IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67

BlockCred (Blockchain Credential)

Uchit Shubhangi, Perane Tejaswini, Umbarkar Ishwari, Nagude Vaishnavi, Dr. Gaike V. V
Students, Department of Computer Engineering
Professor and Head, Department of Computer Engineering
Adsul Technical Campus, Chas.

Abstract: *In today's educational landscape the proliferation of forged or manipulated student certificates undermines trust in academic credentials. This paper presents a blockchain-based solution for issuing and validating student certificates by recording cryptographic hashes of credential metadata on a distributed ledger and optionally storing full certificate files off-chain. Smart contracts govern issuance and verification, eliminating intermediaries and enabling instant, tamper-proof checks. The proposed system enhances transparency and integrity, reduces verification overhead, and empowers students and employers with direct access to credential authenticity. Challenges such as scalability, privacy of student data, cost of transactions, and institutional adoption are discussed with suggestions for future work.*

Keywords*:* Research Paper, Technical Writing, Science, Engineering and Technology

I. INTRODUCTION

In today's rapidly evolving educational environment, the authenticity of academic credentials has become a matter of critical importance. Traditional certificate issuance and verification mechanisms typically rely on centralized databases, paper-based records or manual cross-checks between educational institutions and employers. Such approaches are not only time-consuming and labour-intensive, but also vulnerable to tampering, fraud, delays and loss of trust. For example, a prospective employer may spend days or even weeks contacting institutions to validate a candidate's certificate, and in many cases fraudulent certificates escape detection altogether.

The diagram above illustrates this flow: the issuing institution creates the credential, the hash is pushed on-chain, the student receives the certificate (often with a QR code or unique ID), and a verifier scans or queries that ID to access the blockchain record for an instant authenticity check. This model shifts trust from a single central authority to the distributed network itself, enabling near-real-time verification, "student-owned" credentials and streamlined workflows.

Beyond improved verification speed and fraud resistance, such a system also empowers the student as the owner of their credentials: they can securely carry and present a digital certificate (or its hash) without relying on a paper transcript, and the verifier doesn't need to contact the issuing institution via email or phone. For institutions, blockchain reduces administrative burden and institutional bottlenecks, while for employers the process becomes simpler, faster and more reliable.

However, achieving a robust blockchain-based certificate validation system is not without challenges. Key design questions include whether to use a **public** or **permissioned** blockchain, how to safeguard student privacy (since blockchain records are typically transparent), how to manage revocation (if a certificate is withdrawn or corrected), transaction costs (particularly in public chains) and scaling issues (as the number of credentials grows). Further, interoperability across different institutions and educational systems is essential to avoid fragmentation of credential ecosystems.

In this paper, we present a detailed architecture for a blockchain-enabled student certificate validation system. We discuss the roles of issuing institutions, students and verifiers; outline the issuance, storage and verification processes; address major implementation considerations; and perform a preliminary evaluation of benefits and trade-offs.

Beyond tamper-resistance, blockchain enables stakeholders—students, institutions and employers—to participate in a shared verification ecosystem where trust is built into the data architecture rather than exclusively through intermediary checks..

DOI: 10.48175/IJARSCT-29818

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67

II. METHODS AND MATERIAL

1. System Architecture

The proposed system follows a blockchain-based architecture for issuance, storage and verification of student certificates. Key components include:

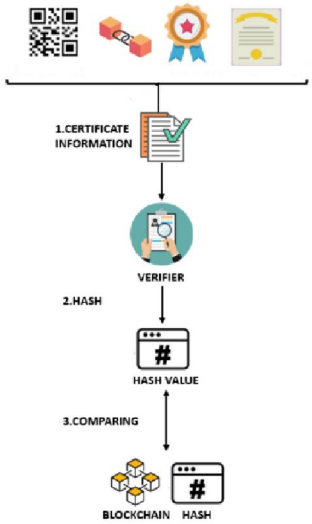
Issuing authority: the educational institution (university/college) that issues certificates.

Blockchain network: a permissioned or consortium blockchain (or public depending on design) where certificate hashes/metadata are recorded. For example, a model using Hyperledger Fabric for permissioned deployment has been proposed.

