SALEM COLLEGE OF ENGINEERING AND TECHNOLOGY

BIOMEDICAL ENGINEERING

NAAN MUDHALVAN

PROJECT DOMAIN: BLOCKCHAIN TECHNOLOGY
PROJECT TITLE: BLOCKCHAIN BASED SMART REAL ESTATE MANAGEMENT SYSTEM
TEAM MEMBERS

REGISTER NUMBER	NAME
622420121049	TAMILAZHAGAN K
622420121048	SUTHEEN R
622420121014	JEEVA S
622420121306	RUGEND S

TABLE OF CONTENT

1. INTRODUCTION

Smart real estate management using blockchain technology is poised to revolutionize the way we buy, sell, and manage properties. Blockchain, a decentralized and immutable ledger, offers transparency, security, and efficiency in the real estate sector. This innovative approach ensures trustworthy transactions, reduces fraud, and streamlines property management, creating a more seamless and reliable real estate experience. In this discussion, we will explore the transformative potential of blockchain technology in real estate management, from property transactions to record-keeping and beyond.

In the ever-evolving world of real estate, the integration of blockchain technology has emerged as a transformative force, ushering in a new era of smart real estate management. Blockchain's decentralized ledger and cryptographic security provide a foundation for transparency and trust in property transactions and management. This introduction delves into the exciting realm of smart real estate management using blockchain, exploring how this innovative technology is poised to revolutionize the way we buy, sell, and manage real estate assets, making the entire process more secure, efficient, and accessible.

Project Overview

The project for smart real estate management using blockchain technology is a cutting-edge initiative designed to revolutionize the real estate industry. Leveraging the power of blockchain, the project aims to enhance transparency, security, and efficiency throughout the real estate lifecycle. Key objectives include securing property transactions through blockchain, verifying ownership records, and implementing smart contracts for automated processes such as lease agreements and rent collection. Additionally, the project explores the tokenization of real estate assets, offering fractional ownership opportunities and increased liquidity. It will establish a secure, permissioned blockchain network

for data sharing among real estate professionals, simplifying collaboration and reducing administrative burdens. By creating user-friendly interfaces and ensuring regulatory compliance, the project seeks to streamline real estate management, making it more accessible, trustworthy, and secure for all stakeholders.

Purpose

The purpose of implementing smart real estate management using blockchain technology is to usher in a new era of efficiency, transparency, and security within the real estate industry. This innovative approach aims to address several critical challenges that have historically plagued the sector. By leveraging blockchain's immutable ledger, the project seeks to instill trust and reduce fraud in property transactions, providing a secure foundation for buyers, sellers, and investors. Furthermore, by verifying property ownership records and implementing smart contracts for lease agreements and rent collection, the project streamlines and automates property management, reducing the need for intermediaries and minimizing the potential for disputes. The exploration of tokenization enhances access to real estate investments, enabling fractional ownership and increasing liquidity. By facilitating data sharing and collaboration among real estate professionals through a secure blockchain network, the project aims to significantly reduce administrative overhead and paperwork. In essence, the purpose of this endeavor is to modernize and optimize the real estate industry, making it more accessible, trustworthy, and efficient for all stakeholders involved.

2.LITERATURE SURVEY

"How Blockchain Is Changing Finance",Auther: A. Tapscott and T. Don,Year:2017,Publisher:IEEE

Blockchain was originally developed as the technology behind cryptocurrencies like Bitcoin. A vast, globally distributed ledger running on millions of devices, it is capable of recording anything of value. Money, equities, bonds, titles, deeds, contracts, and virtually all other kinds of assets can be moved and stored securely, privately, and from peer to peer, because trust is established not by powerful intermediaries like banks and governments, but by network consensus, cryptography, collaboration, and clever code. For the first time in human history, two or more parties, be they businesses or individuals who may not even know each other, can forge agreements, make transactions, and build value without relying on intermediaries (such as banks, rating agencies, and government bodies such as the U.S. Department of State) to verify their identities, establish trust, or perform the critical business logic contracting, clearing, settling, and record-keeping tasks that are foundational to all forms of commerce.

"A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities",Auther: Fahim Ullah & Fadi Al-Turjman,Year:2021,Publisher:IEEE

Blockchains-based smart contracts are disrupting the smart real estate sector of the smart cities. The current study explores the literature focused on blockchain smart contracts in smart real estate and proposes a conceptual framework for its adoption in smart cities. Based on a systematic review method, the literature published between 2000 and 2020 is explored and analyzed. From the literature, ten key aspects of the blockchain smart contracts are highlighted that are grouped into six layers for adopting the smart contracts in smart real estate. The decentralized application and its interactions with Ethereum Virtual

Machine (EVM) are presented to show the development of a smart contract that can be used for blockchain smart contracts in real estate. Further, a detailed design and interaction mechanism are highlighted for the real estate owners and users as parties to a smart contract. A list of functions for initiating, creating, modifying, or terminating a smart contract is presented along with a stepwise procedure for establishing and terminating smart contracts. The current study can help the users enjoy a more immersive, user-friendly, and visualized contracting process, whereas the owners, property technologies (Proptech) companies, and real estate agents can enjoy more business and sales. This can help disrupt traditional real estate and transform it into smart real estate in line with industry 4.0 requirements.

Existing Problem

Implementing blockchain technology in smart real estate management presents a promising solution to several long-standing challenges in the industry. These include issues of transparency and trust, as traditional real estate transactions often lack transparency, making it difficult for parties to trust each other. Blockchain's transparent and immutable ledger can enhance trust by providing a secure and unchangeable record of all transactions and property history. Property fraud is another significant problem, which can be mitigated through blockchain's security features, ensuring that property records and ownership details remain tamper-proof. Furthermore, the inefficiency and high transaction costs associated with traditional real estate transactions due to numerous intermediaries can be reduced by blockchain, enabling smart contracts to streamline the process and expedite transactions. Property ownership verification, data security, cross-border transactions, and regulatory compliance are other challenges that blockchain can address effectively. It achieves this through its decentralized, secure, and automated nature, which ultimately revolutionizes the real estate industry by providing a more secure, efficient, and transparent framework for managing properties.

References

[1] A. Tapscott and T. Don, “How blockchain is changing finance,” Harvard Business Review, vol. 1, no. 9, pp. 2–5, 2017.
[2] Q. I. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du et al., “MeDShare: Trust-less medical data sharing among
cloud service providers via blockchain,” IEEE Access, vol. 5, pp. 14757–14767, 2017.
[3] O. Novo, “Blockchain meets IoT: An Architecture for scalable access management in IoT,” IEEE Internet of
Things Journal, vol. 5, no. 2, pp. 1184–1195, 2018.
[4] J. Sun, Y. Yan and K. Z. Zhang, “Blockchain-based sharing services: What blockchain technology can contribute
to smart cities,” Financial Innovation, vol. 2, no. 1, pp. 1–9, 2016.

Problem Statement Definition

The problem statement for smart real estate management using blockchain technology revolves around the inefficiencies, lack of transparency, and susceptibility to fraud that have long plagued the traditional real estate industry. Conventional real estate transactions are often marred by a lack of transparency, making it difficult for parties involved to trust each other fully. Moreover, the multitude of intermediaries in the process, such as agents, banks, and title companies, results in delays and significant transaction costs. Property fraud, a persistent issue, threatens the integrity of property records and ownership information, and the regulatory landscape can be complex and challenging to navigate.In this context, the adoption of blockchain technology offers a potential solution to these problems. By leveraging blockchain's decentralized and immutable ledger, real estate transactions can be conducted with enhanced transparency and trust, as every step of the process is recorded in a tamper-proof manner. Smart contracts, powered by blockchain, can automate and streamline

various aspects of real estate transactions, reducing the need for intermediaries and accelerating the process.

3.IDEATION & PROPOSED SOLUTION

Ideation:

Smart real estate management using blockchain is an innovative approach to revolutionize the traditional real estate industry. By leveraging blockchain technology, we aim to enhance transparency, security, and efficiency in property transactions and management. Our proposed solution will establish a decentralized network where property records, transactions, and ownership details are securely stored, making them tamper-proof and easily accessible to authorized parties. This system will streamline the buying, selling, and renting processes, reducing the need for intermediaries and associated costs. Additionally, it will introduce smart contracts to automate rental payments, property maintenance, and other aspects of real estate management. By combining blockchain's trust and automation with real estate, our solution has the potential to bring about a more accessible, secure, and cost-effective approach to property management and transactions, benefiting both property owners and tenants alike.Blockchain-based smart real estate management can be a catalyst for innovative solutions in the real estate industry. One idea revolves around the concept of "Property Passport." This digital, blockchain-backed passport would contain comprehensive and immutable information about a property, such as its ownership history, maintenance records, energy efficiency certifications, and past transactions. Property owners, real estate agents, and potential buyers or tenants could access this passport, ensuring complete transparency and trust in the property's history. Smart contracts integrated into the system could automate processes like lease agreements, rental payments, and property transfers, streamlining the property management lifecycle.

