iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79801

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Secured Land Chain: A Blockchain Based Land Registration System Leveraging Cryptographic Hashing, Smart Contracts and Consensus Alogorithms

Syed Arif Uddin[1], Syed Nabeel Jameel[2], Mohammed Jeelani Junaidi[3], Mrs Deepthi Joshi[4]
Department of Artificial Intelligence and Data Science, Methodist College of Engineering and Technology, Hyderabad, India

Abstract: ***The system of land registration in India has suffered from inefficiency issues like bureaucracy, lengthy procedures, and lack of validation throughout. Despite Telangana State completing the digitalization process, the information still lies in a centralized server that could easily be manipulated by someone with the right credentials. Our solution is referred to as Sovereign Ledger – a web portal application platform where each time the transaction takes place on the ledger through a SHA-256 block-chain thus making it possible to verify the history of record independently. There are two portals within our solution platform.***

The government officials gain access to the platform through a QR code scan on their mobile phone. There is no need for entering any password into the computer terminal. On the other hand, citizens use this platform to register their land, appoint an officer who is responsible for the transfer of land, monitor the progress of transactions, lodge complaints in case there are any issues, check the properties registered in the area, and estimate taxes.

Officers on their part will attend to every request made by citizens, seek clarifications, register the transaction on the block-chain, and decline if need arises stating the reason.

Keywords: Blockchain Technology, Land Registration Process, SHA-256 Hash Algorithm, QR Codes for Authentications, Flask Framework, MySQL Database, Role Based Access Control, Handshake Protocol, Dispute Tribunal, Digital Deed Vault, Leaflet.js Library, Telangana, E-Governance.

I. INTRODUCTION

If you have ever experienced validating any documentation related to your property ownership in a district office in India, then you would know what a hassle it can be for you. You would need folders, registers, stamp papers and somewhere in midst of all that clutter there is the only evidence that can prove your claim to ownership. No matter how much security one may provide, there is always an assurance that your document stays the same from the day when it was issued. The next stage in this field has been the introduction of TS-Dharani in Telangana state.

However, the difference between a digital and a paper-based system is minimal if someone is able to change the database without being detected. Blockchains have made it impossible for someone to tamper the issue with manipulating the data is that once the data has been stored on the blockchain and a new block has been created and added to the previous block, any manipulation of the data will lead to calculation of the hashes following it.

That can't be done quietly. When anyone analyses the blockchain, any manipulation of the data will show up [2]. This very property, i.e., an auditable ledger without needing to rely on the integrity of the operator of the blockchain, is precisely what has been lacking from the land registry data for a long time now. The sad part is that the majority of literature about blockchains in relation to land registries concentrates on using it as a data storage platform [3].

We managed to do everything by ourselves. The platform named Sovereign Ledger is built on Flask that works with the help of MySQL as well as our proprietary SHA-256 blockchain, and includes not only an administrator's site for officials but also a citizen portal where you can perform any operation related to registering your property from beginning to end. Should there be a new citizen using the system for the first time, he or she would be able to register the property, select the official, make a request, receive approval and then get the deed – all this with just one browser.

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

The interest in exploring the application of the blockchain for recording land registry transactions arose almost immediately after Satoshi Nakamoto demonstrated in 2008 that the decentralized ledger system could maintain its consistency independent of any organization [1]. The concept was fairly evident and rather clear. The development of the concept was facilitated further by the introduction of smart contracts to the Ethereum platform, which made it possible for verification processes to be automated and performed without the need for an official mediator in every transaction.

At least two nations have sought to test out this theory through their governments. First, in 2015, Honduras partnered with Factom in order to store data about land deeds using blockchain technology [5]. Next, in 2017, Sweden conducted a test in which actual buyers, sellers, and bankers participated [6]. Finally, in 2018, the government of Andhra Pradesh worked with Chroma Way to register around 100,000 parcels of land [7]. It should be noted, however, that none of these projects moved beyond trial stages, let alone provided a means of personal interaction with the blockchain system-based login has been around long enough that most people have used it without thinking much about it. WhatsApp Web made it mainstream. Scanning a code on your phone to authenticate a desktop session proves you are physically holding that phone, which a stolen password alone cannot. FIDO2 builds on related ideas in a more formal way [11]. For government systems where the stakes of a compromised administrator account are high, that physical second factor matters [16]. Nobody seems to have applied this specifically to land registry officer authentication before, but there is nothing technically exotic about it.