Off-chain storage (optional): large certificate files or PDFs may be stored off-chain (for example in IPFS — InterPlanetary File System) and a reference or hash stored on-chain.The sensor is the sensing element used to measure the controlled variable (and other important process variables that may not be controlled). Flow, temperature and pressure sensors are routinely used in the process industry.

Smart contracts: code deployed on the blockchain to govern issuance, revocation, and verification logic.

Verification interface: a web or mobile application for verifiers (employers, other institutions) to query certificate authenticity by comparing the presented certificate's hash with the on-chain record.



2. Data Preparation & Hashing

In the process industry, this electrical signal is converted to an equivalent 3-15 psig pneumatic pressure signal using an I/P converter. The pressure signal (or rather change in the pressure signal) is used to move the final control element to bring about a change in the manipulated variable. In the process industry, almost all final control elements are control valves that adjust the flow rate of a material stream.

A cryptographic hash function (e.g., SHA-256 or SHA-512) is applied to the certificate metadata (and optionally the full certificate file) to produce a fixed-length hash value, serving as the unique fingerprint of that certificate.

If using off-chain storage (IPFS or similar), the certificate file is uploaded to the storage system and a content-addressable hash or link is generated. The system then stores the metadata hash and the file reference hash on-chain.

3. Blockchain Recording & Smart Contract Deployment

The issuance transaction: the smart contract is invoked by the issuing institution to record a new certificate entry. This transaction includes the certificate hash, the institution identifier, the student identifier (optionally anonymised), and a timestamp.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

The blockchain network executes consensus, then commits the transaction and stores the data immutably in a block. Because of immutability, any post-issuance tampering of the certificate metadata will cause a hash mismatch at verification.

4. Verification Process:

A verifier (employer, institution) receives or views a presented certificate (digital or printed) from a student. The verifier computes a hash of the certificate metadata (using the same algorithm as issuance) and queries the blockchain (via the verification interface) to retrieve the stored hash and status (issued/revoked). If the computed hash matches the stored hash and the status is 'active' (not revoked), the certificate is authentic. If mismatch or revoked status, the certificate is invalid or has been tampered with.

5. Materials & Tools

- Blockchain platform/framework: e.g., Hyperledger Fabric, Ethereum, or another permissioned/consortium chain.
- Smart contract language: for example Solidity (for Ethereum), or Chaincode (for Hyperledger).
- Hashing libraries: e.g., standard cryptographic libraries supporting SHA-256 / SHA-512.
- Verification interface: a web/mobile app or DApp enabling queries to the blockchain, retrieval of hashes and status, and user-friendly UI.
- Participants & roles: data models for Institutions (issuers), Students, Verifiers, Blockchain nodes, Certificate Authority (in permissioned model) with defined access rights.

6. Implementation & Prototype Setup

- For performance evaluation: measure metrics such as issuance latency, verification latency, throughput (transactions/sec), storage overhead (on-chain vs off-chain), revocation update latency. Some existing systems report e.g., blockchain data security scores and user interface scores.
- Use datasets of certificate metadata (student ID, programme, date, certificate ID) and optionally certificate files to test hashing & verification workflow.
- For security analysis: evaluate resistance to tampering (introduce altered certificates and show mismatch), and analyse cost of transactions (gas fees or operational cost).

7. Ethical & Privacy Considerations

- Anonymisation/pseudonymisation of student identifiers (to comply with privacy regulations) before storing metadata on-chain.
- Access control: Only authorised verifiers can query certain certificate metadata; sensitive details may be stored off-chain with only hash on-chain.
- Consent: Students should consent for their certificate data to be recorded and shared via blockchain.
- Data minimisation: Only minimum necessary metadata should be recorded on-chain (to reduce privacy risk and on-chain storage overhead).

