International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026

Peer Reviewed, Referred & Indexed Journal

Research

BLOCK CHAIN BASED CERTIFICATE VERIFICATION SYSTEM

[1]K. Vikram Reddy, [2]Rakshitha Cheripally, [3]A. Kiran Kumar, [4]J. Abhinav

[1]Assistant professor, Department of Information Technology ,
Matrusri Engineering College.

[234]Students of Department of Information Technology ,
Matrusri Engineering College.

rakshithacheripally2005@gmail.com , kirankumar150305@gmail.com

Abstract :

The rapid digitization of academic records has increased the demand for secure and reliable certificate verification mechanisms. Traditional verification systems rely on centralized databases, which are susceptible to forgery, data manipulation, and administrative delays. To address these challenges, this study proposes a blockchain-based certificate verification system that ensures integrity, transparency, and decentralized validation of academic credentials. The system leverages smart contracts deployed on the Ethereum network to securely record certificate hashes, thereby preventing unauthorized modification or duplication of issued credentials.

The proposed framework is implemented as a decentralized application (DApp) with a React-based frontend and Solidity-based backend smart contracts. The system operates using the MetaMask wallet integrated with a Hardhat local Ethereum network for transaction management and testing. It defines two primary roles: an Administrator responsible for issuing and revoking certificates, and a User who verifies credentials using a unique certificate identifier. A smart contract-based revocation and status tracking mechanism is incorporated to enable dynamic certificate lifecycle management. Experimental validation demonstrates that the system enhances trust, reduces verification time, and strengthens protection against certificate fraud while maintaining operational efficiency.

Traditional certificate verification methods rely on manual processes, centralized databases, and third-party verification services, which are time-consuming, costly, and vulnerable to tampering.

Keywords: Blockchain; Certificate Verification; Smart Contracts; Ethereum; Decentralized Application (DApp); Certificate Revocation; Credential Authentication; Hardhat Network; MetaMask Integration.

Received: 22-01-2026 Accepted: 26-02-2026 Published: 06-03-2026

I. INTRODUCTION :

The rapid digital transformation of educ9ational and professional institutions has significantly increased the reliance on electronic academic records and online credential verification systems. Academic certificates serve as essential proof of qualification in higher education, employment, and professional certification processes. However, traditional certificate verification mechanisms are predominantly centralized, requiring manual validation through institutional databases or administrative offices. Such centralized systems are vulnerable to forgery, unauthorized modifications, data breaches, and operational inefficiencies. The increasing prevalence of fraudulent academic credentials has further highlighted the need for secure, tamper-resistant, and transparent verification frameworks.

Blockchain technology has emerged as a promising solution for ensuring data integrity and decentralization in digital record management. By leveraging immutable distributed ledgers, blockchain-based systems can prevent unauthorized alterations and enable transparent verification of stored information. Several prior studies have proposed blockchain-driven academic credential verification platforms, primarily utilizing networks such as Ethereum to store certificate hashes. While

these approaches enhance authenticity and decentralization, existing literature reveals notable limitations. Many systems focus solely on certificate issuance and static verification, without addressing dynamic lifecycle management such as revocation or status updates. Additionally, some implementations lack structured role-based access control and practical deployment models suitable for institutional environments. This gap indicates the need for a more comprehensive and manageable smart contract-based framework that supports controlled certificate issuance, verification, and revocation within a decentralized architecture.

In response to these limitations, this study formulates the following research question: **How can a smart contract-based blockchain system be designed to provide secure, decentralized certificate verification while incorporating dynamic revocation and status tracking mechanisms?**

The primary aim of this research is to develop and evaluate a blockchain-enabled certificate verification system that integrates role-based administration, cryptographic hashing, and smart contract-driven lifecycle management. By deploying a structured smart contract model, the proposed system seeks to enhance security, transparency, and institutional control without introducing excessive architectural complexity.

