**Land Registry System Using Blockchain**

Project Report

*Submitted in partial fulfilment of the requirement for the award of the degree of*

**Bachelor of Technology**

**(Information Technology)**

to

****

DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

***By***

DHEERAJ KUMAR (1873613018)

SANGAM (1873613048)

TANYA SRIVASTAVA (1907360139001)

Under the Guidance of

**Mr. Ashok Kumar Yadav**

**(Assistant Professor)**

****

*DEPARTMENT OF INFORMATION TECHNOLOGY*

*RAJKIYA ENGINEERING COLLEGE*

**AZAMGARH-276201-(U.P.)-INDIA**

JUNE- 2022

**Ⓒ RAJKIYA ENGINEERING COLLEGE,**

**AZAMGARH -276201, 2022 ALL RIGHTS RESERVED**

**Plagiarism Report**

We declare that this written project report tested on plagiarism checking software. The plagiarism software showing less than 20% similarity index after excluding cited contents.

Dheeraj Kumar Sangam Tanya Srivastava

(1873613018) (1873613048) (1907360139001)

**Department of Information Technology**

**Rajkiya Engineering College**

**Azamgarh – 276201 U.P**

**CANDIDATE’S DECLARATION**

We declare that this written submission represents our work and ideas in our own words and where other’s ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the University/Institution and can also evoke penal action from the sources which have thus not been properly cited or from whom paper permission has not been taken when needed.

**Dheeraj Kumar**

**(Roll no. 1873613018)**

**Sangam**

**(Roll no. 1873613048)**

**Tanya Srivastava**

**(Roll no. 1907360139001)**

Date:

Place: REC Azamgarh

**Department of Information Technology**

**Rajkiya Engineering College**

**Azamgarh – 276201 U.P**

**CERTIFICATE**

This is to certify that this project report entitled **“Land Registry System using Blockchain”** submitted by **Mr. Dheeraj Kumar, Mr. Sangam** and **Miss. Tanya Srivastava** to the **Rajkiya Engineering College, Azamgarh affiliated to Dr. A.P.J. Abdul Kalam Technical University**, for the award of the degree of **Bachelor of Technology in Information Technology**, is a project work carried out by him under the supervision of **Mr. Ashok Kumar Yadav, Asst. Prof. Department of Information Technology, Rajkiya Engineering College Azamgarh** To the best of my knowledge this work has not been submitted in part or full to any other University or Institution for the award of any degree or diploma.

**Mr. Ashok Kumar Yadav Dr. Tauseef Ahmad External Examiner**

(Project Supervisor) (HOD-ITD)

**ACKNOWLEDGMENT**

It is my great pleasure to express my gratitude to all the people who have supported me greatly during the pursuit of my B.Tech degree. Without their encouragement and useful help, it would not have been possible for me to complete my project. First of all, I would like to give my special and sincere thanks to my project advisor, **Mr. Ashok Kumar Yadav,** who has been devoted to his valuable guidance and encouragement, help, and useful suggestions throughout in my project work. His untiring and diligent effort, methodical approach, and individual help make it possible for me to complete this work on time. I consider myself very lucky for having been associated with the scholar like him. His guidance, affection, and scientific approach served a valuable incentive for completing this work. I would like to extend my appreciation to **Mr. Ashok Kumar Yadav**, Rajkiya Engineering College, Azamgarh for motivating me towards “Land Registration using Blockchain”, from the perspective of research and development by providing his valuable suggestions.

I shall ever remain obliged to the faculty members and staff of Rajkiya Engineering College, Azamgarh for their cooperation, kindness extended towards me during the completion of this project work.

This acknowledgement will remain incomplete, if I fail to express my deep sense of obligation to my parents for their consistent blessing and encouragement. I am deeply obliged to my father and maternal uncle, whose patience, tolerance, encouragement, consistent, pleasing and love has been continuously giving me the braveness to face challenges.

**Dheeraj Kumar**

**(Roll no. 1873613018)**

**Sangam**

**(Roll no. 1873613048)**

**Tanya Srivastava**

**(Roll no. 1907360139001)**

**ABSTRACT**

The existing set-up for land registration is very slow, inconvenient and involves a lot of vulnerabilities & paper works that give rise to disputes and corruption. It also involves a lot of intermediaries like property dealers to put trust in land registration system which leads to additional expenses. This paper discusses about proposed secure land registry system, based on ethereum blockchain that has great potential for storing all the transactions that have taken place during the process of land ownership transfer in a secure and immutable way. It can automatically trigger various events like registration of buyers and sellers, addition of land, approval of requested land and transfer of fund etc.by using the smart contract concept of blockchain technology. The decentralized, immutable and transparent nature of Blockchain technology makes the process of land registration secure, cost-effective, reliable, flexible and reduce the examples of deceit in the channel to a great extent and makes it possible to trace the transfer of land ownership. It will also be beneficial for entities involved in land registration process that is buyer, seller and land inspector to hand over the land possession from the seller to fresh buyer.

**List of Figure**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Caption** | **Page no.** |
| Figure 1.1 | Centralized, Decentralized, Distributed…………………………………... | ……02 |
| Figure 1.2 | Addition of new block to existing chain………………………………...... | ……03 |
| Figure 2.1 | Hash Function…………………………………………………………...... | ……10 |
| Figure 2.2 | Elliptic Curve Cryptography……………………………………………… | ……12 |
| Figure 2.3 | Merkle tree………………………………………………………………... | ……14 |
| Figure 2.4 | Proof of work……………………………………………………………... | ……16 |
| Figure 3.1 | Block Structure…………………………………………………………… | ……21 |
| Figure 3.2 | Blockchain Functionality…………………………………………………. | ……22 |
| Figure 4.1 | Bitcoin ………………………………………………………………......... | ……24 |
| Figure 4.2 | Ethereum …………………………………………………………………. | ……26 |
| Figure 4.3 | Artifacts generated in compilation process on Remix …………………… | ……27 |
| Figure 4.4 | Hyperledger ………………………………………………………………. | ……28 |
| Figure 4.5 | Solana …………………………………………………………………….. | ……29 |
| Figure 6.1 | Land Registration using blockchain ……………………………………… | ……35 |
| Figure 6.2 | Working flowchart of land registration system with Blockchain ………... | ……36 |
| Figure 7.1 | Inter Planetary File System ………………………………………………. | ……38 |
| Figure 7.2 | Ganache …………………………………………………………………... | ……40 |
| Figure 8.1 | Node Modules Installation ……………………………………………….. | ……45 |
| Figure 8.2 | Ganache …………………………………………………………………... | ……46 |
| Figure 8.3 | Ganache Auto-mining configuration …………………………………….. | ……47 |
| Figure 8.4 | Contract compilation using Truffle ………………………………………. | ……49 |
| Figure 8.5 | Contract Deployment using truffle ………………………………………. | ……50 |
| Figure 8.6 | The running DApps ………………………………………………………. | ……51 |
| Figure 8.7 | Contract owner dashboard ……………………………………………….. | ……52 |
| Figure 8.8 | Land inspector dashboard ………………………………………………... | ……52 |
| Figure 8.9 | User Registration…………………………………………………………. | ……53 |
| Figure 8.10 | User verification ………………………………………………………….. | ……53 |
| Figure 8.11 | Add land by user …………………………………………………………. | ……54 |
| Figure 8.12 | Land verification by Land Inspector ……………………………………... | ……54 |
| Figure 8.13 | Transfer Ownership by land inspector …………………………………… | ……55 |
| Figure 8.14 | Transferring ownership by to the buyer ………………………………….. | ……55 |
| Figure 8.15 | Successfully transferred ownership to the buyer ………………………… | ……56 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Name** | **Page no.** |
| Table 1.1 | Comparison of different blockchains …………………………………… | …...06 |
| Table 5.1 | Literature Review 1 …………………………………………………....... | …...32 |
| Table 5.1 | Literature Review 2 …………………………………………………....... | .…..33 |
| Table 5.1 | Literature Review 3 …………………………………………………....... | .…..34 |

**List of Abbreviations**

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **Meaning** | **Page no.** |
| RPC | Remote Procedure Call | 2 |
| HTTPS | Hyper Text Transfer Protocol Secure | 2 |
| HTTP | Hyper Text Transfer Protocol | 2 |
| EVM | Ethereum Virtual Machine | 6 |
| ECDSA | Elliptic Curve Digital Signature Algorithm | 9 |
| RSA | Rivest Shamir Adleman | 9 |
| BLS | Boneh LynnShacham | 10 |
| MD | Message Digest | 11 |
| SHA | Secure Hash Function | 11 |
| RIPEMD | RACE Integrity Primitive Evaluation Message Digest | 11 |
| ECC | Elliptic Curve Cryptography | 12 |
| PoW | Proof of Work | 15 |
| POS | Proof of Stake | 15 |
| PBFT | Practical byzantine fault tolerance | 17 |
| P2P | Peer to Peer | 21 |
| DApps | Decentralized Application | 26 |
| ICO | Initial Coin Offering | 26 |
| DLT | Distributed Ledger technology | 27 |
| PoH | Proof of History | 29 |
| PoC | Proof of Concept | 30 |
| IPFS | Inter Planetary File System | 31 |
| LIS | Land Information System | 32 |
| SARB | South African Reserve Bank | 32 |
| DAG | Direct Acyclic Graph | 38 |
| DHT | Distributed Hash Tables | 38 |
| UI | User Interface | 39 |
| CLI | Command Line | 39 |
| IDE | Integrated Development Environment | 40 |
| E-ADT | Eclipse Android Development Tools | 40 |
| API | Application binary interface | 42 |
| SDK | Software Development Kit | 47 |
| NFT | Non-fungible Tokens | 57 |

**Table of Contents**

**Declaration……………………………………………………………….........................i**

**Certificate……………………………………………………………………………..…ii**

**Acknowledgement.………………………………………………………….…………...iii**

**Abstract………………………………………………………………………………......iv**

**List of Figures …………………………………………………………………………....v**

**List of Tables …………………………………………………………………………….vi**

**List of Abbreviation …………………………………………………………………….vii**

|  |  |
| --- | --- |
| * 1. Centralized, Distributed & Decentralized System ………………...   2. What is blockchain ………………………………………………...   3. Types of blockchain ……………………………………………….      1. Public blockchain …………………………………………      2. Private blockchain ………………………………………...      3. Consortium blockchain ……………………………………   4. Comparison of Different types of blockchain ……………………..   5. Problems in Current Land registry System ……………………….. | …………1  …………3  …………3  …………4  …………4  …………4  …………5  …………6 |

**CHAPTER 1. Introduction ………………………..……………….....….1-7**

**CHAPTER 2. Components of blockchain ……………………………...8-18**

|  |  |
| --- | --- |
| 2.1 Public key cryptography ………………………………………………....  2.2 Digital Signature …………………………………………………………  2.2.1 Definition ……………………………………………………………  2.2.2 Digital signature serve three purpose ………………………………..  2.2.3 Data value used to create a digital signature ………………………...  2.2.4 Digital Signature as a component in blockchain …………………….  2.2.5 Blockchain has a commonly used signature scheme ………………..  2.3 Hash Function ……………………………………………………………  2.3.1 Properties of hash function ………………………………………….  2.3.2 Popular hash functions ………………………………………………  2.4 Elliptic curve cryptography ………………………………………………  2.5 Merkle tree ……………………………………………………………….  2.5.1 Implementation of Merkle tree ……………………………………...  2.6 Consensus algorithm ……………………………………………………..  2.6.1 Proof of Work (PoW)………………………………………………...  2.6.2 Proof of Stake ………………………………………………………..  2.6.3 Practical byzantine fault tolerance (PBFT) …………………………. | ……...8  ……...8  ……...8  ……...9  ……...9  ……...9  ……...9  ……..10  ….….10  ….….11  ….….12  ……..13  ….….14  …….15  …….15  …….17  …….17 |

**CHAPTER 3. Architecture and features of blockchain…………........19-23**

|  |  |
| --- | --- |
| 3.1 Blockchain architecture ………………………………………………............  3.2 Process of blockchain …………………………………………………………  3.3 features of blockchain ………………………………………………………… | ..19  ....21  ..22 |

**CHAPTER 4. Blockchain Development Platforms………………..….24-29**

|  |  |
| --- | --- |
| 4.1 Introduction to Bitcoin ………………………………………………………...  4.2 Introduction to Ethereum ………………………………………………………  4.3 Introduction to Hyperledger …………………………………………………...  4.4 Introduction to Solana …………………………………………………………  4.5 Introduction to Near Blockchain ……………………………………………… | .24  .25  .27  .28  .29 |

**CHAPTER 5. Literature Review………………………………………30-34**

|  |  |
| --- | --- |
| 5.1 Related Works …………………………………………………………………..…  5.2 Related Papers ………………………………………………………………… | .30  .32 |

**CHAPTER 6. Proposed Methodology………………………………...35-37**

|  |  |
| --- | --- |
| 6.1 Proposed Methodology ……………………………………………………...... | ...35 |

**CHAPTER 7. Tools and Technologies………...………….……...…...38-42**

|  |  |
| --- | --- |
| 7.1 IPFS…………………………………………………………………………  7.2 Blockchain platform: Ethereum…………………………………………….  7.3 Ganache……………………………………………………………………..  7.4 Metamask……………………………………………………………….……  7.5 Android Studio………………………………………………………………..  7.6 Flutter framework……………………………………………………………  7.6.1 Web3Dart language………………………………………………….... | ….38  ….39  ….39  ….40  ….40  ….41  ….42 |

**CHAPTER 8. Implementation of Project……..……………………...43-56**

|  |  |
| --- | --- |
| 8.1 Smart Contract……………………………………………………………….  8.2 Installations…………………………………………………………………..  8.3 Configuration…………………………………………………………………  8.4 Deploying smart contract with Truffle……………………………………….  8.5 Running the DApp…………………………………………………………… | ..43  ..45  46  47  51 |

**CHAPTER 9. Conclusion and future work…...………………….......57**

**References ……..…………………………………………………………………....58**

**Chapter 1**

**Introduction**

Blockchain is constantly growing decentralized and immutable ledger that keeps permanent & encrypted records of all the transactions across a peer-to-peer network. All transactions stored in the blockchain are immutable; once a piece of data stored into a blockchain, it is extremely difficult to alter its value (because it facilitates append only feature). There are various platforms which facilitates the creation of decentralized, reliable, immutable, transparent, and secure blockchain-based land registration system. Now, we will see some basic concepts of Blockchain Technology.

* 1. **Centralized, distributed and decentralized systems**

Centralized systems refer to the system that *uses client/server architecture i.e.,* association of a few nodes to a central server. Majority of tasks of such systems are reliant on a central server/master node which results in loss of information, time and control if that central point goes down. Currently, the majority of the services is running under a centralized architecture (for example the records of property is stored on a central server), however such frameworks have some vulnerabilities regarding some parameters which are discussed below-

1. **Single Point of failure:** Centralized system has a significant problem that single point of failure. if the central server fails, the entire network will likely shut down. Because all nodes rely on the central/main server. This failure may cause the loss of crucial information, time and money which result in service interruptions and inconvenience.
2. **Cost:** The significant impact here is that centralized system is more costly because it requires very high cost to setup the entire server which maintains the proper communication (i.e., request and reply) among nodes connected in that system.
3. **Disk Space:** In such a centralized system, once the central server got compromised, we may loss critical data. So, we try to maintain the storage of data at multiple places or to rent a cloud from any cloud service provider to provide backup & avoid sudden data loss. It may take several terabytes of storage disks to maintain data in such a situation.
4. **Time:** Due to network traffic, it takes additional time to exchange data, and some of the time it produces the issue like ’server down’.
5. **Scalability and Reliability:** Centralized system can scale its resources and availability very rapidly. However, it is directly highly dependent on adding more storage, network bandwidth, or processing power etc to the central server. Therefore, it leads to scalability limitations. It seems to be significantly less reliable due to the reason that all the data are stored in a centralized database. It’s more prone to hacks and information leaks.
6. **Difficult server maintenance:** In such framework, there is just one server node, and taking the server down for maintenance is very inefficient. So, the server maintenance is complex.

All these parameters and points of discussion leads to an evolved system, i.e., distributed and decentralized systems.

**Distributed System:** In a distributed system, the nodes are located in different physical locations. That’s why It is highly secure. Various nodes can communicate and coordinate through message passing techniques (i.e., RPC, HTTP, HTTPS etc) i.e., they can send and receive messages to and from each other. It ensures that every node in the network has a replica of the blockchain and that it is getting update in real-time when a transaction happens. Even if one node fails in the network, it will not take down the entire system. Using Distributed Computing along with the peer-to-peer architecture, it maintains concurrency and also eliminates the need of central authority to govern the network thus making the blockchain system cost-effective, transparent, scalable, secure, and immutable.

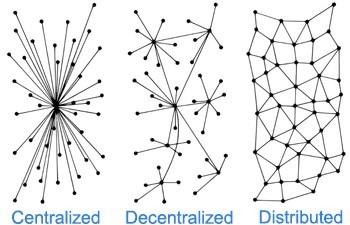
**Decentralized Systems:** Blockchain technology brings decentralization. The main disadvantages of the centralized system are “single point failure” which can be addressed by decentralization. Decentralized system stores data in distributed manner. It offers several advantages like transparency, security, trust etc.

Figure 1.1: Centralized, Decentralized, Distributed.

In centralized system, single points of failure can halt entire System, Decentralized system addressed this issue, means if a node fails due to some reason or goes offline passes their workload to the other nodes in network, and this type of healing power of such system increases the system reliability.

In most of the cases, the system charges some opportunity cost to govern the system's functioning in a centralised environment; however, there is no central authority to pay that opportunity cost in decentralised and distributed systems, which makes system transparent and cost-effective. A smooth functioning can be maintained simply by providing incentives to the nodes involved in same process as in centralized systems.

**1.2 What is Blockchain**

The term ‘Blockchain’ was first referenced in the white paper of Satoshi Nakamoto in 2008. Blockchain can be characterized as the constantly growing ledger of blocks that keeps immutable and permanent records of all the transactions that have taken place. It provides an intelligent way of the utilization of the decentralized, distributed system, public ledger, hash cash, consensus mechanism, merkle tree, cryptography and several hashing techniques for ensuring security, privacy and trust in the P2P network.

“Blockchain Technology can be defined as a platform where people come together to share their thoughts collaboratively to make rational decision making to achieve destined goal in trustless environment”. It forms a chain of blocks by storing hash values of the previous block in each block and a block refers to collection of transactions. Since a block is made up in 10 minutes and before its size was around 1 MB but later it can be extended to 10 MB.

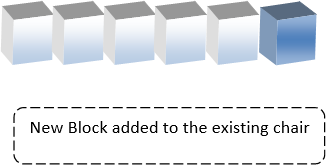
As blockchain technology is distributed and decentralized in nature. So, there is no central authority to govern the system. Here Blockchain system relies on consensus algorithm which ensures a common agreement among nodes in a P2P network. It provides trust in trustless environment and prevents from double spending problem and bring higher security to the blockchain system. It ensures fault tolerance and reliability in distributed system.

Figure 1.2: Addition of new block to existing chain

* 1. **Types of blockchain**

Blockchain is mainly categorized into the three subcategories-

1. Public Blockchain
2. Private Blockchain
3. Consortium Blockchain
   * 1. **Public blockchain**

A public blockchain also referred as permissionless blockchain where any individual is allowed to join the network, see the ledger and take part in the consensus process thus it maintains transparency. It doesn’t have any restrictions and allows all nodes of the network to have equal rights to access the blockchain data, create new blocks of data, and validate blocks of data. A public blockchain is decentralized because there is no central authority which controls the network. It is fully immutable in nature, it means that once a block gets added on the chain, it’s almost impossible to alter it or delete it. It offers append only feature. The public network operates on an incentive and reputation mechanism that encourages new participants to join and actively contribute into network to keep network alive.

* + 1. **Private blockchain**

Private Blockchain comes under the permissioned model that is managed by a network administrator and participants need consent to join the network There are very high security and integrity of the data in the permissioned blockchain. Private blockchains speed up the transaction. it can process thousands of transactions every second and are easily scalable. It controls and manage that ensures only authorized user can participate in the network, execute the consensus protocol that defines the mining rights and rewards, and maintains the distributed shared ledger. Every node connected in that private network is provided with login credentials so that user can easily login to the network and participate in the mining process. It is comparatively easy to track the mining record of the particular node in the system. Private blockchain can be also attacked by its miners when more than half are agreed to act maliciously in the network. Private blockchain model is used for financial institutions, medical industry etc.

