

Blockchain-based Land Registration and Ownership Management System with AI-Assisted Document Verification

Rasad Regmi

Department of Software Engineering
Nepal College of Information Technology
Pokhara University
Kathmandu, Nepal
rasad.221737@ncit.edu.np

Dipesh DC

Department of Software Engineering
Nepal College of Information Technology
Pokhara University
Kathmandu, Nepal
dipesh.221714@ncit.edu.np

Birat Aryal

Department of Software Engineering
Nepal College of Information Technology
Pokhara University
Kathmandu, Nepal
birat.221614@ncit.edu.np

Pranav Subedi

Department of Software Engineering
Nepal College of Information Technology
Pokhara University
Kathmandu, Nepal
pranav.221733@ncit.edu.np

Abstract—Land registration in Nepal relies on a paper-intensive process requiring repeated visits to government offices, with limited transparency and a high risk of document tampering. This paper presents a blockchain-based land registration and ownership management ecosystem that digitizes the workflow while preserving the legally mandated physical and biometric verification of the Land Revenue Office. Citizens submit applications, complete KYC, and upload documents online, while verification officers review claims with an AI-assisted tamper-detection pipeline combining optical character recognition (OCR) and classical machine learning. Only SHA-256 digests of documents and opaque property identifiers are recorded on the Ethereum Sepolia blockchain, preserving privacy, while personal data and legal documents remain in PostgreSQL. A state-machine workflow enforces every step before a new Lalpurja certificate is issued. On a synthetic dataset of 2,000 Lalpurja documents, the model achieves a ROC-AUC of 0.854 and a recall of 70% at a 10% false-positive rate. Measured API latencies confirm the practical feasibility of the proposed approach.

Keywords —*blockchain, land registration, ownership management, smart contracts, Ethereum, data integrity, tamper detection, OCR, KYC, Nepal*

I. INTRODUCTION

Ownership of land is among the most valuable legal and financial assets of citizens in Nepal. The current system of land registration and ownership transfer requires extensive paperwork, manual verification, and multiple visits to government departments. Documents such as the Lalpurja (certificate of ownership), Rajinama (transfer deed), and tax certificates are managed in paper form, which increases the risk of tampering, loss, and ownership disputes. Inconsistencies in record keeping across different government offices further complicate verification.

Although some government services have been digitized, existing systems act largely as electronic archives and do not provide an immutable record of ownership and transaction history. A purely electronic registration system cannot be implemented immediately because Nepalese legislation still mandates physical and biometric verification of ownership. There is therefore a need for a hybrid approach that digitizes document submission and workflow management while recording verified ownership information on a blockchain.

This paper reports the design, implementation, and evaluation of a blockchain-based land registration ecosystem for Nepal. The main contributions are as follows:

- A hybrid blockchain architecture in which only cryptographic hashes and property identifiers are anchored on the Ethereum Sepolia blockchain, while personal data and legal documents remain in an off-chain PostgreSQL database, preserving citizen privacy.
- A state-machine-governed registration and ownership-transfer workflow that enforces every legal step, including duplicate detection, least-loaded officer assignment, physical verification, tax assessment, and multi-party signing, before a new Lalpurja is issued.
- An AI-assisted Lalpurja tamper-detection pipeline that combines OCR with a calibrated RandomForest classifier and a structure-aware decision layer to assist, rather than replace, the verification officer.
- A working end-to-end implementation comprising a responsive web frontend, a secured backend, smart contracts, blockchain synchronization, and an audit trail, validated through unit, integration, API, smart-contract, and machine-learning tests.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed system architecture and workflow. Section IV presents the AI-assisted document tamper-detection module. Section V reports results and evaluation, and Section VI concludes the paper.