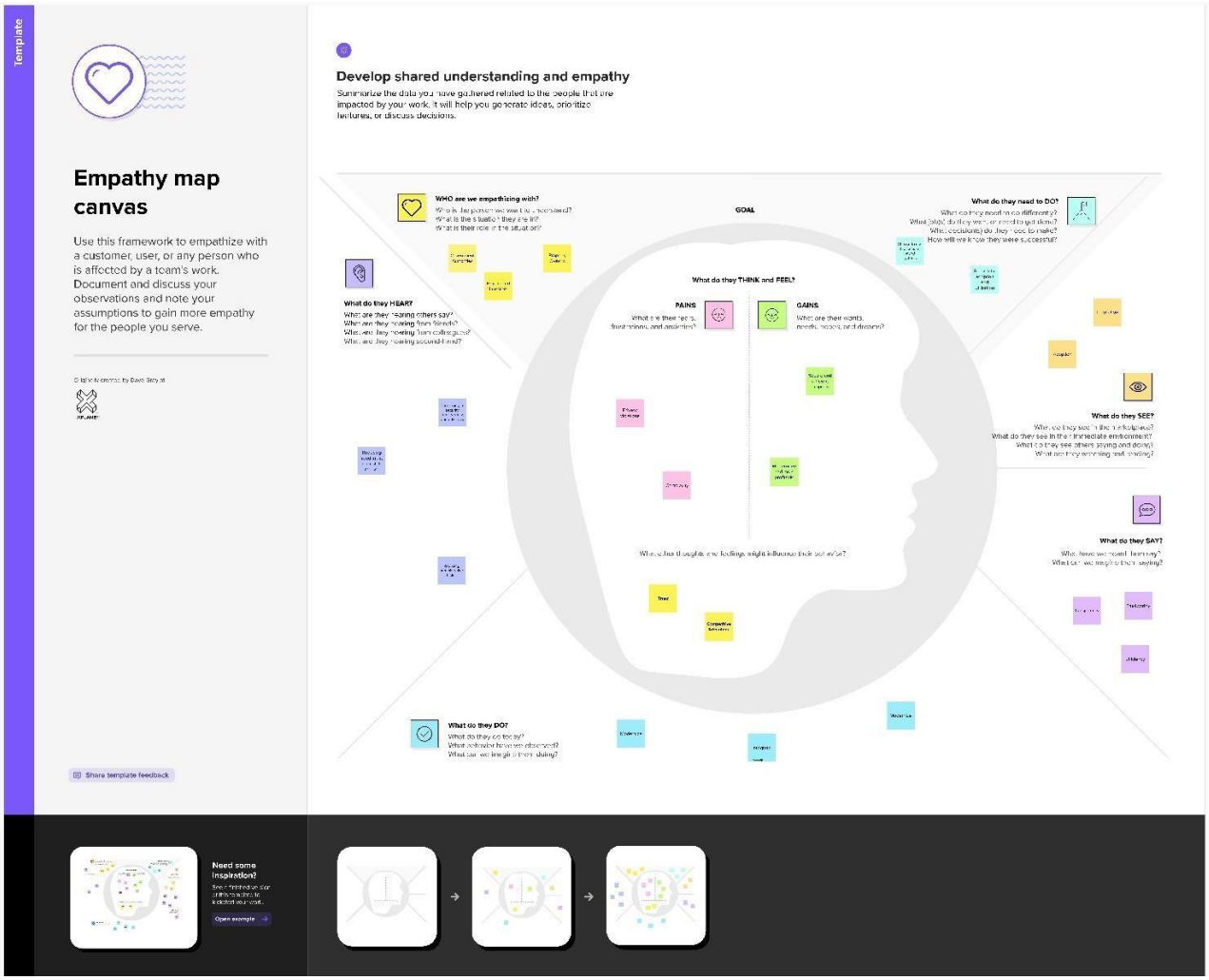
Proposed Solution:

Our proposed smart real estate management system, powered by blockchain technology, is poised to redefine the way we interact with property assets. By integrating blockchain, we aim to create a transparent, secure, and efficient ecosystem for real estate transactions and property management. This system will establish a decentralized ledger to record property details, ownership information, and transactions, ensuring the immutability and accessibility of data. With blockchain's inherent trust and security features, we will reduce fraud and eliminate the need for intermediaries in property transactions. Moreover, we will implement smart contracts, automating rental agreements, payments, and maintenance processes. In doing so, we will streamline real estate operations, reducing costs and complexities for both property owners and tenants. Our vision is to leverage blockchain's potential to usher in a new era of transparency, trust, and convenience in real estate management, benefiting all stakeholders in the industry. blockchain-based smart real estate management centers around the creation of an integrated platform called "RealEstatePro." This comprehensive system would leverage blockchain technology to provide a secure and efficient means of managing real estate properties. Key features of RealEstatePro would include a tamper-proof blockchain ledger for property records, ensuring transparency and trust in property transactions. Smart contracts would be utilized for automating lease agreements, rental payments, and property transfers, reducing the reliance on intermediaries and minimizing disputes. Real Estate Pro would integrate with existing real estate listing services, offering decentralized property listings to reduce centralized control in the industry. Moreover, it would tokenize real estate assets, allowing fractional ownership and investment diversification. The platform would prioritize data security and compliance with evolving regulations, mitigating concerns about privacy and legality.

Empathy Map Canvas

The Empathy Canvas for blockchain-based smart real estate management involves understanding the needs and perspectives of various stakeholders. The Empathy Canvas for blockchain-based smart real estate management involves understanding the needs and perspectives of various stakeholders.

