International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# Advanced Blockchain-Based Secure Academic Certificate Verification System: Trusted Issuer Records, Document Integrity and Signed Attestations

**Spoorthi D, Manasa M K, Bhuvaneshwari H T, Rakshitha P S**
Department of Information Science and Engineering, Aditya College of Engineering and Technology, Bangalore, India
Under the guidance of Prof. Jagdish D, Department of Information Science and Engineering
Corresponding email: dspoorthi11@gmail.com
Co-author emails: manasakushal0@gmail.com, bhuvaneshwariht19@gmail.com, psrakshitha5@gmail.com

**Abstract—Academic certificates are relied on by employers and universities as proof of qualification, yet forged or altered copies, slow manual confirmation with issuing institutions, and fragmented issuer records make that proof difficult to trust. This paper presents the design of CertiChain, an independent academic certificate verification platform for universities, students, employers and administrators. The platform does not issue certificates and does not claim that a blockchain, or an AI score, establishes authenticity. Instead it separates five questions and answers each with its own mechanism: which record is claimed (identification), whether the uploaded document is the recorded document (SHA-256 integrity), whether the issuer is known and approved (issuer authentication), whether the verification result can be checked later (a platform-signed Ed25519 attestation), and whether supporting signals raise concern (heuristic document analysis and an auxiliary blockchain registry). Fixed decision rules ensure that an unknown or unapproved issuer yields "Unable to independently authenticate this certificate" and that a hash mismatch is never overridden by a favourable risk score. The prototype's role-based web interface is implemented; the server-side components are specified, and their status is reported separately. No quantitative evaluation has been carried out, so only design and implementation observations are reported, together with an evaluation plan.**

**Keywords—academic credential verification, document integrity, SHA-256, Ed25519, blockchain registry, role-based access control**

## I. INTRODUCTION

An academic certificate carries a long chain of trust: admission, instruction, examination, result processing and a university that vouches for the result. Employers, other universities and public bodies usually see only a scanned or printed copy, and they must decide whether to rely on it. Systematic reviews of certificate verification describe forgery and slow, manual confirmation as the main motivations behind most proposals for digital solutions [1]. Policy analyses of blockchain in education make a complementary point: a ledger can make tampering with stored records more detectable, but the accuracy of what is recorded still depends on the institutions that record it [2].

This distinction is the practical problem addressed here. A digital copy of a certificate is as easy to edit as any other file, and a certificate number printed on a forged document can be genuine. A verification service therefore has to establish, separately, that the presented document is the one the institution recorded, that the institution is a legitimate and approved issuer, and that its own result can be checked later by someone who was not present. Cryptographic hashes and signatures provide tools for the first and third needs, and distributed ledgers have been proposed as a tamper-evident place to publish digests [3], [4]. None of them can say whether the issuer truly awarded the qualification; that depends on records traceable to the institution.

This paper describes CertiChain, an academic certificate verification platform developed by an undergraduate team. Its contributions are deliberately modest and technical: (i) a verification model that distinguishes identification, document integrity, issuer authentication, attestation and supporting indicators; (ii) fixed decision rules under which failed mandatory checks cannot be overridden; (iii) a clear separation between a platform-generated Ed25519 attestation and a legally recognised signature; and (iv) a role-based web interface for universities, students, verifiers and administrators. The paper separates what is implemented from what is specified or planned, and it reports no performance or user-study results because none have been collected.

The remainder of the paper is organised as follows. Section II reviews related work and states the research gap. Section III presents the methodology: the proposed system, architecture, trust model, verification workflow, decision rules, cryptographic components and implementation. Section IV reports the

implementation observations, Section V discusses them together with an evaluation plan, security, privacy and the expected advantages, and Section VI concludes with the limitations and future scope.

### *A. Problem Statement*

Given institutions that issue academic certificates and parties who must rely on copies of them, the engineering problem is to provide a service through which a verifier can obtain an explainable, repeatable and tamper-evident verification outcome. Four sub-problems follow.

1) *Altered or forged documents.* Editing a name, grade or date is easy, and visual inspection cannot reliably detect it.
2) *Weak identifier-only lookup.* A certificate number alone locates a record but says nothing about the document presented.
3) *Fragmented issuer records.* Institutions keep records in different forms, and no shared, reviewed register of approved issuers exists for verifiers to consult.
4) *Over-trust in technology.* Marketing of blockchain and AI can suggest that a ledger entry or a score proves authenticity, and a platform seal can be mistaken for a legally recognised signature.

