iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.79395

www.ijraset.com

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

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

Blockchain-Based Academic Certificate Verification System

Ruturaj Deshmukh[1], Prasanna Thulkar[2], Prathamesh Somwanshi[3], Suraj Nimbalkar[4], Mrs. Nisha Jagadale[5]
Dept. of Computer Science VPKBIET, Baramati, India

Abstract: ***Counterfeit academic certificates mass production leads to ethical challenges, making people lose confidence in both educational organizations and global employers. The current methods of certificates' validation rely on central databases and administrative procedures, which makes them extremely time- consuming, costly, and vulnerable to single point of failure. Thus, a scalable and fully decentralized approach towards academic certificate verification based on Ethereum blockchain technology and IPFS is proposed, designed, and examined within this paper. The system architecture relies on backend written with the help of Node.js, frontend implemented in React, and lightweight yet safe communication between blockchain and client-side performed by Ethers.js library. Through implementation of a properly designed strategy of research, the roles for each participant of the application: Issuer, User, and Verifier are defined. In order to reduce excessive fees for storing data in the blockchain, an approach of including files in IPFS and keeping only their corresponding CIDs within Ethereum contract is implemented. Finally, support for automatic QR-code generation to make verification easy and reliable is included in the system. Tests were performed to verify the reliability and efficiency of the system, using 300 academic certificates and measur- ing its ability to work efficiently when handling up to 400 concurrent users. The tests show the speed of 10 to 20 seconds for a transaction and minimum gas cost of about 0.49percertificateatproductionleveldeployment. The results obtained clearly confirm that the proposed ap- proach allows to ensure a highly reliable data security, completely exclude any possible counterfeiting of certificates, and provide an economically efficient approach to modern academic credentials management.***

Index Terms: Blockchain, Academic Certificates, Verification Systems, Smart Contracts, Ethereum, IPFS, Node.js, Ethers.js, Hardhat, Decentralization.

I. INTRODUCTION

A. Background

The education sector has played an indispensable role in fostering technological and social progress, and academic cer- tificates have acted as credible proof of an individual's qualifications, abilities, and accomplishments. Academic certificates find widespread application in educational and employment processes, providing a necessary chain of trust between the issuing organization, the graduate, and the employing entity. Historically, the process of verifying certificate authenticity has relied on physical documents with embedded security or centralized digital databases under the authority of the insti- tutions themselves. However, such methods involve lengthy and expensive procedures that involve manual verification via administrative authorities and third-party organizations. Furthermore, centralized databases are highly susceptible to cyber attacks and data breaches since they rely on a single point of control. Blockchain technology presents an innovative solution to the shortcomings associated with conventional methods. By utilizing a distributed ledger, blockchain technology avoids single points of failure and guarantees data integrity. The combination of blockchain technology and IPFS creates a transparent and trustworthy system for verifying academic certificates.

B. Motivation

Although the advantages of blockchain technology are widely recognized, the motivation behind developing this Blockchain-Based Academic Certificate Verification System arises from three major real-world challenges: the growing impact of credential fraud, inefficiencies in traditional verifi- cation systems, and the high operational costs of blockchain- based solutions.

1) Economic and Social Impact of Credential Forgery: With the rapid advancement of digital editing tools and high- quality printing technologies, the creation of counterfeit aca- demic certificates has increased significantly. Fake credentials undermine the credibility of the global workforce and can lead to serious financial losses for organizations, with studies esti- mating an average loss of around $15,000 per fraudulent hire. Beyond financial damage, the issue also raises serious safety concerns, especially when unqualified individuals secure roles in critical sectors such as healthcare, engineering, or finance. Addressing and eliminating credential forgery is therefore a key motivation for this work.

2) Eliminating Administrative Inefficiencies: Traditional certificate verification methods are largely manual, making them slow, error-prone, and dependent on institutional working hours. During peak periods such as mass recruitment or university admissions, these processes create significant delays and operational bottlenecks. This highlights the need for an automated solution that enables instant verification. By lever- aging technologies such as QR codes, the proposed system allows seamless and real-time validation without requiring continuous involvement from issuing authorities.
3) Reducing the Cost of Blockchain Implementation: Despite its strong security guarantees, blockchain adoption in academia is limited due to high operational costs, particularly when storing large files directly on-chain. Storing certificate data such as PDFs on Ethereum leads to high gas fees, making large-scale deployment impractical. To overcome this, the proposed system adopts a hybrid approach by storing documents on IPFS and recording only their corresponding hashes and Content Identifiers (CIDs) on the blockchain. This significantly reduces costs while maintaining security, making the solution practical and scalable for widespread use.

