A Blockchain-Based Secure Land Registry System

Jay Mohan Anmolu
Department of Information Technology
School of Engineering and Technology
D.Y. Patil University, Ambi
Pune, India
jayanmolu14@gmail.com

Rushikesh Jeevan Asawale
Department of Information Technology
School of Engineering and Technology
D.Y. Patil University, Ambi
Pune, India
asawalerushikeshj@gmail.com

Soham Shankar Satardekar
Department of Information Technology
School of Engineering and Technology
D.Y. Patil University, Ambi
Pune, India
sohamsatardekar07@gmail.com

Shubham Suresh Nighot
Department of Information Technology
School of Engineering and Technology
D.Y. Patil University, Ambi
Pune, India
shubhamnighot414@gmail.com

Abstract

Land registration systems play a crucial role in establishing property ownership and ensuring secure transactions. However, traditional systems are largely centralized and paper-based, making them slow, inefficient, and vulnerable to fraud, data manipulation, and disputes. These limitations reduce transparency and weaken trust among stakeholders.

This paper presents a Blockchain-Based Secure Land Registry System that adopts a structured and secure approach to managing land records. The system is designed using modern web technologies to ensure data integrity, controlled access, and efficient transaction processing. It incorporates a user-friendly interface, a backend server, and an administrative verification module to handle key operations such as land registration, ownership verification, and property transfer.

The proposed approach is inspired by blockchain principles such as data immutability, transparency, and traceability. Each transaction is validated before being stored, ensuring consistency and accountability within the system. A transaction history mechanism is maintained to enable tracking of ownership changes and auditing of records.

The current implementation focuses on a secure digital architecture using React.js, Spring Boot, and MySQL, providing a scalable and efficient platform. Integration of a fully decentralized blockchain network and smart contract-based automation is considered as a potential enhancement for future development.

Overall, the system enhances security, improves efficiency, and builds trust in land management processes. It offers a practical and scalable solution for modern real estate systems, with strong potential for adoption in e-governance and smart city applications.

Keywords—Land Registry System, Secure Data Management, Spring Boot, React.js, MySQL, Data Integrity, Property Management, Web-Based Application, Authentication and Authorization, Transaction Management

1. Introduction

In order to establish legal title and facilitate safe property transactions, land registration is essential. Due to their centralisation and heavy reliance on manual procedures, traditional land register systems are laborious, prone to fraud, and susceptible to unauthorised changes. These restrictions frequently lead to disagreements, a lack of openness, and a decline in stakeholder trust.

A safe, effective, and transparent system for keeping land records is becoming more and more necessary as digital technologies improve. Although current digital systems increase accessibility, they still rely on centralised authorities, which poses hazards including data tampering and single-point failures.

In order to improve data quality, accessibility, and process automation, this project suggests a Secure Land Registry System that makes use of contemporary web technology. The system's sturdy and scalable architecture is ensured by the use of a MySQL database, a React.js frontend, and a Spring Boot-based backend. To keep safe and trustworthy land records, it integrates transaction tracking, administrative verification, and authentication methods.

Through an organised and tracked workflow, the system allows users to register land, confirm ownership, and carry out property transfers. The suggested strategy minimises

errors, increases overall transparency, and decreases manual involvement by digitising records and upholding a regulated validation procedure.

The remainder of this paper is organized as follows… an overview of current methods, the suggested system architecture, implementation specifics, and assessment of the created solution.

2. Literature Survey

The shortcomings of conventional, paper-based methods have drawn a lot of attention to the modernisation of land registry systems. Conventional systems are frequently centralised and necessitate physical paperwork and manual verification, which causes delays, inefficiencies, and a greater vulnerability to fraud and data manipulation.

Digital land management systems that store and maintain property information using centralised databases have been proposed by a number of scholars. These solutions decrease paperwork and increase accessibility, but they still rely significantly on a single administrative body. Risks including unauthorised data alteration, a lack of transparency, and possible single-point system failures are introduced by this dependency [5]

Some studies have integrated role-based access control and authentication systems into land registry platforms to improve security. These methods enhance operational control and limit system access to authorised users. They improve application-level security, but they fall short in addressing issues with data integrity and auditability [5]

In order to guarantee the immutability of land records, more modern methods have investigated the use of secure and tamper-resistant technologies [1], [3]. Transparent transaction histories and the prevention of unauthorised changes are the goals of these systems. Despite their benefits, these methods frequently encounter difficulties with scale, complexity, and integration with current administrative frameworks.