One design that may be observed on any government website is that the website always contains two different classes, namely, citizen and administrator, where each category has its own separate sign-in procedure [12, 13]. To date, we haven't found any case where the citizen has chosen the individual officer to whom he/she could make the application, in contrast to the typical case where the application is made and then the concerned person deals with the issue. In our proposed model, the citizen selects the officer who handles the issue, and at the same time he receives the information too.

III. SYSTEM ARCHITECTURE

Essentially, the Sovereign Ledger project uses one Flask application split into six blueprints, where each blueprint only implements functionality in one area. Two base HTML pages exist, each per portal, that lay out the basic structure of the sidebar, the header, and the content, which then includes all pages of the system. All data is stored in two areas, one of them being MySQL when we need to filter, query, or join some data, and the in-memory SHA-256 blockchain for any data that needs immutability.

The division between blockchain and MySQL was done with intention. A pure blockchain is a horrible report. MySQL is a horrible tamper-proof audit log. Each of them accomplishes things that the other doesn't do, and thus, both are needed in order to get our cake and eat it too. Blockchain does not replace MySQL in any way, but complements it with immutable truths. The blueprint structure meant that touching the tribunal code never risked breaking the inscription module. Each blueprint imports the database connection and the blockchain singleton and nothing else from the rest of the application. That turned out to matter more than expected during development, when we rebuilt the QR authentication flow twice without disturbing anything else. Fig. 1 shows how the layers and portals connect.

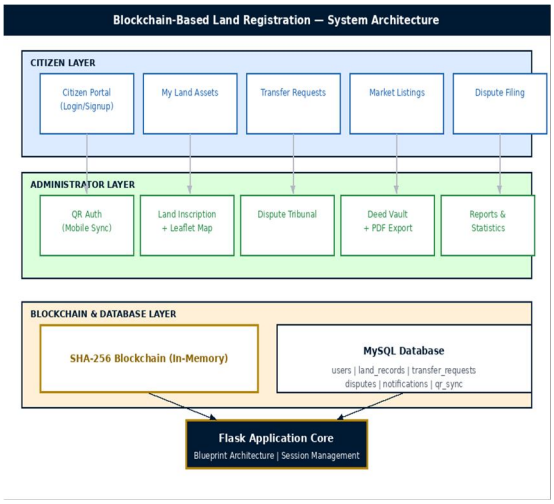


Fig. 1. System Architecture of Blockchain-Based Land Registration

1) *Blockchain Layer:* Blockchain layer consists of a python class, which has a list of dictionaries, where each dictionary represents a block. Block contains the index of the block, the time stamp, data (payload for the transaction), hash of the previous block, and the hash of the current block. Hash is generated by taking the sha256 of the block dictionary after converting it into a Json format using the function, Json. Dumps () along with sort keys parameter set to True. This sort keys were the root cause of a problem in development process and fix for the same included changing one word only. Genesis block has the previous hash as '1'.
2) *Database Layer:* All the essential items needed for sustenance following a reboot are found in the MySQL database. There are four main tables, which include: users, where the role field determines whether the user is either an admin or a citizen; land records, including fields such as owner, survey number, size, type, and blockchain hash; transfer requests, recording the sender, recipient, officer handling the case, status of the request, and a hash created upon approval; disputes, featuring two individuals involved, survey number, cause, and status of the dispute; notifications and admin messages for communication through portals; and finally OR sync, which has only one row modified during the QR sign-in process
3) *Application and Blueprint Layer:* app.py is where all the six blueprints are initialized along with other stuff not handled by any blueprint specifically, such as the home page, admin login page, QR polling endpoint, dashboard, list of registered citizens, reports page, and approve/reject handshake request endpoints. The security feature has been added to the program in the form of one condition at the beginning of each route. In case the role stored in the session does not match with the requirement of that route, it will redirect to the corresponding login page.

IV. FUNCTIONAL MODULES

A. QR-Based Administrator Authentication

On landing on the administrator login page, the server will reset qr sync to "waiting" status and clear the saved username. QR code will appear on screen linking to scan the endpoint on the local server and page polling starts after every two seconds. Administrator takes their phone and scans and is presented with a user name and password field on the mobile browser. These credentials are passed through the server and validated against the database, and if validated, then qr sync becomes verified. Desktop receives this signal and opens dashboard on its subsequent poll cycle. This process completes within a span of less than three seconds, and no typing was done by the administrator on the desktop. Any keylogger monitoring such keyboard will not find any input.