C. *Drawbacks in Conventional Systems*

As of now, verification of educational qualifications involves checking information stored in centralized databases, hosted by either the institution awarding certificates or independent background verification agencies. Although such conventional methods have prevailed as the global standard for decades, the infrastructural and procedural vulnerabilities plaguing them make them increasingly unreliable:

1) Single Point of Failure and Inherent Security Risks: Centralized databases involve storing data in a single server or a network cluster managed by a particular authority. Should the centralized server experience any form of DDoS attacks, hardware faults, or security breaches, the entire infrastructure will be compromised or rendered unresponsive.
2) Vulnerability to Insider Threats and Alterations: The database systems being run manually require human intervention, in which case the system's administrators can access overarching credentialing keys. Such de- pendence on human trust makes traditional databases vulnerable to malicious insiders and administrators who may manipulate information.
3) Administrative Bottlenecks and Increased Latency Times: Verification of candidate's background history requires human interaction—such as sending emails to the registrar at the relevant university and physically searching for the required documents in archives. This process is inherently time-consuming and takes days or even weeks to be completed, causing major bottlenecks in hiring.
4) Extremely High Cost of Management: To ensure data safety and reduce security risks, institutions must invest in robust server infrastructures, regular updates of fire- walls and antivirus software, and employing dedicated verification specialists, leading to high expenses.

D. *Blockchain Framework*

Blockchain technology is based on the revolutionary paradigm of decentralized verification using cryptography and distributed consensus algorithms . Essentially, blockchain in- volves maintenance of the ledger on the peer-to-peer network of computers rather than a centralized server. By distributing transactions among multiple nodes, blockchain mathematically eliminates the vulnerability to hacking due to the absence of the single point of failure characteristic of traditional databases.

In terms of certificate verification, blockchain framework provides a reliable and "trustless" platform. Upon issuance of the certificate, all information about a document is encrypted in a process known as hashing. Once generated, the crypto- graphic signature is written on the block, making it permanent and irreversible due to consensus protocols.

Additionally, a blockchain-based system employs self- executing programs known as "smart contracts." Whenever a malicious user attempts to alter a single element of the certificate, the resulting hash will inevitably differ. As soon as this difference is noted by a smart contract, it is flagged as a manipulation, allowing a verifier to reject a forged document . The absolute immutability of data stored on a blockchain allows employers to verify a document within seconds regardless of the issuing institution's involvement.

II. SYSTEM ARCHITECTURE AND CORE METHODOLOGY

The architectural design of the Blockchain-Based Academic Certificate Verification System has been carefully structured to support scalability as a hybrid decentralized application (DApp). The solution seamlessly combines conventional web services with decentralized cryptographic mechanisms to de- liver a convenient and intuitive user experience. For proper separation of concerns and system reliability, the architecture has been logically divided into several distinct layers that work together toward a common goal.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

These essential technical layers are the Presentation Layer (Frontend), the Orchestration Layer (Backend), the Decen- tralized Storage Layer (IPFS), and the Blockchain Layer. By distributing responsibilities among these different layers, the system effectively deals with user interaction, large file stor- age, and cryptographic state transitions without overloading any particular network node. This separation also enables efficient resource utilization and allows each layer to scale independently based on demand. As a result, the system can handle increased workloads without compromising performance or security. It further ensures a smooth integration between Web2 and Web3 technologies.

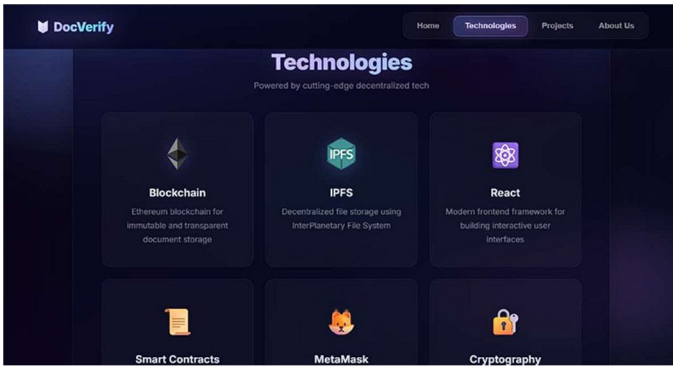


Fig. 1. Key technologies behind the hybrid decentralized document verifica- tion architecture.

A. Presentation and Interaction Layer (Frontend)

The Presentation Layer acts as a crucial mediator between the end-user and the blockchain infrastructure. The move from Web2 to Web3 often poses a high risk of user experience (UX) friction because of issues related to cryptographic wallets, gas, and latency. Thus, the client interface of the app has been developed as a very efficient SPA using React.js.

This technology offers a lot of advantages. In particular, its component model combined with the use of the Virtual DOM allows handling dynamic states effectively. Communication with a blockchain, which operates asynchronously, requires constantly changing the application's state in a timely manner (e.g., receiving IPFS data, waiting for Metamask signature approvals, and listening to mined transactions). React frontend handles all the states, which guarantees real-time notifications of important events (transaction hashes, loading icons, etc.).