II. RELATED WORK

Existing blockchain-based land registration systems fall into two broad categories. The first uses permissioned platforms such as Hyperledger Fabric: Sen et al. combined Fabric with IPFS for inter-agency collaboration in e-governance property registration [1]; Biswas et al. proposed a secured system for Bangladesh aimed at reducing fraud in land-transfer records [2]; Ali et al. exposed property records through REST APIs on Fabric [3]; and Pilare et al. developed a Fabric-based system in which all peers endorse every transaction, increasing transparency but not addressing document-level forgery [4]. The second uses public blockchains: Zhang et al. built an Ethereum time-stamping and registration system using Solidity, MetaMask, and IPFS [5]; Krishnapriya and Sarath validated SHA-256 hash anchoring on 200 transactions [6]; Priya et al. combined smart-contract-based user verification, document hashing, and role-based ownership transfer with an audit trail [7]; and Bisht et al. implemented an Ethereum prototype with IPFS, MongoDB, and Flask, noting scalability and regulatory obstacles [8]. Two systems are particularly relevant to existing national registries: Lazuashvili et al. analyzed Georgia's national blockchain land registry and concluded that blockchain serves as a tamper-evident verification layer that complements, rather than replaces, existing government registries [9], while Humdullah et al. proposed a data-storage framework that separates document storage from on-chain verification [10].

Table I compares the main characteristics of these systems. While all demonstrate the value of blockchain for tamper-evident records, none simultaneously address three requirements specific to Nepal's legal framework: (1) a hybrid on-chain/off-chain storage model that records only cryptographic digests while keeping personal data off-chain for privacy, (2) an AI-assisted document tamper-detection pipeline that can identify forged Lalpurja documents before the verification officer, and (3) a state-machine-governed workflow that enforces every legally mandated step—including duplicate detection, physical and biometric verification at the Land Revenue Office, tax assessment, and multi-party signing—before a new ownership certificate is issued. Our system uniquely addresses all three, preserving the mandatory physical verification procedure while digitizing every other aspect of the registration and transfer process.

III. PROPOSED SYSTEM

The proposed ecosystem connects citizens, verification officers, and administrators through dedicated web portals integrated with a backend API that supports authentication, land registration, document handling, ownership transfer, verification, audit logging, and role-based access control. Structured

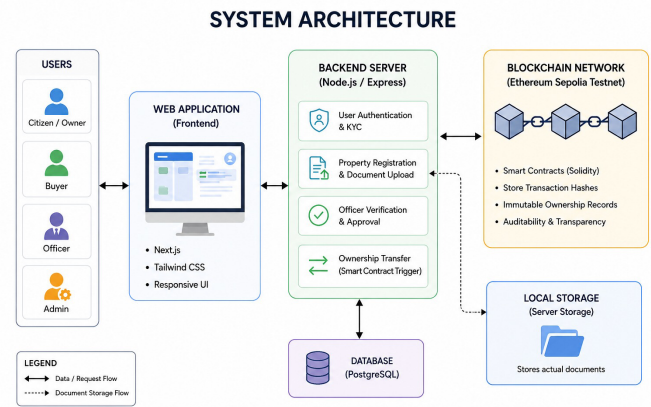


Fig. 1. System architecture of the proposed blockchain-based land registration and ownership management ecosystem.

data such as user records and transaction history are stored in PostgreSQL through the Prisma ORM, while documents are stored locally. A blockchain sync service integrates the backend with the Ethereum Sepolia blockchain and anchors verified property states, document hashes, transfer events, and approval records. To preserve privacy, personal information and full documents remain off-chain; only SHA-256 hashes and opaque property identifiers are recorded on-chain, yielding a tamper-evident ownership trail without exposing sensitive data. The overall system architecture is illustrated in Fig. 1.

The platform is organized into ten functional modules: an *Authentication* module managing registration, login, JWT access and refresh tokens, password recovery, and role-based permissions; a *Property* module handling property creation, duplicate detection, and ownership tracking; a *Document* module managing upload, categorization, and SHA-256 hash generation; a *Verification* module handling officer review queues, approvals, rejections, and correction requests; a *Transfer* module implementing the ownership-change workflow; a *Property Listing* module providing a public marketplace that prevents double-selling; a *Blockchain* module anchoring verified records; a *Ledger and History* module exposing ownership traceability; an *Admin* module governing users and configuration; and an *Audit* module recording actions, timestamps, and record references for accountability.