* + 1. **Consortium blockchain**

Consortium blockchain can be defined as the semi-decentralized model of one or more organizations, it has private and public blockchain features. In this blockchain, the consensus processes are controlled by predefined nodes. It has a validator node that initiates, receives and validates transactions. Member nodes can receive or initiate transactions. It tends to be highly secure, scalable and efficient than a permissionless blockchain network.

* 1. **Comparison of Different Types of Blockchain**

Depending on various use cases and applications, blockchain technology is mainly categorized into three types: private blockchain, public blockchain, and consortium blockchain. A public blockchain is open to all. A private blockchain is only open to only approved member of one organization. Consortium blockchain can be said to be hybrid form of public and private blockchain. All these types of blockchain have their own specific applications and each of these have their own pros and cons. The detailed comparison of all the three types of blockchain are shown in below table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Public** | **Private** | **Consortium** |
| **Consensus** **Determinat**i**on** | All member | Approved member of one organization | Approved member of multiple organizations |
| **Centralizat**i**on** | Fully decentralised | Fully decentralised | Partially Decentralised |
| I**mmutabili**t**y** | Impossible to temper the public ledger | May be tempered if majority of node want temper | May be tempered if majority of node want temper |
| **Read** **Access** | anyone | Only Authenticated users of one organization | Only Authenticated users of multiple organization |
| **Write** **Access** | Anyone | Authenticated users only | Authenticated users only |
| **Network** s**ize** | Large (10000) | Small (10-100) | Small (10- 100) |
| **Network** **synchrony** | Asynchronous / partially synchronous | Partially synchronous / synchronous | Partially synchronous / synchronous |
| **Node** **Identity** | Pseudonymous | Revealed | Revealed |
| **Transparency** | Open | Closed | Open |
| **Network** **synchrony** | Asynchronous / partially synchronous | Partially synchronous / synchronous | Partially synchronous / synchronous |
| **Network** **connectivity** | Low | High (fully connected) | High (fully connected) |
| **Consensus** | Permissionless | Permissioned | permissioned |
| **Cost** | Cheap | Costly | Costly |
| **Efficiency** | Very low | Very high | High |
| **Ownership** | Nobody | Single entity | Group of entities |
| **Application** **examples** | Cryptocurrency, smart contract, public record. | Business contract, supply chain. | Inter-bank clearing, supply chain |

Table 1.1: Comparison of different blockchains

* 1. **Problems in current Land Registry System**

There are many problems in land registry system which are listed below-

* Time consuming
* Lack of transparency
* Less secure
* Forgery
* Costly
* Error prone

There are various platforms which facilitates the creation of decentralized, reliable, immutable, transparent, and secure blockchain-based land registration system. Among these systems, Smart contract based on Ethereum blockchain are gaining traction Ethereum is a decentralized, public blockchain platform that allows anyone to join and participate in the blockchain ecosystem. Smart Contracts can be deployed on the Ethereum Virtual Machine (EVM). Once deployed it can be accessed globally due to its transparent nature. These contracts verified by the parties that we call miners. Miners are multiple computers who validates new blockchain transactions and records them on the blockchain by solving a complex mathematical problem based on cryptographic hash algorithm. Miners are getting paid with something called Gas, refers to the cost to run a smart contract. To perform a transaction on the network i.e., execute a function of smart contract, or transfer amount to another account, you must pay some ether that gets converted into gas. Though blockchain technology is especially popular in finance because it came to existence with Bitcoin cryptocurrency still its implementation is not only limited to cryptocurrencies. It’s impacting a variety of sectors and playing an essential role in each part of day-to-day life.

Land Registry is also one of the use cases of this emerging technology. The transparent nature of Blockchain can make it possible to track who owns which pieces of the property. The traditional system for land registration is very slow, cumbersome, full of duplicacy and inefficiencies which raises major issues about data fraud, security of sensitive data, and the risk of centralized system failure due to natural disasters. Almost 66% court cases in our country are related to land conflicts spending a hefty Rs.58,000 crore in litigation. The land dispute between the claiming owner and the legal owner; the claiming owner can file a case against the legitimate owner to win the ownership of that property/asset. In that case, the judiciary labelled the property as disputed property or disputed land. That cannot be sold or bought by anyone who claims to be an actual owner. In this case, the judiciary takes seizes the property till he decides who is the legal owner of that property which makes the inter-process difficult. Land Registration system using blockchain technology offers distributed and decentralized environment that will keep records of all the transactions made during the process of land registry. It will be helpful for all the entities involved in land registration process as well as it will also speed up the process of land registration. Blockchain made it easier to trace who is the actual owner of a particular assets and also at what time any particular transaction took place.

**Here is the list of few countries planning to introduce blockchain technology in land registry system to improve the land registry and transfer processes.**

1. United States
2. Ukraine
3. Netherlands
4. Brazil
5. Sweden
6. United Kingdom

**Chapter 2**

**Components of blockchain**

The unique properties of blockchain are only because of the unique features which it has. The features like public key cryptography, data structures, hashing algorithms, digital signature, decentralized peer-to-peer network etc. In this chapter, we will explore each of these features in detail.

**2.1 Public key cryptography**

Public key cryptography (also called “asymmetric cryptography”) is a core part of modern -day information security. Public key cryptography uses unique keys to secure information. we use public key cryptography to crate the public -private key pair. they are considered a “pair” because the public key is derived from the private key. the private key controls access by being the unique piece of information needed to create digital signatures, which are required to sign transactions to spend any funds in the account. keys are created with the assistance of cryptographic calculations which depends on the numerical issues to deliver one way works. A single direction work is the capacity which can’t be turned around. The public key cryptography is different from the symmetric cryptography where just a solitary key is utilized for encryption just as decoding.

The main business applications for public-key cryptography are:

1. **Digital signatures:** Message is digitally signed with an individual’s private key and is verified by the individual’s public key.
2. **Encryption:** Message is encrypted using a recipient’s public key and can only be decrypted with the recipient’s private key

**2.2 Digital Signature**

**2.2.1 Definition:**

Digital signatures are similar to handwritten signatures. This includes electronic verification of the sender and digital signatures of the type that encrypt documents with digital codes that are particularly difficult to identical. Digital Signature is widely used in software distributions, financial transactions, and other cases where it is important to identify counterfeiting or tampering. Digital signatures are very famous with email users. The digital signature acts as a powerful tool and is now recognized as legally binding in many countries. These can be used as part of contract and notary document authentication, personal and corporate authentication, and more complex protocols. Digital signatures also allow the secure distribution and transmission of public keys, and thus serve in the truest sense of the word as the basis of all public key cryptography.

**2.2.2 Digital signatures serve three purposes**

1. **Authentication:** The digital signature gives the receiver a reason to believe that the message was created and sent by the declared sender.
2. **Non-repudiation:** With a digital signature, the sender cannot deny that the message was sent later.
3. **Integrity:** Digital signatures ensure that the message has not been modified in transit.

The digital signature contains the sender's private key, which the signer keeps safe. He or she converts the message into ciphertext and signs the document, a hash algorithm is used, and the recipient decrypts the message using the sender's public key for digitally signed data using a hash function, convert it to a hash value, encrypt the hash with the sender's private key, add a certificate to the signature, and finally create the digitally signed data.

**2.2.3 Data value used to create a digital signature**

**Hash function:** Converts data into fixed and compressed values called hashes.

**Private key:** A private key or variable used for encryption.

**Certificate:** Sender Indicates the ID to which the message belongs to the sender and is bound to the sender's or personal's public key.

**2.2.4 Digital Signature as a component in Blockchain**

It is rare to forge a digital signature because it uses mathematical theory to guarantee functionality and uses a system called public key cryptography to allow users to build pairs using both public and private keys. The public key can be thought of as the owner's ID, and the private key can be thought of as private information that can prove that the owner has the public key. Digital signatures are a fundamental part of the blockchain. These are primarily used to check the authenticity of transactions. When a user submits a transaction, all nodes in the system must demonstrate their right to use their funds while making them unavailable to other users. Each node on the network checks the conditions of the transaction sent and makes sure that all other nodes are functioning to agree to the correct state.

* + 1. **Blockchain has a commonly used signature scheme**

1. **ECDSA**

The Elliptic Curve Digital Signature Algorithm (ECDSA) is now using in Bitcoin. Compared to RSA, ECDSA uses shorter keys and requires less computationally intensive requests while maintaining a high level of security.

1. **SCHNORR SIGNATURES (Bitcoin)**

ECDSA does not have an efficient way to compress and verify signatures together. Therefore, there is a Schnarr signature to improve the scalability, efficiency and privacy of Bitcoin. Schnarr signatures are clearly secure and immutable using standard cryptographic assumptions (individual protocols) (third parties modify valid signatures to differ for the same key and message). Provides linearity (multiple parties can work together to create different valid signatures for the same key and message), create a single signature. This applies to all public keys.

1. **BLS SIGNATURES (ETH2)**

BLS (BonehLynnShacham) is used in the Proof of Stack of Ethereum Transitions. It is based on a type of encryption called pairing-based encryption. The ETH2 chain involves thousands of verifiers on every committee, and each committee needs to create thousands of signatures in a very short amount of time.

**2.3 Hash function**

A hash function is “any function that can be used to map data of arbitrary size to data of a fixed size”. The input to a hash function is called a pre-image, the message, or simply the input data. The output is called the hash. Cryptographic hash functions are a special subcategory that have specific properties that are useful to secure platforms, such as Ethereum. Values returned by a hash function are called message digest or simply hash values.

The following picture illustrated hash function: -

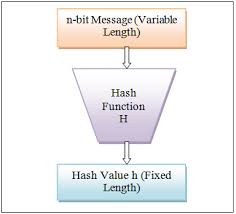


Figure 2.1: Hash Function.

**2.3.1 Properties of hash functions:**

In order to be an effective cryptographic tool, the hash function is desired to possess following properties-

1. **Pre-Image Resistance:** This property means that it should be computation-ally hard to reverse a hash function. In other words, if a hash function h produced a hash value z, then it should be a difficult process to find any input value x that hashes to z. This property protects against an attacker who only has a hash value and is trying to find the input.
2. **Second Pre-Image Resistance:** This property means given an input and its hash it should be hard to find a different input with the same hash. In other words, if a hash function h for an input x produces hash value h(x), then it should be difficult to find any other input value y such that h(y) = h(x). This property of hash function protects against an attacker who has an input value and its hash, and wants to substitute different values as a legitimate value in place of original input value.
3. **Collision Resistance:** This property means it should be hard to find two different inputs of any length that result in the same hash. This property is also referred to as collision free hash function. In other words, for a hash function h, it is hard to find any two different inputs x and y such that h(x) = h(y). Since, hash function is compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find. This property makes it very difficult for an attacker to find two input values with the same hash.
4. **Efficiency of Operation:** Generally, for any hash function h with input x, computation of h(x) is a fast operation. Computed hash functions are much faster than a symmetric encryption.
5. **Fixed Length Output (Hash Value):** Hash function coverts data of arbitrary length to a fixed length. This process is often referred to as hashing the data. In general, the hash is much smaller than the input data, hence hash functions are sometimes called compression functions. Since a hash is a smaller representation of a larger data, it is also referred to as a digest. Hash function with n bit output is referred to as an n-bit hash function. Popular hash functions generate values between 160 and 512 bits.

**2.3.2 Popular hash functions**

Let us briefly see some popular hash functions-

**Message Digest (MD):** Message digest is also known as hash code or simply hash which obtained with the help of a given input and a given hash function.

**Secure Hash Function (SHA):** Family of SHA comprise of four SHA algorithms; SHA-0, SHA-1, SHA-2, and SHA-3. Though from same family, they are structurally different.

**RIPEMD:** The RIPEMD is an acronym for RACE Integrity Primitives Evaluation Message Digest. This set of hash functions was designed by open research community and generally known as a family of European hash functions.

**2.4 Elliptic curve cryptography**

Elliptic Curve Cryptography (ECC) is an open key encryption method dependent on elliptic bends hypothesis that can be utilized to make quicker, smaller, and increasingly proficient cryptographic keys. It is Asymmetric Public key cryptosystem. It provides Equal security with smaller key size (as compared to RSA) as compared to Non ECC algorithms. i.e., small key size and high security. It makes use of Elliptic curves Elliptic curves are defined by some mathematical functions such as cubic function.

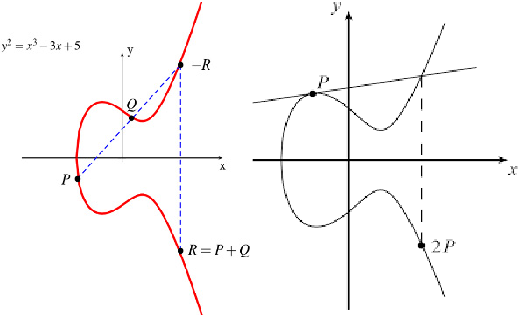
e.g.: y2 = x3 + ax + b (equation of degree 3)

Figure 2.2: Elliptic Curve Cryptography (ECC)

**Properties-**

1. Symmetric to x-axis.
2. If we draw a line, it will touch a Max of 3 points.

Let Eq (a, b) be the elliptic curve. consider the equation Q=k\*P Where Q, P 🡪 Points on curve and k<n, if k and P 🡪 given, if should be easy to find Q but if we know Q and P, it should be extremely difficult to find k. This is called the discrete logarithm problem for elliptic curves. It is one way function like Trapdoor function (a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information).

**ECC- ALGORITHM**

* **ECC-KEY EXCHANGE**

**Global Public Elements**

Eq (a, b): elliptic curve with parameters a, b and q(prime no. Or an integer of the form 2^m).

G: Points on the curve/elliptic curve whose order is large value of n.

**User A key Generation**

Select Private key nA : nA<n

Calculate public key PA :PA= nA\*G

**User B key Generation**

Select private key nB : nB<n

Calculate Public key PB : PB=nB\*G

**Calculation of secret key by user A**

kA= k=nA\*PB

**Calculation of secret key by user B**

**K=nB\*PA**

* **ECC ENCRYPTION**
* Let the message be M.
* First encode this message M into a point on elliptic curve.
* Let this point be Pm (now this point is encrypted).

For encryption, chose a random positive integer k

The Cipher point will be-

Cm={kG, Pm+ kPB} (for encryption public key of B used)

This point will be sent to the receiver.

* **ECC DECRYPTION**

For decryption, multiply 1st point in the pair with receiver’s secret key

i.e., kG\*nB (for decryption private key of B used)

Then Subtract it from 2nd point / coordinate in the pair

i.e., Pm + kPB - (kG\*nB), but we know PB = nB\*G

So,

= Pm + kPB - kPB

= PM (original point).

So, receiver gets the same point.

**2.5 Merkle tree**

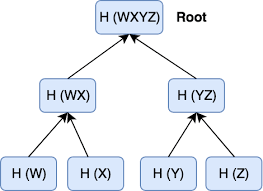
Merkle tree also known as hash tree is a data structure used for data verification and synchronization. It is a tree data structure where each non-leaf node is a hash of its child nodes. All the leaf nodes are at the same depth and are as far left as possible. It maintains data integrity and uses hash functions for this purpose. Merkle trees are a fundamental component of blockchains that underpin their functionality. They allow for efficient and secure verification of large data structures, and in the case of blockchains, potentially boundless data sets. The leaf nodes are the lowest tier of nodes in the tree. At first, it may sound difficult to comprehend, but if you look at the commonly used figure below, it will become much easier to understand.

Figure 2.3: Merkle Tree

The example above is the most common and simplest form of a Merkle tree known as a Binary Merkle Tree. As you can see, there is a top hash that is the hash of the entire tree, known as the root hash. Essentially, Merkle trees are a data structure that can take n number of hashes and represent it with a single hash. The structure of the tree allows for efficient mapping of arbitrarily large amounts of data and enables easy identification of where changes in that data occur. This concept enables Merkle proofs, with which, someone can verify that the hashing of data is consistent all the way up the tree and in the correct position without having to actually look at the entire set of hashes. Instead, they can verify that a data chunk is consistent with the root hash by only checking a small subset of the hashes rather than the entire data set. As long as the root hash is publicly known and trusted, it is possible for anyone who wants to do a key-value lookup in a database to use a Merkle proof to verify the position and integrity of a piece of data within a database that has a particular root. When the root hash is available, the hash tree can be received from any non-trusted source and one branch of the tree can be downloaded at a time with immediate verification of data integrity, even if the whole tree is not yet available. One of the most important benefits of the Merkle tree structure is the ability to authenticate arbitrarily large sets of data through a similar hashing mechanism that is used to verify much smaller amounts of data. The tree is advantageous for distributing large sets of data into manageable smaller parts where the barrier for the verification of integrity is substantially reduced despite the overall larger data size. The root hash can be used as the fingerprint for an entire data set, including an entire database or representing the entire state of a blockchain.

**2.5.1 Implementation of merkle trees**

Bitcoin was the first cryptocurrency that employed Merkle trees effectively. To ensure that the hash values are protected and cannot be reversed easily, it utilizes the famous Secure Hashing Algorithm SHA-256. This also means that the hash values output is 256 bits long. At the core, Merkle trees are used to store data and also prune transactions. Ethereum blockchain also utilizes Merkle trees. However, the approach here is different than that of how bitcoin utilized it. In Ethereum, Merkle Patricia Tree is used which is a complex version of the Merkle tree. This is possible because Ethereum is Turing-complete.

Finally, Merkle trees are important component of distributed version control systems such as Git and IPFS. Their ability to easily ensure and verify the integrity of data shared between computers in a P2P format makes them invaluable to these systems. Merkle trees are an integral component of blockchains and effectively allow them to function with provable immutability and transaction integrity. Understanding the role that they play in distributed networks and their underlying technology of cryptographic hash functions is crucial to grasping the basic concepts within cryptocurrencies as they continue to develop into larger and more complex systems.

**2.6 Consensus algorithms**

A Consensus algorithm is a process in computer science used to achieve agreement on a single data value among distributed processes or systems. Consensus algorithms are designed to achieve reliability in a network involving multiple unreliable nodes, the blockchain consensus protocol consists of some specific objectives such as coming to an agreement, collaboration, co-operation, equal rights to every node, and mandatory participation of each node in the consensus process. A consensus mechanism refers to any number of methodologies used to achieve agreement, trust, and security across a decentralized computer network. In this way, an accord calculation targets finding a typical understanding that is a success for the whole system. Now, we will discuss various consensus algorithms and how they work.

**2.6.1 Proof of Work (PoW)**

The idea for Proof of Work (PoW) was first published in 1993 by Cynthia Dwork and Moni Naor and was later applied by Satoshi Nakamoto in the Bitcoin paper in 2008. cryptographic proof has different forms in which proof of work is one of them in which prover proves to verifiers that a certain amount of specific computational effort has been expanded. Verifiers have authority to confirm this expenditure subsequently with less effort. The term “proof of work” was first used by Markus Jakobsson and Ari Juelsin a publication in 1999.

**Principle:** A consensus mechanism's objective is to bring all nodes into agreement, that is, to trust one another, in an environment where nodes do not trust one another. The new block's transactions are then validated, and the new block is put to the blockchain. The block with the greatest block height will be added to the chain. Miners (special machines on the network) conduct computing labour in order to solve a complicated mathematical problem in order to add the block to the network, thus the term Proof-of-Work. The mathematical issue grows increasingly complicated with time.