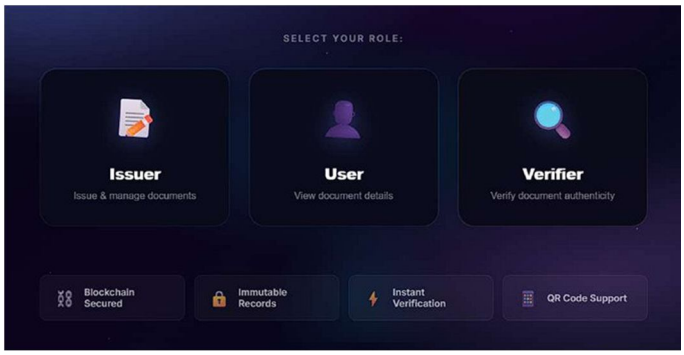


Fig. 2. Role-Based Access Control (RBAC) interface that routes users to appropriate React components based on their cryptographic clearance.

As opposed to the classic web application backend stor- ing user passwords, this layer directly interacts with the user through the Metamask extension, a browser-based Web3 cryptographic wallet. By connecting to Metamask, one can access the Ethereum Provider API injected to the browser global namespace as window.ethereum following the stan- dard EIP-1193. When trying to send a transaction, such as uploading a document, the frontend constructs a transac- tion payload and invokes a cryptographic signature operation via MetaMask. With this architecture, one ensures a "Zero- Knowledge" security posture in which the user's private key is kept encrypted in the browser extension and never passed to Node.js backend server.

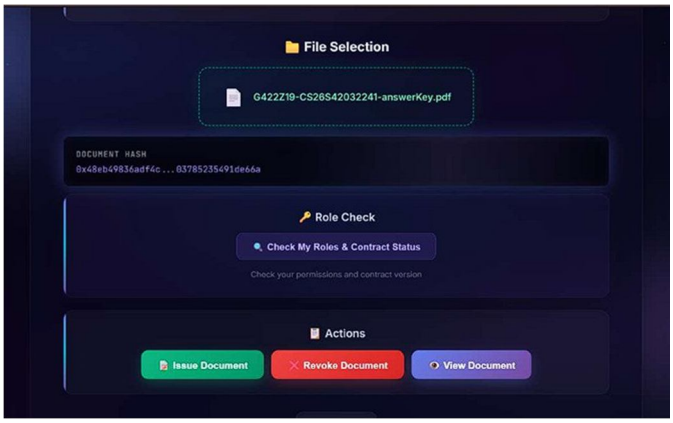


Fig. 3. Issuer Dashboard at the moment of choosing the document, displaying SHA-256 hash of the file.

To provide a better experience for different types of users, the frontend architecture consists of highly-specialized dash- boards controlled by the Role-Based Access Control (RBAC):

1) The Issuer Dashboard: a special dashboard designed to allow university officials to securely upload physical PDF documents via React frontend components. It pro- vides the frontend with an opportunity to compute the SHA-256 hash of the file that is then shown to the user before sending the payload to the backend;

2) The User Dashboard: this dashboard gives students read-only access to their educational records. The fron- tend executes view functions that return the relevant information about the IPFS CIDs and issuance times- tamps;

3) The Verifier Dashboard: for maximum convenience, corporate employees have access to this dashboard to instantly verify a document. The React components here feature a built-in QR-code scanner. After scanning the QR code, the frontend decodes the transaction metadata, queries the Ethereum blockchain using an accessible RPC node, and receives a binary "valid"/"invalid" cryp- tographic proof.

B. Orchestration and API Layer (Backend)

Although it is possible to make a DApp perform all the operations between the frontend and the blockchain, it would lead to severe security issues due to exposing the decentralized storage API keys in addition to making the client do too much data-formatting. For that reason, this project architecture uses an optimal Orchestration and API Layer built on top of Node.js.

Node.js was selected based on its asynchronous and non- blocking nature, as performing operations with decentralized systems— like making an IPFS node pin a file, for example— requires significant time, thus resulting in long-running re- quests. Using Node.js for such purposes will make use of the asynchronous nature of the platform and allow running such operations in parallel while maintaining a very efficient system with low overhead. This architectural decision makes it possible to serve as many as 400 concurrent connections at once without having any crashes or API timeouts.

The Orchestration Layer acts as a central component of the platform, taking care of three main operational components:

1) Secure Data Ingestion and Payload Formatting: After the Issuer uploads an academic certificate, the Node.js backend acts as the first point of contact. It receives the data uploaded through a multipart form, checks the file consistency, and formats the metadata payload. Before interacting with the decentralized networks, the backend guarantees that the payload is properly prepared to minimize future on-chain transaction costs.

