A Decentralized Application (DApp) on Trading of Real Estate Assets using Blockchain and Artificial Intelligence

A PROJECT REPORT

Submitted by

DARIN JOSHUA D	**(211418104043)**
GLADWIN JOSEPH SOLOMON B	**(211418104064)**
JOEL NITHISH KUMAR M	**(211418104102)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MAY 2022

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“A Decentralized Application (DApp) on Trading of Real Estate Assets using Blockchain and Artificial Intelligence”** is the bonafide work of “**Darin Joshua D (211418104043), Gladwin Joseph Solomon B (211418104064), Joel Nithish Kumar M (211418104102)”** who carried out the project work under my supervision.

SIGNATURE

Dr. S. MURUGAVALLI ,M.E., Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mr. A. KARTHIKEYAN, M.Tech.,
SUPERVISOR
ASSISTANT PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above mentioned students were examined in End Semester project viva-voice held on ______________________.

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **Darin Joshua D (211418104043), Gladwin Joseph Solomon B (211418104064), Joel Nithish Kumar M (211418104102)** hereby declare that this project report titled **“A Decentralized Application (DApp) on Trading of Real Estate Assets using Blockchain and Artificial Intelligence”**, under the guidance of **Mr. A. KARTHIKEYAN, M.Tech.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHI KUMAR, M.E., Ph.D** and **Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S. MURUGAVALLI , M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our Project Guide **Mr. A. KARTHIKEYAN, M.Tech.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

We would also like to thank our Project Coordinator **Dr. G. SENTHIL KUMAR, Ph.D,** and all our family members, friends for their moral support and guidance.

Darin Joshua D	**(211418104043)**
Gladwin Joseph Solomon B	**(211418104064)**
Joel Nithish Kumar M	**(211418104102)**

ABSTRACT

Real estate is private property in the form of buildings and land. Real estate can be used for residential, commercial or industrial purposes and includes all resources of the land such as water and minerals. It is often the most valuable investment a person owns, and the value of real estate is an important indicator of economic stability. Real estate is a legal term that describes an individual's right to land or a building. In most cases, real estate includes land and real estate as defined by the local ordinance in which the real estate resides. Land or real estate that no one owns is not considered real estate. It is a common form of investment. The land and property you own cannot only increase the value and bring stable profits to the owner, but also provide stable income and even financial stability. In recent years, the growth of real estate technology has had an influence on how the sector operates and has transformed the face of the business. One such technology is Blockchain. Due to its wide range of applications in the fields of computer science and technology, blockchain technology has gotten a lot of interest recently. The financial industry has been affected by the rapid development of blockchain technology, which has resulted in the emergence of a new crypto-economy and the use of various crypto-currencies today. Then came smart contracts, which established confidence between several untrustworthy parties without the involvement of a trusted third-party. Artificial intelligence is another popular technology that allows machines to think for themselves. The combination of blockchain with artificial intelligence (AI) can help make apps more secure and work without the need for third-party assistance (in most cases). Decentralized Applications, or DApps, are applications that use Blockchain to allow both parties involved in a transaction to engage and conduct transactions directly without the use of a traditional intermediary that works for a fee or transaction fee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SYMBOL	TITLE
DApp	Decentralized Application
AI	Artificial Intelligence
ER	Entity Relationship
UML	Unified Modeling Language
CSS	Cascading Style Sheets
JS	JavaScript
UI	User Interface
UX	User Experience
API	Application Programming Interface
USD	United States Dollar
PoW	Proof of Work
PoS	Proof of Stake
TF	Term Frequency
IDF	Inverse Document Frequency
RL	Reinforcement Learning
PUT	Parameterized Unit Tests
SRS	System Requirement Specification
CDBC	Central Bank Digital Currency

CHAPTER 1

INTRODUCTION

Artificial intelligence (AI) and blockchain have become two of the most trending and disruptive technologies, especially with the rise of Web 3.0, which makes the Internet safer and more reliable than ever before than the predecessor technologies. The combination of AI and blockchain technology allows distributed artificial intelligence to access the shared data store, transactions and logs in a decentralized, secure and trusted way, with decentralized algorithms. However, they are given a common goal. A decentralized application (DApp) is a digital application or program that is stored and runs on your computer's blockchain or peer-to-peer (P2P) network, and not on a single computer. Here, we describe and build an efficient decentralized application (DApp) that helps you to transact and validate real estate assets using blockchain and artificial intelligence features. The DApp is named "Real Trado" and is built on a blockchain network with a ledger payment gateway and AI features that allow users to buy/rent/sell assets securely and efficiently on one platform. only. This system stores transaction details on the blockchain, making it more secure against attacks and tampering of data and information. "Real Trado" is a unique solution offered as a DApp in the first phase of future Web 3.0 development, providing users with an intuitive, secure platform to invest in real estate without any hassle like traditional methods, platform, or system.

Fig 2.1 Logo of the DApp platform

1.1 PROBLEM DEFINITION

Real estate and asset transactions require a ton of paperwork to verify financial information and ownership and then transfer deeds and titles to new owners. The current system is centralized and has separate services which uses a client/server architecture and is maintained & owned by a central organization. E.g., Airbnb, Home Away, etc. There is also a hidden transaction fee or commission for every transaction involved. It also isn’t in any means a secure or straightforward approach.

Each transaction and customer details are stored in a central server and the user has no control over their data as it is owned and managed by an organization which can manipulate the prices, information and resources according to their business needs. There is no graceful degradation due to the Centralized server. It also has lesser security and privacy of the user resources and data.

Using blockchain technology and artificial intelligence features to record real estate transactions, listings, research and payment can provide a more secure, accessible means of verifying and transferring ownership. A decentralized medium can speed up transactions, reduce paperwork, remove need of middleware or commissions and save money enhancing privacy and direct dealings with buyer and seller.

CHAPTER 2
LITERATURE SURVEY

[1] A. Tapscott and T. Don, How blockchain is changing finance

This explains how two or more parties, whether businesses or individuals who may or may not know each other, can use blockchain to avoid forged agreements, make transactions, and build value without relying on intermediaries to verify their identities, establish trust, or perform critical business logic like contracting, clearing, settling, and record-keeping tasks that are foundational to all forms of commerce.

[2] Blockchain for Real Estate Yarlagadda Jyotsna, Keerthi Gampala

Despite the importance of real estate in a country's economy, the existing system has numerous flaws ranging from property searches to property agreements, sale and purchase, transaction of money, and the role of middleman. Real estate and blockchain can work together. The land business process can use the blockchain model to firmly establish the real estate concept in more than one of its wearing exercises.

[3] An Overview on Integrating Machine Learning with Blockchain Kevin Shaju Varghese

Machine learning and blockchain are two of the most disruptive technologies that will radically alter how we live, work, and interact. Furthermore, smart contracts enable blockchain to manage interactions between members without the need for an intermediary or a trusted third party. For many components of machine learning, including as data, algorithms, and computing power, blockchain can power decentralized marketplaces and coordination platforms.

[4] Decentralized Applications: The Blockchain-Empowered Software System - Wei Cai, Zehua Wang

Blockchain systems provide infrastructure for decentralized applications by combining cryptography, peer-to-peer networking, and consensus methods. In this paper they have analyzed the history of blockchain technology and defined their popular meanings in this article. They have proposed DApp application scenarios, which we believe will be the focus of future blockchains. They have also covered ideal DApp traits and recent blockchain development initiatives, such as payment channels, unique consensus methods, and non-public blockchains.

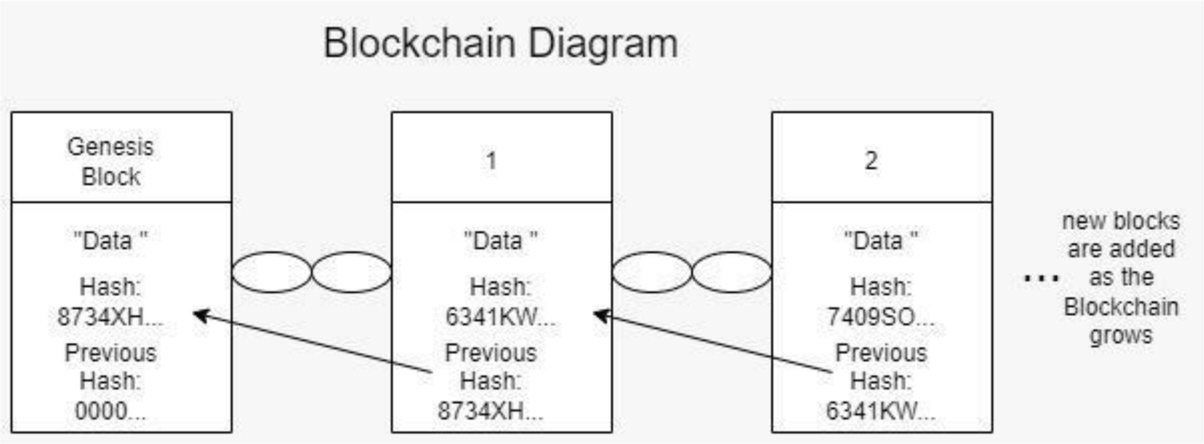


Fig 2.2 Blockchain diagram

CHAPTER 3
SYSTEM ANALYSIS

Various aspects of the real estate industry, its systems, methods and criteria are taken into perspective for the system analysis to prepare and formulate an effective solution.

3.1 EXISTING SYSTEM

In a typical real estate business, any plyer looking to purchase or sell/rent a property will be unable to do so on their own in an effective, quick, and profitable manner without the assistance of any intermediaries in the form of brokerage websites or brokers. These middlemen suggest some sellers and buyers who meet their criteria. These middlemen then negotiate with the opposite party to determine the legal and transactional fees. Finally, they reach an agreement when the buyer accepts the deal and the transaction is completed. It should be mentioned that the buyer and seller must pay a fee to the intermediary for the transaction.

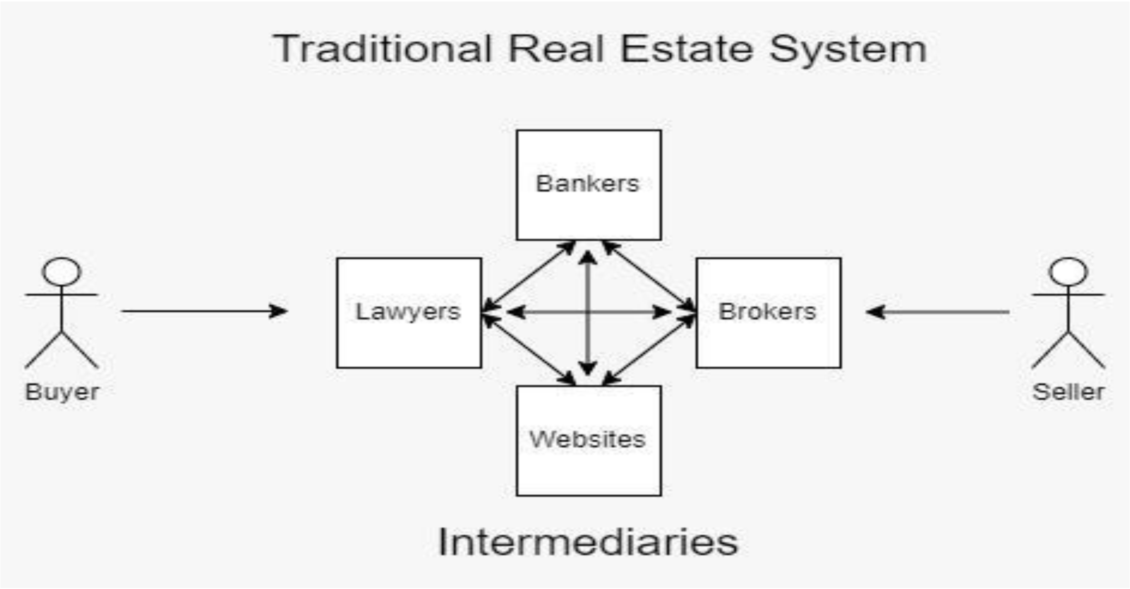


Fig 2.3 Traditional real estate system

3.2 PROBLEMS IN EXISTING SYSTEM

Regardless of whoever the intermediaries/middlemen be, they are always subject to the following limitations:

Legitimacy of the process:

Customers cannot trust a typical transaction technique since it is not legitimate enough. All of the market's middlemen or intermediates are in it to make a profit, and they don't care about the seller's or buyer's interests. They all have a shared purpose or business strategy that guides purchasers toward the most profitable purchases. These deals are not always beneficial to both buyers and sellers. These middlemen can also limit the possibilities available to the seller or prioritize the ones that are more profitable for them. As a result, there's a significant risk the buyers would lose out on the ideal property for them. The other major issue is that middlemen make misleading promises, which increases excitement among purchasers and ultimately results in a loss for them.

Genuineness of the current system:

These days, there are countless online scams. If a buyer has an outstanding website and a few con artists, they can easily be misled into acquiring a home that is not available or does not even exist. Not all purchasers are cautious enough to investigate them, and many do not have the finances or time to complete background investigations on these middlemen and the property. As a result, many legal suits have been brought, and a large amount of money has been lost in these processes; this is likely to recur frequently in the future unless people are made aware.

Middleman cost:

If the seller wants to sell the property and the buyer wants to acquire it, they must pay the middlemen or intermediaries a fee or commission for their services. Frequently, the costs charged by middlemen do not even justify the services provided. And, instead of getting better bargains and making the purchase process easy for both the buyer and seller, they wind up inflicting more harm than good. Furthermore, in many circumstances, the broker works with the seller to sell a home at a higher price than the market standard.

Time consuming procedures:

The buyer must decide and confirm the property they wish to purchase over the course of several months. Some intermediaries will demand a fee for assisting clients in finding the correct home, which is unnecessary. Even if they find the ideal property for them, the registration process will take a long time because of the inefficient workflow, which involves a lot of paperwork and personnel, and which may be avoided by adopting an efficient system.

Lack of availability:

A traditional real estate system involving middlemen or intermediaries may not always be available due to a lot of possible reasons. This can be of critical importance for someone who is wanting to buy or sell an asset in a very short time. This can be avoided by a system which can always work regardless of the availability of manpower or human support in the background which can make it a lot easier for both the buyers and sellers.

3.3 PROPOSED SYSTEM

Real estate and asset sales necessitate a large amount of paperwork to verify financial information and ownership before transferring deeds and titles to new owners. Using blockchain technology and artificial intelligence to record real estate transactions can provide a more safe, quick, and accessible method of validating and transferring ownership. This can decrease paperwork, eliminate the need for middlemen or commissions, and save money. There is no risk of downtime because it is decentralized and not reliant on a single server or entity.

The suggested system's aim is to create an efficient, safe, and user-centric platform that focuses on the trading of physical assets such as houses, land, and so on to buy/sell/rent/lease via a decentralized network that is peer-to-peer (P2P) and implemented utilizing blockchain technology. When a new transaction occurs, a new block is added to the blockchain. To validate a transaction, the network nodes must first validate it, and the nodes must reach an agreement using a consensus method.

By combining Artificial Intelligence features such as automatic price range prediction, chatbots, and recommendation engines with a Blockchain-based Decentralized Application for trading real assets, a better user experience and a variety of other benefits can be achieved. It allows users' data to be stored on a decentralized server rather than a centralized server controlled by any organization that can corrupt the data.

3.4 FEASIBILITY STUDY

Executive Summary:

The Decentralized application on Real Estate is a peer-to-peer based system which does not involve any intermediary or middleman for the transfer of properties between buyers and sellers. Blockchain technology makes it impossible for people to tamper the data and hence making it the most secure means of transaction involving no transaction fees or commission. The addition of Artificial Intelligence in the application provides reliable, related, and best fitting property for the buyer’s requirements and considering the benefits it can provide for them.

Market Feasibility:

The residential real estate market is expected to register a CAGR of more than 9% during the forecast period (2022-2027). The residential real estate (RRE) markets were impacted by the COVID-19 pandemic in several ways. On one hand, lockdowns and the increased use of remote working practices are expected to increase the demand for RRE, and accommodative monetary policies are anticipated to improve its affordability. The economic downturn and increases in unemployment are expected to weigh negatively on demand. Due to lockdowns, most of the construction activity and property transactions came to a halt during the pandemic. In 2021, as soon as the lockdown relaxation took place, the residential real estate market surged.

The residential real estate market is the cornerstone of the well-being of any economy. Shelter is considered a basic need for humans and lies at the base of the famous hierarchy of needs pyramid (Maslow). Therefore, it is understood that the manner in which the residential real estate market moves has a rippling effect on people around the world. The growth of population will also directly impact the demand of real estate properties in a positive way.

Residential properties such as apartments, bungalows, and villas are bought and sold on the market. The residential real estate market in emerging nations is mostly driven by urbanization. Major cities in emerging nations such as India, China, Brazil, Argentina, and South Africa are fast expanding and require additional housing to accommodate people migrating from various regions of the country.

Furthermore, government measures promoting affordable housing stimulate market expansion. Even the low mortgage interest rates are fueling the residential real estate market in countries like the United States, Canada, India, and Australia.

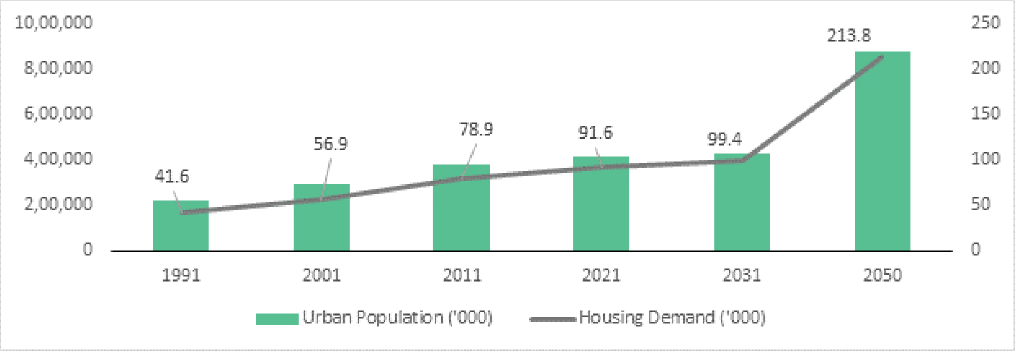
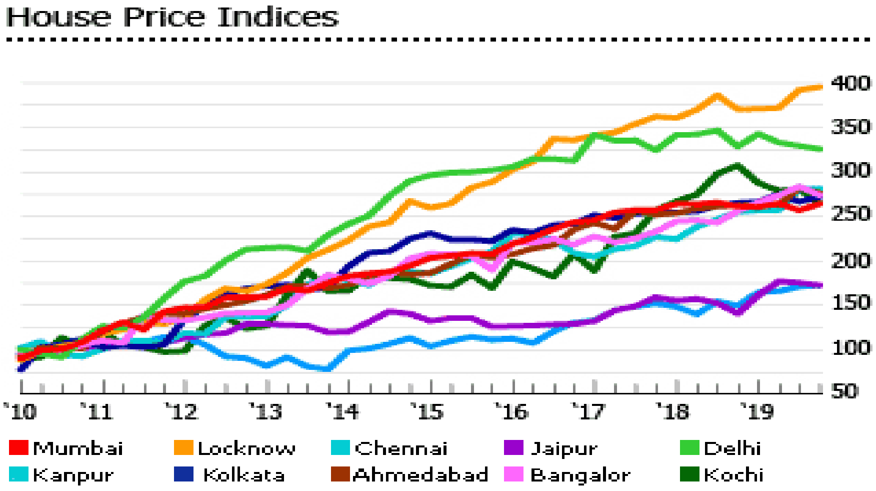


Fig 2.4 Predicted demand

The above image depicts the predicted increase in demand in the coming years due to the rise in urban population. This directly impacts the cost per square feet in every area which is depicted in the image below.



Source: Reserve Bank of India, Global Property Guide

Fig 2.5 House price indices

With real estate having a highly likely positive future, it is important to have a stable, secure, and service oriented system which can help the buyers and sellers. There cannot be a better technology other than Blockchain and Artificial Intelligence.

Technical Feasibility:

Blockchain technology promises to facilitate fast, secure, low-cost international payment processing services (and other transactions) through the use of encrypted distributed ledgers that provide trusted real-time verification of transactions without the need for intermediaries such as correspondent banks and clearing houses. Blockchain technology was initially used to support the digital currency Bitcoin, but is now being explored for a wide variety of applications that don’t involve bitcoin. The Blockchain technology has the potential to transform the property business. The potential shake-up would significantly speed up transactions and increase transparency. With smart contracts, every part of a lease or sale agreement is automated, and payments are received instantly – even outside of business hours. Smart contracts would also speed up pre-lease due diligence. Blockchain technology can help verify identities, making the background check process faster. Parties involved in a contract can access it with a personal digital key, arguably reducing the likelihood of fraud. In the industrial and logistics sector particularly, Blockchain's transparency could be of benefit to investors and occupiers alike. As corporations have expanded globally, supply chains have become longer and more complex—and the challenge of tracking inventory has become acute. A further attraction of blockchain technology is its use in recording land titles – a historically challenging area to access with most information still kept offline. The technology has the potential to dramatically cut the traditionally lengthy process of recording and transferring titles, with the added benefit of virtually bullet-proof transparency. And more and more governments are looking to apply the secure ledger to store and easily access historical title records.

AI technology is evolving by leaps and bounds! It has advanced to a point where mass-market utilization of AI has become highly affordable and feasible for all kinds of businesses and industries. A key area of application of AI in Real Estate is an in-depth analysis using multiple data-points. This typically analyses a property for its financial and operational performance and helps investors make the right decision. This analysis is performed with the massive amounts of data that the Real Estate industry generates. Digital transformation using Artificial intelligence in Real Estate is not futuristic anymore; it is the present. AI has revolutionized several industries, and there is no reason to exclude the Real Estate industry. AI helps in data collation from various sources and analyses market trends and provides analysis at a region, city, area, budget and property level. Along with this, details like the customer's past-preferences and past-interactions when fed into an AI-enabled system can offer better property recommendations to the customers. All these can help sales and marketing improve customer experience. Details like location, size, amenities etc., when combined with a self-learning AI system, can help predict the change in value of the property. This happens as the AI algorithms can make sense of the thousands of data points, and offer tangible insights to the realtor, from an investment standpoint.

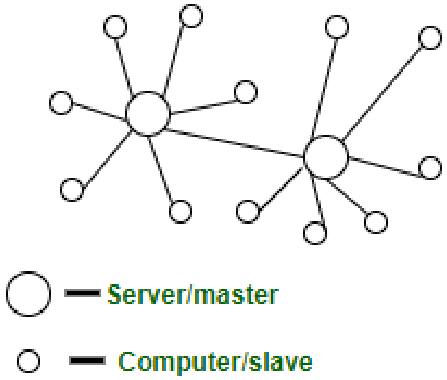


Fig 2.6 Traditional architecture

Conclusion:

There has been a change in customer demography, buying behavior, and the buying journey that has compelled developers to adopt efficient, digital mediums to engage with them. Driving sales velocity, customer engagement, cost reduction, and improving margins are the major focus areas for developers and Blockchain along with Artificial Intelligence can provide all those benefits. Overall, the study revealed that the use of Blockchain and Artificial Intelligence in Real Estate can be the turning point from now on. All these inferences lead to a perfect result of making real estate properties more accessible and facilitated to the users in a very secure and decentralized way. The traditional architecture cannot keep pace with the modern web 3.0 based developments. It will boom the sector; the data confirms that real estate is ready for change that can make it more accessible to the community.

3.5 HARDWARE REQUIREMENTS

A Desktop or a Laptop (also can be accessed through a smartphone) with a good internet connection and the below requirements: -

Processor	:	Intel Pentium Dual Core 2.00GHz or above
Hard disk	:	500 GB or above
RAM	:	8 GB (minimum)
Network	:	3 Mbps or above

3.6 SOFTWARE REQUIREMENTS

Operating system must run a stable build with the latest version of a browser supporting web 3.0 and pertaining to the below requirements: -

Domain	:	Blockchain and Artificial Intelligence
OS	:	Linux Distros / Windows 7 / macOS 10.15 (Catalina) or above
Browser	:	Latest Version of Chrome, Edge or Firefox
Platforms	:	Visual Studio Code Editor, Ethereum Platform, GitHub, DApp Test
Languages	:	Python, Javascript, Solidity, HTML, CSS, JSON, etc
Packages	:	All the required dependencies, packages and requirements
Tools	:	AI/ML frameworks (TensorFlow, Pytorch, Keras, Streamlit, etc), React NextJS, Ganache, Metamask, Ethereum, Auth0, CollectChat, Tally, Heroku, Vercel, Project Management (Clickup), Node, CLI

CHAPTER 4
SYSTEM DESIGN

The System is carefully designed after taking into consideration the overall user experience and security. It is designed in a way that even a naive user will be able to operate and do transactions in the application because of its simplicity. The following sections give a brief overview of the system.

4.1 ER DIAGRAM

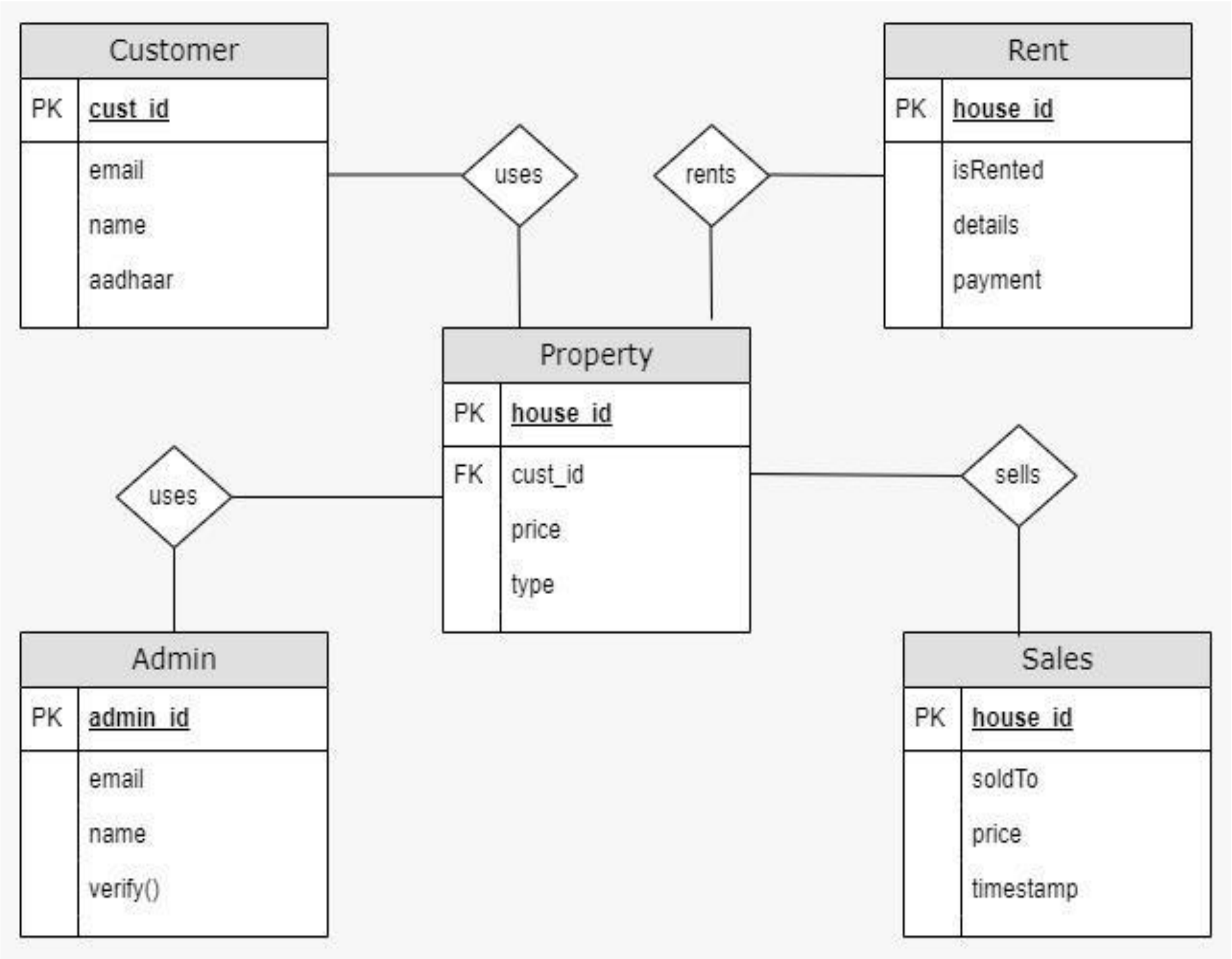


Fig 2.7 ER Diagram

4.2 DATA FLOW DIAGRAM

Level 0 DFD

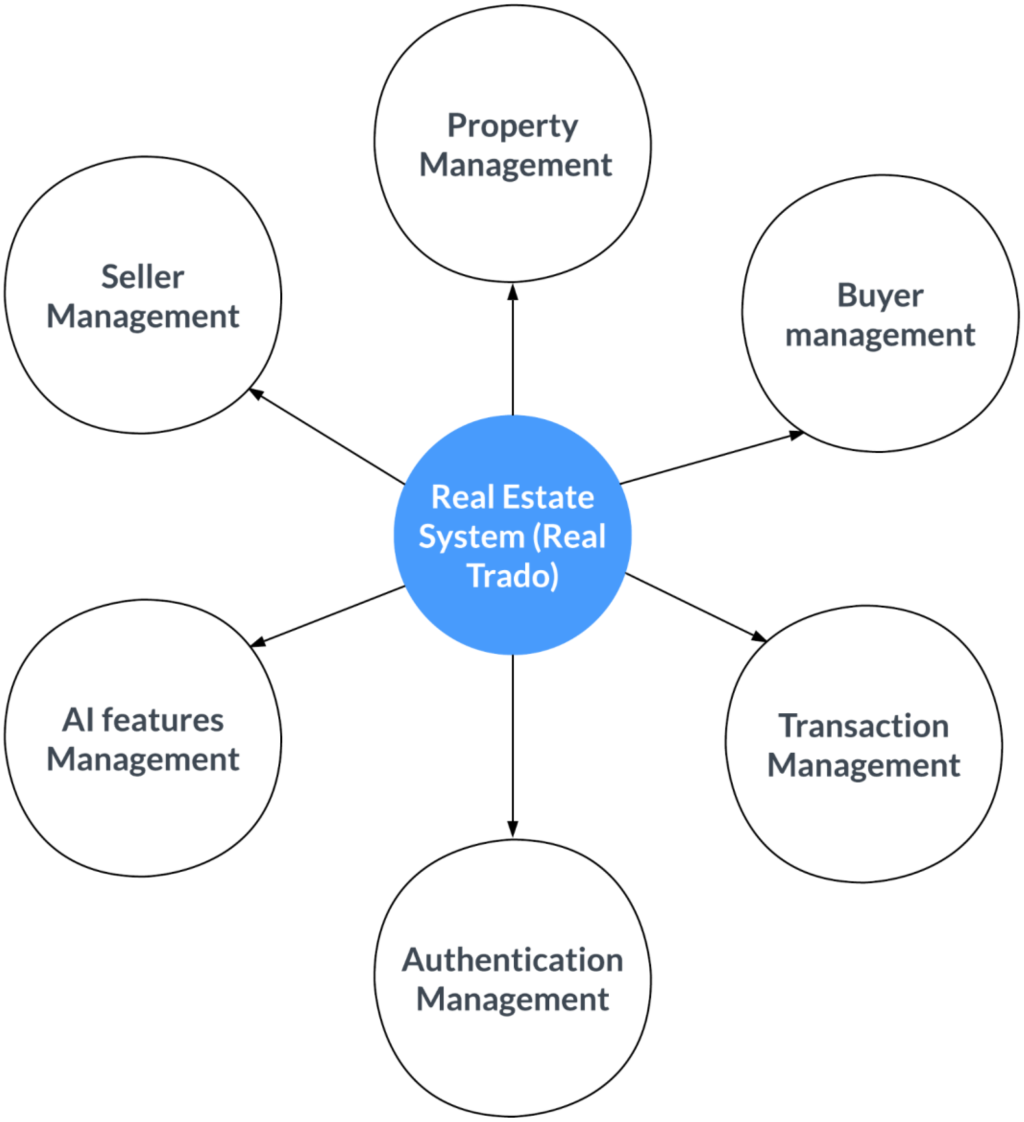


Fig 2.8 Level 0 DFD

Level 1 DFD

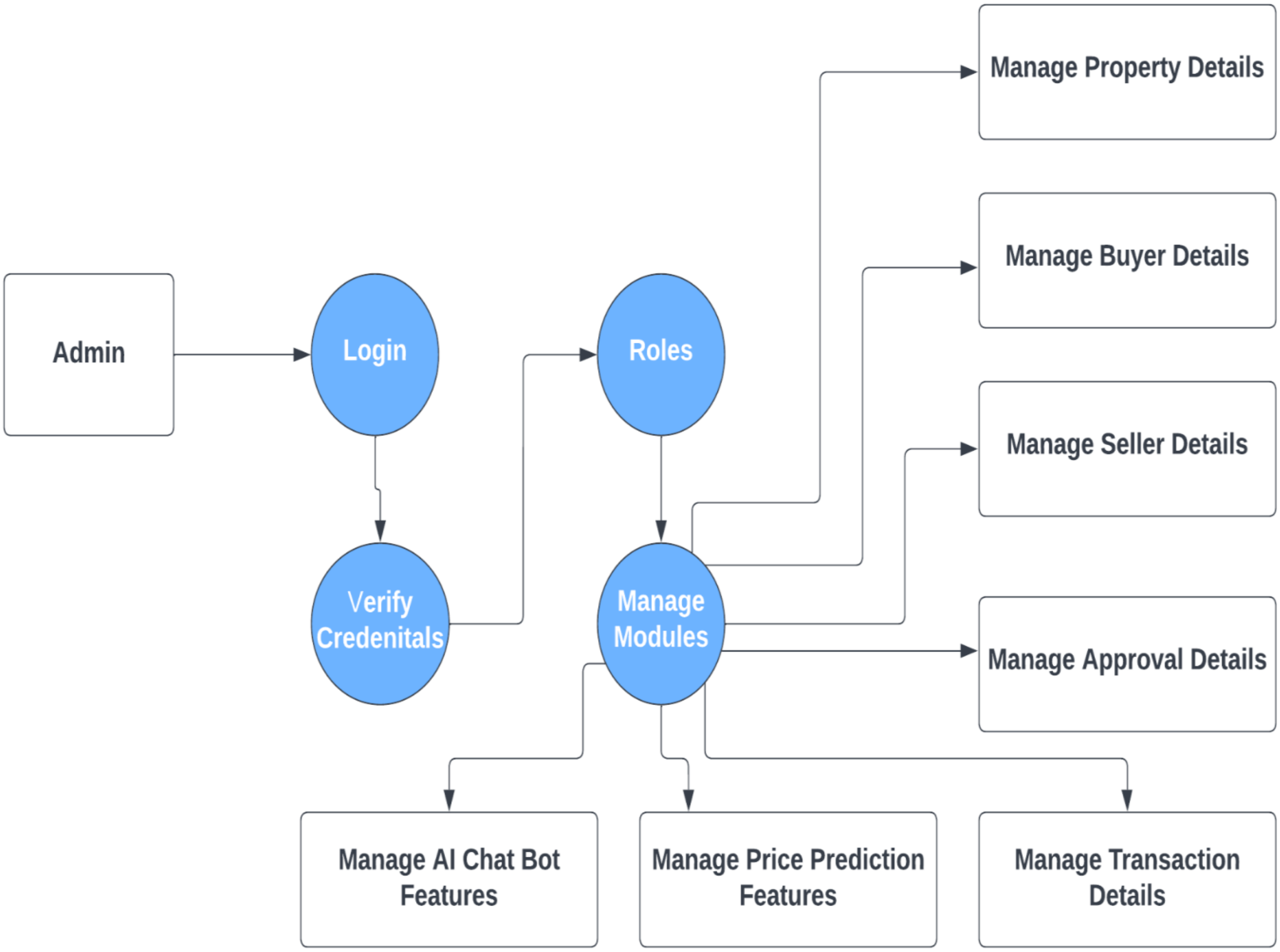


Fig 2.9 Level 1 DFD

4.3 UML DIAGRAM

Class Diagram

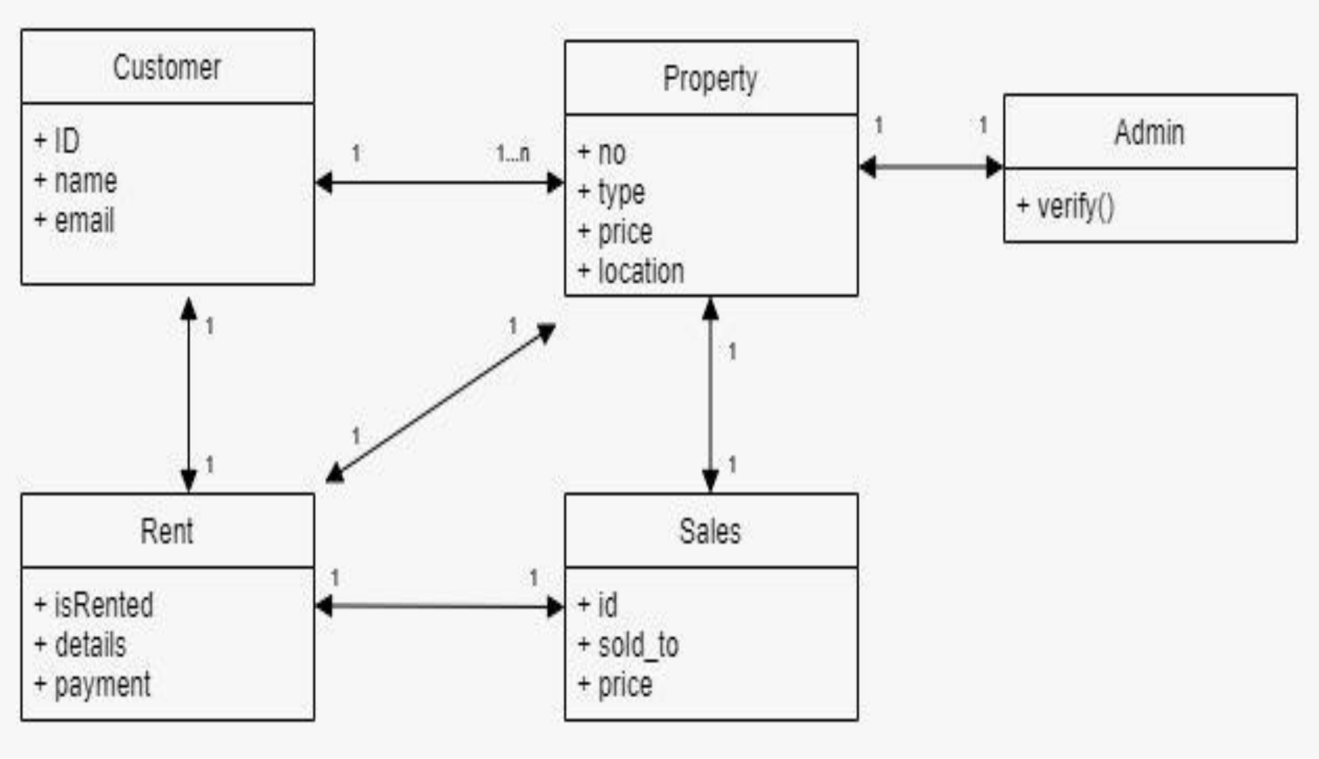


Fig 2.10 Class Diagram

CHAPTER 5
SYSTEM ARCHITECTURE

To use the system, buyers and sellers must join up for the platform, which contains multiple AI capabilities such as price range prediction, a recommendation engine, and a chat bot for enquiries. These AI functions aid the user's productivity flow.

Lessor and lessee will meet on the internet platform, where everyone will have access to verified property information such as ownership details, geographic location, chain of custody/ownership, and many other details. The marketplace or online platform can also be integrated with government systems, banks, or third-party verification businesses to assure the legitimacy of the information and identity verification. A distributed peer-to-peer system that allows the user to use a unified interface throughout the buying/selling/renting of real estate properties.

The Decentralized application (DApp), can be accessible via a variety of devices, assists in connecting users' accounts to the distributed network and enabling a secure environment for the procedures. The transaction takes place when the vendor and the buyer reach an agreement.

The system is finally completed by the integration of the various modules into one single unified platform. The blockchain network is secure and handles all the transactions, along with the AI features that help boost the productivity of the DApp for the users. Authentication is precise and all the buyer’s and seller’s details are promptly sent to the respective users through their mail. Thus, the newer blockchain real estate system with all these services and secure features is more efficient than the traditional real estate system as depicted in the below architecture diagram of the proposed system.

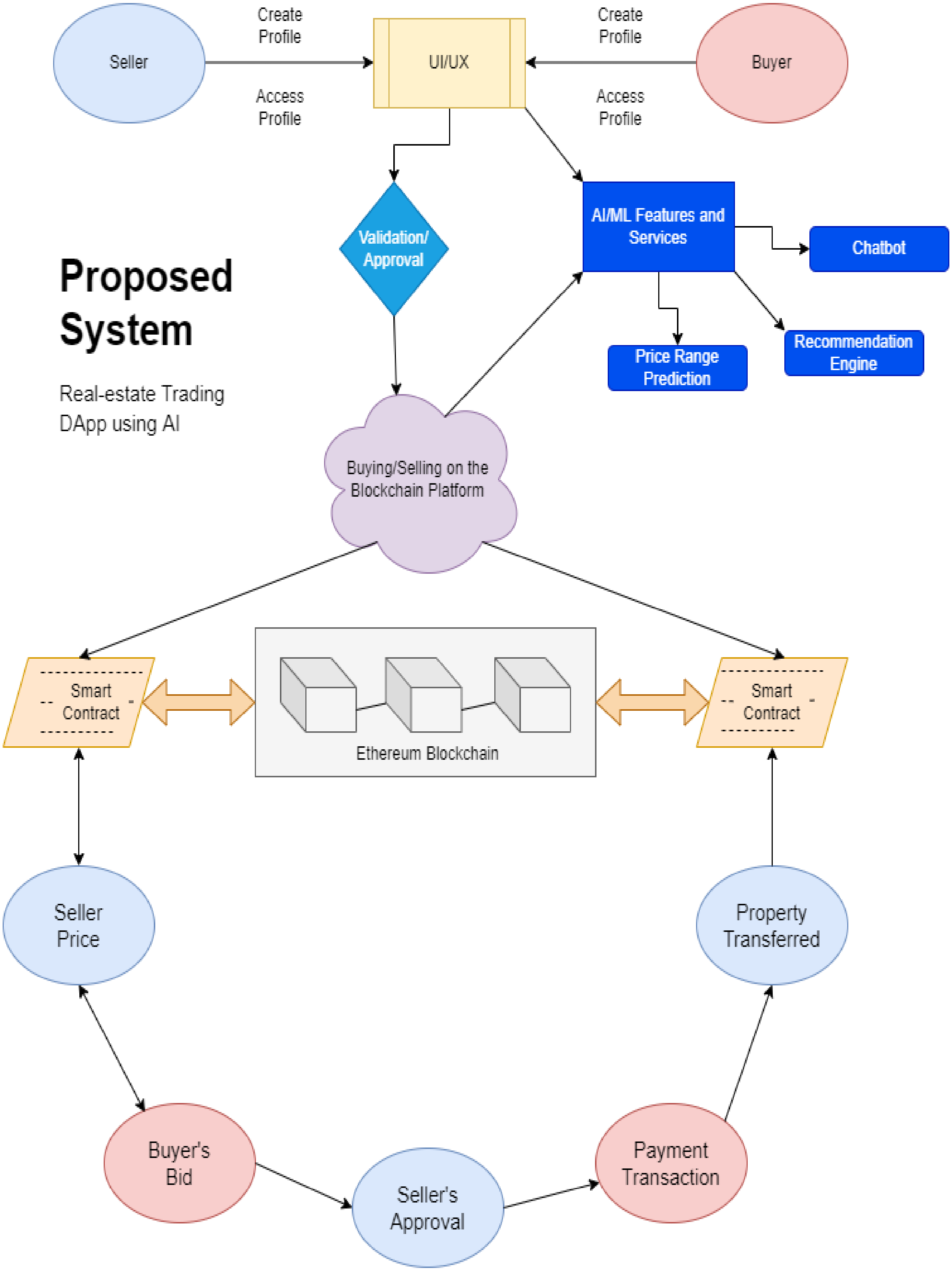


Fig 2.11 Architecture diagram

5.1 MODULE DESIGN SPECIFICATION

The different modules were built and integrated and their designs are described below.

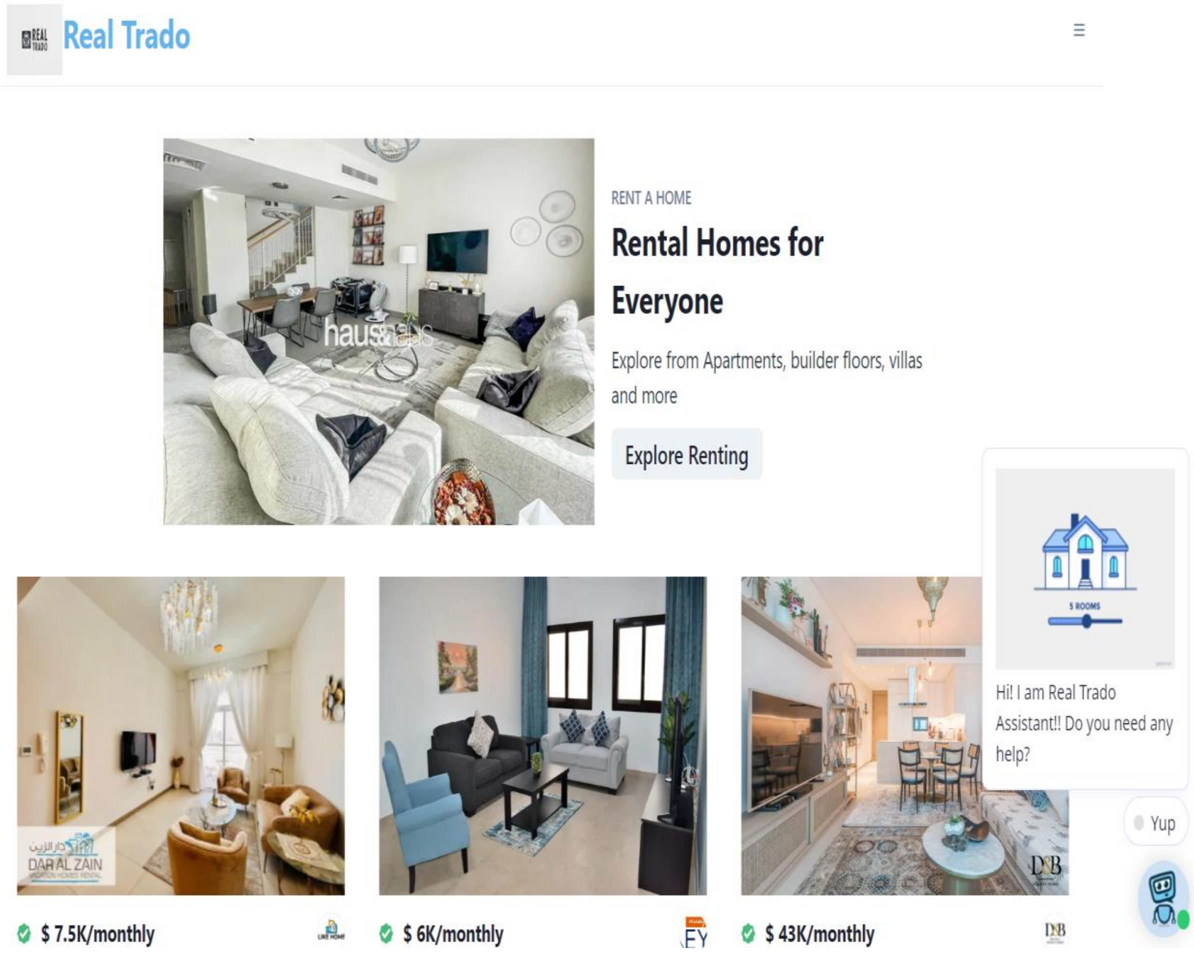
5.1.1 FRONTEND UI/UX DESIGN

The frontend is built as an intuitive and interactive user interface or user experience on the blockchain network with a combination of various AI features. All are integrated as a unified DApp platform. It is built using the React NextJS framework which is based on Node to create a wholesome and formulated structure to the overall application. NextJS is one of the most used frameworks and will be great for Web 3.0 applications. The site gets data from various seller listings posted into the site after admin approval also includes an API integration of the different properties available.

It is solely built using JavaScript and CSS which provides the users with a faster response, quick retrieval time and responsiveness. The buying, renting and selling of the properties is executed through the interface, especially the buying/renting. The selling built with tally gets the user inputs, Aadhaar number of user and information of the properties which are later listed in the DApp after approval by the admins. It also is linked with the Auth0 for authentication of the login and signup

into the platform for identification, security and personalization.

The frontend is integrated with the private blockchain network and payment gateway which tracks the transactions based on the ledgers of a decentralized method. Each property is linked to the corresponding recipient id and the amount will be transferred through the blockchain. The frontend UI/UX also holds the AI features such as chatbot and recommendation and price comparison engine.



Rental Homes for Everyone

Explore from Apartments, builder floors, villas and more

Hi! I am Real Trado Assistant!! Do you need any help?

Fig 2.12 UI/UX design

5.1.2 BLOCKCHAIN NETWORK AND PAYMENT GATEWAY

Once the buyer has picked their choice of property, they can confirm their interest by doing the payment through the DApp. The payment (transactions) in this DApp is done through the Blockchain and ideally a person needs to have money in the wallet which can be in the form of any crypto currency or in their native currency (₹ in case of India, for this project it is $). In this DApp, we have used a local Blockchain network using Ganache and have minted DAI tokens which are stable coins with a standard value and converted them to dollars to show the working of the Blockchain. DAI (formerly SAI) is a stable coin cryptocurrency on the Ethereum blockchain which aims to keep its value as close to one United States Dollar (USD). The transaction will be completed instantly no matter how much money is involved. Also, there are no hidden charges. After the person does the payment through the application, their transaction can be seen in the application and the transaction details are stored in the Blockchain. The transfer of ownership or the renting process gets completed here.

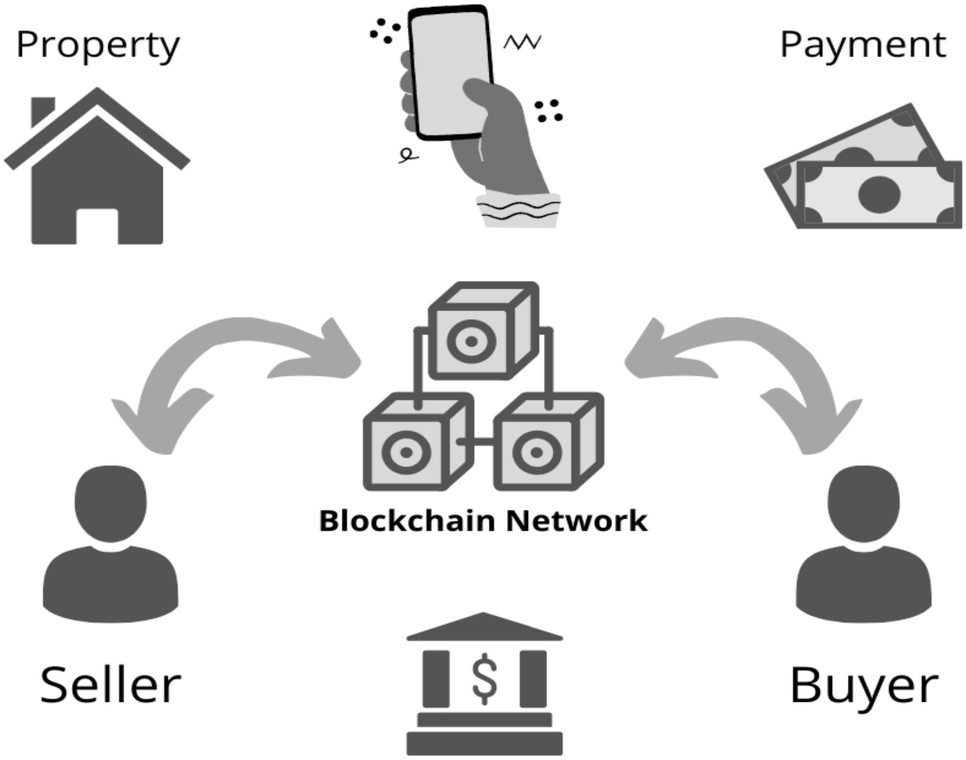


Fig 2.13 Blockchain network payment gateway

5.1.3 CONVERSATIONAL AI (CHATBOT)

The conversational AI or chatbot plays an indisputable role in the modern web applications and the Web 3.0 based DApp will continue the trend to provide support and query responses to various queries of the users. This AI feature not only helps the users to interact with the platform easily but also gives the users and the admins the freedom of answering the questions and getting various responses. It is an excellent support tool, the chatbot is built using the AI collect chat API and is trained to give a variety of responses from support to booking calls to the admins. The user interface and the chatbot is integrated, called as the “Real Trado Assistant” which enables the functions of the unified platform.

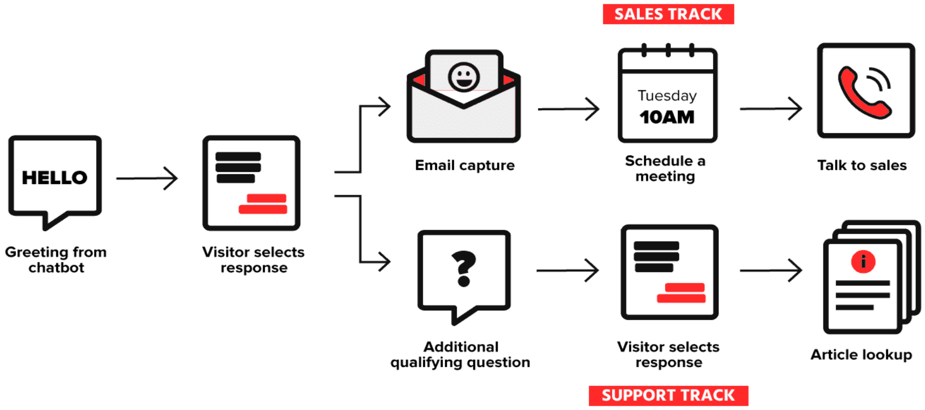


Fig 2.14 AI Chatbot Process Flow

The chatbot analyzes the user’s request and then identifies the different entities and intents of the train set of instructions and data. It composes coherent responses for the particular variety of questions by the user to give the prompt and apt response.

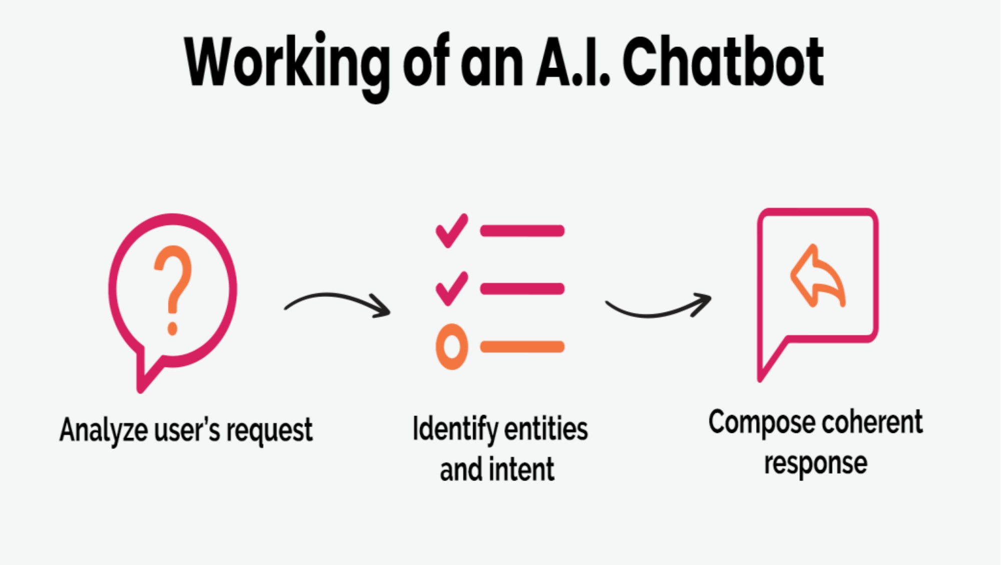

Working of an A.I. Chatbot
Analyze user's request
Identify entities and intent
Compose coherent response

Fig 2.15 AI Chatbot Working

5.1.4 AI RECOMMENDATION AND PRICE COMPARISON ENGINE

AI Recommendation and Price comparison combines two of the most powerful machine learning models where the train data uses nearly 5000 different data of the real estate properties dataset. The properties under the listings and API are used as the test data and the results are provided with high accuracy.

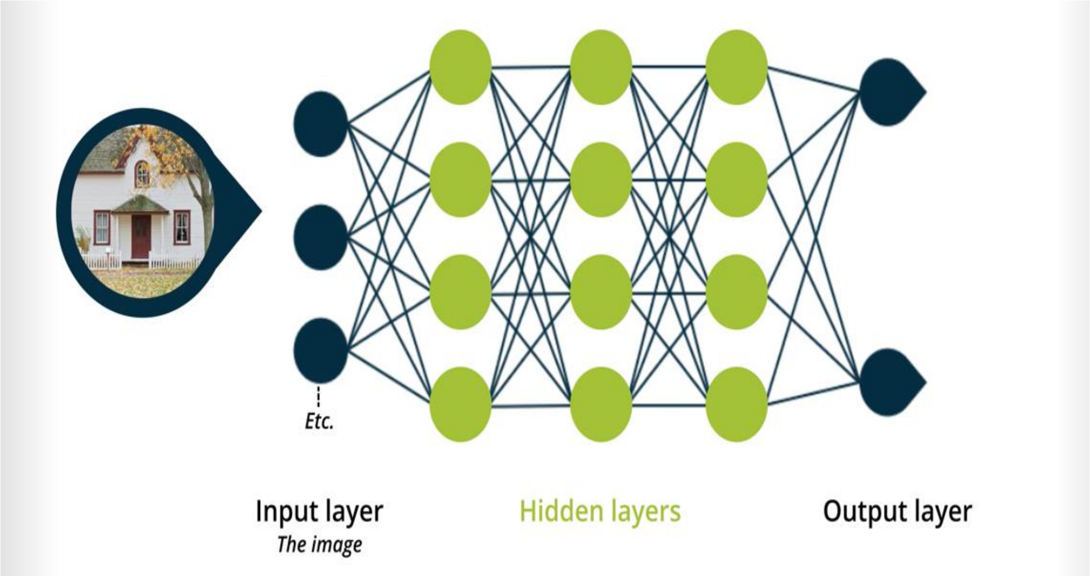


Fig 2.16 Price Comparison neural network

The code is written in python and streamlit framework along with other machine learning frameworks are used to depict and compare the prices of the various properties and also gives the recommendation and research for the users all based on macro reports and recommendation reports to choose the best suitable property. It is integrated as a feature of the DApp called as Real Trado Recommend. It also shows the various recommendations on the home page of the DApp.

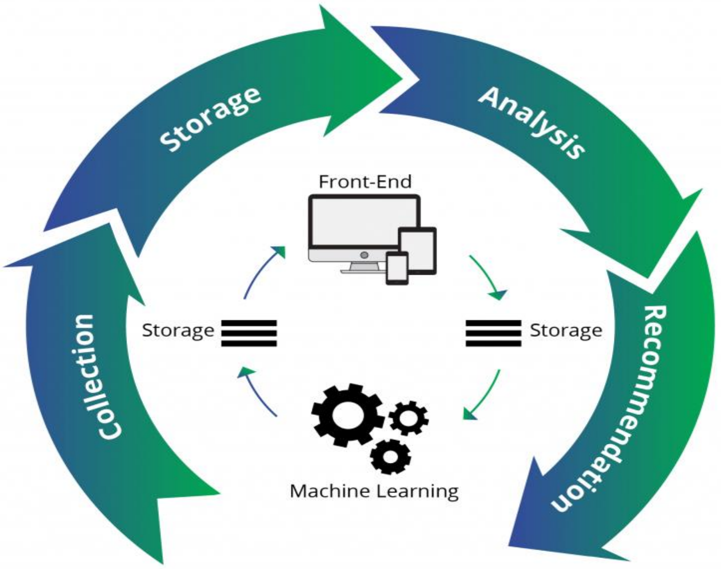


Fig 2.17 House Prediction process flow

5.2 ALGORITHMS

The below mentioned are the algorithms for blockchain and artificial intelligence that are used in the development of the project: -

5.2.1 BLOCKCHAIN ALGORITHMS

Proof of Work (PoW)

The Proof of Work algorithm is the most widely used algorithm. This algorithm is used by crypto currencies such as Bitcoin and Ethereum, each one with its own differences. PoW algorithm is used to confirm transactions and produce new blocks in the blockchain. Using Proof of Work, miners compete against each other to complete the transaction on the network and get rewarded. In this algorithm, main

work is to solve the mathematical puzzle. It is an issue that requires a lot of computational power to solve. Miners solve the puzzle then confirm the transaction and form the new block. When other nodes confirm that the transaction is valid then only block is added to the chain permanently. The problem should not be too complex to solve, if it is like that, block generation takes a lot of time. But in other scenario if problem is too easy, it is prone to DOS attack and spam. The solution needs to be easily checked by other nodes otherwise not all nodes are capable of analyzing that calculation is correct. Thus, it will have to trust other nodes and it violates one of the important features of blockchain - transparency. Complexity of puzzle is depending on the number of users, the current power and the network load. The hash value of each block contains the previous block's hash value which increases the security. The genesis block is an exception as it has no parent block, so its hash value is completely zeros. Bitcoin is the foundation of this kind of consensus. The puzzle called as Hash cash. The Proof of Work algorithm allows changing the complexity of a puzzle based on the total power of the network. The average time of the formation of any block is approx. 10 minutes. The main disadvantages of this algorithm are huge expenditure, uselessness of computations and 51 percent attack.

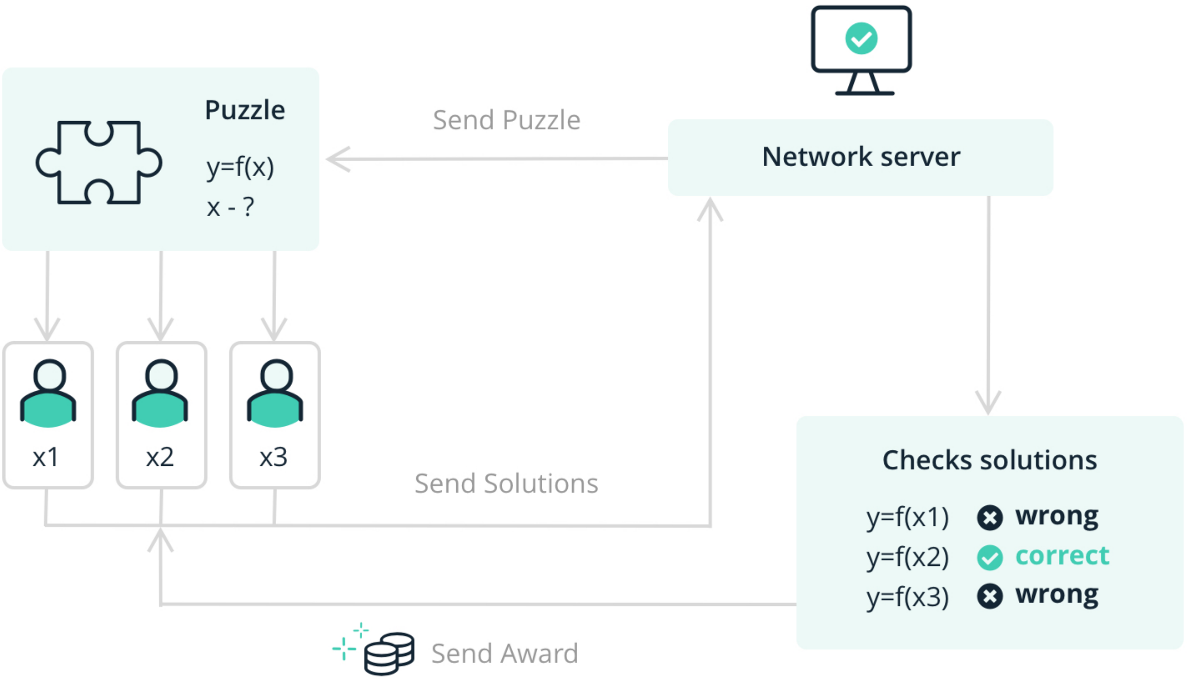
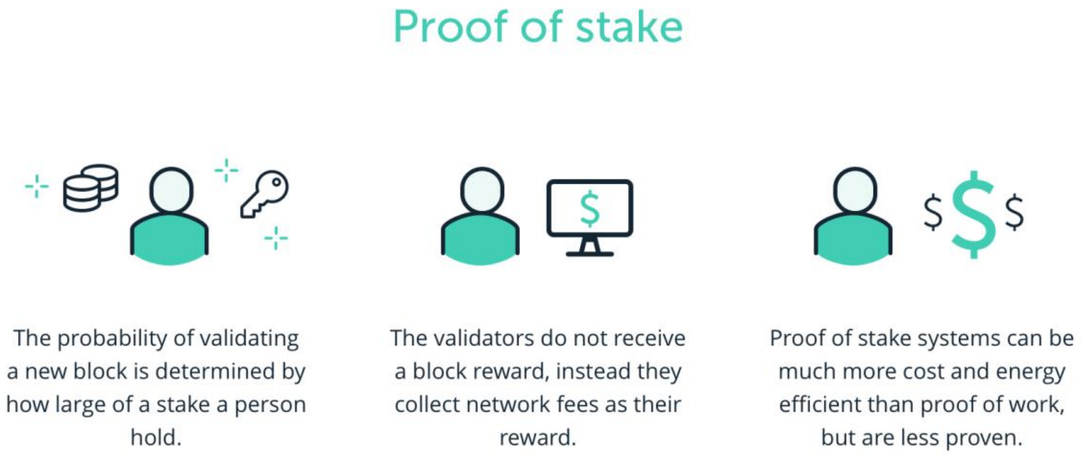


Fig 2.18 Proof of Work

Proof of Stake (PoS)

Proof of Stake has been mentioned in the first bitcoin project, but it was not used in bitcoin because of its robustness and other reasons. It is different from the Proof of Work algorithm in which hashing algorithm is used to validate the transaction. Proof of Stake is most used as the replacement of PoW in Peercoin. Traditionally in this algorithm, selection of miners is based on the account balance i.e., higher the balance in account, higher the chance to become the miner. So, the richest person has the probability to become a permanent miner as it has high balance in his account and that leads to being unfair to other persons. So, this process leads to the centralization that is why several other processes of selection has been devised. In peer coin crypto currencies, proof of Stake combines the randomization with the concept of “coin age”.

Coins that have been unspent for at least 30 days, being competing for the next block. To signing the next block, the older and largest set of coins have greater possibility. Then once the Stake of coin has been selected to sign a block, they must start with zero ‘coin age’ and then wait at least 30 more days before signing another block. So, this process secures the network and produce new coin without much computational power. Proof of Stake is more efficient than Proof of Work.



Proof of stake

The probability of validating a new block is determined by how large of a stake a person hold.

The validators do not receive a block reward, instead they collect network fees as their reward.

Proof of stake systems can be much more cost and energy efficient than proof of work, but are less proven.

Fig 2.19 Proof of Stake

5.2.2 AI ALGORITHMS

Recommendation Engine

Content-Based systems recommends items to the customer similar to previous version of the highly sought or history by the customer. It uses the features and properties of the item. From these properties, it can calculate the similarity between the items.

In a content-based recommendation system, first, we need to create a profile for each item, which represents the properties of those items. From the user profiles are inferred for a particular user. We use these user profiles to recommend the items to the users from the catalog and create the algorithm.

To create an item profile, firstly, we need to perform the TF-IDF vectorizer, here TF (term frequency) of a word is the number of times it appears in a document and The IDF (inverse document frequency) of a word or search is the measure of how significant that term is in the whole corpus. These can be calculated by the following formula:-

The term-frequency can be calculated by:

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}}$$

where fij is the frequency of term(feature) i in document(item) j.

The inverse-document frequency can be calculated with:

$$IDF_i = log_e \frac{N}{n_i}$$

where, ni number of documents that mention term i. N is the total number of docs.

Therefore, the total formula is:

$$TF - IDFscore(w_{ij}) = TF_{ij} * IDF_i$$

Price Comparison Engine

- **Reinforcement learning model:**

Reinforcement learning (RL) is a goal-directed machine learning model which aims to achieve the highest rewards by learning from environment data. An RL dynamic pricing model explores data about customers' demand, taking into account seasonality, competitors' prices, and the uncertainty of the market, with high revenue being the final goal.

- **Decision tree model:**

Decision trees are classification machine learning models that output a tree-like model of decisions and their possible consequences, including the possibility of a certain outcome, resource costs, and utility. Decision tree dynamic pricing algorithms help businesses understand which parameters have the most effect on the prices, and which of these price ranges predicts the highest revenues, and using this information, the algorithm predicts the best price range for each product.

CHAPTER 6
SYSTEM IMPLEMENTATION

To create a well-structured and an intuitive DApp requires the use of precise coding and the use of various tools, frameworks, software, and resources. By using a module-based approach to build each part of the system and integrating them makes a solidified outcome to the platform.

The frontend consists of the buy and rent UI/UX, search, sort, authentication, selling interface along with the integration of the blockchain network and payment gateway, AI features like chatbot, recommendation engine and make it a unified platform. Few of the sample coding for each are below.

6.1 FRONTEND CODING

GitHub Repository:

https://github.com/DarinJoshua-dev/Real-Trado (or)
https://github.com/GladwinJosephSolomon/Real-Trado (or)
https://github.com/JoelNithishKumar/Real-Trado

App.js :

```
import "../styles/globals.css";
import Router from "next/router";
import Head from "next/head";
import NProgress from "nprogress";
import { ChakraProvider } from "@chakra-ui/react";
import Script from "next/script";
import React from "react";
import { UserProvider } from "@auth0/nextjs-auth0";
import Layout from "../components/Layout";

function MyApp({ Component, pageProps }) {
  NProgress.configure({ showSpinner: false });
  Router.events.on("routeChangeStart", () => {
    NProgress.start();
  });

  Router.events.on("routeChangeComplete", () => {
    NProgress.done();
  });

  return (
    <>
      <Head>
        <link
          rel="stylesheet"
          href="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.css"
          integrity="sha512-
42kB9yDlYiCEfx2xVwq0q7hT4uf26FUgSIZBK8uiaEnTdShXjwr8Ip1V4xGJMg3mHk
Ut9nNuTDxunHF0/EgxLQ=="
          crossOrigin="anonymous"
          referrerPolicy="no-referrer"
        />
        <script
          type="text/javascript"
          dangerouslySetInnerHTML={{
            __html: `
                      (function(w, d) { w.CollectId = "627d53fcfaa7943a0bbb2472"; var h =
d.head || d.getElementsByTagName("head")[0]; var s = d.createElement("script");
s.setAttribute("type", "text/javascript"); s.async=true; s.setAttribute("src",
"https://collectcdn.com/launcher.js"); h.appendChild(s); })(window, document);,
          }}
        />
      </Head>
      <ChakraProvider>
        <Layout>
          <UserProvider>
            <Component {...pageProps} />
          </UserProvider>
        </Layout>
      </ChakraProvider>
    </>
  );
}
export default MyApp;
```

Index.js:

```
import Link from "next/link";
import Image from "next/image";
import { Flex, Box, Text, Button } from "@chakra-ui/react";
import Property from "../components/Property";
import { baseUrl, fetchApi } from '../utils/fetchApi'

const Banner = ({
  purpose,
  title1,
  title2,
  desc1,
  desc2,
  buttonText,
  linkName,
  imageUrl,
}) => (
  <Flex flexWrap="wrap" justifyContent="center" alignItems="center" m="10">
    <Image src={imageUrl} width={500} height={300} />
    <Box p="5">
      <Text color="gray.500" fontSize="sm" fontWeight="medium">
        {purpose}
      </Text>
      <Text fontSize="3xl" fontWeight="bold">
        {title1}
        <br />
        {title2}
      </Text>
      <Text fontSize="lg" paddingTop="3" paddingBottom="3" color="gray.700">
        {desc1}
        <br />
        {desc2}
      </Text>
      <Button fontSize="xl">
        <Link href={linkName}>
          <a>{buttonText}</a>
        </Link>
      </Button>
    </Box>
  </Flex>
);

export default function Home({ propertiesForSale, propertiesForRent }) {
  console.log(propertiesForSale, propertiesForRent);

  return (
    <Box>
      <Banner
        purpose="RENT A HOME"
        title1="Rental Homes for"
        title2="Everyone"
        desc1=" Explore from Apartments, builder floors, villas"
        desc2="and more"
        buttonText="Explore Renting"
        linkName="/search?purpose=for-rent"
        imageUrl="https://bayut-production.s3.eu-central-
1.amazonaws.com/image/145426814/33973352624c48628e41f2ec460faba4"
      />
      <Flex flexWrap="wrap">
        {propertiesForRent.map((property) => <Property property={property}
key={property.id} />)}
      </Flex>
      <Banner
        purpose="BUY A HOME"
        title1=" Find, Buy & Own Your"
        title2="Dream Home"
        desc1=" Explore from Apartments, land, builder floors,"
        desc2=" villas and more"
        buttonText="Explore Buying"
        linkName="/search?purpose=for-sale"
        imageUrl="https://bayut-production.s3.eu-central-
1.amazonaws.com/image/110993385/6a070e8e1bae4f7d8c1429bc303d2008"
      />
      <Flex flexWrap="wrap">
        {propertiesForSale.map((property) => <Property property={property}
key={property.id} />)}
      </Flex>
    </Box>
  );
}

// fetchApi
// Nextjs default function
export async function getStaticProps() {
  const propertyForSale = await fetchApi(`${baseUrl}/properties/list?locationExternalIDs=5002&purpose=for-sale&hitsPerPage=6`)
  const propertyForRent = await fetchApi(`${baseUrl}/properties/list?locationExternalIDs=5002&purpose=for-rent&hitsPerPage=6`)

  return {
    props: {
      propertiesForSale: propertyForSale?.hits,
      propertiesForRent: propertyForRent?.hits,
    }
  }
}
```

[Id].js:

```
import { Box, Flex, Text, Spacer, Avatar } from "@chakra-ui/react";
import { FaBed, FaBath } from "react-icons/fa";
import { BsGridFill } from "react-icons/bs";
import { GoVerified } from "react-icons/go";
import { baseUrl, fetchApi } from "../../utils/fetchApi";
import ImageScrollbar from "../../components/ImageScrollbar";
import millify from "millify";
import Link from "next/link";

const PropertyDetails = ({
  propertyDetails: {
    price,
    rentFrequency,
    rooms,
    title,
    baths,
    area,
    agency,
    isVerified,
    description,
    type,
    purpose,
    furnishingStatus,
    amenities,
    photos,
  },
}) => (
  <Box maxWidth="1000px" margin="auto" p="4">
    {photos && <ImageScrollbar data={photos} />}
    <Box w="full" p="6">
      <Flex paddingTop="2" alignItems="center" justifyContent="space-between">
        <Flex alignItems="center">
          <Box paddingRight="3" color="green.400">
            {isVerified && <GoVerified />}
          </Box>
          <Text fontWeight="bold" fontSize="lg">
            $ {millify(price)}
            {rentFrequency && `/${rentFrequency}`}
          </Text>
        </Flex>
        <Box>
          <Avatar size="sm" src={agency?.logo?.url} />
        </Box>
      </Flex>
      {/* For Rooms */}
      <Flex
        alignItems="center"
        p="1"
        justifyContent="space-between"
        w="250px"
        color="blue.400"
      >
        {rooms} <FaBed /> | {baths} <FaBath /> | {millify(area)} sqft{" "}
        <BsGridFill />
        <Link href="https://real-trado-blockchain-pay.herokuapp.com/" passHref>
          <a className="button">Buy/Rent</a>
        </Link>
      </Flex>
      {/* For title and description */}
      <Box marginTop="2">
        <Text fontSize="lg" marginBottom="2" fontWeight="bold">
          {title}
        </Text>
        <Text lineHeight="2" color="gray.600">
          {description}
        </Text>
      </Box>
      <Flex
        flexWrap="wrap"
        textTransform="uppercase"
        justifyContent="space-between"
      >
        <Flex
          justifyContent="space-between"
          w="400px"
          borderBottom="1px"
          borderColor="gray.100"
          p="3"
        >
          <Text>Type</Text>
          <Text fontWeight="bold">{type}</Text>
        </Flex>
        <Flex
          justifyContent="space-between"
          w="400px"
          borderBottom="1px"
          borderColor="gray.100"
          p="3"
        >
          <Text>Purpose</Text>
          <Text fontWeight="bold">{purpose}</Text>
        </Flex>
        {furnishingStatus && (
          <Flex
            justifyContent="space-between"
            w="400px"
            borderBottom="1px"
            borderColor="gray.100"
            p="3"
          >
            <Text>Furnishing Status</Text>
            <Text fontWeight="bold">{furnishingStatus}</Text>
          </Flex>
        )}
      </Flex>
      {/* Amenity */}
      <Box>
```

```
        {amenities.length && (
          <Text fontSize="2xl" fontWeight="black" marginTop="5">
            Amenities
          </Text>
        )}
        <Flex flexWrap="wrap">
          {amenities.map((item) =>
            item.amenities.map((amenity) => (
              <Text
                fontWeight="bold"
                color="blue.400"
                fontSize="l"
                p="2"
                bg="gray.200"
                m="1"
                borderRadius="5"
                key={amenity.text}
              >
                {amenity.text}
              </Text>
            ))
          )}
        </Flex>
      </Box>
    </Box>
  </Box>
);

export default PropertyDetails;

export async function getServerSideProps({ params: { id } }) {
  const data = await fetchApi(`${baseUrl}/properties/detail?externalID=${id}`);

  return {
    props: {
      propertyDetails: data,
    },
  };
}
```

6.2 BLOCKCHAIN CODING

GitHub Repository:

https://github.com/GladwinJosephSolomon/blockchain-network-payment (or)
https://github.com/DarinJoshua-dev/Blockchain-Network-Pay (or)
https://github.com/JoelNithishKumar/Blockchain-Payment-system

App.js:

```
import React, { Component } from "react";
import RealTrado from "../RealTrado.png";
import "./App.css";
import Web3 from "web3";
import DaiTokenMock from "../abis/DaiTokenMock.json";

class App extends Component {
  async componentWillMount() {
    await this.loadWeb3();
    await this.loadBlockchainData();
  }

  async loadWeb3() {
    if (window.ethereum) {
      window.web3 = new Web3(window.ethereum);
      await window.ethereum.enable();
    } else if (window.web3) {
      window.web3 = new Web3(window.web3.currentProvider);
    } else {
      window.alert(
        "Non-Ethereum browser detected. You should consider trying MetaMask!"
      );
    }
  }

  async loadBlockchainData() {
    const web3 = window.web3;
    const accounts = await web3.eth.getAccounts();
    this.setState({ account: accounts[0] });
    const daiTokenAddress = "0x7e81Faa7875113F7dC3E25d18Ac0697Df01Dd316";
// Replace DAI Address Here
    const daiTokenMock = new web3.eth.Contract(
      DaiTokenMock.abi,
      daiTokenAddress
    );
    this.setState({ daiTokenMock: daiTokenMock });
    const balance = await daiTokenMock.methods
      .balanceOf(this.state.account)
      .call();
    this.setState({ balance: web3.utils.fromWei(balance.toString(), "Ether") });
    const transactions = await daiTokenMock.getPastEvents("Transfer", {
      fromBlock: 0,
      toBlock: "latest",
      filter: { from: this.state.account },
    });
    this.setState({ transactions: transactions });
    console.log(transactions);
  }

  transfer(recipient, amount) {
    this.state.daiTokenMock.methods
      .transfer(recipient, amount)
      .send({ from: this.state.account });
  }

  constructor(props) {
    super(props);
    this.state = {
      account: "",
      daiTokenMock: null,
      balance: 0,
      transactions: [],
    };

    this.transfer = this.transfer.bind(this);
  }

  render() {
    return (
      <div>
        <nav className="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0
shadow">
          <a
            href="https://real-trado-dapp.vercel.app/"
            className="navbar-brand col-sm-3 col-md-2 mr-0">
            Real Trado
          </a>
        </nav>
        <div className="container-fluid mt-5">
          <div className="row">
            <main role="main" className="col-lg-12 d-flex text-center">
              <div
                className="content mr-auto ml-auto"
                style={{ width: "500px" }}
              >
                <a href="https://real-trado-dapp.vercel.app/">
                  <img src={RealTrado} width="150" />
                </a>
                <h1> $ {this.state.balance} </h1>
                <form
                  onSubmit={(event) => {
                    event.preventDefault();
                    const recipient = this.recipient.value;
                    const amount = window.web3.utils.toWei(
                      this.amount.value,
                      "Ether"
                    );
                    this.transfer(recipient, amount);
                  }}
                >
                  <div className="form-group mr-sm-2">
                    <input
                      id="recipient"
                      type="text"
                      ref={(input) => {
                        this.recipient = input;
                      }}
                      className="form-control"
                      placeholder="Recipient Address"
                      required
                    />
                  </div>
                  <div className="form-group mr-sm-2">
                    <input
                      id="amount"
                      type="text"
```

```
   ref={(input) => {
    this.amount = input;
   }}
   className="form-control"
   placeholder="Amount"
   required
  />
 </div>
 <button type="submit" className="btn btn-primary btn-block">
  Send
 </button>
</form>
<table className="table">
 <thead>
  <tr>
   <th scope="col">Recipient</th>
   <th scope="col">value</th>
  </tr>
 </thead>
 <tbody>
  {this.state.transactions.map((tx, key) => {
   return (
    <tr key={key}>
     <td>{tx.returnValues.to}</td>
     <td>
      {window.web3.utils.fromWei(
       tx.returnValues.value.toString(),
       "Ether"
      )}
     </td>
    </tr>
   );
  })}
 </tbody>
</table>
<button
 type="submit"
 className="btn btn-primary btn-block"
 onClick={(event) => (window.location.href = "success.html")}
>
 Finish Now
</button>
```

```
              </div>
            </main>
          </div>
        </div>
      </div>
    );
  }
}

export default App;
```

1_initial_migration.js:

```
const Migrations = artifacts.require("Migrations");
const DaiTokenMock = artifacts.require("DaiTokenMock");

module.exports = async function(deployer) {
  await deployer.deploy(Migrations);
  await deployer.deploy(DaiTokenMock);
  const tokenMock = await DaiTokenMock.deployed()
  // Mint 1,000000 Dai Tokens for the deployer
  await tokenMock.mint(
    '0x1e578566044D50954a10a295229a66176678d098',
    '1000000000000000000000000'
  )
};
```

DaiTokenMock.sol:

```
pragma solidity ^0.5.0;

import "@openzeppelin/contracts/token/ERC20/ERC20Mintable.sol";

contract DaiTokenMock is ERC20Mintable {
  string  public name;
  string  public symbol;
  uint256 public decimals;

  constructor() public {
    name = "Dai Stablecoin (DAI)";
    symbol = "DAI";
    decimals = 18; }}
```

6.3 ARTIFICIAL INTELLIGENCE CODING

GitHub Repository: https://github.com/DarinJoshua-dev/RealEstate_Price-Recommendation-Engine

Houses_recommendations.ipynb:

```
import pandas as pd
import numpy as np
import warnings
import seaborn as sns
import plotly.express as px
import scipy.stats as _stats
import datetime as dt
import itertools
import folium
import geopandas

from matplotlib import pyplot as plt
from IPython.display import display, HTML, Image
from helper_functions import *
from scipy import stats
from folium.plugins import MarkerClusteg

def jupyter_settings():
    %matplotlib inline

    plt.style.use( 'bmh' )
    plt.rcParams['figure.figsize'] = [20, 12]
    plt.rcParams['font.size'] = 24

    display( HTML( '<style>.container { width:100% !important; }</style>') )

    pd.options.display.max_columns = 200
    pd.options.display.max_rows = 200
    pd.set_option( 'display.expand_frame_repr', False )
    pd.set_option('display.float_format', lambda x: '%.2f' % x)

    sns.set()

def descriptive_stats(data: pd.DataFrame, bound_dist: int):
    mean = pd.DataFrame(data.mean())
    std = pd.DataFrame(data.std())
    median = pd.DataFrame(data.median())
    mode = pd.DataFrame(data.mode().T.iloc[:,0])

    q1,q3 = pd.DataFrame(data.quantile(0.25)), pd.DataFrame(data.quantile(0.75))

    iqr = q3[0.75] - q1[0.25]
    lower_bound, upper_bound = q1[0.25] - (bound_dist*iqr), q3[0.75] +
(bound_dist*iqr)

    maximum = pd.DataFrame(data.max())
    minimum = pd.DataFrame(data.min())

    p1,p5,p95,p99 =
pd.DataFrame(data.quantile(0.01)),pd.DataFrame(data.quantile(0.05)),pd.DataFram
e(data.quantile(0.95)),pd.DataFrame(data.quantile(0.99))

    skew = pd.DataFrame(data.skew())
    kurtosis = pd.DataFrame(data.kurtosis())

    descriptive_statistics =
pd.concat([mean,std,minimum,p1,p5,lower_bound,q1,mode,median,q3,upper_bound,p
95,p99,maximum,skew,kurtosis],axis=1)

    descriptive_statistics.columns =
['mean','std','minimum','p1','p5','lower_bound','q1','mode','median','q3','upper_bound',
'p95','p99','maximum',
                              'skew','kurtosis']

    return descriptive_statistics

def group_descriptive_stats(data: pd.DataFrame, agg_feature: str, info_feature: str,
bound_dist: int):

    grouped_data = data[[agg_feature,info_feature]].groupby(agg_feature)

    mean = grouped_data.mean()
    std = grouped_data.std()
    median = grouped_data.median()
    mode = grouped_data.agg([stats.mode]).apply(lambda x: x[0][0][0],axis=1)
```

```
q1,q3 = grouped_data.quantile(0.25), grouped_data.quantile(0.75)

    iqr = q3 - q1
    lower_bound, upper_bound = q1 - (bound_dist*iqr), q3 + (bound_dist*iqr)

    maximum = grouped_data.max()
    minimum = grouped_data.min()

    p1,p5,p95,p99 = grouped_data.quantile(0.01),grouped_data.quantile(0.05),grouped_data.quantile(0.95),grouped_data.quantile(0.99)

    skew = grouped_data.skew()
    kurtosis = grouped_data.apply(pd.DataFrame.kurt).drop(agg_feature,axis=1)

    grouped_descriptive_statistics = pd.concat([mean,std,minimum,p1,p5,lower_bound,q1,mode,median,q3,upper_bound,p95,p99,maximum,skew,kurtosis],axis=1)

    grouped_descriptive_statistics.columns = ['mean','std','minimum','p1','p5','lower_bound','q1','mode','median','q3','upper_bound','p95','p99','maximum','skew','kurtosis']

    return grouped_descriptive_statistics

def select_num_attr(data: pd.DataFrame):

    return data.select_dtypes(include=['int64', 'float64'])

def get_geofile(url):
    geofile = geopandas.read_file(url)
    return geofile

def z_score(data, k_dev):
    mean = np.round(data.mean(), decimals=2)
    std_dev = np.round(data.std(), decimals=2)

    z_scores = [ (x - mean)/std_dev for x in data ]
    return data[(np.abs(z_scores) > k_dev)]

def modified_zscore(data, thresh):
    median = np.median(data)
    median_absolute_deviation = np.median([np.abs(x - median) for x in data])

    modified_z_scores = [0.6745 * (x - median) / median_absolute_deviation for x in
data]
    return data[(np.abs(modified_z_scores) > thresh)]

def iqr(data, dist):

    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1

    return data[(data < (Q1 - (dist * IQR))) | (data > (Q3 + (dist * IQR)))]

def number_iqr_outliers(data: pd.DataFrame,dist: int) -> pd.DataFrame:

    outliers = {}
    for feature in data.columns:
    #     print(f"{feature} outliers", end="\n")
    #     print(iqr(num_attributes[feature], 1.5))
        outliers[feature] = iqr(num_attributes[feature], dist)
    num_outliers = [outliers[feature].shape[0] for feature in outliers.keys()]
    perc_records = [(n_outliers/data.shape[0])*100 for n_outliers in num_outliers]
    outliers_data =
{'num_iqr_outliers':num_outliers,'perc_outlier_records':perc_records}
    iqr_outliers = pd.DataFrame(data=outliers_data, index=data.columns.to_list())

    return iqr_outliers

orig_df = pd.read_csv('data/kc_house_data.csv')
house_df_eda = orig_df.copy(deep=True)

rows, cols = choose_grid(len(num_attributes.columns))
iterate_through_data(num_attributes,num_attributes.columns,rows,cols)

multiple_scatter_plots(house_df_eda.loc[:, house_df_eda.columns != 'id'])
group_descriptive_stats(num_attr_fs,'zipcode','m2_living',bound_dist)

# Calculate its median price per region per living size and per lot size.
```

```
zipcode_median_price = house_df_ins[['price', 'zipcode','living_size','lot_size']].groupby(['zipcode','living_size','lot_size']).median().reset_index()
zipcode_median_price.columns = ['zipcode','living_size','lot_size','median_price']

house_df_ins = pd.merge(house_df_fs,zipcode_median_price,on=['zipcode','living_size','lot_size'],how='inner')

# Calculate its median price per m2 per region per living size and per lot size.

zipcode_median_price_m2 = house_df_ins[['price_m2', 'zipcode','living_size','lot_size']].groupby(['zipcode','living_size','lot_size']).median().reset_index()
zipcode_median_price_m2.columns = ['zipcode','living_size','lot_size','median_price_m2']

house_df_ins = pd.merge(house_df_ins,zipcode_median_price_m2,on=['zipcode','living_size','lot_size'],how='inner')
# Create house recommendation system

house_df_ins['status'] = house_df_ins[['price','condition','median_price']].apply(lambda x: 'to buy' if (x[0] < x[2]) & (x[1] > 3)\
                                                                          else 'to consider' if (x[0] < x[2]) & (x[1] == 3)\
                                                                          else 'to compare' if (x[0] > x[2]) & (x[1] > 3)\
                                                                          else 'not worth buying',  axis = 1)

recommended_houses_ordered = house_df_ins.loc[(house_df_ins['perc_value_below_median_price'] > 0) & (house_df_ins['status'] == 'to buy'),:].copy(deep=True)
```

Dashboard.py:

```
import geopandas
import streamlit as st
import pandas as pd
import numpy as np
import math
import folium

from PIL import Image
from numerize.numerize import numerize
from streamlit_folium import folium_static
from folium.plugins import MarkerCluster
from datetime import datetime
import time

import plotly.express as px
from matplotlib import pyplot as plt

hide_streamlit_style = """
            <style>
            #MainMenu {visibility: hidden;}
            footer {visibility: hidden;}
            </style>
            """
def main():

    status = 'initial_page'

    set_page_header(status)

    option = data_size_choice()

    if option != '':
        status = dashboard_choice(status)

    path = 'kc_house_data.csv'
    url =
'https://opendata.arcgis.com/datasets/83fc2e72903343aabff6de8cb445b81c_2.geojson
'

    if status == 'macro_analysis':
        macro_dashboard(path,url,option)

    if status == 'report_analysis':
        report_dashboard(path,url,option)

    set_page_footer()
```

```
def set_page_header(status):

    # Visualization Setup
    st.set_page_config(page_title='Real Trado Recommend', page_icon='Real
Trado.png',
                layout='wide', initial_sidebar_state='expanded')

    #Set header
    c1, c2 = st.columns((1, 5))

    # image
    with c1:
        photo = Image.open('Real Trado.png')
        st.image(photo)

    # title and subtitle
    with c2:
        st.write('')
        HR_format = '<div> <p style="font-family:sans-serif;' \
                    'color:#105088 ;' \
                    'font-size: 40px;' \
                    'font-weight: bold;' \
                    'font-style: normal;' \
                    'text-align: left;">' \
                    'Real Trado Investment Dashboard <br> AI Recommendation & Price
Comparison Engine</p> </div>'
        st.markdown(HR_format, unsafe_allow_html=True)

    return None

def set_page_footer():
    st.markdown('---')
    st.subheader('Investment Dashboard Purpose:')

    st.markdown('The **Macro Dashboard** allows the clients to check a data
overview and do some basic statistical analysis.')

    st.markdown('The **Recommendation Report Dashboard** allows the clients to
check his required report on various data and recommendation.')

    st.markdown(hide_streamlit_style, unsafe_allow_html=True)
    url = "https://real-trado-dapp.vercel.app/"
    st.markdown("[Back to Home Page](%s)" % url)
    st.write('')

def set_report_sidebar(data):

    # filters
    invest_opt = st.sidebar.multiselect('Investment Option', data.status.unique(),
default=['to buy'])
    house_id = st.sidebar.multiselect('House Id', np.sort(data.id.unique()))
    f_zipcode = st.sidebar.multiselect('Enter zipcode',
np.sort(data['zipcode'].unique()))
    f_attributes = st.sidebar.multiselect('Enter columns', data.columns,
default=['id','zipcode','price','selling_price','profit_est','best_season_selling_price','be
st_season_profit_est','status'])
    f_living_size = st.sidebar.multiselect('Living Size', data.living_size.unique())
    f_lot_size = st.sidebar.multiselect('Lot Size', data.lot_size.unique())
    min_price = int(data.price.min())
    max_price = int(data.price.max())
#     st.sidebar.subheader('Select Min Price')
#     f_price = st.sidebar.slider('Price', min_price,
#                                          max_price,
#                                          min_price)
#     min_profit = int(data.profit_est.min())
#     max_profit = int(data.profit_est.max())
#     st.sidebar.subheader('Select Min Profit')
#     f_profit = st.sidebar.slider('Profit', min_profit,
#                                        max_profit,
#                                        min_profit)


    filters_dict = {'status':invest_opt,
                   'id':house_id,
                   'zipcode':f_zipcode,
                   'attributes':f_attributes,
                   'living_size':f_living_size,
                   'lot_size':f_lot_size}#,'price':f_price,'profit_est':f_profit}
    return filters_dict

def set_macro_sidebar(data):

    # filters
    f_attributes = st.sidebar.multiselect('Enter columns', data.columns)
    house_id = st.sidebar.multiselect('House Id', np.sort(data.id.unique()))
    f_zipcode = st.sidebar.multiselect('Enter zipcode',
np.sort(data['zipcode'].unique()))
    f_order_column = st.sidebar.selectbox('Order Data Overview by
column:',data.columns)
    f_ascending = st.sidebar.checkbox('Ascending')

    min_year_built = int(data['yr_built'].min())
    max_year_built = int(data['yr_built'].max())

    st.sidebar.subheader('Select Max Year Built')
    f_year_built = st.sidebar.slider('Year Built', min_year_built,
                                     max_year_built,
                                     max_year_built)

    # transform date attribute data type
    data.date = pd.to_datetime(data.date).dt.strftime('%Y-%m-%d')

    min_date = datetime.strptime(data['date'].min(), '%Y-%m-%d')
    max_date = datetime.strptime(data['date'].max(), '%Y-%m-%d')

    st.sidebar.subheader('Select Max Date')
    f_date = st.sidebar.slider('Date', min_date,
                                     max_date,
                                     max_date)

    min_price = int(data['price'].min())
    max_price = int(data['price'].max())
    avg_price = int(data['price'].mean())

    st.sidebar.subheader('Select Max Price')
    f_price = st.sidebar.slider('Price', min_price, max_price, max_price)
    st.sidebar.title('Attributes Options')

    f_bed = st.sidebar.selectbox('Max number of bedrooms',

sorted(set(data['bedrooms'].unique())),index=len(data['bedrooms'].unique())-2)
    f_bath = st.sidebar.selectbox('Max number of bathrooms',

sorted(set(data['bathrooms'].unique())),index=len(data['bathrooms'].unique())-1)
    f_floors = st.sidebar.selectbox('Max number of floors',
sorted(set(data['floors'].unique())),index=len(data['floors'].unique())-1)
    f_water = st.sidebar.checkbox('With water view')

    st.sidebar.markdown('*For more information about Real Estate Investment
Dashboard, please go to '
            '[Additional Information](#additional-information) section by the end of this
page.*')

    filters_dict = {'id':house_id,
                    'zipcode':f_zipcode,
                    'attributes':f_attributes,
                    'yr_built':f_year_built,
                    'date':f_date,
                    'price':f_price,
                    'bedrooms':f_bed,
                    'bathrooms':f_bath,
                    'floors':f_floors,
                    'waterfront':f_water}
    return filters_dict, f_order_column, f_ascending

def dashboard_choice(status):

    st.write('If it is taking too long for the page to load, '
                        'please select a smaller database size above')

    st.markdown("**Feel free to change between dashboards:**")

    f_dashboard = st.selectbox('Choose Dashboard',['','Macro Dashboard','Report
Dashboard'])
    if status == 'initial_page':
        if f_dashboard == 'Macro Dashboard':
            status = 'macro_analysis'
        if f_dashboard == 'Report Dashboard':
            status = 'report_analysis'

    return status
```

CHAPTER 7
SYSTEM TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects) and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- achieves the general result its stakeholder’s desire.

7.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

7.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

7.3 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

7.4 Acceptance Testing

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

In systems engineering it may involve black-box testing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

7.5 TEST CASES AND REPORTS:

Test Case ID	Description	Test Steps	Test Data	Expected Results	Actual Results	Test Pass/Fail
1	Login page	1)Enter email 2)Enter password	dgj@x.com D@rin	Login successful	Login successful	Pass
2	Buying a property	Click on the property	N/A	Show UI, payment page	Show UI, payment page	Pass
3	Selling a property	Click on Sell option in Navbar	N/A	Show selling details page	Show selling details page	Pass
4	Renting a property	Click on the property	N/A	Show UI, payment page	Show UI, payment page	Pass
5	AI recommend and price comparison	Click on dashboard and check results, research	API, Property listings and User inputs	Show price ranges, graphs, maps, charts, recommend and sorting	Show price ranges, graphs, maps, charts, recommend and sorting	Pass
6	Blockchain	1)Enter	0x86dBA09	Show	Show	Pass

	Network and Payment	Recipient Address 2)Enter Amount 3)Click Send	57B1597ae87697B28026E533eC62C1598	payment success page	payment success page	

Thus, the Real Trado application was able to pass all the test cases and was found to be working as per the proposed system.

CHAPTER 8
CONCLUSION

The Traditional Real Estate system is replaced with the Decentralized Application (DApp) through which the transactions and agreements can be made in a more efficient way. Moreover, it provides transparency and security which is of utmost importance in the present times. Some of the conclusions that can be made are listed below.

(i) Blockchain provides a secure, fast and reliable means of transferring the assets.

(ii) Artificial Intelligence makes it possible for the user to see the expected price and also provides better suggestions based on the input from the user.

(iii) The overall process can be more environmentally friendly than the traditional system as it doesn’t involve the transportation, manpower and paperwork that the traditional system needs. But it is equally important to know that the Blockchain needs a considerable amount of computational power to store the transactions in each block which might be the only downside of a system using Blockchain.

(iv) Building DApps will take some time and effort but the outcome of a DApp is a purely peer to peer system that provides a secure way of transferring data without the intermediary which is always present in a traditional system.

8.1 RESULTS AND DISCUSSION

The project prototype of the DApp is built after refinements and made available for users over the internet as it is hosted and deployed. This platform would be one of the beginning concepts of Web 3.0 that has been made into a working system. It enables all the specifications and features listed in the above headings required for real estate trading. The blockchain and artificial intelligence concepts well integrated makes it an extremely useful real-world application with few more refinements can make it much more refined and more feature rich.

Final Deployment of the project prototype:

https://real-trado-dapp.vercel.app/ (or) https://real-trado.vercel.app/

8.2 FUTURE ENHANCEMENTS

The use of Blockchain and Artificial Intelligence will attract both the buyers and sellers due to the various advantages it comes along with. Some of the work which will be done in the near future are mentioned below:

1. Implementation of an automated identity, document verification within the DApp with the help of Government agencies, services which further authenticates and validates the user and does property verification.
2. Creation of an additional payment method which would use digital currencies to do transactions. Also integrate it with the upcoming CDBC of India, the Digital Rupee, which will be very useful and futuristic. It can also be converted back into real world physical money.
3. Using AI to validate the seller's property, registration and formalities for faster and effectiveness instead of manual property validation of seller's properties. With an automated listing mechanism to the site after validation which reduces the DApp admin's work.
4. A token-based approach by which the sellers or buyers can also sell/buy a part of the property instead of getting it based on customer demand.

APPENDICES

A.1 Screenshots

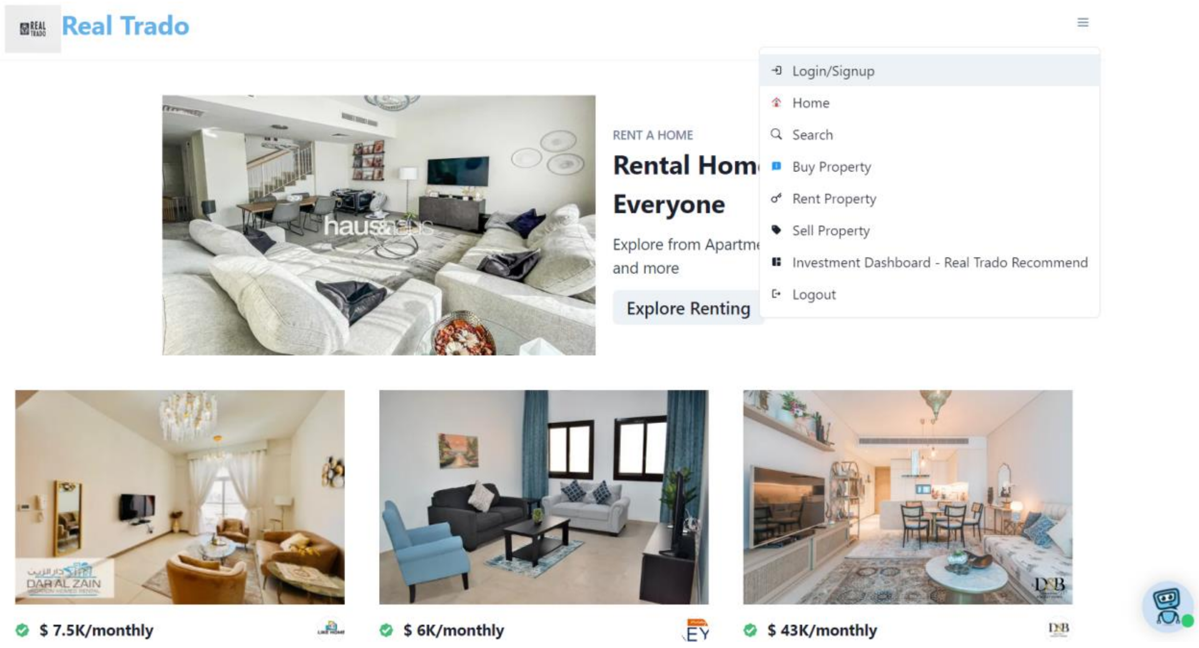
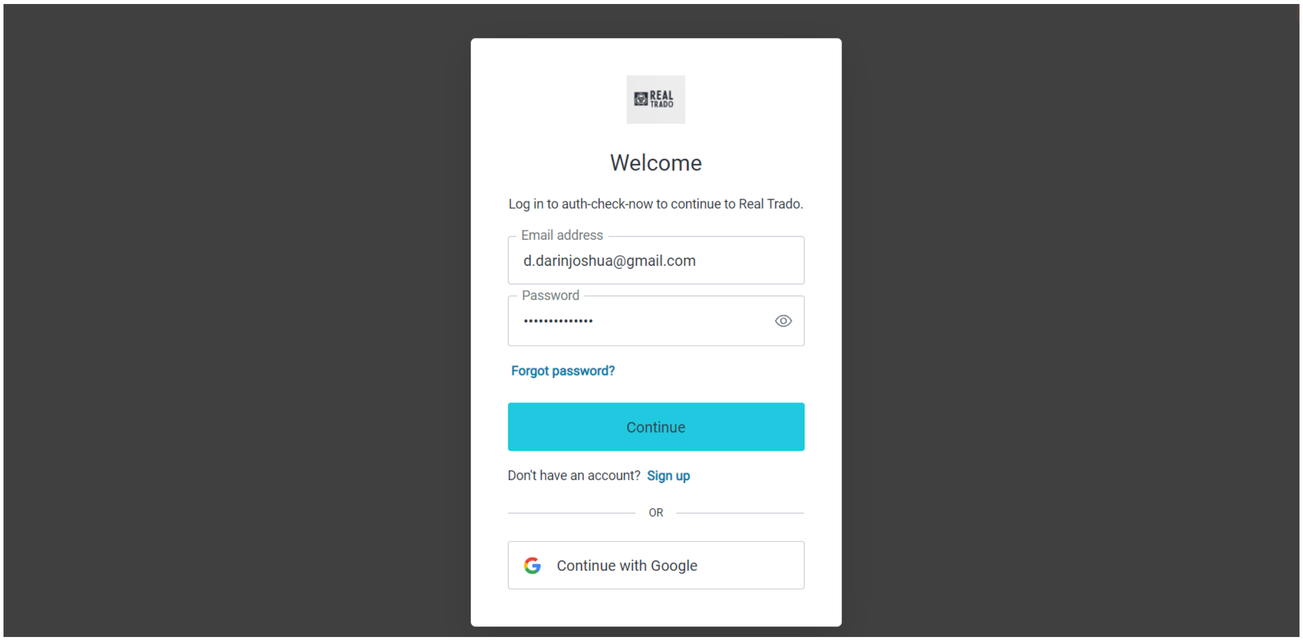


Fig 2.20 Home page with all features in the menu

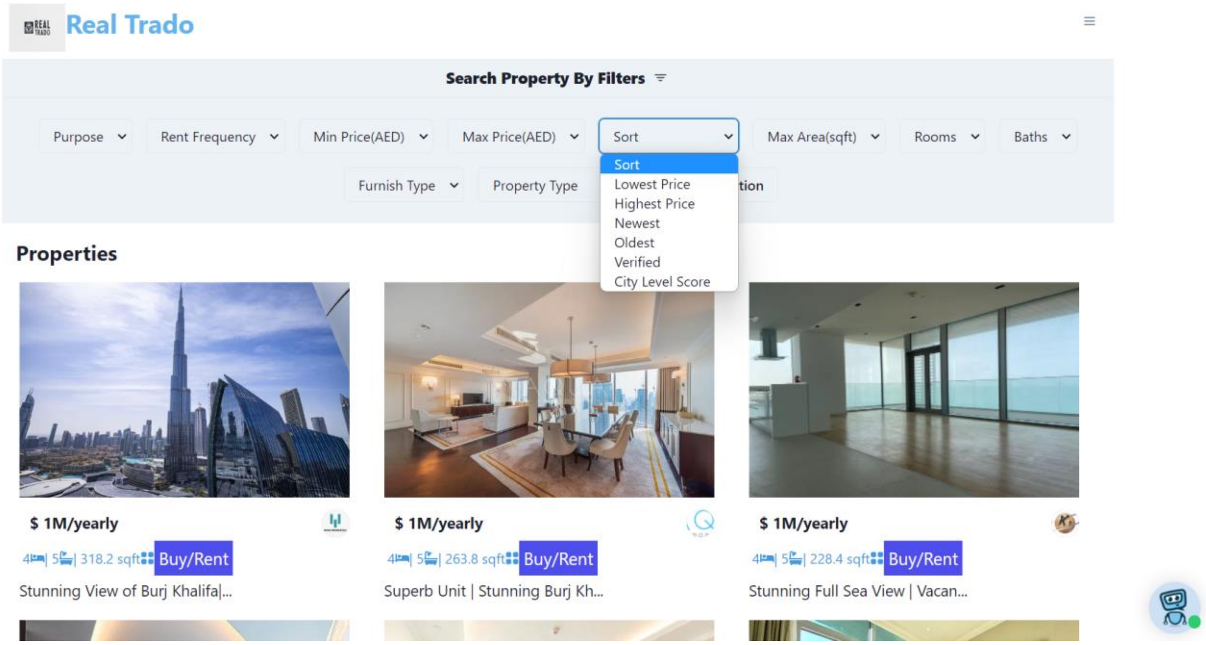

Welcome

Log in to auth-check-now to continue to Real Trado.

Email address: d.darinjoshua@gmail.com

Password: ••••••••••••••

Fig 2.21 Login/Signup authentication



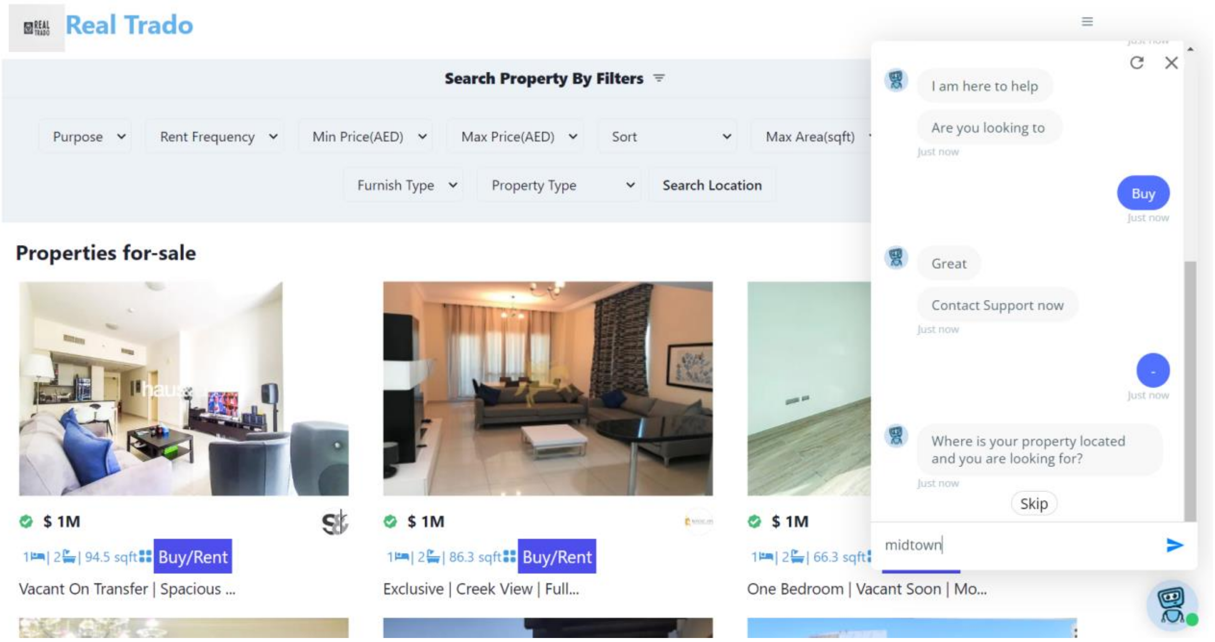
Properties

$ 1M/yearly
4 | 5 | 318.2 sqft
Stunning View of Burj Khalifa|...

$ 1M/yearly
4 | 5 | 263.8 sqft
Superb Unit | Stunning Burj Kh...

$ 1M/yearly
4 | 5 | 228.4 sqft
Stunning Full Sea View | Vacan...

Fig 2.22 Sorting and searching features of the site



Properties for-sale

$ 1M
1 | 2 | 94.5 sqft
Vacant On Transfer | Spacious ...

$ 1M
1 | 2 | 86.3 sqft
Exclusive | Creek View | Full...

$ 1M
1 | 2 | 66.3 sqft
One Bedroom | Vacant Soon | Mo...

I am here to help

Are you looking to
Just now

Buy
Just now

Great

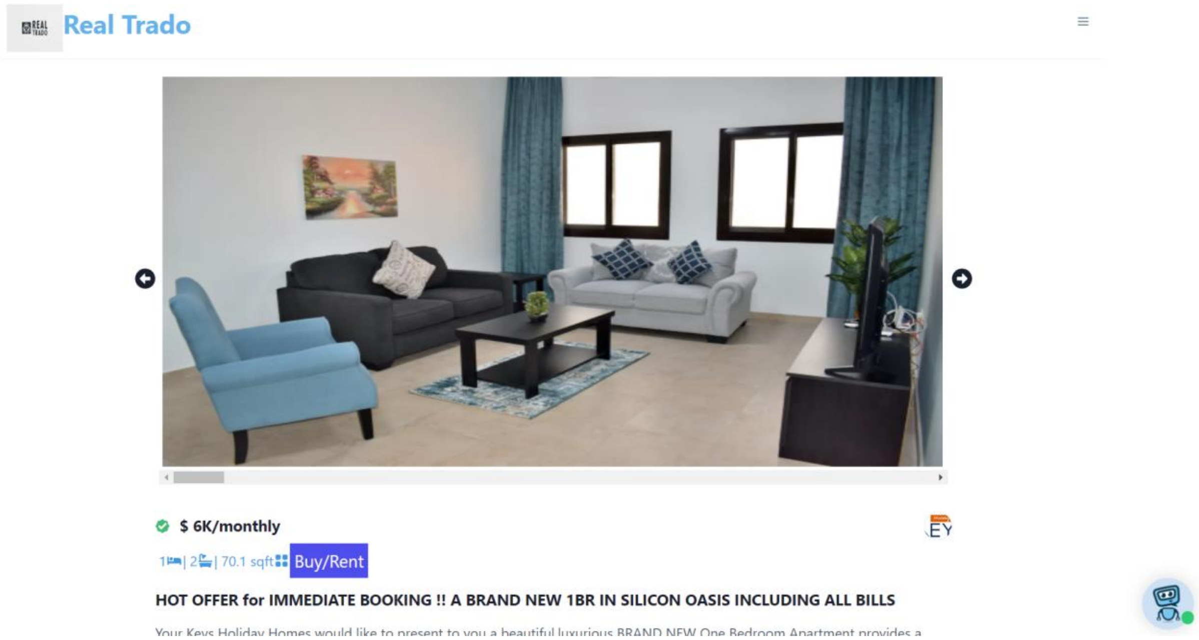
Contact Support now
Just now

-
Just now

Where is your property located and you are looking for?
Just now

midtown

Fig 2.23 AI chatbot for support, interaction and service



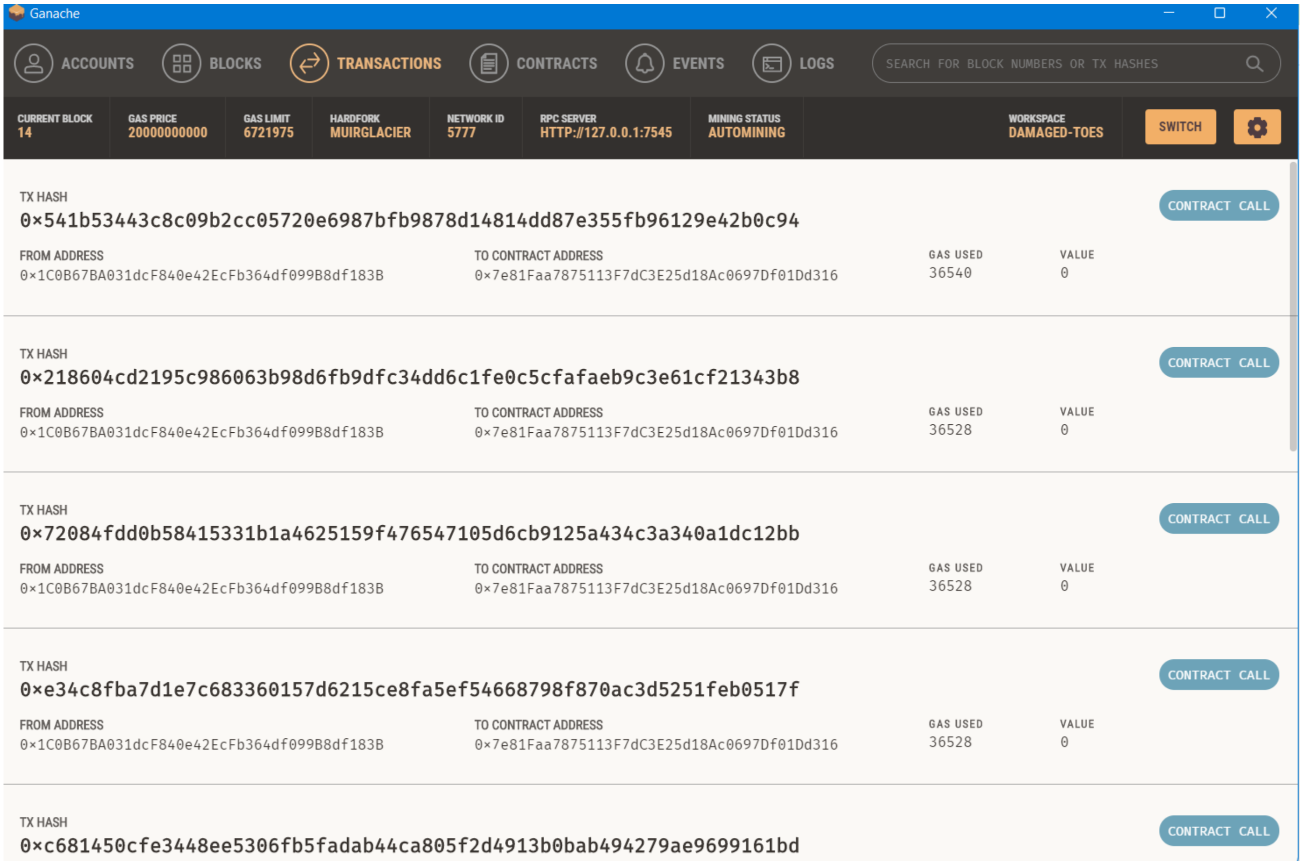
$ 6K/monthly

1 | 2 | 70.1 sqft

HOT OFFER for IMMEDIATE BOOKING !! A BRAND NEW 1BR IN SILICON OASIS INCLUDING ALL BILLS

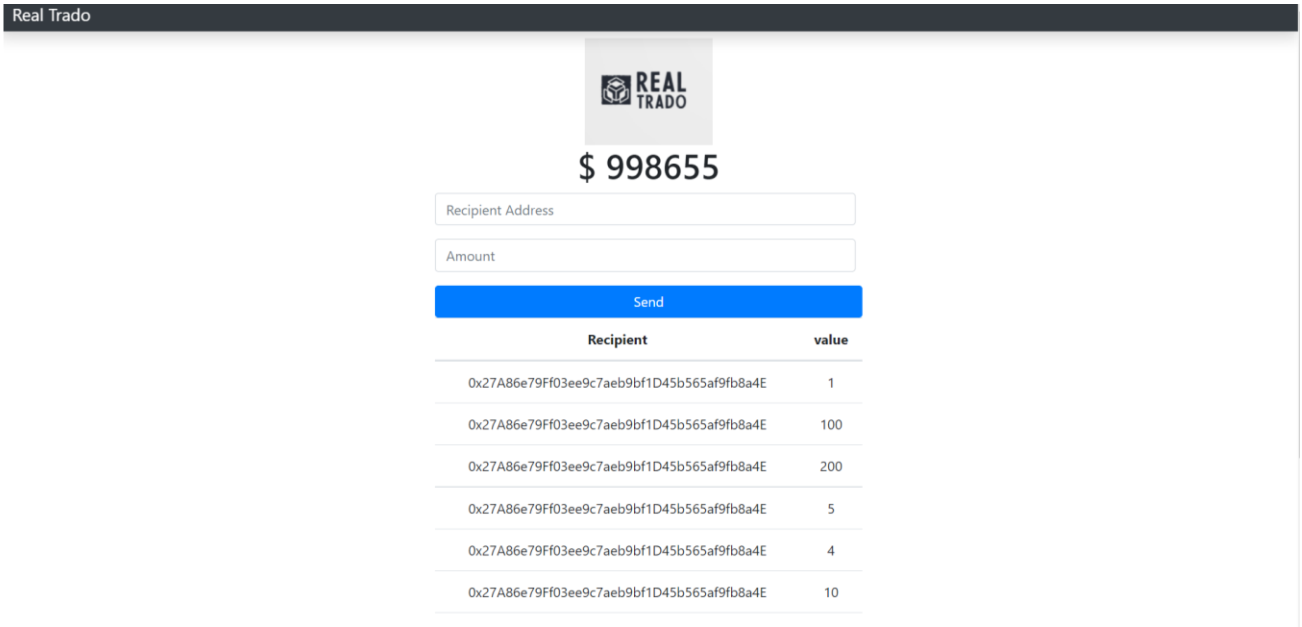
Your Keys Holiday Homes would like to present to you a beautiful luxurious BRAND NEW One Bedroom Apartment provides a

Fig 2.24 Specific property page



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK     NETWORK ID  RPC SERVER             MINING STATUS   WORKSPACE
14             20000000000  6721975    MUIRGLACIER  5777        HTTP://127.0.0.1:7545  AUTOMINING      DAMAGED-TOES

TX HASH
0×541b53443c8c09b2cc05720e6987bfb9878d14814dd87e355fb96129e42b0c94
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED   VALUE
0×1C0B67BA031dcF840e42EcFb364df099B8df183B   0×7e81Faa7875113F7dC3E25d18Ac0697Df01Dd316   36540      0

TX HASH
0×218604cd2195c986063b98d6fb9dfc34dd6c1fe0c5cfafaeb9c3e61cf21343b8
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED   VALUE
0×1C0B67BA031dcF840e42EcFb364df099B8df183B   0×7e81Faa7875113F7dC3E25d18Ac0697Df01Dd316   36528      0

TX HASH
0×72084fdd0b58415331b1a4625159f476547105d6cb9125a434c3a340a1dc12bb
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED   VALUE
0×1C0B67BA031dcF840e42EcFb364df099B8df183B   0×7e81Faa7875113F7dC3E25d18Ac0697Df01Dd316   36528      0

TX HASH
0×e34c8fba7d1e7c683360157d6215ce8fa5ef54668798f870ac3d5251feb0517f
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED   VALUE
0×1C0B67BA031dcF840e42EcFb364df099B8df183B   0×7e81Faa7875113F7dC3E25d18Ac0697Df01Dd316   36528      0

TX HASH
0×c681450cfe3448ee5306fb5fadab44ca805f2d4913b0bab494279ae9699161bd
```

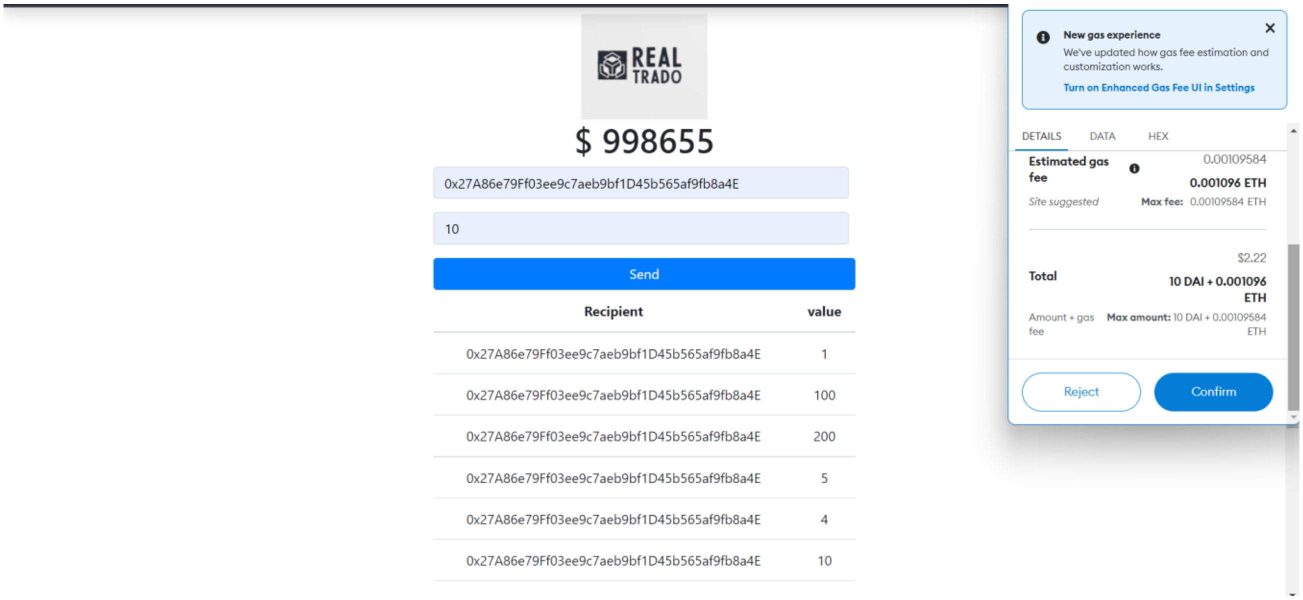
Fig 2.25 Private blockchain network based on Ganache



$ 998655

Recipient	value
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	1
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	100
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	200
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	4
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	10

Fig 2.26 Blockchain payment page



$ 998655

0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E

10

Recipient	value
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	1
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	100
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	200
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	4
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	10

New gas experience

We've updated how gas fee estimation and customization works.

Turn on Enhanced Gas Fee UI in Settings

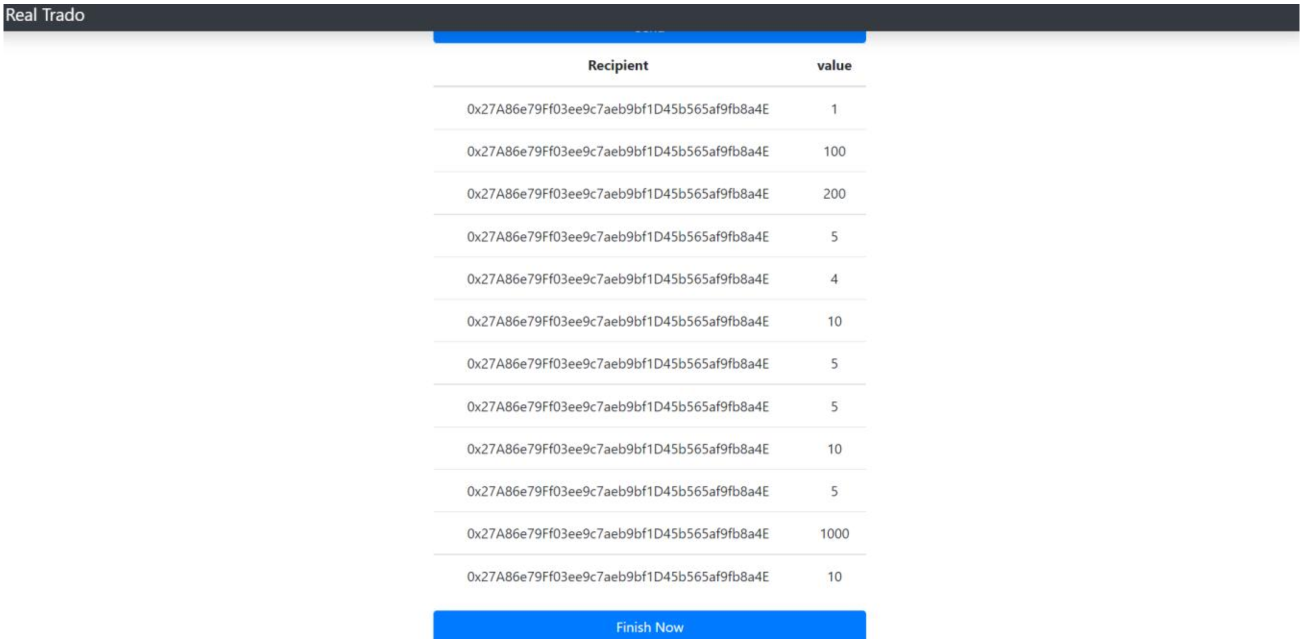
Estimated gas fee 0.00109584 **0.001096 ETH**

Site suggested **Max fee:** 0.00109584 ETH

Total $2.22 **10 DAI + 0.001096 ETH**

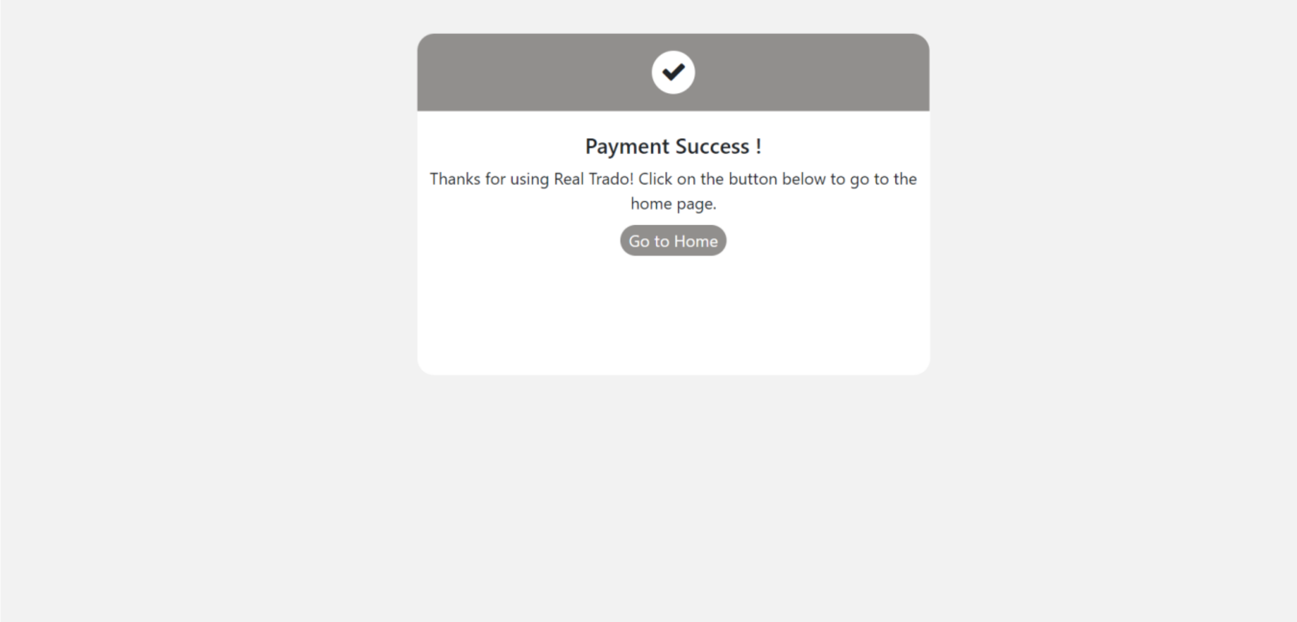
Amount + gas fee **Max amount:** 10 DAI + 0.00109584 ETH

Fig 2.27 Blockchain transaction done through MetaMask



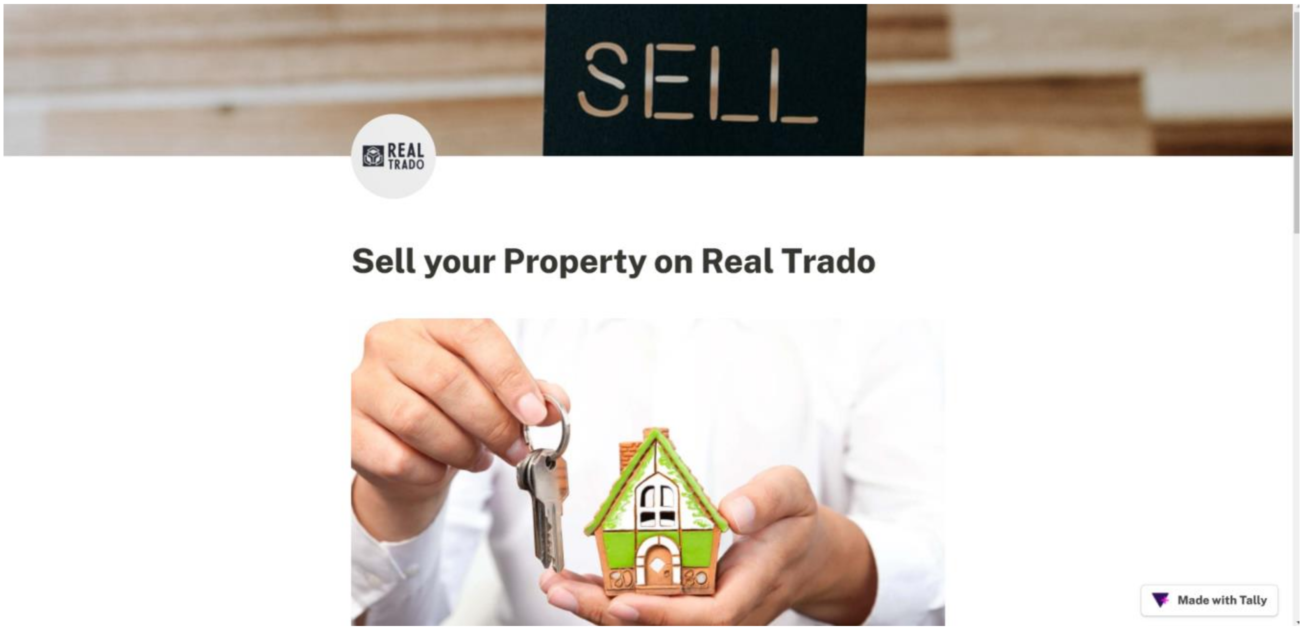
Recipient	value
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	1
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	100
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	200
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	4
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	10
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	10
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	5
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	1000
0x27A86e79Ff03ee9c7aeb9bf1D45b565af9fb8a4E	10

Fig 2.28 Transaction ledger history in the page


Payment Success !

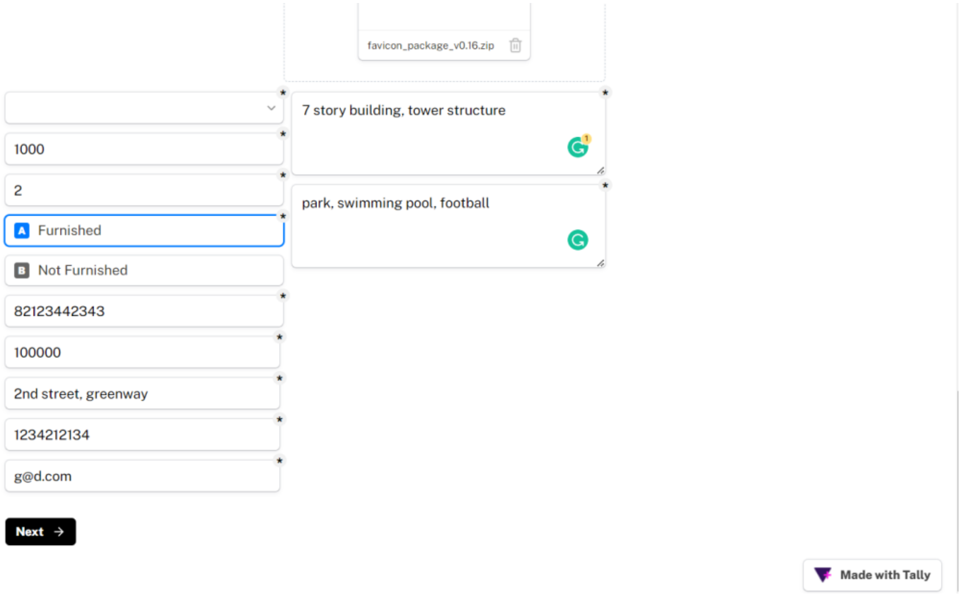
Thanks for using Real Trado! Click on the button below to go to the home page.

Fig 2.29 Payment success page



Sell your Property on Real Trado

Fig 2.30 Property selling page



favicon_package_v0.16.zip

7 story building, tower structure

park, swimming pool, football

1000

2

A Furnished

B Not Furnished

82123442343

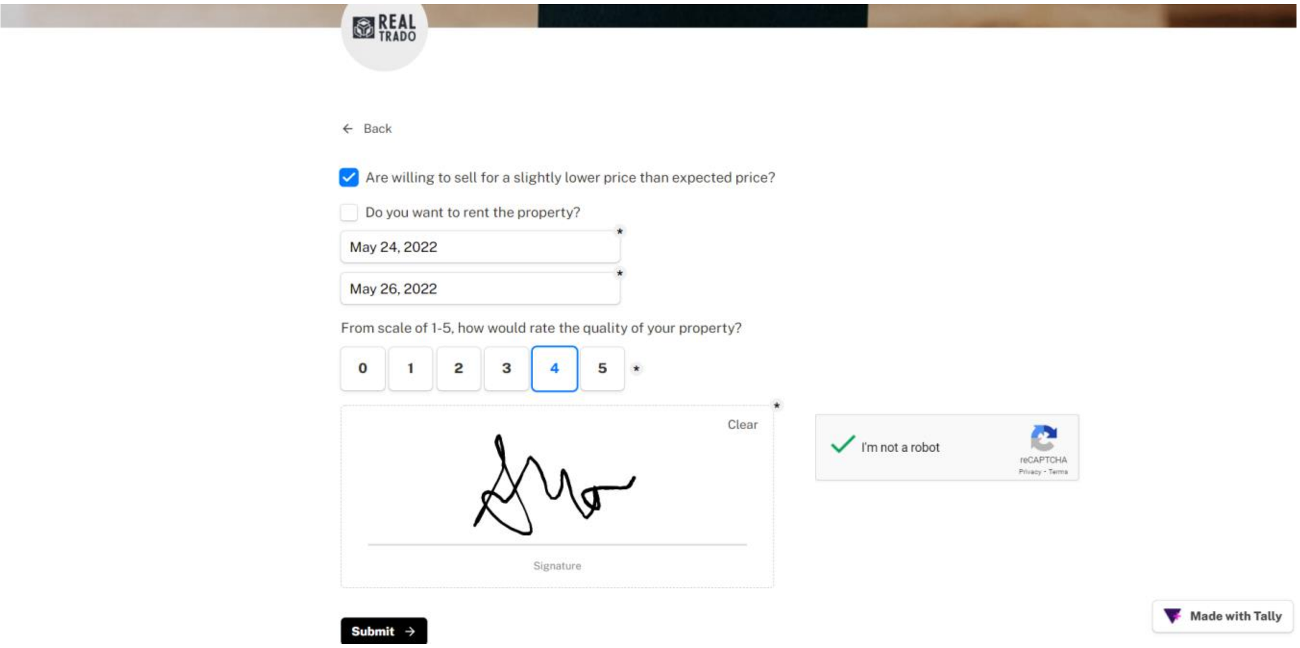
100000

2nd street, greenway

1234212134

g@d.com

Fig 2.31 Details required to sell property



← Back

☑ Are willing to sell for a slightly lower price than expected price?

□ Do you want to rent the property?

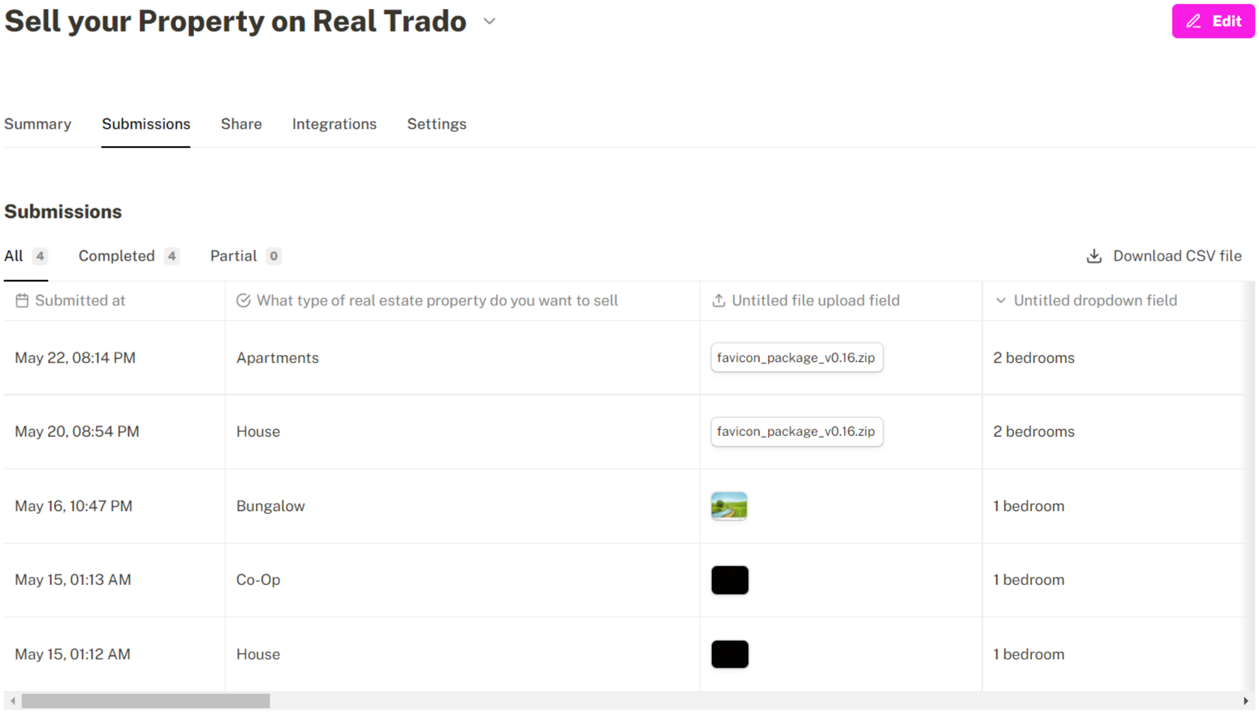
May 24, 2022 *

May 26, 2022 *

From scale of 1-5, how would rate the quality of your property?

0	1	2	3	4	5	*

Fig 2.32 Property listing period and confirmation

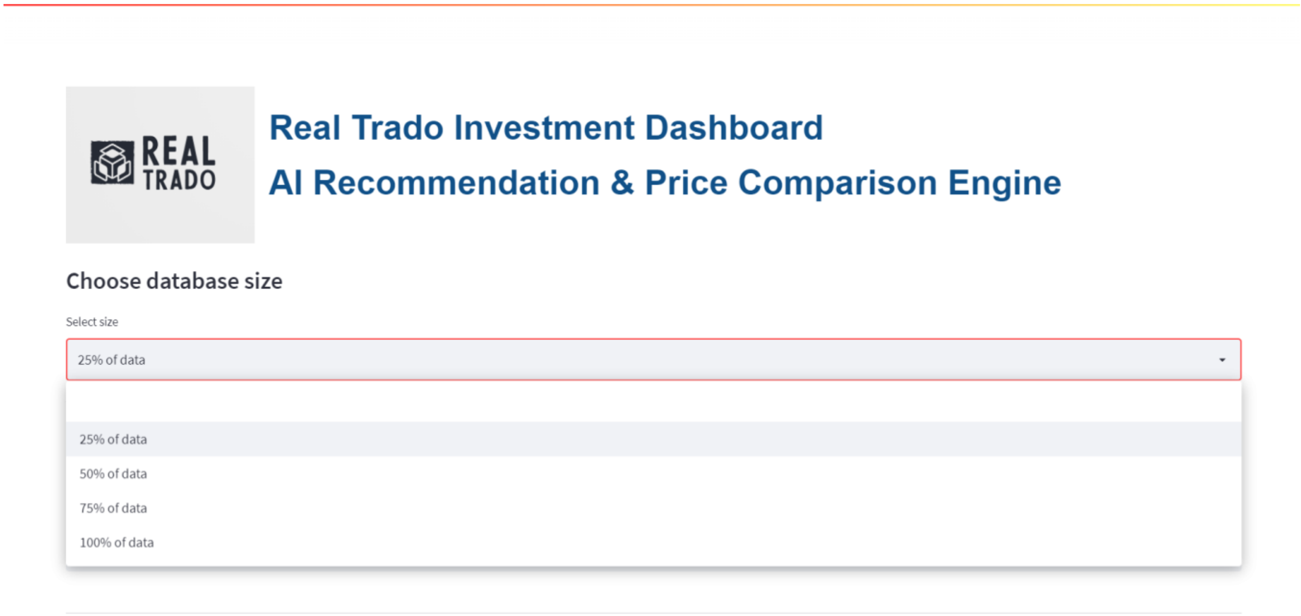


Sell your Property on Real Trado

Submissions

Submitted at	What type of real estate property do you want to sell	Untitled file upload field	Untitled dropdown field
May 22, 08:14 PM	Apartments	favicon_package_v0.16.zip	2 bedrooms
May 20, 08:54 PM	House	favicon_package_v0.16.zip	2 bedrooms
May 16, 10:47 PM	Bungalow		1 bedroom
May 15, 01:13 AM	Co-Op		1 bedroom
May 15, 01:12 AM	House		1 bedroom

Fig 2.33 Admin page to authenticate the details of property of the seller



Real Trado Investment Dashboard

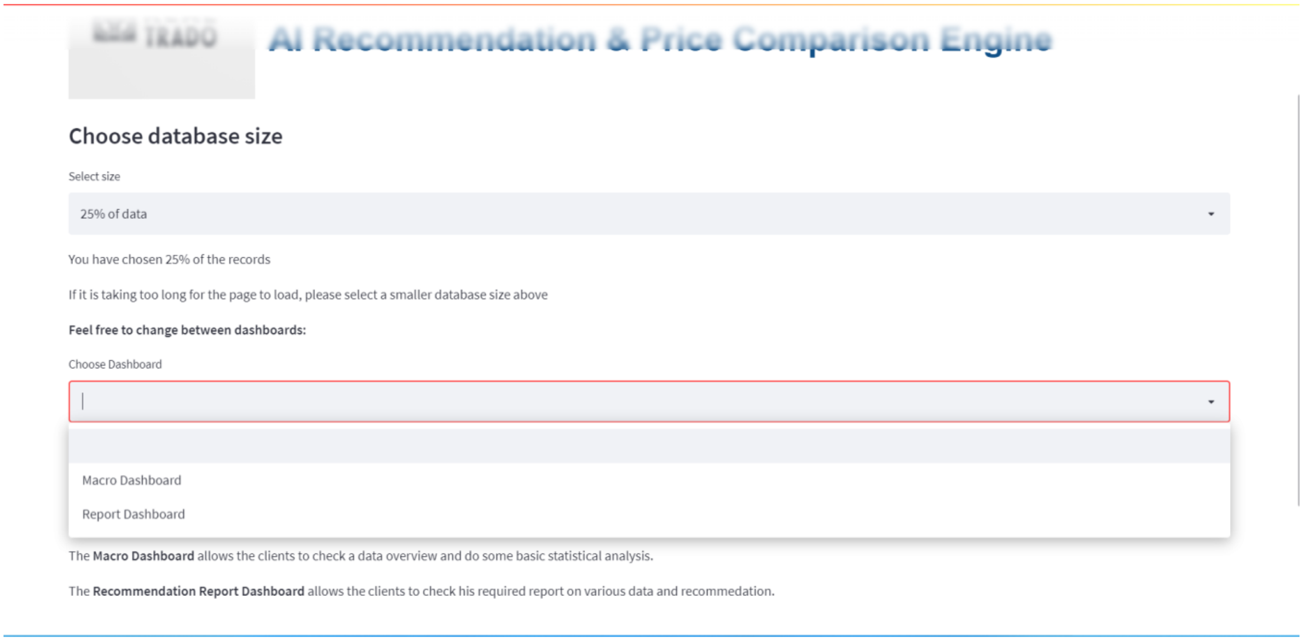
AI Recommendation & Price Comparison Engine

Choose database size

Select size

25% of data

Fig 2.34 Investment dashboard to access AI features



AI Recommendation & Price Comparison Engine

Choose database size

Select size

25% of data

You have chosen 25% of the records

If it is taking too long for the page to load, please select a smaller database size above

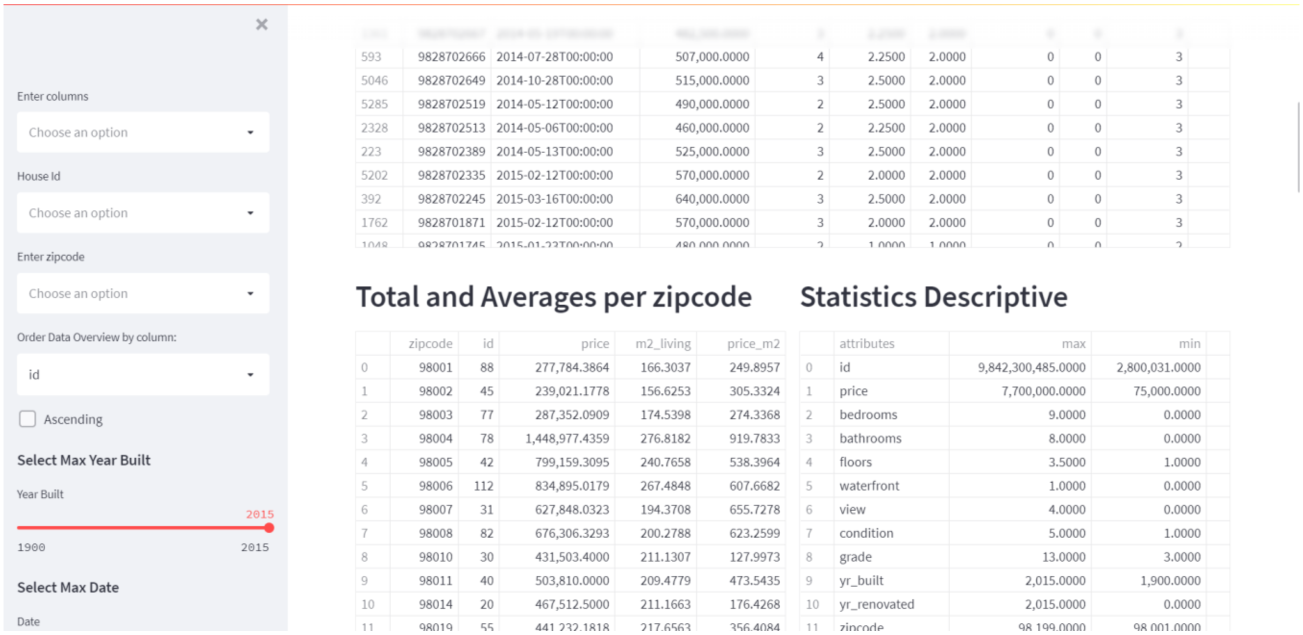
Feel free to change between dashboards:

Choose Dashboard

The **Macro Dashboard** allows the clients to check a data overview and do some basic statistical analysis.

The **Recommendation Report Dashboard** allows the clients to check his required report on various data and recommedation.

Fig 2.35 Choosing database size and report for research



Enter columns

Choose an option

House Id

Choose an option

Enter zipcode

Choose an option

Order Data Overview by column:

id

□ Ascending

Select Max Year Built

Year Built

2015

1900 2015

Select Max Date

Date

593	9828702666	2014-07-28T00:00:00	507,000.0000	4	2.2500	2.0000	0	0	3
5046	9828702649	2014-10-28T00:00:00	515,000.0000	3	2.5000	2.0000	0	0	3
5285	9828702519	2014-05-12T00:00:00	490,000.0000	2	2.5000	2.0000	0	0	3
2328	9828702513	2014-05-06T00:00:00	460,000.0000	2	2.2500	2.0000	0	0	3
223	9828702389	2014-05-13T00:00:00	525,000.0000	3	2.5000	2.0000	0	0	3
5202	9828702335	2015-02-12T00:00:00	570,000.0000	2	2.0000	2.0000	0	0	3
392	9828702245	2015-03-16T00:00:00	640,000.0000	3	2.5000	2.0000	0	0	3
1762	9828701871	2015-02-12T00:00:00	570,000.0000	3	2.0000	2.0000	0	0	3
1048	9828701745	2015-01-23T00:00:00	480,000.0000	2	1.0000	1.0000	0	0	2

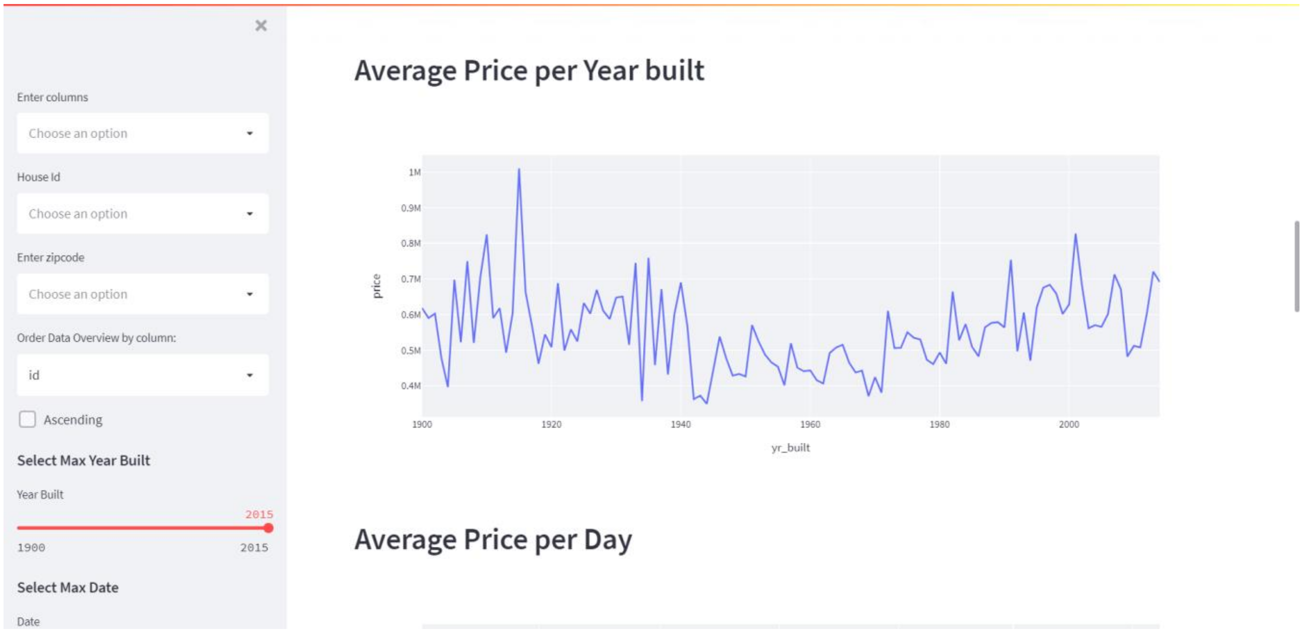
Total and Averages per zipcode

	zipcode	id	price	m2_living	price_m2
0	98001	88	277,784.3864	166.3037	249.8957
1	98002	45	239,021.1778	156.6253	305.3324
2	98003	77	287,352.0909	174.5398	274.3368
3	98004	78	1,448,977.4359	276.8182	919.7833
4	98005	42	799,159.3095	240.7658	538.3964
5	98006	112	834,895.0179	267.4848	607.6682
6	98007	31	627,848.0323	194.3708	655.7278
7	98008	82	676,306.3293	200.2788	623.2599
8	98010	30	431,503.4000	211.1307	127.9973
9	98011	40	503,810.0000	209.4779	473.5435
10	98014	20	467,512.5000	211.1663	176.4268
11	98019	55	441,232.1818	217.6563	356.4084

Statistics Descriptive

	attributes	max	min
0	id	9,842,300,485.0000	2,800,031.0000
1	price	7,700,000.0000	75,000.0000
2	bedrooms	9.0000	0.0000
3	bathrooms	8.0000	0.0000
4	floors	3.5000	1.0000
5	waterfront	1.0000	0.0000
6	view	4.0000	0.0000
7	condition	5.0000	1.0000
8	grade	13.0000	3.0000
9	yr_built	2,015.0000	1,900.0000
10	yr_renovated	2,015.0000	0.0000
11	zipcode	98,199.0000	98,001.0000

Fig 2.36 Macro Report of total and averages per zip code



Enter columns

Choose an option

House Id

Choose an option

Enter zipcode

Choose an option

Order Data Overview by column:

id

□ Ascending

Select Max Year Built

Year Built

2015

1900 2015

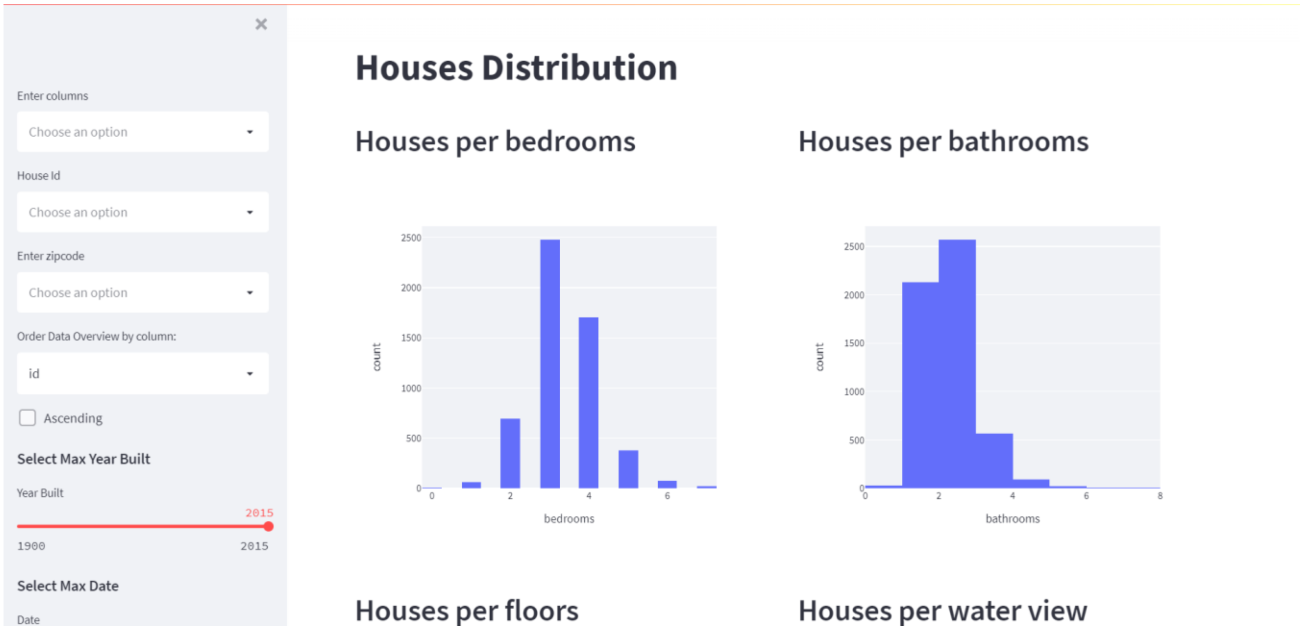
Select Max Date

Date

Average Price per Year built

Average Price per Day

Fig 2.37 Average price per year built



×

Enter columns

Choose an option

House Id

Choose an option

Enter zipcode

Choose an option

Order Data Overview by column:

id

□ Ascending

Select Max Year Built

Year Built

2015

1900 2015

Select Max Date

Date

Houses Distribution

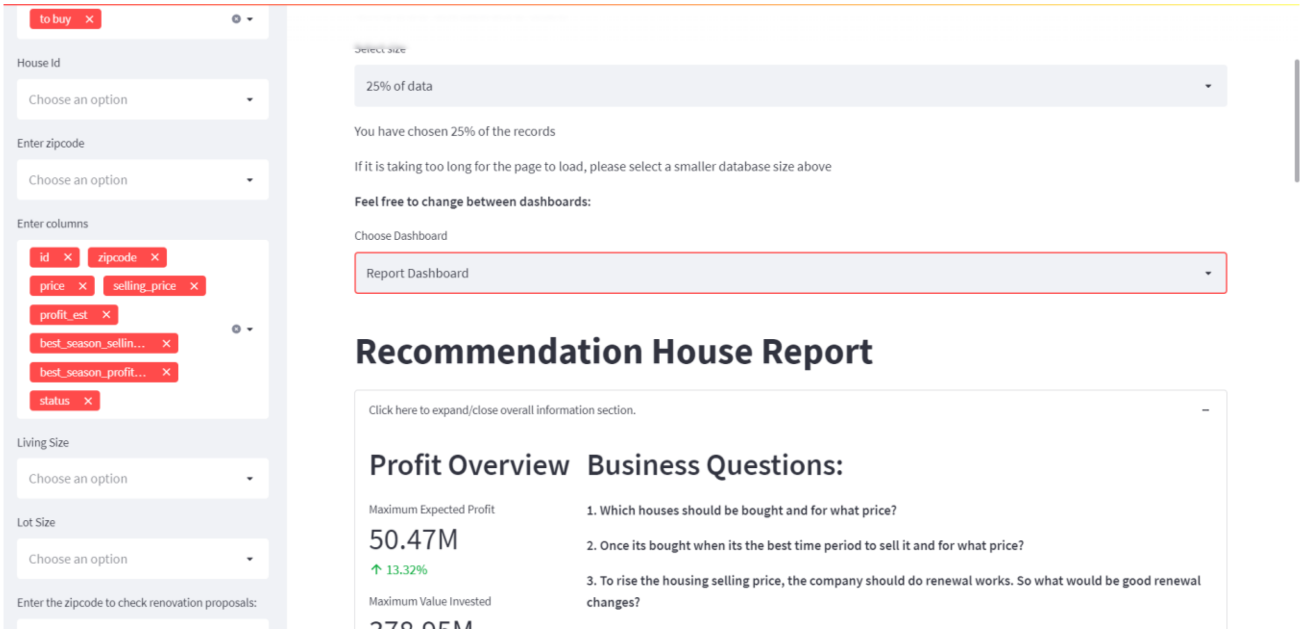
Houses per bedrooms

Houses per bathrooms

Houses per floors

Houses per water view

Fig 2.38 Houses Distribution Analysis



to buy ×

House Id

Choose an option

Enter zipcode

Choose an option

Enter columns

id × zipcode ×

price × selling_price ×

profit_est ×

best_season_sellin... ×

best_season_profit... ×

status ×

Living Size

Choose an option

Lot Size

Choose an option

Enter the zipcode to check renovation proposals:

Select size

25% of data

You have chosen 25% of the records

If it is taking too long for the page to load, please select a smaller database size above

Feel free to change between dashboards:

Choose Dashboard

Report Dashboard

Recommendation House Report

Click here to expand/close overall information section.

Profit Overview

Maximum Expected Profit

50.47M

↑ 13.32%

Maximum Value Invested

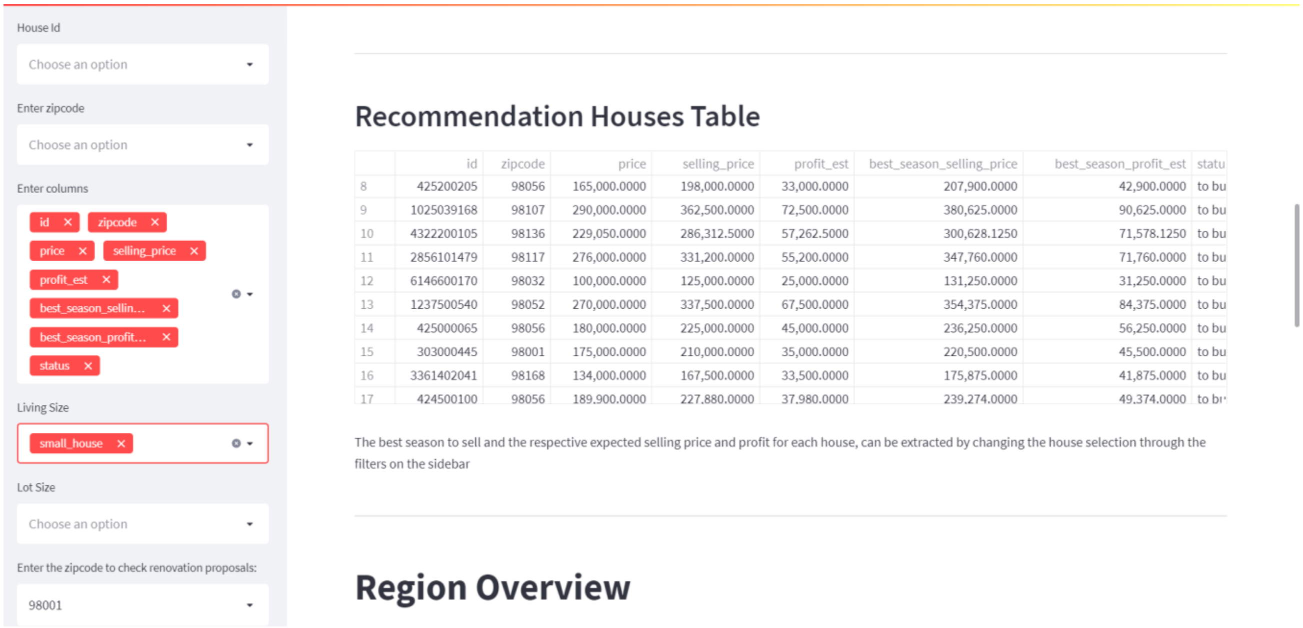
Business Questions:

1. Which houses should be bought and for what price?

2. Once its bought when its the best time period to sell it and for what price?

3. To rise the housing selling price, the company should do renewal works. So what would be good renewal changes?

Fig 2.39 Recommendation House Report



House Id

Choose an option

Enter zipcode

Choose an option

Enter columns

id ✕ zipcode ✕ price ✕ selling_price ✕ profit_est ✕ best_season_sellin... ✕ best_season_profit... ✕ status ✕

Living Size

small_house ✕

Lot Size

Choose an option

Enter the zipcode to check renovation proposals:

98001

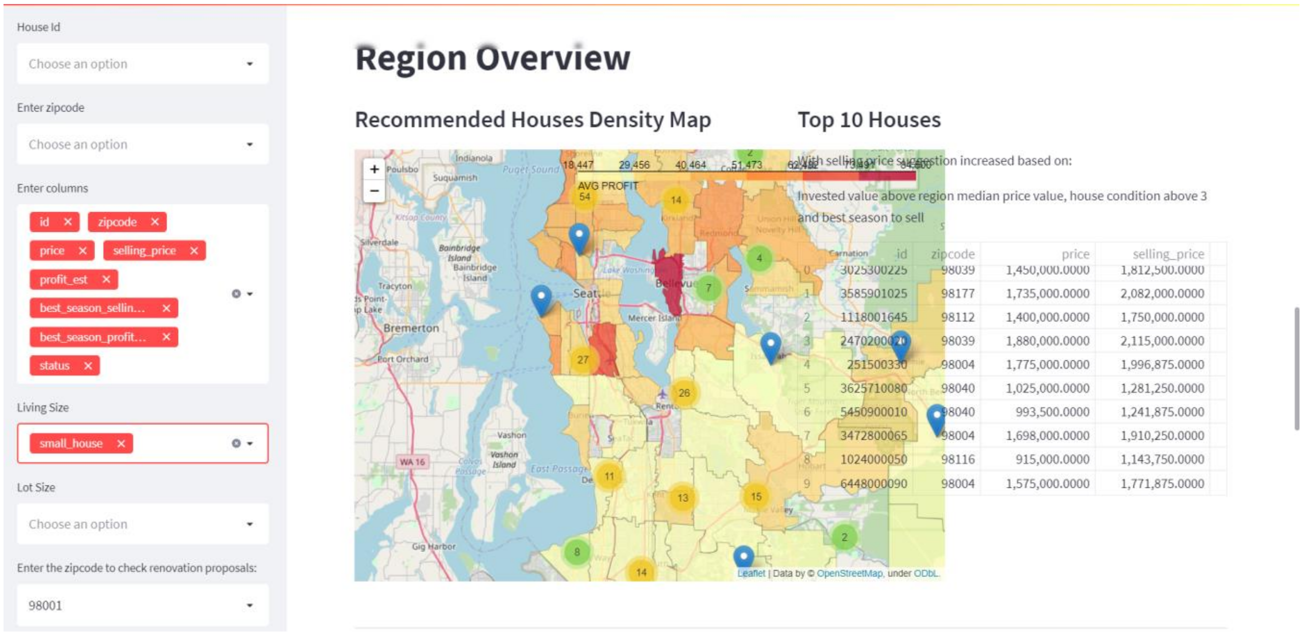
Recommendation Houses Table

	id	zipcode	price	selling_price	profit_est	best_season_selling_price	best_season_profit_est	statu
8	425200205	98056	165,000.0000	198,000.0000	33,000.0000	207,900.0000	42,900.0000	to bu
9	1025039168	98107	290,000.0000	362,500.0000	72,500.0000	380,625.0000	90,625.0000	to bu
10	4322200105	98136	229,050.0000	286,312.5000	57,262.5000	300,628.1250	71,578.1250	to bu
11	2856101479	98117	276,000.0000	331,200.0000	55,200.0000	347,760.0000	71,760.0000	to bu
12	6146600170	98032	100,000.0000	125,000.0000	25,000.0000	131,250.0000	31,250.0000	to bu
13	1237500540	98052	270,000.0000	337,500.0000	67,500.0000	354,375.0000	84,375.0000	to bu
14	425000065	98056	180,000.0000	225,000.0000	45,000.0000	236,250.0000	56,250.0000	to bu
15	303000445	98001	175,000.0000	210,000.0000	35,000.0000	220,500.0000	45,500.0000	to bu
16	3361402041	98168	134,000.0000	167,500.0000	33,500.0000	175,875.0000	41,875.0000	to bu
17	424500100	98056	189,900.0000	227,880.0000	37,980.0000	239,274.0000	49,374.0000	to bu

The best season to sell and the respective expected selling price and profit for each house, can be extracted by changing the house selection through the filters on the sidebar

Region Overview

Fig 2.40 Recommendation Houses Table



House Id

Choose an option

Enter zipcode

Choose an option

Enter columns

id ✕ zipcode ✕ price ✕ selling_price ✕ profit_est ✕ best_season_sellin... ✕ best_season_profit... ✕ status ✕

Living Size

small_house ✕

Lot Size

Choose an option

Enter the zipcode to check renovation proposals:

98001

Region Overview

Recommended Houses Density Map

Top 10 Houses

With selling price suggestion increased based on:

Invested value above region median price value, house condition above 3 and best season to sell

	id	zipcode	price	selling_price
0	3025300225	98039	1,450,000.0000	1,812,500.0000
1	3585901025	98177	1,735,000.0000	2,082,000.0000
2	1118001645	98112	1,400,000.0000	1,750,000.0000
3	2470200020	98039	1,880,000.0000	2,115,000.0000
4	251500330	98004	1,775,000.0000	1,996,875.0000
5	3625710080	98040	1,025,000.0000	1,281,250.0000
6	5450900010	98040	993,500.0000	1,241,875.0000
7	3472800065	98004	1,698,000.0000	1,910,250.0000
8	1024000050	98116	915,000.0000	1,143,750.0000
9	6448000090	98004	1,575,000.0000	1,771,875.0000

Fig 2.41 Region overview

A.2 Publication

Joel Nithish Kumar M, Karthikeyan A, Darin Joshua D and Gladwin Joseph Solomon B. A Decentralized Application (DApp) on Trading Real Estate Assets using Blockchain and Artificial Intelligence, May 10, 2022, Journal Of Algebraic Statistics (JOAS), Indexed in the Emerging Sources Citation Index - Web of Science (Clarivate Analytics), Volume 13, No. 1, 2022, p. 216-224 https://publishoa.com ISSN: 1309-3452 216.

REFERENCES

[1] A. Jeppsson and O. Olsson, Jun. 2017. ‘‘Blockchains as a solution for traceability and transparency,’’ Packag. Logistics, Lund Univ., Lund, Sweden, Student Paper, Tech. Rep.,

[2] A. Narayanan, J. Bonneau and E. Felten, Bitcoin and Cryptocurrency Technologies: A comprehensive Introduction. 1st ed. Princeton, New Jersey: Princeton University Press, 2016

[3] Alisha Kuvalekar, Shivani Manchewar, Sidhika Mahadik, Shila Jawale. Posted: 3 Apr 2020. House Price Forecasting Using Machine Learning, Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3565512

[4] Ch. Rupa, Divya Midhunchakkaravarthy, Mohammad Kamrul Hasan, Hesham Alhumyani 4, Rashid A. Saeed, 2021. Industry 5.0: Ethereum blockchain technology based DApp smart contract.

[5] G. Naga Satish, Ch. V. Raghavendran, M.D.Sugnana Rao, Ch.Srinivasulu. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-9, July 2019. House Price Prediction Using Machine Learning. https://www.ijitee.org/wp-content/uploads/papers/v8i9/I7849078919.pdf

[6] Gharahighehi, A.; Pliakos, K.; Vens, Appl. Sci. 2021, 11, 7502. Recommender Systems in the Real Estate Market—A Survey. https://doi.org/10.3390/app11167502

[7] IBM, Published: 2018. Conversational AI - Watson Assistant: The Shortcut to Great Customer Service.

[8] Jennifer Conway, 2010, University of California, Berkeley. Artificial Intelligence and Machine Learning: Current Applications in Real Estate https://dspace.mit.edu/bitstream/handle/1721.1/120609/1088413444-MIT.pdf

[9] K. Christidis and M. Devetsikiotis, vol. 4, 2016. ‘‘Blockchains and smart contracts for the Internet of Things,’’ IEEE Access.

[10] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, ‘‘Blockchain,’’ Bus. Inf. Syst. Eng., vol. 59, no. 3, pp. 183–187, Mar. 2017.

[11] 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

[12] Paul Tostevin, September 2021. The total value of global real estate. https://www.savills.com/impacts/market-trends/

[13] 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

[14] R. Manjula, Shubham Jain, Sharad Srivastava and Pranav Rajiv Kher, Real estate value prediction using multivariate regression models, Materials Science and Engineering Conference Series, volume 263, issue 4, Posted: 2017
https://iopscience.iop.org/article/10.1088/1757-899X/263/4/042098

[15] Rajalakshmi Krishnamurthi, A Brief Analysis of Blockchain Algorithms and Its Challenges.

[16] S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[17] S. Raval, O’Reilly Media, 2016. Decentralized Applications: Harnessing Bitcoin’s Blockchain Technology, 1st ed. Newton, MA, USA.

[18] Shafaq Naheed Khan, Faiza Loukil, Chirine Ghedira-Guegan, Elhadj Benkhelifa &Anoud Bani-Hani. Peer-to-Peer Networking and Applications volume 14, pages2901–2925 (2021). Blockchain smart contracts: Applications, challenges, and future trends

[19] Sudeep Tanwar , Qasim Bhatia , Pruthvi Patel , Aparna Kumari , Pradeep Kumar Singh , Wei-chiang Hong , Senior member, IEEE. Machine Learning adoption in Blockchain-based Smart Applications: The challenges, and a way forward.

[20] Wei Cai, Zehua Wang, Jason B. Ernst, Zhen Hong, Chen Feng, Victor C. M. Leung (Members and Fellows, IEEE), Digital Object Identifier 10.1109/ACCESS.2018.2870644, Decentralized Applications: The Blockchain-Empowered Software System.

[21]Yugesh Verma, October 2021, How Machine Learning can be used with Blockchain Technology. https://analyticsindiamag.com/how-machine-learning-can-be-used-with-blockchain-technology/