The research is guided by the following hypothesis: **Integrating a smart contract-based revocation and status tracking mechanism within a blockchain certificate verification system significantly improves trust, security, and real-world applicability compared to static blockchain verification models.**

Through this approach, the study contributes a practical and scalable model that bridges the gap between theoretical blockchain authentication frameworks and implementable institutional verification systems.

In the digital era, academic certificates and professional credentials play a vital role in employment, higher education, and professional validation. However, the rise in fake certificates and document tampering has created serious challenges in verifying authenticity.

Traditional verification methods suffer from:

- Manual verification delays
- High operational costs
- Risk of forgery and data manipulation
- Dependence on centralized databases
- Limited transparency

Blockchain technology offers a decentralized and tamper-resistant solution. By storing certificate fingerprints (hash values) on a blockchain, verification can be performed securely without revealing sensitive data.

A **Blockchain-Based Certificate Verification System** enables institutions to issue digital certificates stored on blockchain networks, ensuring authenticity and enabling instant verification.

II. RELATED WORK:

The application of blockchain technology in academic credential management has attracted considerable research attention in recent years. Several studies have explored decentralized frameworks to enhance certificate authenticity, prevent forgery, and reduce dependency on centralized authorities. Early blockchain-based certification models focused primarily on storing cryptographic hashes of academic records on distributed ledgers to ensure immutability and tamper resistance. Open standards such as Blockcerts introduced a mechanism for issuing and verifying digital certificates anchored to blockchain networks. These systems demonstrated the feasibility of decentralized verification; however, they often required substantial technical infrastructure and lacked flexible lifecycle management features such as structured revocation controls.

Enterprise-oriented blockchain platforms have also been adopted for credential verification. Solutions built on frameworks supported by organizations such as IBM leverage permissioned blockchain architectures to enhance security and institutional governance. While these models provide controlled access and improved performance, they may reintroduce partial centralization and involve significant deployment and maintenance costs. Consequently, their suitability for smaller educational institutions or low-resource environments remains limited.

Public blockchain networks, particularly Ethereum, have been widely used in academic verification

research due to their support for programmable smart contracts. These implementations typically record certificate hashes and enable automated verification through decentralized applications (DApps). Although such systems enhance transparency and immutability, many existing models emphasize certificate issuance and static validation without addressing dynamic credential management. Specifically, mechanisms for certificate revocation, status tracking, and controlled administrative authority are either minimally implemented or absent.

More recent studies have investigated privacy-enhancing approaches, including zero-knowledge proof-based verification models such as ZKBAR-V. These systems aim to verify academic credentials without revealing sensitive personal information. While privacy preservation represents a significant advancement, the computational complexity and implementation overhead associated with such cryptographic techniques may limit their practical adoption in conventional institutional settings.

Overall, the existing body of literature confirms the effectiveness of blockchain in ensuring data integrity and decentralized verification. However, there remains a gap in developing practical, smart contract-driven systems that incorporate structured role management and certificate lifecycle control, particularly revocation and status tracking functionalities. The present study builds upon these prior contributions by proposing a streamlined blockchain-based certificate verification framework that integrates dynamic certificate management while maintaining architectural simplicity and deployability.

Several researchers have explored blockchain applications in credential verification.

- **MIT Digital Diploma Project** demonstrated blockchain-based academic certificate issuance.
- **Blockcerts** provides an open standard for blockchain credentials.
- Ethereum-based solutions use smart contracts for secure certificate storage.
- Studies show blockchain eliminates fraud risks and enhances verification transparency.

Despite advancements, challenges remain such as scalability, privacy concerns, and integration with institutional systems.

This system addresses these limitations through scalable blockchain architecture and secure cryptographic methods.