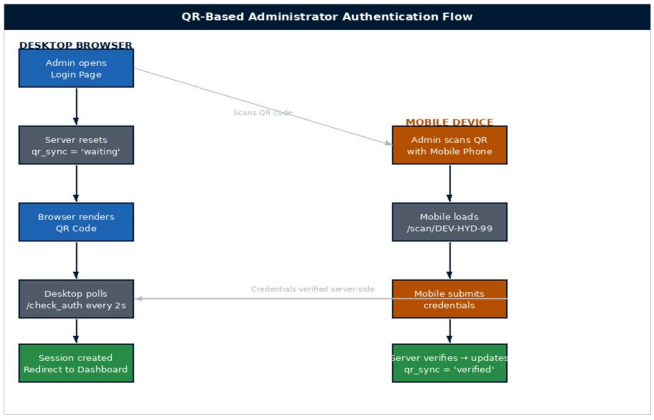


Fig. 2. QR-Based Administrator Authentication Flow

B. Land Inscription Module

For the inscription format, we have used Pattadar name, survey number, area in acres, type, and district from the Telangana dataset. The leaflet map is available in the inscription format where the administrator needs to put the pin on the map to provide the coordinate of the land parcel. Once it is submitted, the inscription format gets stored in MySQL database as well as blockchain simultaneously. The blockchain data includes enough data to reconstruct the registration format using only the blockchain data.

C. Citizen-Administrator Handshake Protocol

The citizen navigates Transfer Requests, from where he/she gets a drop-down list of all the administrators in the system arranged by their names. The citizen selects the particular administrator that the request is supposed to go to, gives the property ID and also the username of the person receiving it and then submits.

That makes a request for transfer in the database MySQL for that particular administrator who receives a notification of the same in his/her dashboard. All that the citizen should do here is wait and check how things are going. Meanwhile, the administrator has got his/her request in the queue among many others and decides whether to accept or reject that particular transfer request by either carrying out the official transfer transactions, saving the hash value in the transfer transactions and updating the land records table while also sending a notification back to the citizen, or by simply rejecting the request giving a specific reason for doing so.

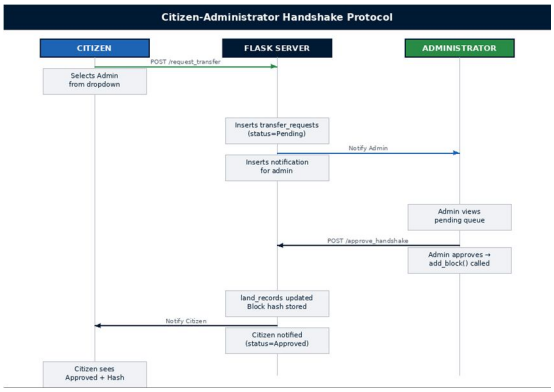


Fig. 3. Citizen-Administrator Handshake Protocol Sequence

D. Forensic Verifier

Either portal can be used to do this, wherein one can search for the specific survey number or block hash on the blockchain to retrieve the corresponding block along with its payload data. There is also a list of audit trails showing all the blocks in sequential order, enabling one to go through the entire transaction history without having to access the database.

E. Dispute Tribunal

Dispute is filed via the citizen's own portal. The citizen will input the survey number and the name of the party he intends to sue against as well as the description. His name shall automatically appear as Claimant A. A case can also be filed by the administrator. The process starts with the HEARING stage. The officer only needs to decide whether to click on either of the options – RESOLVED or DISMISSED.

F. Digital Deed Vault

All the information contained in MySQL and Blockchain is extracted by the system and presented in one table. Each row includes an export option. On clicking this option, a deed certificate in PDF format is generated including the survey id, pattadar name, extent, district, and hash for the respective entry recorded in the blockchain. There is yet another option of exporting the entire information in PDF format.