The scope is limited to the verification platform and its prototype. Issuing degrees, replacing university records and making hiring or admission decisions are outside it.

## II. LITERATURE REVIEW

### *A. Blockchain for Academic Credentials*

Early work proposed distributed systems for educational records, reputation and reward [5], and later platforms such as EduCTX recorded higher-education credit on a blockchain [6]. Surveys discuss potential uses of the technology in education [7], and a systematic review of certificate-verification systems identifies recurring architectures and open challenges such as scalability, privacy and governance [1]. Across this literature the ledger is usually treated as the place where certificate digests are registered; the quality of the registration process is the open question.

### *B. Revocation Scalability and Fraud Detection*

Tan-Vo et al. combine smart contracts with an optimistic-rollup approach to certificate revocation and a machine-learning component for fraud detection, and report lower cost and latency for revocation [8]. This direction addresses scale and detection. The present work is complementary: it concentrates on the decision rules that govern a single verification and on the honest labelling of each piece of evidence.

### *C. Cryptographic Building Blocks*

SHA-256 is specified in the NIST Secure Hash Standard [9]. Ed25519 is an EdDSA instance designed for speed and security [10], standardised in RFC 8032 [11] and approved in the NIST Digital Signature Standard [12]. Legally recognised signatures in PDF follow profiles such as PAdES [13] and may use trusted timestamps [14]; in India, their legal effect is governed by the Information Technology Act, 2000 [15] and the framework administered by the Controller of Certifying Authorities [16].

### *D. Access Control and Secure Development*

Role-based access control (RBAC) assigns permissions to roles instead of individuals [17], which suits a platform whose universities, students, verifiers and administrators need different privileges. The OWASP Top Ten is a common checklist of web application risks [18].

### *E. Research Gap*

Many proposals treat registration on a ledger as proof of authenticity or focus on issuance rather than on documents already in circulation. Fewer state explicit outcomes for unknown issuers, forbid a favourable analysis score from overriding a failed check, or distinguish an attestation from a university certificate and from a legal signature. The project aims to occupy that space. This statement is limited to the literature reviewed here. Table I summarises the comparison.

TABLE I. COMPARISON OF REVIEWED APPROACHES

| Approach | Main focus | Relevance / gap |
|---|---|---|
| Systematic review [1] | Taxonomy of blockchain certificate systems | Survey; no decision rules |
| Kudos system [5] | Distributed educational records | Concept level; no issuer review |
| EduCTX [6] | Higher-education credit on a ledger | Credit tokens, not document checks |
| Education review [7] | Potential blockchain uses | Context only |
| Rollups + ML [8] | Revocation scale; fraud detection | Layer-2 and ML emphasis |
| This work | Explicit evidence model and decision rules | Implementation status in Table II |

## III. METHODOLOGY

### *A. Overview*

CertiChain is an independent verification service. Institutions register and are reviewed by an administrator; approved institutions supply trusted issuer records that include a trusted SHA-256 digest of each issued document. A verifier or student uploads a certificate, and the backend validates the file, computes its SHA-256 digest from the uploaded bytes, identifies

the claimed institution, checks that it is approved, finds the matching issuer record and compares digests. Registry information, heuristic document analysis and OCR are added as supporting evidence. Only if the mandatory checks succeed does the platform sign an Ed25519 attestation and produce a QR code and report.

### *B. Roles*

The University role registers an institution, submits it for independent review and, after approval, adds verified issuer records. The Student role holds certificates linked to the account and can start a verification. The Verifier (employer) role submits a certificate identifier and document for an independent check and keeps a history of checks. The Administrator role reviews institutions and manages users. A public page allows anyone to submit an identifier and document without an account.

### *C. Implemented Versus Specified Functionality*

Because this distinction determines how later sections should be read, Table II states the status of each function. "Interface implemented" means the screen was observed in the running development build; it does not by itself establish server-side behaviour.