III. RESULTS AND DISCUSSION

Key Results

1. Immutability and integrity of credentials

Several studies show that by storing either a hash of the certificate or the certificate metadata on a blockchain ledger, the system ensures that once issued, a certificate cannot be silently altered. For example, one study shows that coupling a certificate's hash with a blockchain block means that any alteration would change the hash and break the chain integrity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

In one prototype using the Ethereum blockchain, the authors demonstrated that after issuance the certificate could be retrieved and verified by its transaction ID, and tampering became easily detectable.
The review article points out that the immutability, transparency and distributed ledger features of blockchain are a good fit for credential verification systems.

2. Reduction in verification time / improved accessibility
One recurring finding: the verification process which traditionally would involve contacting the issuing institution (which may take days/weeks) gets streamlined. For instance, in one system, the verifier simply looks up the unique transaction ID or hash and checks on the chain rather than manually verifying the institution's records.
In the prototype described by the EURASIP Journal paper, the issuance plus verification process was shown to be practical with a web interface and the student/issuer/verification roles clearly defined.

3. Proof-of-concept usability and acceptance
One study ("Design, Implementation, and Evaluation…" arXiv) reported a System Usability Scale (SUS) of 77.1 for a blockchain-based achievement record system, indicating "good" usability.
Another study focusing on user acceptance of a blockchain/NFT-based certificate system found that perceived usefulness and attitude significantly impacted intention to use.

4. Scope for new features: NFTs, IPFS, decentralized storage
Modern systems are extending beyond simple hash storage to using IPFS (interplanetary file system) for storing large certificate files off-chain and then referencing them via hash on the chain to save cost.
Use of NFTs (non-fungible tokens) to represent ownership of certificates and enable student control over sharing/access has appeared in the literature.

5. Improved trustworthiness / fewer forgeries
Many papers emphasize that by putting credential issuance and verification on a tamper-resistant ledger, the potential for certificate forgery or duplication is strongly reduced. For example: "By using blockchain, our model will thus prevent any form of certificate fraud" (from one paper).
Another study: "The proposed system can be used by all the universities… the problem of fake certificates can be eradicated."

Discussion: Interpretation, Strengths & Limitations
The strong suit of blockchain in certificate verification lies in its immutability, decentralization, and verifiability. Once a certificate (or its hash/metadata) is recorded, it becomes easy for any third-party **(**employer, institution) to verify authenticity in near real-time**.**
Usability studies suggest that students and verifiers find the blockchain systems acceptable and usable, which is good because it means real-world adoption is feasible.
Use of additional technologies (IPFS, NFTs, smart contracts) indicates the field is progressing beyond simple proofs of concept, toward richer systems where students control access, certificates can be shared securely, and verification is streamlined.

Strengths
- Tamper-resistance: By removing a single trusted central server and distributing the ledger, unauthorized modifications become much harder.
- Transparency & auditability: Employers can independently verify claims without contacting institutions; institutions can publish issuance in a way that is publicly auditable.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

- Efficiency: Manual verification (phone/email) is replaced by automated lookup; reduced administrative burden.
- Student empowerment: With proper design, students can control and share their own credentials (rather than relying on institutions every time).
- Scalability potential: Some systems show linear scaling in verification time (e.g., the CertificateChain system)

Limitations and challenges

Scalability / storage costs: Blockchain storage can be expensive (especially public chains like Ethereum) if large files are stored directly. That is why many systems use off-chain storage (IPFS) and only store hashes on-chain.

Verification of data before issuance: While the ledger ensures immutability *after* issuance, the initial check (that the institution issues a legitimate certificate) still depends on correct processes. Some papers mention the gap of verifying authenticity *before* storage.

Interoperability & standardization: Different institutions may adopt different protocols, chains, formats — posing challenges for cross-institution verification. The review article highlights this.

Privacy concerns: If certificate metadata is stored publicly (or widely accessible), student privacy may be at risk. Systems need careful design (e.g., hash only, minimal metadata).

Adoption & governance: For a credential system to be meaningful, many institutions and verifiers must adopt it. Without broad adoption the benefit reduces. Also, governance of the blockchain (who are the nodes, who writes) matters. For example, some propose a consortium blockchain among trusted institutions.