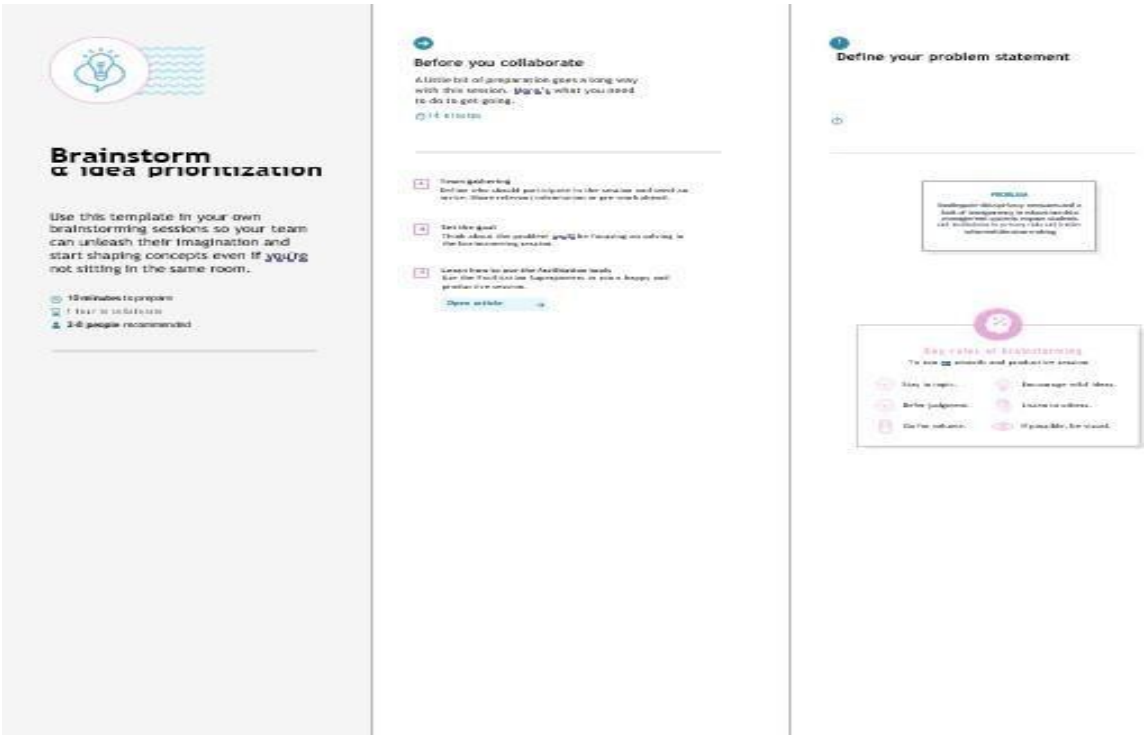
Transparent Education Data Management



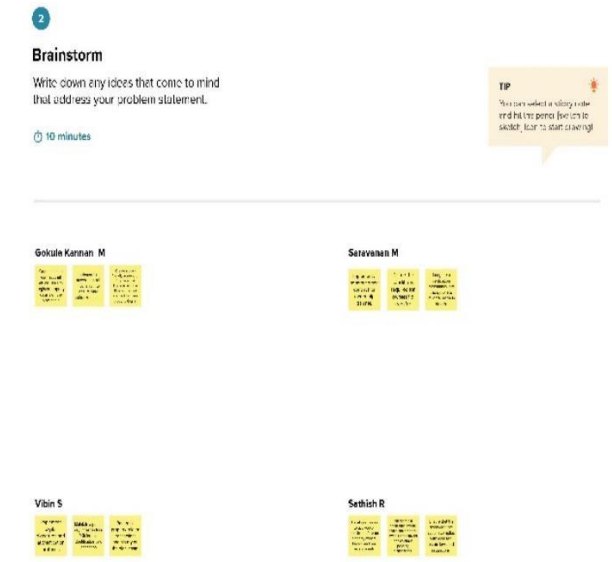
Ideation & Brainstorming

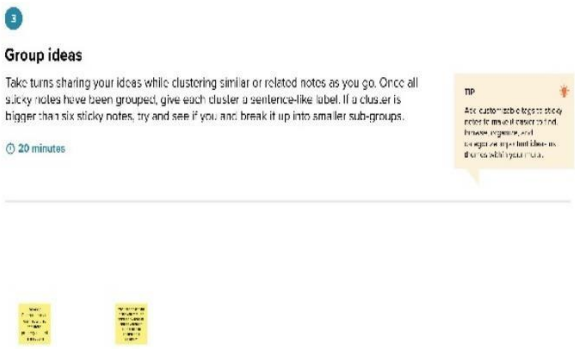
Brainstorming provides a free and open environment that encourages everyone within a teamto participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants areencouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

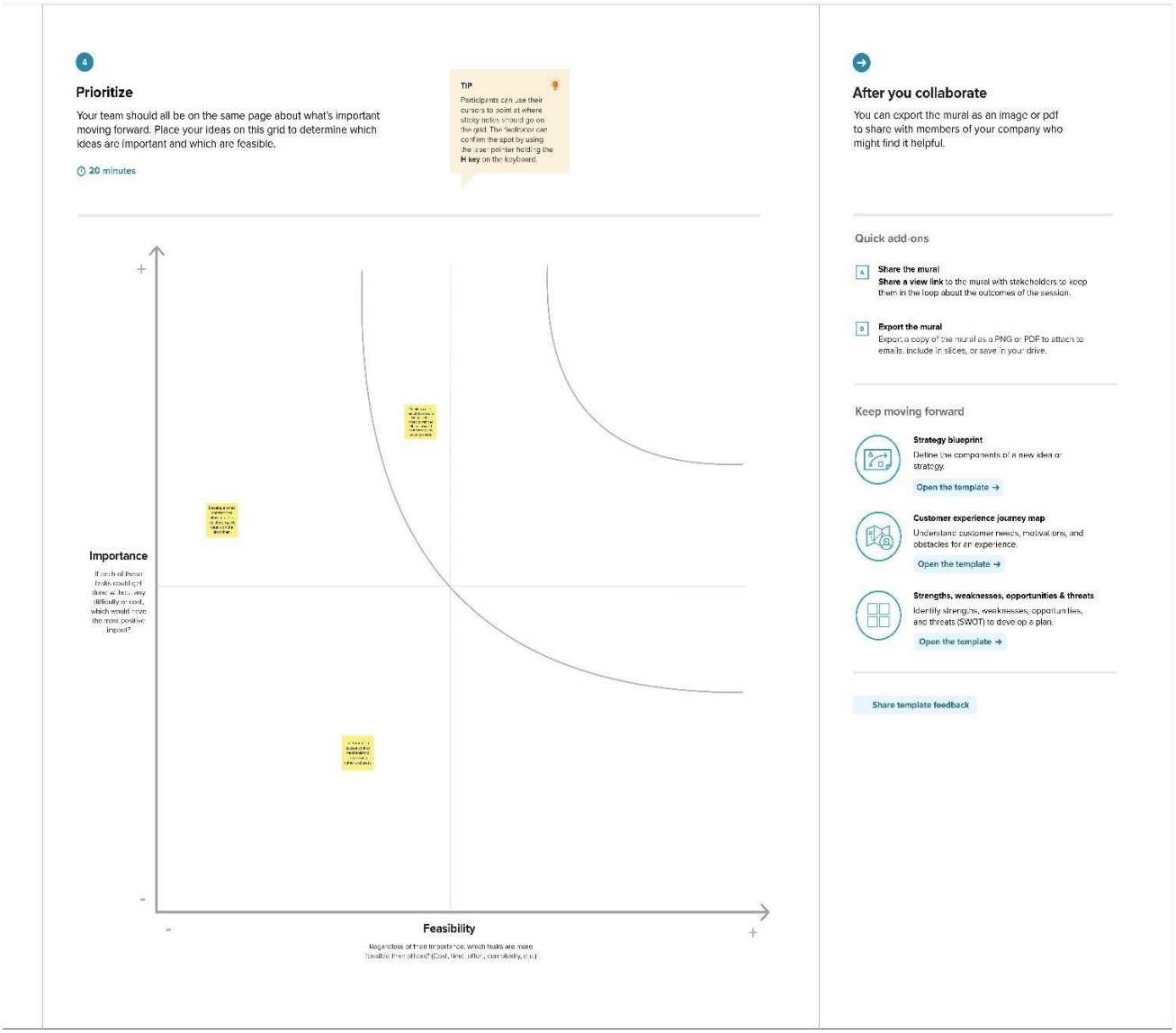


Step-2: Brainstorm, Idea Listing and Grouping





Step-3: Idea Prioritization



4.REQUIREMENT ANALYSIS

Functional Requirements

A blockchain-based smart real estate management system must address several key functional requirements to streamline and secure property transactions, ensure transparency, and enhance overall management. Firstly, the system should enable secure and immutable recording of property ownership and transaction history. This includes creating a blockchain ledger for each property, where all relevant details like property title, ownership changes, and transaction records are securely stored. Access to these records should be permissioned and accessible to relevant parties, such as property owners, real estate agents, and government agencies.Additionally, the system should support smart contracts that automate various aspects of real estate management, such as lease agreements, rent collection, and property maintenance. These contracts can execute automatically based on predefined conditions, reducing the need for intermediaries and minimizing the risk of disputes. Integration with IoT devices and sensors can provide real-time data on property conditions and security, further enhancing management capabilities. Overall, a blockchain-based smart real estate management system should prioritize data security, transparency, and automation to create a more efficient and trustworthy ecosystem for property transactions and management. The system should integrate with payment gateways to facilitate secure and automated rent collection, property maintenance, and utility payments using digital currencies, improving overall efficiency and reducing administrative overhead. Overall, the system should offer a comprehensive solution that streamlines real estate management processes, enhances security, and promotes trust and transparency within the industry.