TABLE II. STATUS OF PLATFORM FUNCTIONS

| Function | Status |
|---|---|
| Landing, about and workflow pages | Interface implemented |
| Registration with role selection | Interface implemented |
| Login for institution, student, employer and administrator workspaces | Interface implemented |
| University workspace: institution-review form, issuer-record form | Interface implemented; server logic not yet verified |
| Student workspace | Interface implemented |
| Verifier workspace (ID + document, independent check) | Interface implemented |
| Public verification page (ID + document, up to 10 MB) | Interface implemented |
| Administrator review screen | Not observed |
| Server-side SHA-256 comparison and decision rules | Specified; not yet verified |
| Ed25519 attestation, QR code, PDF report | Specified; not yet verified |
| Blockchain registry integration | Specified; extent not yet verified |
| Heuristic document analysis, OCR | Specified; not yet verified |
| Authorised eSign / DSC / PAdES, trusted timestamp | Future scope |

### *D. System Architecture*

The platform follows the three logical layers in Table III and Fig. 1. The layers are conceptual divisions of the design.

TABLE III. LOGICAL LAYERS

| Layer | Contents | Responsibility |
|---|---|---|
| Presentation | React single-page interface [19] | Role-specific screens, public page |
| Application | FastAPI services [20] | Authentication, validation, verification, signing, reports |
| Data and trust | Relational database, document storage, registry contract, key store | Records, evidence, tamper-evident reference |

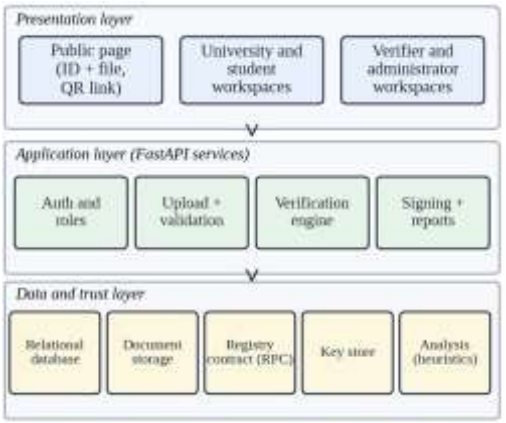



Fig. 1. Logical architecture of the platform (conceptual).

All trust decisions are made by the backend. The interface only collects input and displays the outcome; a modified client therefore cannot turn a failed check into a success. Role restrictions are enforced by the application layer as well as by screen routing.

### *E. Design Principles and Trust Model*

The design rests on five distinct questions, each with a mechanism and a stated limit (Table IV).

TABLE IV. WHAT EACH MECHANISM CAN AND CANNOT ESTABLISH

| Mechanism | Can establish | Cannot establish |
|---|---|---|
| Certificate ID lookup | A record with that identifier exists | That the presented file is the recorded file |
| SHA-256 comparison | Byte-level identity with the trusted document | Who issued it; whether content is true |
| Approved issuer record | Issuer is known and reviewed | That every record is correct |
| Registry entry | A digest was registered and not silently rewritten | That the registrant was entitled to register |
| Ed25519 attestation | The stored result is unaltered since signing | University approval; legal recognition |
| Heuristics / AI cues | Inconsistencies for human attention | Authenticity or forgery |

Two consequences are built into the design. First, a favourable analysis score can add information but can never change a failure into a success. Second, an unknown issuer is reported as impossible to authenticate rather than guessed to be fake, because absence of records is not evidence of forgery.

### F. Verification Workflow

Fig. 2 shows the verification pipeline. The steps are as follows.

1) The user uploads a certificate.
2) The backend validates the file (type, size).
3) The backend computes SHA-256 directly from the uploaded bytes; a client-supplied digest is not trusted.
4) The claimed institution is identified.
5) The institution's approval status is checked.
6) Document information is compared with trusted issuer records.
7) The uploaded hash is compared with the stored trusted hash.
8) Registry information is checked to the extent supported by the implementation.
9) Heuristic analysis and OCR indicators are evaluated as supporting evidence.
10) The system decides whether independent authentication is possible.
11) On success, an Ed25519 attestation is signed.
12) A QR code and PDF report are generated, if implemented.
13) The public page re-checks the stored attestation signature.

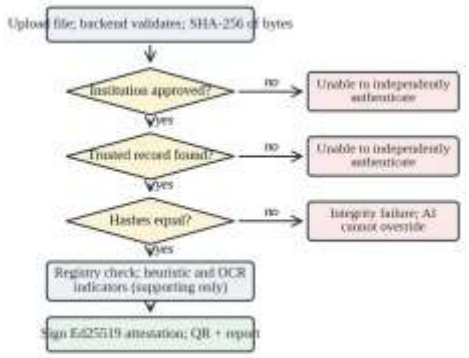



Fig. 2. Verification decision flow (conceptual).

### G. Decision Rules

Table V lists the decision conditions. Rules 1–3 are mandatory checks.

TABLE V. VERIFICATION DECISION CONDITIONS

| No. | Condition | Result |
|---|---|---|
| 1 | Institution unknown or not approved | Unable to independently authenticate this certificate |
| 2 | No matching trusted issuer record | Unable to independently authenticate this certificate |
| 3 | Trusted record found, hashes differ | Integrity failure; not authenticated |
| 4 | Approved, record found, hashes equal | Authenticated on issuer and integrity grounds |
| 5 | Case 4, registry or OCR unavailable | Authenticated; the missing check is reported |
| 6 | Certificate ID supplied without file | Insufficient; integrity cannot be established |

The rules also require that the original document remain unchanged, that a platform attestation never be presented as the original certificate, and that public verification expose no more student information than necessary.

### H. Cryptographic and Evidence Components

#### 1) SHA-256 Integrity Check

The backend computes the SHA-256 digest [9] of the exact uploaded bytes and compares it with the trusted digest held in the issuer record, using a constant-time comparison. Because comparison is byte-exact, a re-exported copy of a certificate will not match even if it looks identical; institutions must therefore register the digest of the original document that is circulated.

#### 2) Ed25519 Attestation

After a successful decision, the platform signs a canonical payload containing a verification identifier, the document digest, institution and certificate references, the outcome, the verification time and a key identifier. The signature is stored with the key identifier so that the public page can select the right public key, and so that old attestations remain checkable after key rotation. Table VI states how this artefact differs from the others.

TABLE VI. CERTIFICATE, ATTESTATION AND LEGAL SIGNATURE

| Property | University certificate | Platform attestation | Legal e-signature |
|---|---|---|---|
| Issued by | University | Platform | Signer with licensed CA certificate |
| Proves | The award | Stored result unaltered | Signer identity and integrity under law |
| Legal effect | Per institution and law | None assumed | Per IT Act [15] |
| Trusted timestamp | N/A | Not provided | Where used [14] |

#### 3) Registry, QR Code and Analysis

A Solidity registry contract may store digests for lookup and revocation; its presence does not prove that a complete workflow operates, so each capability (deployment, registration, confirmation, lookup, revocation, events) must be shown separately. The registry is an auxiliary tamper-evident reference and not proof of authenticity. A QR code [21] links to a public page; the security lies in the signature check that page performs, not in the code itself. Heuristic analysis inspects document metadata and structure, and OCR, which depends on an external engine such as Tesseract [22], extracts text for comparison; when the engine is absent the step must be reported as not performed. These outputs are supporting indicators only.

### *I. Implementation*

#### *1) Environment and Technology*

The project's technology stack consists of a React and Vite frontend [19], a Python FastAPI backend [20], a relational database, a Solidity contract managed with Hardhat for the registry, SHA-256 and Ed25519 cryptography, and containerised deployment. Statements about the backend are design statements unless Table II says otherwise.

#### *2) Frontend*

The running development build provides the following routes: a landing page, an about page, a public verification page, login, registration and three role dashboards (university, student and verifier). Fig. 3 shows the public page and the verifier workspace. The public page accepts a certificate ID and a PDF, JPG or PNG document up to 10 MB, and states that AI analysis is a supporting risk assessment and that authenticity requires an approved issuer record and a matching document hash. The verifier workspace states that a certificate ID alone can locate issuer records and that the document must be uploaded so that its hash can be compared and a signed attestation issued. The university workspace shows an issuer-status banner ("not registered" until independent administrator approval), an institution-verification form with supporting evidence upload (up to 20 MB), and an issuer-record form.

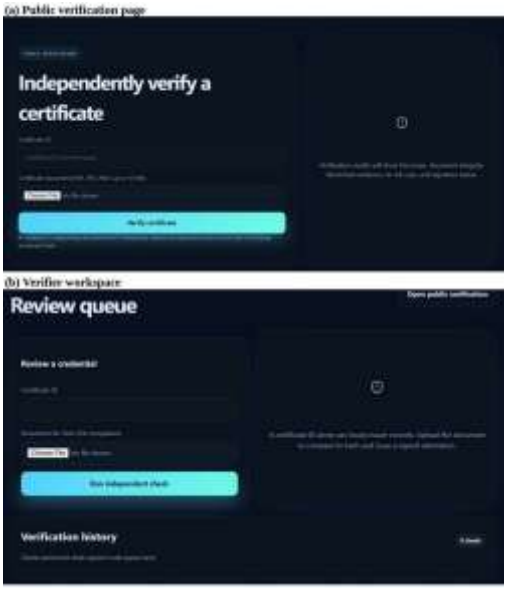



Fig. 3. Screenshots of the running interface: (a) public verification page, (b) verifier workspace.

#### *3) Backend and Cryptography*

The hashing service, decision engine, signing module, report and QR generation, registry client and analysis modules are specified in Sections III-F to III-H. Their implementation status has not been verified and is not asserted here.

## IV. RESULTS

No quantitative testing has been performed and none is reported. In particular, the paper offers no figures for detection rate, false accept or reject rate, response time, throughput, usability scores or user numbers. Table VII lists the implementation observations.

TABLE VII. IMPLEMENTATION OBSERVATIONS

| Observation | Status |
|---|---|
| Role-based registration, login and separate dashboards | Interface observed |
| Institution-review form and issuer-record form | Interface observed |
| Verifier and public pages require ID and document upload | Interface observed |
| Interface text states AI is supporting evidence only | Observed |
| Server-side verification, signing and registry behaviour | Not yet evidenced |

## V. DISCUSSION

### *A. Interpretation*

These observations show that the user-facing foundation of a role-based verification platform exists. They do not show that the platform detects forgery; that claim depends on the server-side pipeline and on test evidence that this paper does not provide.

### *B. Proposed Evaluation Plan*

A future study could build a synthetic corpus of certificates with registered digests and measure (i) the rejection of single-byte and visual edits (expected to be deterministic for the hash check), (ii) correct handling of unknown, unapproved and revoked issuers, (iii) behaviour when the registry node, OCR engine or signing key is unavailable, (iv) verification latency on typical hardware, (v) usability through a standard questionnaire, and (vi) for the heuristic indicators, false-alarm and miss rates on a labelled set, which would first require such a set to be created.

### *C. Security and Privacy Considerations*

*Authentication and authorisation.* Credentials must be stored using slow, salted hashing and sessions must expire. Routing to different dashboards is a user-interface measure; permissions must also be enforced on the server by role and ownership [17].

*Upload handling.* Files should be validated by type, size and content signature, stored without modification and never executed. Rate limiting of upload and public endpoints should be considered.

*Key management.* The signing key must be kept outside the repository and database, with restricted access, backup and a rotation procedure that retains old public keys [23].

*Public-page privacy.* The public page should show only the outcome, issuer name, certificate reference, time and signature status, not the document or unnecessary student details.

*Web risks.* The OWASP Top Ten [18] can guide security testing. No security testing has been reported, and the controls above are requirements and recommendations rather than verified features.

*Data protection.* Student names, e-mail addresses and certificate details are personal data; applicable Indian data-protection requirements must be reviewed before public deployment.

*Long-term verification.* Keeping attestations checkable over time needs secure backups, retention policies, integrity monitoring, key management and rotation, preservation of verification evidence and ongoing maintenance; unconditional lifetime validity cannot be promised.

### D. Advantages

The design offers several potential advantages, all to be validated in practice.

- *Explainable outcomes.* Each result names the check that passed or failed.
- *No identifier-only acceptance.* The document hash is mandatory for authentication.
- *Reusable evidence.* A signed attestation and QR link can be rechecked by third parties holding the platform public key.
- *Honest labelling.* Attestation, certificate and legal signature are kept distinct.
- *Role separation.* Each stakeholder sees only the screens and data its role needs.

## VI. CONCLUSION

This paper addressed the difficulty of trusting academic certificate copies. It presented CertiChain, an independent verification platform whose design separates identification, document integrity, issuer authentication, attestation and supporting indicators, and whose decision rules prevent failed mandatory checks from being overridden. The role-based interface for universities, students, verifiers and administrators is implemented in the prototype; the server-side pipeline, signing and registry components are specified and their status is reported conservatively.

The work has clear boundaries. It does not issue certificates, does not treat a ledger or AI score as proof, does not equate a platform seal with a legal signature, and offers no quantitative evaluation. Its value at this stage is a structured, honest foundation. The next steps are to confirm and test the server-side pipeline, harden authorisation and key management, evaluate with a synthetic corpus and real users, and pursue integration with institutional systems and authorised signature providers.

### A. Limitations

- Only institutions that are approved and supply records can be authenticated; others yield “unable to independently authenticate”.
- Byte-exact hashing rejects re-exported copies.
- The platform seal is not a Digital Signature Certificate, a PAdES signature or a trusted timestamp.
- Heuristic analysis and OCR are supporting evidence; OCR needs an external engine.
- Registry entries do not prove academic authenticity.
- Server-side behaviour has not been verified by testing, and no evaluation exists.
- A platform of this kind needs institutions to join; a small institutional base limits usefulness whatever the technical quality.

### B. Future Scope

None of the following is a current capability.

1) Integration with verified university information systems.
2) External accreditation verification.
3) Complete blockchain registration, confirmation and revocation workflows, if not already present.
4) Integration with an authorised Indian eSign provider for legally recognised signatures.
5) Long-term signature validation using PAdES [13] and trusted timestamps [14].
6) Improved document forensics validated on labelled data.
7) Mobile application, multilingual support and improved archival infrastructure.
8) A structured field evaluation using the measures in Section V-B.

## ACKNOWLEDGMENT

The authors thank Prof. Jagdish D, Department of Information Science and Engineering, Aditya College of Engineering and Technology, for guidance throughout this project.

## REFERENCES

[1] A. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, “A systematic literature review on blockchain-based systems for academic certificate verification,” IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3289598.

[2] A. Grech and A. F. Camilleri, “Blockchain in education,” Publications Office of the European Union, Luxembourg, JRC Sci. Policy Rep. EUR 28778 EN, 2017.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper.

[5] M. Sharples and J. Domingue, "The blockchain and Kudos: A distributed system for educational record, reputation and reward," in Adaptive and Adaptable Learning (EC-TEL 2016), LNCS vol. 9891. Cham, Switzerland: Springer, 2016, pp. 490–496.

[6] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, "EduCTX: A blockchain-based higher education credit platform," IEEE Access, vol. 6, pp. 5112–5127, 2018.

[7] G. Chen, B. Xu, M. Lu, and N.-S. Chen, "Exploring blockchain technology and its potential applications for education," Smart Learn. Environ., vol. 5, art. no. 1, 2018.

[8] K. Tan-Vo et al., "Optimizing academic certificate management with blockchain and machine learning: A novel approach using optimistic rollups and fraud detection," IEEE Access, vol. 12, pp. 168135–168159, 2024, doi: 10.1109/ACCESS.2024.3486029.

[9] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," FIPS PUB 180-4, Aug. 2015.

[10] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," J. Cryptogr. Eng., vol. 2, no. 2, pp. 77–89, 2012.

[11] S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (EdDSA)," IETF, RFC 8032, Jan. 2017.

[12] National Institute of Standards and Technology, "Digital Signature Standard (DSS)," FIPS PUB 186-5, Feb. 2023.

[13] European Telecommunications Standards Institute, "Electronic signatures and infrastructures (ESI); PAdES digital signatures; Part 1: Building blocks and PAdES baseline signatures," ETSI EN 319 142-1.

[14] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato, "Internet X.509 public key infrastructure: Time-stamp protocol (TSP)," IETF, RFC 3161, Aug. 2001.

[15] Government of India, "The Information Technology Act, 2000 (Act No. 21 of 2000)."

[16] Controller of Certifying Authorities, Ministry of Electronics and Information Technology, Government of India. [Online]. Available: https://cca.gov.in

[17] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," IEEE Computer, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[18] OWASP Foundation, "OWASP Top Ten." [Online]. Available: https://owasp.org/www-project-top-ten/

[19] Meta Open Source, "React documentation." [Online]. Available: https://react.dev

[20] "FastAPI documentation." [Online]. Available: https://fastapi.tiangolo.com

[21] ISO/IEC 18004:2015, "Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification," 2015.

[22] R. Smith, "An overview of the Tesseract OCR engine," in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), 2007, vol. 2, pp. 629–633.

[23] E. Barker, "Recommendation for key management: Part 1 – General," NIST Special Publication 800-57 Part 1 Rev. 5, May 2020.