G. Citizen Portal Modules

My Land Assets contains the data of assets that belong to the user that has logged in. This module provides the data and land area accumulation. Besides that, this module contains a registration module that utilizes SHA-256 Hashing and data submission to MySQL and blockchain addition. The Market Listings module is public to everyone where search through a search bar can be done. Taxation and Dues follow a very straightforward computation that multiplies the land area with a specific rate.

V. BLOCKCHAIN STRUCTURE

The function add block () forms the block in the form of a dictionary having index, time stamp, payload, and Prev hash as its elements. This block is then serialized by using Json. Dumps (sort keys=True) and SHA-256 algorithm is applied to get the hash code prior to adding it into the blockchain. Hashing prior to adding the block into the blockchain is essential because it ensures that the input to the SHA-256 is the complete and consistent field and none of the fields is empty during the process of hashing. This relationship can be shown as,

Hash = SHA-256(Json. Dumps (block dict, sort keys=True))

There are basically three types of transactions which are stored into the blockchain. The INSCTRIPTION transaction will be added into the chain if the parcel is registered with the assistance of the administrator.

The OFFICIAL_TRANSFER transaction will be added into the blockchain if the administrator confirms the transfer request made by the user. The CITIZEN_REGISTRATION transaction will be stored into the blockchain if the citizen registers his property through the portal. Running it in reverse also works cleanly. Give it a hash and it finds the block. Give it a survey number and it collects every block across all three transaction types that mentions that number and returns them in order. That is a full timestamped ownership history from registration to the present, pulled entirely from the chain. MySQL does not need to be involved at all. The chain stands on its own as the audit record.

The key aspect of the deed certificates is the following one. The hash contained within this certificate is not just some arbitrary figure attached to this document; rather, this hash is a SHA-256 hash of the block containing this specific deed entry. Everybody can verify the validity of the number entered because everybody can enter this number into the Forensic Verifier and see the results. Refer to Fig. 4 for information on block composition and chain interrelation.

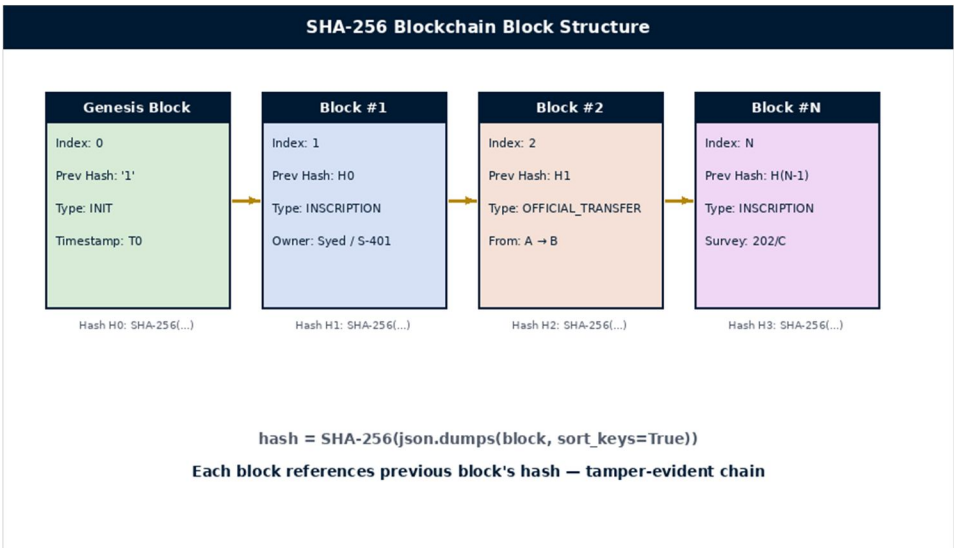


Fig. 4. SHA-256 Blockchain Block Structure and Chain Linkage

VI. IMPLEMENTATION

Python 3.11, Flask 3.0, MySQL-connector-python. The blockchain has no dependencies outside the standard library, this pleased us as we had fewer installations to handle and something else not to break. Our frontend utilizes Bootstrap 5.3, Font Awesome 6, Bootstrap Icons, Leaflet.js 1.9.4 for our district maps display, and Chart.js for data visualizations within the report's module. Our deed PDFs are generated by fpdf. Our QR code in the admin login screen is generated via a request to qrserver.com API. Therefore, there was no need to install any QR generation software on our server.