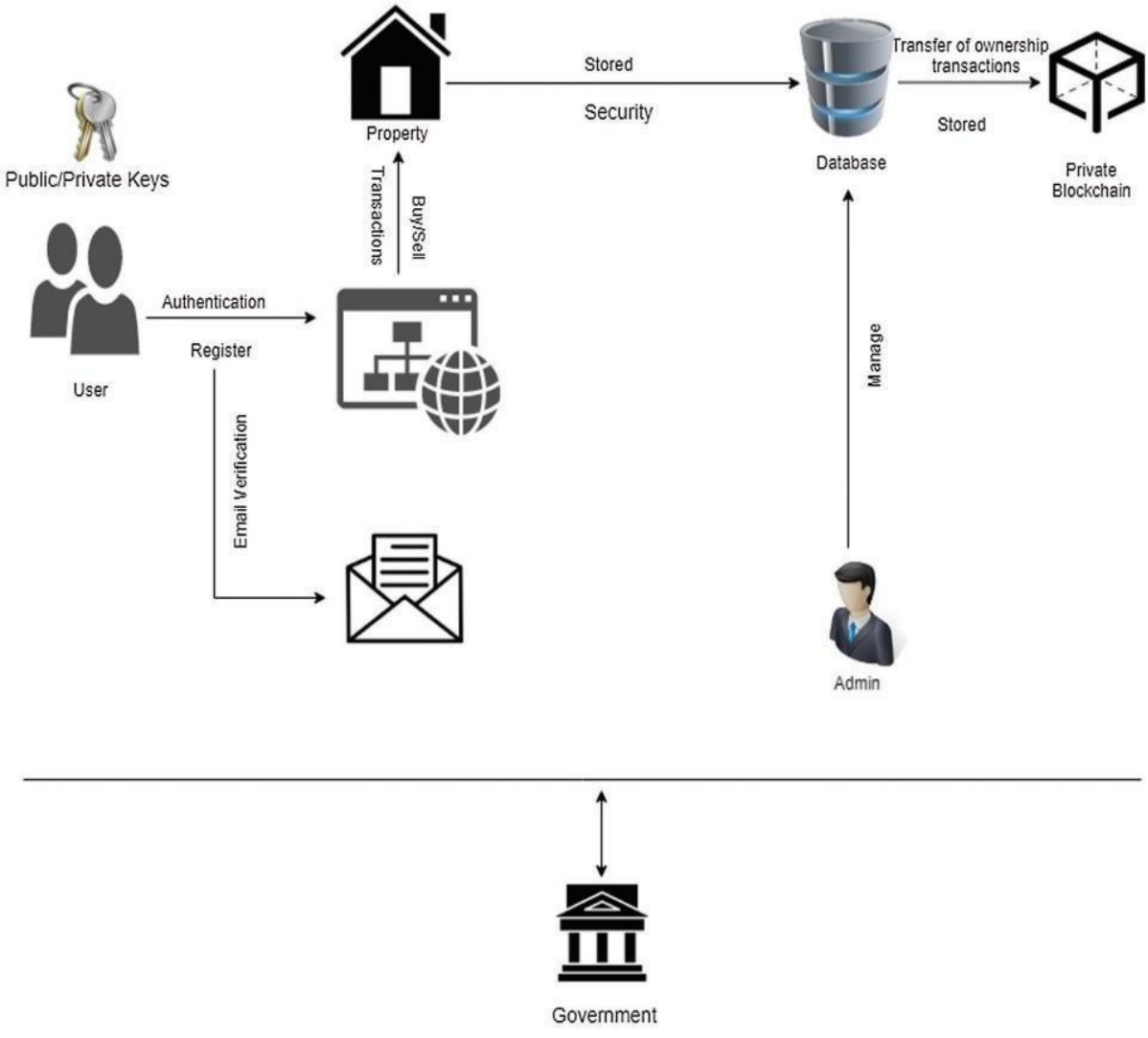
Non Functional Requirements

The Security given the sensitive nature of real estate transactions and data, the system must prioritize robust security measures. This includes data encryption, access control, and cryptographic techniques to safeguard the integrity and confidentiality of information. Compliance with regulatory and legal standards, such as GDPR for data protection, should also be considered. Additionally, the system should have mechanisms to prevent fraud and unauthorized access.The Scalability and Performance as real estate management systems can accumulate vast amounts of data and transactions, scalability and performance are vital non-functional requirements. The blockchain network should be capable of handling a growing number of users and transactions efficiently. This involves optimizing the consensus mechanism, network latency, and transaction confirmation times to ensure a responsive and high-performing system. Scalability solutions, such as sharding or sidechains, should be considered to accommodate increasing workloads while maintaining responsiveness. The Compliance and Regulatory Requirements system should adhere to local and international regulations governing real estate transactions, such as Know Your Customer (KYC) and Anti-Money Laundering (AML) requirements, and enable the easy auditing and reporting of transactions to comply with legal obligations. The Performance Optimization tuning should be an ongoing process to ensure that the blockchain network operates efficiently. This includes optimizing transaction speeds, reducing confirmation times, and minimizing resource consumption to keep operational costs in check. The blockchain system should be designed to seamlessly integrate with other systems and services used in the real estate industry, such as property valuation tools, legal document management systems, and financial institutions, ensuring efficient data exchange and workflows.

5.PROJECT DESIGN

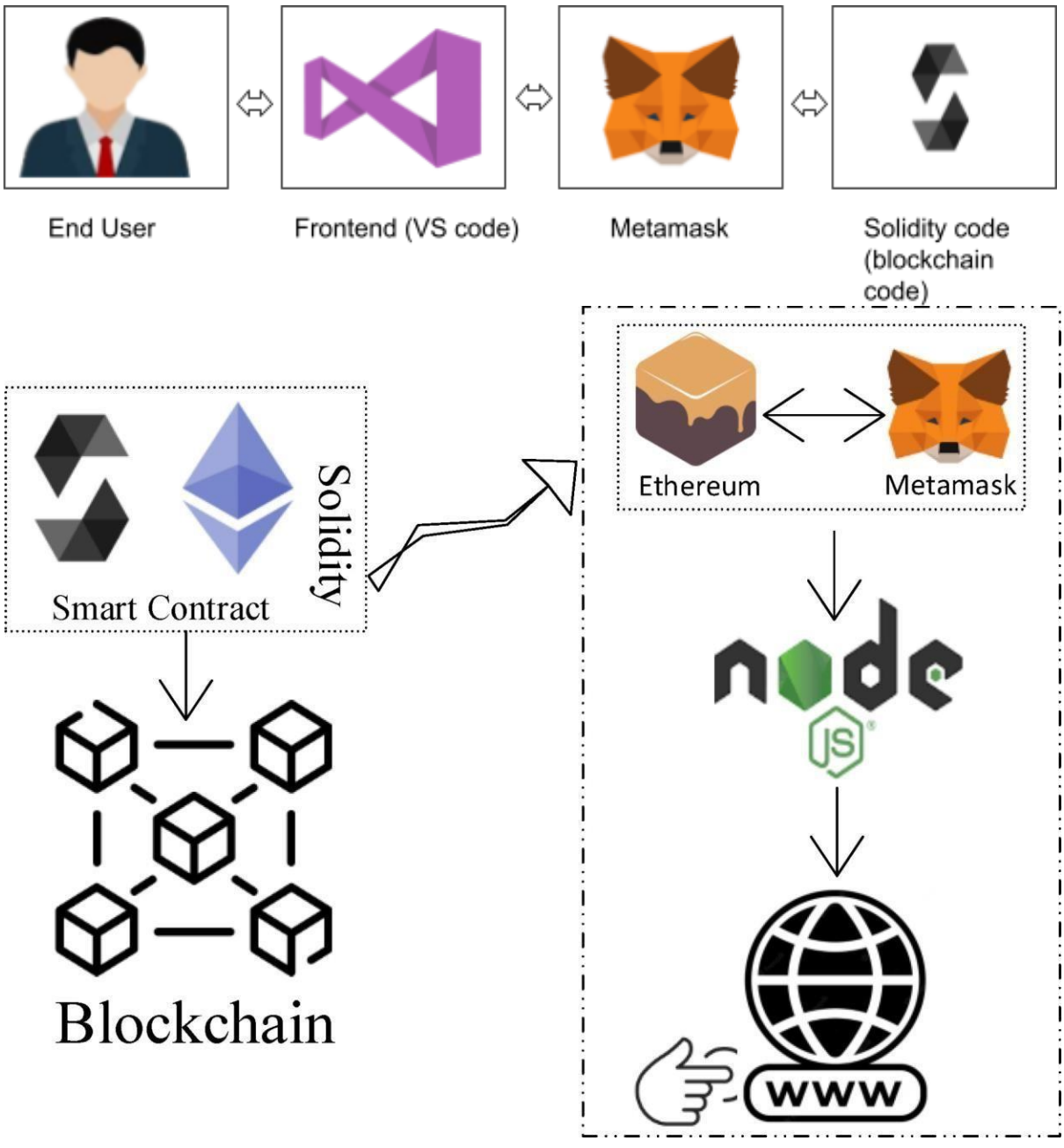
The project aims to implement a blockchain-based system to securely and transparently manage education data, including academic records, certificates, and transcripts, while ensuring data integrity, authenticity, and privacy. The system will serve educational institutions, students, and employers, enabling them to store, access, and verify educational credentials efficiently.

Data Flow Diagrams & User Stories



Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

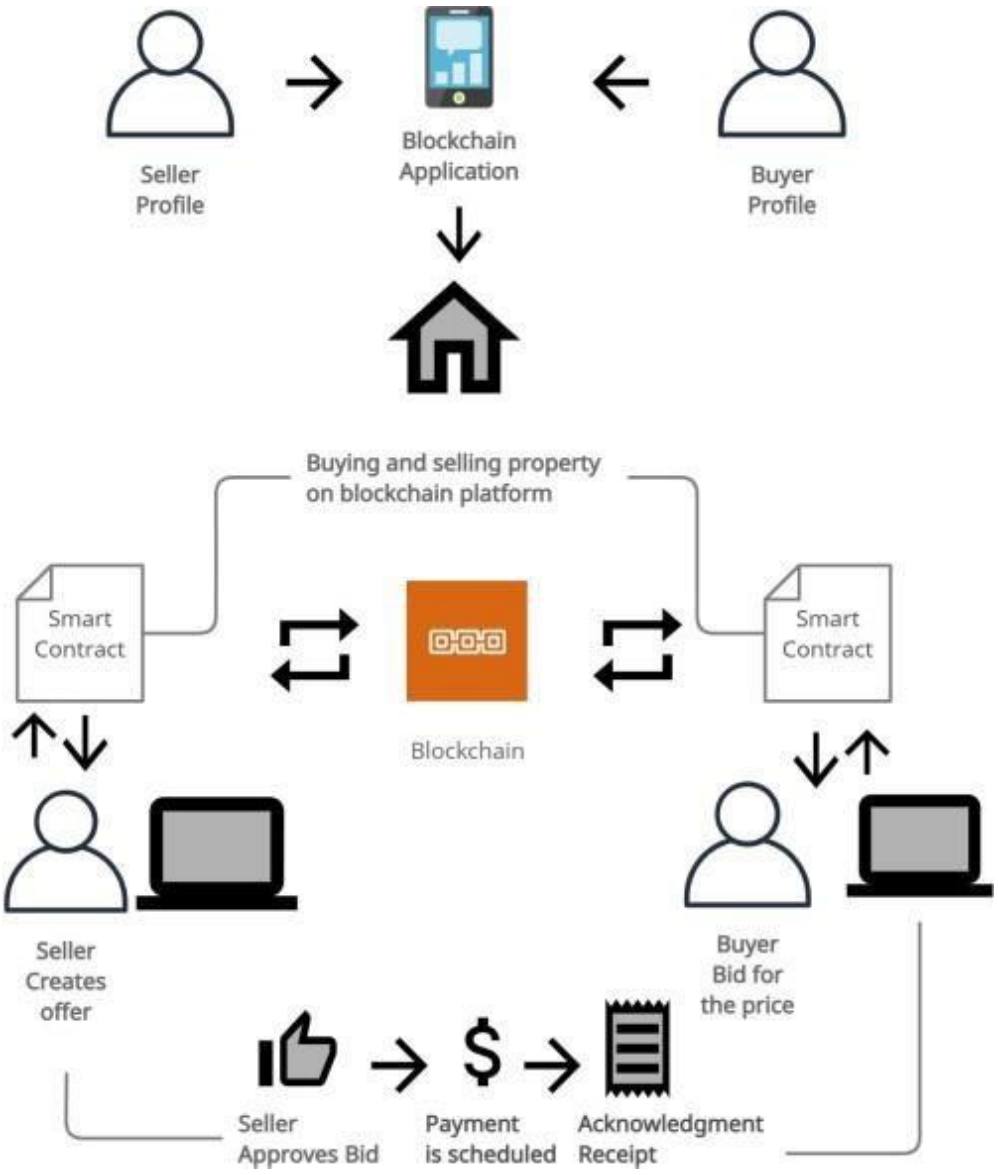


Interaction between web and the Contract

6.PROJECT PLANNING AND SCHEDULING

Project planning and scheduling for transparent education data management is crucial to ensure that educational institutions can effectively collect, store, analyze, and share data while maintaining data security and transparency.

Technical Architecture



Sprint Planning & Estimation

Sprint planning and estimation in a blockchain-based smart real estate management project are critical phases in the development process, contributing to the project's success and efficient delivery. During sprint planning, the development team, which may include blockchain developers, smart contract experts, and real estate domain specialists, collaborates to define the scope of work for the upcoming sprint. This involves breaking down the project into manageable tasks and user stories that align with the overarching project goals. In the context of blockchain-based real estate management, these tasks could include enhancing the blockchain's capabilities, optimizing smart contract functionality, or improving user interfaces.Estimation is an integral part of sprint planning, as it helps in setting realistic goals for the sprint. The team assigns story points or time estimates to each task to gauge the effort required for its completion. Accurate estimation is particularly important in blockchain projects due to the complex and sometimes unpredictable nature of blockchain technology.In the smart real estate management context, the team must consider the intricacies of blockchain, such as consensus algorithms, cryptographic operations, and data management, when estimating development efforts. Additionally, the team needs to assess the impact on real estate management processes, ensuring that the solution aligns with industry standards and legal requirements.Both sprint planning and estimation enable the project team to prioritize tasks, allocate resources effectively, and ensure that the blockchain-based smart real estate management system evolves in a structured and agile manner. Regular sprint reviews and retrospectives help fine-tune the process and adapt to changing requirements, ultimately resulting in a more efficient and responsive real estate management solution.

Sprint Delivery Schedule

The sprint delivery schedule for a blockchain-based smart real estate management project plays a pivotal role in achieving timely and incremental progress. Sprint delivery schedules are typically organized into time-bound iterations, usually lasting two to four weeks, where the development team delivers a set of completed and tested features and enhancements. In the context of smart real estate management, each sprint should be designed to deliver tangible value and improve the platform's functionality. The sprint delivery schedule begins with sprint planning, where the team identifies the scope of work for the upcoming iteration, outlines specific tasks and goals, and assigns story points or time estimates to each task. This information is then used to map out a timeline for the sprint, allocating resources and responsibilities to ensure that the objectives are met within the defined timeframe. Blockchain-based real estate management projects often involve complex components such as smart contracts, data validation, and user interfaces. Each sprint's schedule should accommodate the development, testing, and integration of these components, with a focus on maintaining high-quality standards and security, which are crucial in real estate transactions. By adhering to a well-structured sprint delivery schedule, the project team can incrementally build and enhance the blockchain-based smart real estate management system. Regular sprint reviews provide stakeholders with insights into the project's progress and allow for adjustments as needed. This agile approach ensures that the platform evolves systematically, adapts to changing requirements, and remains on track to deliver a robust and efficient solution for real estate management. For a smart real estate management system based on blockchain, these tasks may include enhancing security protocols, implementing new smart contract features, integrating with external data sources, or improving the user interface. Each sprint is planned to deliver tangible and valuable increments to the system.

7 CODING & SOLUTIONING

Developing a blockchain-based smart real estate management system

involves a combination of smart contract development, front-end design, and back-end infrastructure. Below is a simplified example of a code snippet for property tokenization using Solidity (a smart contract language) and a high-level solution.

Feature

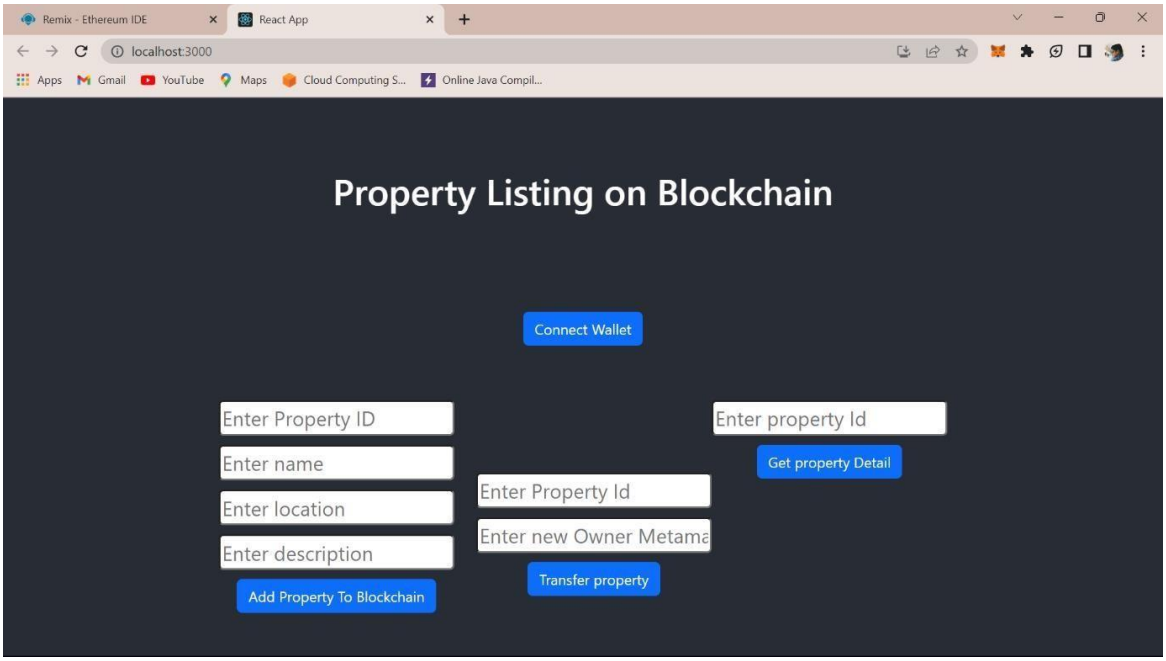
```
pragma solidity ^0.8.0;

contract RealEstateOwnership {
    address public owner;

    struct Property {
        address owner;
        uint256 value;
    }

    mapping(uint256 => Property) public properties;
    uint256 public propertyCount;

    event PropertyTransferred(uint256 indexed propertyId, address from, address to);

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "Only the owner can call this function.");
        _;
    }

    function createProperty(uint256 _value) public onlyOwner {
        propertyCount++;
        properties[propertyCount] = Property(msg.sender, _value);
    }

    function transferProperty(uint256 _propertyId, address _to) public {
        Property storage property = properties[_propertyId];
        require(property.owner == msg.sender, "You don't own this property.");
        property.owner = _to;
        emit PropertyTransferred(_propertyId, msg.sender, _to);
    }
```

}

Feature

1. Transparency and Immutability: Blockchain technology ensures transparent and immutable property records. All transactions and ownership changes are recorded in a tamper-proof ledger that is accessible to authorized parties. This feature enhances trust and reduces the risk of fraud and disputes, as historical data cannot be altered. It provides a clear and verifiable history of property ownership and transactions.

2. Smart Contracts: Smart contracts are self-executing contracts with the terms of the agreement written directly into code. In smart real estate management, these contracts automate various processes, such as property transfers, lease agreements, and rental payments. They execute automatically when predefined conditions are met, reducing the need for intermediaries, streamlining transactions, and minimizing the potential for human errors or delays. This feature improves efficiency and cost-effectiveness in real estate management



8 PERFORMANCE METRICS

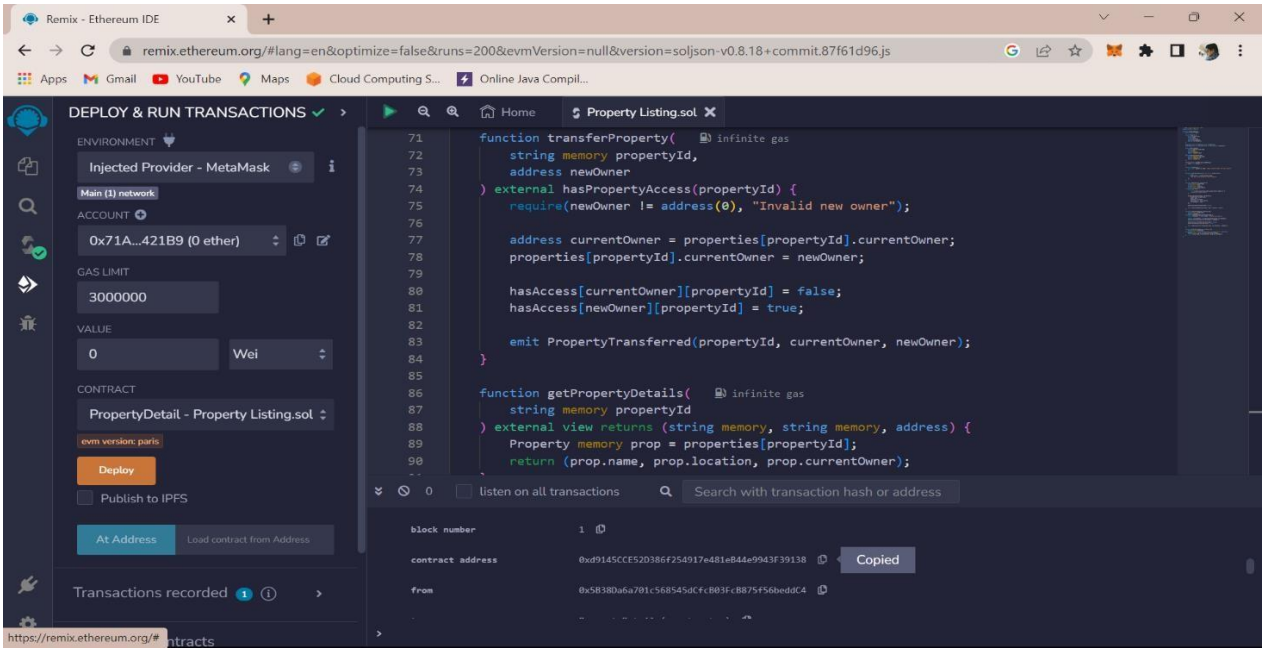
Performance Metrics

Performance metrics for a blockchain-based smart real estate management system encompass several key aspects, including transaction throughput, latency, scalability, and security. Transaction throughput measures the number of real estate transactions processed per second, ensuring that the system can handle high transaction volumes efficiently. Latency assesses the time it takes for a transaction to be confirmed, providing insights into the responsiveness of the platform. Scalability metrics evaluate the system's ability to expand and accommodate a growing number of properties and users while maintaining performance. Security metrics focus on the prevention of unauthorized access, data integrity, and compliance with privacy regulations, guaranteeing that sensitive real estate information is protected from potential threats. Additionally, other metrics like uptime and auditability are crucial for ensuring system availability and regulatory compliance in a blockchain-based smart real estate management solution. crucial to ensure its efficiency and effectiveness. Key metrics to monitor include transaction processing speed, blockchain network scalability, data storage capacity, and response time. Measuring the system's security against potential threats, such as unauthorized access or fraudulent transactions, is vital. User adoption and satisfaction should also be tracked, as they reflect the system's usability and overall success. Ensuring compliance with industry standards, regulatory requirements, and legal obligations is another critical performance aspect. Regular monitoring and optimization of these metrics are essential to maintain a robust and reliable real estate management system that leverages blockchain technology. A balance between performance, security, and scalability should guide the choice of metrics to guarantee a robust and responsive real estate management solution.

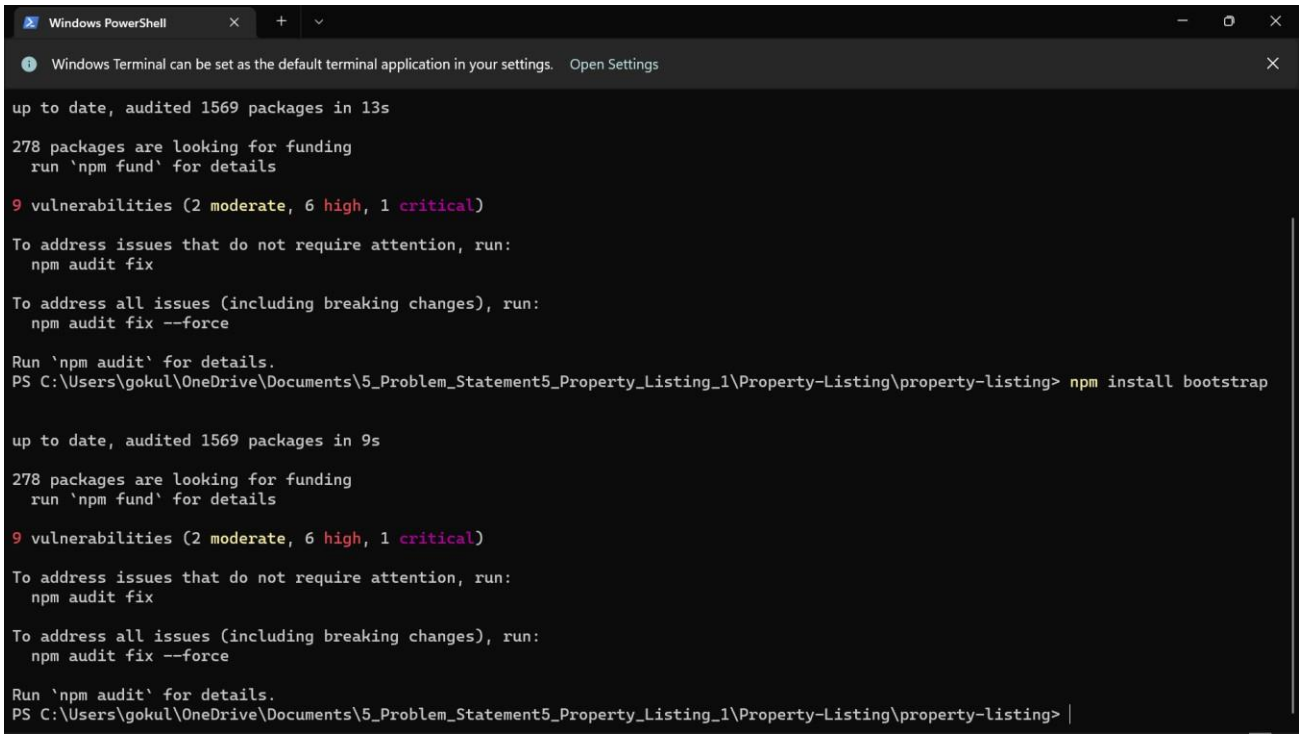
9 RESULTS

Blockchain-based smart real estate management is a transformative approach that holds the potential to revolutionize the traditional real estate industry. By leveraging the blockchain's inherent features of transparency, security, and efficiency, this technology offers a promising array of benefits. It ensures transparent and tamper-proof property transactions, reducing fraud and disputes, while smart contracts automate complex processes, saving time and resources. Blockchain's robust security features protect sensitive property data and enable property verification, enhancing trust in ownership records. Moreover, tokenization enables fractional ownership and accessibility for a broader range of investors. However, challenges such as scalability, regulatory hurdles, and the need for widespread adoption remain. Despite these challenges, the adoption of blockchain-based smart real estate management is an exciting development that has the potential to streamline real estate processes, reduce costs, and make the industry more accessible and efficient.

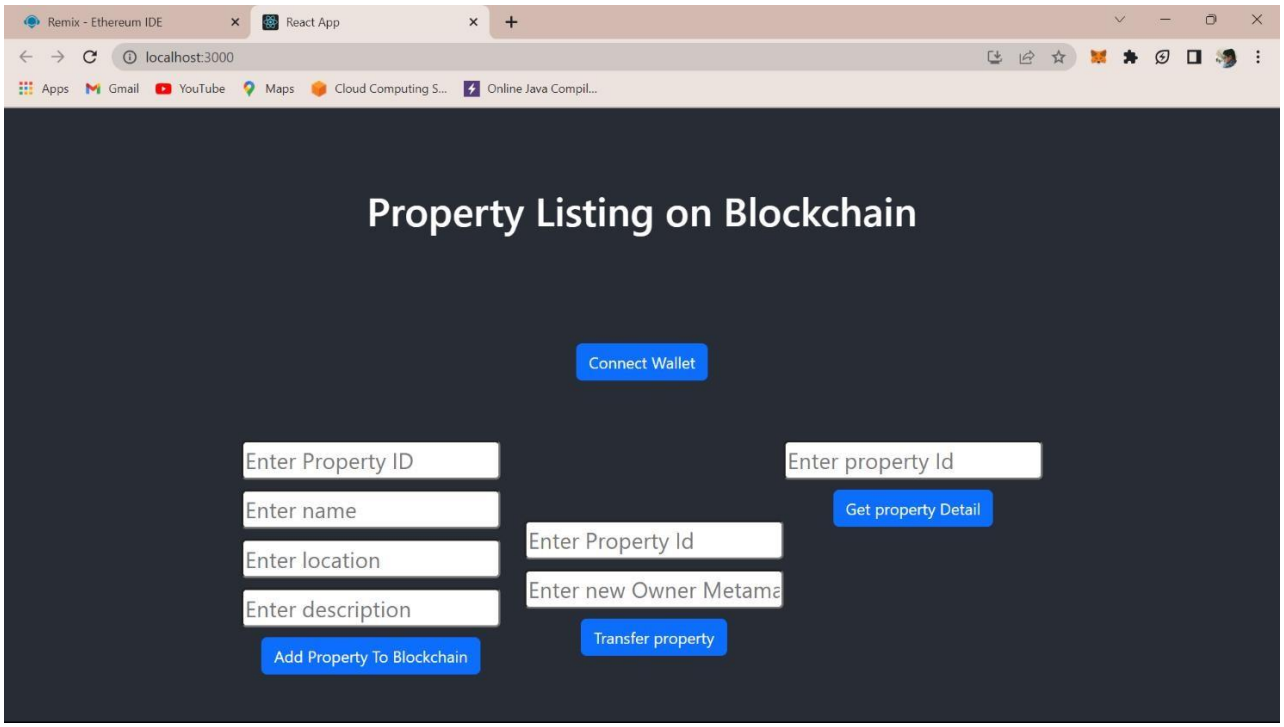
9.1 Output Screenshots



Creating smart contract



Installing packages



Output Screenshot

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

Blockchain-based smart real estate management offers several advantages that can revolutionize the industry. It enhances transparency and trust. By storing real estate transaction data on a secure, tamper-resistant blockchain, stakeholders, including buyers, sellers, and government authorities, can access a transparent and immutable ledger of property records. This reduces the risk of fraud, ensures the accuracy of ownership information, and streamlines property transfers, ultimately boosting trust in the real estate market.Blockchain streamlines and automates processes. Smart contracts, which are self-executing contracts with predefined rules, can facilitate automated property transactions, reducing the need for intermediaries like banks and lawyers. This not only saves time and costs but also increases the efficiency of real estate management by automating tasks such as payment processing and property transfer.

DISADVANTAGES:

Blockchain-based smart real estate management offers several advantages, but it also comes with disadvantages. One notable drawback is the complexity and cost of implementation. Developing and maintaining blockchain systems requires specialized skills and can be resource-intensive, especially for smaller real estate firms. Moreover, the adoption of blockchain in the real estate industry may require changes to existing processes, which can be met with resistance from traditional stakeholders. Scalability and speed issues can also arise, particularly if the blockchain network becomes congested, leading to slower transaction processing times. Lastly, regulatory uncertainties and legal challenges can pose obstacles as the legal framework for blockchain-based real estate transactions is still evolving, which can result in compliance and governance concerns.

11. CONCLUSION

In conclusion, the adoption of blockchain technology in the realm of smart real estate management represents a significant step towards a more efficient, transparent, and accessible real estate industry. This innovative approach leverages blockchain's inherent qualities of security, immutability, and trust to streamline various aspects of the real estate lifecycle. From the tokenization of properties and the use of smart contracts to property verification and real-time data tracking, the system offers a robust solution for property owners, investors, and tenants. Furthermore, its potential to integrate with DeFi platforms, enhance sustainability practices, and simplify cross-border transactions highlights its adaptability to the evolving needs of the real estate market. As this technology matures and user-friendly applications become widespread, the blockchain-based smart real estate management system is poised to reshape the industry, offering greater accessibility and efficiency while reducing friction and risks in real estate transactions. It is an exciting development that holds the promise of unlocking new opportunities and transforming the way we approach real estate management in the future.

12. FUTURE SCOPE

The future scope for blockchain-based smart real estate management systems is highly promising, poised to revolutionize the traditional real estate industry. Blockchain technology offers a secure, transparent, and efficient framework for property transactions and management. One key area of potential growth is the tokenization of real estate, which can enable fractional ownership and make property investment more accessible. Smart contracts will continue to automate various aspects of real estate transactions, reducing reliance on intermediaries and minimizing disputes. Property verification and title records will become more reliable, reducing fraud and enhancing trust in the system. Integrating the Internet of Things (IoT) with blockchain will provide real-time

property data, enabling proactive maintenance and management. Decentralized property listings will challenge the dominance of centralized listing services, and cross-border transactions will become more streamlined. Data security and regulatory compliance will be significantly improved, instilling confidence among stakeholders. Furthermore, blockchain-based real estate systems have the potential to integrate with decentralized finance (DeFi) platforms, offering new ways to invest, lend, and trade real estate-related assets. As this technology matures and user-friendly applications are developed, the future of blockchain-based real estate management appears to be dynamic and transformative, offering benefits for property owners, investors, and tenants alike.