The system combines three complementary layers of assurance, which address distinct parts of the verification problem and should not be confused. First, *bit-level document integrity* is enforced by SHA-256 hashing: any modification of an uploaded document changes its digest, so the on-chain and off-chain hash registries detect even a single-bit alteration. Second, *content-level tamper detection* is performed by the AI-assisted module described in Section IV, which compares an uploaded document with the distribution of genuine Lalpurja certificates and flags likely forgeries (replaced fields, erased areas, removed signatures) before the officer reviews the claim. Third, *legal ownership verification* is a human decision: the verification

TABLE I
COMPARISON OF BLOCKCHAIN-BASED LAND REGISTRATION SYSTEMS

Work	Country	Blockchain	Hybrid storage	AI document verification	Legal workflow	Scope
Sen et al. [1]	India	Hyperledger Fabric + IPFS	✓	×	×	Registration
Biswas et al. [2]	Bangladesh	Undisclosed private chain	×	×	×	Transfer
Ali et al. [3]	Not stated	Hyperledger Fabric	×	×	×	Registration
Pilare et al. [4]	India	Hyperledger Fabric	×	×	×	Registration
Zhang et al. [5]	Not stated	Ethereum + IPFS/Infura	✓	×	×	Registration + transfer
Krishnapriya & Sarath [6]	India	Undisclosed	×	×	×	Registration
Priya et al. [7]	India	Ethereum	×	×	✓	Registration + transfer
Bisht et al. [8]	India	Ethereum + IPFS	✓	×	×	Registration + transfer
Lazuashvili et al. [9]	Georgia	Undisclosed	×	×	Partial	Registry enhancement
Humdullah et al. [10]	Malaysia	Undisclosed	×	×	×	Data storage
This work	Nepal	Ethereum Sepolia	✓	✓	✓	Full registration + transfer

officer examines the claim, the AI output, and the physical and biometric checks conducted at the Land Revenue Office, and only then approves the property to *verified*. The state machine of Section III-A coordinates these layers—the document hash and AI verdict are produced during *under review*, and only a verified property reaches the blockchain-anchored *active* state. Blockchain integrity therefore guarantees what happened to a certified document in the past, while the AI pipeline and the officer jointly determine whether an uploaded document is genuine today.

A. Registration and Verification Workflow

A citizen creates a property claim, and the system checks for duplicates against the Kitta number, parcel number, plot number, and ward. A match returns a conflict and creates a *PropertyConflict* record; otherwise, the property is created in a *draft* state with the creator as the primary owner. On submission, the property is assigned to the least-loaded verification officer, who progresses it through *submitted* → *under review* and either approves it to *verified* or requests corrections, returning the claim for resubmission. A verified property is registered by creating a *PropertyRegistry* entry, which advances its state to *active*. The state machine is illustrated in Fig. 2.

B. Ownership Transfer

A transfer begins after a seller or buyer initiates a purchase request. Once the seller accepts, the system validates the request and schedules an appointment; a signed QR appointment slip and token are generated. During the visit to the Land Revenue Office, the officer performs physical verification, assesses government fees and capital-gains tax, and confirms payment. The transfer then enters a *blockchain-pending* state, is anchored on-chain, and is marked completed after the ownership registry is updated. Completion requires the signatures of the buyer, seller, and verification officer. Properties listed on a public

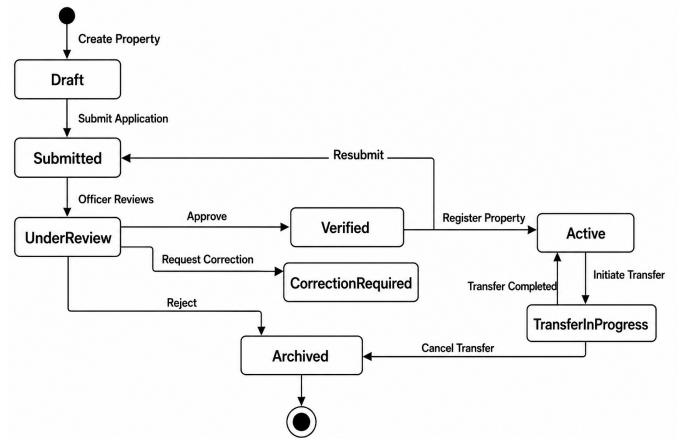


Fig. 2. State machine governing the land registration and verification workflow.

marketplace automatically transition to *pending* while a transfer is active and to *sold* on completion, preventing double-selling.

C. Blockchain Layer

The *PropertyRegistry* smart contract, written in Solidity version 0.8.24 and deployed on the Ethereum Sepolia test network using Hardhat and ethers.js, records only opaque identifiers and hash digests. Its primary functions are *registerProperty*, which creates an on-chain proof-of-existence fingerprint and the current owner; *recordTransfer*, which updates the owner and replaces the canonical hash while preserving the previous hash in the ownership history; *getCurrentOwner* and *getOwnershipHistory*, which expose the immutable ownership trail; *verifyProperty*, which performs an on-chain tamper check; and *propertyCount*. Blockchain transactions are identified by their Keccak-256 digest, and document integrity relies on the SHA-256 hash function. On the Sepolia

test network, block confirmation is typically reached in about 12–15 seconds.