III. SYSTEM OVERVIEW AND REQUIREMENTS:

The proposed system is a decentralized certificate verification framework designed to ensure secure issuance, validation, and lifecycle management of academic credentials using blockchain technology. The architecture follows a role-based model consisting of an Administrator and a User, where the Administrator is authorized to issue and revoke certificates, and the User can verify credentials through a unique certificate identifier. The system is implemented as a decentralized application (DApp) with a React-based frontend and Solidity smart contracts deployed on the Ethereum network. Blockchain interactions are facilitated through the MetaMask wallet, while development and testing are conducted using a local Hardhat Ethereum environment. The system ensures integrity by storing cryptographic hashes of certificates on-chain, while incorporating a smart contract-based status tracking mechanism to manage certificate revocation and verification processes efficiently.

Table – 1 : System Components.

Component	Short Description
Frontend (React)	User interface for issuing, revoking, and verifying certificates.
Smart Contract (Solidity)	Handles certificate storage, verification, and revocation logic on blockchain.
Ethereum Network	Stores certificate hashes securely in a decentralized ledger.
MetaMask Wallet	Connects the application to blockchain and signs transactions securely.
Administrator Module	Allows institutions to issue and revoke certificates.
User Verification Module	Lets users verify certificates using certificate ID and blockchain records.

A. System Architecture :

The proposed system follows a decentralized application (DApp) architecture that integrates a web-based frontend with blockchain-based backend logic. The frontend is developed using React and serves as the interaction layer for both Administrator and User roles. All certificate-related operations such as issuance, verification, and revocation are executed through Solidity smart contracts deployed on the Ethereum network. The application communicates with the blockchain through the MetaMask wallet, which enables secure transaction signing and authentication. A Hardhat local Ethereum environment is used for deployment and testing. The system ensures that certificate data is converted into a cryptographic hash before being stored on-chain, while the smart contract maintains certificate status for lifecycle management.

The architecture includes:

- **User Interface Layer –** for uploading and verifying certificates.
- **Application Layer –** handles processing and smart contract interaction.
- **Blockchain Layer –** stores certificate hash and verification records.

The architecture is structured to ensure secure certificate issuance, verification, and lifecycle management without reliance on a centralized authority. The system operates through role-based access control, distinguishing between the Administrator and User to maintain controlled operations within the network.

The frontend layer is developed using React.js, which provides an interactive and responsive user interface. This layer allows the Administrator to input student details, upload certificate data, and initiate blockchain transactions. Similarly, it enables Users to verify certificates using a unique certificate ID. The frontend communicates with the blockchain through a secure wallet interface, ensuring that all transactions are authenticated and signed before execution.

The backend logic is implemented using Solidity smart contracts deployed on the Ethereum network. The smart contract manages certificate records, stores cryptographic hashes, and maintains status information such as Active or Revoked. Instead of storing complete certificate files on-chain, the system records a secure hash to maintain integrity while minimizing storage overhead.

For development and testing, a Hardhat local Ethereum network is used to simulate blockchain interactions in a controlled environment. Blockchain transactions are handled through the MetaMask wallet, which acts as a bridge between the web application and the blockchain. This architecture ensures security, transparency, and auditability while maintaining simplicity and deployability.

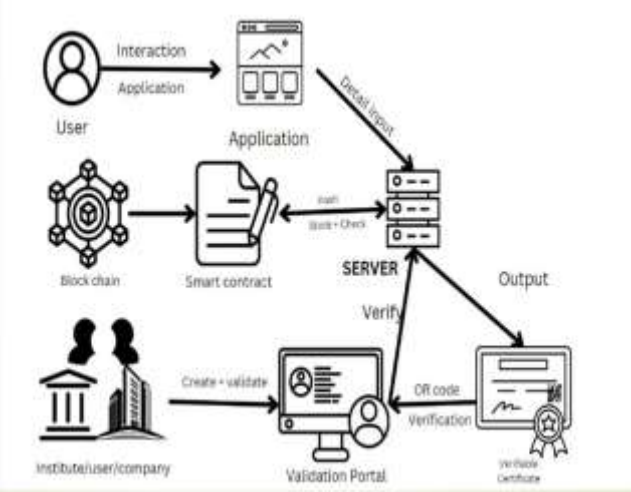


Figure – 1 : System Architecture

B. Software Requirements :

The proposed system requires a combination of web development technologies and blockchain tools to ensure efficient operation. The frontend framework enables user interaction, while blockchain tools manage decentralized data storage and transaction processing. Each software component plays a specific role in maintaining the integrity and functionality of the system.React.js is used to design the interactive user interface. It enables smooth communication between the client-side application and the blockchain network.