Additionally, web-based solutions that make use of contemporary development frameworks have been established to enhance system efficiency and usability. While React.js offers responsive and engaging user interfaces, technologies like Spring Boot facilitate the creation of scalable backend services. However, strong validation, safe data management, and successful transaction workflows are necessary for these systems to be effective [5]

It is clear from the analysis of current systems that while digitisation has made land record administration better, issues with security, openness, and effective processing still exist. A dependable and scalable system that guarantees safe data management, restricted access, and efficient transaction processing is thus required [5]

3. Proposed System

The suggested system is a Secure Digital Land Registry System made to offer a dependable, open, and effective platform for handling property transactions and land records. The solution ensures safe data handling and restricted access to sensitive information by substituting a structured digital workflow for conventional manual methods.

A MySQL database, a Spring Boot backend, and a React.js frontend make up the system's contemporary web-based architecture. Scalability, reactivity, and effective data management are guaranteed by this combination. The frontend offers a user-friendly interface for administrators and users, while the backend manages transaction processing, authentication, and business logic.

The User Module and the Admin Module are the two main modules that make up the system. People can register, log in, submit requests for land registration, confirm ownership information, and start property transfer procedures using the User Module. Verifying provided records, accepting or denying requests, and preserving the system's integrity fall within the purview of the Admin Module.

The system uses authorisation and authentication procedures to provide security, enabling only authorised users to carry out tasks. Before being entered into the database, every transaction is verified to guarantee data consistency and stop illegal changes. Every action is traceable and accountable because to the utilisation of organised workflows.

Additionally, the system keeps a transaction history, which allows administrators and users to keep tabs on previous actions and track ownership changes. This lessens the possibility of disagreements and increases transparency. Furthermore, all data entries are examined prior to final clearance thanks to the centralised validation procedure, which reduces mistakes and fraudulent activity.

Overall, by digitising procedures, increasing accessibility, and guaranteeing safe and dependable transaction handling, the suggested system improves the effectiveness of land record management. It offers a workable solution that can be expanded for practical uses in e-governance and property management systems.

4. System Architecture

The frontend, backend, and database layer are the three primary components of the proposed system's multi-tier web-based architecture. This design ensures modularity, scalability, and effective communication between system components.

The overall system architecture is illustrated in Fig.1. The frontend layer is developed using React.js, which provides a dynamic and responsive user interface. Through functionalities such as registration, login, land record

submission, and property transfer requests, it enables users to interact efficiently with the system. The user-friendly interface ensures smooth navigation across different modules.

Spring Boot is used to implement the backend layer, which manages the application's main business logic. It handles user requests, controls authorisation and authentication, and guarantees safe connection between the database and the frontend. For object-relational mapping, Spring Data JPA and Hibernate are utilised, allowing for effective database interaction.

MySQL is used by the database layer to store system logs, transaction information, land records, and user data. It facilitates organised information storage and guarantees data permanence. To ensure data security and integrity, all operations including record insertion, changes, and retrieval are controlled by the backend services.

The system uses a client-server communication model, in which the backend APIs receive HTTP requests from the frontend. Postman is used to test these APIs to guarantee correct operation and dependability. Before returning a response to the frontend, the backend verifies each request, completes the necessary tasks, and communicates with the database.

The system maintains an organised workflow whereby user actions, including transferring ownership and registering land, are initially submitted for administrative verification. The modifications are only permanently stored in the database after the administrator has given their approval. This stops illegal activity and guarantees controlled data modification.

Overall the architecture ensures safe data management, effective user request processing, and a clear division of responsibilities, making the system reliable and scalable for practical uses.

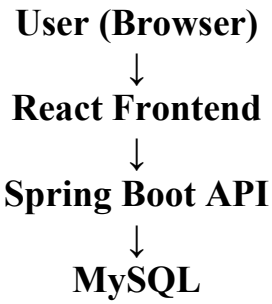

Fig. 1. System Architecture of the Proposed Land Registry System

5. *Methodology*

To guarantee safe and effective management of land registration and property transactions, the suggested system adheres to a standardised process. From user interaction to final record storage, the methodology is intended to provide a clear sequence of processes, enforce controlled access, and preserve data integrity.

Users must first register and authenticate themselves in the system. User information is safely saved in the database after registration, and authentication procedures guarantee that only legitimate users can access system features. After being verified, users can utilise the interface to log in and communicate with the system.

Following a successful login, users can start a land registration request by providing the required information, including ownership and property details. These requests are sent to the administration module for validation rather than being stored as verified records right away.

Maintaining system dependability depends heavily on the administrative verification procedure. After reviewing submitted requests and confirming the accuracy of the information supplied, the administrator decides whether to approve or reject the request. The land records are only permanently kept in the database following approval.

For property transfer, the system follows a similar controlled workflow. The current owner initiates a transfer request, which includes details of the new owner. This request is again subject to administrative verification to prevent unauthorized or fraudulent transfers. Upon approval, the ownership details are updated in the database, and the transaction is recorded.