The integrity guarantees of the system are grounded in the properties of cryptographic hash functions. Each uploaded document D is represented by a unique digest $DH = H(D)$, where $H(\cdot)$ is the SHA-256 function; the digest is stored both in the off-chain document hash registry and, after canonicalization, anchored on-chain. Each block of the underlying ledger is identified by a block hash

$$BH = H(PH \parallel MR \parallel T \parallel N), \quad (1)$$

where PH is the previous block hash, MR the Merkle root, T the timestamp, and N the nonce. Linking every block to its predecessor makes the chain immutable [11], and an ownership transition $O_{\text{new}} = f(O_{\text{old}}, \text{Buyer}, \text{PropertyID})$ is recorded on-chain as a transaction with digest $TH = H(\text{TransactionData})$, providing a tamper-evident trail of every ownership change.

D. Security Design

Security is enforced across multiple layers: short-lived JWT access tokens with rotating refresh tokens and bcrypt password hashing for authentication; role-based access control (citizen, officer, admin) with per-resource checks for authorization; SHA-256 document digests verified against the on-chain records for integrity; CORS and helmet headers, rate limits, and Zod-validated request payloads for transport and input security; and the blockchain ownership ledger for immutability and a tamper-evident audit trail.

IV. AI-ASSISTED DOCUMENT TAMPER DETECTION

Rule-based verification alone cannot reliably distinguish a genuine Lalpurja from a skillfully forged one, because forgers manipulate the same fields that a verification officer checks. A machine-learning pipeline was therefore developed to perform a forensic comparison of an uploaded document against the distribution of genuine documents and produce an explainable tampering verdict that assists the officer.

A. Pipeline

The pipeline consists of four stages, as shown in Fig. 3. The uploaded Lalpurja image is processed by OCR (PaddleOCR, used solely as an OCR engine) to extract text regions, bounding boxes, and confidence scores. A verification stage computes field similarity against expected values when metadata is available. A feature builder derives hand-crafted features that are passed to a RandomForest classifier with 300 trees, a maximum depth of 12, balanced class weights, and a fixed random seed, followed by Platt sigmoid calibration. Finally, a structure-aware decision layer combines the model output with a document-structure score to produce the final verdict.

A RandomForest was chosen for three reasons relevant to the deployment setting: it is robust to the heterogeneous scales and non-normal distributions of hand-crafted OCR and computer-vision features without requiring normalization; with balanced class weights it tolerates the moderate class imbalance of the training data; and its built-in feature-importance ranking

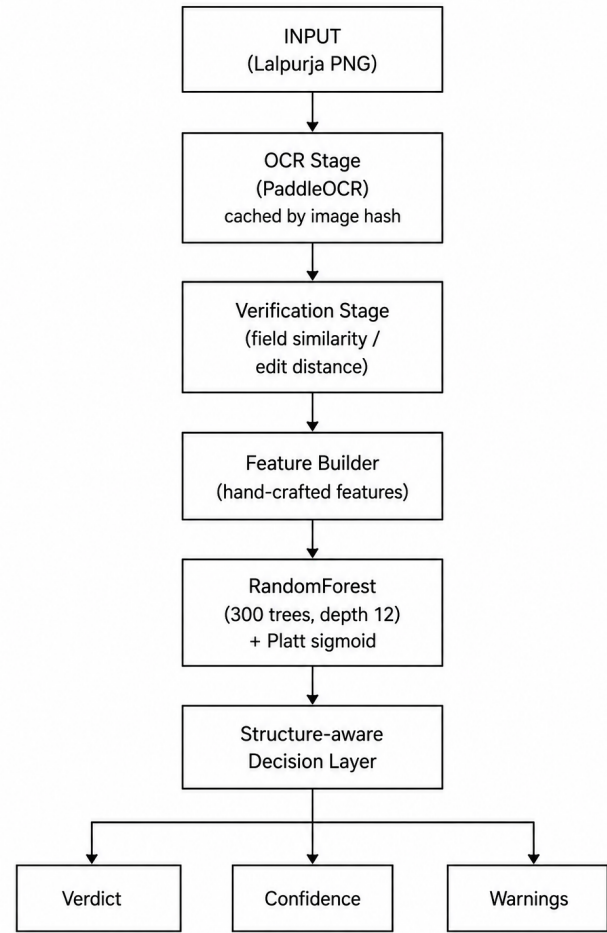


Fig. 3. Architecture of the AI-assisted document tamper-detection pipeline.

supports the explainable-verdict goal, allowing the officer to see which features drove the decision. Platt sigmoid calibration is applied because the downstream structure-aware decision layer thresholds the tamper probability directly, and the raw tree fractions are not well calibrated. The final verdict of the pipeline is displayed within the officer's review queue (Section III), where the officer makes the binding approval decision; the pipeline therefore assists, rather than replaces, human verification.

The experimental design was refined across three model generations so that the model learns document integrity rather than image-quality artifacts; Section IV-D details this evolution and its effect on the false-positive rate.

B. Dataset and Features

A balanced synthetic dataset of 2,000 Lalpurja documents was generated, comprising 1,000 genuine and 1,000 tampered documents, and split into training (1,400), validation (300), and test (300) sets as shown in Table II. Base documents were rendered from an HTML template using the Playwright headless browser, with content generated from real-distribution Nepali

TABLE II
SYNTHETIC DATASET COMPOSITION

Split	Genuine	Tampered	Total
Training	700	700	1,400
Validation	150	150	300
Test	150	150	300
Total	1,000	1,000	2,000

data including names, districts, municipalities, Kitta numbers, and land areas. Synthetic corpora are widely used in document-forensics research; Sidère et al. constructed a synthetic payslip corpus with realistic forgeries for forgery detection and spotting [12]. Six tampering methods were simulated with the following shares of the tampered class: field replacement 30%, erasure 22%, signature replacement 15%, font mismatch 13%, seal manipulation 10%, and stamp removal 10%.

From the extracted content, 43 hand-crafted OCR and computer-vision features were derived across eight categories: OCR confidence, per-field verification, layout, grid texture, content, font consistency, global grid, and seal region. Two feature subsets were produced. The *full* model uses 39 of the 43 features; four features were excluded because they require runtime values that are not available in the deployment environment. The full model also includes the per-field verification features, which require expected field values as metadata. The *auto* model uses the 22 metadata-independent features, enabling standalone verification of a document image without manual field entry. This two-model design lets an officer maximize accuracy when expected values are available, while retaining the ability to screen documents automatically when they are not.

C. Model Evaluation

Both models were evaluated on the held-out test set of 300 documents. Table III summarizes the overall results: the full model achieves a test accuracy of 79.7%, a ROC-AUC of 0.8542, and a recall of 70.0% at a 10% false-positive rate; the auto model reaches 68.3% accuracy, a ROC-AUC of 0.7165, and a recall of 43.3% at the same operating point. At the default threshold of 0.5, the full model produced 110 true positives, 21 false positives, 40 false negatives, and 129 true negatives (Table IV), corresponding to a precision of 84.0%, a recall of 73.3%, a false-positive rate of 14%, and an F_1 -score of 78.3%. Table V reports the operating characteristics of the full model across false-positive rate limits; the design operating point of 10% is realized at a threshold of 0.5576, and higher recall can be obtained by accepting a higher false-positive rate.

Per-tamper recall at the 10% false-positive operating point is reported in Table VI. Feature-importance analysis of the trained forest explains the large performance gap between categories. The *field consistency score*—available only when expected field values are supplied—is the most discriminative feature of the full model, which is why it detects field replacement (95.7%) and erasure (100.0%) much better than the auto model (21.3% and 92.3%). For the standalone auto model, the

TABLE III
COMPARISON OF THE FULL AND AUTO TAMPER-DETECTION MODELS

Model	Features	Accuracy	ROC-AUC
v3 full (with metadata)	39	79.7%	0.8542
v3 auto (standalone)	22	68.3%	0.7165

TABLE IV
CONFUSION MATRIX OF THE FULL MODEL ON THE HELD-OUT TEST SET AT THE DEFAULT THRESHOLD OF 0.5

Predicted	Actual	
	Tampered	Genuine
Tampered	110	21
Genuine	40	129

TABLE V
OPERATING CHARACTERISTICS OF THE FULL MODEL ACROSS FALSE-POSITIVE RATE LIMITS

FPR limit	Achieved FPR	Best recall	Threshold
5%	4.0%	59.3%	0.7254
10%	9.3%	70.0%	0.5576
15%	14.0%	74.0%	0.4778
20%	20.0%	79.3%	0.3984

TABLE VI
PER-TAMPER RECALL AT THE 10% FPR OPERATING POINT

Tampering method	Full model	Auto model
Erasure	100.0%	92.3%
Field replacement	95.7%	21.3%
Signature replacement	86.2%	79.3%
Font mismatch	26.1%	30.4%
Stamp removal	18.8%	6.3%
Seal manipulation	0.0%	0.0%

most influential features are char-width coefficient of variation, region brightness variance, block-spacing anomaly, and region contrast variance, which capture the layout and texture artifacts left by erasure and signature replacement but are insensitive to the remaining categories. The hard cases are therefore the ones that hide within normal rendering variation: font mismatch alters glyph shapes in ways that OCR-based features average out, stamp removal leaves a smooth but structurally plausible region, and seal manipulation is essentially undetectable by hand-crafted features—for example, the seal-region features read near zero on real documents because real seal geometry does not match the synthetic seal bounding box used during training. These three categories are explicitly treated as open limitations in Section VI and are the target of the deep-learning-based future work.

D. Model Evolution

The development progressed through three major versions. The first version (v1) was a proof-of-concept trained on 200 samples, which demonstrated feasibility but overfitted due to the limited dataset. The second version (v2) expanded the dataset

to 2,500 samples, but analysis revealed that the model learned acquisition-related image degradation rather than document integrity, since compression, blur, and noise patterns overlapped with tampering effects, producing a high false-positive rate of 28.4%. Version 3 redesigned the experimental methodology by applying acquisition effects equally to genuine and tampered documents and modifying only the document content for the tampered class, forcing the model to learn document-integrity features. The final v3 models are the *full* model (39 features) for maximum accuracy and the *auto* model (22 features) for standalone verification.

E. Structure-Aware Decision Layer

Because the models were trained on synthetic documents, real scanned documents can fall outside the training feature distribution. A document-structure validator was therefore added that checks for the presence of required keywords, field labels, and field values and computes a structure score:

$$\text{structure} = 0.35 \cdot k + 0.30 \cdot l + 0.35 \cdot v, \quad (2)$$

where k , l , and v are the keyword, field-label, and field-value scores, respectively. The weights reflect the diagnostic value of each component: keywords and field values are the strongest indicators that a certificate is legitimate, because a genuine Lalpurja must contain the correct keywords and the expected values in the right locations, whereas field labels are slightly weaker because label wording varies across certificate versions. For ambiguous cases, the final confidence is

$$\text{Conf} = 0.8 \times \text{Structure} + 0.2 \times (1 - \text{TamperProb}). \quad (3)$$

Because the classifier is trained on synthetic documents, structural validity is the more reliable signal when the model and the structure score disagree, hence the larger weight (0.8) on structure; the 0.2 weight on the tamper probability preserves sensitivity to genuine content anomalies. The override fires only within the ambiguous band $0.50 \leq \text{TamperProb} < 0.85$ when the structure score is at least 0.85; a strong model signal of 0.85 or above always wins. These thresholds were selected so that the override changes a verdict only when the model is genuinely uncertain (its probability is below the strong-signal bound) and the document is structurally valid; outside this band the classifier signal is trusted. Applying this layer, a genuine real-world Lalpurja (structure score 0.905, tamper probability 0.7143) that the auto model had flagged was correctly classified as genuine with a confidence of 0.7811, while a random noise image with a structure score of zero was correctly rejected.

V. RESULTS AND EVALUATION

The system was implemented end-to-end and evaluated at multiple levels. This section first describes the completed implementation (Section V-A) and the automated test coverage (Section V-B), then presents the measured performance of the running system (Section V-C), an assessment of the complete workflow (Section V-D), and a comparison with the existing paper-based process (Section V-E).

A. Implementation Results

The frontend was developed with Next.js, React, TypeScript, Tailwind CSS, and TanStack Query, providing responsive, role-aware interfaces for citizens, officers, and administrators. The backend was built with Node.js and Express.js, exposing RESTful APIs for authentication, registration, verification, ownership transfer, and document handling, secured with JWT, role-based authorization, Zod validation, and rate limiting. The PostgreSQL database was fully implemented with the Prisma ORM, covering users, properties, documents, ownership records, transfers, verification records, conflicts, and audit logs. The *PropertyRegistry* smart contract was deployed and verified on the Sepolia test network, with property fingerprints, canonical document hashes, and ownership transitions anchored on-chain and kept synchronized with the database by the blockchain service.

B. Testing

To verify that the components described in Section V-A behave as designed, the system was tested at the unit, integration, API, and smart-contract levels. Business logic for authentication, property registration, documents, transfers, and auditing was covered with Vitest; Express routes were exercised with Supertest, validating request handling, validation errors, and role-based access decisions; smart-contract tests on Sepolia verified `registerProperty`, `recordTransfer`, `verifyProperty`, and input guards including duplicate registration and empty owner identifiers; and the tamper-detection model was evaluated on the held-out test set of 300 documents, as reported in Section IV.

C. Performance Evaluation

The functional tests described in Section V-B were complemented by a latency benchmark of the running backend. The backend was benchmarked in a Dockerized environment against the Sepolia test network. Table VII reports the median latency over ten requests per endpoint. All transactional read/write paths stay below 30 ms; the authentication endpoint is dominated by bcrypt hashing (cost factor 12) and JWT signing, and the blockchain endpoints are dominated by the public JSON-RPC round trip to Sepolia rather than by application logic. Because each endpoint was exercised with a single-request-at-a-time dockerized deployment and a modest sample of ten requests, these figures indicate practical feasibility rather than full production capacity; a proper load test under concurrent traffic is left to future work.

D. System-Level Evaluation

At the system level, the tests and benchmarks above are combined to assess whether the complete workflow satisfies its legal and functional requirements. Table VIII summarizes the evaluation of the complete system against the paper-based process. All legal state transitions are enforced by the state machine, duplicate claims are flagged, ownership transfers complete only with the signatures of the buyer, seller, and officer and are anchored on-chain, and transaction finality is reached in about 12–15 seconds on the Sepolia test network.

TABLE VII
MEASURED API LATENCY (MEDIAN OF TEN REQUESTS)

Endpoint	Operation	Median
GET /health	Liveness check	2.0 ms
GET /auth/me	Authenticated profile	2.7 ms
GET /registry/my	List my properties	14.6 ms
GET /registry/search	Property search	14.3 ms
GET /registry/:id	Property detail	16.3 ms
POST /auth/login	Login (bcrypt + JWT)	≈1.1 s
GET /blockchain/verify/:id	On-chain verification	≈0.85 s
GET /blockchain/ownership/:id	Ownership history	≈0.34 s

TABLE VIII
SYSTEM-LEVEL EVALUATION SUMMARY

Aspect	Outcome
Registration workflow	All legal state transitions enforced by the state machine; duplicate claims flagged with <i>PropertyConflict</i>
Ownership transfer	Completion requires buyer, seller, and officer signatures; on-chain <code>recordTransfer</code> executed on Sepolia
Tamper detection	Full-model ROC-AUC of 0.8542; document hashes verified against on-chain records
Transaction finality	Approximately 12–15 seconds block confirmation on Sepolia

TABLE IX
COMPARISON BETWEEN THE EXISTING PAPER-BASED PROCESS AND THE DEVELOPED SYSTEM

Aspect	Existing paper-based process	Our implementation
Application submission	Manual filling and repeated office visits	Online portal available around the clock
Document handling	Physical originals and photocopies	Digital upload with SHA-256 fingerprinting
Document authenticity	Visual inspection by the officer only	OCR plus ML tamper detection before review
Verification	Manual physical verification	Physical verification preserved and digitally recorded
Ownership records	Paper mutation registers	Blockchain-anchored ownership record
Transparency	Limited public access	On-chain hash verifiable by anyone

E. Comparison with the Existing System

Table IX compares the developed system with the current paper-based process at the Land Revenue Office. Compared with the existing process, the system reduces repeated office visits, provides online application tracking, automates tax assessment, and produces an immutable, tamper-evident ownership ledger while preserving the mandatory physical verification.