2) IPFS Gateway Protocol (Decentralized Storage Bridging): For securely interfacing with the InterPlan- etary File System (IPFS) without exposing the IPFS Administrative API Keys in the client browser, a secure backend proxy was used. With this approach, the Node.js server creates dedicated IPFS clients and uses them for uploading the physical document to the network. After ensuring that the file is safely pinned within the decen- tralized network, the backend receives the immutable Content Identifier (CID) which will be later securely passed to the frontend and included within the smart contract transaction.

3) Web3 RPC Bridging via Ethers.js: Connecting to the Ethereum blockchain requires a reliable backend solution, thus Ethers.js package has been used. Ethers.js is preferred over Web3.js because of being lighter- weight, well-documented, and better at handling Appli- cation Binary Interfaces (ABIs). To interact with the Ethereum blockchain, the Node.js backend uses the JsonRpcProvider to connect to the Ethereum JSON-RPC nodes. While the frontend (MetaMask extension) writes transactions

(thus hiding the private key of the user), the backend uses Ethers.js extensively to read informa- tion from the blockchain. Upon receiving a verification request from the Employer, the backend quickly checks the smart contract state, comparing the provided Content Identifier against the Document Hash and sending it back to the frontend within 2 seconds.

C. Decentralized Storage Layer (IPFS)

However, one of the key architectural limitations of any Ethereum mainnet, as well as other public blockchains, is the enormous cost of storing data in on-chain smart contracts. Blockchain technology is primarily intended for recording transactions related to state changes in a decentralized manner. Storing even a regular PDF certificate, that takes a few hundreds kilobytes of memory, in on-chain storage will cost thousands of dollars of Gas (network transaction fee). Thus, storing a huge number of educational certificates issued yearly by the university becomes completely infeasible.

In order to overcome this notorious problem of "Storage Bottleneck", the suggested architecture incorporates an on- chain/off-chain data structure based on InterPlanetary File System (IPFS). IPFS is a P2P hypermedia protocol that was introduced as a means to move away from conventional web (HTTP) location-based addressing and use **content-based addressing** that is highly secure.

Once an administrator uploads a certificate of a student via the React frontend to a designated university Node.js backend, the IPFS node handles the uploading of a document according to the following cryptography process:

1) Merkle DAG Building: A document is divided into pieces (most often around 256 Kilobytes each). Such pieces are then constructed in Merkle DAG.
2) Generation of CID: CID or Content Identifier, which is usually encoded into Base58 and starts with "Qm..." is generated for such Merkle DAG (basically its root hash).
3) The Avalanche Effect: Since CID is determined based on the content of a document, any alteration in such file, be it a grade, date, or single pixel will yield an entirely different root hash and hence a different CID. Such property guarantees absolute security.

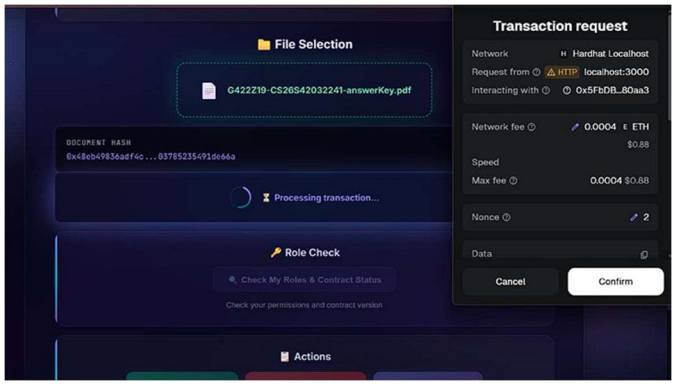


Fig. 4. User Dashboard retrieving the immutable Content Identifier (CID) of an IPFS-hosted file stored in the blockchain record.

As can be seen from Fig. 5, instead of uploading the actual certificate as part of on-chain state storage, the Ethereum smart contract only saves its lightweight CID. The reduction in size goes from several Megabytes of storage to exactly 46 Bytes. It is exactly the reason why such solution achieves such a low transaction cost ($0.49 per issuance).

It is worth mentioning that IPFS nodes regularly conduct garbage collection in order to release some extra storage for new files. In order to counteract this phenomenon, the Node.js backend incorporates a "Pinning" mechanism. Once a file gets uploaded by the administrator, a corresponding message to the IPFS node instructing it to pin the CID is sent. That way, the university certificate is permanently stored on IPFS and cannot get lost due to routine garbage collection or other common server issues.

Whenever an employer attempts to check the validity of the document, they only need to retrieve such CID and download the original academic certificate.

D. Decentralized Storage Layer (IPFS)

However, one of the key architectural limitations of any Ethereum mainnet, as well as other public blockchains, is the enormous cost of storing data in on-chain smart contracts. Blockchain technology is primarily intended for recording transactions related to state changes in a decentralized manner.