**HOW PoW WORK?**

According to an edX course, "the Proof of Work consensus algorithm requires solving a computationally difficult problem in order to produce new blocks on the Bitcoin network." The process is colloquially known as mining,' and the nodes in the network that participate in mining are known as miners. Mining transactions are motivated by economic payoffs, in which competing miners are rewarded with bitcoins and a tiny transaction fee." It doesn't require much vitality or time to confirm the exchanges in the square to be included, arrange these exchanges in a sequential request in the square, and declare the just mined square to the whole system. When a miner discovers the correct answer, the node publishes it. It is distributed to the whole network at the same time, with a cryptocurrency prize (the reward) supplied via the PoW algorithm. With more miners, the time it takes to mine the next block will inevitably shorten. This implies that new blocks are discovered more quickly. To ensure that one block is found every ten minutes, the Bitcoin network modifies the difficulty level of mining a new block on a regular basis. To comprehend PoW, we must first become acquainted with the network's miner nodes. Miner nodes are specialized nodes that are in charge of producing new blocks on the Blockchain. When the blocks are successfully created, these miner nodes are paid with tokens. The incentives guarantee that the blocks inside the network remain active. The miner node in PoW is determined via a computational issue. The nodes are given a mathematical challenge in which they must discover a value called nonce such that after hashing the nonce and the previous block data, the new hash created equals the specified hash. Every block in PoW is produced after 10 minutes. The core concept of PoW is a mathematical conundrum that is difficult to solve but simple to verify. The mining process in PoW demands a lot of computing power, therefore miner nodes are paid with tokens for using their processing power and putting in the effort to mine.

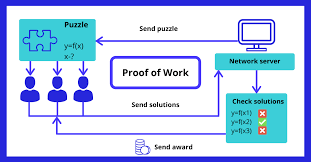


Fig 2.4: Proof of Work

**Common cryptographic protocols used in proof of work system:** The most extensively used proof-of-work consensus is SHA-256, which was developed as part of the Bitcoin. Scrypt, SHA-3, scrypt-jane, scrypt-n, and more algorithms are also available.

**Features of Proof of Work System:** The following are the primary factors that have led to the widespread success of this consensus protocol:

1. Finding a solution to the mathematical issue is difficult.

2. It is simple to check the solution's accuracy.

**Main issue with the Proof-of-Work consensus:** The Proof-of-Work consensus process has the following flaws:

1. **The 51 percent risk**: If a controlling entity possesses 51 percent or more of the network's nodes, the entity can damage the blockchain by obtaining control of the majority of the network.
2. **Time-consuming**: Miners must examine multiple nonce values in order to locate the correct answer to the problem that must be solved in order to mine the block, which is a time-consuming operation.
3. **Consumption of resources:** Miners use a lot of computational power to solve difficult mathematical puzzles. It results in the squandering of valuable resources (money, energy, space, hardware). Transaction confirmation takes between 10 and 60 minutes. As a result, it is not an immediate transaction; it takes time to mine the transaction and add it to the blockchain, so committing the transaction.

**2.6.2 Proof of Stake**

This is the most commonly used alternative to PoW. Ethereum has switched from PoW to PoS consensus. Instead, then investing in expensive technology to solve a complicated challenge, validators invest in the system's currency by locking up part of their coins as stake. After that, all validators will begin verifying the blocks. If validators uncover a block that they believe may be added to the chain, they will validate it by putting a wager on it. Based on the actual blocks uploaded to the Blockchain, all validators receive a payout proportionate to their bets, and their stake increases proportionately. Finally, depending on their economic stake in the network, a validator is picked to produce a new block. As a result, PoS encourages validators to establish a consensus through an incentive mechanism.

**2.6.3 Practical byzantine fault tolerance (PBFT)**

Practical Byzantine Fault Tolerance is a consensus method developed by Barbara Liskov and Miguel Castro in the late 1990s. PBFT was designed to perform well in asynchronous (no upper constraint on when the response to the request will be received) systems. It is designed to have a reduced overhead time. Its purpose was to overcome many of the difficulties associated with existing Byzantine Fault Tolerance methods. Distributed computing and blockchain are two examples of application areas. Byzantine Fault Tolerance (BFT) is a distributed network characteristic that allows it to establish consensus (agreement on the same value) even when some nodes in the network fail to reply or respond with inaccurate information. A BFT mechanism's goal is to protect against system failures by using collective decision making (both right and faulty nodes) to lessen the effect of faulty nodes. BFT is an abbreviation for the Byzantine Generals' Problem.

**Types of byzantine failures:**

There are two categories of failures that are considered. One is fail-stop (in which the node fails and stops operating) and another is arbitrary-node failure. Some of the arbitrary node failures are given below,

1. Failure to return a result
2. Respond with an incorrect result
3. Respond with a deliberately misleading result
4. Respond with a different result to different parts of the system.

**How PBFT works?**

PBFT attempts to provide a viable Byzantine state machine replication that can function even in the presence of malevolent nodes in the system. In a PBFT-enabled distributed system, nodes are sorted progressively, with one node serving as the primary (or leader node) and the rest serving as secondary (or the backup nodes). It should be noted that any eligible node in the system can become the primary by switching from secondary to primary (typically, in the case of a primary node failure). The objective is for all honest nodes to contribute to obtaining an agreement on the state of the system using the majority rule. A real-world Byzantine Fault Tolerant systems can work if the greatest number of malicious nodes is less than or equal to one-third of all nodes in the system. The system becomes more secure as the number of nodes grows.

**Limitations of PBFT:**

Due to the substantial communication cost that rises exponentially with each additional node in the network, the PBFT consensus model works efficiently only when the number of nodes in the distributed network is limited.

1. **Sybil attacks:** The PBFT systems are vulnerable to Sybil attacks, which occur when a single entity (party) controls several identities. Sybil attacks become increasingly harder to carry out as the number of nodes in the network grows. However, because PBFT processes have scalability concerns, they are employed in conjunction with other mechanisms.
2. **Scalability:** PBFT does not scale effectively because to the overhead of communication (with all the other nodes at every step). The time it takes to react to a request rises as the number of nodes in the network grows (increases as O(nk), where n is the number of messages and k is the number of nodes).

**Chapter 3**

**Architecture and features of blockchain**

In previous chapters, we have already discussed about its definition and the components of the blockchain. In this chapter, we will explain how those components are organized together to form a blockchain architecture and what are the features and characteristics which blockchain can provide us to implement our project.

* 1. **Blockchain architecture**

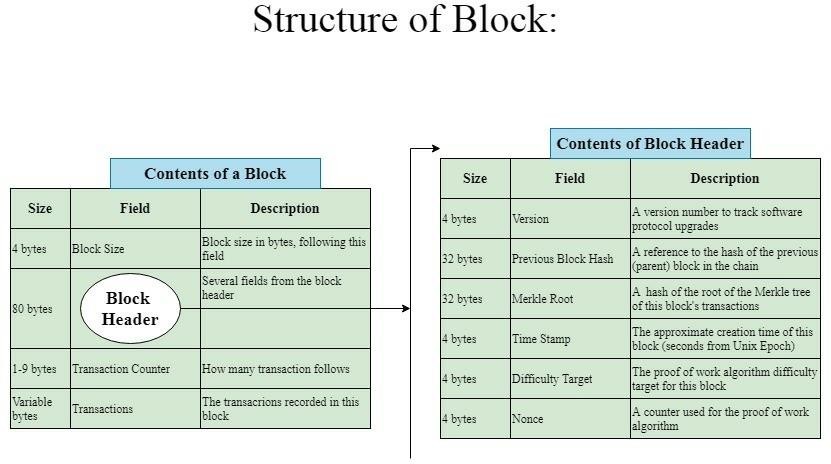
1. **Decentralization (Peer to Peer)** is the fundamental implication of Blockchain Technology. Decentralization can be considered as a platform where various nodes participate to generate blocks. These blocks have some transaction records. They also cooperate in decision making as well. Every node connected to the network will contribute the same to make changes in the public ledger if applicable. Decentralization helps as there is no single point of failure. If any node fails during the execution of any transaction, it won’t affect the transaction much. There will be other peers to make their own separate sub-network in such conditions. Decentralization is extremely important because it is almost impossible to succumb to illegal attacks of hacking or misuse of information. The reason being every peer has a copy of the public ledger. That peer would update the network when any other transaction commits in the blockchain.
2. **Hashcash** is also known as Proof-of-Work. It is a mining algorithm used in Bitcoin. The concept of Hash-cash was suggested by Adam Back in 1997. Hashcash works in the condition as the sender sends the message in encrypted format using SHA-1 algorithms. Brute Force is the only way to get hash-cash to be implemented.
3. **Consensus** as blockchain technology is distributed and decentralized in nature. So, there is no central authority to govern the system. Here Blockchain system relies on a consensus algorithm that ensures a common agreement among nodes in a P2P network. It provides trust in a trustless environment and prevents the double-spending problem and brings higher security to the blockchain system. It ensures fault tolerance and reliability in a distributed system.
4. **Synchronous message passing system** is a system in which a message is passed from one person to the next inside a time frame. If the peers in a blockchain network impart the message that they have answers for the select nonce with the goal, this becomes risky. An off chance that isn’t, at that point it would be a waste of resources and time. Also, an agreement won’t be set up in most likelihood.
5. **Asynchronous message-passing system** is a system in which a message starts broadcasting from one companion to the next with no restriction over time to reach. For instance, everybody cannot expect to the postpone time. This allows a sender to continue doing other things as soon as the message has been sent.
6. **Public Ledger** can be defined as an information book of every successful transaction that is available to all peers (in the peer-to-peer network). The concept of the public ledger has been highly efficient to reduce the attacks on any blockchain network (specially Permission-less networks). If any miner does a malicious activity at the time of adding information to the block, that will be visible to every peer. Hence, it is safe as any malicious activity can be rectified easily by other nodes in the network. Let’s take a simple example to understand. Assume A, B, C, and D are users in the same network. If A makes a transaction with B, then a copy of that transaction will be sent to A, B, C and D. This prevents any malicious activity from happening.
7. **Timestamping in 1991,** Harber and Stornetta used timestamping for digital documents for the first time. Timestamping is utilized in the advanced archive when any report is made or refreshed. In a blockchain, squares joined with themselves have a record of Time Stamp TS, square id, Number of access worth and hash work. A useful implication of Timestamping is in the verification of an advanced archive.
8. **In a Permission-less system** any person/organization interested can do a transaction in open conditions. Bitcoin uses, such a consent-less framework dependent on the Proof of Work algorithm. A fundamental issue advanced from such framework is to get acknowledgment and distinguishing proof of individual who is executing and whether their exchange is at risk to submit or not. This brought forth the idea of Smart Contracts. Today, excellent agreements are written in Bitcoin scripts maintaining distance from the confirmation issue.
9. **In the permissioned model** user is given, appropriate login credentials to the wallet. They are provided with the client login and secret key. With this, the user can log in to their wallet sum and make their transaction. The verification is done by open and private key encryption. The sender will encrypt the transaction with their own private key. The receiver will decrypt the transaction with the public key of the sender. It can be understood that the receiver receiving the transaction might be in a broken state. Thus, the idea of State machine replication came into the picture. In it, numerous companions are given smart contracts. Thus, smooth execution of the framework is possible.
10. **Nonce:** Nonce is a short form for ‘Number Only used Once’. Every block has a nonce associated with it.
11. **Block:** A block in the blockchain is the collection of transactions. Besides Transactions, blocks also stored other metadata like previous block hash, Merkle root hash, etc. A simple block is shown in the figure given below.

Figure 3.1: Block Structure

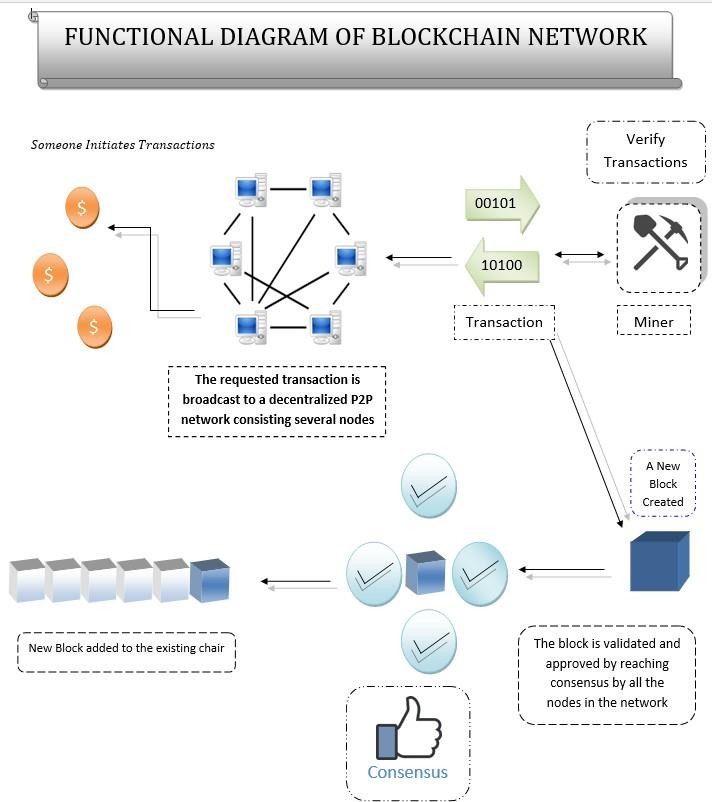
* 1. **Process of blockchain**
     1. Firstly, a transaction is initiated by a user.
     2. This transaction is broadcasted over the P2P decentralized network.
     3. Every peer in the network verifies the transaction to check for any malicious activity using a copy of the public ledger.
     4. This transaction record is again broadcasted over the network with a cryptographic puzzle to solve to maintain the integrity of the hash values of the blockchain.
     5. The primary challenge in such a cryptographic puzzle is to find nonce (Number used only once).
     6. Every miner starts the mining process to find that nonce and the first one to do so correctly is awarded some fixed reward.
     7. Miner commits the transaction record in the block with the solved cryptographic puzzle i.e., nonce.
     8. After committing the block, the miner pushes it to the distributed network where other miners validate the block.
     9. After this the consensus algorithm comes into the picture. Once the final consensus has been reached, then the block is committed to the blockchain.
     10. The block will be added on the long blockchain to avoid fork.

Figure 3.2: Blockchain Functionality

* 1. **Features of blockchain**

1. **Peer-to-peer connection:** A peer-to-peer network(P2P) refers to a decentralized and distributed network of interconnected computers(nodes) that collectively store, upload and share files where each node/computer acts as an individual peer. In a P2P network, nodes can communicate without the need for any central administration or server, which means all nodes have equal privilege and perform the same tasks. In the P2P network model, each peer or node can act as both a client and a server simultaneously. It offers better security compared to the traditional client-server model. It helps in validating and verifying transactions or blocks in the blockchain ecosystem.
2. **Scalable:** Blockchain is a scalable system in which new nodes can be added by a set of guidelines. This scalability mostly enhances the efficiency of the Blockchain.
3. **Reliable:** If the main network fails due to any fault, then peers of the network can form their own sub-network and work under the existing blockchain model. This makes the blockchain systems reliable.
4. **Immutability:** Immutability is one of the key features of blockchain technology. It ensures that any transactions recorded on the blockchain is permanent and cannot be reversed or tampered with by anyone. Immutability becomes one of the most common and attractive features when we are storing confidential data/ information like land records.
5. **Secured:** Blockchain is the one of most secured systems available currently. This has become possible because of many cryptographic algorithms which have been implemented in the core side of blockchain. Some of the attributes of this secured system are Merkle tree, Timestamping, Digital Signatures, Hashing (SHA 256), Cryptography (MD 5), etc.
6. **51% majority:** This concept is also known as the 51% attack and shows a potential shortcoming of the blockchain. When more than half of the peers agree to behave maliciously then there is no solution to get the real information. That’s why we need 51% ethical votes to validate the truth of the information.
7. **No third party:** Blockchain has no central authority to rule. So, using blockchain can provide much more efficient functionality helping with the third-party costs.
8. **Decentralization:** The main bane of centralized systems is “single point failure”. This is where Blockchain comes into the picture. Bringing decentralization. Decentralized system stores data in a distributed manner with peer-to-peer nodes. It offers several benefits like trust, transparency, security, etc.
9. **Transparency:** Transparency is built by making data available for everyone in a distributed network. It is a crucial feature to build trust among different peers. Blockchain makes the system more transparent.

**Chapter 4**

**Blockchain development platforms**

As we know there are many blockchain development platforms are available in the market. But we will discuss about some top blockchain development platforms which are for ease to use and open-source platforms. So, there are four basic blockchain development platforms named as Bitcoin, Ethereum, Hyperledger fabric and Solana. Ethereum is the best one of them and was introduced by Vitalik Buterin in his whitepaper in 2014. Ethereum enables secure digital ledgers to be publicly created and maintained and focused on the ease of transaction and removal of third party to reduce the costs. The explanations about all the three platforms are discussed below.

**4.1 Introduction to Bitcoin**

Bitcoin is the most popular and primarily known as a cryptocurrency. It was introduced by Satoshi Nakamoto in his whitepaper in 2008. Bitcoin was focused only on enabling ease of transactions and removal of central authority. Bitcoin platform uses various hashing (like SHA-256) and cryptographic algorithms (ECC) to provide more security and integrity of the transactions. Bitcoin platform uses consensus algorithm proof of work (PoW) in which many nodes compete to solve a mathematical puzzle for creating the blocks. This process is called as mining who mines first, he gets rewarded for the mining. ECDSA (elliptic curve digital signature algorithm) is an important feature of bitcoin which is used to generate the public keys and private keys for encryption and decryption.



Figure 4.1: Bitcoin

Bitcoin cryptocurrency was developed on its own platform and only exchange or transfer of currencies is performed in this platform in a secured manner. Bitcoin scripting language is used to implement the business logics in which rules and conditions have been written to execute the business logics. There are some important features of bitcoin explained below-

1. **Full Validation:** Bitcoin Core ensures every block and transaction is valid. It increases the security and helps to prevent miners and banks from taking control of bitcoin.
2. **Merkle Tree:** Merkle trees are the data structures in which every non-leaf node is a hash of its respective child nodes. Merkle trees are the integral part of the blockchain which ensures transaction integrity and immutability. Further, it is also helpful in synchronization of various blocks which were not active at some point of time. The tree data structure is also helpful in verification of integrity of transactions in more time efficient manner.
3. **Hashing:** Hashing is an algorithm which converts an input of variable length into a fixed length output. Because of it is irreversible and collision-resistant, it helps to maintain the integrity of transactions. Bitcoin uses hashing algorithms such as Race Integrity Primitives Evaluation Message Digest (RIPEMD) and SHA-256 to generate a hash value of fixed length.
4. **Elliptic Curve Cryptography:** Elliptic curve cryptography is a method of public key cryptography which uses elliptic curve equation to create public and private keys. It uses the concept of generator function to generate the keys which is much faster and more efficient.
5. **Transaction Scripting Language:** Bitcoin uses transaction scripting language to execute the business logics on its own platform. Some important properties of transaction scripting language are written below-
6. Stack based language
7. Stateless language
8. Turing Incomplete language
9. **Nonce:** Nonce is a random number which is used only once in mining process. Miners have to find a value in the process of solving cryptographic puzzle and then the nonce has to be append to hash value, then the resultant value should be match with a predefined output which is previous block hash.

**4.2 Introduction to Ethereum**

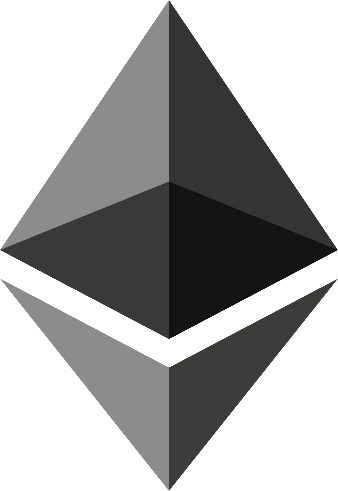
Ethereum blockchain is widely used and most preferrable blockchain development platform. Ethereum was proposed by Vitalik Buterin in his white paper in 2013 and made available for public use. Ethereum platform can be used for both permissioned or permissionless blockchain model. Ethereum platform provides the feature of not only exchange or transfer of currencies but creating any type of applications. These types of application known as decentralized application (DApps) in which Solidity language is used to write smart contracts.

Figure 4.2: Ethereum

Some of the features explained in bitcoin platform like P2P network, Merkle Tree and Consensus algorithms are also same in ethereum platform. Apart from these features, some other features and components of ethereum platform are explained below-

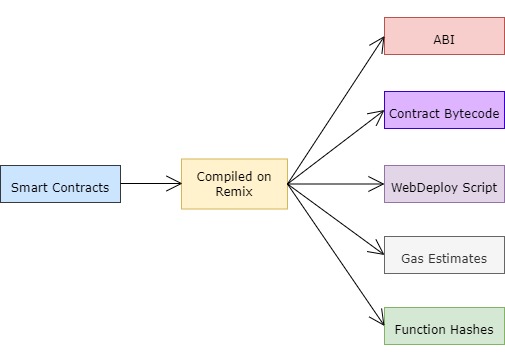
1. **Smart Contract:** The smart contract is a primary feature of Ethereum. Smart contract is defined as a program which contains set of rules or protocols which required to implement the business logics. Smart contract is used to develop decentralized application using Solidity language. It provides separate protocols and logics without being involved in the network and implemented by nodes in a network.
2. **Initial Coin Offering (ICO):** Initial coin offering is an unregulated method of fund raising for a new cryptocurrency. It is used by startups to regulated capital-raising process.
3. **Turing Completeness:** Ethereum platform is turing complete. Turing complete can provide appropriate output to all types of inputs. It gives a benefit to the peers who are validating the transaction.
4. **Gas:** Gas is defined as fees which required to complete the transaction. It ensures that a specified amount of gas is given to each transaction and prevents the forever running of a particular transaction by being terminated of transaction process after consuming allocated gas.
5. **Ethereum Virtual Machine (EVM):** Ethereum virtual machine is a virtual machine which is used to run the byte code of the contract. By compiling smart contract, it generates byte code for a particular smart contract and then EVM provides flexibility to run this byte code any EVM compatible system
6. **Remix IDE:** Remix IDE is an open-source tool which is written in JavaScript. It is used to create, compile, deploy and debug Solidity code. It provides several types of environments to deploy the smart contract which are JavaScript VM, Injected Web3, Web3 Provider etc. Remix produces artifacts which are useful in the blockchain application development, from which some artifacts are given in the figure-

Figure 4.3: Artifacts generated in compilation process on Remix

1. **Metamask:** Metamask is an extension which acts as a crypto wallet where we can store our cryptocurrencies. This allows a particular node to interact with decentralized ethereum applications.
2. **Wallets and Ethers:** Ether is the virtual crypto-currency of ethereum which is used for exchange or transfer as well as used as a required fees while deploying the smart contracts and performing transactions.

**4.3 Introduction to Hyperledger**

Hyperledger is an open source blockchain application development platform which is used to develop business logics and bring the transparency and efficiency of Distributed Ledger Technology (DLT) to the enterprise market. Hyperledger is not a cryptocurrency like bitcoin but it can be called as open-source project which is developed by The Linux Foundation. The hyperledger follows the permissioned approach and is based on the concept of “Plug and Play”. Hyperledger contains a lot of frameworks and tools for creating blockchain applications.

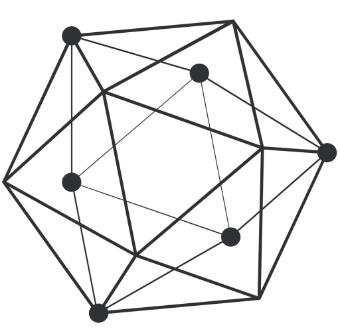


Figure 4.4: Hyperledger

Hyperledger Fabric is one of the most used hyperledger framework which is developed by IBM. Some important components of hyperledger are discussed below-

1. **Chaincodes:** Chaincodes are the business logics which are similar to smart contracts in ethereum. Programming languages such as Java, JavaScript, and Golang are used to create Chaincodes.
2. **Identity:** All nodes are prior known to each other in permissioned model. The certifying authority gives them a unique identity.
3. **Ledger:** Ledger is defined as transaction logs which stores all transactions that are validated by peers available in a network.
4. **Channelization:** Confidentiality is the major problem in cross-industry collaboration because hyperledger is used as a collaborative approach. So, channelization has been implemented to provide confidentiality. Channelization is the process of creating two different channels, one of which is open to everyone and the other of which is only utilized by authenticated peers.
5. **Endorsing Peer:** Endorsing peers are the nodes available in a network which are used to take decision. The peers either accept or reject the transaction according to predefined rules and policy.
6. **Committing Peer:** These peers keep all the blocks which are broadcasted to them and maintain the ledger of all blocks. These are responsible for final committing the transactions in the block.
7. **Ordering Peer:** Ordering peer takes all transactions in proper sequential order, create a block by ordering transactions and broadcast the block to all the peers.
8. **World State:** It is a component of hyperledger which provides current ownership of any asset.

**4.4 Introduction to Solana**

Solana is a public, highly functional, layer-1 open-source blockchain platform designed to host decentralized, scalable applications. Solana was created in 2017 by Anatoly Yakovenko. It implements a new, high-performance, permissionless blockchain that supports smart contracts. The Solana Foundation is based in Geneva, Switzerland and maintains the open-source project. It is much faster in terms of the number of transactions (50,000 TPS (Transactions per second) and 400ms Block Times) it can process and has comparatively lower transaction fees.

****

Figure 4.5: Solana

It implements an innovative hybrid consensus model that integrates a unique proof-of-history (PoH) consensus algorithm with the lightning-fast synchronization engine, that's a version of proof-of-stake (PoS). It is built on RUST programming language which make it robust and safe. SOL is the Solana's native token.

**4.5 Introduction to Near Blockchain**

Near is a layer-1, sharded, permissionless, proof-of-stake blockchain built for usability and scalability. It is smart contract-capable blockchain that is designed to be a next-generation platform for developing DApps.

It is comparatively cheaper, faster and easier to use which provides 1000x lower transaction fees for users, 1.2 second block production time & 30% of gas fees are paid out to the smart contract developers. We can write near smart contracts using rust programming or Assembly Script. It's climate-neutral blockchain and EVM compatible with Project Aurora.

**Chapter 5**

**Literature Review**

The blockchain is such a technology which has vast applications in almost every aspect of the world. This technology could be extremely helpful in the field of land registry system. In this chapter, we have discussed all the related works and researches which were carried out earlier and have the similar concept as our project.

**5.1 Related works**

Land registration can be defined as a process of transfer the ownership by registering on the documents and other transactions related to the properties. Land registry is a huge system for the government in the world by which the user can purchase and sell the property. In current era as we know the land registry system is more difficult to maintain the integrity and track its transaction records because there is a lot of issues occur like record tampering, third party involvement, protection from competing claims due to data unavailability etc. So, for fix these problems we proposed to use Blockchain Technology. Blockchain technology is a peer-to-peer decentralized ledger of the transactions that provides all transaction records, transparency, immutability, reliabilities and removes third party involvement. Here we have reviewed few research papers and other resources to understand its fundamental proposed model and what type of consensus mechanism and algorithms are used. [1] Most of the problem comes from the current land registration systems which is either having legacy paper document trails or from centralized systems where the records kept non-transparent. Fraudulent users may try to modify electronic records and to forge paper documents to change the land ownership record. This white paper offers a mechanism which keeps more secure records and address these issues using a blockchain-based system. This system can generate records of real assets into unchangeable liquid blockchain-based token assets. [2] This white paper describes a proof of concept (PoC) that integrates blockchain into a land registration system. It reduces processing steps and makes better storage reliability. The PoC uses the procedure of borrowing with banks to develop and test an Ethereum-based private blockchain. The performance of this model can be handled by at least 26 transactions per second. [3] The distributed ledger technology underlying blockchain is expected to restructure land registration by providing a secure architecture for storing land transactions using cryptographic protocols. This brings the benefits of increased reliability, processing efficiency, and cost reduction. [4] This document proposes a framework for auto-maintaining records for registration records. By using this framework, they can also resolve the biggest problem of clearing loans and creates transparency in government electronic governance. Instead of the existing common manual processor online registry records, the systems processed by the blockchain network include: Apply multi-level authentication schemes to secure blockchain-based frameworks, a quick verification, for all the batch holders such as registry departments, banks, buyers, and sellers), and application of smart contracts is to automatically check the loan process according to government rules and facilitate buyers and sellers for selling and purchases. [5] In this paper, some claim that the blockchain-based approach to title registry properties can significantly improve the efficiency of transactions and even prevent fraud. It is also claimed that property transactions can be processed on the blockchain in a manner similar to payments between parties using digital currencies. [6] This paper aims to use blockchain technology as a property registration mechanism for Cyprus country and other disputed land situations to support land conflicts that can promote peace and harmony. [7] Making tamper proof land registration system is a very big problem for any country. Specially in developing country like Pakistan, due to corruption and poverty, a centralized system has many loopholes. For solving this problem, this paper proposed a decentralized system by using proof-of-concept technique. This model also consists the necessary component of land registry system. [8] This research paper has a case study of using blockchain technology in land registry system of two countries, Georgia and Honduras. This paper also discusses the difficulties to implement a modern technology in a traditional system. It highlights issues of the political and unaware of modern technologies. By analysing these two cases, it discusses the factors affecting the decentralization of public entities and finds the best solution for identification problem. [9] The land registry paper contains very essential and sensitive information. Thus, it needs to keep secure and tamperproof. But the current centralized registry system doesn’t have any high secured and tamperproof database. So, this paper proposed a blockchain based decentralized system with high security and immutability of data. In this proposed model when a person tries to register for a property, he is provided a hard copy for the registration by the government officials and also a copy stored in a decentralized Inter Planetary File System (IPFS). For making more secure and tamper proof, a hash value of document provided by IPFS is stored in Ethereum network. [10] The land administration system represents the record of ownership of a land. The records of land registration system are considered as always true from the legal point of view that’s why correctness and security of data should be maintained. But this land administration system has many problems like it can be influenced by a powerful person. Thus, correctness of data cannot be maintained in this system. So, this paper proposed a model to use blockchain technology to remove all these vulnerabilities. It also discusses the features of blockchain like decentralization, immutability etc. [11] It is very difficult to prove the ownership of a land on paper work. So, this paper focuses on problems with the manual land registration process like transparency, mutability, security and authenticity, and provides best ways to resolve these issues using blockchain technology. This paper also provides a comparison of digitalized systems property registration based on blockchain in different countries. [12] This document proposes solution based on blockchain technology that provides information integration and transparency, accessibility, immutability, and a faster and cheaper solution. They introduced a gradual Blockchain adoption model that starts with a public Blockchain ledger and later steadily contains levels of hybrid blockchain. [13] This study describes a blockchain-based framework for digitizing real estate transactions, which reduces the risk of document fabrication and other fraudulent acts. To decentralise the whole ecosystem, the suggested architecture is built on blockchain technology. The framework integrates multiple area register offices in the state/country seamlessly by using the InterPlanetary File System (IPFS), which is a Peer-to-Peer (P2P) swarm network. [14] Distributed ledger technology (DLT) is utilized to build this digital log additionally secure and proceed it in a decentralized method, decreases the paperwork in trading land. The proposed MRRCM consensus algorithm achieves less block generation time with an average of 3.05% round-robin approach and 95.48% PoW. [15] In this paper, they analysed transaction management in the Serbian land authority and how this process can be backed up by contemporary ledger technologies like blockchain. To investigate how blockchain can be executed to support Land Information System (LIS) transactions, it is compulsory to figure out the cadastral and transactional processes in the LIS, as well as the parliamentary and regulatory aspects. [16] This article illustrates a type of blockchain solution for handling present records that represents three different design arrangements. It deals with distinct types of solutions regarding the significance for record keeping and long-period conservancy of legitimate documents. [17] In this paper5, we introduce LandLedger, a blockchain-based land administration system that enables responsible, transparent, efficient, secure, and scalable land property management. LandLedger's proposed architecture achieves property verification, registration, and revocation through specially designed transactions on a permissioned blockchain, which is managed by various departments such as the Registrar's office, the Income Tax Department, and the Revenue Department, among others. [18] Blockchain technology has the potential to greatly enhance the processes involved in keeping electronic records for landed property. The deeds office and the South African Reserve Bank (SARB) must keep control of the procedures in order to protect the integrity of the property record.

**5.2 Related papers**

|  |  |  |  |
| --- | --- | --- | --- |
| Research Paper | Their Work | Our Work | Link to papers |
| Blockchain - can this new technology really revolutionize the land registry system? | Analysis of blockchain based land registry system. | Implemented blockchain technology in land registry system. | <https://www.notariesofeurope.eu/wp-content/uploads/2021/09/Land-and-Poverty-Conference_Blockchain-Presentation.pdf> |
| Blockchain as a tool for land rights: ownership of land in Cyprus | This paper raises the historical and present land issues in Cyprus and highlights that blockchain technologies could work as a tool to record disputed property rights on the Island. | Focused on its implementation, issues and used tools and technologies which are useful for any country. | <https://www.emerald.com/insight/content/doi/10.1108/JPPEL-02-2020-0010/full/html> |
| Blockchain-Based Land Registration System: A Conceptual Framework | This fixes the flaws of centralized systems and a proof-of-concept system used for implementation. | Understood the proof-of-concept for the implementation. | <https://www.hindawi.com/journals/abb/2022/3859629/> |
| Digitalization of Land Records: From Paper to Blockchain | Discussed two country use cases of land record modernization. | We are working on modernization of land registry system for our country. | <https://cutt.ly/GJtUlf1> |
| Land Registry Using Blockchain - A survey of existing systems and proposing a feasible solution | Proposed blockchain based system for secure storage of property papers and introduced IPFS for document storing. | Focused on IPFS which is used to store the property papers. | <https://ieeexplore.ieee.org/abstract/document/9129289> |
| Blockchain and land administration: possible applications and limitations | Prospects of using different implementations of blockchain technology, mainly using permissionless or permissioned blockchain is also presented. | We are using permissionless model where anyone can get the information of any person which is broadcasted to the peers available in network. | <https://www.researchgate.net/profile/Miroslav-Stefanovic-2/publication/329650717_Blockchain_and_Land_Administration_Possible_applications_and_limitations/links/5c79204e458515831f78b72c/Blockchain-and-Land-Administration-Possible-applications-and-limitations.pdf> |
| A Novel Framework for Implementation of Land Registration and Ownership Management via Blockchain in Bangladesh | This paper highlights issues related to manual land registrations processes like transparency, centralization, authenticity, reliability, etc and proposes a better method to overcome these problems using blockchain. | Working on maintaining its transparency, immutability, integrity and reliability etc. | <https://ieeexplore.ieee.org/abstract/document/9230721> |
| A Blockchain-based Land Title Management System for Bangladesh | Blockchain-based solution that offers [data synchronization](https://www.sciencedirect.com/topics/computer-science/data-synchronization) and transparency, ease of access, immutable records management, a faster and cheaper solution. | We are working with ethereum which provides flexibility to write smart contracts. | <https://www.sciencedirect.com/science/article/pii/S1319157820304912> |
| Digitization of Land Record Through Blockchain-based Consensus Algorithm | Proposed a blockchain-based framework for digitizing property transactions that mitigate the risk of document forgery and other fraudulent activities which uses IPFS to integrate different region registry offices. | Focusing on reduce risk of document forgery using IPFS concept in public blockchain. | <https://www.tandfonline.com/doi/abs/10.1080/02564602.2021.1908859> |
| Query Optimization in a Blockchain-Based Land Registry Management System | Distributed Ledger Technology (DLT) is used for making this digitized record more secure and process it in a decentralized way, and reduces paperwork in selling and buying of land. | Implementing DLT technologies for security and transparency of the data. | <https://www.researchgate.net/profile/Amrendra-Yadav/publication/350547407_Query_Optimization_in_a_Blockchain-Based_Land_Registry_Management_System/links/60677d86299bf1252e24458a/Query-Optimization-in-a-Blockchain-Based-Land-Registry-Management-System.pdf> |

Table 5.1: Literature Survey

**Chapter 6**

**Proposed Methodology**

Land registration in India, as in many other countries of the world, is a time-consuming and inconvenient process. Due to the millions of land records to keep, current land registration and verification processes are prone to fraud and the loss of documents and court proceedings. The idea behind creating this was to make the land registration process more robust and to reduce incidences of fraud in the process. Validation of the lands is also achievable using the system, as immutable transactions are kept in the public ledger. As a result, the Land Registration system based on block-chain is a distributed system that will keep all transactions done during the land purchase process.

**6.1 Proposed methodology**

In this model, we have proposed three modules in our system. These modules such as Contract Owner, User dashboard and land inspector dashboard where each module defines its own individual functionalities and working. we created the smart contract using solidity and deploy the contract where mainly we used User registration(seller/buyer), and land inspector functions. Using the flutter framework, we have created the dashboard of the user registration, contract owner and land inspector and interconnect them using web3Dart to make a web application as well as android application. This model will save not only time but also the efforts that are supposed to be put for land registering in traditional manner. For the front-end we are using *flutter Framework, web3Dart, Meta-mask Chrome Extension*, and for the back-end we are using *Ethereum Blockchain (Truffle Suite), Solidity, Ganache.* We are using meta-mask as a wallet for performing transactions and using Ganache for testing solidity smart contracts. The transaction is verifiable on Ethereum platform which history we can see in etherscan.

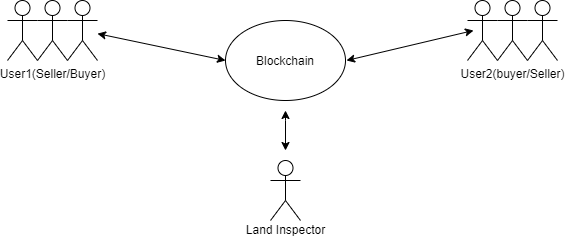


Figure.6.1: Land Registration using Blockchain

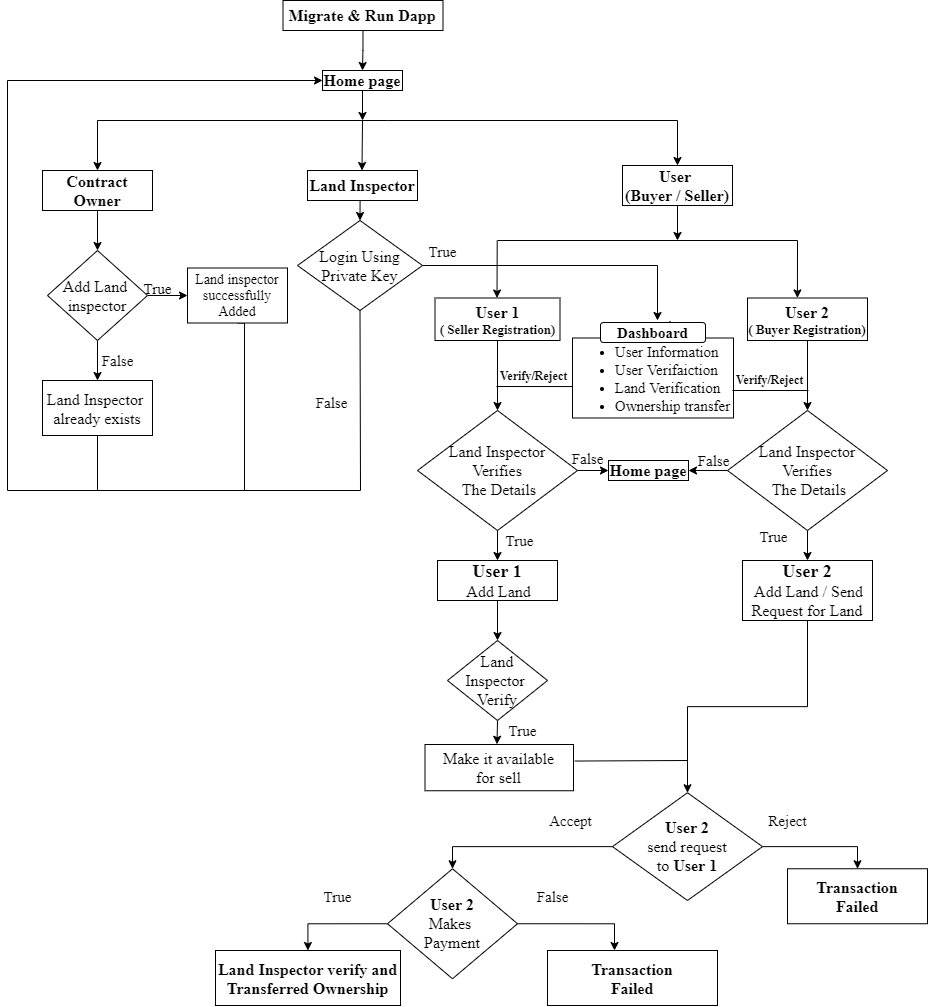


Figure 6.2: working flowchart of land registration system with Blockchain

The working of the proposed model can be understood as shown in fig.6.2. first of all, contract owner who have deploy the smart contract and after login with his private key, he can add land inspector on the decentralized platform if there is failed in registration then land inspector is already exists but if it is success land inspector will be successfully added.

Land inspector will use his private key to login in his dashboard if there is false it will return to home page. There is three function which is done by the land inspector it verifies the user account, added lands and verify the ownership transfer. after the verification of the user account, user can use his dashboard as a buyer as well as seller. If the verification is not approved by the land inspector it will go to home page. If the user1(seller) want to sell the land he will add the land and land inspector will verify the land and make it available for the sell and user2(buyer) will send a request for land. if request is rejected by the user1(seller) transaction will failed and if request is accepted then a request of payment will send to the user2(buyer) after the successful payment land inspector will verify transaction and ownership transferred. if the verification is failed transaction will failed.

The proposed model uses the Blockchain to solve the Land Registration problem. We can see the Paper System of Land registry System is Slow, Unsecure, inefficient and time Consuming so there is proposed solution using Blockchain. There are various functions which are used to design the model- There is buyer registration land inspector seller registration these are the main functions.

***User Registration:*** This function is used for registration one who wants to buy/sell the land.

***Land Inspector:*** Land inspector is provided by government authority who verifies property details.

***Land Inspector Dashboard:*** This function has user information, user verification land verification and ownership transferred details.

***User Dashboard:*** In this dashboard, the profile has both seller and buyer functions. It can add lands, request for lands and have owned land details.

***Add Land:*** When the land is certified by the land inspector it is added to the land gallery.

***Approve Land Request:*** This functionality is used when land is approved that is request by the user(buyer) to buy.

***My Land:*** This function has details of hand over lands by the buyer.

***Make payment:***When the land request is approved the payment is transferred to the seller.

***Land gallery:***There is full information about lands with images.

***Land Ownership Transfer:***When the payment is verified by the land inspector and details land is hand over to the buyer from the seller.

**Chapter 7**

**Tools and Technologies**

The implementation of any project requires certain sets of tools and software. The blockchain related works are always accompanied with complex tools and software like flutter Framework, web3Dart, Meta-mask Chrome Extension etc. This chapter will give the details of tools and software used for the project Land Registration.

**7.1 IPFS (Inter planetary file system)**

The Inter Planetary File System (IPFS) is a protocol and peer-to-peer network that allows data to be stored and shared in a distributed file system. IPFS use content-addressing to uniquely identify each file in a global namespace that connects all computing devices. IPFS aspires to build a permanent and distributed web. It accomplishes this by employing a content-addressed system rather than HTTP's location-based approach. Instead of a geographical address, IPFS addresses the content using a representation of the content itself. This is accomplished by performing a cryptographic hash on a file, which serves as the address. Other items can be located in the path of the hash, which symbolises a root object. The working of IPFS involves the use of direct acyclic graphs (DAG), content-based addressing and the distributed hash tables (DHT). Each and every file stored on the IPFS is accessed with the help of its content and this is known as content-based addressing. If any file is updated later, then its content gets changed and due to which its hash also gets changed. Therefore, the IPFS is also helpful in keeping a track of the different versions of the file. The IPFS makes the use of Merkle DAG to represent the contents which are coming from different sources or peers.

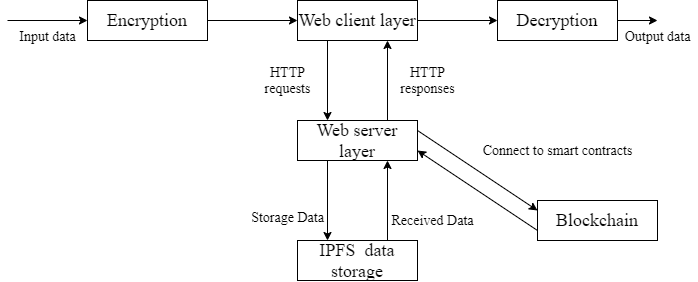


Figure 7.1 Inter Planetary File System

**7.2 Blockchain platform: Ethereum**

Ethereum blockchain platform was introduced by Vitalik Buterin in 2013.Smart contracts, which are pieces of code that accomplish certain functions without the assistance of a third party, were introduced by Ethereum. The Ethereum platform facilitates the creation of decentralized applications (DApps) that allow users to conduct transactions in a decentralized way. It has its own cryptocurrency, ether, which is the second most valuable cryptocurrency in terms of market value after Bitcoin. When a transaction initiates a smart contract, all network nodes execute all instructions. To do this, Ethereum creates an execution environment on the blockchain known as the Ethereum Virtual Machine (EVM). As part of the block verification protocol, all nodes on the network use the EVM. Various Ethereum tools used in our work are discussed below-

* **Solidity:** Solidity is a turing-complete programming language for creating smart contracts. It is comparable to Java and adheres to object-oriented programming principles.
* **Remix:** Remix is an IDE for creating smart contracts using the Solidity programming language. It features a very user-friendly interface for creating, compiling, and deploying smart contracts.
* **Ethereum Virtual Machine (EVM):** EVM helps in the execution of the smart contracts.

**7.3 Ganache**

Ganache is a personal Ethereum Blockchain which is used in the development, deployment and testing of decentralized application. Ganache is available in two flavors’: UI and CLI. Ganache UI is a desktop application that works with both Ethereum and Corda. For Ethereum development, the command-line tool ganache-cli (previously known as the TestRPC) is now accessible. Ganache is an Ethereum simulator that makes developing Ethereum applications faster, easier, and safer. It includes all popular RPC functions and features (like events) and can be run deterministically to make development a breeze.

* Zero-config Mainnet Forking
* Fork any Ethereum network without waiting to sync
* Ethereum JSON-RPC support
* Snapshot/revert state
* Mine blocks instantly, on demand, or at an interval
* Fast-forward time
* Impersonate any account (no private keys required!)
* Listens for JSON-RPC 2.0 requests over HTTP/WebSockets
* Programmatic use in Node.js
* Pending Transactions

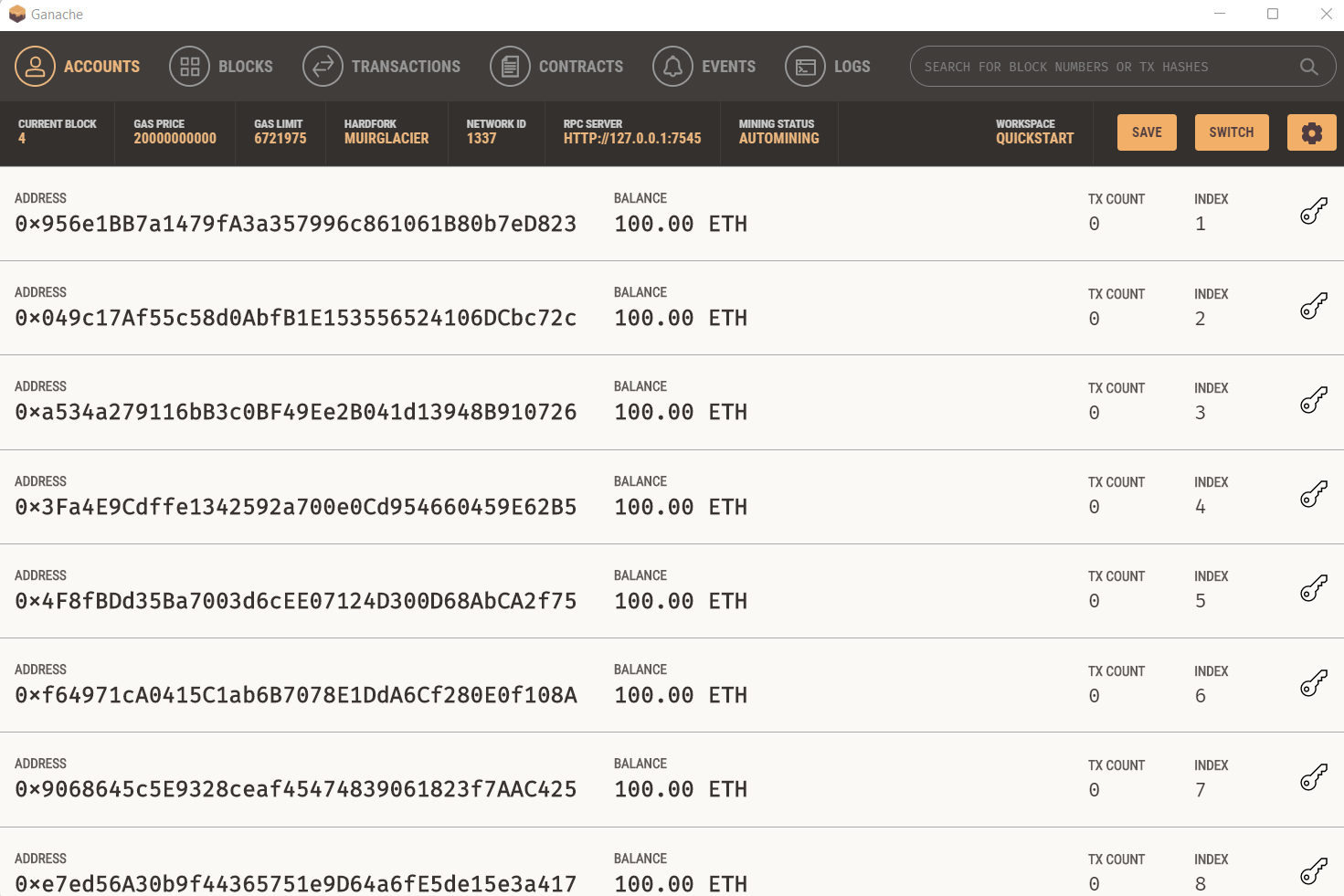


Figure 7.2: Ganache

**7.4 Metamask**

Metamask is a cryptocurrency wallet that works with the Ethereum Blockchain. To access the Ethereum wallet, it is utilized as a browser extension or a mobile application. It is the real location where ethers and cryptographic keys are stored. Users may use Metamask to store and manage account keys, broadcast transactions, transfer and receive Ethereum-based coins and tokens, and securely connect to decentralized apps using a suitable web browser or the built-in browser of the mobile app. Developers integrate Metamask to their decentralized apps by defining interactions between Metamask and Smart Contracts using a JavaScript plugin such as Web3js or Ethers.

The Metamask app features an integrated tool for trading Ethereum tokens by combining the best exchange rates from many decentralized exchanges (DEXs).

**7.5 Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, based on JetBrains' IntelliJ IDEA software and customized exclusively for Android development. It is a substitute for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development. Android Studio supports all of the programming languages supported by IntelliJ (and CLion), including Java, C++, and others with extensions such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version”. Android Studio is a single development environment that allows you to create apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to break your project into functional components that you can create, test, and debug separately. Android Studio employs a Gradle-based build system, emulator, code templates, and GitHub integration to allow application development within the Android operating system. Every Android Studio project includes one or more modalities containing source code and resource files. Android app modules, Library modules, and Google App Engine modules are among these modalities.

## **Features of Android Studio**

* It has a flexible Gradle-based build system.
* It has a fast and feature-rich emulator for app testing.
* Android Studio has a consolidated environment where we can develop for all Android devices.
* Apply changes to the resource code of our running app without restarting the app.
* Android Studio provides extensive testing tools and frameworks.
* It supports C++ and NDK.
* It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.

**7.6 Flutter framework**

Flutter is a Google open-source framework for creating attractive, natively built, multi-platform apps from a single codebase. Flutter is a cross-platform user interface toolkit that allows code reuse between operating systems such as iOS and Android while also letting apps to interact directly with underlying platform functions. The objective is for developers to be able to produce high-performance programmes that feel natural across platforms, embracing differences where they exist while sharing as much code as feasible. Flutter apps run in a VM during development, allowing for stateful rapid reload of changes without requiring a full recompile. Flutter apps are compiled straight to machine code, whether Intel x64 or ARM instructions, or to JavaScript if they are intended for the web. The framework is open source and distributed under the permissive BSD licence, with a vibrant community of third-party packages that enhance the core library capabilities.

This overview is broken into the following sections:

* The layer model is made up of the elements that make up Flutter.
* Reactive user interfaces are a fundamental element in the creation of Flutter user interfaces.
* **An overview of widgets:** The core components of Flutter user interfaces.
* **The rendering method:** Flutter's method for converting UI code into pixels.
* **A summary of the platform embedders:** The code that allows mobile and desktop operating systems to run Flutter apps.
* **Flutter integration with other code:** Information on the many approaches accessible to Flutter applications.
* **Web-based assistance:** Final thoughts on Flutter's properties in a browser context.

**7.6.1 Web3Dart language**

A dart library that connects to the Ethereum blockchain to communicate with it. It communicates with an Ethereum node in order to send transactions and engage with smart contracts.

**Features**

* Connect to an Ethereum node with the rpc-api, call common methods
* Send signed Ethereum transactions
* Generate private keys, setup new Ethereum addresses
* Call functions on smart contracts and listen for contract events
* Code generation based on smart contract ABI for easier interaction

**Chapter 8**

**Implementation of Project**

In this chapter, we will discuss how implemented our project. This includes downloading, deploying of smart contract and configuration of tools and software used. This will mainly focus right from the starting of smart contracts to the finally running decentralized application.

* 1. **Smart Contract**

The smart contract is a primary feature of Ethereum. Smart contract is defined as a program which contains set of rules or protocols which required to implement the business logics. Smart contract is used to develop decentralized application using Solidity language. It provides separate protocols and logics without being involved in the network and implemented by nodes in a network.

**A simple pseudocode Smart Contract for Land Registry System:**

Contract Land {

**// creating land Registration**

*struct Landreg {  
 uint id;  
 uint area;  
 string landAddress;  
 uint landPrice;  
 string allLatitudeLongitude;  
 uint propertyPID;  
 string physicalSurveyNumber;  
 string document;  
 bool isforSell;  
 address payable ownerAddress;  
 bool isLandVerified;  
}*

**// creating user details**

*struct User {  
 address id;  
 string name;  
 uint age;  
 string city;  
 string aadharNumber;  
 string panNumber;  
 string document;  
 string email;  
 bool isUserVerified;  
}*

**// creating land Inspector details**

*struct LandInspector {  
 uint id;  
 address \_addr;  
 string name;  
 uint age;  
 string designation;  
 string city;  
}*

**// adding Land Inspector on decentralized platform**

*function addLandInspector (address \_addr, string memory \_name, uint \_age, string memory \_designation, string memory \_city) public returns(bool) {  
 if(contractOwner!=msg.sender)  
 return false;  
 require(contractOwner==msg.sender); // check contractOwner is the contract deployer  
 // adding all details   
}*

**// adding user by giving his details**

*function registerUser (string memory \_name, uint \_age, string memory \_city, string memory \_aadharNumber, string memory \_panNumber, string memory \_document, string memory \_email) public {  
 require(!RegisteredUserMapping[msg.sender]); // check if user already registered  
 // adding all details of user  
}*

**// adding land details by seller**

*function addLand (uint \_area, string memory \_address, uint landPrice, string memory \_allLatiLongi, uint \_propertyPID, string memory \_surveyNum, string memory \_document) public {  
 require(isUserVerified(msg.sender)); // checking user is verified before adding land  
 // adding land details   
}*

**// requesting to buy land by buyer to seller**

*function requestforBuy(uint \_landId) public {  
 require(isUserVerified(msg.sender) && isLandVerified(\_landId));  
 // implements for buying land from seller  
}*

**// accepting land request of buyer**

*function acceptRequest(uint \_requestId) public  
{  
 require(LandRequestMapping[\_requestId].sellerId==msg.sender);  
 LandRequestMapping[\_requestId].requestStatus=reqStatus.accepted;  
}*

**// making payment by buyer to seller**

*function makePayment(uint \_requestId) public payable  
{  
 require(LandRequestMapping[\_requestId].buyerId==msg.sender &&*

*LandRequestMapping[\_requestId].requestStatus == reqStatus.accepted);  
 // implemented code for making payment   
}*

**// transfer ownership to buyer from seller by land inspector**

function transferOwnership(uint \_requestId,string memory documentUrl) public returns(bool)  
{  
 require(isLandInspector(msg.sender));  
 if(LandRequestMapping[\_requestId].isPaymentDone==false)  
 return false;  
 documentId++;  
 LandRequestMapping[\_requestId].requestStatus=reqStatus.commpleted;  
 // implementation of transfer ownership by land inspector to buyer   
}

}

* 1. **Installations**

There are NodeJS and npm are required to work and execute our project. So, all installations and all other required dependencies are given below.

* **Node modules**

1. Open command line terminal and set to your project directory.

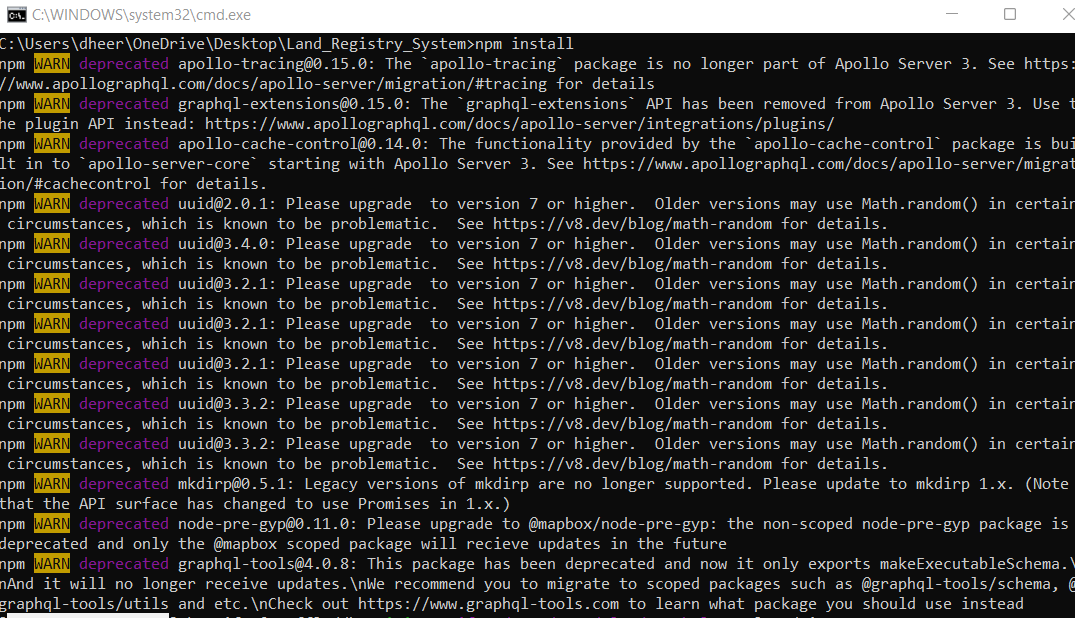
2. Run npm install.

Figure 8.1: Node Modules Installation

* **Ganache**

1. Go to [Ganache homepage](https://trufflesuite.com/ganache/) to download Ganache.

2. If you are using Linux, you must have got an *.appimage* file and follow further

installation process. [Click here](https://itsfoss.com/use-appimage-linux/).

* **IPFS**

1. Go to [download page](https://docs.ipfs.io/) of IPFS and follow the instructions given in this doc file.

* **Metamask**

1. Metamask is a browser extension available for Google Chrome, Mozilla Firefox and

Brave browser.

2. Go to [Metamask download](https://metamask.io/) and add Metamask to your browser.

* **Android studio**

1. Android studio provides the various tools to the user for building android applications.

2. Go to [Android Studio](https://developer.android.com/studio) to download.

* **Flutter**

1. Flutter is an open-source framework which is used to create beautiful applications from

a single codebase.

2. You can download Flutter by clicking [here](https://docs.flutter.dev/get-started/install).

**8.3 Configuration**

* **Ganache**

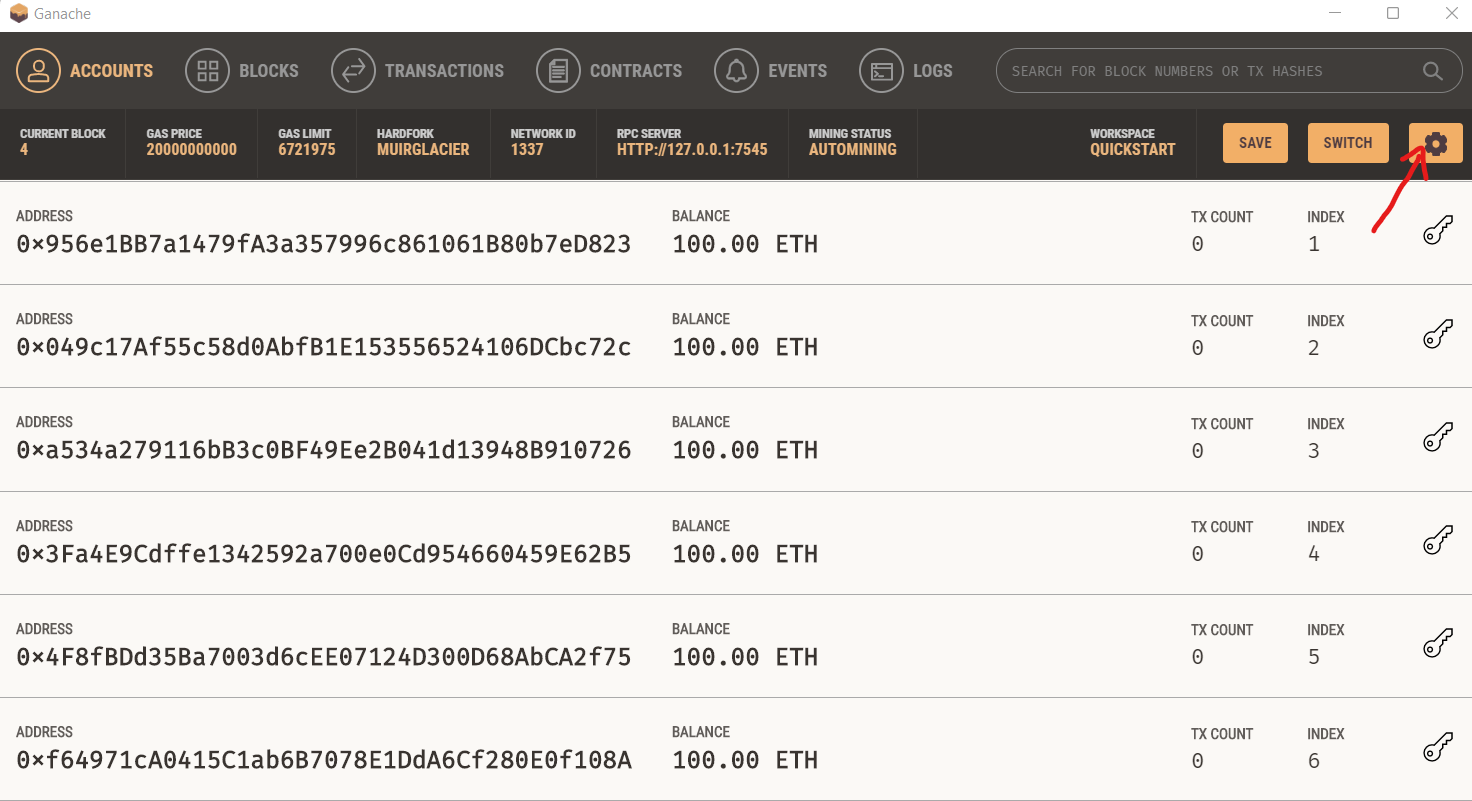
1. Install and open Ganache then click on settings button in the top right corner.

Figure 8.2: Ganache

2. Go to **Server** section:

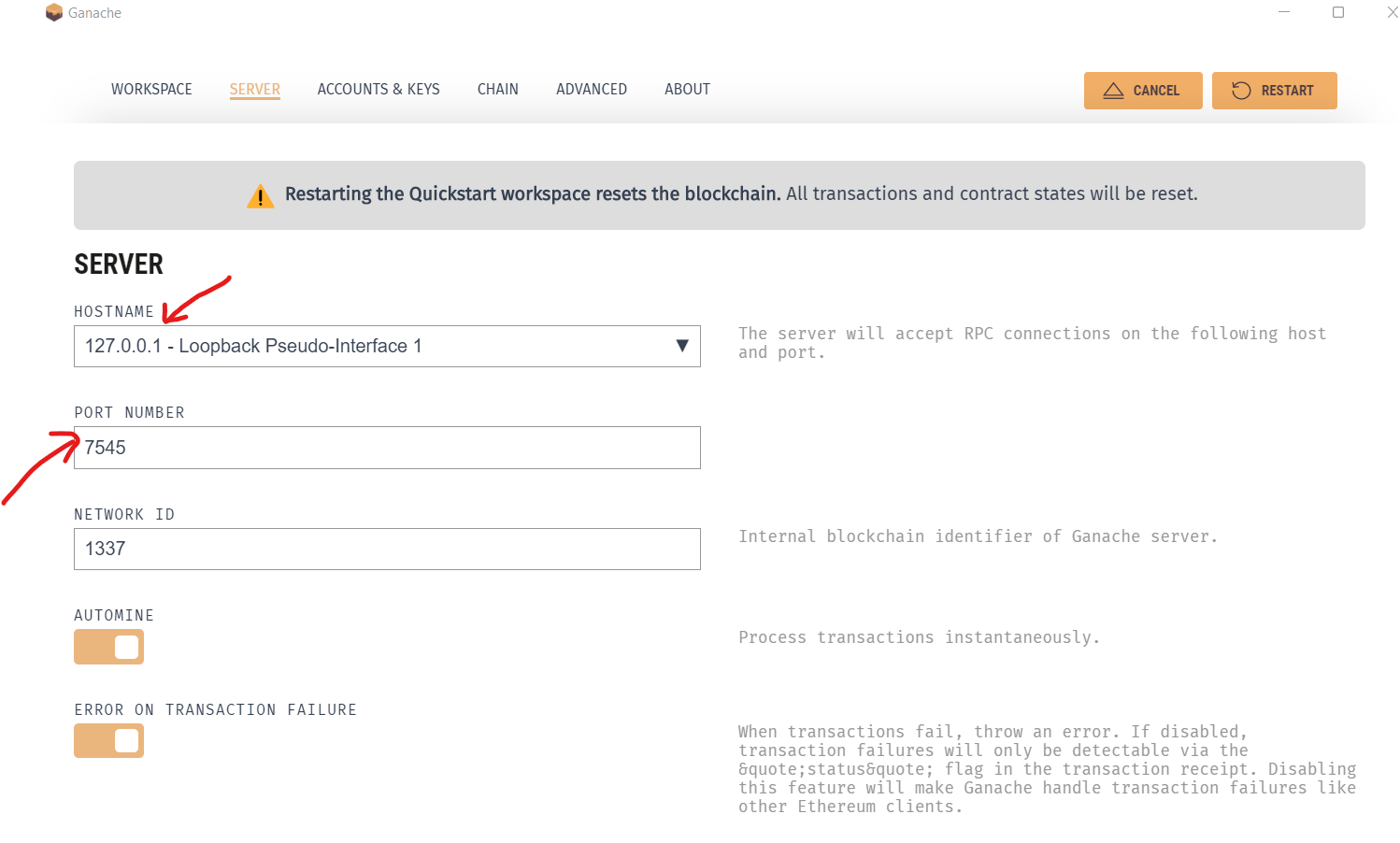
1. Select Hostname to 127.0.0.1 -Loopback Pseudo-Interface1
2. Set Port Number to 7545
3. Enable Automine

Figure 8.3: Ganache auto-mining configuration

* **Metamask**

1. Install Metamask and open on your browser.

2. You will see *Get Started* button. Click on that button.

3. Click on *Create a Wallet* button. If you have already created a wallet then just click on

*Import Wallet* button.

4. Click on *I Agree* button by accepting all terms and conditions.

5. Create a Password and click to unlock backup phrase and after selecting all phrase words

in sequence as you got by unlocking backup phrase.

5. Now you are ready with Metamask wallet.

* **Flutter SDK**

After installing flutter, we have to give SDK path to our project for execution of front end. So, download flutter SDK and then give SDK directory path to the project.

**8.4 Deploying smart contract with Truffle:**

**Tools used:**

* Solidity
* Metamask
* Truffle
* Ganache
* Smart contract

**Front-end:**

* Flutter
* Web3Dart

**About Truffle:**

Truffle, a framework, is used to develop, test, compile and deploy our smart contracts for creating decentralized application. It helps in better simulation of Blockchain environment.

**Getting Started:**

* Install the truffle package by using the given command in command line terminal:

npm install -g truffle.

* Go to the project directory in which smart contract is available.
* Create a truffle project using the following command:

truffle init

* This command will create four files in our project directory-

1. Contracts

2. Migrations

3. Test

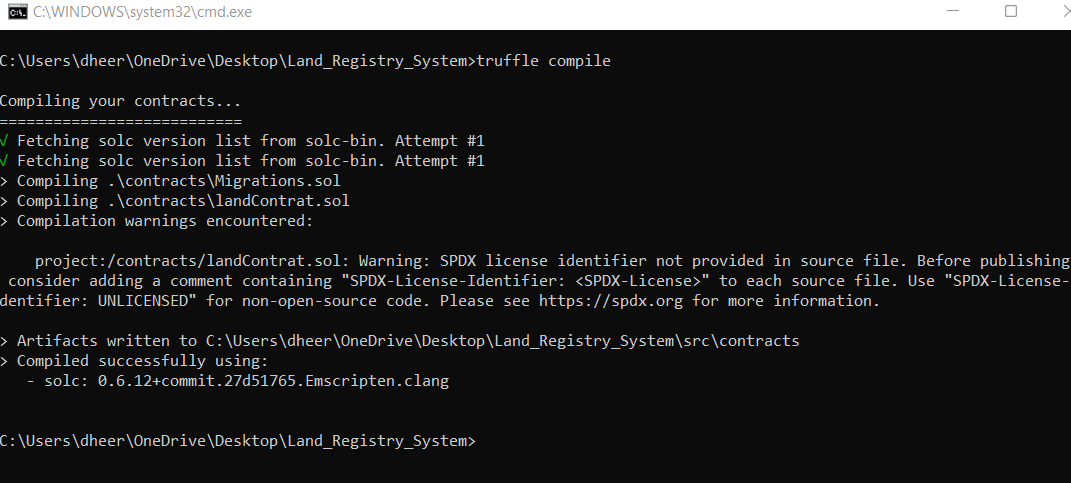
4. Truffle-config.js

**Compiling the contract:**

* Edit your truffle-config.js file according to your smart contract and network used.

module.exports = {  
 networks: {  
 development: {  
 host: "127.0.0.1",  
 port: 7545,  
 network\_id: "\*", *// Match any network id* },  
 advanced: {  
 websockets: true,  
 },  
 },  
};

* Now you’ll get trffle-confir.js file and compile your smart contract using following command: *truffle compile*

Figure 8.4: Contract compilation using Truffle

**Starting Local Development Blockchain Network:**

* Open Ganache.
* Make sure to configure it the way mentioned above.

**Deploying the smart contract using Truffle:**

* Create a new file named as **2\_deploy\_migrations.js** in the migration directory and place the following code:

*const landContract = artifacts.require(****"****Land****"****);  
  
module.exports = function(deployer) {  
 deployer.deploy(landContract);  
};*

* Let’s open **truffle-config.js** and change the networks section which is given below:

*networks: {  
 development: {  
 host: "127.0.0.1",  
 port: 7545,  
 network\_id: "\*", // Match any network id  
 },*

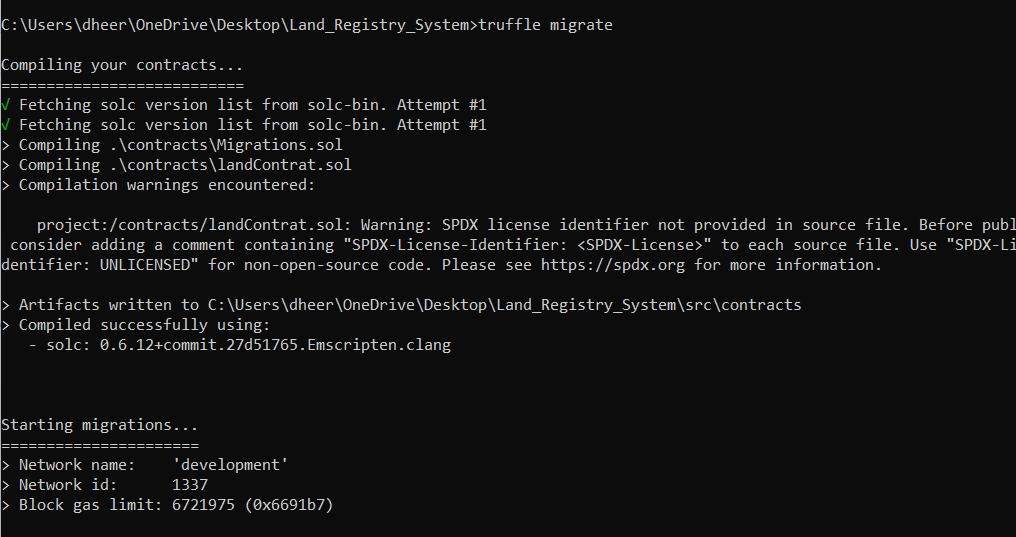
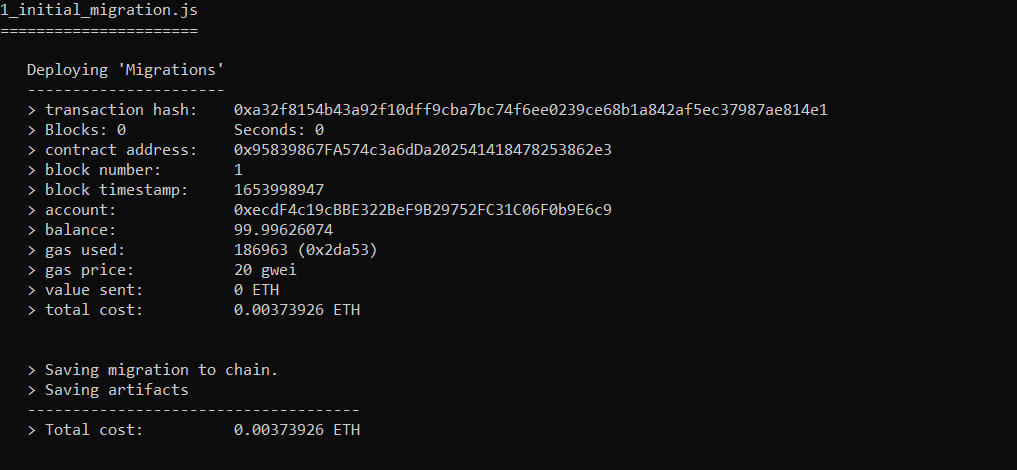
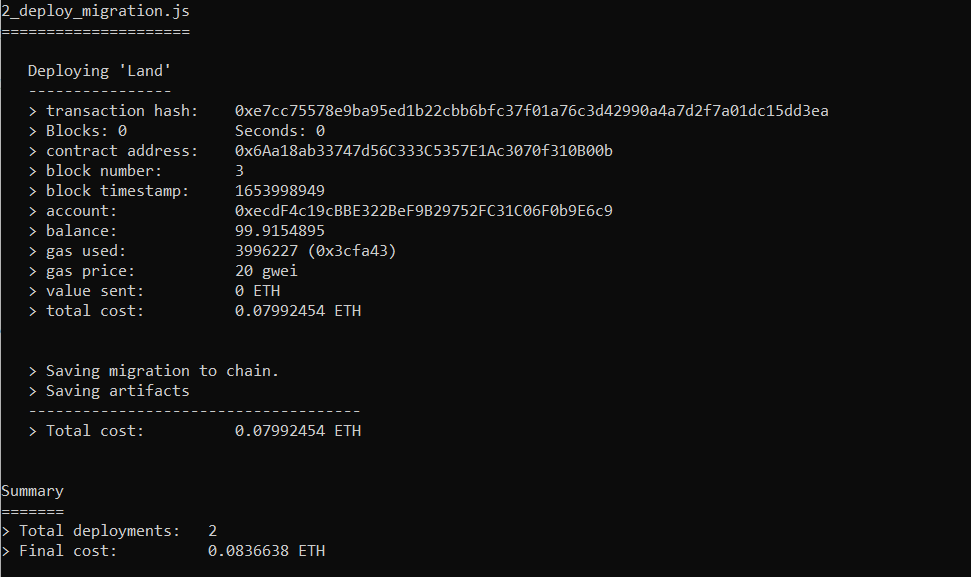
* Open the terminal and deploy smart contracts using truffle migrate.

Figure 8.5: Contract Deployment using Truffle

* Copy deployed contract address to **constant/constants.dart**.
* Create mapbox api key from https://www.mapbox.com/ and add it with 'mapBoxApiKey' constant.dart file.
* Create nftstorage api key from https://nft.storage/ and add it with ‘nftStorageApiKey’ constant.dart file.

**8.5 Running the DAPP**

* Run flutter web app by giving the command in terminal:

*flutter pub get*

*flutter run -d web-server --web-port 5555*

* Open localhost:54582 on your browser.
* **Home Page:** This is the home page of land registry system where contract owner, land

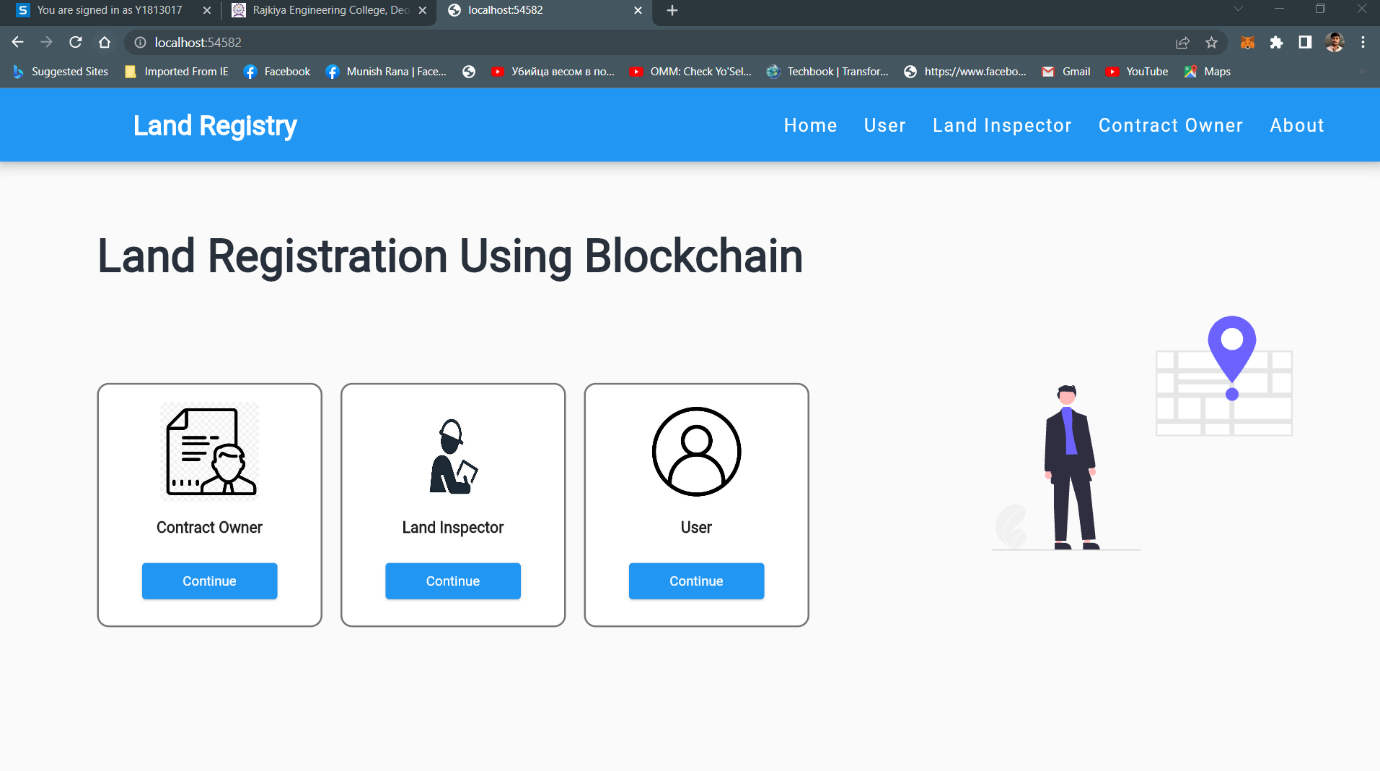
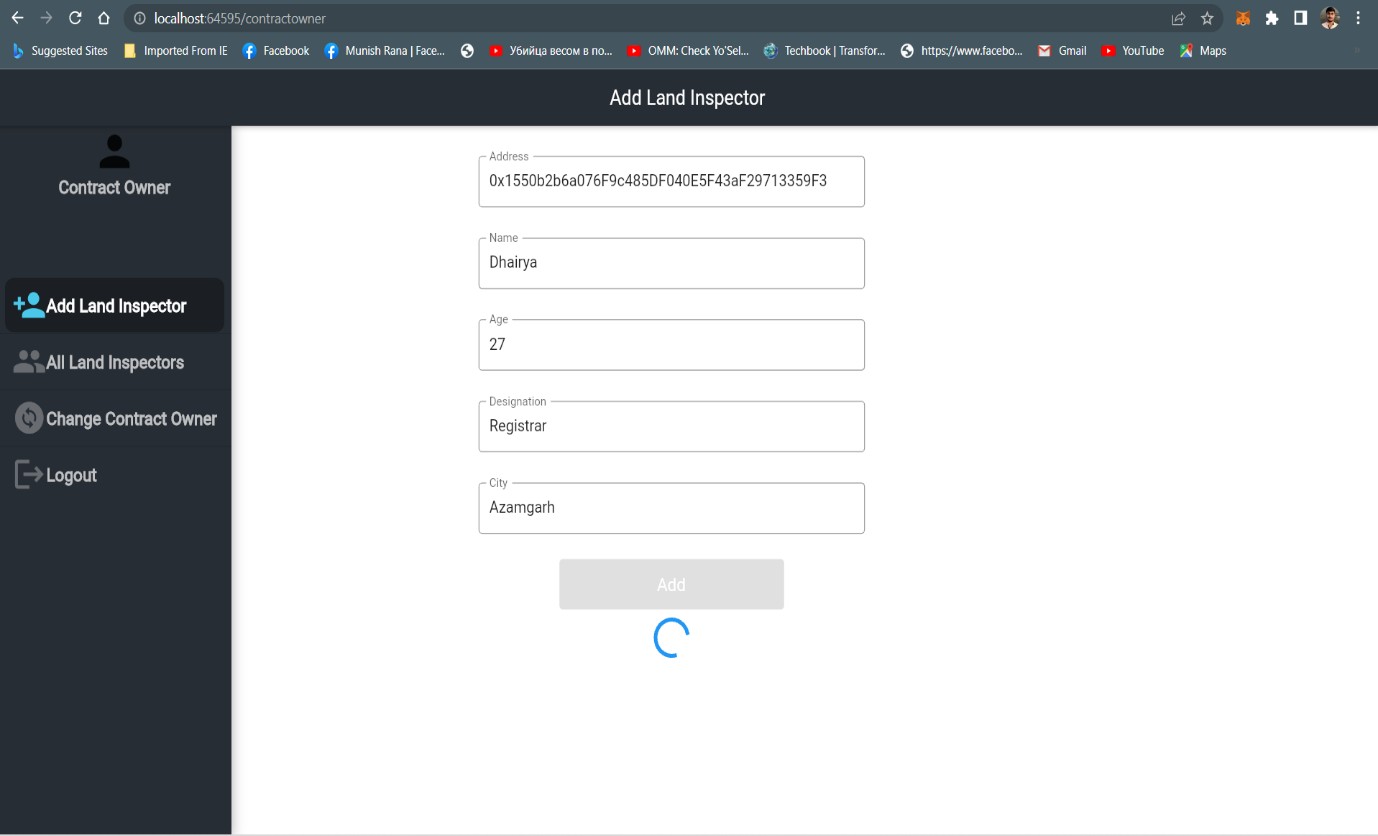
 inspector and user interface are available.

Figure 8.6: The running DApp

* **Contract Owner Dashboard:** Contract owner is known as government authority who adds land inspector as registrar/sub-registrar.

Figure 8.7: Contract Owner Dashboard

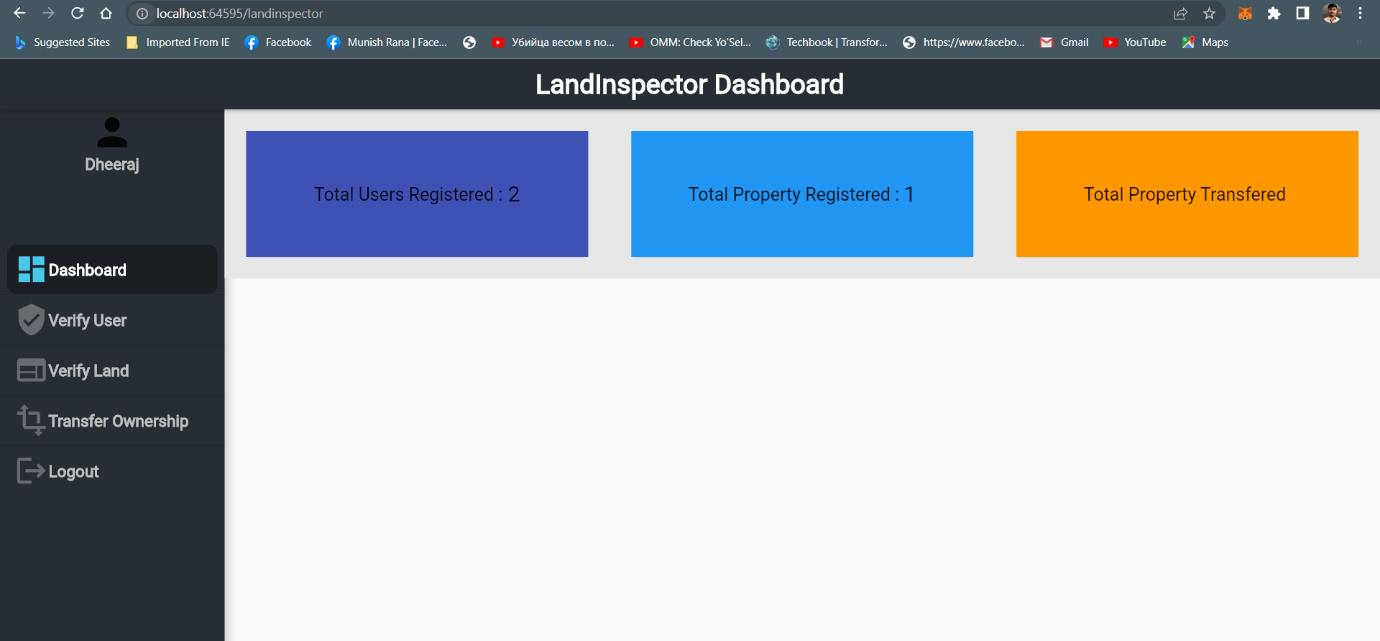
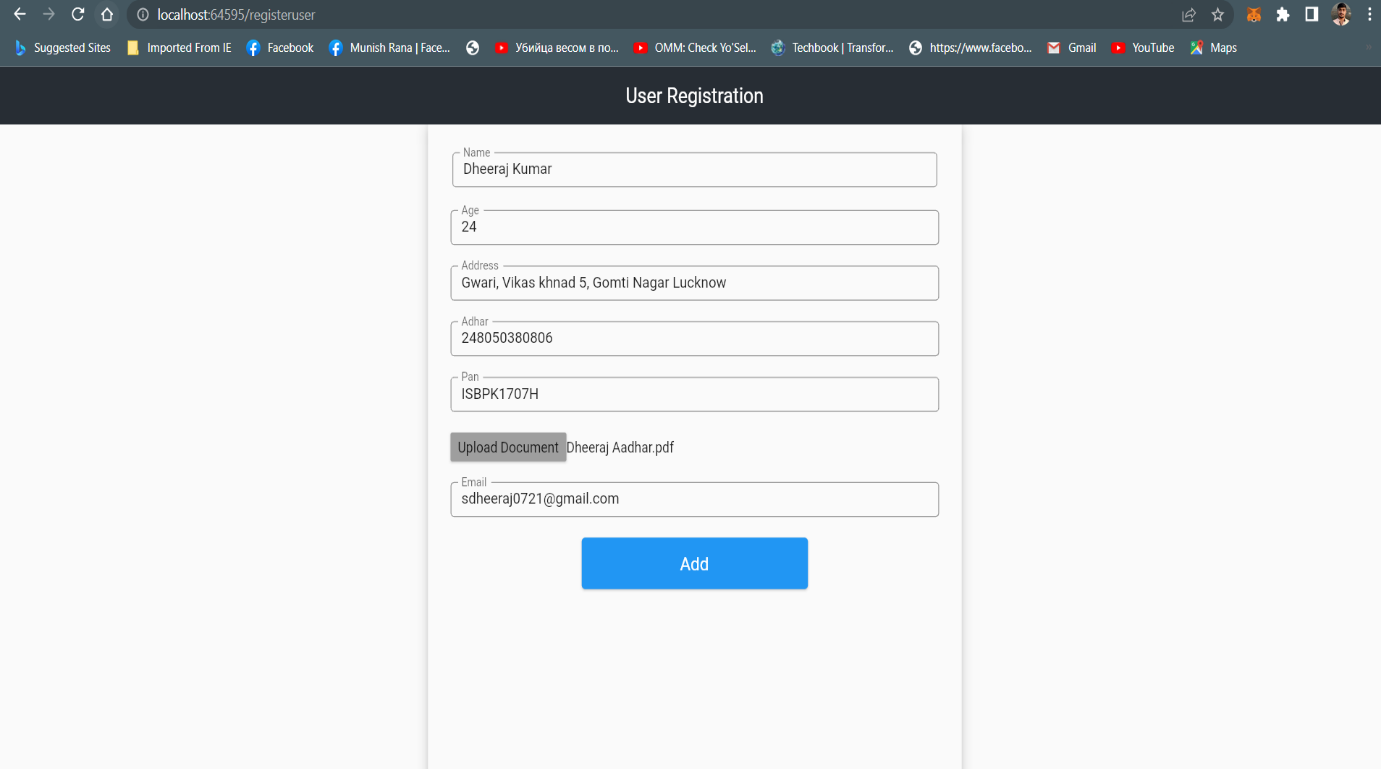
* **Land Inspector Dashboard:** Land inspector login using his private key which is added contract owner and go to his dashboard.

Figure 8.8: Land Inspector Dashboard

* **User Registration:** Users can register with their personal details and then login with private key.

****Figure 8.9: User Registration

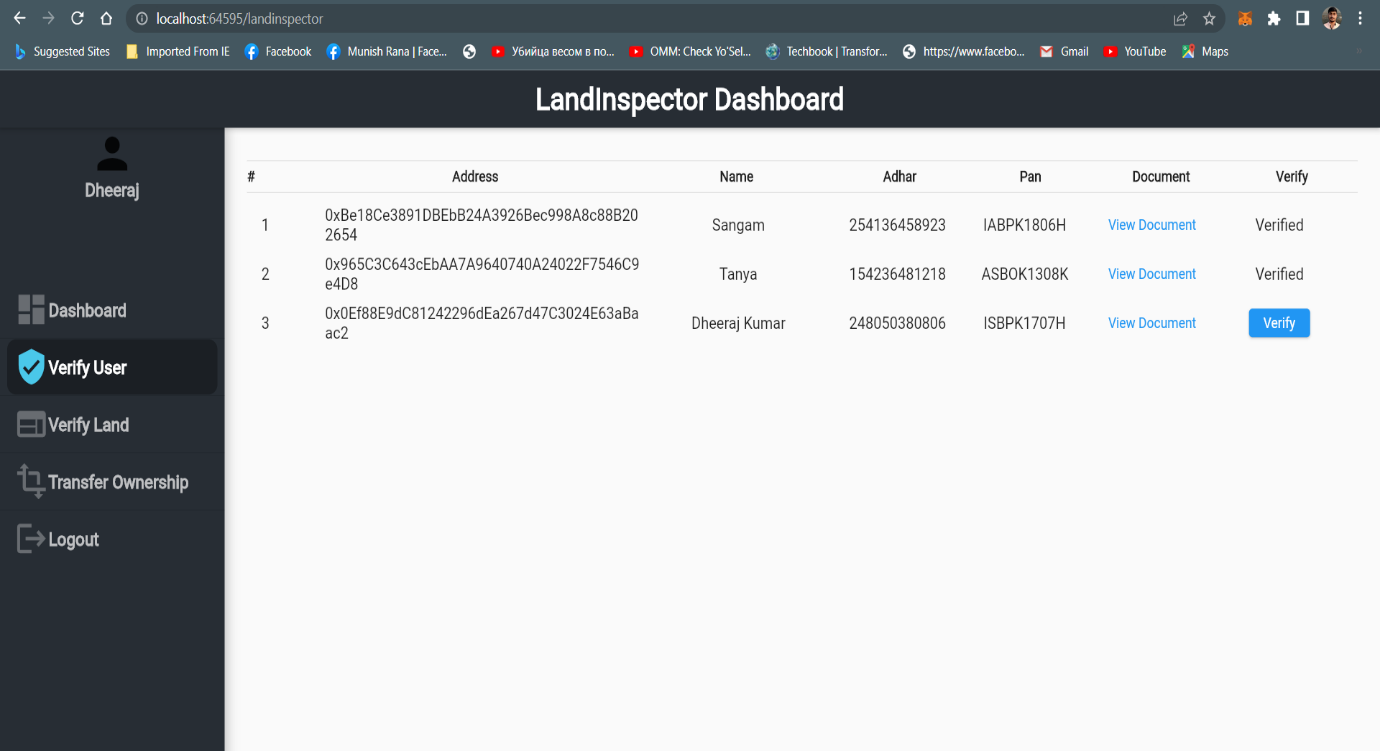
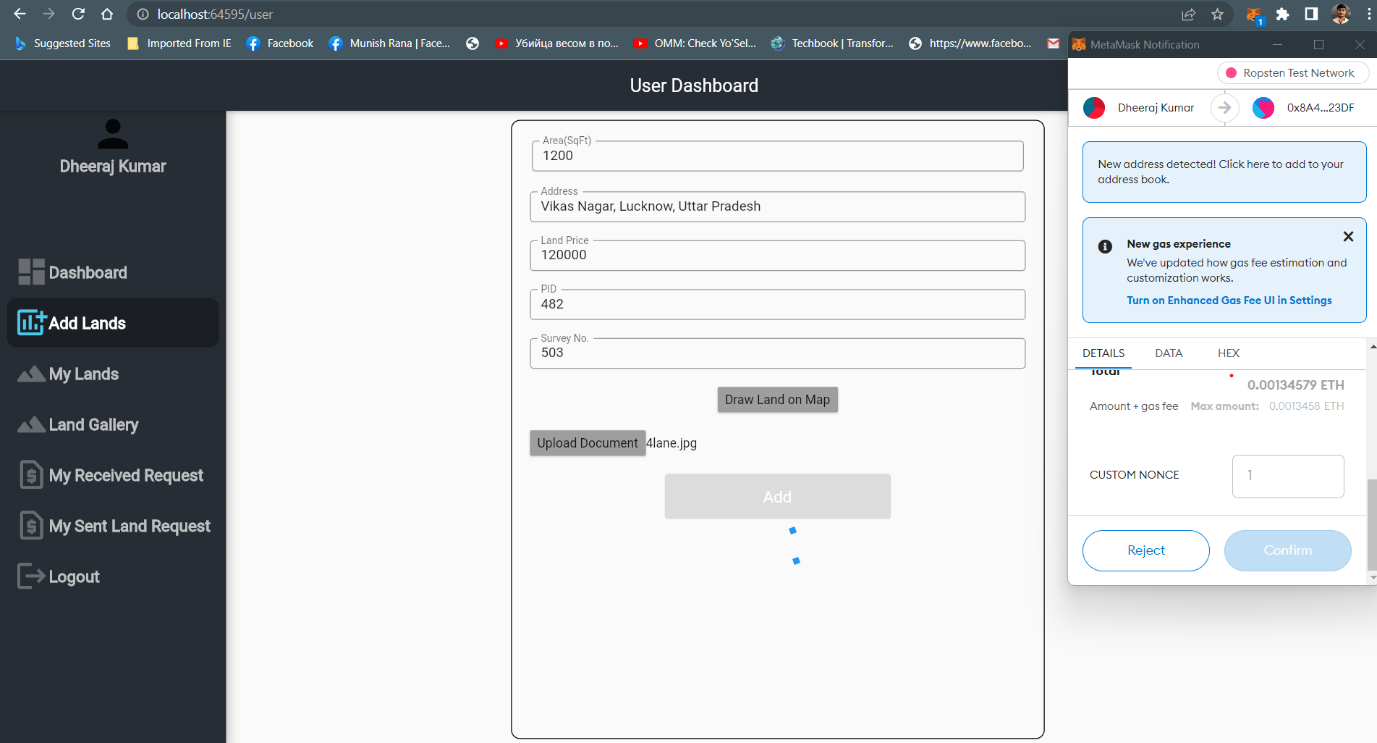
* **User Verification:** When user adds his details for registration then land inspector verifies his details for authentication, after that user can login with his private key.

Figure 8.10: User Verification

* **Add Land:** After successfully login, user can add land for sell.

Figure 8.11: Add Land by User

* **Land Verification:** When user adds land in his dashboard then land inspector verifies the land details of ownership. After that user makes land for sell.

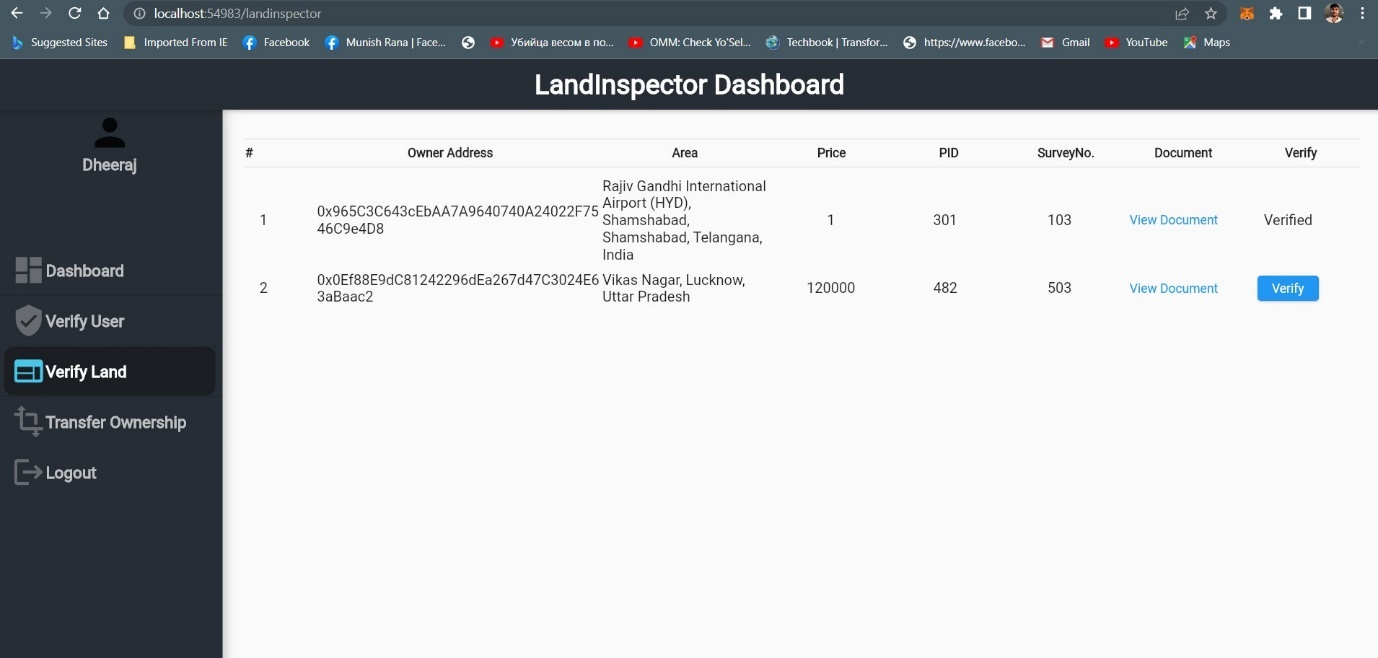
****

Figure 8.12: Land Verification by Land Inspector

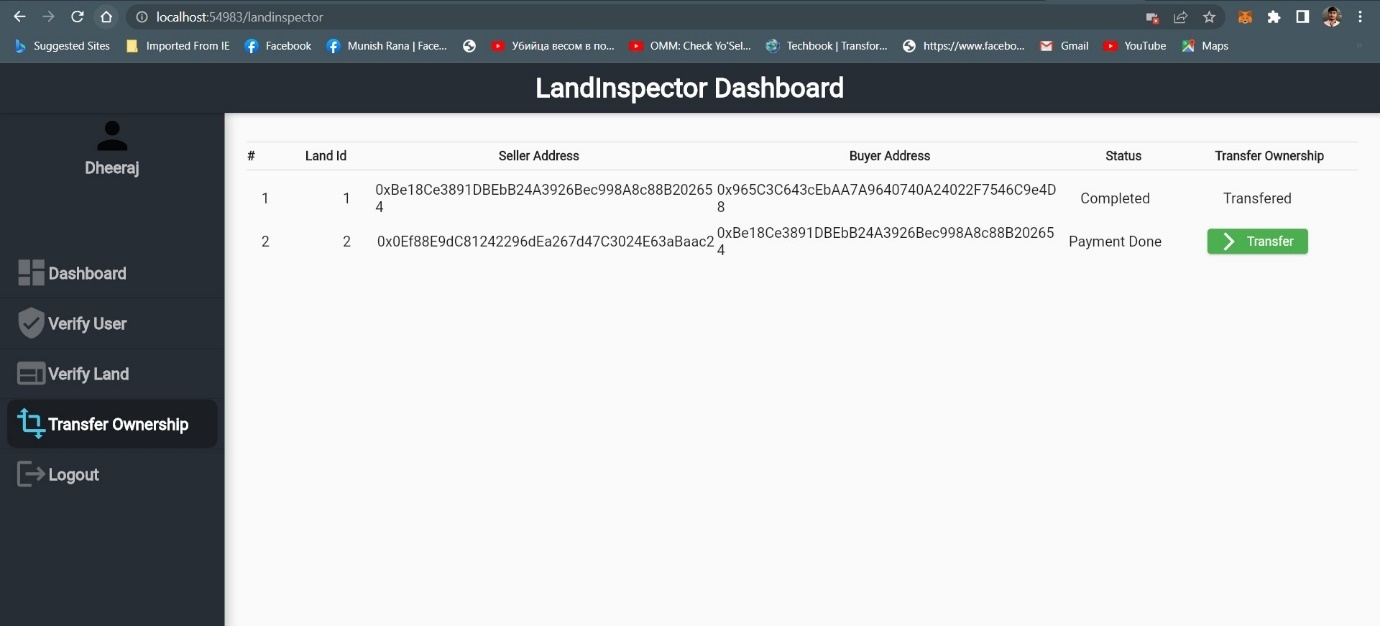
* **Send request by buyer for land:** Another user (buyer) can send request to seller for buying land.
* **Accept request by seller:** Seller accepts the request sent by user (buyer). Then buyer makes payment to the seller.
* **Ownership Transfer:** Land inspector verifies the payment status and then transfer the ownership to the buyer from the seller.

Figure 8.13: Transfer Ownership by Land Inspector

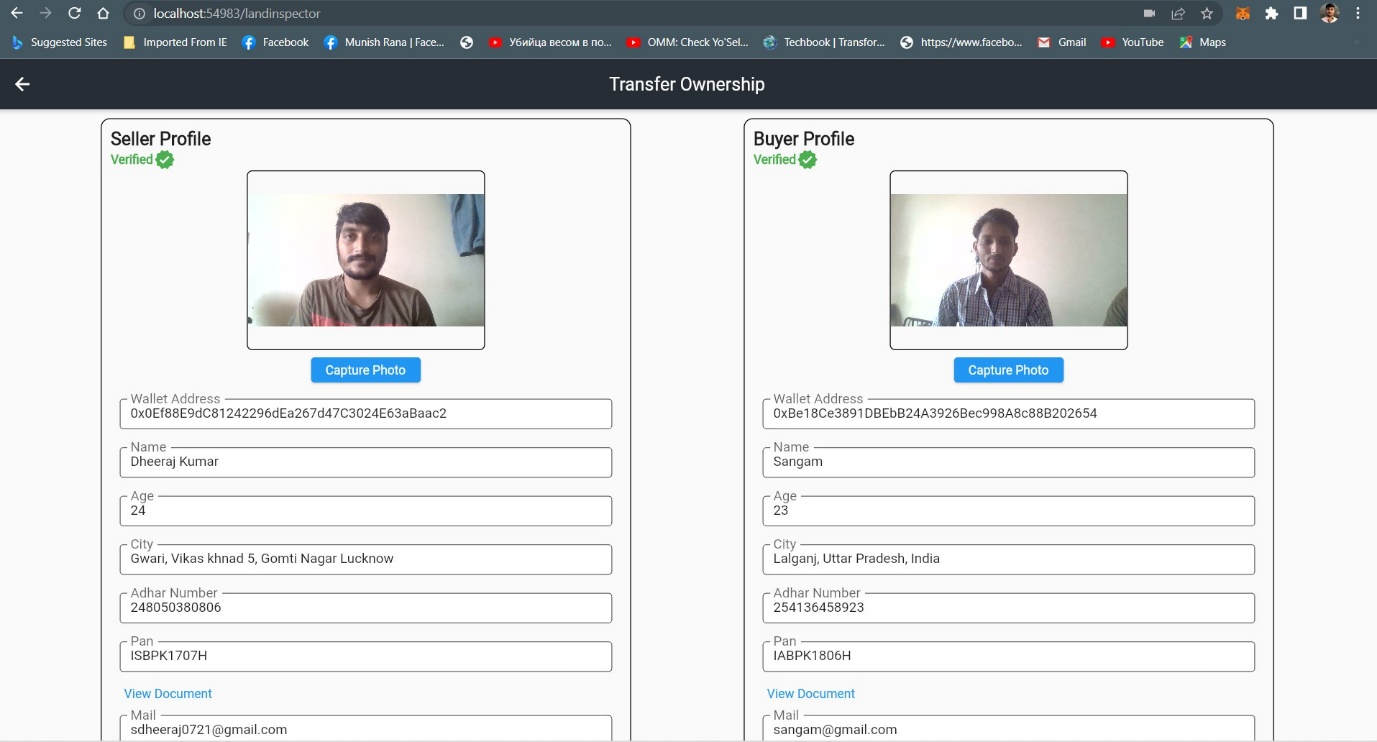
****

Figure 8.14: Transferring ownership to the buyer

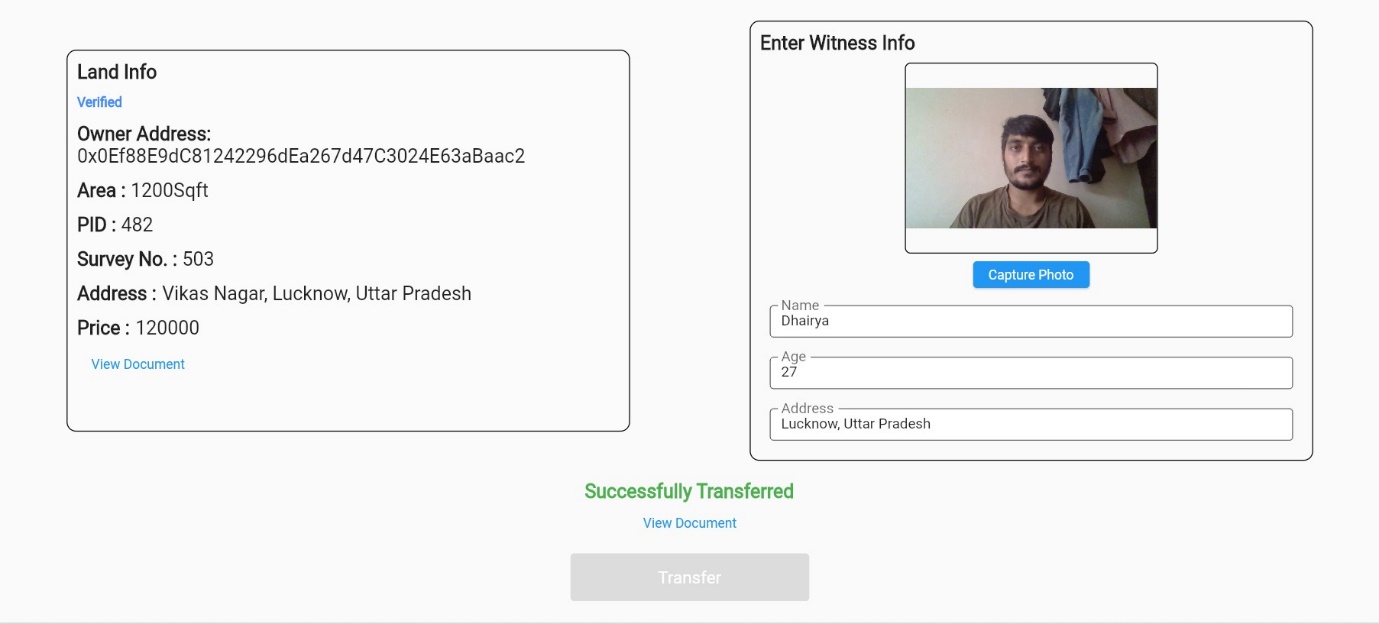
****

Figure 8.15: Successfully transferred ownership to the buyer

**Chapter 9**

**Conclusion and future work**

After exploring the potential of blockchain technology in the field of the land registry, it’s proven as a great move to eliminate land disputes, reduce risk & corruption and make execution faster, highly secure transactions due to hashed blocks have made blocks immutable and secure. All the transactions are approved and verified using proof-of-work consensus algorithm. The steps involved in the process of land registration are shown in flowchart in the paper. Blockchain technology has huge potential to replace paper money with digital currency which offers secured environment to perform transactions and in return boost economic growth. Blockchain features such as immutability, transparency, authenticity, confidentiality, data integrity, non-repudiation, and data availability can help a most of the developing countries like India to turn into a cash-less economy. We have discussed various views on which this project relies, in comparison to existing system, but the way of handling the events seems to be complicated sometimes using these technologies. It is because handling of various modules like web3Dart, smart contract, flutter, simultaneously, but finally this project reduces the aggregation cost, opportunity cost, management cost and solved the interoperability problem, transparency problem, security issues and land registration problems etc. These types of advantages will lead to the betterment of both user and government. This project will also cut the intermediary costs to a great extent. Several types of limitations can be found in the land registration system like data transparency which needs to be improved for the betterment of user as well as land inspector. This project has proposed a solution to the limitations of land registration through the help of Blockchain Technology. It has been observed that majority of the problems present in land registration because of its centralized architecture. So, switching from centralized architecture to decentralized architecture could be a potential solution for addressing the problems. Further, some other evolving technologies like inter-planetary file system (IPFS) and non-fungible tokens (NFT) can also be used to provide more security, resiliency and flexibility as well as ensuring the interests and engagements of the entities present in land registration system. As this project is developed on Ethereum platform, there is a lot more to do with it like a feature of using renting property.

**References**

1. Nandi, Meghali, Rajat Kanti Bhattacharjee, Amrit Jha, and Ferdous A. Barbhuiya. "A secured land registration framework on Blockchain." In *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, pp. 130-138. IEEE, 2020.
2. Pongnumkul, Suporn, Chanop Khonnasee, Swiss Lertpattanasak, and Chantri Polprasert. "Proof-of-concept (PoC) of land mortgaging process in blockchain-based land registration system of Thailand." In *Proceedings of the 2020 The 2nd International Conference on Blockchain Technology*, pp. 100-104. 2020.
3. Kaczorowska, Maria. "Blockchain-based land registration: Possibilities and challenges." *Masaryk University Journal of Law and Technology* 13, no. 2 (2019): 339-360.
4. Soner, Swapnil, Ratnesh Litoriya, and Prateek Pandey. "Exploring blockchain and smart contract technology for reliable and secure land registration and record management." *Wireless Personal Communications* 121, no. 4 (2021): 2495-2509.
5. Barbieri, Maurice, and Dominik Gassen. "Blockchain-can this new technology really revolutionize the land registry system." In *Responsible Land Governance: Towards an Evidence Based Approach: Proceedings of the Annual World Bank Conference on Land and Poverty*, pp. 1-13. 2017.
6. Yapicioglu, Balkiz, and Rebecca Leshinsky. "Blockchain as a tool for land rights: Ownership of land in Cyprus." *Journal of Property, Planning and Environmental Law* (2020).
7. Khalid, Muhammad Irfan, Jawaid Iqbal, Ahmad Alturki, Saddam Hussain, Amerah Alabrah, and Syed Sajid Ullah. "Blockchain-Based Land Registration System: A Conceptual Framework." *Applied Bionics and Biomechanics* 2022 (2022).
8. Benbunan-Fich, Raquel, and Arturo Castellanos. "Digitization of land records: From paper to blockchain." (2018).
9. Shinde, Disha, Snehal Padekar, Siddharth Raut, Abdul Wasay, and S. S. Sambhare. "Land Registry Using Blockchain-A Survey of existing systems and proposing a feasible solution." In *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, pp. 1-6. IEEE, 2019.
10. Stefanović, Miroslav, D. Pržulj, S. Ristic, and Darko Stefanović. "Blockchain and land administration: Possible applications and limitations." In *Proceedings of the 5th International Scientific Conference on Contemporary Issues in Economics, Business and Management EBM 2018*. 2018.
11. Islam, Md Sakibul, Fahmid Shahriar Iqbal, and Muhaimenul Islam. "A novel framework for implementation of land registration and ownership management via blockchain in bangladesh." In *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 859-862. IEEE, 2020.
12. Alam, Kazi Masudul, JM Ashfiqur Rahman, Anisha Tasnim, and Aysha Akther. "A blockchain-based land title management system for Bangladesh." *Journal of King Saud University-Computer and Information Sciences* (2020).
13. Yadav, Amrendra Singh, and Dharmender Singh Kushwaha. "Digitization of land record through blockchain-based consensus algorithm." *IETE Technical Review* (2021): 1-18.
14. Yadav, Amrendra Singh, and Dharmender Singh Kushwaha. "Query Optimization in a Blockchain-Based Land Registry Management System." *Ingénierie des Systèmes d Inf.* 26, no. 1 (2021): 13-21.
15. Sladić, Goran, Branko Milosavljević, Siniša Nikolić, Dubravka Sladić, and Aleksandra Radulović. "A blockchain solution for securing real property transactions: a case study for Serbia." *ISPRS International Journal of Geo-Information* 10, no. 1 (2021): 35.
16. Lemieux, Victoria L. "A typology of blockchain recordkeeping solutions and some reflections on their implications for the future of archival preservation." In *2017 IEEE international conference on big data (Big Data)*, pp. 2271-2278. IEEE, 2017.
17. Lemieux, Victoria L., Darra Hofman, Danielle Batista, and Alysha Joo. "Blockchain technology & recordkeeping." *ARMA International Educational Foundation* (2019).
18. Biswas, Milon, Jabed Al Faysal, and Kazi Asif Ahmed. "LandChain: A Blockchain Based Secured Land Registration System." In *2021 International Conference on Science & Contemporary Technologies (ICSCT)*, pp. 1-6. IEEE, 2021.
19. Vinothiyalakshmi, P., C. Muralidharan, Y. Mohamed Sirajudeen, and R. Anitha. "Digitized Land Registration Using Blockchain Technology." In *Blockchain Technology*, pp. 73-86. CRC Press, 2022.
20. Ooi, Vincent, Soh Kian Peng, and Jerrold Soh. "Blockchain land transfers: Technology, promises, and perils." *Computer Law & Security Review* 45 (2022): 105672.
21. McMurren, Juliet, Andrew Young, and Stefaan Verhulst. "Addressing transaction costs through blockchain and identity in swedish land transfers." *Blockchain Technologies for Social Change, GovLab* (2018).
22. Mintah, Kwabena, Festival Godwin Boateng, Kingsley Tetteh Baako, Eric Gaisie, and Gideon Kwame Otchere. "Blockchain on stool land acquisition: Lessons from Ghana for strengthening land tenure security other than titling." *Land Use Policy* 109 (2021): 105635.
23. Yadav, Amrendra Singh, and Dharmender Singh Kushwaha. "Blockchain-based digitization of land record through trust value-based consensus algorithm." *Peer-to-Peer networking and applications* 14, no. 6 (2021): 3540-3558.
24. Schmidt, Kai, and Philipp Sandner. "Solving challenges in developing countries with blockchain technology." *FSBC Work. Pap* (2017).
25. Beznosov, A. G., E. A. Skvortsov, and E. G. Skvortsova. "Prospects for application of blockchain technology in land administration." In *IOP Conference Series: Earth and Environmental Science*, vol. 699, no. 1, p. 012045. IOP Publishing, 2021.
26. Amadi-Echendu, Anthea P. "Using blockchain technology to facilitate property transactions." *SA Journal of Information Management* 23, no. 1 (2021): 7.
27. Alketbi, Ahmed, Qassim Nasir, and Manar Abu Talib. "Blockchain for government services—Use cases, security benefits and challenges." In *2018 15th Learning and Technology Conference (L&T)*, pp. 112-119. IEEE, 2018.
28. Gupta, Nidhi, Manik Lal Das, and Sukumar Nandi. "LandLedger: blockchain-powered land property administration system." In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6. IEEE, 2019.
29. Wüst, Karl, and Arthur Gervais. "Do you need a blockchain?." In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 45-54. IEEE, 2018.
30. Miscione, Gianluca, Christine Richter, and Rafael Ziolkowski. "Authenticating Deeds/Organizing Society Considerations for Blockchain-Based Land Registries." *DeVries WT et al.(2020)“Responsible and Smart Land Management Interventions: An African Context” CRC Press (Taylor & Francis)* (2020).
31. V. L. Lemieux, "Evaluating the use of blockchain in land transactions: An archival science perspective", European Property Law Journal, vol. 6, no. 3, pp. 392-440, 2017.
32. G. Chavez-Dreyfuss, "Sweden tests blockchain technology for land registry", Reuters, vol. 16, June 2016.
33. A. Anand, M. McKibbin and F. Pichel, "Colored coins: Bitcoin blockchain and land administration", Annual World Bank Conference on Land and Poverty, 2016.
34. "Indian land system on blockchain", [online] Available: https://qz.com/india/1325423/ indias-andhra-state-is-using-blockchain-to-build-capital-amaravati/.
35. N. Atzei, M. Bartoletti and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)", International Conference on Principles of Security and Trust, pp. 164-186, 2017.
36. A. Baliga, "Understanding blockchain consensus models", Persistent, 2017.
37. J. M. Graglia and C. Mellon, "Blockchain and property in 2018: At the end of the beginning", Innovations: Technology Governance Globalization, vol. 12, no. 1-2, pp. 90-116, 2018.
38. S. Enemark, I. Williamson and J. Wallace, "Building modern land administration systems in developed economies", Journal of Spatial Science, vol. 50, no. 2, pp. 51-68, 2005.
39. N. S. Chiteji and F. P. Stafford, "Asset ownership across generations", 2000.
40. M. Pilkington, "11 blockchain technology: principles and applications", Research handbook on digital transformations, vol. 225, 2016.