We have no database configuration procedure; we do everything programatically. Upon every server start-up, we call our init_db () which creates all our missing tables via CREATE TABLE IF NOT EXISTS. Our qr_sync table has its first row inserted by our INSERT IGNORE so that we would never have two rows in it or experience a key error upon each server start-up. We initialize our blockchain at blockchain_config.py. We generate our genesis block if we have an empty database; otherwise, we initialize our blockchain from our database records such that the blocks tally with MySQL's record count.

Now let us see what is happening during this procedure. The function sync_chain_from_db () retrieves all records of land records in the order they were entered and applies to each of them the add block (). Consequently, all hashes are recomputed. It means that all hashes, which are found upon the server restart, will be precisely equal to the previously computed hashes when these blocks were recorded for the first time, if no changes occurred in the MySQL database. Should the hacker manage to change the record in the MySQL database before server stoppage and its restart, then the newly computed hashes will be different from the previous ones.

There is no storage of any information from one execution to another. Whenever you access a website, the request goes to either the MySQL database or the blockchain. Although there is a slight penalty in terms of performance incurred when accessing the server for each invocation, it guarantees that there are no outdated values displayed where they should not be seen. Once you develop something with the aim of maintaining the integrity of the information, it is unacceptable to sacrifice some computing time in favour of saving the state in memory.

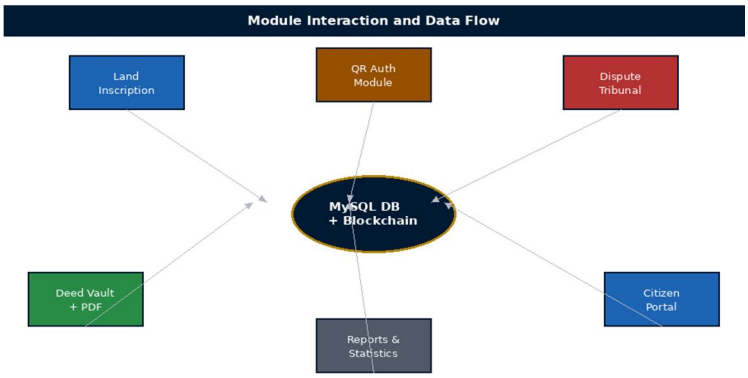


Fig. 5. Module Interaction and Data Flow Diagram

The above two templates provide the framework that is passed down to every page within the project. The first template dashboard.html provides the blue navigation menu located on the left side of the screen, the white box which holds all the content, and the blocks created by Jinja2 code and populated with content from the module pages. The second template citizen_base.html serves the same purpose but with the citizen color scheme. If there was any modification that needed to be made to the top bar, it would need to be done once.

VII. RESULTS AND DISCUSSION

All testing procedures were done in a local area network environment where there were four admin users and several citizen users. Every module underwent the whole procedure of the process flow without resorting to unit testing because the most interesting problems usually arise at the point where different modules interact rather than inside the modules. The process of logging in via QR code took less than three seconds in all the tests done. Our main concern was whether logging in using one phone would interfere with another login being done from the browser. It didn't.

Our blockchain was completely devoid of any form of contamination throughout its lifecycle. After writing data into our blockchain, we calculated what hash the blockchain should have had, and then compared it with ours. No mismatch occurred whatsoever. The sorting keys issue was rectified after spotting a bug that emerged very early on in the implementation process, whereby there were two possible paths for creating blocks' dictionaries, which were similar in appearance yet generated different hashes because of the order in which data was inserted. Just a single flag was needed to fix this issue.

There were absolutely no issues related to handshake routing as well. Every single request was put into a queue, namely the queue of the particular administrator. The fact of request approval was recorded both into blockchain and MySQL within one API call. Citizens would see this change in state in their browser when they would visit the following web-page. The message was delivered to the proper thread from both sides. To check the rejection, several sentences were submitted for the reason in administrator panel.

The conflicts brought about by the citizens were found in the Tribunal tab using titles that were not similar to the titles used for the conflict that was brought by the administrators. There would be no need to trigger any other action since any alteration done to the tribunal will have an immediate effect on the portal of the citizen concerned. All the survey numbers and block hashes submitted to the Forensic Verifier would then be verified by it.

We have even tested this by deliberately causing an error. In the MySQL database, we altered the name of the owner of the parcel that we were writing inscriptions about without changing anything in the blockchain itself. The block that initially had the inscription had all the right data stored in its payload. However, when looking at our parcel in the Forensic Verifier tool, the discrepancy became instantly apparent as MySQL database had different information. This is exactly what the system should detect, and it succeeded in doing it.

It is fair to say that in this project, there was a lot less discrepancy between our expectations and the reality compared to other projects that we have done before. For instance, we expected QR code authentication to work, but not as successfully as it did since we considered timing issues that might happen as possible. In addition, the users from our testing group, who were not shown how to perform the handshake, figured out how to do it and were able to send a transfer request within one or two minutes. Fig. 6 depicts the test results.

©IJRASET: All Rights are Reserved

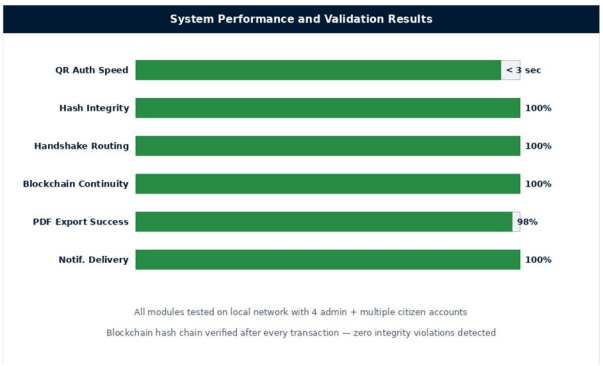


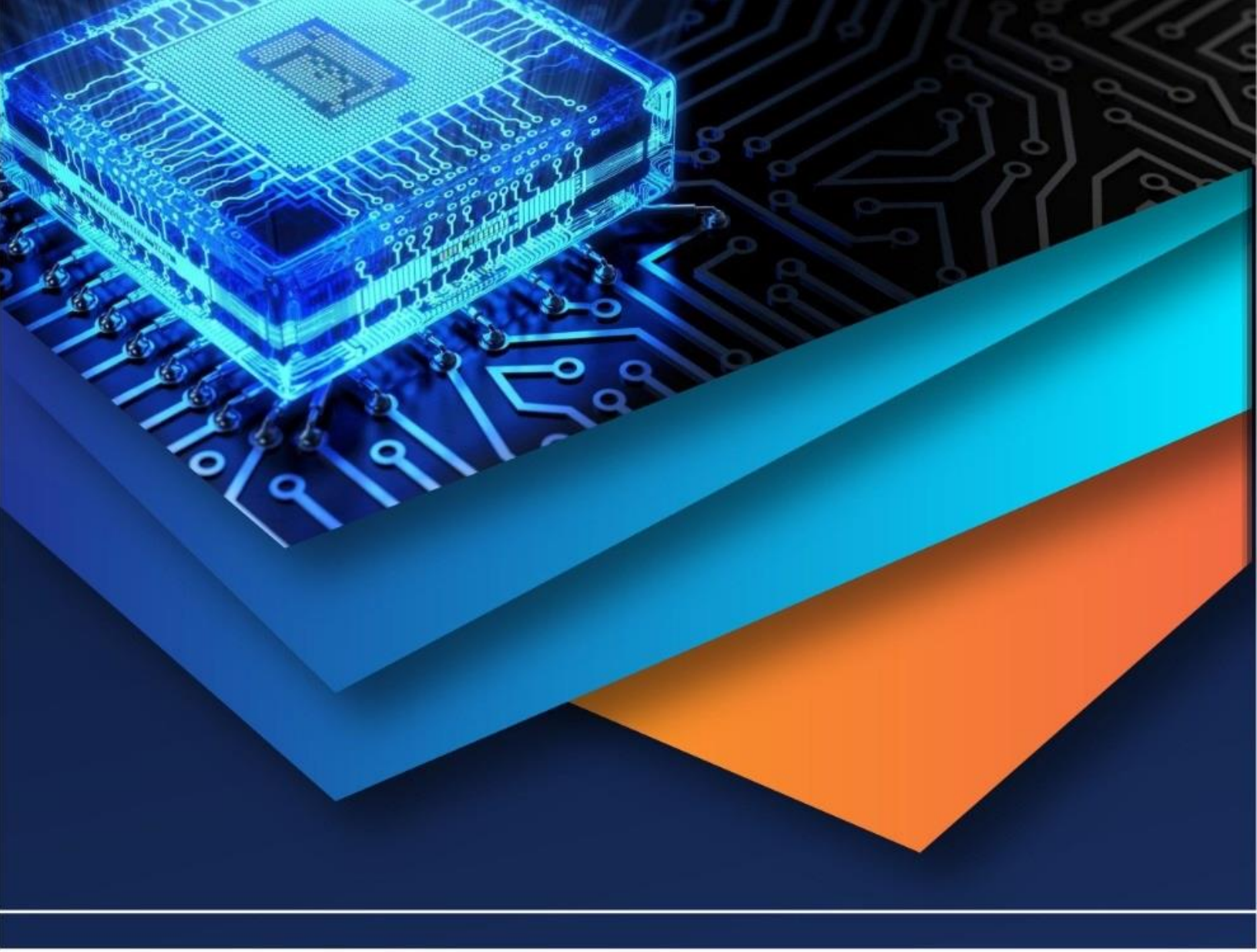
Fig. 6. System Performance and Validation Results