Storing even a regular PDF certificate, that takes a few hundreds kilobytes of memory, in on-chain storage will cost thousands of dollars of Gas (network transaction fee). Thus, storing a huge number of educational certificates issued yearly by the university becomes completely infeasible.

In order to overcome this notorious problem of "Storage Bottleneck", the suggested architecture incorporates an on- chain/off-chain data structure based on InterPlanetary File System (IPFS). IPFS is a P2P hypermedia protocol that was introduced as a means to move away from conventional web (HTTP) location-based addressing and use **content-based addressing** that is highly secure.

Once an administrator uploads a certificate of a student via the React frontend to a designated university Node.js backend, the IPFS node handles the uploading of a document according to the following cryptography process:

1) Merkle DAG Building: A document is divided into pieces (most often around 256 Kilobytes each). Such pieces are then constructed in Merkle DAG.

2) Generation of CID: CID or Content Identifier, which is usually encoded into Base58 and starts with "Qm..." is generated for such Merkle DAG (basically its root hash).

3) The Avalanche Effect: Since CID is determined based on the content of a document, any alteration in such file, be it a grade, date, or single pixel will yield an entirely different root hash and hence a different CID. Such property guarantees absolute security.

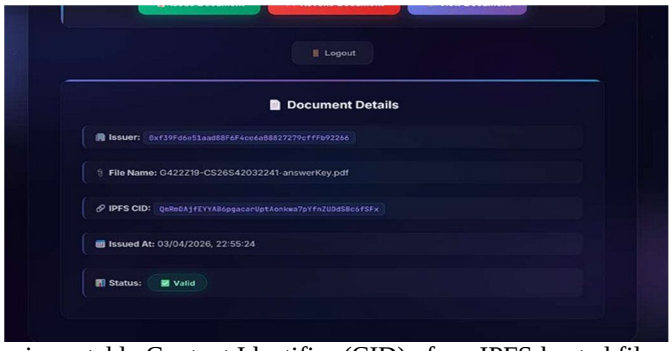


Fig. 5. User Dashboard retrieving the immutable Content Identifier (CID) of an IPFS-hosted file stored in the blockchain record.

As can be seen from Fig. 5, instead of uploading the actual certificate as part of on-chain state storage, the Ethereum smart contract only saves its lightweight CID. The reduction in size goes from several Megabytes of storage to exactly 46 Bytes. It is exactly the reason why such solution achieves such a low transaction cost ($0.49 per issuance).

It is worth mentioning that IPFS nodes regularly conduct garbage collection in order to release some extra storage for new files. In order to counteract this phenomenon, the Node.js backend incorporates a "Pinning" mechanism. Once a file gets uploaded by the administrator, a corresponding message to the IPFS node instructing it to pin the CID is sent. That way, the university certificate is permanently stored on IPFS and cannot get lost due to routine garbage collection or other common server issues.

Whenever an employer attempts to check the validity of the document, they only need to retrieve such CID and download the original academic certificate.

III. IMPLEMENTATION METHODOLOGY

This section describes the actual algorithms involved in exe- cuting our decentralized application, alongside a chronology of the process. The methodology includes two major processing pipelines: Certificate Issuance Protocol and Zero-Knowledge Verification Protocol.

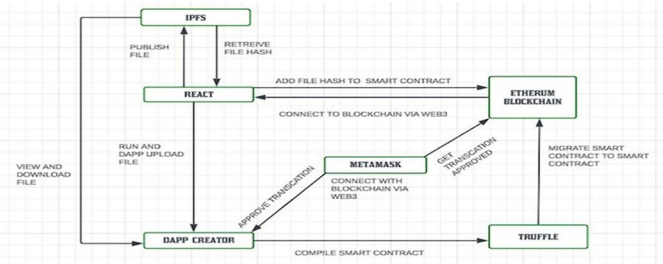


Fig. 6. Diagram illustrating the entire algorithmic workflow involving the DApp, IPFS, and Ethereum smart contract.

A. *Certificate Issuance and IPFS Pinning Protocol*

The immutability of the public ledger is guaranteed through an algorithmically rigorous process that seeks to conserve gas expenses while ensuring data integrity:

1) Ingest Document and Localize Hash: The algorithm begins when the issuer (an authorized administrator from the issuing university) gains access to the React fron- tend. They upload the physical academic certificate and enter the associated metadata. Afterward, the backend processes the binary stream of the file through a SHA- 256 encryption hashing function to create a localized hash of the document.
2) Decentralized Storage and CID Creation: To circum- vent the prohibitive expense of storing data directly on the blockchain, the backend interacts with the Inter- Planetary File System (IPFS) API. The physical PDF document is sent to an IPFS node, where it is formatted mathematically into a Merkle DAG and assigned a unique Content Identifier (CID). The backend ”pins” this CID to guarantee its off-chain availability.
3) MetaMask Cryptographic Signature: From here, the backend sends the generated hash and IPFS CID back to the frontend. The frontend formats the transaction parameters and invokes the injected MetaMask provider in order to obtain a cryptographic signature from the Issuer without sending their credentials to the server side.
4) Transaction Execution and Confirmation: Following the successful signing, the transaction is sent across the Ethereum network and executed by the smart con- tract's issueCertificate function. It maps the document's bytes32 hash to a struct containing the IPFS CID and the timestamp at which the block was mined.

