Blockchain-based Land Registration System

First Presentation

Supervisor: Professor Yiu Siu Ming

Group members:
RAHMAN Mohammad Abdur
BHATIA Divtej Singh
AGARWAL Rahul
GOLI Smaran

TABLE OF CONTENTS

PROBLEM DEFINITION

Existing Land Registration System Challenges in Hong Kong:

01 Legal uncertainties and complexities in property transactions

02 Inconvenience and Higher costs

03 Difficulty in secure storage of paper deeds

Opportunities with Blockchain Technology

01 Reliable, decentralized, transparent and secure land registration system.

02 Adoption of smart contracts on public blockchain platforms

03 Potential for enhancing security and maintenance of land data

Literature Review

An overview of the key points discussed in ***"A Systematic Literature Review on Blockchain for Real Estate Transactions: Benefits, Challenges, Enablers, and Inhibitors"*** (https://doi.org/10.21203/rs.3.rs-2823844/v1)

Note: Articles were selected from four sources, EBSCO Discovery Service (816), Web of Science (190), IEEE (90) and Springer (261), till May 2022. Data was collected by an exhaustive reading of the 75 articles, published mostly in journals and conferences

Challenges in Real Estate Transactions

- Inefficiency, high costs, and complex processes.
- Corruption, fraud, and outdated technology.
- Lack of trust, transparency, and certainty.

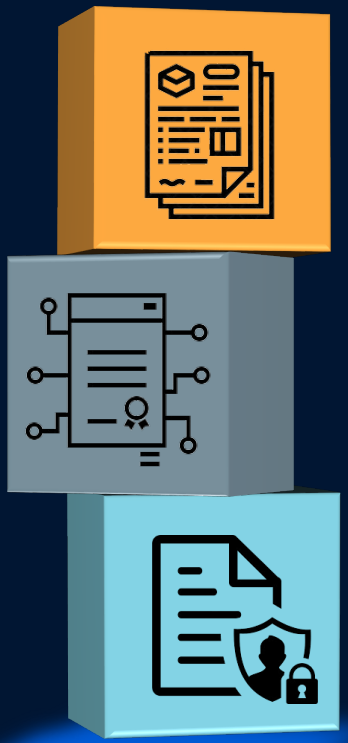
Benefits of Blockchain in Real Estate

- Increased transparency and security.
- Reduction of intermediaries, fraud, time, and costs.
- Accelerated execution and traceable transactions.

Challenges to overcome

- Lack of regulation and legal enforcement.
- Poor data quality and incomplete information.
- Compliance with data protection regulations.

PROJECT OBJECTIVES



ESTABLISHING A BLOCKCHAIN-BASED LAND REGISTRY

AUTOMATING PROPERTY TRANSACTIONS

ENSURING DATA SECURITY AND PRIVACY

METHODOLOGY

01

System Architectural Design

How the system works as a whole ?

02

Smart Contract Design

Initial Implementation of Smart Contracts

System Architectural Design

1. **Ethereum Blockchain:**

- **Digital Ledger:**
  - Ethereum blockchain is a decentralized and distributed ledger that securely records transactions across a network of computers. Each block in the chain contains a list of transactions, and these blocks are linked together using cryptographic hashes, forming a secure and transparent chain of data.

System Architectural Design

1. **Ethereum Blockchain:**

- **Digital Ledger:**
  - Ethereum blockchain is a decentralized and distributed ledger that securely records transactions across a network of computers. Each block in the chain contains a list of transactions, and these blocks are linked together using cryptographic hashes, forming a secure and transparent chain of data.

- **Land Transactions:**
  - In the context of land transactions, each entry on the Ethereum blockchain represents a unique transaction related to real estate, such as buying, selling, or transferring ownership.

System Architectural Design

2. **Smart Contracts (Solidity):**

- **Digital Agreements:**
  - Smart contracts are self-executing contracts with the terms of the agreement directly written into code. Solidity is the programming language we are using to create these smart contracts on the Ethereum platform.

System Architectural Design

2. **Smart Contracts (Solidity):**

- **Digital Agreements:**
  - Smart contracts are self-executing contracts with the terms of the agreement directly written into code. Solidity is the programming language we are using to create these smart contracts on the Ethereum platform.

- **Automation:**
  - Smart contracts automate processes by executing predefined actions when specific conditions are met. For example, in the context of land transactions, a smart contract could automatically transfer ownership of a piece of land to the buyer once the payment is confirmed, eliminating the need for intermediaries.

System Architectural Design

3. User Interface:

1. **Face of the System:** It provides a user-friendly way for individuals to engage with the underlying technology without needing to understand the complex coding or cryptographic concepts.

2. **Functionality:** In our land transactions scenario, the UI offers features such as a

   - Registration portal for new land transactions,
   - A dashboard for checking ownership details, and
   - Options for initiating and tracking the progress of transactions.

SMART CONTRACT Design

The 4 Main Use Cases:

REGISTER

For Land Registration

VERIFY / VIEW

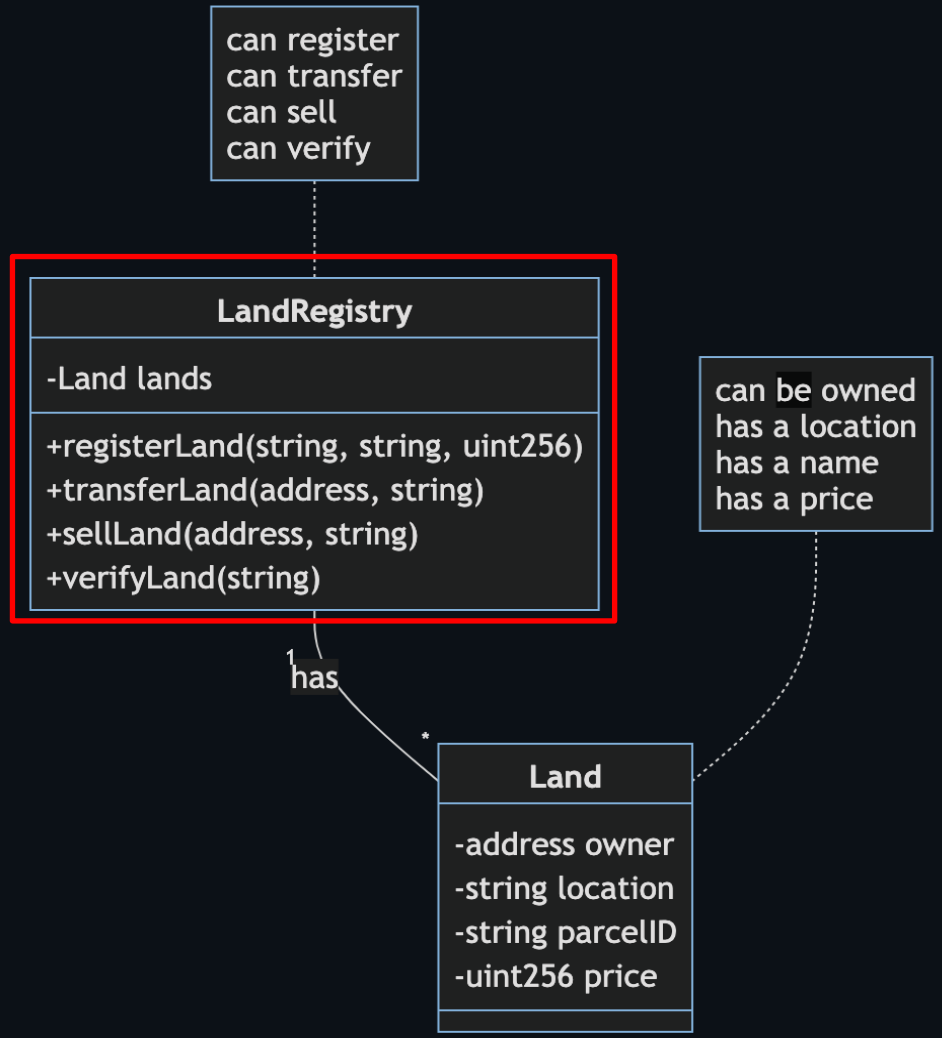
To verify the authenticity of records

TRANSFER

Land Transfer

SELL

For Property Sales



Smart Contracts

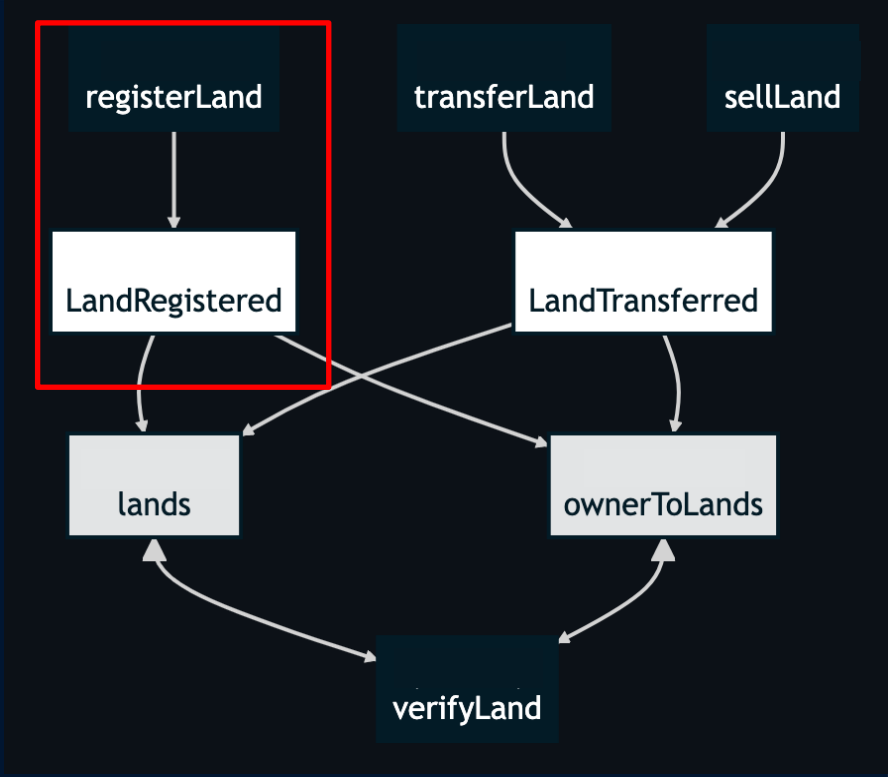
- Class Overview

Shows how data and functions interact with each other.

Interactions

How Different Functions and Events in our system interact.

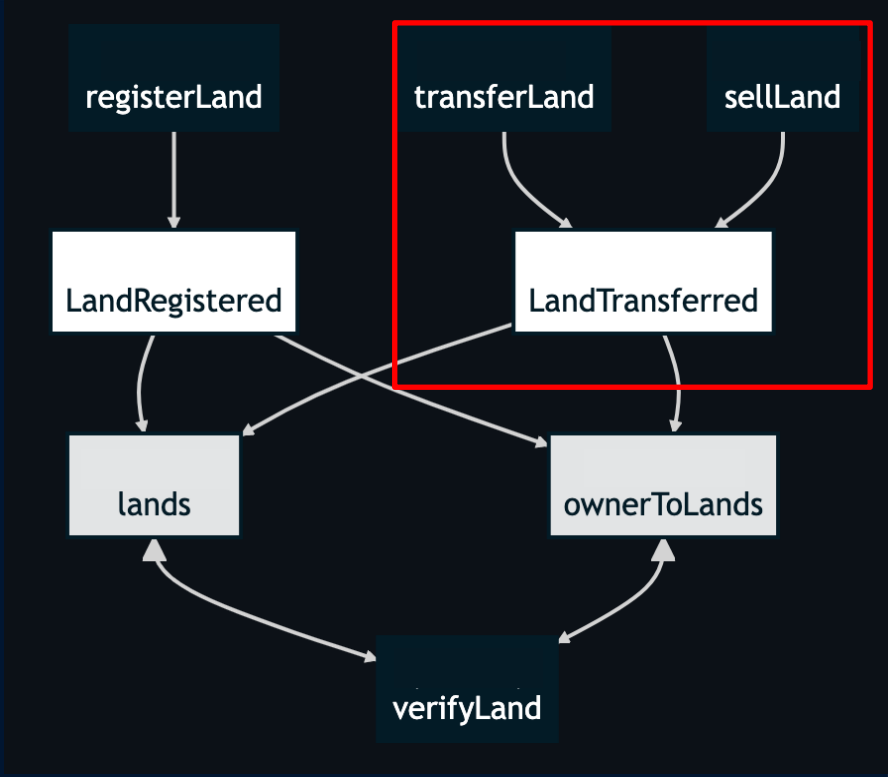
- **registerLand()** allows an address to register a new parcel of land with its location and a unique parcel ID. Emits a LandRegistered event when land is registered.



- **registerLand()** allows an address to register a new parcel of land with its location and a unique parcel ID. Emits a LandRegistered event when land is registered.

- **transferLand()** allows the current owner of a land parcel to transfer it to a new owner. Emits a LandTransferred event when land is transferred.

- **sellLand()** allows a landowner to sell their land to a buyer. This also involves the transfer of digital currency from the buyer to the seller, representing the payment for the land.



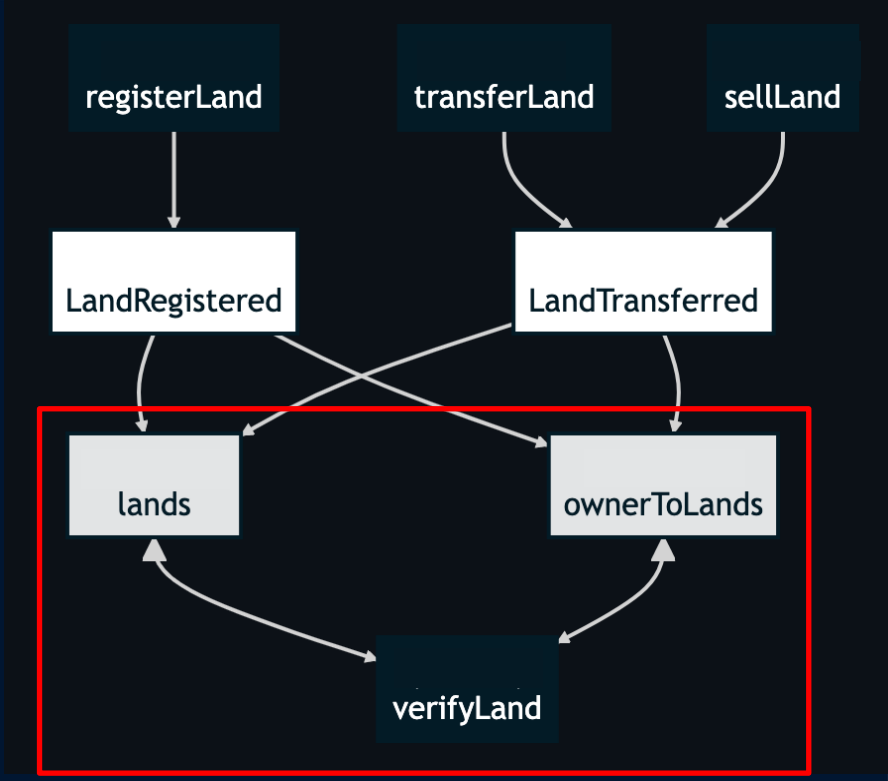
- **registerLand()** allows an address to register a new parcel of land with its location and a unique parcel ID. Emits a LandRegistered event when land is registered.

- **transferLand()** allows the current owner of a land parcel to transfer it to a new owner. Emits a LandTransferred event when land is transferred.

- **sellLand()** allows a landowner to sell their land to a buyer. This also involves the transfer of digital currency from the buyer to the seller, representing the payment for the land.

- **verifyLand()** allows anyone to verify the details of a land parcel.

- **listLandsByOwner()** allows for querying all land parcels owned by a specific owner, making it easier to get an overview of one's properties



BhatiaDivtej / RealEstateBlockchain

Type / to search

Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

RealEstateBlockchain

Private

Unwatch 1 | Fork 0 | Star 0

main | 1 Branch | 0 Tags | Go to file t | Add file | Code

id110067 Add .DS_Store to .gitignore — f0a5791 · last week — 7 Commits

Real Estate Blockchain Registry	Add .DS_Store to .gitignore	last week
Readme.md	Update Readme.md	3 months ago

README

Real Estate Blockchain - Land Registration / Tokenization

RealEstateBlockchain is a decentralized application built on a blockchain network designed to revolutionize the real estate industry. By leveraging the power of blockchain technology, RealEstateB transparency, efficiency, and security in real estate transactions.

Website to our Notion Page: https://www.notion.so/FYP-2023-2024-d6b32d8

Table of Contents

About

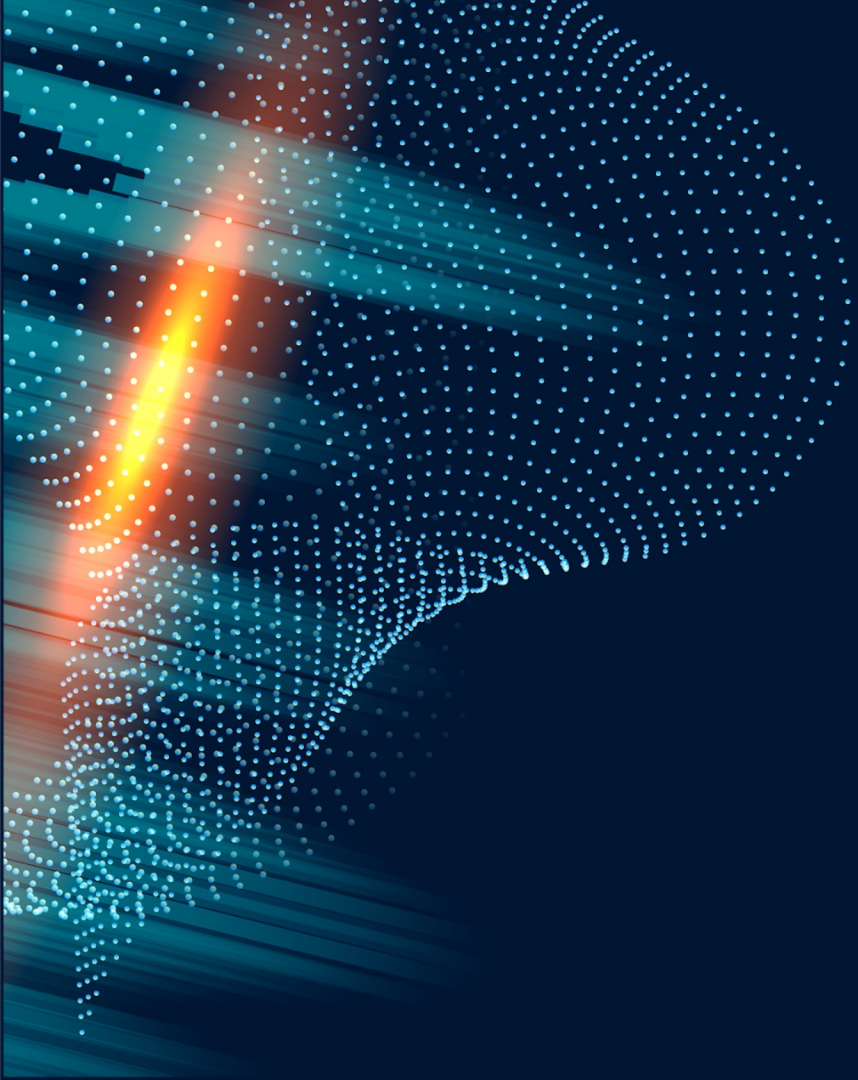
Project to tokenise land and property records and store in a blockchain.

Readme
Activity
0 stars
1 watching
0 forks

Releases

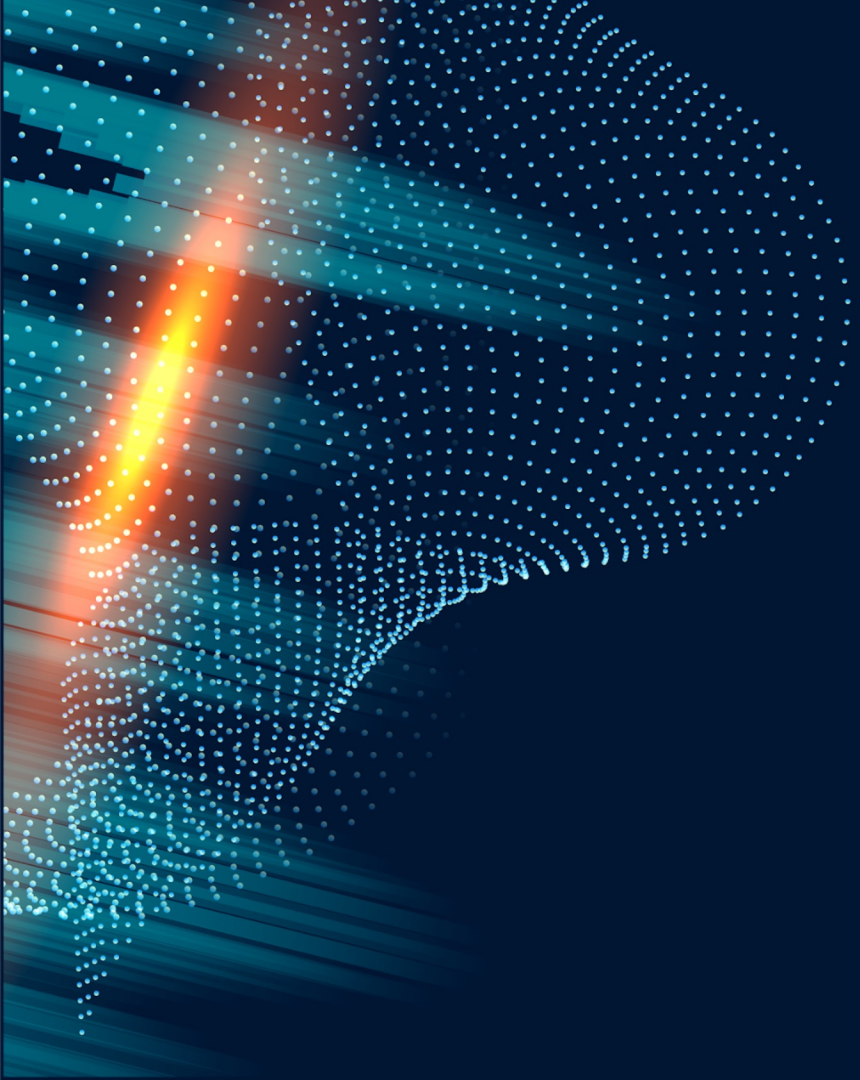
No releases published
Create a new release

Name	Last commit message	Last commit date
..		
contracts	Add files and scripts for Real Estate Blockchain Registry	last week
scripts	Add files and scripts for Real Estate Blockchain Registry	last week
.gitattributes	Add files and scripts for Real Estate Blockchain Registry	last week
.gitignore	Add .DS_Store to .gitignore	last week
brownie-config.yaml	Add files and scripts for Real Estate Blockchain Registry	last week



DEMO!

Video Link



Components

- Solidity Smart Contract:
- - LandRegistry.sol

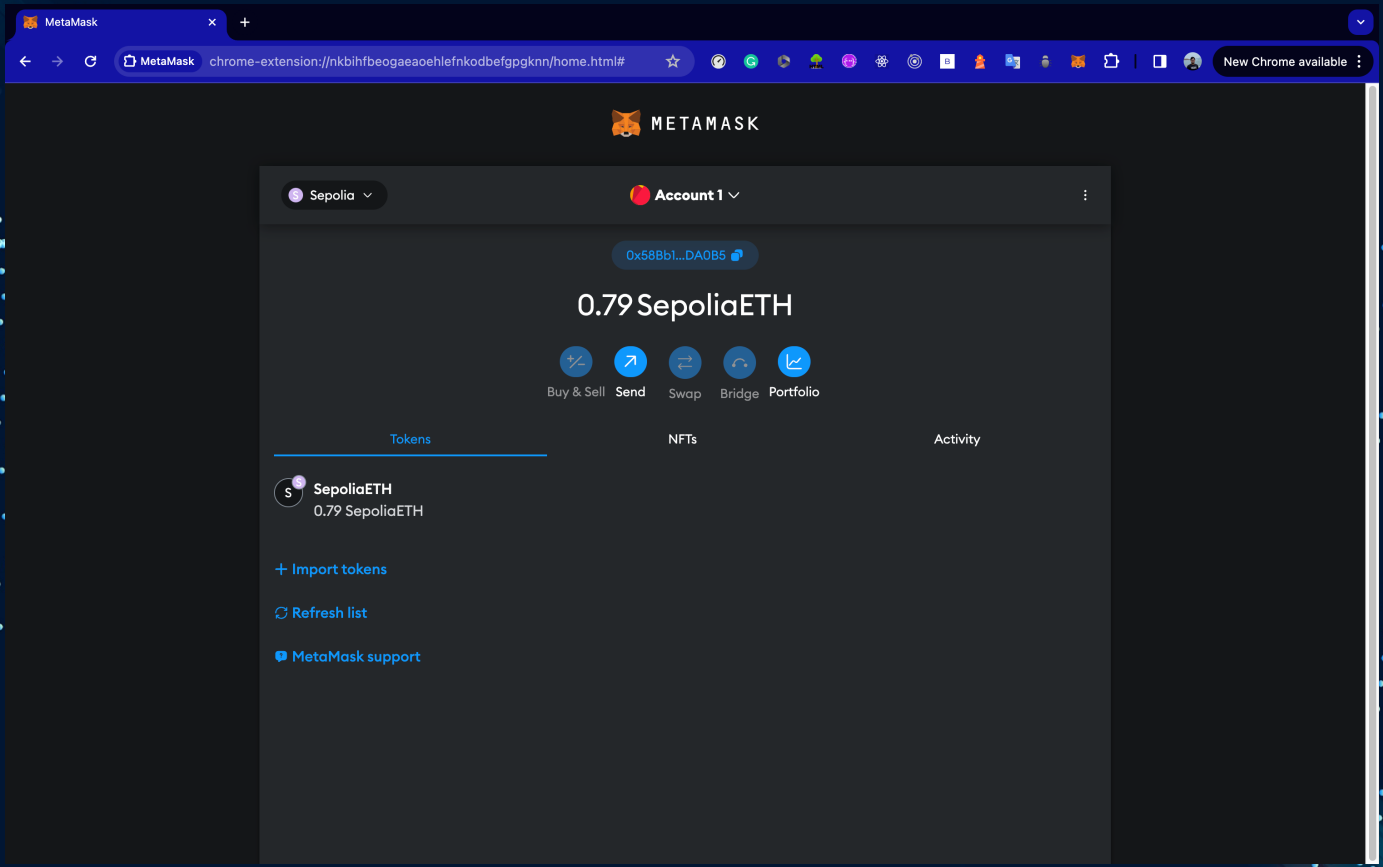
- Metamask Wallet

- Sepolia Testnet

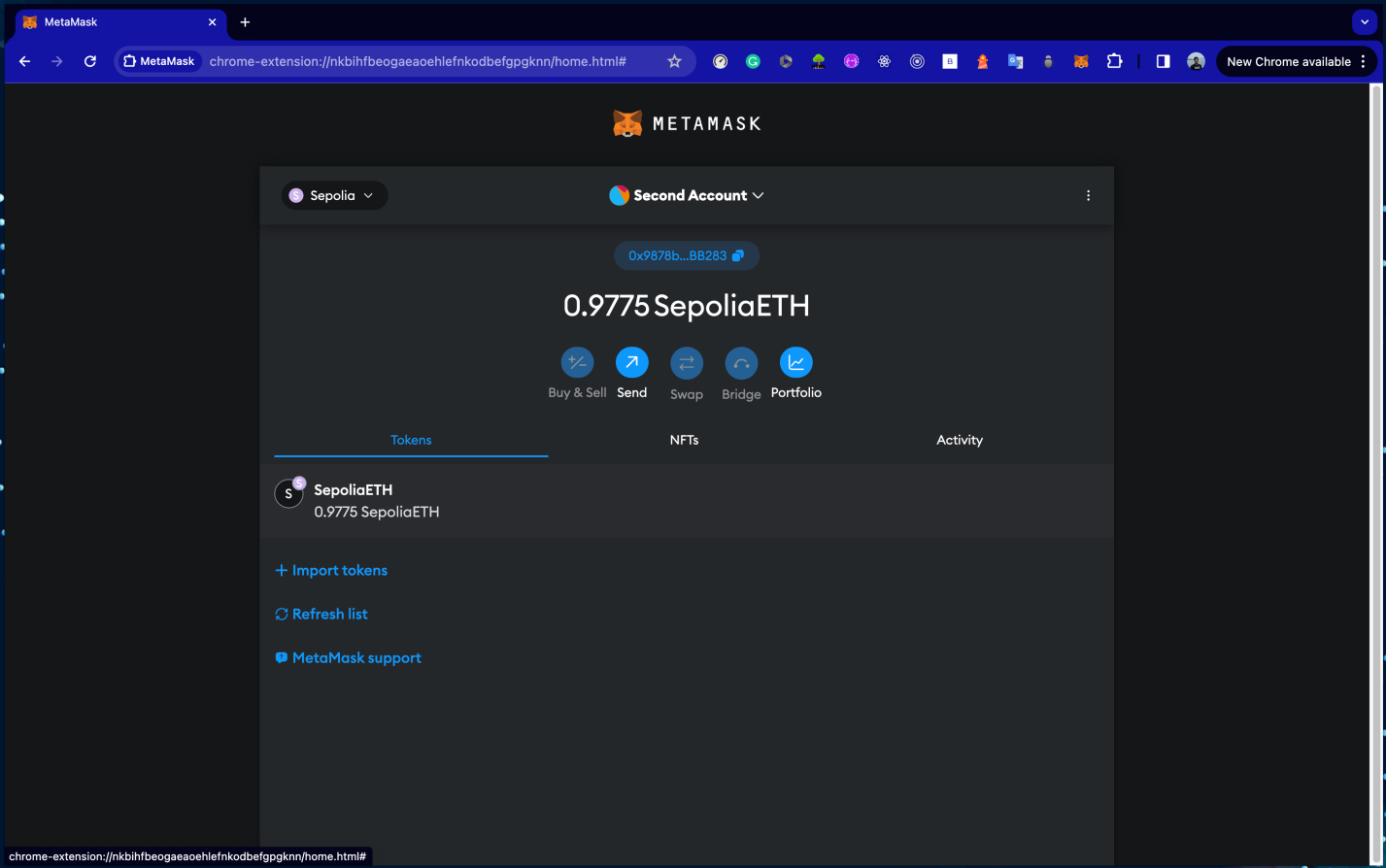
- Infura EVM

Python Scripts (using brownie framework):
- - deploy.py
- - register_land.py
- - verify_land.py
- - list_lands_by_owner.py
- - transfer_land.py
- - initiate_purchase.py
- - confirm_sale.py

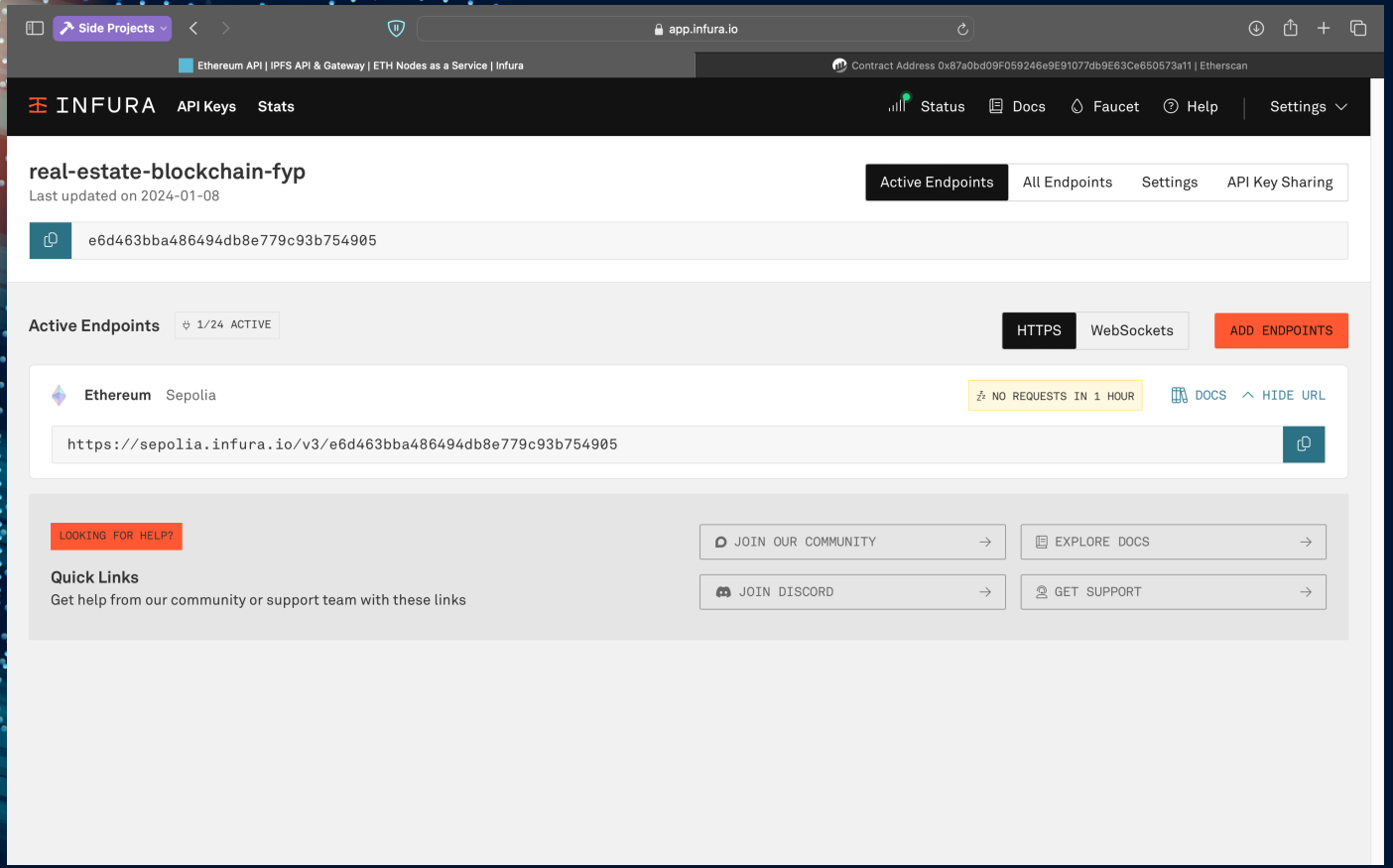
MetaMask Accounts



MetaMask Accounts



Infura EVM


Side Projects
app.infura.io
Ethereum API | IPFS API & Gateway | ETH Nodes as a Service | Infura
Contract Address 0x87a0bd09F059246e9E91077db9E63Ce650573a11 | Etherscan
INFURA
API Keys
Stats
Status
Docs
Faucet
Help
Settings
real-estate-blockchain-fyp
Last updated on 2024-01-08
Active Endpoints
All Endpoints
Settings
API Key Sharing
e6d463bba486494db8e779c93b754905
Active Endpoints
1/24 ACTIVE
HTTPS
WebSockets
ADD ENDPOINTS
Ethereum
Sepolia
NO REQUESTS IN 1 HOUR
DOCS
HIDE URL
https://sepolia.infura.io/v3/e6d463bba486494db8e779c93b754905
LOOKING FOR HELP?
Quick Links
Get help from our community or support team with these links
JOIN OUR COMMUNITY
EXPLORE DOCS
JOIN DISCORD
GET SUPPORT

Solidity Smart Contract

Structs

```
struct Land {
    address payable owner;
    string location;
    string parcelID;
    uint256 price;
    // Uncomment this if you want to include a flag indicating whether a
    // parcel of land is registered or not.
    // bool isRegistered;
}
```

Solidity Smart Contract

Mappings

```
mapping(string => Land) public lands;
```

```
mapping(address => string[]) public ownerToLands;
```

```
mapping(string => address payable) public pendingPurchases;
```

Solidity Smart Contract

Events

```
event LandRegistered(address indexed owner, string parcelID);
```

```
event PurchasePending(
    address indexed buyer,
    string parcelID,
    uint256 price
);
```

```
event LandSold(        You, 5 days ago •
    address indexed seller,
    address indexed buyer,
    string parcelID,
    uint256 price
);
```

```
event LandTransferred(
    address indexed oldOwner,
    address indexed newOwner,
    string parcelID
);
```

Solidity Smart Contract

Modifiers

```
modifier onlyLandOwner(string memory _parcelID) {
    require(
        lands[_parcelID].owner == msg.sender,
        "Only the current owner can perform this operation."
    );
    _;
}
```

Solidity Smart Contract

Methods

```
function registerLand(
    string memory _location,
    string memory _parcelID,
    uint256 _price
) public {
    // Check if the land parcel is already registered
    require(
        lands[_parcelID].owner == address(0),
        "This land parcel is already registered."
    );

    // Create a new Land struct and store it in the lands mapping
    lands[_parcelID] = Land(
        payable(msg.sender),
        _location,
        _parcelID,
        _price
    );

    // Add the parcelID to the list of lands owned by the sender
    ownerToLands[msg.sender].push(_parcelID);

    // Emit the event
    emit LandRegistered(msg.sender, _parcelID);
}
```

Solidity Smart Contract

Methods

```
function transferLand(
    address payable _newOwner,
    string memory _parcelID
) public onlyLandOwner(_parcelID) {
    // Transfer the land to the new owner
    address oldOwner = lands[_parcelID].owner;
    lands[_parcelID].owner = _newOwner;

    // Update the ownerToLands mapping for the old owner
    removeLandFromOwner(oldOwner, _parcelID);

    // Update the ownerToLands mapping for the new owner
    ownerToLands[_newOwner].push(_parcelID);

    // Emit the event
    emit LandTransferred(oldOwner, _newOwner, _parcelID);
}
```

Solidity Smart Contract

Methods

```
function initiatePurchase(string memory _parcelID) public payable {
    uint256 landPrice = lands[_parcelID].price;
    require(msg.value == landPrice, "Incorrect Ether value.");
    require(lands[_parcelID].owner != address(0), "Land not registered.");

    pendingPurchases[_parcelID] = payable(msg.sender);

    emit PurchasePending(msg.sender, _parcelID, msg.value);
}
```

Solidity Smart Contract

Methods

```
function confirmSale(
    string memory _parcelID
) public onlyLandOwner(_parcelID) {
    address payable buyer = pendingPurchases[_parcelID];
    require(buyer != address(0), "No pending purchase for this land.");

    // Transfer the land to the new owner
    address payable oldOwner = lands[_parcelID].owner;
    lands[_parcelID].owner = buyer;

    // Update the ownerToLands mappings for both old and new owners
    removeLandFromOwner(oldOwner, _parcelID);
    ownerToLands[buyer].push(_parcelID);

    // Transfer the funds to the previous owner
    uint256 price = lands[_parcelID].price;
    oldOwner.transfer(price);

    // Clear the pending purchase
    delete pendingPurchases[_parcelID];

    emit LandSold(oldOwner, buyer, _parcelID, price);
}
```

Solidity Smart Contract

Methods

```
function removeLandFromOwner(
    address _owner,
    string memory _parcelID
) internal {
    uint256 len = ownerToLands[_owner].length;
    for (uint256 i = 0; i < len; i++) {
        if (
            keccak256(bytes(ownerToLands[_owner][i])) ==
            keccak256(bytes(_parcelID))
        ) {
            ownerToLands[_owner][i] = ownerToLands[_owner][len - 1];
            ownerToLands[_owner].pop();
            break;
        }
    }
}
```

Solidity Smart Contract

Methods

```
function verifyLand(
    string memory _parcelID
) public view returns (address, string memory, string memory, uint256) {
    // Check if the land parcel is registered
    require(
        lands[_parcelID].owner != address(0),
        "This land parcel is not registered."
    );

    // Return the land details
    return (
        lands[_parcelID].owner,
        lands[_parcelID].location,
        lands[_parcelID].parcelID,
        lands[_parcelID].price
    );
}
```

Solidity Smart Contract

Methods

```
function listLandsByOwner(
    address _owner
) public view returns (string[] memory) {
    return ownerToLands[_owner];
}
```

.env File

Real Estate Blockchain Registry > .env

Click here to ask Blackbox to help you code faster

```
export PRIVATE_KEY_REGISTRAR=0x049db6e655d056dbde0e004acf241b2a3dff69a656d914a83730bca07ac6d12b
export PRIVATE_KEY_BUYER=0xefb55c200007a1da16e9d1ad87d85517caf1d956044bfd9d61f694f27f976adf
export WEB3_INFURA_PROJECT_ID=e6d463bba486494db8e779c93b754905
```

brownie-config.yaml File

Real Estate Blockchain Registry > brownie-config.yaml > { } networks > { } sepolia > host

You, 5 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```yaml
dotenv: .env
networks:
  default: sepolia
  sepolia:
    host: https://sepolia.infura.io/v3/${WEB3_INFURA_PROJECT_ID}      You, 5 days ago • Add files and script
    chain_id: 11155111
    explorer: https://sepolia.etherscan.io
wallets:
  from_key_registrar: ${PRIVATE_KEY_REGISTRAR}
  from_key_buyer: ${PRIVATE_KEY_BUYER}
```

Python Scripts

deploy.py

RealEstateBlockchain

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Estate... 1
- list_lands_by_owner.py R... 1
- deploy.py Real Estate Bloc... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py 1
    - initiate_purchase.py 1, M
    - list_lands_by_owner.py 1
    - register_land.py 1, M
    - transfer_land.py
    - verify_land.py 1
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

LandRegistry.sol | register_land.py 1, M | verify_land.py 1 | list_lands_by_owner.py 1 | deploy.py 1 | initiate_pur

Real Estate Blockchain Registry > scripts > deploy.py > get_account

You, 5 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import accounts, config, LandRegistry, network

def deploy_land_registry():
    account = get_account()
    print(f"Deploying from account: {account}")
    land_registry = LandRegistry.deploy({"from": account})
    print(f"LandRegistry deployed at: {land_registry.address}")

def get_account():
    active_network = network.show_active()
    print(f"Active network: {active_network}")
    if active_network == "development":
        return accounts[0]
    else:
        return accounts.add(config["wallets"]["from_key_registrar"])

def main():
    deploy_land_registry()
```

You, 5 days ago • Add files and scripts for Real Estate Blockchain …

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS | zsh - Real Estate Bloc

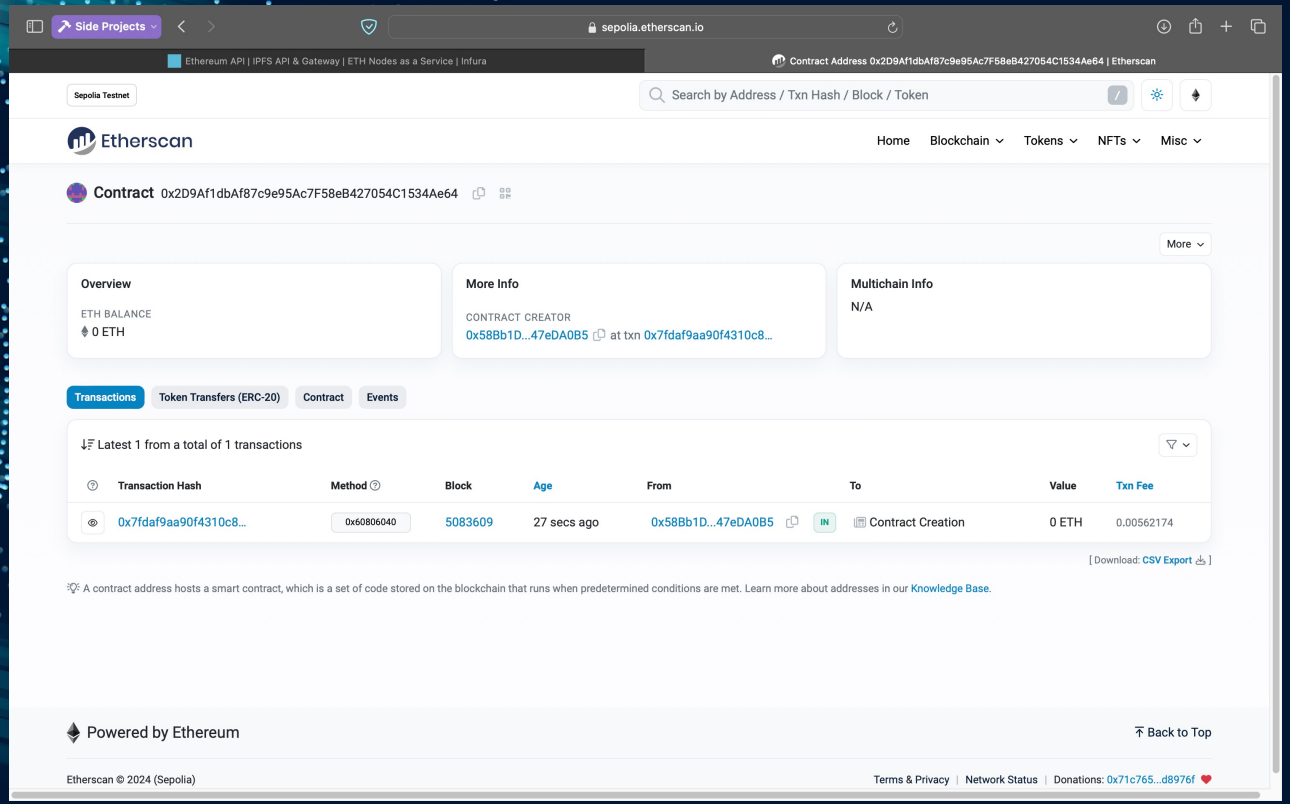
```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/deploy.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/deploy.py::main'...
Active network: sepolia
Deploying from account: 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5
Transaction sent: 0x7fdaf9aa90f4310c831b3a19ac5d012bb41ee154048314e5f1698f063331386b
  Gas price: 4.347547938 gwei   Gas limit: 1422392   Nonce: 15
  LandRegistry.constructor confirmed   Block: 5083609   Gas used: 1293084 (90.91%)
  LandRegistry deployed at: 0x2D9Af1dbAf87c9e95Ac7F58eB427054C1534Ae64

LandRegistry deployed at: 0x2D9Af1dbAf87c9e95Ac7F58eB427054C1534Ae64
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Confirm contract deployment on Etherscan



Python Scripts

register_land.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real Est... 1
- verify_land.py Real Estate... 1
- list_lands_by_owner.py R... 1
- deploy.py Real Estate Bloc... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py 1
    - initiate_purchase.py 1, M
    - list_lands_by_owner.py 1
    - register_land.py 1
    - transfer_land.py
    - verify_land.py 1
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

LandRegistry.sol | register_land.py 1 | verify_land.py 1 | list_lands_by_owner.py 1 | deploy.py 1 | initiate_purchase.p

Real Estate Blockchain Registry > scripts > register_land.py > main

You, 5 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config

def register_land(account, _location, _parcelID, _price):
    land_registry = LandRegistry[-1]
    tx = land_registry.registerLand(_location, _parcelID, _price, {'from': account})
    tx.wait(1)
    print(f"Land with parcel ID {_parcelID} registered by {account.address}")
    return tx

def main():
    # Use registrar account      You, 5 days ago • Add files and scripts for Real Estate Blockchain …
    registrar_account = accounts.add(config["wallets"]["from_key_registrar"])

    # Dummy data for demonstration
    _location = "6 Lung Wah Rd, Kennedy Town"
    _parcelID = "kennedytown1"
    _price = 20000000000000000 # 0.02 ETH in wei

    # Dummy data for demonstration
    # _location = "6B Sassoon Road, Pok Fu Lam"
    # _parcelID = "pokfulam1"
    # _price = 10000000000000000 # 0.01 ETH in wei

    register_land(registrar_account, _location, _parcelID, _price)
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS | zsh - Real Estate Blockchain Regi

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/register_land.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/register_land.py::main'...
Transaction sent: 0x84f74049bfd823e8daee808aca632bd0e46e1a7ca2553e39ba018c78db2bd012
  Gas price: 3.761713106 gwei   Gas limit: 176882   Nonce: 16
  LandRegistry.registerLand confirmed   Block: 5083626   Gas used: 160802 (90.91%)

  LandRegistry.registerLand confirmed   Block: 5083626   Gas used: 160802 (90.91%)

Land with parcel ID kennedytown1 registered by 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

register_land.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Estate... 1
- list_lands_by_owner.py R... 1
- deploy.py Real Estate Bloc... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py 1
    - initiate_purchase.py 1, M
    - list_lands_by_owner.py 1
    - register_land.py 1, M
    - transfer_land.py
    - verify_land.py 1
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

OUTLINE

TIMELINE

LandRegistry.sol | register_land.py 1, M | verify_land.py 1 | list_lands_by_owner.py 1 | deploy.py 1 | initiate_purc

Real Estate Blockchain Registry > scripts > register_land.py > main

Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config

def register_land(account, _location, _parcelID, _price):
    land_registry = LandRegistry[-1]
    tx = land_registry.registerLand(_location, _parcelID, _price, {'from': account})
    tx.wait(1)
    print(f"Land with parcel ID {_parcelID} registered by {account.address}")
    return tx

def main():
    # Use registrar account
    registrar_account = accounts.add(config["wallets"]["from_key_registrar"])

    # Dummy data for demonstration
    # _location = "6 Lung Wah Rd, Kennedy Town"
    # _parcelID = "kennedytown1"
    # _price = 20000000000000000 # 0.02 ETH in wei

    # Dummy data for demonstration
    _location = "6B Sassoon Road, Pok Fu Lam"
    _parcelID = "pokfulam1"
    _price = 10000000000000000 # 0.01 ETH in wei      You, 1 second ago • Uncommitted changes

    register_land(registrar_account, _location, _parcelID, _price)
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS

zsh - Real Estate Blockchain Registry

```
Running 'scripts/register_land.py::main'...
Transaction sent: 0x84f74049bfd823e8daee808aca632bd0e46e1a7ca2553e39ba018c78db2bd012
  Gas price: 3.761713106 gwei   Gas limit: 176882   Nonce: 16
  LandRegistry.registerLand confirmed   Block: 5083626   Gas used: 160802 (90.91%)

  LandRegistry.registerLand confirmed   Block: 5083626   Gas used: 160802 (90.91%)

Land with parcel ID kennedytown1 registered by 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/register_land.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/register_land.py::main'...
Transaction sent: 0x7c1bcbb1e5858163705616733a35c999075f9f28ef3d49d0677461df75b56e41
  Gas price: 4.754122334 gwei   Gas limit: 158032   Nonce: 17
  LandRegistry.registerLand confirmed   Block: 5083656   Gas used: 143666 (90.91%)

  LandRegistry.registerLand confirmed   Block: 5083656   Gas used: 143666 (90.91%)

Land with parcel ID pokfulam1 registered by 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

verify_land.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Est... 1, M
- list_lands_by_owner.py R... 1
- deploy.py Real Estate Bloc... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py 1
    - initiate_purchase.py 1, M
    - list_lands_by_owner.py 1
    - register_land.py 1, M
    - transfer_land.py
    - verify_land.py 1, M
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

LandRegistry.sol | register_land.py 1, M | verify_land.py 1, M | list_lands_by_owner.py 1 | deploy.py 1 | initiate_p

Real Estate Blockchain Registry > scripts > verify_land.py > main

... | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config

def verify_land(_parcelID):
    land_registry = LandRegistry[-1]
    return land_registry.verifyLand(_parcelID)

def main():
    # Dummy data for demonstration
    # _parcelID = "kennedytown1"      You, 1 second ago • Uncommitted changes
    _parcelID = "pokfulam1"

    land_info = verify_land(_parcelID)
    print(f"Land info: Owner: {land_info[0]}, Location: {land_info[1]}, Parcel ID: {land_info[2]}, Price: {land_info[3]}")
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS

zsh - Real Estate Blockchain Registry

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/verify_land.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/verify_land.py::main'...
Land info: Owner: 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5, Location: 6 Lung Wah Rd, Kennedy Town, Parcel ID: kennedytown1, Price: 20000000000000000
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/verify_land.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/verify_land.py::main'...
Land info: Owner: 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5, Location: 6B Sassoon Road, Pok Fu Lam, Parcel ID: pokfulam1, Price: 10000000000000000
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

list_lands_by_owner.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Est... 1, M
- list_lands_by_owner.p... 1, M
- deploy.py Real Estate Bloc... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py 1
    - initiate_purchase.py 1, M
    - list_lands_by_owner.... 1, M
    - register_land.py 1, M
    - transfer_land.py
    - verify_land.py 1, M
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

LandRegistry.sol | register_land.py 1, M | verify_land.py 1, M | list_lands_by_owner.py 1, M | deploy.py 1 | initiate

Real Estate Blockchain Registry > scripts > list_lands_by_owner.py > main

You, 1 second ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config

def list_lands_by_owner(_owner):
    land_registry = LandRegistry[-1]
    return land_registry.listLandsByOwner(_owner)

def main():
    # Use any account to list lands, here we use the registrar
    owner_account = accounts.add(config["wallets"]["from_key_registrar"])
    # owner_account = accounts.add(config["wallets"]["from_key_buyer"])      You, 1 second ago • Uncommitted changes

    land_list = list_lands_by_owner(owner_account.address)
    print(f"Lands owned by {owner_account.address}: {land_list}")
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS | zsh - Real Estate Blockchain Regis

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/list_lands_by_owner.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/list_lands_by_owner.py::main'...
Lands owned by 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5: ('kennedytown1', 'pokfulam1')
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

transfer_land.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Est... 1, M
- list_lands_by_owner.p... 1, M
- transfer_land.py Real Esta... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py
    - initiate_purchase.py 1, M
    - list_lands_by_owner.... 1, M
    - register_land.py 1, M
    - transfer_land.py 1
    - verify_land.py 1, M
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

LandRegistry.sol | register_land.py 1, M | verify_land.py 1, M | list_lands_by_owner.py 1, M | transfer_land.py 1 | initia

Real Estate Blockchain Registry > scripts > transfer_land.py > ...

You, 5 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config
    You, 5 days ago • Add files and scripts for Real Estate Blockchain …
def transfer_land(account, new_owner, parcel_id):
    land_registry = LandRegistry[-1]  # Fetch the latest deployed LandRegistry contract
    tx = land_registry.transferLand(new_owner, parcel_id, {'from': account})
    tx.wait(1)  # Wait for the transaction to be mined
    print(f"Land with parcel ID {parcel_id} transferred to {new_owner}.")
    return tx

def main():
    # Use transferer account, which is set in the brownie-config.yaml
    transferer_account = accounts.add(config["wallets"]["from_key_registrar"])

    # Dummy data for demonstration
    new_owner = accounts.add(config["wallets"]["from_key_buyer"]).address
    parcel_id = "kennedytown1"

    # Call the transfer function
    transfer_land(transferer_account, new_owner, parcel_id)
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS | zsh - Real Estate Blockchain Registry

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/transfer_land.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/transfer_land.py::main'...
Transaction sent: 0xba611a3f793c1adc300a564ed8ef91311baf28644e3214358c6a634309a8ec36
  Gas price: 4.796551886 gwei   Gas limit: 103595   Nonce: 18
  LandRegistry.transferLand confirmed   Block: 5083663   Gas used: 89378 (86.28%)

  LandRegistry.transferLand confirmed   Block: 5083663   Gas used: 89378 (86.28%)

Land with parcel ID kennedytown1 transferred to 0x9878bcc327791CB49B9722B5f18Cd8d1aD9BB283.
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

initiate_purchase.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Est... 1, M
- list_lands_by_owner.p... 1, M
- initiate_purchase.py R... 1, M
- transfer_land.py Real ... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py
    - initiate_purchase.py 1, M
    - list_lands_by_owner.... 1, M
    - register_land.py 1, M
    - transfer_land.py 1, M
    - verify_land.py 1, M
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

egister_land.py 1, M | verify_land.py 1, M | list_lands_by_owner.py 1, M | initiate_purchase.py 1, M | transfer_land.py 1, M

Real Estate Blockchain Registry > scripts > initiate_purchase.py > ...

You, 2 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config, network
```

You, 6 days ago • Add files and scripts for Real Estate Blockchain …

```python
def initiate_purchase(buyer, parcel_id, price):
    land_registry = LandRegistry[-1]  # Fetch the latest deployed LandRegistry contract

    # The buyer initiates the land purchase
    tx = land_registry.initiatePurchase(parcel_id, {'from': buyer, 'value': price})
    tx.wait(1)

    print(f"Purchase for land with parcel ID {parcel_id} initiated by {buyer}.")
    return tx

def main():
    # Use buyer account
    buyer = accounts.add(config['wallets']['from_key_buyer'])

    # Dummy data for demonstration
    parcel_id = "pokfulam1"
    land_registry = LandRegistry[-1]

    # Access the land's details from the contract and then get its price
    land_details = land_registry.lands(parcel_id)
    land_price = land_details[3]

    # Call the initiate purchase function
    initiate_purchase(buyer, parcel_id, land_price)
```

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS

zsh - Real Estate Blockchain Registry

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/initiate_purchase.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/initiate_purchase.py::main'...
Transaction sent: 0xf1d7f0a1455f6e3b732cb67eccc8d4bad1dad303fa3ad01ce17a7f29621927b4
  Gas price: 18.156371518 gwei   Gas limit: 57252   Nonce: 8
  LandRegistry.initiatePurchase confirmed   Block: 5084201   Gas used: 52048 (90.91%)

  LandRegistry.initiatePurchase confirmed   Block: 5084201   Gas used: 52048 (90.91%)

Purchase for land with parcel ID pokfulam1 initiated by 0x9878bcc327791CB49B9722B5f18Cd8d1aD9BB283.
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

Python Scripts

confirm_sale.py

EXPLORER

OPEN EDITORS
- LandRegistry.sol Real Estate B...
- register_land.py Real ... 1, M
- verify_land.py Real Est... 1, M
- list_lands_by_owner.p... 1, M
- transfer_land.py Real Esta... 1
- initiate_purchase.py R... 1, M
- confirm_sale.py Real E... 1, M

REALESTATEBLOCKCHAIN
- Real Estate Blockchain R...
  - build
  - contracts
    - LandRegistry.sol
  - interfaces
  - reports
  - scripts
    - __pycache__
    - confirm_sale.py 1, M
    - deploy.py
    - initiate_purchase.py 1, M
    - list_lands_by_owner.... 1, M
    - register_land.py 1, M
    - transfer_land.py 1
    - verify_land.py 1, M
  - tests
  - .env
  - .gitattributes
  - .gitignore
  - brownie-config.yaml
  - Readme.md

verify_land.py 1, M | list_lands_by_owner.py 1, M | transfer_land.py 1 | initiate_purchase.py 1, M | confirm_sale.py 1, M

Real Estate Blockchain Registry > scripts > confirm_sale.py > main

You, 2 days ago | 1 author (You) | Click here to ask Blackbox to help you code faster

```python
from brownie import LandRegistry, accounts, config, network

def confirm_sale(seller, parcel_id):
    land_registry = LandRegistry[-1]  # Fetch the latest deployed LandRegistry contract

    # The seller confirms the sale
    tx = land_registry.confirmSale(parcel_id, {'from': seller})
    tx.wait(1)

    print(f"Sale for land with parcel ID {parcel_id} confirmed by {seller}.")
    return tx

def main():
    # Use seller account
    seller = accounts.add(config['wallets']['from_key_registrar'])

    # Dummy data for demonstration
    parcel_id = "pokfulam1"

    # Call the confirm sale function
    confirm_sale(seller, parcel_id)
```

You, 5 days ago • Add files and scripts for Real Estate Blockchain …

PROBLEMS 6 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS | SEARCH ERROR | GITLENS | COMMENTS | zsh - Real Estate Blockchain Registry

```
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry % brownie run scripts/confirm_sale.py --network sepolia
Brownie v1.19.3 - Python development framework for Ethereum

RealEstateBlockchainRegistryProject is the active project.

Running 'scripts/confirm_sale.py::main'...
Transaction sent: 0x393c817aec0fa6b6dd67d2b885ac184e39ae45cedb05d213caecd1bae6a34eaf
  Gas price: 4.151357007 gwei   Gas limit: 94955   Nonce: 19
  LandRegistry.confirmSale confirmed   Block: 5083672   Gas used: 71923 (75.74%)

  LandRegistry.confirmSale confirmed   Block: 5083672   Gas used: 71923 (75.74%)

Sale for land with parcel ID pokfulam1 confirmed by 0x58Bb1D08d9a13EbA3592C60BB45086DD47eDA0B5.
abdur8606@Mohammad-Abdurs-MacBook-Pro Real Estate Blockchain Registry %
```

CHALLENGES

INTEGRATING LAND REGISTRATION DATA

LEGAL AND TECHNOLOGICAL COMPLEXITIES

SMART CONTRACT COMPLEXITIES

SECURITY

PROJECT SCHEDULE

PHASE 1

SEPT – OCT

Phase 1 Deliverables

Project Plan (October 1, 2023)

Phase 2 Deliverables

First Presentation (January 8-12th 2024)

Interim Project Report (January 21st)

NOV - JAN

PHASE 2

PHASE 3

FEB - MAR

Key Milestones

Back testing and optimisations

Performance testing

Final documentation

Phase 4 Deliverables

Final Project Report

Final Project Presentation (Due date: April 15-19th 2023)

APR

PHASE 4

CONCLUSION

Next Steps

Further Implementation and Testing

Blockchain Revolution

Property Transaction Transformation

Milestone Achievements

Data Integration and Implementation of Smart Contracts

Future Outlook

Total Asset Digitalization

THANKS!

Do you have any questions?