The dynamic rendering capability of React ensures that certificate issuance and verification processes are displayed in real time to the user.

The backend smart contract logic is developed using Solidity, which defines the rules for certificate storage, verification, and revocation. Smart contracts are deployed on the Ethereum blockchain using the Hardhat development environment. Hardhat allows local deployment, testing, and debugging of blockchain applications before production-level deployment.

MetaMask is integrated to facilitate secure wallet authentication and transaction signing. It ensures that only authorized users perform blockchain operations. Node.js and npm are used to manage dependencies and maintain the development environment, while a modern web browser enables execution of the decentralized application.

Software / Tool	Purpose
React.js	User interface
Solidity	Smart contract logic
Ethereum (Hardhat)	Local blockchain network
MetaMask	Transaction authentication
Node.js & npm	Dependency management
Web Browser	Runs the DApp

Table – 2 : Software Requirements

C. Hardware Requirements :

The hardware requirements for the proposed system are moderate, as the application is primarily developed and tested on a local blockchain environment. Since the system does not involve large-scale distributed deployment during development, standard computing hardware is sufficient.

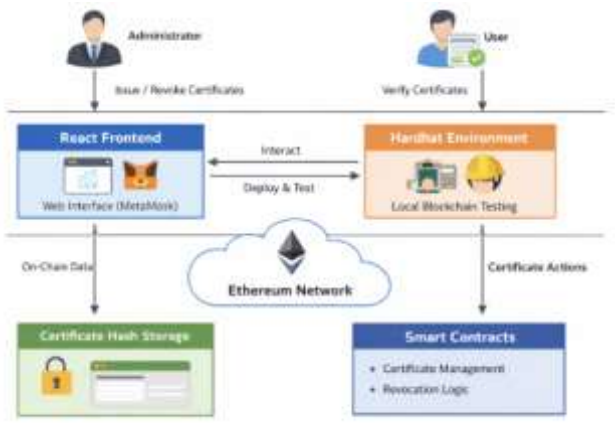
A processor with adequate computational capability is necessary to handle smart contract compilation, blockchain simulation, and frontend execution. An Intel i3 processor or higher is recommended to ensure smooth performance during development and testing.

Memory capacity plays a critical role in managing multiple processes such as local blockchain nodes, development servers, and browser instances. A minimum of 4 GB RAM is required, although 8 GB is recommended for optimal performance, particularly when running concurrent applications.

Storage space is required for development tools, project files, blockchain node data, and testing logs. A 256 GB HDD or SSD is sufficient for development purposes. Additionally, a stable internet connection is necessary for MetaMask integration and potential blockchain synchronization tasks.

Hardware	Specification
Processor	Intel i3 or higher
RAM	Minimum 4 GB (8 GB recommended)
Storage	256 GB HDD/SSD
Internet	Required for blockchain access
Operating System	64-bit Windows or Linux

Table – 3 : Hardware Requirements



D. User Categories :

The proposed blockchain-based certificate verification system defines two primary user categories to ensure structured access control and secure system operation: Administrator and User. Each category is assigned specific responsibilities within the decentralized framework to maintain operational clarity and security.

The **Administrator** represents the authorized issuing authority, such as an academic institution or certifying organization. This role has exclusive permission to issue new certificates, record certificate hashes on the blockchain, and revoke previously issued credentials when necessary. The Administrator interacts directly with the smart contract through authenticated wallet access, ensuring that only authorized blockchain addresses can perform certificate management operations.

