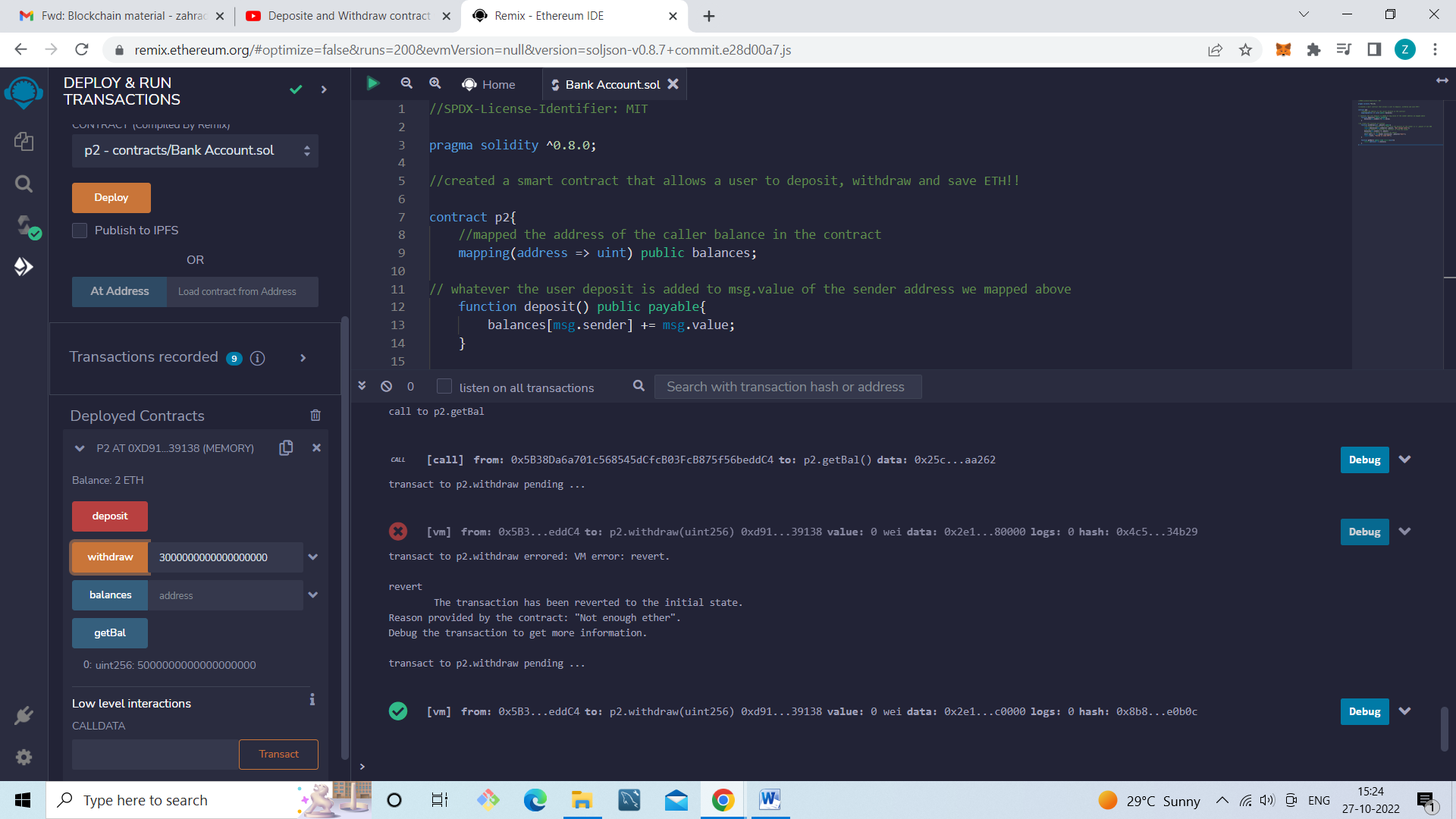
Deposit of: 5 ETHER



Checking Balance:



Withdraw of 3 ETHER:



Checking Balance after withdraw:

