Developing a Smart Contract for Land Registry Using Blockchain

Saket Chaudhary

Veer Bahadur Singh Purvanchal University

B.Tech. (CSE)

PROBLEM STATEMENT

The traditional land registry system is prone to fraud, inefficiencies, and unauthorized access. Paper-based records and centralized databases are vulnerable to tampering, mismanagement, and lack of transparency. There is a need for a secure, decentralized, and transparent system to store and manage land ownership data.

Project Description

This project aims to develop a blockchain-based land registry system using smart contracts. Blockchain technology ensures a tamper-proof, decentralized ledger that enhances security, transparency, and accessibility. The smart contract automates property registration and ownership transfers, reducing fraud and inefficiencies. By leveraging Ethereum and Solidity, the project ensures that land ownership records remain immutable, reducing dependency on intermediaries and increasing trust among stakeholders.

Key Features:

- *Decentralized Storage:* Eliminates the risk of data manipulation.
- *Enhanced Security*: Ensures data integrity through cryptographic hashing.
- *Improved Access*: Provides a transparent and efficient property registration system.
- *Smart Contract Automation:* Reduces manual intervention, ensuring faster and error-free transactions.

WHO ARE THE END USERS?

- Government agencies and land registration authorities
- Property owners and real estate investors
- Legal professionals and law firms
- Blockchain developers and researchers

Technology Used

- Blockchain Platform: Ethereum
- Programming Language: Solidity
- Development Environment: Remix IDE
- Smart Contract Deployment: Ethereum Virtual Machine (EVM)
- Security Features: Cryptographic hashing and digital signatures

RESULTS

- Successfully developed and deployed a smart contract for land registration.
- Demonstrated a secure, transparent, and immutable ledger for property transactions.
- Automated property ownership transfers, reducing manual errors and delays.
- Enhanced data security, eliminating unauthorized modifications.
- Potential for integration with government and legal systems for wider adoption.

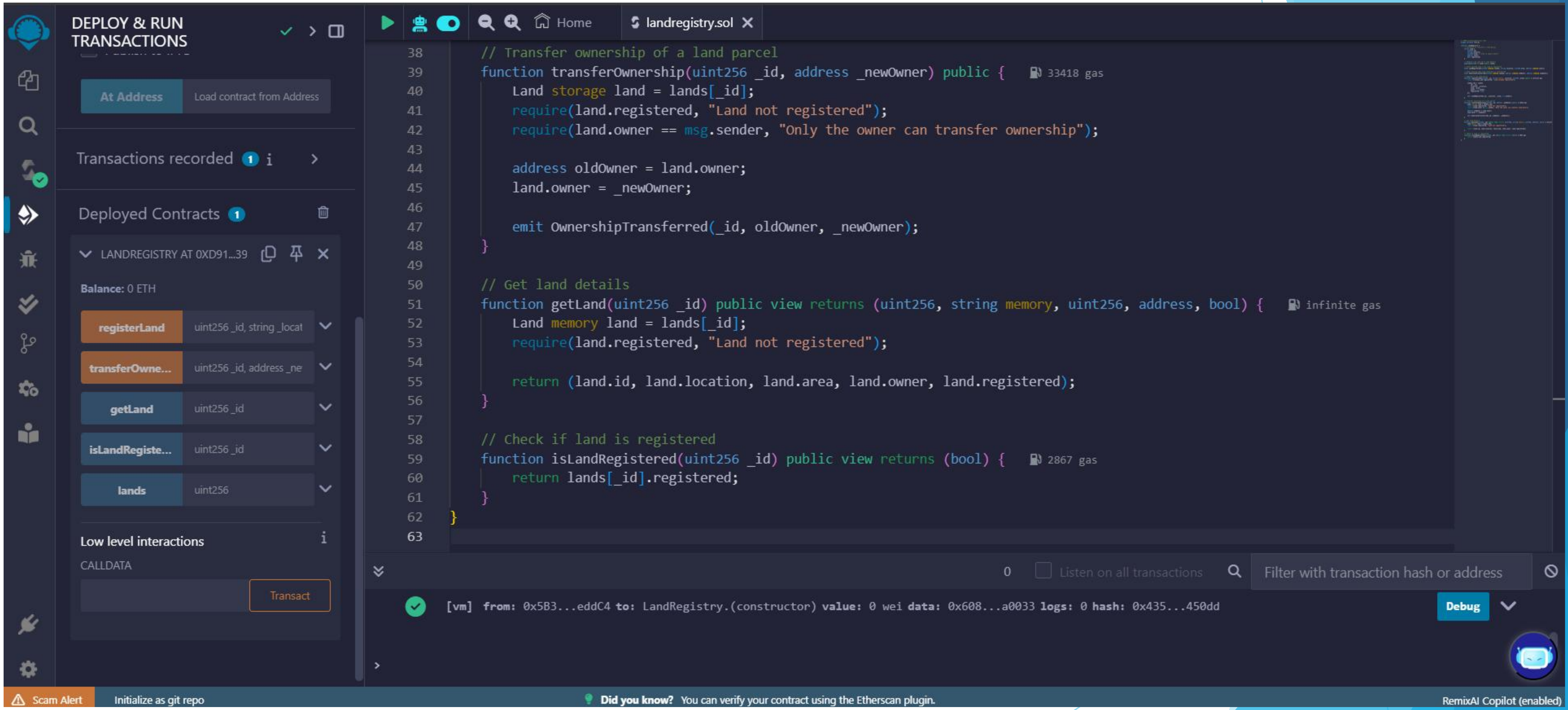
GitHub Repository

For more details and the complete code for the Blockchain-based Land Registry, visit the GitHub repository:

[Blockchain-Land-Registry]

SnapShots

Deployment Screen:



DEPLOY & RUN TRANSACTIONS

At Address | Load contract from Address

Transactions recorded 1

Deployed Contracts 1

LANDREGISTRY AT 0XD91...39

Balance: 0 ETH

registerLand	uint256 _id, string _locat
transferOwne...	uint256 _id, address _ne
getLand	uint256 _id
isLandRegiste...	uint256 _id
lands	uint256

Low level interactions

CALLDATA

Transact

Home | landregistry.sol

```
    // Transfer ownership of a land parcel
    function transferOwnership(uint256 _id, address _newOwner) public {    33418 gas
        Land storage land = lands[_id];
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner = _newOwner;

        emit OwnershipTransferred(_id, oldOwner, _newOwner);
    }

    // Get land details
    function getLand(uint256 _id) public view returns (uint256, string memory, uint256, address, bool) {    infinite gas
        Land memory land = lands[_id];
        require(land.registered, "Land not registered");

        return (land.id, land.location, land.area, land.owner, land.registered);
    }

    // Check if land is registered
    function isLandRegistered(uint256 _id) public view returns (bool) {    2867 gas
        return lands[_id].registered;
    }
}
```

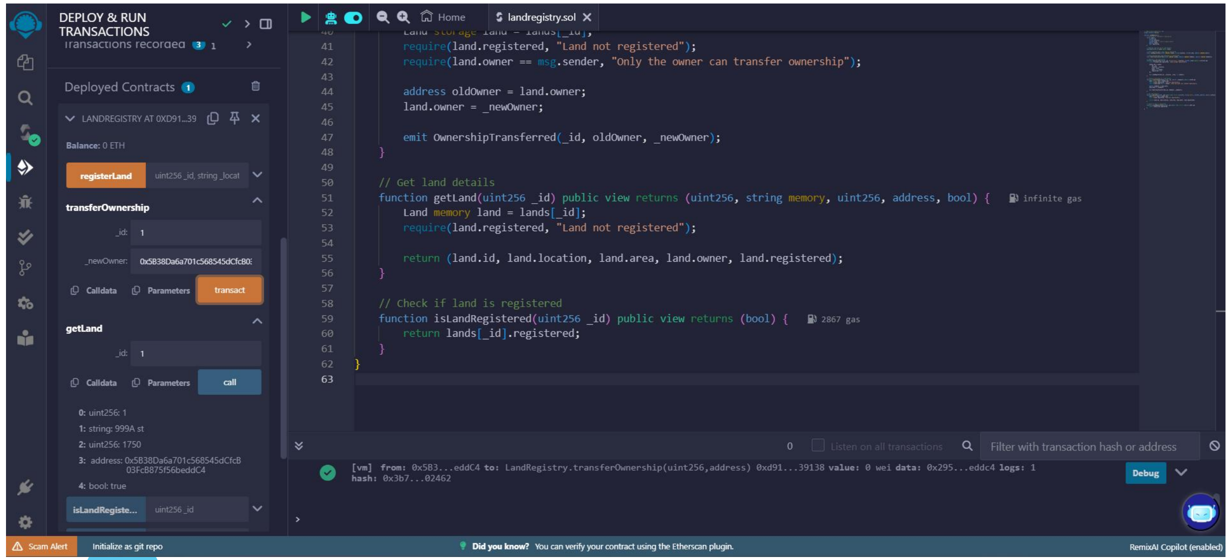
0 Listen on all transactions | Filter with transaction hash or address

```
[vm] from: 0x5B3...eddC4 to: LandRegistry.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x435...450dd
```

Debug

Scam Alert | Initialize as git repo | **Did you know?** You can verify your contract using the Etherscan plugin. | RemixAI Copilot (enabled)

Blockchain Transaction Logs:


DEPLOY & RUN TRANSACTIONS

Transactions recorded 3 1

Deployed Contracts 1

LANDREGISTRY AT 0XD91...39

Balance: 0 ETH

registerLand | uint256 _id, string _locat

transferOwnership

_id:	1
_newOwner:	0x5B38Da6a701c568545dCfcB0:

Calldata · Parameters · transact

getLand

_id:	1

Calldata · Parameters · call

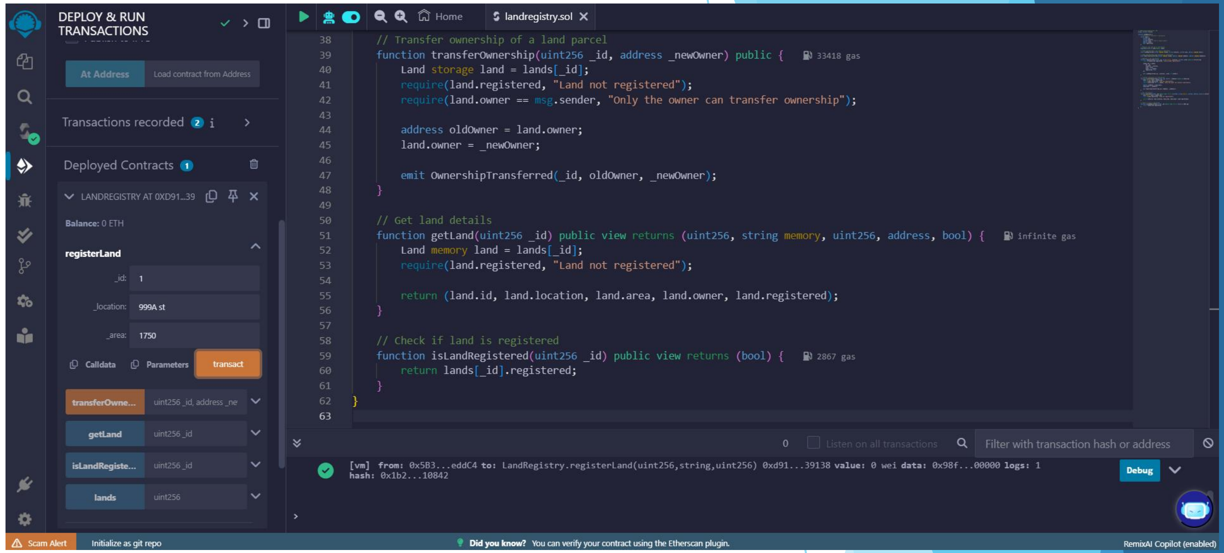
```
0: uint256: 1
1: string: 999A st
2: uint256: 1750
3: address: 0x5B38Da6a701c568545dCfcB
            03FcB875f56beddC4
4: bool: true
```

isLandRegiste... | uint256 _id

```
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner = _newOwner;

        emit OwnershipTransferred(_id, oldOwner, _newOwner);
    }

    // Get land details
    function getLand(uint256 _id) public view returns (uint256, string memory, uint256, address, bool) {    infinite gas
        Land memory land = lands[_id];
        require(land.registered, "Land not registered");

        return (land.id, land.location, land.area, land.owner, land.registered);
    }

    // Check if land is registered
    function isLandRegistered(uint256 _id) public view returns (bool) {    2867 gas
        return lands[_id].registered;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: LandRegistry.transferOwnership(uint256,address) 0xd91...39138 value: 0 wei data: 0x295...eddc4 logs: 1
hash: 0x3b7...02462
```

Did you know? You can verify your contract using the Etherscan plugin.


DEPLOY & RUN TRANSACTIONS

At Address · Load contract from Address

Transactions recorded 2 i

Deployed Contracts 1

LANDREGISTRY AT 0XD91...39

Balance: 0 ETH

registerLand

_id:	1
_location:	999A st
_area:	1750

Calldata · Parameters · transact

transferOwne... | uint256 _id, address _ne

getLand | uint256 _id

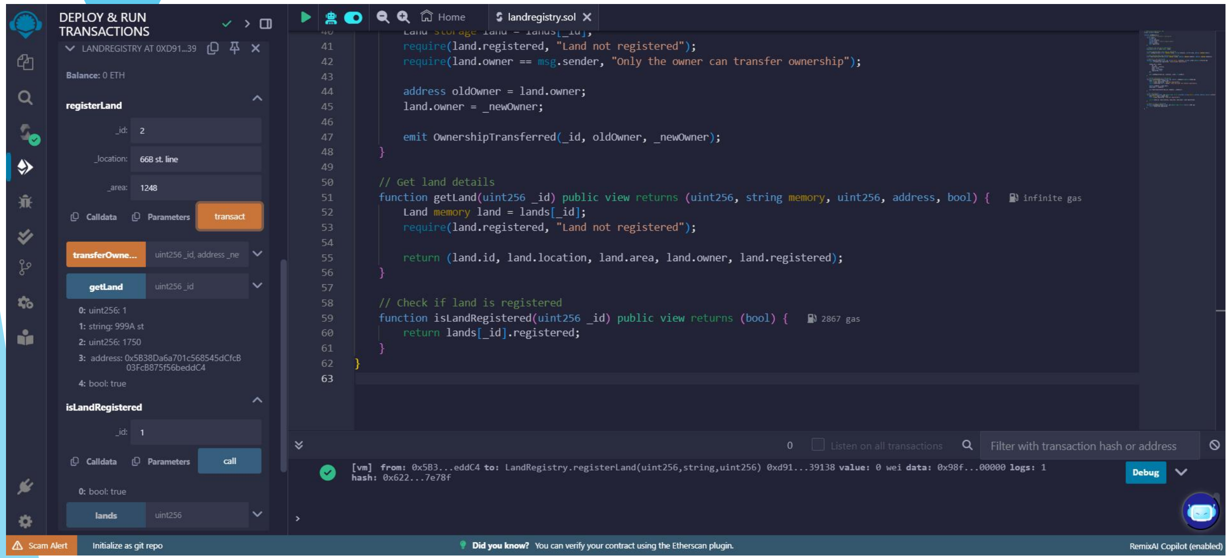
isLandRegiste... | uint256 _id

lands | uint256

```
    // Transfer ownership of a land parcel
    function transferOwnership(uint256 _id, address _newOwner) public {    33418 gas
        Land storage land = lands[_id];
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner = _newOwner;

        emit OwnershipTransferred(_id, oldOwner, _newOwner);
    }

    // Get land details
    function getLand(uint256 _id) public view returns (uint256, string memory, uint256, address, bool) {    infinite gas
        Land memory land = lands[_id];
        require(land.registered, "Land not registered");

        return (land.id, land.location, land.area, land.owner, land.registered);
    }

    // Check if land is registered
    function isLandRegistered(uint256 _id) public view returns (bool) {    2867 gas
        return lands[_id].registered;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: LandRegistry.registerLand(uint256,string,uint256) 0xd91...39138 value: 0 wei data: 0x98f...00000 logs: 1
hash: 0x1b2...10842
```

Did you know? You can verify your contract using the Etherscan plugin.


DEPLOY & RUN TRANSACTIONS

LANDREGISTRY AT 0XD91...39

Balance: 0 ETH

registerLand

_id:	2
_location:	66B st. line
_area:	1248

Calldata · Parameters · transact

transferOwne... | uint256 _id, address _ne

getLand | uint256 _id

```
0: uint256: 1
1: string: 999A st
2: uint256: 1750
3: address: 0x5B38Da6a701c568545dCfcB
            03FcB875f56beddC4
4: bool: true
```

isLandRegistered

_id:	1

Calldata · Parameters · call

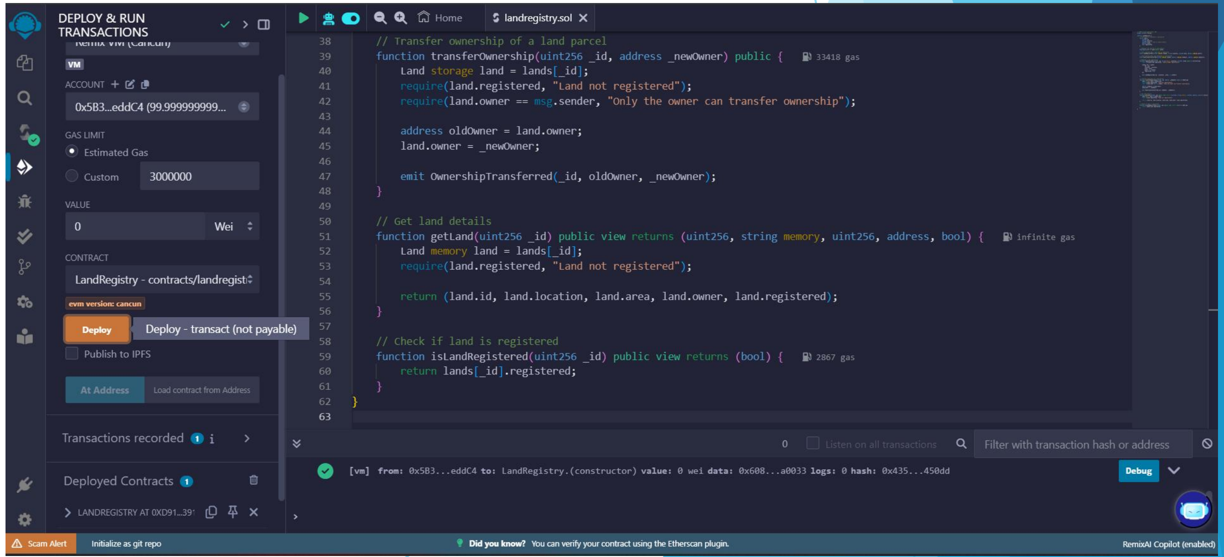
```
0: bool: true
```

lands | uint256

```
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner = _newOwner;

        emit OwnershipTransferred(_id, oldOwner, _newOwner);
    }

    // Get land details
    function getLand(uint256 _id) public view returns (uint256, string memory, uint256, address, bool) {    infinite gas
        Land memory land = lands[_id];
        require(land.registered, "Land not registered");

        return (land.id, land.location, land.area, land.owner, land.registered);
    }

    // Check if land is registered
    function isLandRegistered(uint256 _id) public view returns (bool) {    2867 gas
        return lands[_id].registered;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: LandRegistry.registerLand(uint256,string,uint256) 0xd91...39138 value: 0 wei data: 0x98f...00000 logs: 1
hash: 0x622...7e78f
```

Did you know? You can verify your contract using the Etherscan plugin.


DEPLOY & RUN TRANSACTIONS

VM

ACCOUNT: 0x5B3...eddC4 (99.999999999...

GAS LIMIT: [✓ Estimated Gas] Custom 3000000

VALUE: 0 Wei

CONTRACT: LandRegistry - contracts/landregist

