pragma solidity ^0.6;



contract banking

{

 mapping(address=>uint) public user\_account;

 mapping(address=>bool) public user\_exists;

 function create\_account() public payable returns(string memory)

 {

 require(user\_exists[msg.sender]==false,'Account already created');

 if(msg.value==0)

 {

user\_account[msg.sender]=0;

user\_exists[msg.sender]=true;

 return "Account created";

 }

 require(user\_exists[msg.sender]==false,"Account already created");

user\_account[msg.sender]=msg.value;

user\_exists[msg.sender]=true;

 return "Account created";

 }

 function deposit() public payable returns(string memory)

 {

 require(user\_exists[msg.sender]==true,"Account not created");

 require(msg.value>0,"Value for deposit is Zero");

user\_account[msg.sender]=user\_account[msg.sender]+msg.value;

 return "Deposited Successfully";

 }

 function withdraw(uint amount) public payable returns(string memory)

 {

 require(user\_account[msg.sender]>amount,"Insufficient Balance");

 require(user\_exists[msg.sender]==true,"Account not created");

 require(amount>0,"Amount should be more than zero");

user\_account[msg.sender]=user\_account[msg.sender]-amount;

msg.sender.transfer(amount);

 return "Withdrawl Successful";

 }

 function transfer(address payable userAddress, uint amount) public returns(string memory)

 {

 require(user\_account[msg.sender]>amount,"Insufficient balance in Bank account");

 require(user\_exists[msg.sender]==true,"Account is not created");

 require(user\_exists[userAddress]==true,"Transfer account does not exist");

 require(amount>0,"Amount should be more than zero");

user\_account[msg.sender]=user\_account[msg.sender]-amount;

user\_account[userAddress]=user\_account[userAddress]+amount;

 return "Transfer Successful";

 }



 function send\_amt(address payable toAddress, uint256 amount) public payable returns(string



memory)

 {

 require(user\_account[msg.sender]>amount,"Insufficeint balance in Bank account");

 require(user\_exists[msg.sender]==true,"Account is not created");

 require(amount>0,"Amount should be more than zero");

user\_account[msg.sender]=user\_account[msg.sender]-amount;

toAddress.transfer(amount);

 return "Transfer Success";

 }

 function user\_balance() public view returns(uint)

 {

 return user\_account[msg.sender];

 }

 function account\_exist() public view returns(bool)