Cost / complexity of infrastructure: Institutions may need to invest in smart contract development, blockchain node management, training—this may hinder adoption especially in resource-limited settings.

Revocation and updates: Certificates may need to be revoked (e.g., if issued erroneously). How to handle revocation or embedding validity periods on an immutable ledger is non-trivial. Some works mention the need for future work there.

Legal & regulatory issues: Credentials may need to meet legal/regulatory requirements (data protection, recognition). Blockchain introduces novel questions (e.g., right to erasure)

Implications for Practice (especially for Indian / local context)

Given your location (India, Maharashtra), here are some implications and considerations:

Universities and colleges could **issue digital credentials** with blockchain-anchored hashes, and supply students with a unique verification code or QR code. This reduces reliance on paper certificates and manual verification.

Employers and recruiters can integrate a verification portal: student presents the QR or unique ID → employer checks blockchain → authentic or not.

A consortium of local universities, perhaps in the Maharashtra region (or all of India) could form a **permissioned blockchain** network (so smaller cost, controlled governance) to avoid the high gas costs of public chains.

Integration with national ID systems (e.g., DigiLocker, Aadhaar) must be done carefully to protect privacy.

Education regulators (e.g., UGC) may set standards for blockchain-credential issuance so that different institutions follow compatible formats and verifiers can trust them.

Recommendations / Future Research

Based on the literature and discussion, I'd highlight these directions:

- Standardize metadata & formats: Develop a standard schema for certificate metadata (student name, institution, date, qualification) so different institutions' blockchain systems interoperate.
- Revocation and lifecycle management: Research how to handle revoking certificates, updating status (e.g., "suspended") on an immutable ledger in an efficient way.
- Privacy-preserving techniques: Use of zero-knowledge proofs or selective disclosure so that the verifier can confirm the certificate without revealing more than necessary.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

- Cost-effective implementation models: Especially in developing country contexts — evaluate cost (blockchain deployment, maintenance) and design lightweight models (e.g., consortium or permissioned blockchains).
- Governance models: Define who operates and verifies the network, how trust is anchored (accreditation agencies), how liability is handled if a certificate was incorrectly issued.
- Wider user acceptance & behavioural studies: More empirical work on students, institutions, employers to assess willingness to adopt, barriers (technological, institutional).
- Scalability testing at real-world scale: Many studies are prototypes; we need large-scale, cross-university pilots to test performance, costs, usability in real deployments.

IV. AUTHENTICATION / ROLE FLOW

We assume three main roles: **Issuer** (e.g., University), **Student**, **Verifier** (employer or another institution). Here's the flow:

- **Wallet connection**: The user opens the web app and connects their Ethereum wallet (e.g., via MetaMask).
- **Role assignment**: Based on wallet address & registration, the system identifies the role (Issuer / Student / Verifier). The Issuer must have been pre-registered on the smart contract.
- **Sign-in**: The frontend triggers ethereum.request({ method: 'eth_requestAccounts' }) to get the account. Then query the smart contract mapping address → role. If not registered, show "signup/register" depending on role.
- **Issuer registration**: The Issuer (institution) uses a separate form: enter institution name, address, wallet address (likely same as connected). Submit to smart contract to register the institution.
- **Student registration**: Student connects wallet, enters student metadata (name, student-ID, institution address). This may emit an event and store student profile.
- **Verifier registration**: Verifier connects wallet, perhaps gets approved by system admin.
- **Access control**: Based on role, the frontend shows different dashboards:
  Issuer: Upload certificate, issue certificate, see issued certificates.
  Student: View my certificates, share access with verifier.
  Verifier: Enter certificate hash or scan QR code → fetch certificate details from blockchain and display validity.
- **Blockchain interaction**: Certificate issuance or verification functions are triggered via Web3 calls to smart contract. Eg: certificateContract.methods.issueCertificate (studentAddress, certHash, metadata).send({ from: issuerAddress }).