13. APPENDIX

Source Code

Property Listing.sol

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

contract PropertyDetail{
    address public owner;

    struct Property {
        string propertyId;
        string name;
        string location;
        string discription;
        address currentOwner;
    }

    mapping(string => Property) public properties;
    mapping(address => mapping(string => bool)) public hasAccess;

    event PropertyAdded(
        string indexed propertyId,
        string name,
        string  location,
        address indexed owner
    );
```

```
event PropertyTransferred(
    string indexed propertyId,
    address indexed from,
    address indexed to
);

constructor() {
    owner = msg.sender;
}

modifier onlyOwner() {
    require(msg.sender == owner, "Only contract owner can call this");
    _;
}

modifier hasPropertyAccess(string memory propertyId) {
    require(
        hasAccess[msg.sender][propertyId],
        "You don't have access to this property"
    );
    _;
}

function addProperty(
    string memory propertyId,
    string memory name,
    string memory location,
    string memory _description
) external onlyOwner {
    require(
        bytes(properties[propertyId].propertyId).length == 0,
        "Property already exists"
    );

    properties[propertyId] = Property({
        propertyId: propertyId,
        name: name,
        location: location,
        discription : _description,
        currentOwner: owner
    });

    hasAccess[owner][propertyId] = true;
```

```
        emit PropertyAdded(propertyId, name, location, owner);
    }

    function transferProperty(
        string memory propertyId,
        address newOwner
    ) external hasPropertyAccess(propertyId) {
        require(newOwner != address(0), "Invalid new owner");

        address currentOwner = properties[propertyId].currentOwner;
        properties[propertyId].currentOwner = newOwner;

        hasAccess[currentOwner][propertyId] = false;
        hasAccess[newOwner][propertyId] = true;

        emit PropertyTransferred(propertyId, currentOwner, newOwner);
    }

    function getPropertyDetails(
        string memory propertyId
    ) external view returns (string memory, string memory, address) {
        Property memory prop = properties[propertyId];
        return (prop.name, prop.location, prop.currentOwner);
    }
}
```

connector.js

```
const { ethers } = require("ethers");

const abi = [
 {
  "inputs": [],
  "stateMutability": "nonpayable",
  "type": "constructor"
 },
 {
  "anonymous": false,
  "inputs": [
   {
    "indexed": true,
    "internalType": "string",
    "name": "propertyId",
    "type": "string"
```

```
  },
  {
   "indexed": false,
   "internalType": "string",
   "name": "name",
   "type": "string"
  },
  {
   "indexed": false,
   "internalType": "string",
   "name": "location",
   "type": "string"
  },
  {
   "indexed": true,
   "internalType": "address",
   "name": "owner",
   "type": "address"
  }
 ],
 "name": "PropertyAdded",
 "type": "event"
},
{
 "anonymous": false,
 "inputs": [
  {
   "indexed": true,
   "internalType": "string",
   "name": "propertyId",
   "type": "string"
  },
  {
   "indexed": true,
   "internalType": "address",
   "name": "from",
   "type": "address"
  },
  {
   "indexed": true,
   "internalType": "address",
   "name": "to",
   "type": "address"
  }
```

```
 ],
 "name": "PropertyTransferred",
 "type": "event"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "propertyId",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "name",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "location",
   "type": "string"
  },
  {
   "internalType": "string",
   "name":  "_description",
   "type": "string"
  }
 ],
 "name": "addProperty",
 "outputs": [],
 "stateMutability": "nonpayable",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "propertyId",
   "type": "string"
  }
 ],
 "name": "getPropertyDetails",
 "outputs": [
  {
   "internalType": "string",
```

```
   "name": "",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  },
  {
   "internalType": "address",
   "name": "",
   "type": "address"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "address",
   "name": "",
   "type": "address"
  },
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  }
 ],
 "name": "hasAccess",
 "outputs": [
  {
   "internalType": "bool",
   "name": "",
   "type": "bool"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [],
 "name": "owner",
```

```
 "outputs": [
  {
   "internalType": "address",
   "name": "",
   "type": "address"
  }
 ],
 "stateMutability": "view",
 "type": "function"
},
{
 "inputs": [
  {
   "internalType": "string",
   "name": "",
   "type": "string"
  }
 ],
 "name": "properties",
 "outputs": [
  {
   "internalType": "string",
   "name": "propertyId",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "name",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "location",
   "type": "string"
  },
  {
   "internalType": "string",
   "name": "discription",
   "type": "string"
  },
  {
   "internalType": "address",
   "name": "currentOwner",
   "type": "address"
```

```
   }
  ],
  "stateMutability": "view",
  "type": "function"
 },
 {
  "inputs": [
   {
    "internalType": "string",
    "name": "propertyId",
    "type": "string"
   },
   {
    "internalType": "address",
    "name": "newOwner",
    "type": "address"
   }
  ],
  "name": "transferProperty",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
 }
]

if (!window.ethereum) {
 alert('Meta Mask Not Found')
 window.open("https://metamask.io/download/")
}

export const provider = new ethers.providers.Web3Provider(window.ethereum);
export const signer = provider.getSigner();
export const address = "00xd9145CCE52D386f254917e481eB44e9943F39138"

export const contract = new ethers.Contract(address, abi, signer)
import React, { useState } from "react";
import { Button, Container, Row, Col } from 'react-bootstrap';
import '../../node_modules/bootstrap/dist/css/bootstrap.min.css';
import { contract } from "./connector";

function Home() {
 const [Id, setId] = useState("");
 const [name, setname] = useState("");
 const [location, setLocation] = useState("");
```

```
const [des, setDes] = useState("");
const [TransferPropertId, setTransferPropertId] = useState("");
const [MeatAddr, setMeatAddr] = useState("");
const [PropDetailId, setPropDetailId] = useState("");
const [PropDetails, setPropDetails] = useState("");
const [Wallet, setWallet] = useState("");




const handleId = (e) => {
 setId(e.target.value)
}

const handleName = (e) => {
 setname(e.target.value)
}

const handleLocation = (e) => {
 setLocation(e.target.value)
}

const handleDes = (e) => {
 setDes(e.target.value)
}

const handleAddProperty = async() => {
 try {
  let tx  = await contract.addProperty(Id.toString(),name,location,des)
  let wait = await tx.wait()
  console.log(wait);
  alert(wait.transactionHash)
 } catch (error) {
  alert(error)
 }
}

const handleTransferPropertyId = (e) => {
 setTransferPropertId(e.target.value)
}

const handleMetaAddr = (e) => {
 setMeatAddr(e.target.value)
}
```

```
const handletransferProperty = async () => {
 try {
  let tx = await contract.transferProperty(TransferPropertId.toString(), MeatAddr)
  let wait = await tx.wait()
  console.log(wait);
  alert(wait.transactionHash)

 } catch (error) {
  alert(error)
 }
}

const handlePropId = (e) => {
 setPropDetailId(e.target.value)
}

const handlePorpDetails = async() => {
 try {
  let tx = await contract.getPropertyDetails(PropDetailId.toString())
  let arr = []

  tx.map(e => arr.push(e))

  setPropDetails(arr)
  console.log(tx);
  // alert(tx)
 } catch (error) {
  alert(error)
 }
}

const handleWallet = async () => {
 if (!window.ethereum) {
  return alert('please install metamask');
 }

 const addr = await window.ethereum.request({
  method: 'eth_requestAccounts',
 });

 setWallet(addr[0])

}
```

```
 return (
  <div>
   <h1 style={{ marginTop: "30px", marginBottom: "80px" }}>Property Listing on
Blockchain</h1>
   {!Wallet ?

    <Button onClick={handleWallet} style={{ marginTop: "30px", marginBottom:
"50px" }}>Connect Wallet </Button>
    :
    <p style={{ width: "250px", height: "50px", margin: "auto", marginBottom:
"50px", border: '2px solid #2096f3' }}>{Wallet.slice(0, 6)}... {Wallet.slice(-6)}</p>
   }
   <Container style={{ display: "flex" }}>
    <Row >
     <Col>
      <div>


       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleId} type="number" placeholder="Enter Property ID" value={Id}
/> <br />

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleName} type="string" placeholder="Enter name" value={name} />
<br />

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleLocation} type="string" placeholder="Enter location"
value={location} /><br />

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleDes} type="string" placeholder="Enter description" value={des}
/><br />

       <Button onClick={handleAddProperty} style={{ marginTop: "10px" }}
variant="primary">Add Property To Blockchain</Button>
      </div>
     </Col>
     <Col>
      <div style={{marginTop:"80px"}}>

       <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleTransferPropertyId} type="string" placeholder="Enter Property
```

```
Id" value={TransferPropertId} /><br />


      <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handleMetaAddr} type="string" placeholder="Enter new Owner
Metamask Address" value={MeatAddr} /><br />


      <Button onClick={handletransferProperty} style={{ marginTop: "10px" }}
variant="primary">Transfer property</Button>

     </div>
    </Col>
   </Row>
   <Row>
    <Col>
     <div>

      <input style={{ marginTop: "10px", borderRadius: "5px" }}
onChange={handlePropId} type="string" placeholder="Enter property Id"
value={PropDetailId} /><br />


      <Button onClick={handlePorpDetails} style={{ marginTop: "10px" }}
variant="primary">Get property Detail</Button>
       {PropDetails ? PropDetails?.map(e => {
        return <p>{e}</p>
       }) : <p></p>}
      </div>
     </Col>
    </Row>

   </Container>

  </div>
 )
}

export default Home;
```