evm version: cancun

Deploy · Deploy - transact (not payable)

Publish to IPFS

At Address · Load contract from Address

Transactions recorded 1 i

Deployed Contracts 1

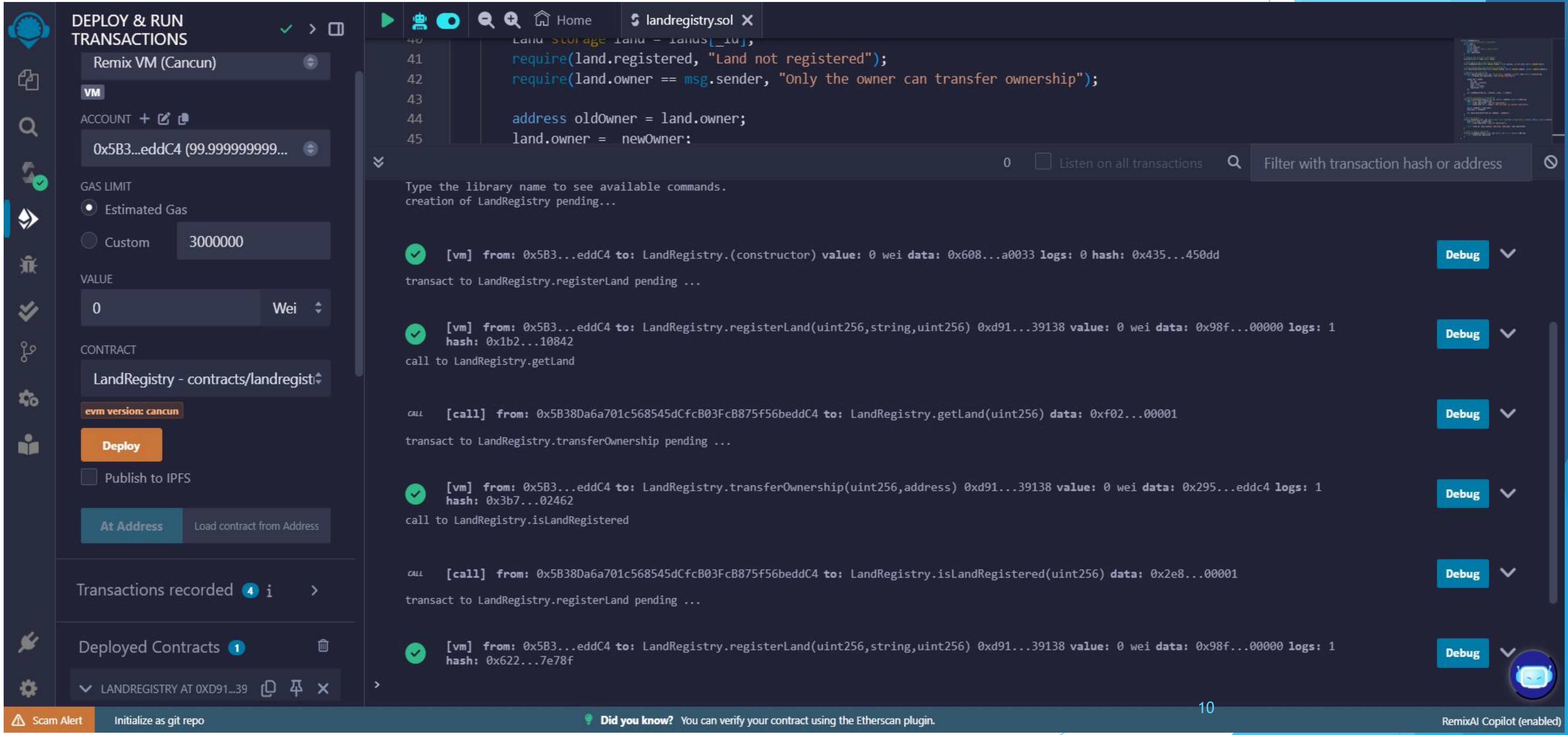
LANDREGISTRY AT 0XD91...39

```
    // Transfer ownership of a land parcel
    function transferOwnership(uint256 _id, address _newOwner) public {    33418 gas
        Land storage land = lands[_id];
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner = _newOwner;

        emit OwnershipTransferred(_id, oldOwner, _newOwner);
    }

    // Get land details
    function getLand(uint256 _id) public view returns (uint256, string memory, uint256, address, bool) {    infinite gas
        Land memory land = lands[_id];
        require(land.registered, "Land not registered");

        return (land.id, land.location, land.area, land.owner, land.registered);
    }

    // Check if land is registered
    function isLandRegistered(uint256 _id) public view returns (bool) {    2867 gas
        return lands[_id].registered;
    }
}
```

```
[vm] from: 0x5B3...eddC4 to: LandRegistry.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x435...450dd
```