B. *Zero-Knowledge Verification Protocol and QR Code Generation*

The verification protocol is entirely "trustless" in the sense that the potential employer does not have to communicate with the institution in question in order to verify the legitimacy of the document.

1) *Step 1: Automated QR Code Creation*

Following the confirmation of the transaction, the React frontend dynamically creates a Quick Response (QR) code that embeds the verification URL appended with the unique bytes32 document hash and IPFS CID.

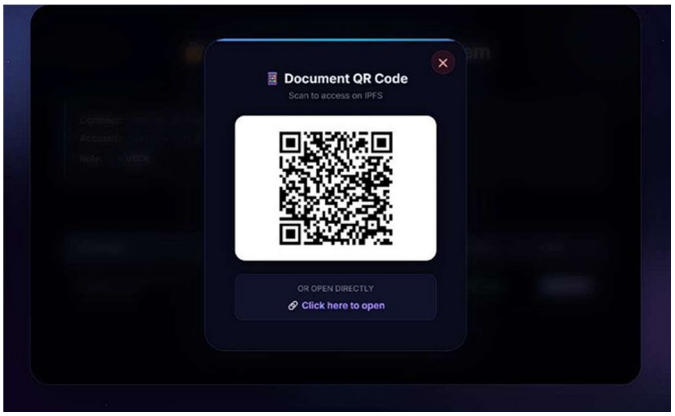


Fig. 7. Automated QR Code creation that links directly to the IPFS CID and the transaction hash stored in the blockchain for instant validation on any mobile device.

2) *Step 2: Decentralized Query Processing*

When receiving a digital copy of the student's certificate, the employer uses any standard mobile scanning app to scan the embedded QR code or enters the document's hash into the Verifier Dashboard. The frontend establishes a JSON RPC connection to the public Ethereum node using Ethers.js and calls the verifyCertificate view function of the smart contract.

3) *Step 3: Document Hash Comparison and Validation*

Since view functions do not change the blockchain's state, there is no gas fee consumed in the query process. The smart contract returns the CID and Issuer wallet address associated with the given bytes32 transaction hash. At the same time, the frontend queries the IPFS node with the returned CID. The document is fetched, and a hash of it is computed. This computed hash is then compared with the immutable one stored on the blockchain. If there is a match, a "Valid" cryptographic proof will be provided on-screen. Otherwise, the document is deemed counterfeit.

IV. RESULTS AND DISCUSSION

To rigorously evaluate the efficiency, scalability, and eco- nomic feasibility of the proposed decentralized verifica- tion system, experiments were conducted in a live de- ployment environment. The testing included end-to-end is- suance, IPFS storage, and verification of 300 academic cer- tificates under varying network conditions. :contentRefer- ence[oaicite:0]index=0

A. Mathematical and Economic Cost Model

One of the major challenges in blockchain adoption is the high and variable transaction cost (gas fees). To analyze this, a mathematical cost model based on Ethereum gas pricing is used.
The total transaction cost in USD is given by:

The QR code is automatically overlaid onto the student's certificate.

Where:

$$TC = \frac{EP \times TGC \times GP}{10^9} \quad (1)$$

- TC = Total Cost in USD
- EP = Ether Price (USD)
- TGC = Total Gas Consumed
- GP = Gas Price in Gwei

In Ethereum, the SSTORE operation is the most expensive, costing 20,000 gas per storage slot. Storing large files such as PDFs directly on-chain would exceed limits and result in extremely high costs.
To overcome this, the system stores only the SHA-256 hash and IPFS CID on-chain, while keeping the actual document off-chain. This optimization significantly reduces gas con- sumption, resulting in an average cost of $0.49 per certificate. For institutions issuing 10,000 certificates annually, the total cost is approximately $4,900, which is significantly lower than maintaining centralized infrastructure and manual verification systems.

B. Scalability and Concurrency Analysis

To evaluate system performance, stress testing was con- ducted on the backend under high load conditions. The system was tested with up to 400 concurrent users, including both issuers and verifiers.