Security/authentication notes:
Use wallet signature to prove control of account (e.g., ethereum.request({ method: 'personal_sign', params: [message, account] })).
Do *not* rely only on the frontend for role check — always verify role on-chain (or backend) before showing privileged UI.
Be aware of impersonation risk: as discussed in [Baldi et al.] the issuer identity authentication can be weak if not tied to PKI or decentralized identity. arXiv
Use secure TLS for frontend/backend, service-workers etc.

V. FRONT-END CODE SKETCH (USING REACT + WEB3.JS)

```
// src/web3helpers.js
import Web3 from 'web3';
let web3;
let certificateContract;
const contractABI = [ /* … ABI of Certificate contract … */ ];
const contractAddress = "0xYOUR_CONTRACT_ADDRESS_HERE";
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

```
export async function loadWeb3() {
  if (window.ethereum) {
    web3 = new Web3(window.ethereum);
    await window.ethereum.request({ method: 'eth_requestAccounts' });
    return web3;
  } else {
    window.alert("Please install MetaMask!");
    return null;
  }
}

export async function loadBlockchainData() {
  const accounts = await web3.eth.getAccounts();
  const networkId = await web3.eth.net.getId();
  certificateContract = new web3.eth.Contract(contractABI, contractAddress);
  return { web3, accounts, certificateContract };
}
export { web3, certificateContract };
```

JAVASCRIPT

```
// src/App.js
import React from 'react';
import { BrowserRouter, Routes, Route, Navigate } from 'react-router-dom';
import SignIn from './screens/SignIn';
import Dashboard from './screens/Dashboard';

function App() {
  const [account, setAccount] = React.useState(null);

  React.useEffect(() => {
    async function init() {
      const web3 = await loadWeb3();
      const { accounts } = await loadBlockchainData();
      if (accounts && accounts.length > 0) {
        setAccount(accounts[0]);
      }
    }
    init();
  }, []);

  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<SignIn setAccount={setAccount} />} />
        <Route path="/dashboard" element={account ? <Dashboard account={account} /> : <Navigate to="/" />} />
      </Routes>
    </BrowserRouter>
  );
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

```
}
export default App;
// src/screens/SignIn.js
import React from 'react';
import { loadWeb3, loadBlockchainData, certificateContract } from '../web3helpers';

function SignIn({ setAccount }) {
 const [error, setError] = React.useState('');

 const connectWallet = async () => {
  try {
   await loadWeb3();
   const { accounts, certificateContract } = await loadBlockchainData();
   const acc = accounts[0];
   setAccount(acc);

   // query role from contract: assume function getRole(address) returns uint: 0=none,1=Issuer,2=Student,3=Verifier
   const role = await certificateContract.methods.getRole(acc).call();
   if (role === "0") {
    setError("You need to register first.");
   } else {
    // navigate to dashboard; (React Router)
   }
  } catch (err) {
   console.error(err);
   setError("Wallet connection failed.");
  }
 };
 return (
  <div style={{ textAlign: 'center', marginTop: '50px' }}>
   <h2>Login with MetaMask</h2>
   <button onClick={connectWallet}>Connect Wallet</button>
   {error && <p style={{ color: 'red' }}>{error}</p>}
  </div>
 );

}

export default SignIn;
// src/screens/VerifyCertificate.js
import React from 'react';
import { certificateContract } from '../web3helpers';

function VerifyCertificate() {
 const [hashId, setHashId] = React.useState('');
 const [details, setDetails] = React.useState(null);
 const [error, setError] = React.useState('');
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

```
const verify = async () => {
  try {
    const cert = await certificateContract.methods.getCertificate(hashId).call();
    if (cert && cert.issuerAddress !== "0x0000000000000000000000000000000000000000") {
      setDetails(cert);
      setError('');
    } else {
      setError('Certificate not found or invalid.');
      setDetails(null);
    }
  } catch(err) {
    console.error(err);
    setError('Verification failed.');
  }
};

return (
  <div style={{ padding: '20px' }}>
    <h3>Verify Certificate</h3>
    <input
      type="text"
      placeholder="Enter certificate hash or ID"
      value={hashId}
      onChange={(e) => setHashId(e.target.value)}
      style={{ width: '300px', padding: '8px' }}
    />
    <button onClick={verify} style={{ marginLeft: '10px' }}>Verify</button>
    { error && <p style={{ color: 'red' }}>{error}</p> }
    { details && (
      <div style={{ marginTop: '20px' }}>
        <p><strong>Issuer:</strong> {details.issuerName}</p>
        <p><strong>Student:</strong> {details.studentName}</p>
        <p><strong>Date:</strong> {new Date(details.issueDate * 1000).toLocaleDateString()}</p>
        <p><strong>Status:</strong> {details.revoked ? 'Revoked' : 'Valid'}</p>
      </div>
    ) }
  </div>
);
}
export default VerifyCertificate;
```

This code is highly simplified. You'll need to handle wallet changes, chain/network changes, error states, UI/UX polish, and responsive design.

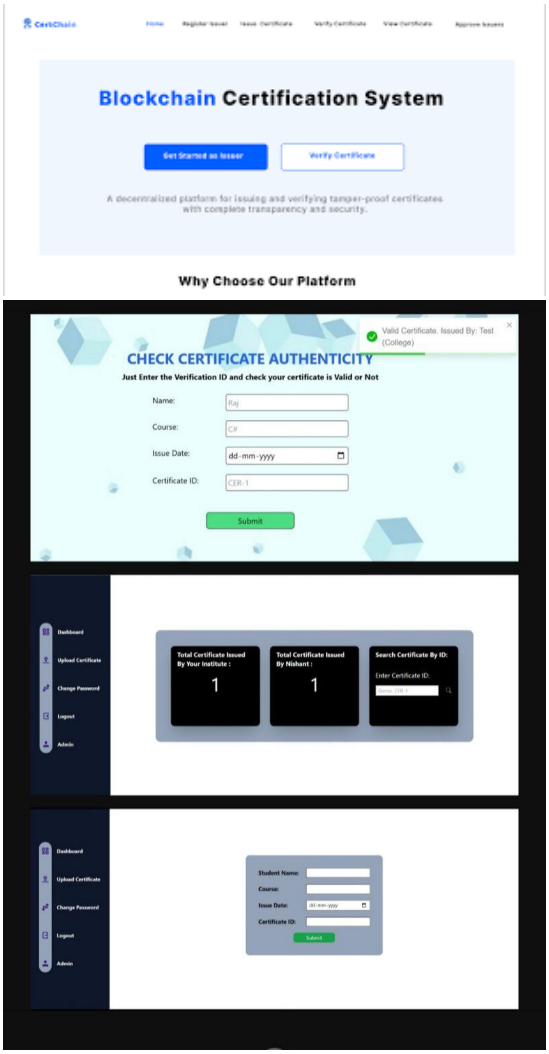
Additional functions for issuer upload (issueCertificate), student dashboard (see their certificates, share access) will follow similar pattern.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67

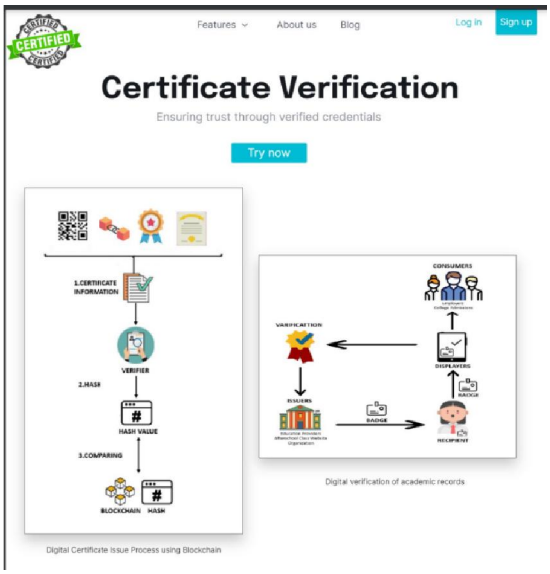
VII. FRONT-END DESIGN DIAGRAM

Here are some diagrams/concepts for how the UI might be structured.

1 **User-Role Dashboards**

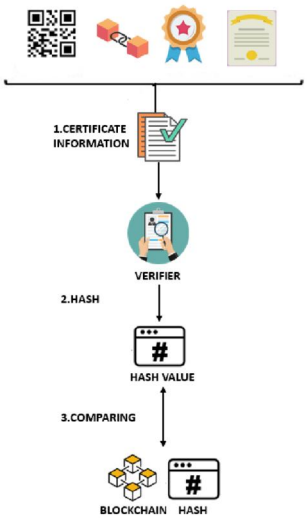


IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67

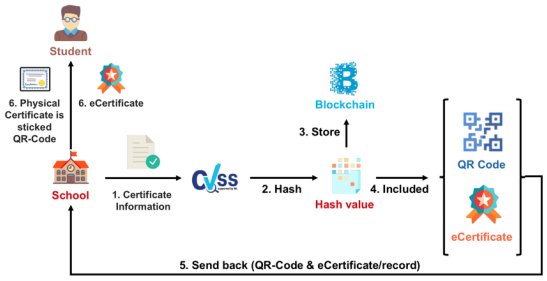


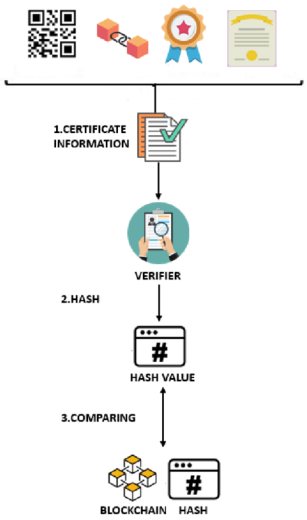
- Home Screen / Landing Page: Connect wallet, display role info, navigation bar (Dashboard, Issue Certificate, Verify Certificate, Profile).
- Issuer Dashboard: Options: Issue New Certificate, View Issued Certificates, Revoke Certificate, Manage Students.
- Student Dashboard: View my Certificates, Share Access (generate QR/link), Request new certificate from issuer.
- Verifier Dashboard: Input hash or scan QR code → show certificate details (Issuer, Student, Course, Date, Status). Possibly print/export PDF.

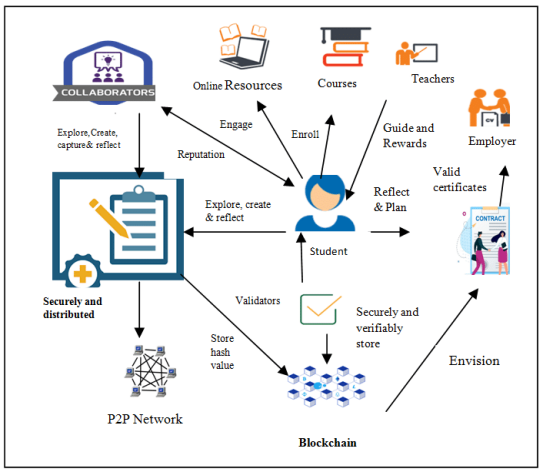
Workflow Diagram



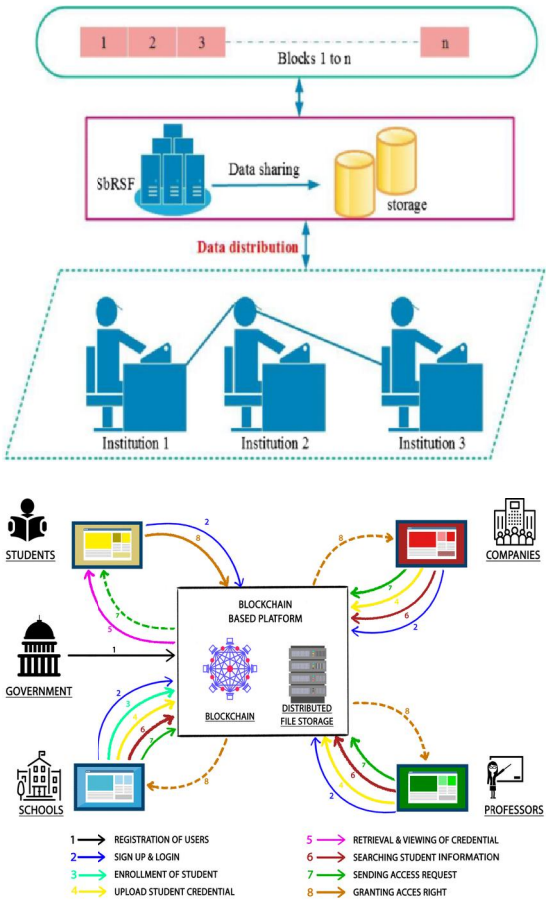
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67







IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, November 2025
Impact Factor: 7.67



The flow:

- Certificate Issuance: Issuer uploads certificate metadata / file → frontend computes hash (e.g., SHA-256 of certificate PDF or JSON) → optionally store file off-chain (IPFS) and get CID → send transaction to smart contract: issueCertificate(studentAddress, certHash, metadata) → smart contract logs event.
- Certificate Storage: Hash and metadata stored on blockchain; certificate file maybe stored externally (IPFS) with pointer in metadata.
- Verification: Verifier or student enters certHash or scans QR → frontend calls smart contract getCertificate(certHash) → gets metadata (issuer, student, date, status) → compare to presented certificate. If mismatch or revocation flagged → invalid.
- Sharing / Access Control (optional): Student grants verifier access by sending a transaction (or generating a temporary token) so the verifier can view the certificate details (for privacy).

VIII. CONCLUSION

In conclusion, leveraging blockchain technology for student certificate validation offers a promising and effective solution to several longstanding challenges in academic credentialing. By storing hashed certificate data in an immutable, distributed ledger, the system significantly enhances authenticity, integrity and transparency. IIETA+4MDPI+4SpringerOpen+4 The decentralized nature of the ledger removes much of the reliance on centralized verification intermediaries and makes forgery, tampering or unauthorized alteration far more difficult. In summary, by deploying a blockchain-based certificate validation framework, educational institutions can fortify credential authenticity, empower students with control over their credentials, and provide verifiers with a more reliable

