Real Estate Transactions

BY TARQUIN 2022524

Throughout my CA I am going to explain why I think we can streamline the Real Estate Business. This will include the the storing of documentation and transaction records on a public ledger. This would reduce costs and increase efficiency throughout the Real Estate world.

1 - Immutable Property Records

Why Implement a **blockchain-based solution** to the real estate world? We can create a secure and unalterable record of all property ownership. This would mean no transaction can be changed or deleted. The benefit of adding Blockchain technology starts with preventing fraud and disputes of property ownership due to the implementation of an entire transparent record.

2 - Decentralized Marketplace

A real estate **marketplace** with the implementation of **blockchain-based** technology would allow buyers and sellers to interact directly with each other. This would remove the need of a real estate agent or lawyers if wanted by both parties. The benefit of this would be a decrease in fees which inflate the current market value of property due to the involvement of mentioned estate agents and lawyers.

3 - Tokenization

Tokenization will involve converting a property asset into a digital token for a blockchain. This would represent fractional ownership. What does fractional owenership mean? For example if a large property is being traded it can be invested by multiple token owners. This could allow for people to invest into property without needing to buy an entire house. This would be similar to a crypto currency like bitcoin but it's physical real estate instead. This could cause challenges with private assets like commercial housing however if this can be regulated it can be good for the economy.

4 - Smart Contracts

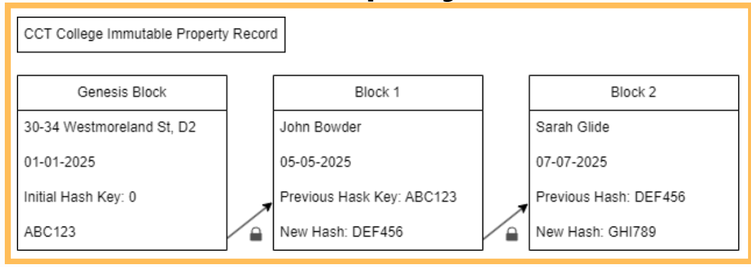
Why and how can we implement **smart contracts** to real estate transactions? Like previously mentioning the removal of lawyes (and brokers) when it comes to buying or selling of property, it can reduce the cost and time it takes to complete a sale. We can start by adding smart contracts thay will execute a certain code once the conditions of the contract are met. To put this into context I will provide an example of code:

4.2 Code Explenation

Code #1: I started off by adding the solidity license and version. For Visual Studio purposes I had to use 0.8.0. I made a **smart contract** under the name **RealEstateContract,** I added the struct Property, this would allow me to bundle a lot of useful related data together. propertyCount is there to assign unique IDs for the listed properties. Events go as follows: **PropertyListed:** Logs when a property is listed. **TokenPurchased:** Logs tokens purchased for fractional ownership. **OwnershipTransferred:** Logs when a property ownership changes.

Code #2: This part of the code will allow for the transfer of ownership. This is also codded with the **onlyOwner** modifier.

1.1 - Immutable Property Record



1.2 - Property Record Explained

To explain how the implementation of a **blockchain-based solution** will work with an **immutable property record** I have provided an example diagram to the left. This showcases the Genesis Block (The original record of the property log) and 2 updated blocks. The Genesis Block will contain the address of the property and the initial starting date of the log. The hash key process is the same as any toher blockchain. The Property record will be updated once the property is sold to its new owner, this will also contain the updated date for the propert record. This process will make the Property Record immutable meaning it cannot be changed.

4.1 - Smart Contracts CODE

Code #1

RealEstateSmartContract.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

// Smart Contract for Real Estate Transactions
contract RealEstateSmartContract {
    struct Property {
        uint id;
        string name;
        uint price;
        address owner;
        uint totalTokens;
        uint tokensAvailable;
        mapping(address => uint) tokenHolders;
    }

    // Counting property ID
    uint public propertyCount = 0;
    mapping(uint => Property) public properties;

    event PropertyListed(uint propertyId, string name, uint price, address owner);
    event TokensPurchased(uint propertyId, address buyer, uint tokens);
    event OwnershipTransferred(uint propertyId, address previousOwner, address newOwner);

    modifier onlyOwner(uint propertyId) {
        require(msg.sender == properties[propertyId].owner, "Not the property owner");
        _;
    }

```

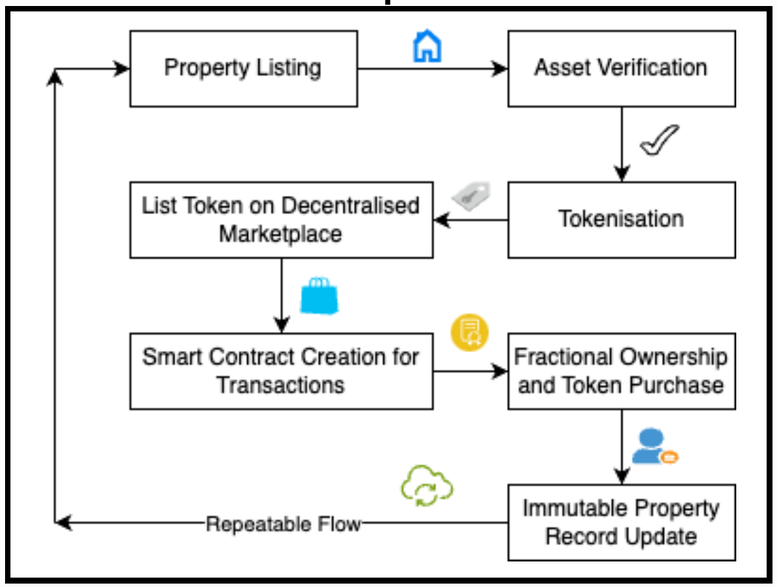
Code #2

```
    // Function will allow for transfering ownership of property
    function transferOwnership(uint256 propertyId, address newOwner) public onlyOwner(propertyId) {
        require(newOwner != address(0), "New owner cannot be the 0 address");

        address previousOwner = properties[propertyId].owner;
        properties[propertyId].owner = newOwner;

```

For the purpose of committing my code to a repository I used Visual Studio. However the deployment of the smart contract was done through remix.ethereum.org

Github link to code: https://github.com/2022524/RealEstateSmartContract/commits/main/

5 -Process and Example



6 - Conclusion

Throughout my poster I have demonstrated how we an add a blockchain-solution into the world of real estate transactions. I have displayed the immutable records and process with diagrams. Went over theory on how we can add Tokenization. All of this will contribute to a decentralized market. To further explain and visualize the vision of my CA we have the solidity code too. This will be demostrated on my video.

7 - References

1 - ChatGPT (2024). *ChatGPT - Basic Solidity Contract.* [online] ChatGPT. Available at: https://chatgpt.com/share/673f2c7f-d688-800b-9708-9a14f840986f [Accessed 18 Nov. 2024].

2 - Consensys (n.d.). *Blockchain in Real Estate | Real World Blockchain Use Cases.* [online] Consensys. Available at: https://consensys.io/blockchain-use-cases/real-estate [Accessed 15 Nov. 2024].

3 - Solidity (2016). *Solidity — Solidity 0.8.28 documentation.* [online] Soliditylang.org. Available at: https://docs.soliditylang.org/en/v0.8.28/ [Accessed 7 Nov. 2024].