VI. CONCLUSION AND FUTURE WORK

This paper presented a blockchain-based solution for land registration and ownership transfer in Nepal that addresses the limitations of the conventional process, including document

forgery, lack of transparency, and inefficient record keeping. The system integrates a modern web application, secure backend services, blockchain technology, and AI-assisted document verification into a single solution while preserving the mandatory physical and biometric verification procedure of the Land Revenue Office. Property fingerprints and ownership transactions stored on the Ethereum blockchain provide a tamper-evident ownership history; only hashes and opaque identifiers are recorded on-chain, so the personal data and documents of citizens remain off-chain. The prototype was validated through functional, integration, API, and smart-contract testing, and the measured API latencies confirm the practical feasibility of the approach.

The results must be interpreted in light of three important limitations. First, the tamper-detection models were trained and evaluated on synthetic Lalpurja documents because a large labelled corpus of real forgery samples is not publicly available; although the structure-aware decision layer partially bridges the synthetic-to-real domain gap, it does not eliminate it. Second, although the full model reaches a ROC-AUC of 0.854 and a recall of 70% at a 10% false-positive rate, detection of the hard cases remains weak: font mismatch (26.1%), stamp removal (18.8%), and seal manipulation (0.0%) are largely undetected by hand-crafted features, and the seal-region features in particular do not transfer to real documents. The AI pipeline is therefore an assisting screening tool, not a substitute for officer verification, and its recommendations should be treated accordingly. Third, the system has been evaluated in a test environment only; it has not been deployed in a real-world setting, and the latency benchmark reflects a small, non-concurrent sample.

Several areas remain open for future work: deployment on a mainnet or a national permissioned ledger with a funded account, gas-cost analysis, and a formal security audit of the smart contracts; strengthening tamper detection for the hard cases using deep-learning-based feature extraction (e.g., CNN or transformer extractors) and a larger, more diverse dataset that better represents real seal and stamp geometry; integration with official government systems such as the National Land Information System and the Land Revenue Information Management System; integration of digital payment gateways and digital signatures; full production load testing under concurrent traffic; and compliance with privacy and public-record regulations.

REFERENCES

- [1] S. Sen, S. Mukhopadhyay, and S. Karforma, "A blockchain based framework for property registration system in e-governance," *International Journal of Information Engineering and Electronic Business*, vol. 13, no. 4, pp. 30–46, 2021.
- [2] M. Biswas, J. A. Faysal, and K. A. Ahmed, "Landchain: A blockchain based secured land registration system," in *2021 International Conference on Science & Contemporary Technologies (ICSCT)*. IEEE, 2021, pp. 1–6.
- [3] T. Ali, A. Nadeem, A. Alzahrani, and S. Jan, "A transparent and trusted property registration system on permissioned blockchain," in *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*. IEEE, 2020, pp. 1–6.
- [4] G. M. Pilare, P. Mohanty, and K. Wasnik, "Land administration system using Hyperledger Fabric," in *2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)*. IEEE, 2022.

- [5] 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*, vol. 5, no. 1, p. 100175, 2024.
- [6] S. Krishnapriya and G. Sarath, "Securing land registration using blockchain," *Procedia Computer Science*, vol. 171, pp. 1708–1715, 2020.
- [7] S. Priya, K. Srilekha, V. Sarojini, P. Tharun, and A. Thinagaran, "Blockchain based land registry system using ethereum blockchain," *International Journal of Engineering & Extended Technologies Research*, vol. 8, no. 2, 2026.
- [8] A. Bisht, N. Teotia, A. Sharma, T. Agarwala, and S. Pandey, "Blockchain-enabled land registration system: A comprehensive review and implementation framework," *SSRN Electronic Journal*, 2025, preprint. [Online]. Available: <https://ssrn.com/abstract=6422578>
- [9] N. Lazuaashvili, 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 Workshops (BPM 2019)*. Springer, 2019, pp. 219–231.
- [10] S. Humdullah, S. H. Othman, M. N. Razali, and H. K. Mammi, "Secured data storage framework for land registration using blockchain technology," in *2021 3rd International Cyber Resilience Conference (CRC)*. IEEE, 2021, pp. 1–6.
- [11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>
- [12] N. Sidère, F. Cruz, M. Coustaty, and J.-M. Ogier, "A dataset for forgery detection and spotting in document images," in *2017 Seventh International Conference on Emerging Security Technologies (EST)*. IEEE, 2017, pp. 26–31.