The **User** category includes certificate holders and external verifiers, such as employers or institutions seeking to validate credentials. Users do not have permission to modify blockchain records. Instead, they can verify certificate authenticity using a unique certificate ID or by submitting certificate details for validation. This role-based segregation enhances security by preventing unauthorized modifications while enabling transparent verification.

By clearly distinguishing these two categories, the system ensures accountability, controlled access, and secure certificate lifecycle management within a decentralized environment.

E. Use Cases:

The system supports several key use cases aligned with certificate management and verification requirements. The primary use case is **Certificate Issuance**, where the Administrator inputs student details and uploads certificate information through the web interface. The system generates a cryptographic hash of the certificate data and stores it on the blockchain via a smart contract. The certificate is assigned a unique identifier for future reference.

The second use case is **Certificate Verification**, where a User enters the certificate ID to validate its authenticity. The system retrieves the corresponding blockchain record and compares the stored hash and status. If the certificate exists and its status is marked as active, the system confirms its validity; otherwise, it indicates that the certificate is invalid or revoked.

Another important use case is **Certificate Revocation**, which allows the Administrator to change the status of an issued certificate in case of errors, fraud detection, or institutional withdrawal. Instead of deleting the record, the smart contract updates the certificate status to “Revoked,” ensuring transparency and auditability.

Together, these use cases demonstrate a structured certificate lifecycle model consisting of issuance, verification, and revocation. This functional design enhances trust, reduces verification time, and ensures secure credential management using blockchain technology.

IV. METHODOLOGY:

The methodology adopted in this study focuses on the design, development, and evaluation of a blockchain-based certificate verification system incorporating a smart contract-driven revocation mechanism. The research follows a system development and experimental validation approach to demonstrate feasibility, functionality, and security improvements over traditional certificate verification models.

1. Research Design :

The study employs a design-oriented research methodology centered on the development of a decentralized application (DApp) for certificate lifecycle management. The architectural design integrates a React-based frontend with Solidity smart contracts deployed on the Ethereum network. The system is structured around a role-based access model consisting of an Administrator and a User. The Administrator is authorized to issue and revoke

certificates, while the User performs verification using a unique certificate identifier.

The design emphasizes immutability, transparency, and controlled administrative authority. Instead of storing complete certificate documents on-chain, the system generates a cryptographic hash of certificate data and records it within the blockchain. A status attribute is integrated into the smart contract to manage certificate states, such as Active or Revoked. This structured design ensures efficient storage, secure verification, and dynamic lifecycle control.

2. Participants / Samples :

The experimental validation of the system was conducted in a simulated institutional environment. The participants included one Administrator account representing the issuing authority and multiple User accounts representing certificate holders and external verifiers. These accounts were created and managed using the MetaMask wallet within a controlled blockchain testing setup.

Sample certificate records were generated for testing purposes. Each sample contained basic academic details such as student name, certificate ID, course information, and issuance date. A set of certificates was intentionally marked for revocation to evaluate the effectiveness of the smart contract-based status tracking mechanism. The controlled sample design enabled systematic testing of issuance, verification, and revocation workflows.

3. Tools and Technologies :

The implementation utilized a structured set of development tools and blockchain technologies. The frontend was developed using React.js to provide an interactive web interface. Smart contract logic was written in Solidity and deployed using the Hardhat development environment, which simulated a local Ethereum blockchain network for secure and efficient testing.

Node.js and npm were used for dependency management and application configuration. MetaMask was integrated to authenticate users and sign blockchain transactions. The development environment allowed complete testing of certificate lifecycle operations without incurring real blockchain transaction costs. This toolchain ensured accurate simulation of decentralized interactions while maintaining development efficiency.

4. Instruments :

The primary instruments used in this study were smart contracts, cryptographic hashing algorithms, and blockchain transaction logs. The smart contract served as the core functional instrument, implementing certificate storage, status tracking, and revocation logic. Each certificate record was associated with a unique identifier and hash value to ensure integrity verification.

