International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Secure-Docs

Dr P Deepthi[1],Maheen Fathima[2],Nuha Qadeer Ahmed[3],Panchagudi Parvathi[4]

[1]Associate Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women, Hyderabad, India.

[2,3,4]B.Tech Students; ; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women, Hyderabad, India.

Mail Id; the.maheen.1133@gmail.com[2],nuhaqadeer@gmail.com[3],panchagudiparvathi@gmail.com[4]

Abstract

The increasing demand for reliable digital credential verification has highlighted the limitations of traditional document management systems, which are often vulnerable to forgery and unauthorized modifications. This paper presents ***Secure-Docs****, a web-based platform that leverages blockchain technology and the InterPlanetary File System (IPFS) to enable secure issuance and verification of academic documents. In the proposed system, educational institutions upload certificates to IPFS, where files are stored in a decentralized manner. A unique cryptographic hash generated for each document is recorded on the blockchain, ensuring immutability and traceability.The integration of blockchain with decentralized storage eliminates dependence on centralized authorities and significantly reduces the risk of document tampering. Through an intuitive web interface, students can manage their credentials securely, while employers and other stakeholders can instantly verify document authenticity without relying on intermediaries. This approach enhances transparency, strengthens digital trust, and improves verification efficiency. Furthermore, the Secure-Docs framework can be extended to support validation of government records, legal agreements, and professional certifications, making it applicable across multiple sectors. The proposed solution demonstrates how decentralized technologies can transform document authentication processes into a secure, transparent, and scalable system.*

Keywords—*Blockchain Technology, InterPlanetary File System (IPFS), Digital Credential Verification, Cryptographic Hashing, Decentralized Storage, Document Authentication, Smart Contracts, Data Integrity, Secure Document Management, Transparency.*

INTRODUCTION

The rapid growth of digital education and online recruitment has increased the need for reliable verification of academic credentials. Traditional methods of validating certificates often involve manual procedures that are time-consuming and vulnerable to manipulation. To address these challenges, this project proposes a secure anti-forgery mechanism for academic documents using blockchain technology, InterPlanetary File System (IPFS), and cryptographic hash functions. By combining decentralized storage with immutable ledger technology, the system ensures that certificates remain authentic and resistant to unauthorized modifications. The proposed platform enhances trust among educational institutions, students, and employers while reducing verification delays and operational costs.

Scope

The scope of this project is to design and develop a secure and transparent platform for issuing and verifying academic documents using blockchain and IPFS technologies. In the proposed framework, certificates are stored in IPFS, while their corresponding hash values are recorded on the blockchain to maintain integrity. The system supports multiple stakeholders, including universities, students, and recruiters, enabling fast and tamper-resistant verification without reliance on third-party authorities. Furthermore, the architecture can be extended to additional applications such as government document validation, legal record authentication, and professional certification management, thereby increasing its usability across various domains.

Existing System

In conventional document verification systems, academic credentials such as mark sheets, certificates, and identity documents are typically issued in paper format or stored in centralized digital databases. Verification is usually performed through manual inspection, email communication, or direct confirmation from issuing institutions. These approaches often require significant time and administrative effort. Additionally, centralized storage increases vulnerability to unauthorized access, while the absence of standardized verification methods limits transparency and reliability.

Proposed System

The proposed system introduces a blockchain-enabled document verification platform integrated with IPFS for decentralized file storage. Academic certificates are uploaded to IPFS, and the generated cryptographic hash is stored on the blockchain network. This ensures that any modification to the document results in a mismatch, immediately

indicating tampering. Educational institutions can issue verified credentials, students can securely access and manage their documents, and employers can validate authenticity instantly. The decentralized nature of the system eliminates dependency on intermediaries and significantly improves trust and efficiency in credential verification.