Throughout the process, the system maintains a transaction history log, ensuring that all activities such as registrations, approvals, and transfers are recorded. This enhances transparency and allows for easy tracking of ownership changes over time.

Additionally, the system ensures data consistency by validating all user inputs and maintaining a structured workflow. This approach enhances reliability and reduces the possibility of errors during land registration and property transactions.

All interactions between the frontend and backend are handled through secure API calls. Each request is validated at the backend before processing, ensuring that only authorized operations are executed. This step-by-step workflow ensures that the system remains secure, reliable, and resistant to unauthorized modifications. ***The step-by-step workflow of the proposed system is illustrated in Fig. 2.***

The proposed system follows a structured workflow to ensure secure and efficient handling of land registration and property transactions. The methodology is designe to maintain data integrity, enforce controlled access, and provide a clear sequence of operations from user interaction to Final record storage.

1. Users register and authenticate themselves in the system usiinin the system. During registration, user details are securely stored in the database, after valida- ting the registration information or ensurijing finciilaties.
2. Authenticated users submit land registration requests by providing property details such as owner information and land documents. These requests are for- warded to the administrative module.
3. The admin verifies the submitted land registration requests, checking ownershi details, property documents, and other relevant information. The admin either approves or rejects the requests based on verification results.
4. Upon approval, the land records are permanently stored in the database. This ensures that the approved land registration requests are securely and immutably documented.
5. For transferring property ownership, a similar process is followed where owners transfer requests from current owners are subject to admin verification. Upon approval, the ownership details are updated in the database.

Fig. 2. Workflow of the Secure Digital Land Registry System

operation and dependability. To stop illegal activity, every request is verified before being carried out.

A. User Interface and System Screenshots:

The developed system includes multiple user interface components that facilitate interaction with the land registry system. The dashboard view of the system is shown in Fig. 3, providing an overview of registered records and system status. The land registration interface is illustrated in Fig. 4, allowing users to submit property details. The blockchain explorer used for viewing transaction blocks and verification details is presented in Fig. 5.

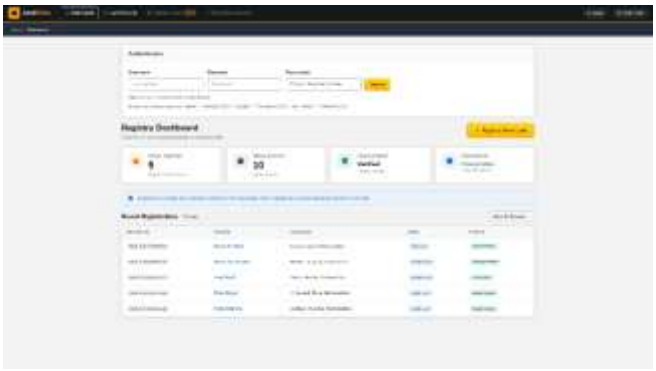

Fig. 3. Dashboard showing system overview, land records, and blockchain status

6. Implementation

In order to assure effective and secure operation, the proposed Secure Digital Land Registry System is designed utilising a contemporary web-based architecture that blends frontend, backend, and database components. MySQL is used for data storage, React.js is used for frontend development, and Java and Spring Boot are used for backend processing.

Business logic, user authentication, request validation, and administrative verification procedures are all handled by the backend. Land records and transaction details may be stored and retrieved with ease because to the effective database interaction provided by Spring Data JPA and Hibernate.

React.js is used in the development of the frontend, giving users an interactive and responsive experience. It makes features like record viewing, land registration, user login, and property transfer requests possible. The interface guarantees easy navigation between various parts and is made to be user-friendly.

The frontend delivers requests to the backend for processing via RESTful APIs, which facilitate system communication. Postman is used to test these APIs to guarantee correct