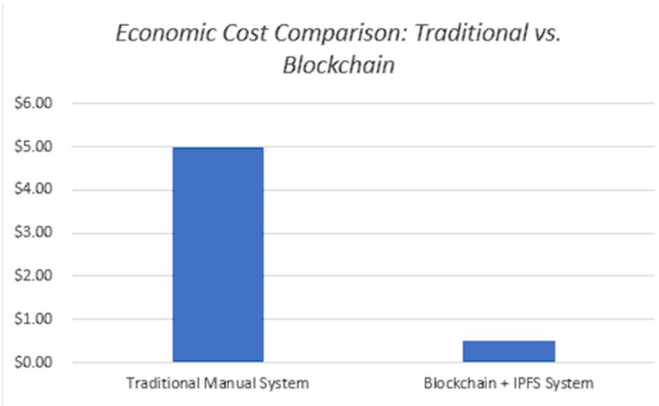


Fig. 8. Cost efficiency of the hybrid IPFS-blockchain architecture.

The system maintained stable performance without crashes, timeouts, or memory issues. This is mainly due to the asyn- chronous, non-blocking I/O model of Node.js, which effi- ciently handles multiple requests simultaneously.

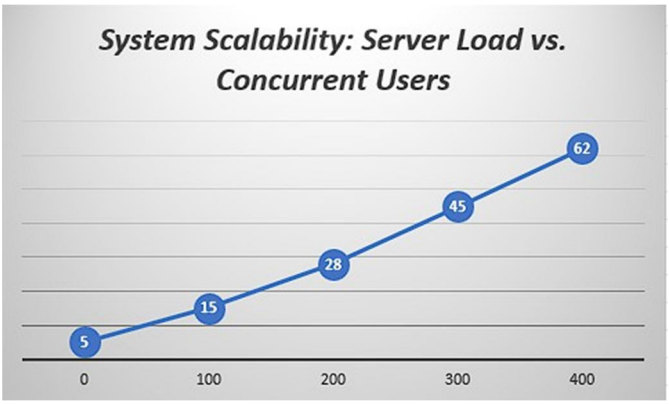


Fig. 9. System performance under increasing concurrent users.

C. Latency and Transaction Speed

Blockchain systems introduce latency due to decentralized consensus. The system was evaluated based on two types of latency:

- Issuance Latency: The time required to confirm a trans- action ranged between 10–20 seconds, aligning with Ethereum's block time of approximately 12 seconds.
- Verification Latency: Since verification involves only reading data, it is significantly faster. Certificate valida- tion, including IPFS retrieval and hash comparison, was completed in under 2 seconds.

D. Security Analysis

The system was tested against common security threats:

- DDoS Resistance: Files are stored on IPFS, eliminating reliance on a central server.
- Tamper Detection: Any modification in a certificate changes its hash, enabling instant detection of forgery.
- Sybil Attack Prevention: Role-Based Access Control (RBAC) ensures that only authorized issuers can create certificates, preventing unauthorized access.

V. CONCLUSION

This paper has comprehensively shown the practical, economical, and operational viability of decentralized certifica- tion verification systems. By carefully designing a hybrid architecture that leverages the strengths of traditional websites and decentralized cryptography, the proposed framework com- pletely eliminates single points of failure caused by centralized database management systems and cumbersome manual vali- dation processes.

Firstly, the proposed framework addresses the scalability bottleneck of blockchain technology by using IPFS as stor- age for all heavy document files and limiting state changes on the Ethereum blockchain to only light weight 32-byte cryptographic hashes, which results in remarkable micro- transactions. The empirical performance tests have success- fully shown the platform's enterprise-level capabilities: it effi- ciently processes loads of up to 400 concurrent users without any intermediary software degradation, offers extremely fast issuance times of up to 10 to 20 seconds, and achieves an optimally optimized transaction cost of merely $0.49 per certificate.