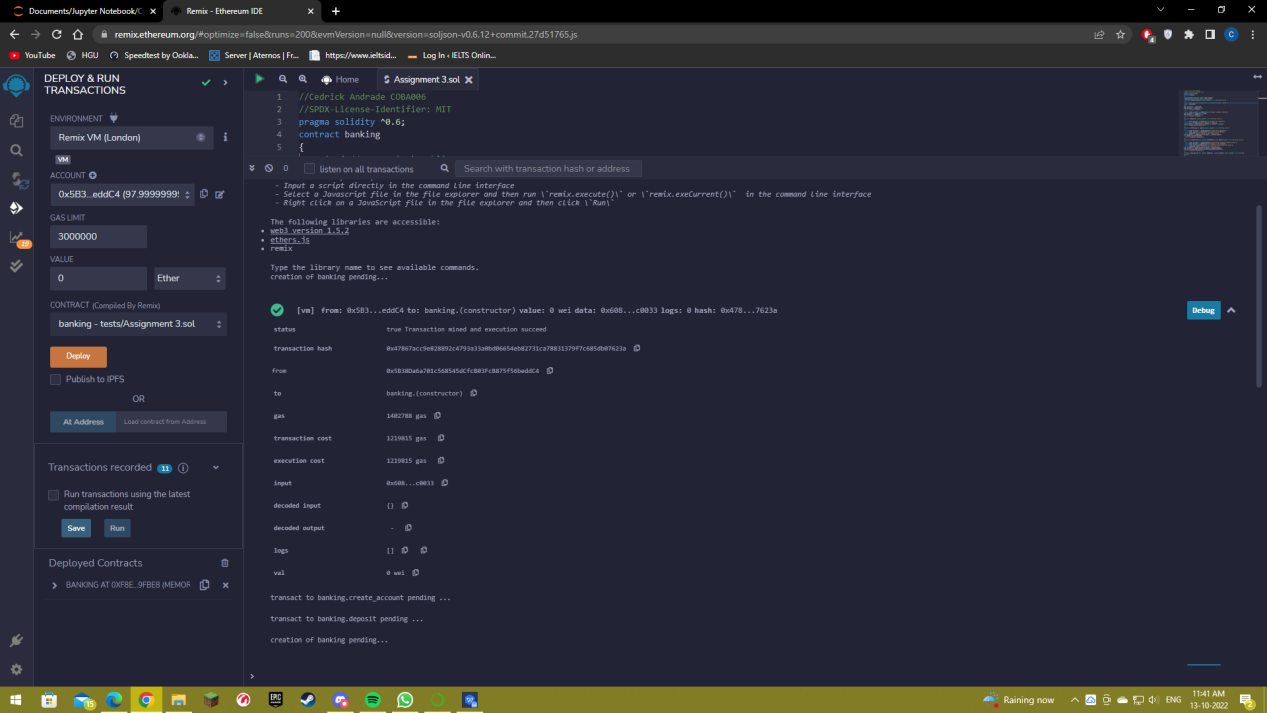
 {

 return user\_exists[msg.sender];

 }

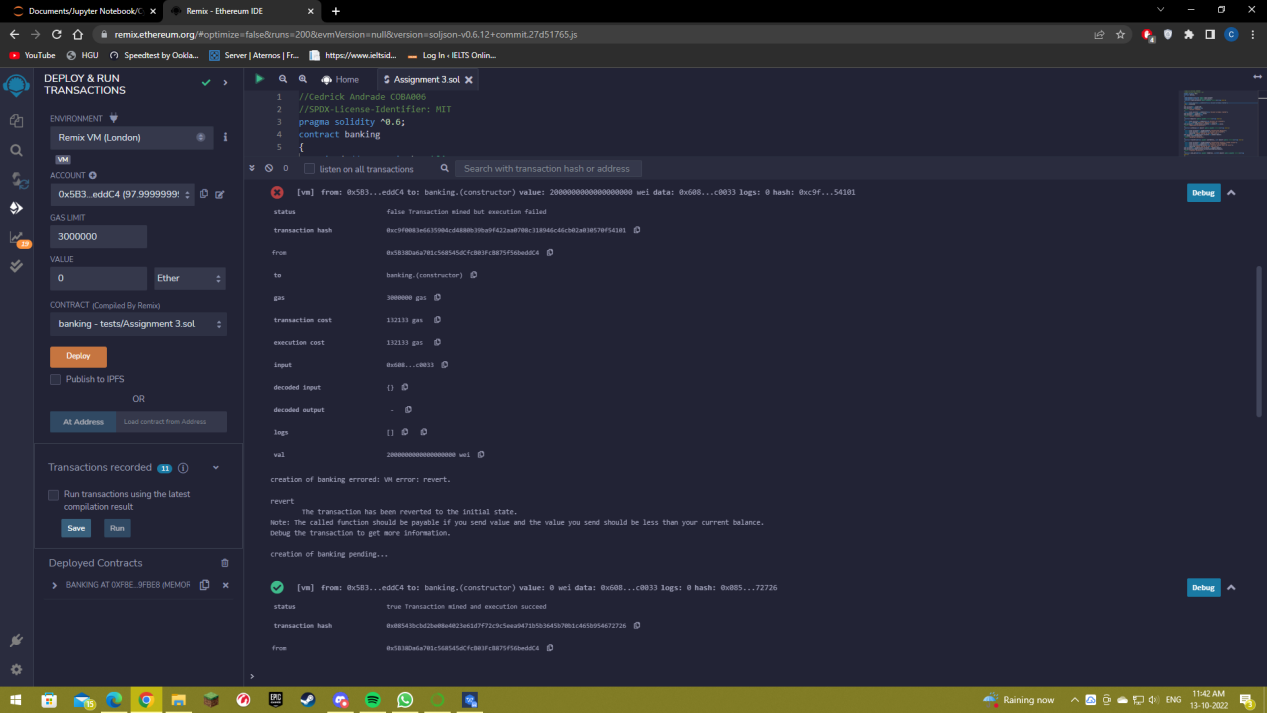
}

**Deploying Contract:**

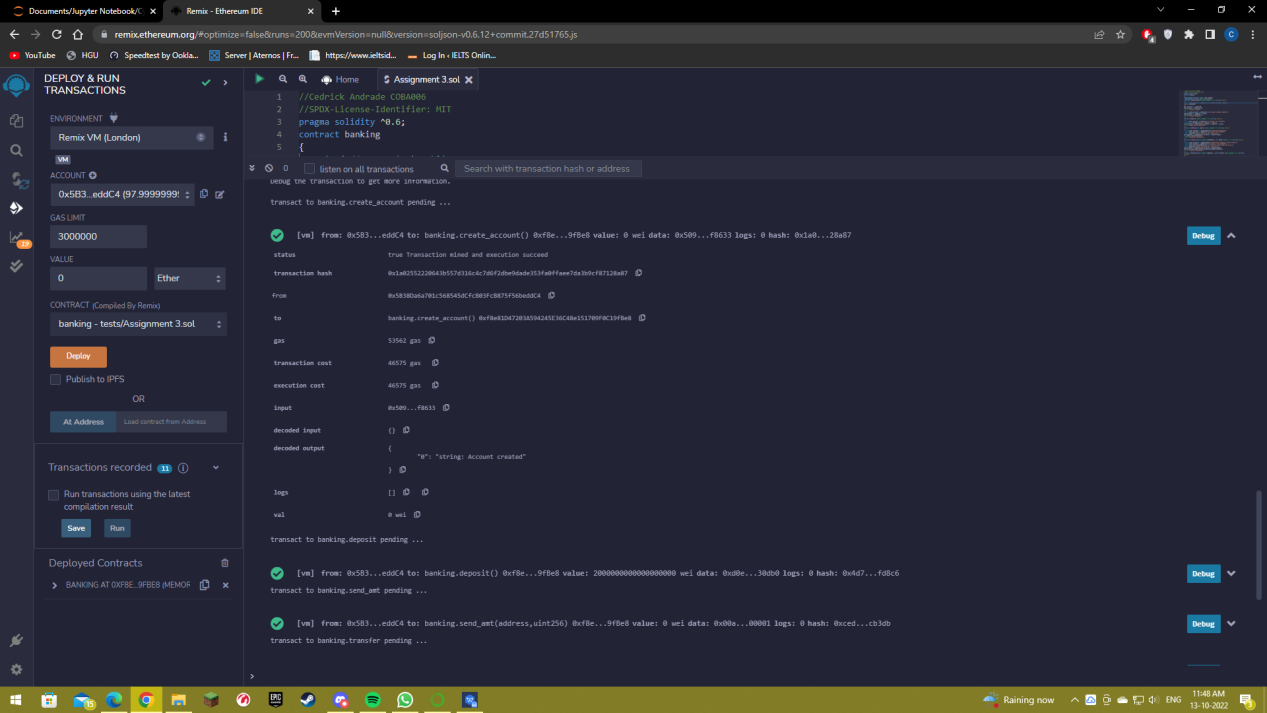




**Deposit Failure due to no Account creation:**

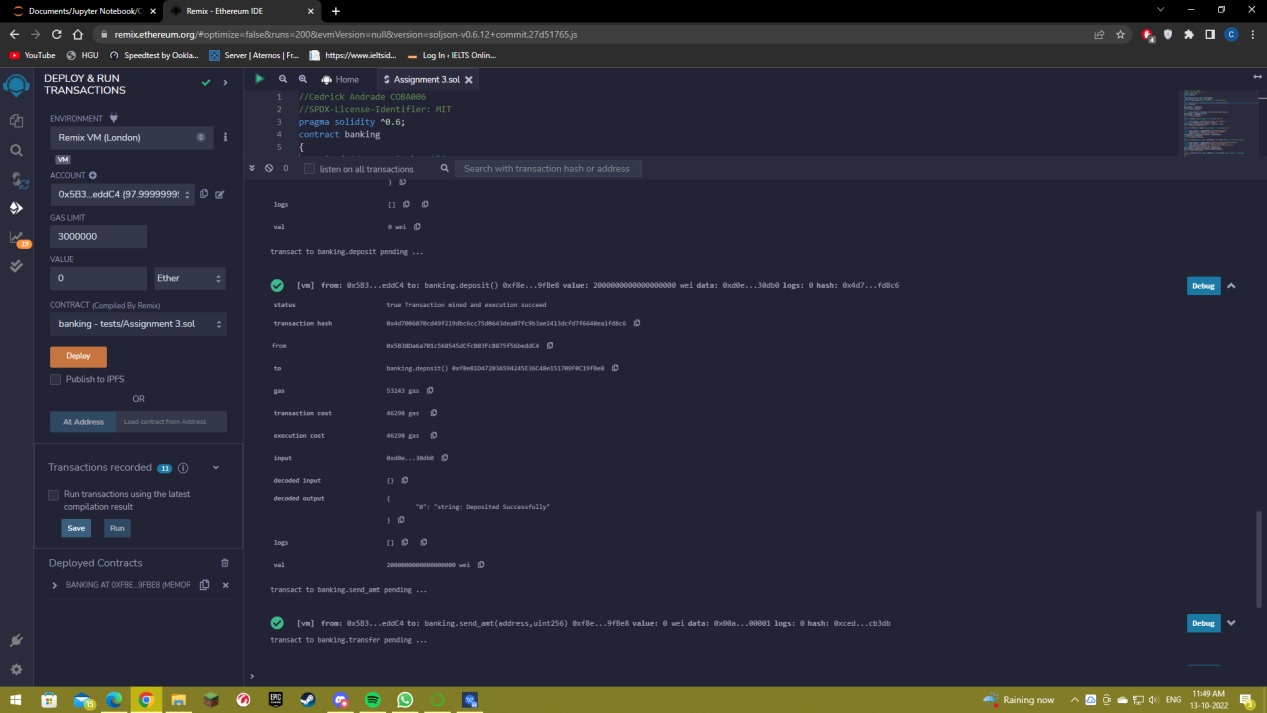


**Account Creation:**

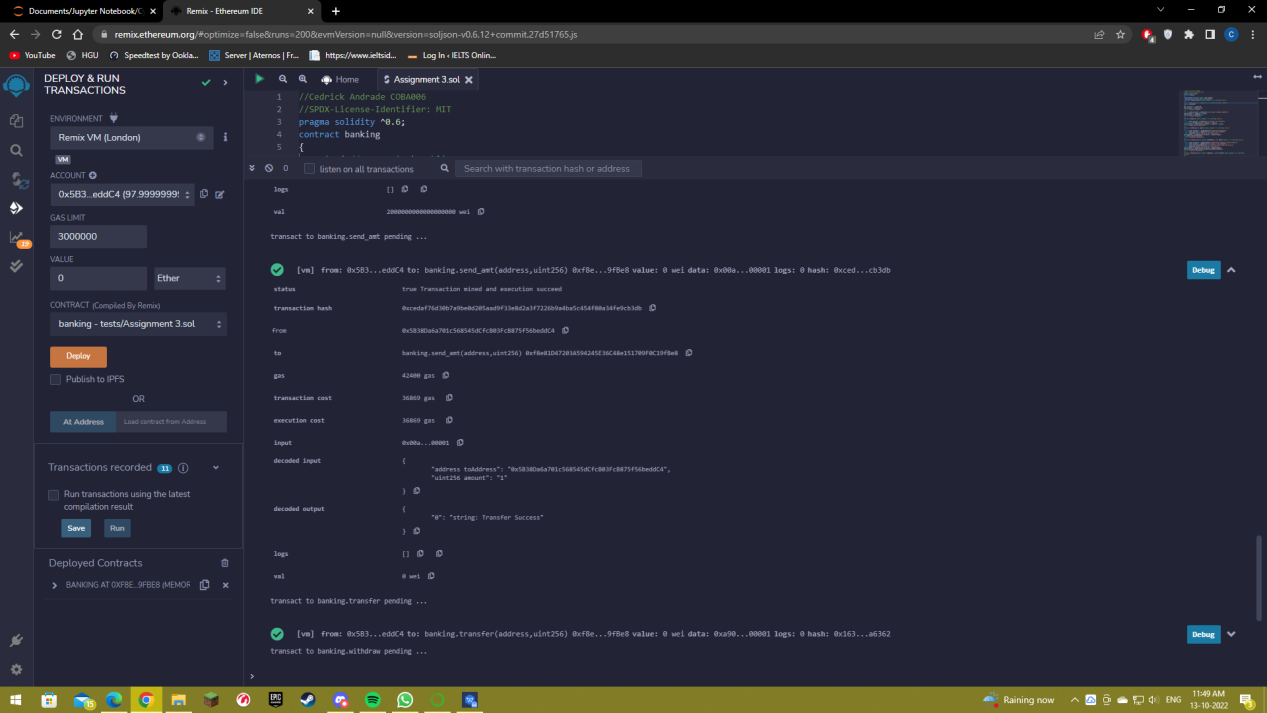