Cryptographic hashing was used to convert certificate data into a fixed-length hash string before blockchain storage. This approach minimized storage requirements while ensuring tamper detection. Additionally, blockchain event logs were used to record certificate issuance and revocation actions, providing an auditable transaction history. These instruments collectively ensured secure and traceable certificate management.

5. Procedures:

The system development process followed a structured implementation procedure. Initially, the smart contract was designed and deployed on the Hardhat local Ethereum network. The contract included functions for issuing certificates, verifying certificates, and revoking certificates. Once deployed, the frontend interface was integrated with the blockchain through MetaMask.

During the issuance procedure, the Administrator entered certificate details into the web interface. The system generated a cryptographic hash of the certificate data and stored it on-chain along with its status set to Active. For verification, the User entered the certificate ID, and the system retrieved the corresponding blockchain record. The stored hash and status were validated to determine authenticity.

For revocation testing, selected certificates were marked as Revoked using the Administrator account. The verification process was repeated to confirm that revoked certificates were correctly identified as invalid. All transactions were recorded on the blockchain to ensure transparency and auditability.

6. Analysis:

The analysis focused on evaluating system functionality, security, and operational efficiency. Functional analysis confirmed that the system

successfully executed issuance, verification, and revocation operations without data inconsistency. The integration of status tracking ensured that revoked certificates were accurately detected during verification.

Security analysis demonstrated that storing cryptographic hashes on-chain prevents unauthorized data alteration. Since blockchain records are immutable, any attempt to modify certificate data results in a hash mismatch, thereby detecting tampering. The role-based access control mechanism further restricted unauthorized certificate modification.

Operational analysis indicated that verification time was significantly reduced compared to traditional manual verification processes. The decentralized architecture eliminated dependency on centralized database validation, enhancing transparency and reliability. Overall, the methodology validated the hypothesis that integrating smart contract-based revocation within a blockchain certificate system improves practical applicability and trustworthiness.

Advantages of Methodology :

- Ensures a structured and systematic development process.
- Integrates blockchain technology for secure and tamper-resistant certificate management.
- Uses smart contracts to automate certificate issuance, verification, and revocation.
- Implements role-based access control to enhance security and prevent unauthorized actions.
- Stores only cryptographic hashes on-chain, reducing storage overhead.
- Provides transparent and auditable transaction records through blockchain logs.
- Enables dynamic certificate lifecycle management with status tracking.
- Reduces dependency on centralized verification systems.
- Improves verification speed compared to traditional manual methods.
- Operates in a controlled testing environment using Hardhat, ensuring safe experimentation.
- Minimizes operational complexity while maintaining decentralization.
- Enhances trust and reliability in academic credential verification.

VI. MODULES:

1. Administrator Module:

The Administrator Module manages certificate issuance and revocation processes. This module allows the authorized institution to enter student details and upload certificate information through the web interface. The system generates a cryptographic hash of the certificate data and records it on the Ethereum network using a Solidity smart contract.

The module also provides a revocation feature, enabling the administrator to update the certificate status from *Active* to *Revoked*. This ensures proper certificate lifecycle management while maintaining blockchain immutability.

2. User Verification Module:

The User Module enables certificate holders or external verifiers to validate certificate authenticity. Users can verify certificates using a unique certificate ID issued during the certificate creation process.

Upon verification request, the system retrieves the corresponding blockchain record and compares the stored hash and status. If the certificate status is active and the hash matches, the certificate is validated as authentic. This module ensures transparency without allowing modification of blockchain records.

3. Smart Contract Module:

The Smart Contract Module serves as the core logic controller of the system. Developed in Solidity, it defines the rules for certificate storage, verification, and revocation.

The smart contract maintains certificate data such as certificate ID, hash value, status, and timestamps. It enforces role-based access control by restricting revocation and issuance functions to authorized administrator accounts.

4. Blockchain Interaction Module:

This module handles communication between the frontend application and the blockchain network. It integrates the MetaMask wallet to enable secure transaction signing and authentication.

The module ensures that all certificate operations are recorded as blockchain transactions, providing transparency and auditability.