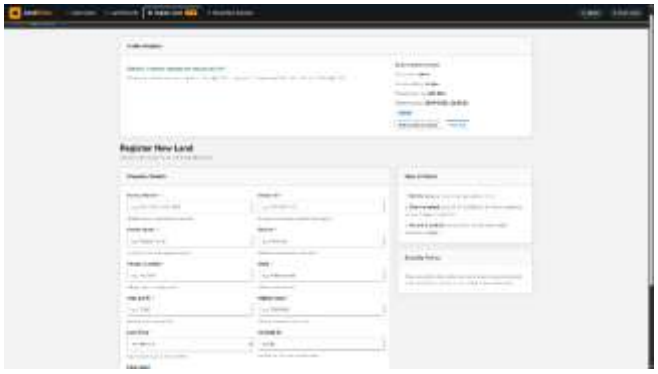

Fig. 4. User interface for registering new land records with property details

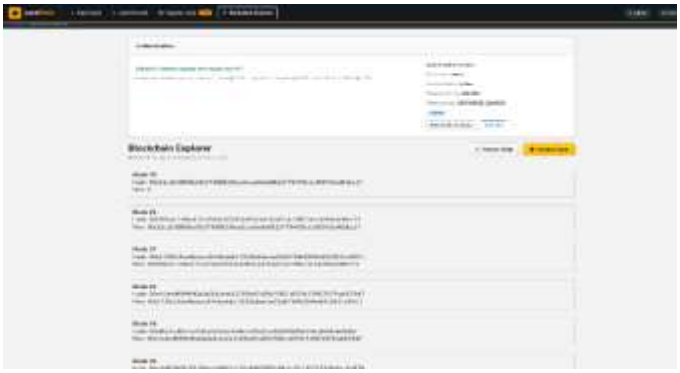

Fig. 5. Blockchain explorer displaying block details, hash values, and transaction records

7. Results and Discussion

Using the specified architecture and technology, the suggested Secure Digital Land Registry System was successfully created and tested. Through a methodical and safe workflow, the system exhibits efficient management of land registration, ownership verification, and property transfer procedures.

In order to prevent unwanted access, the implementation makes sure that only users who have been authenticated can access system functionalities. An extra degree of protection is added by integrating administrative verification, which guarantees that all requests for land registration and transfers are verified before being permanently entered into the database. This greatly lowers the likelihood of fraud and data tampering.

By keeping track of every transaction, the system offers increased transparency. By tracking ownership changes and reviewing historical records, users and administrators can improve accountability and reduce disagreements. The suggested method eliminates the need for physical documentation and speeds up processing when compared to conventional manual systems.

React.js guarantees a responsive user interface, whereas Spring Boot facilitates effective backend processing from a performance standpoint. The system efficiently manages numerous user requests while preserving dependability and uniformity throughout operations. Structured queries and appropriate data management techniques optimise database operations using MySQL.

Postman testing of API endpoints verifies that all system features, such as user login, data submission, verification, and updates, function as intended. In addition to effectively handling invalid or unauthorised requests, the system accurately replies to valid inputs.

Overall, the system's goal of offering a safe, effective, and user-friendly platform for managing land records is accomplished. The system has great promise for real-world use with additional improvements and integration, even though it is created in a controlled environment.

8. Conclusion

In order to increase the effectiveness, dependability, and transparency of land record management, this article proposed a Secure Digital Land Registry System. The suggested method introduces a structured digital workflow backed by contemporary web technologies in order to overcome the drawbacks of conventional manual and centralised systems.

To provide a scalable and user-friendly platform, the system combines a MySQL database, a Spring Boot backend, and a React.js-based frontend. Secure and controlled data management is ensured by essential features such property transfer, administrative verification, land registration, and user authentication. Transparency is improved and the possibility of unauthorised changes is decreased by incorporating transaction tracking systems.

The system's ability to effectively handle land records while preserving data accessibility and integrity is demonstrated by the deployment. Overall, the suggested strategy provides a workable method for updating land register systems and enhancing confidence in real estate transactions.

9. Future Scope

By adding new features and cutting-edge technologies, the suggested system can be improved even more. Adding a completely decentralised blockchain network could be one way to strengthen immutability and do away with the need for centralised validation.

In order to facilitate real-world implementation and interoperability, future work may possibly incorporate integration with government land record databases. Usability and accessibility can be further improved with features like mobile application compatibility, real-time notifications, and intuitive dashboards.
Furthermore, the system would be able to effectively manage high data and user volumes by including sophisticated security measures and scalability enhancements. Better decision-making and system monitoring may also be supported by the addition of data analytics and reporting capabilities.

Acknowledgment

The authors would like to express their sincere gratitude to our project guide, Prof. Pradeep Shinde, for his valuable guidance, continuous support, and encouragement throughout the development of this project. His insights and

expertise greatly contributed to the successful completion of this work.

10. References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[3] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[4] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," 1996.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 2004–2021, 2017.

[6] R. Buyya et al., *Mastering Cloud Computing*. New York, NY, USA: McGraw-Hill, 2013.

[7] R. Johnson, "Secure Data Management in Web Applications," *International Journal of Computer Applications*, vol. 180, no. 25, pp. 1–6, 2018.

[8] Oracle, "MySQL Documentation," [Online]. Available: https://dev.mysql.com/doc/

[9] Spring, "Spring Boot Documentation," [Online]. Available: https://spring.io/projects/spring-boot

[10] React, "React Documentation," [Online]. Available: https://react.dev/