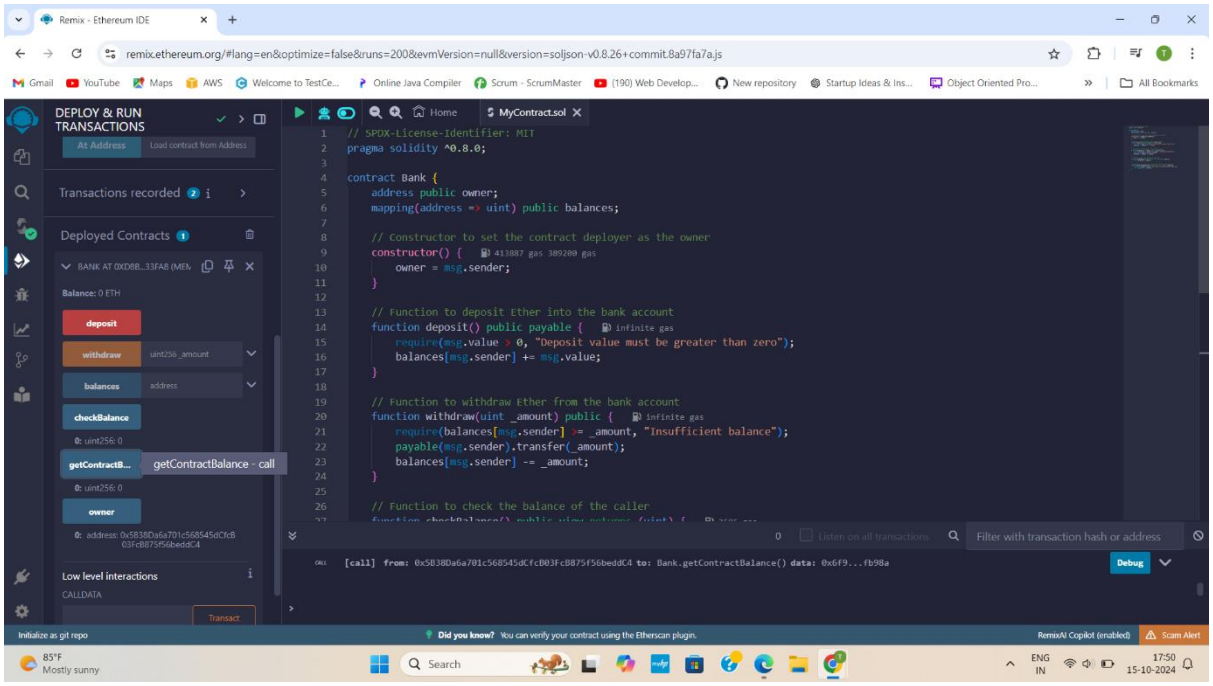
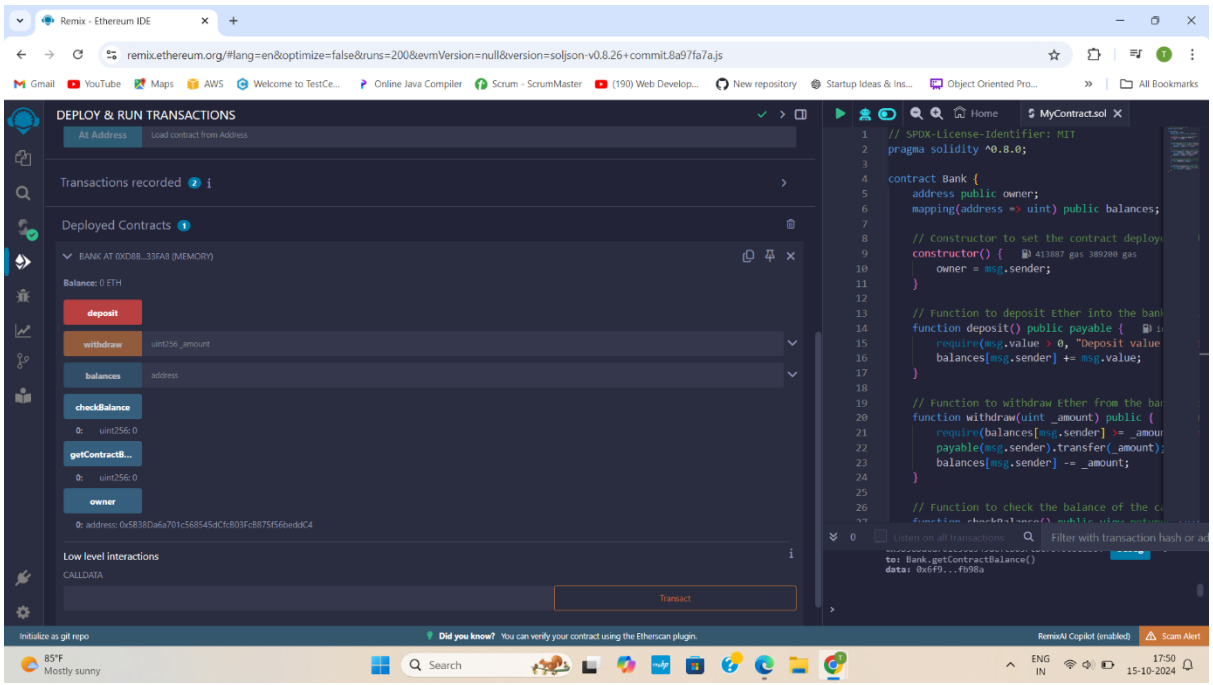
```
pragma solidity ^0.8.0;
contract Bank {
address public owner;
mapping(address => uint) public balances;

constructor() {
owner = msg.sender;}

function deposit() public payable {
require(msg.value > 0, "Deposit value must be greater than zero");
balances[msg.sender] += msg.value;  }

function withdraw(uint _amount) public {
require(balances[msg.sender] >= _amount, "Insufficient balance");
payable(msg.sender).transfer(_amount);
balances[msg.sender] -= _amount;  }

function checkBalance() public view returns (uint) {
return balances[msg.sender];   }

function getContractBalance() public view returns (uint) {
require(msg.sender == owner, "Only owner can check contract balance");
return address(this).balance; }}
```


Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js
DEPLOY & RUN TRANSACTIONS
At Address
Load contract from Address
Transactions recorded
Deployed Contracts
BANK AT 0XD8B...33FA8 (MEM
Balance: 0 ETH
deposit
withdraw
uint256 _amount
balances
address
checkBalance
0: uint256: 0
getContractB...
getContractBalance - call
0: uint256: 0
owner
0: address: 0x5B38Da6a701c568545dCfcB
03FcB875f56beddC4
Low level interactions
CALLDATA
Transact
MyContract.sol
// 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
[call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: Bank.getContractBalance() data: 0x6f9...fb98a
Debug