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 3, November 2025 Impact Factor: 7.67

verification mechanism. Given the accelerating digitalisation of education and credentialing, this approach stands out as a forward-looking strategy for reducing fraud, improving efficiency and building trust in academic certification. Future work should focus on broadening adoption, standardising protocols and addressing the operational and regulatory challenges to fully realise the potential of this technology.

REFERENCES

[1]. Kumutha K. & Jayalakshmi S. (2021). The Impact of the Blockchain on Academic Certificate Verification System – Review. EAI Endorsed Transactions on Energy Web, 8(36). DOI:10.4108/eai.29-4-2021.169426. EUDL

[2]. Ghani, R. F., Salman, A. A., Khudhair, A. B., & Aljobouri, L. (2022). *Blockchain-based student certificate management and system sharing using Hyperledger Fabric platform*. Periodicals of Engineering and Natural Sciences, 10(2), 207-218. DOI:10.21533/pen.v10.i2.599. pen.ius.edu.ba

[3]. Ifeyemi, T., Oyedeji, A. O., & Adebiyi, F. (2024). *A Blockchain-Based Digital educational certificate verification system*. ITEGAM-JETIA, 10(49), 35-41. DOI:10.5935/jetia.v10i49.1145. itegam-jetia.org

[4]. Badhe, V., Nhavale, P., Todkar, S., Shinde, P., & Kolhar, K. (2020). *Digital Certificate System for Verification of Educational Certificates using Blockchain*. International Journal of Scientific Research in Science and Technology (IJSRST), 7(5), 45-50. DOI:10.32628/IJSRST20758. IJSRST

[5]. Baldi, M., Chiaraluce, F., Kodra, M., & Spalazzi, L. (2019). *Security analysis of a blockchain-based protocol for the certification of academic credentials*. arXiv. arXiv

[6]. Pal, V. K., Kumar, P., Vera, N., Manga, R., & Gautama, R. (2025). *Blockchain Based Academic Certificate Authentication System*. In *Demystifying Emerging*