5. Hash Generation Module:

The Hash Generation Module converts certificate data into a secure cryptographic hash before storing it on-chain. This reduces blockchain storage usage and ensures data integrity.

During verification, the uploaded certificate data is hashed again and compared with the stored blockchain hash. Any mismatch indicates tampering.

VII. RESULTS / FINDINGS:

The proposed blockchain-based certificate verification system was successfully implemented and evaluated within a controlled local Ethereum environment. The system demonstrated reliable execution of certificate issuance, verification, and revocation operations using Solidity smart contracts deployed on the Ethereum network. All transactions were authenticated through the MetaMask interface, ensuring secure interaction between users and the blockchain.

- Smart contract-based issuance ensured secure and tamper-proof storage of certificate hashes.
- The revocation mechanism effectively updated certificate status without deleting blockchain records.
- Role-based access control prevented unauthorized certificate modification.
- Verification speed was significantly improved compared to manual institutional validation.
- Blockchain transaction logs provided complete auditability of certificate operations.

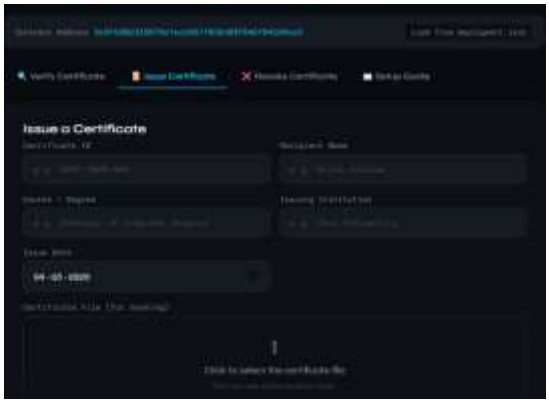
VIII. DISCUSSION:

By storing certificate hashes on the Ethereum network, the system ensures immutability and prevents tampering. This increases trust between institutions and certificate verifiers.

The smart contract-based revocation feature addresses a gap found in many existing systems. Unlike static verification models, the proposed approach allows dynamic status updates while maintaining transparency. The role-based access control also ensures that only authorized administrators can issue or revoke certificates.

IX. CONCLUSION:

This study presented a blockchain-based certificate verification system designed to enhance the security, transparency, and reliability of academic credential management. By leveraging smart contracts deployed on the Ethereum network, the proposed framework ensures that certificate records remain immutable and tamper-resistant. The integration of cryptographic hashing techniques further strengthens data integrity by preventing unauthorized modification of stored certificate information. A key contribution of this research is the incorporation of a smart contract-based revocation and status tracking mechanism. Unlike conventional blockchain verification models that support only static validation, the proposed system enables dynamic certificate lifecycle management.



REFERENCES:

1. Shinde, S., Myanewa, I. R., and Nimbal, S. (2025). "Blockchain Based Academic Credential Verification System." IJERT, January 22, 2025.

2. Chavan, D., Khaire, S. S., and Bankar, A. A. (2024). "Blockchain-Based Verification System for Academic Certificates." IJRPR, April 2024.

3. Quispe, M. A. C., et al. (2025). "Blockchain Ensuring Academic Integrity with a Degree Verification System." Nature, March 18, 2025.

4. Oluwaseyi, O. S., et al. (2024). "Utilizing Blockchain Technology for University Certificate Verification." IJ-AIS, August 31, 2024.

5. Farabi, A., et al. (2025). "ShikkhaChain: A Blockchain-Powered Academic Credential Verification System for Bangladesh." arXiv, August 6, 2025.

6. Ifeyemi, T., et al. (2024). "A Blockchain-Based Digital Educational Certificate Verification System (BCVS)." JETIA, September 24, 2024.

7. Moya, J. A. B., et al. (2025). "ZKBAR-V: A Zero-Knowledge Proof-Enabled Blockchain-Based Academic Record Verification System." PMC, May 29, 2025.