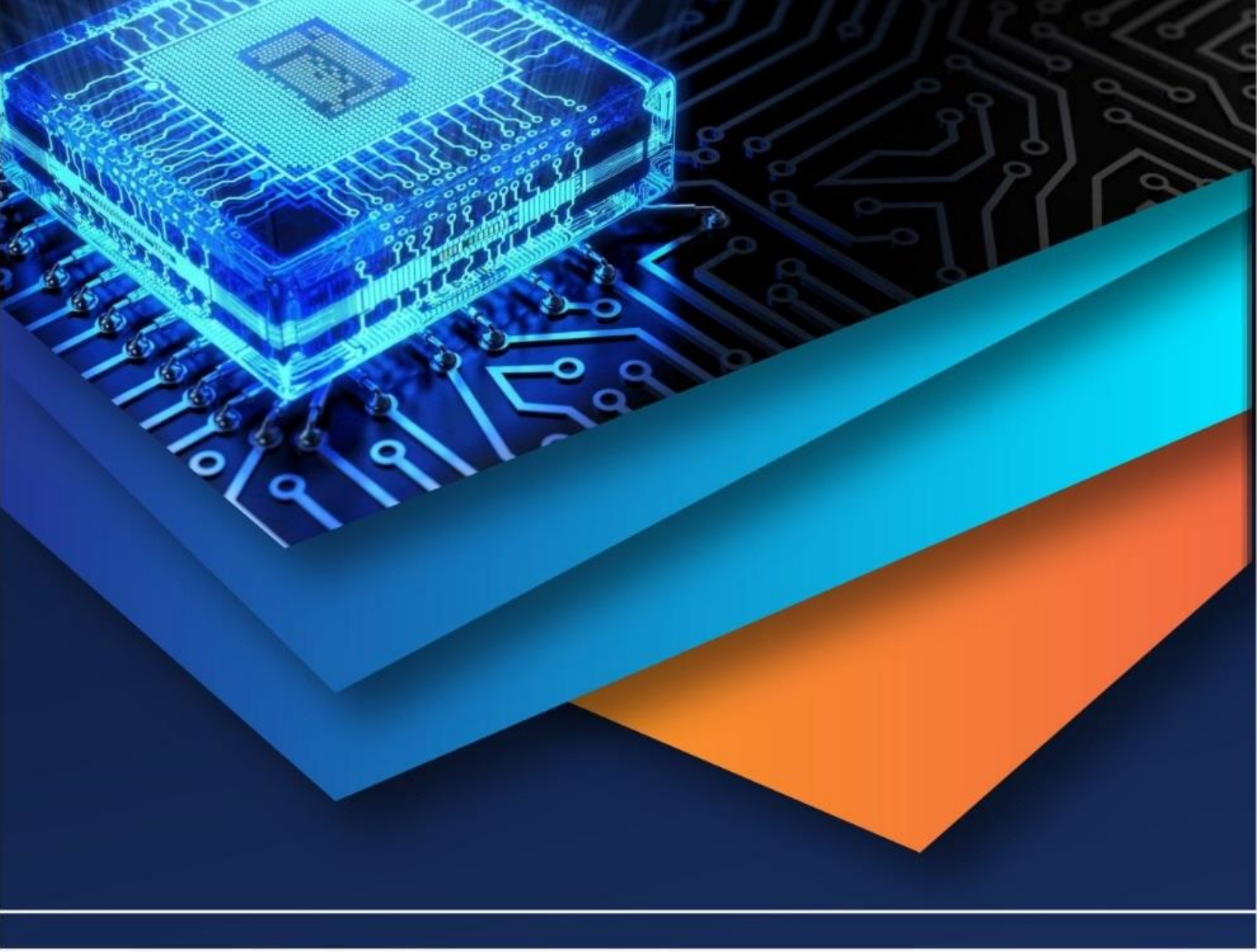
Additionally, the use of strict Role-Based Access Control (RBAC) via Solidity modifiers as well as automated QR- code validation makes the platform extremely secure while remaining intuitive and easy to use at the same time for Issuers, Students, and corporate Verifiers. By leveraging the immutable nature of the SHA-256 hash algorithm and the blockchain network consensus, this framework restores absolute, zero- knowledge trust back to the global academic credentials veri- fication process.

REFERENCES

[1] M. M. Rahman, M. T. K. Tonmoy, S. R. Shihab, and R. Farhana, "Blockchain-Based Certificate Authentication System with Enabling Correction," Journal of Computer and Communications, vol. 11, pp. 73–82, Mar. 2023.

[2] S. Gangwar and A. Chaurasia, "Blockchain-based Authentication and Verification System for Academic Certificate using QR Code and Decen- tralized Applications," International Journal of Computer Applications (IJCA), vol. 186, no. 26, pp. 1–5, Jun. 2024.

[3] N. Vikhankar, A. Andhare, and I. Barne, "E-Certificate Verification Using Blockchain," International Journal of Engineering Research & Technology (IJERT), vol. 13, no. 5, pp. 188–193, May 2024.
[4] A. Mishra, S. Mehta, B. Oza, S. Kumar, and H. Kasturiwale, "Blockchain-Based Decentralized Document Verification and Its Appli- cations," Journal of Information Systems Engineering and Management (JISEM), vol. 10, no. 1, pp. 1–9, 2025.
[5] R. Priyadarshini, R. Pandey, A. K C, D. Bhandari, B. Khadka, R. K. Barik, and M. J. Saikia, "A Faster, Integrated and Trusted Certificate Authentication and Issuer Validation System based on Blockchain," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3539180.
[6] N. Jagadale, P. Somwanshi, P. Thulkar, R. Deshmukh, and S. Nimbalkar, "Blockchain-Based Academic Certificate Verification System," Project Presentation, VPKBIET, Baramati, India, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY