Land registration system using Permissioned Blockchain

Ali Khatami
Department of Computer Science and Engineering
International Islamic University Chittagong
Chattogram , Bangladesh
aliiuc86@gmail.com

Abdullahil Kafi
Department of Computer Science and Engineering
International Islamic University Chittagong
Chattogram , Bangaldesh
abkafi@gmail.com

Mohammad Rokib Khan
Department of Computer Science and Engineering
International Islamic University Chittagong
Chattogram , Bangladesh
rokibk548@gmail.com

Md Rafidul Islam
Department of Computer Science and Engineering
International Islamic University Chittagong
Chattogram , Bangaldesh
rafidulislam3549@gmail.com

***Abstract*—This project aims to create a decentralised land registration system for Bangladesh. The solution uses the Raft consensus algorithm, which helps to make the network distributed by allowing each organisation to have its own peers. Hyperledger Fabric, a well-known permissioned blockchain technology, is used to store and secure important registration data and land details. The Raft ordering service, along with the permissioned nature of Hyperledger Fabric, ensures that citizens' data cannot be changed. To address worries about transaction fees, we've taken steps to prevent regular people from being burdened, making sure everyone can use the system. We've also planned for the system's growth as the population increases. We're working to clear up legal uncertainties related to blockchain in countries like Bangladesh, where people might have concerns about using cryptocurrency. In tests, users were able to log in, submit new land registration applications, and securely store details on the ledger. Ministry of Land administrators can review and approve applications, and Land Revenue Office administrators can smoothly register approved lands. If someone owns land registered on the blockchain, they can transfer or sell ownership. We use Hyperledger Fabric, Express.js, Node.js, Go, and MongoDB to ensure data security, access control, and a strong framework. In short, this project makes land registration easier and tackles challenges in decentralised systems, providing a safe, transparent, and efficient way to manage land records in Bangladesh.**

***Index Terms*—Blockchain, Hyperledger Fabric, Land Registration, Consensus Algorithm, Membership Service Providers, Chaincode.**

I. Introduction

In our effort to create a better land registration system in Bangladesh, we're setting up a decentralised solution using the Raft [1] algorithm and Hyperledger Fabric [2] technology for safety and efficiency. We want everyone using our system to be treated fairly. Despite some legal questions about using blockchain [3] in Bangladesh, our main goal is to build a safe and effective system that secures land records. To make it happen, we'll build the user application using the Express.js [4] framework of Node.js [5]. For logging in, we'll use MongoDB. This way, things will be easy for everyone using the system.

Regular ways of registering land have problems like data being at risk, people paying fees for transactions, struggling to handle more people, not being sure about the law with blockchain, and everything being too centralized. Our project uses a special blockchain for land registration that needs permission. We want to fix these issues by making sure data can't be changed, reducing fees for people, handling more users, clearing up legal questions, and using a decentralised system. Our goal is to change how land registration works, making it more trustworthy, efficient, and fair for everyone.

Some bad people work with helpers and dishonest officials to make fake papers and take control of land by force. This leaves the real owners in long fights in court to get their land back. These land sharks use bribes to change papers, selling the land to people they don't know, causing big legal problems between the true owner and the new buyer with fake documents. The original landowners might make many agreements, taking money from different buyers using a special paper called a pledge deed (Binah Dalil), leading to common court problems. Fixing these issues usually means making friendly agreements or going to the police. But dealing with the police in the legal system brings more money and time problems, lasting for months or even years.

The main objective of this project are:

- To create a system that stops bad people from making fake papers to take over land, make a secure and easy way to check who owns land, and keep clear records of land deals to stop dishonest officials from cheating.
- To implement the Raft ordering service,which is better than Kafka and other ordering services
- To ensure that citizens' data is completely immutable
- To prevent common individuals from being burdened by transaction fees

- To prevent a gradual increase in the load on the blockchain system and transactions due to population growth
- Avoid the use of cryptocurrency to prevent violation of laws regarding cryptocurrency in many countries like Bangladesh

In Chapter 2 of our report we discussed the the related work, and at chapter 3 we focused on methodology we used , and at chapter 4 we focused on Result and discussion and at last chapter 5 we have discussed the conclusion and our future work.

II. Related Work

At [6] the proposal suggests leveraging blockchain, particularly smart contracts, to document property transactions encompassing sales, inheritance, court orders, and land acquisition. Successfully addressed the issue of implementing both public and private blockchain on land registration systems in India and successfully solved it. Their limitation is that the load on the blockchain system will gradually increase, and the transactions will increase because of population growth. In the paper [7]they implemented their work on Hyperledger Fabric v1.2 and assessed performance on AWS t2.large with 2 vCPUs and 8 GB of memory, considering transaction density and node failures. AM2 unifies channels, employing a single chaincode and island-wide land ledger, outperforming AM1's separate channels. But using Kafka for ordering services doesn't decentralise nodes when operated by different organisations, as all connect to a single-controlled Kafka cluster. At [8] they have used the Bitcoin blockchain. technology. They achieved heightened safety, security for citizen data, and improved transparency, ensuring traceability of information. But the limitation is that citizen data can be altered before being stored on immutable blockchain storage. In the paper [9] author's work is based on Ethereum-deployed smart land title contracts using Solidity. They used Ganache-cli for local testing, Remix as an online IDE, and Metamask for browser blockchain interaction. They efficiently manage land ownership, ensure traceability, and minimise travel time and costs. But here, the transaction fee varies with the ETH value, which will not be acceptable for common people.In the paper [10] hey built their system on the Ethereum blockchain network. It cuts real estate authentication expenses, speeds up transactions, and eliminates third-party verification. Their limitation is that they are uncertain about how blockchain legality impacts practical use for businesses and organisations.

III. Methodology

A. Tools used

The tools used are:

TABLE I
Tools used for our project work

Technology	Components
Backend Server:	Node.js (Express)
Fabric API:	Node.js
Chaincode:	Go
Authentication Database:	MongoDB
Data storage Databse:	LevelDB
API Testing:	Postman
Testing with frontend:	ngrok
Consensus Algorithm:	Raft
Certificate format:	X.509
Hashing Algorithm:	SHA-256

B. Algorithms used

Hashing Algorithm:

```
###############################################################################
# BCCSP (BlockChain Crypto Service Provider) section is used to select which
# crypto library implementation to use
###############################################################################
bccsp:
    default: SW
    sw:
        hash: SHA2
        security: 256
        filekeystore:
            # The directory used for the software file-based keystore
            keystore: msp/keystore

###############################################################################
```

Fig. 1. Hashing Algorithm.

The algorithm used for computing the hash values encoded into the blocks of the blockchain.

Consensus Algorithm

```
# SampleDevModeEtcdRaft defines a configuration that differs from the
# SampleDevModeSolo one only in that it uses the etcd/raft-based orderer.
SampleDevModeEtcdRaft:
    <<: *ChannelDefaults
    Orderer:
        <<: *OrdererDefaults
        OrdererType: etcdraft
        Organizations:
            - <<: *SampleOrg
              Policies:
                  <<: *SampleOrgPolicies
                  Admins:
                      Type: Signature
                      Rule: "OR('SampleOrg.member')"
```

Fig. 2. Orderer Consensus

Transactions must be written to the ledger in the order in which they occur, even though they might be between different sets of participants within the network. For this to happen, the order of transactions must be established and a method for rejecting bad transactions that have been inserted into the ledger in error (or maliciously) must be put into place.

```
# SampleAppChannelEtcdRaft defines an application channel configuration
# that uses the etcd/raft-based orderer.
SampleAppChannelEtcdRaft:
    <<: *ChannelDefaults
    Orderer:
        <<: *OrdererDefaults
        OrdererType: etcdraft
        Organizations:
            - <<: *SampleOrg
              Policies:
                  <<: *SampleOrgPolicies
                  Admins:
                      Type: Signature
                      Rule: "OR('SampleOrg.member')"
```

Fig. 3. Channel Consensus

There are many ways to achieve it, each with different trade-offs. Hyperledger Fabric has been designed to allow network starters to choose a consensus mechanism that best represents the relationships that exist between participants. The default mechanisms previously used were SOLO, and KAFKA. But the latest version of Hyperledger Fabric uses RAFT as the default consensus mechanism which is more efficient than the previous ones.

C. Network Deployment Steps:

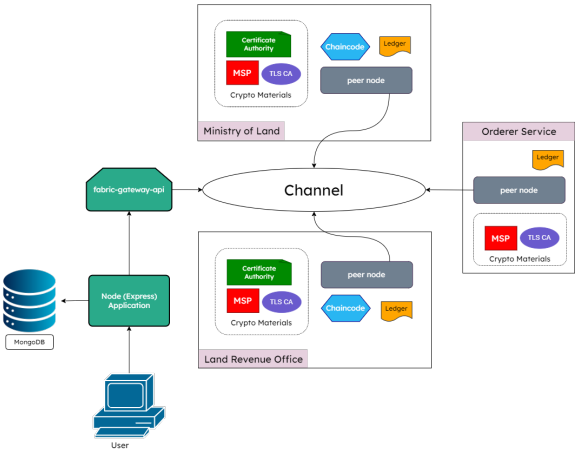

Fig. 4. Network Architecture

1. Fabric Certificate Authority Client (CA client)

The deployment sequence involves starting with the Fabric Certificate Authority Client (CA client), responsible for creating TLS certificates and Membership Service Providers (MSP) for every peer, orderer organisation, and any associated nodes. These credentials are organised into distinct folders for each entity.TLS ensures secure communication and facilitates the handshake process between different organizations.The MSP folder contains certificates, public-private keys, and signed certificates for each organization. When users from any organisation register and enrol via their CA server, they must interact through the CA **2. The Fabric Certificate Authority Server (CA server):** It initiates by creating a self-signed root certificate, endorsed by TLS CA, serving as the shared public key for all interacting clients. It generates the private key for the CA server, storing it in the MSP folder, and establishes the CA server administrator for user registration. **3. Orderer:**The Orderer's primary role is to sequence transactions into blocks, subsequently validated and committed by peers to their ledgers. Peers endorse client transactions, send them to the orderer, which organizes them into transaction blocks distributed to all peer nodes in the channel. **4. The Ministry of Land and Land Revenue Office** representing peer organizations, hosts the ledger and client application. Transaction requests from clients are sent to these organizations, where they undergo endorsement and processing.

D. Land Registration Flow:

Every transaction request initiated by the user follows this path:

The user (client, Ministry of Land and Land Revenue Office) submits a transaction request, directed to the Node.js API, which verifies the user's login status. The transaction is then transmitted to Fabric-API, which forwards it to the relevant Ministry of Land (peer) (LRO peer for LRO admin). The peer node at the Ministry of Land endorses the transaction, appends a signed certificate, and includes the TLS CA from its local directory. This endorsed package is sent to the orderer, which validates the transaction using the provided certificate. After confirming validity, the orderer organises the transaction into a block and sends it back to the peer nodes. The peer nodes add the new block to the ledger, completing the transaction process.

E. Step by step process of our Hyperledger Fabric based land registration system:

Login: Verified users will login with valid credentials into their account.

Dashboard: After login users can see their most recent 10 applications and some statistics about Pending, Rejected, Approved and Registered application.

Apply for Registration: After logging in users can apply for land registration from Register Application menu. They will provide valid information of the land with NEC. If no previous record is already been submitted for the same land the application will be stored to ledger for verification.

Approve application Ministry of Land: Any admin from Ministry of Land will see all pending application in their dashboard. They will verify the applications and approve if information is correct and reject if not.

Register application Land Revenue Office: Admin from Land Revenue office will see all application approved by Ministry of Land in their dashboard. They will verify the applications and register the land and generate Deed if information is correct and reject if not.

Transfer Ownership: Users can transfer the ownership of their registered land to other users. After successful transfer of ownership the land will again have to be verified from Ministry of Land and Land Revenue Office to register the land to new user.

F. Entity Relationship Diagram:

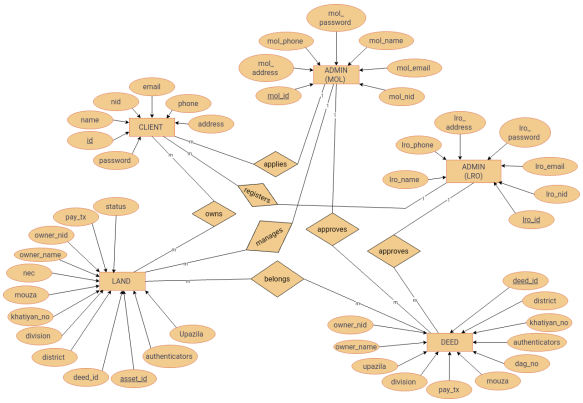

Fig. 5. Entity Relationship Diagram of our permissioned blockchain based Land registration system

G. Use case diagram

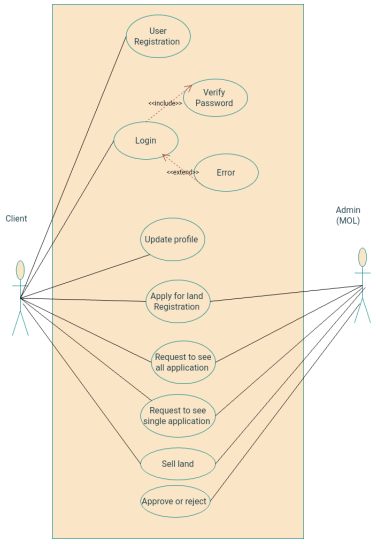

Fig. 6. Use case diagram of Client and Admin of Ministry of Land

Initially, all client requests, such as land registration applications, request to see all applications, requests to see single applications, and land sales, are routed to the Ministry of Land's admin. During this stage, the admin from the Land Revenue Office is unable to access these requests, creating a distinct separation in the process. The initial point of contact for these requests is the admin at the Ministry of Land, and access to these inquiries is restricted for the admin at the Land Revenue Office during this phase.

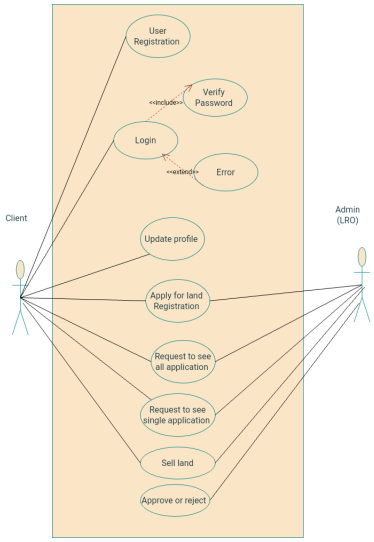

Fig. 7. Use case diagram of Client and Admin of Land Revenue Office

H. Sequence Diagrams:

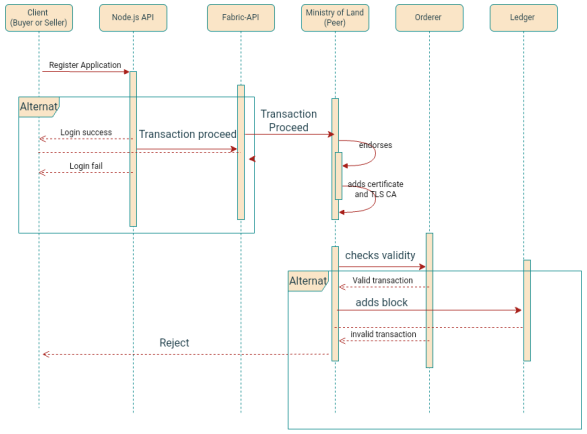

Fig. 8. Sequence diagram for Client side

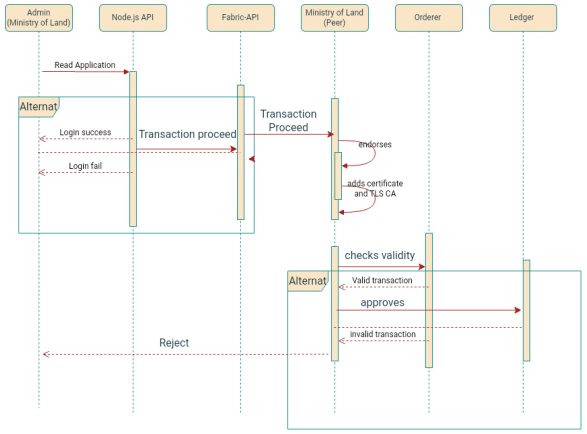

Fig. 9. Sequence diagram for Admin of Ministry of Land side

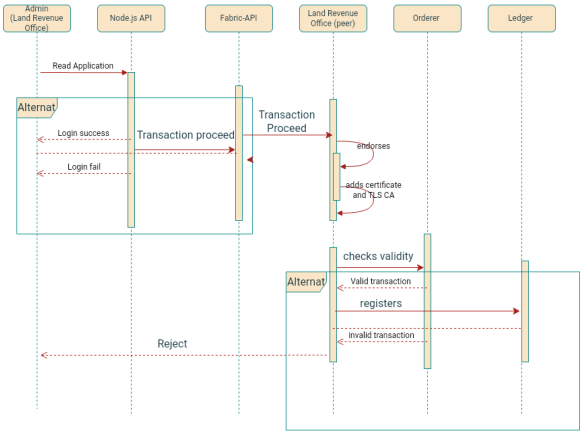

Fig. 10. Sequence diagram for Admin of Ministry of Land side

I. User Interface

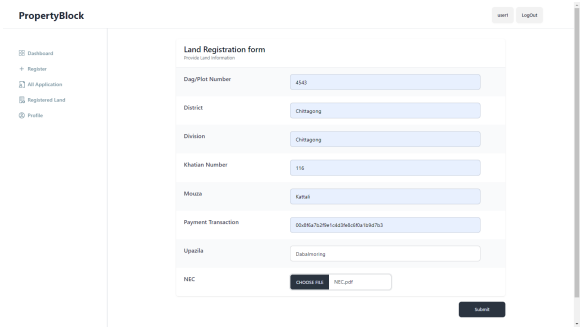

Fig. 11. User Interface for Land registration

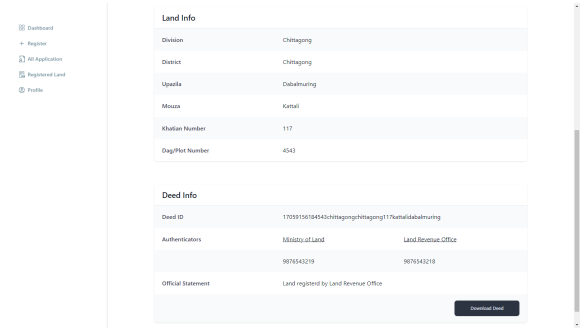

Fig. 12. Details of Application

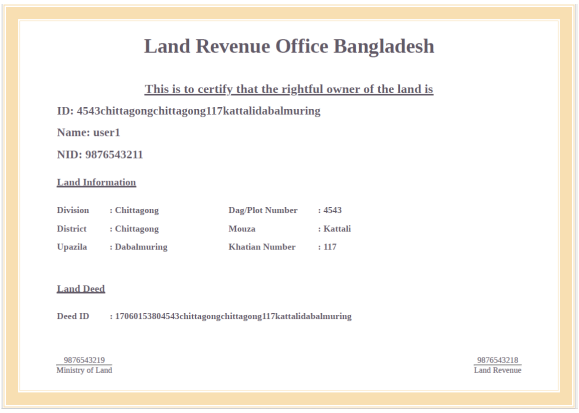

Fig. 13. Generated Certificate after approval by admins of Ministry of Land and Land Revenue Office.