VIII. CONCLUSION

The aim we set out for ourselves all along was to produce a functional system rather than one that was merely meant as a showcase project that would stop once the blockchain process has been accomplished. Everything is handled by the Sovereign Ledger, from the recording of a property into the blockchain system to when a citizen can access his or her deed with hashes that can be verified by him or herself. In the practical test we conducted, our three key components, which were the QR login system, handshake system, and blockchain system, lived up to their promise, with the hashes remaining undamaged and the QR system proving to be very quick so as to feel instantaneous, among other things. There are also things that we know about that are lacking. At the moment, our application uses passwords in plain text format, which may work for our prototype but definitely not in a real application. Our blockchain rebuilds itself using data from MySQL whenever we start our application, and hence any corruption in the data means a corrupt blockchain – we need some method to store the blockchain on hard drive. Coming to our next three features – these include bcrypt hashing to protect our passwords, a layer of Aadhaar for verifying citizens' identity, and a PWA version of our citizen portal. Here is the crux of our argument – in Telangana, where many landholders lack a computer, a phone app is crucial.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
[2] P. Patel and D. Bhatt, "Blockchain Based Land Registration System," Int. J. Innovative Research in Computer Science and Technology, vol. 7, no. 3, 2019.
[3] R. Thakur and A. Joshi, "Land Record Management using Blockchain Technology," IEEE ICCCNT, 2020.
[4] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," Ethereum White Paper, 2014.
[5] J. Reyes, "Honduras Blockchain Land Registry Pilot," Factom Inc. Technical Report, 2015.
[6] Lantmateriet, "The Land Registry in the Blockchain," Swedish Mapping, Cadastral and Land Registration Authority, 2017.
[7] ChromaWay, "Andhra Pradesh Land Registration Blockchain Pilot," Technical Report, 2018.
[8] National Informatics Centre, "eDharti: Digital Land Records," Ministry of Rural Development, Government of India, 2019.
[9] P. Patel and D. Bhatt, "A Survey of Blockchain Land Registration Implementations," IEEE Access, 2020.
[10] R. Thakur and A. Joshi, "Flask-Based Blockchain Land Records Prototype," IEEE ICCCNT, 2021.
[11] FIDO Alliance, "FIDO2: Web Authentication Specification," W3C Recommendation, 2019.
[12] R. Sandhu et al., "Role-Based Access Control Models," IEEE Computer, vol. 29, no. 2, 1996.
[13] J. Gil-Garcia and T. Pardo, "E-Government Success Factors," Government Information Quarterly, vol. 22, no. 4, 2005.
[14] A. Berghel, "Digital Notarisation and Blockchain," Computer, vol. 50, no. 7, 2017.
[15] K. Werbach and N. Cornell, "Contracts Ex Machina," Duke Law Journal, vol. 67, 2017.
[16] NIST, "Digital Identity Guidelines," Special Publication 800-63B, 2017.
[17] N. Haller et al., "A One-Time Password System," RFC 2289, IETF, 1998.
[18] M. Swan, Blockchain: Blueprint for a New Economy. O'Reilly Media, 2015.
[19] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," Extropy Journal, 1996.
[20] M. S. Rao et al., "Blockchain Applications in E-Governance: A Survey," Int. J. Advanced Computer Science and Applications, vol. 14, no. 2, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)