Initialize as git repo
Did you know? You can verify your contract using the Etherscan plugin.
RemixAI Copilot (enabled)
Scam Alert
85°F
Mostly sunny
Search
ENG
IN
17:50
15-10-2024


Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js
DEPLOY & RUN TRANSACTIONS
At Address
Load contract from Address
Transactions recorded
Deployed Contracts
BANK AT 0XD8B...33FA8 (MEMORY)
Balance: 0 ETH
deposit
withdraw
uint256 _amount
balances
address
checkBalance
0: uint256: 0
getContractB...
0: uint256: 0
owner
0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
Low level interactions
CALLDATA
Transact
MyContract.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract Bank {
address public owner;
mapping(address => uint) public balances;
// Constructor to set the contract deploy
constructor() {
owner = msg.sender;
}
// Function to deposit Ether into the ban
function deposit() public payable {
require(msg.value > 0, "Deposit value
balances[msg.sender] += msg.value;
}
// Function to withdraw Ether from the ba
function withdraw(uint _amount) public {
require(balances[msg.sender] >= _amou
payable(msg.sender).transfer(_amount);
balances[msg.sender] -= _amount;
}
// Function to check the balance of the c
to: Bank.getContractBalance()
data: 0x6f9...fb98a
Initialize as git repo
Did you know? You can verify your contract using the Etherscan plugin.
RemixAI Copilot (enabled)
Scam Alert
85°F
Mostly sunny
Search
ENG
IN
17:50
15-10-2024