Did you know? You can verify your contract using the Etherscan plugin.

Test Results:



DEPLOY & RUN TRANSACTIONS

Remix VM (Cancun)

VM

ACCOUNT

0x5B3...eddC4 (99.999999999...

GAS LIMIT

[✓ Estimated Gas]

Custom 3000000

VALUE

0 Wei

CONTRACT

LandRegistry - contracts/landregist

evm version: cancun

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 4

Deployed Contracts 1

LANDREGISTRY AT 0XD91...39

Home landregistry.sol

```
        Land storage land = lands[_id];
        require(land.registered, "Land not registered");
        require(land.owner == msg.sender, "Only the owner can transfer ownership");

        address oldOwner = land.owner;
        land.owner =  newOwner;
```

0 Listen on all transactions Filter with transaction hash or address

```
Type the library name to see available commands.
creation of LandRegistry pending...

[vm] from: 0x5B3...eddC4 to: LandRegistry.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0x435...450dd
transact to LandRegistry.registerLand pending ...

[vm] from: 0x5B3...eddC4 to: LandRegistry.registerLand(uint256,string,uint256) 0xd91...39138 value: 0 wei data: 0x98f...00000 logs: 1
hash: 0x1b2...10842
call to LandRegistry.getLand

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: LandRegistry.getLand(uint256) data: 0xf02...00001
transact to LandRegistry.transferOwnership pending ...

[vm] from: 0x5B3...eddC4 to: LandRegistry.transferOwnership(uint256,address) 0xd91...39138 value: 0 wei data: 0x295...eddc4 logs: 1
hash: 0x3b7...02462
call to LandRegistry.isLandRegistered

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: LandRegistry.isLandRegistered(uint256) data: 0x2e8...00001
transact to LandRegistry.registerLand pending ...

[vm] from: 0x5B3...eddC4 to: LandRegistry.registerLand(uint256,string,uint256) 0xd91...39138 value: 0 wei data: 0x98f...00000 logs: 1
hash: 0x622...7e78f
```

Did you know? You can verify your contract using the Etherscan plugin.

Thank you