**Deposit Success:**

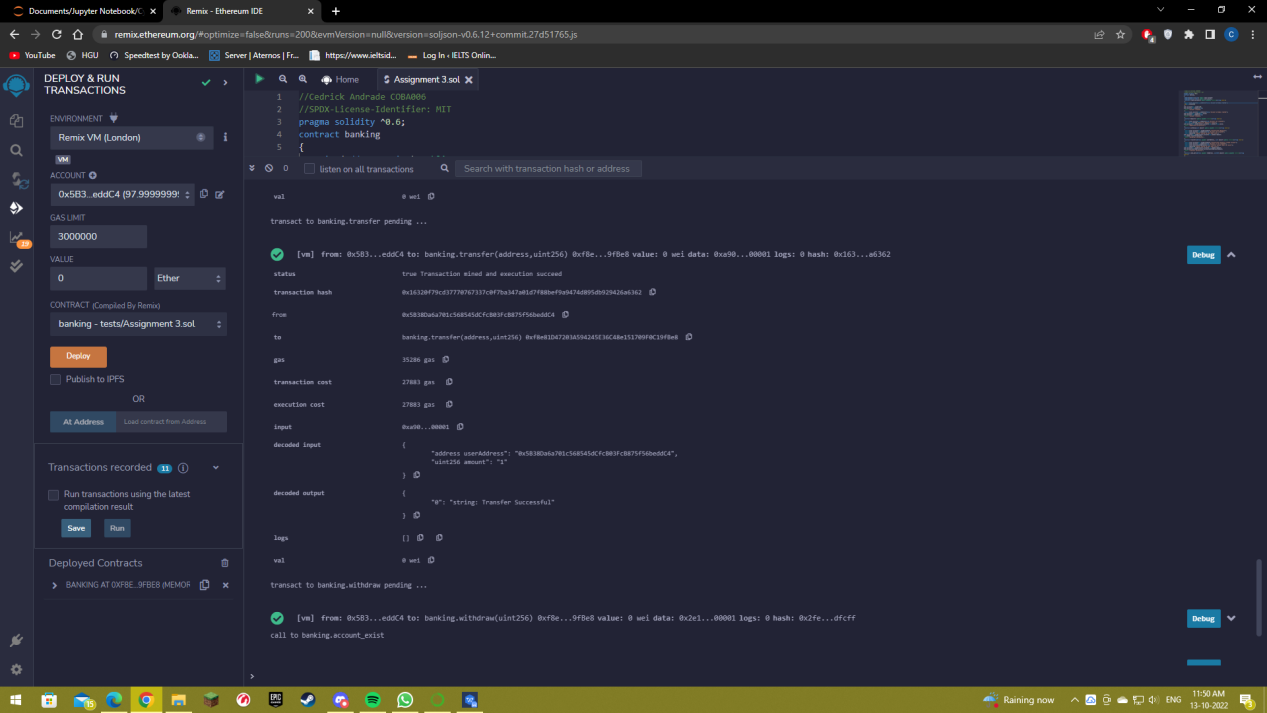


**Send Amount:**

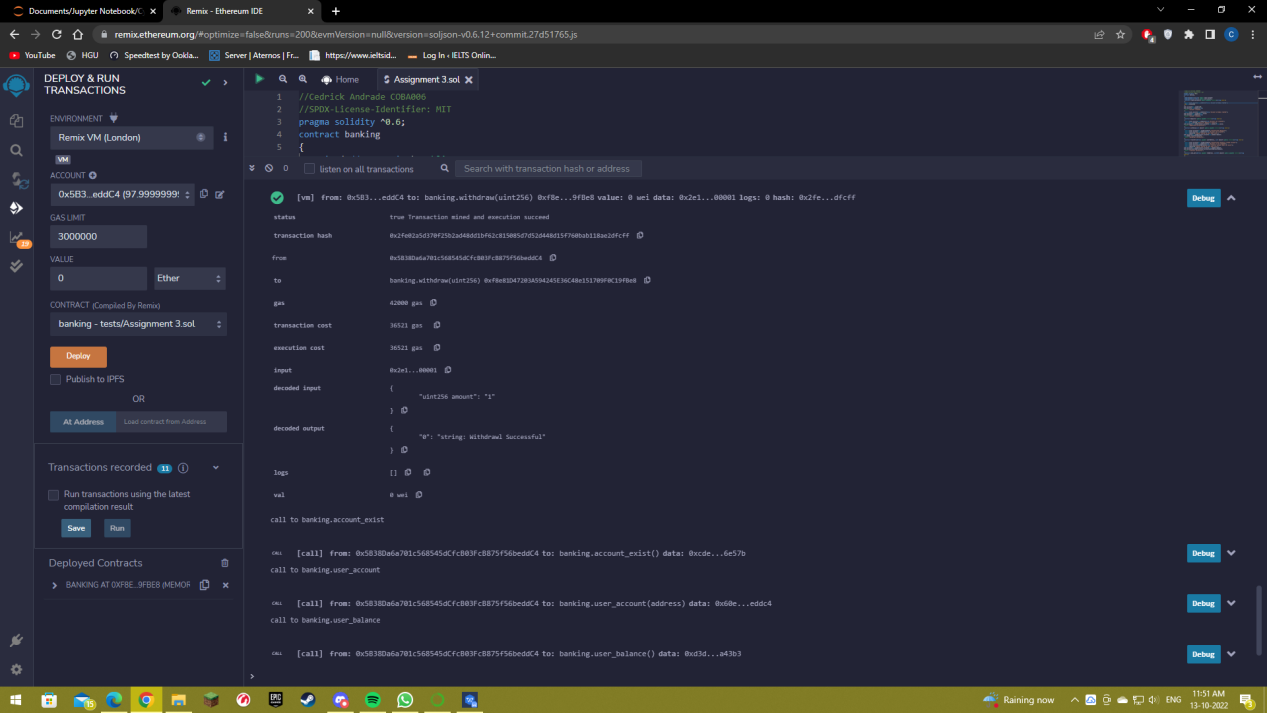