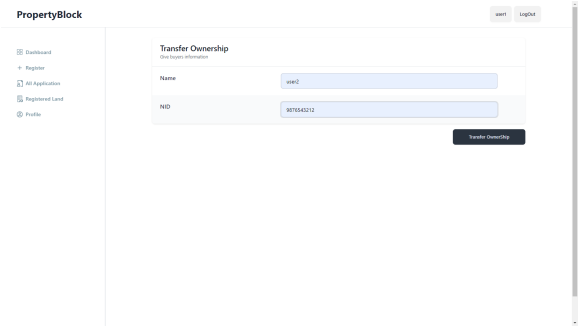

Fig. 14. User Interface for transfer ownership page

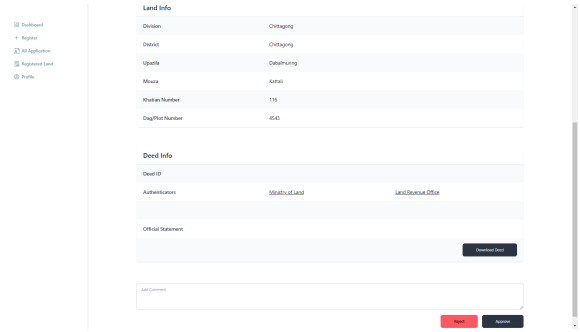

Fig. 15. User Interface for approve or reject applications by admin of MOL

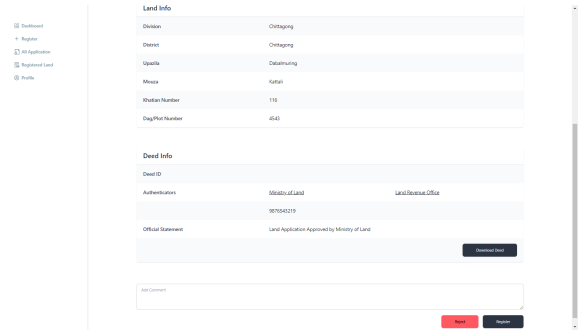

Fig. 16. User Interface for register or approve by admin of LRO

IV. Result and Discussion

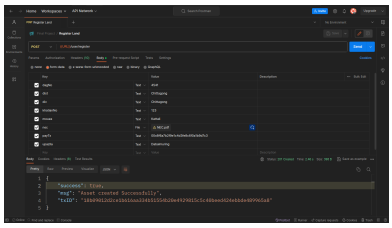

Fig. 17. API test for Registration Application

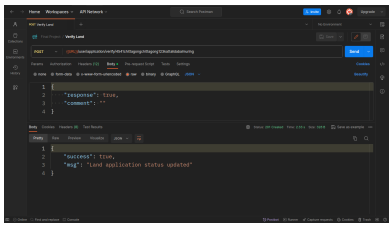

Fig. 18. API test for Land Application Verification

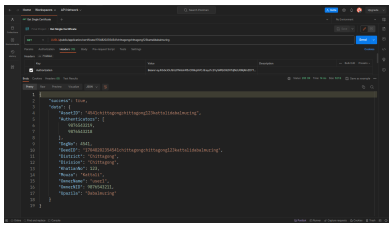

Fig. 19. API test for Deed of Registered Land

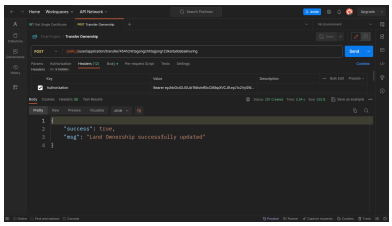

Fig. 20. API test for transfer Land Ownership

Fig. 21. Here we can see when we are going to change the data of the application we can see that the above error which is endorsement failure during invoke occurs and status shown 500 with message "Not authorized" shows

The conducted tests indicate that users are able to log in to their accounts, submit new land registration applications, and have the application details securely stored on the ledger. Administrators from the Ministry of Land have the capability to retrieve pending applications, review the provided information, and approve applications that meet the necessary criteria.

V. Conclusion

Our project successfully implemented a decentralized land registration system for Bangladesh using the Raft consensus algorithm and Hyperledger Fabric. It ensures secure and transparent land registration, addressing concerns about transaction fees and accessibility. Testing proved seamless application submission and review processes. In upcoming versions, users will use their own CA certificates for ledger access, eliminating MongoDB for authentication.

VI. Acknowledgment

First and foremost, we express gratitude to the Almighty Allah for His mercy, which enabled us to complete our thesis despite numerous obstacles. Secondly, we want to convey our appreciation to our supervisor, Abdullahil Kafi, for providing continuous guidance and support since the inception of our research.

VII. References

References

[1] Hyperledger Fabric, "The Ordering Service." https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html. Accessed: 06/12/2023.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, pp. 1–15, 2018.

[3] M. Xu, X. Chen, and G. Kou, "A systematic review of blockchain," *Financial Innovation*, vol. 5, no. 1, pp. 1–14, 2019.

[4] A. Mardan and A. Mardan, "Using express. js to create node. js web apps," *Practical Node. js: Building Real-World Scalable Web Apps*, pp. 51–87, 2018.

[5] M. F. Karagoez and C. Turgut, "Design and implementation of restful wireless sensor network gateways using node. js framework," in *European Wireless 2014; 20th European Wireless Conference*, pp. 1–6, VDE, 2014.

[6] V. Thakur, M. Doja, Y. K. Dwivedi, T. Ahmad, and G. Khadanga, "Land records on blockchain for implementation of land titling in india," *International Journal of Information Management*, vol. 52, p. 101940, 2020.

[7] D. Fernando and N. Ranasinghe, "Permissioned distributed ledgers for land transactions; a case study," in *Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum, Vienna, Austria, September 1–6, 2019, Proceedings 17*, pp. 136–150, Springer, 2019.

[8] N. Lazuashvili, A. Norta, and D. Draheim, "Integration of blockchain technology into a land registration system for immutable traceability: a casestudy of georgia," in *Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum, Vienna, Austria, September 1–6, 2019, Proceedings 17*, pp. 219–233, Springer, 2019.

[9] K. M. Alam, J. A. Rahman, A. Tasnim, and A. Akther, "A blockchain-based land title management system for bangladesh," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 3096–3110, 2022.

[10] L. Zhang, L. Ci, Y. Wu, and B. Wiwatanapataphee, "The real estate time-stamping and registration system based on ethereum blockchain," *Blockchain: Research and Applications*, p. 100175, 2023.