**Transfer:**

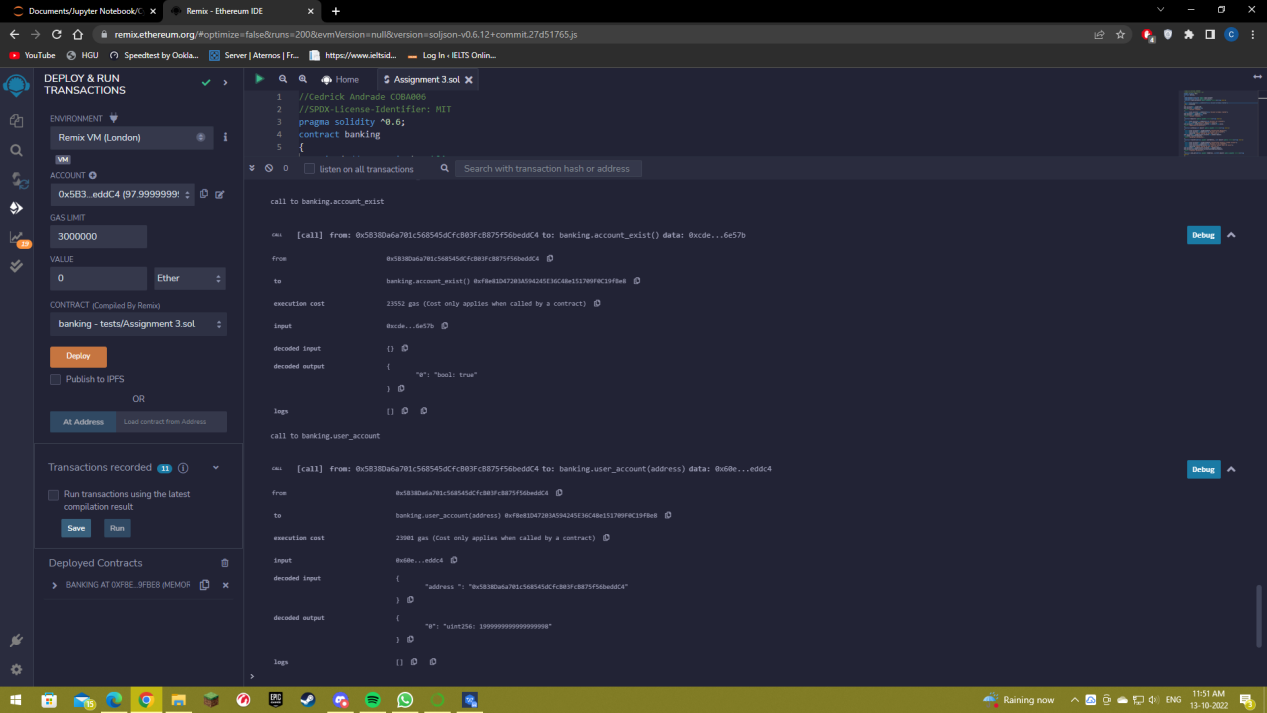


**Withdraw:**





**User account exist and bank exist functions:**



**User Balance and Exist:**