REQUIREMENT ANALYSIS

Functional Requirements

The Secure-Docs system is designed with three primary user roles: administrator, student, and verifier, each with specific functionalities. The administrator module is responsible for overall system management, including secure login access, uploading student certificate files in PDF format, and generating unique cryptographic hashes for each uploaded document. After hash generation, the administrator issues certificates to the respective student accounts and maintains records by viewing, updating, or deleting certificate details. Additionally, the administrator reviews verifier registration requests, approves authorized users, and monitors all registered students and verifiers before logging out securely.The student module enables users to register and create personal accounts, log in securely, and access their dashboard. Students can view certificates issued by authorized institutions and download them in PDF format for official use. The module also allows students to log out safely after completing their tasks.The verifier module is designed for employers or organizations that need to validate document authenticity. Verifiers first register and request access to the system. Once approved, they can log in to the verification panel, enter the certificate hash value, or scan the QR code associated with the document. The system checks blockchain records and confirms authenticity. Upon successful verification, the verifier can view certificate details and securely exit the system.

Non-Functional Requirements

The system is designed to meet several non-functional requirements to ensure quality and reliability. Performance is maintained by enabling quick certificate upload and verification operations. Scalability is considered so that the system can handle an increasing number of users and documents without performance degradation. Usability is emphasized through a simple and intuitive interface suitable for institutions, students, and employers. Reliability ensures consistent system operation with minimal downtime. Security is achieved using blockchain immutability, authentication mechanisms, and decentralized storage. Compatibility allows the application to function across modern browsers and operating systems. Maintainability supports future updates and improvements, while accessibility ensures that users can access the system anytime through the web-based platform.The development of the Secure-Docs system requires both software and hardware resources. The software environment includes Windows 11 as the operating system and Visual Studio Code as the development environment. JavaScript is used as the primary programming language, with React.js for frontend development and Node.js with Express.js for backend services. Ethereum is utilized as the blockchain platform, and Web3.js is used for blockchain interaction. Smart contracts are developed using Solidity, with Truffle serving as the development framework and Ganache providing a local blockchain network for testing. IPFS is integrated for decentralized file storage, and MetaMask is used as a browser extension for blockchain connectivity and user authentication.

The hardware requirements include a system with at least an Intel Core i5 processor or higher, a minimum of 8 GB RAM, and approximately 50 GB of available hard disk space to ensure smooth development and execution of the application.

Life Cycle Model

The development of the Secure-Docs system follows the Waterfall model, which is a sequential approach where each phase is completed before moving to the next. Initially, during the requirement analysis phase, all functional and non-functional requirements were identified and documented to create the Software Requirements Specification. In the system design phase, the architecture of the application, user interface layout, and module structure were prepared. Data flow diagrams and smart contract designs were also created to represent system functionality.During the implementation phase, smart contracts were developed using Solidity, and blockchain testing was conducted using Truffle and Ganache. IPFS was integrated for decentralized document storage, while the web interface was built using React.js along with HTML, CSS, and JavaScript. MetaMask was integrated for blockchain interaction and authentication. In the integration and testing phase, all modules were combined into a single application, and unit, functional, and performance testing were carried out to verify system behavior.

DESIGN

System design defines the overall structure of an application and explains how different components are organized to perform required tasks. A well-planned design improves system efficiency, enhances security, and ensures ease of use. In the Secure-Docs project, the design is based on three primary modules: administrator, student, and verifier. The administrator uploads academic certificates, generates SHA-256 hashes, and stores them on the blockchain, while the actual files are stored in IPFS. Students are provided with secure access to view and download their issued certificates. Verifiers can confirm the authenticity of

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

certificates using either a hash value or a QR code. The frontend of the application is developed using React, the backend is implemented using Node.js, and MetaMask is integrated to enable secure blockchain transactions.

Architecture

The architecture of the Secure-Docs system represents the organization of components and the flow of data between them. It explains how requests are processed and which modules are involved in handling different operations. The architecture is divided into two main categories: software architecture and technical architecture. These layers provide a structured representation of the system and support scalability, maintainability, and secure communication between components.

Software Architecture

Software architecture describes the logical structure of the application and defines interactions between modules. The Secure-Docs system follows a role-based architecture to clearly separate functionalities. The administrator module is responsible for managing the system, including uploading certificates, generating SHA-256 hashes, issuing certificates, and approving verifier requests. The student module allows users to register, log in, and securely access their certificates for viewing or downloading. The verifier module provides functionality for validating certificates by entering hash values or scanning QR codes, after which authenticity results are displayed. All modules communicate with a central processing system that coordinates blockchain interaction and decentralized storage. This structured architecture ensures proper role separation, controlled access, and efficient workflow management.

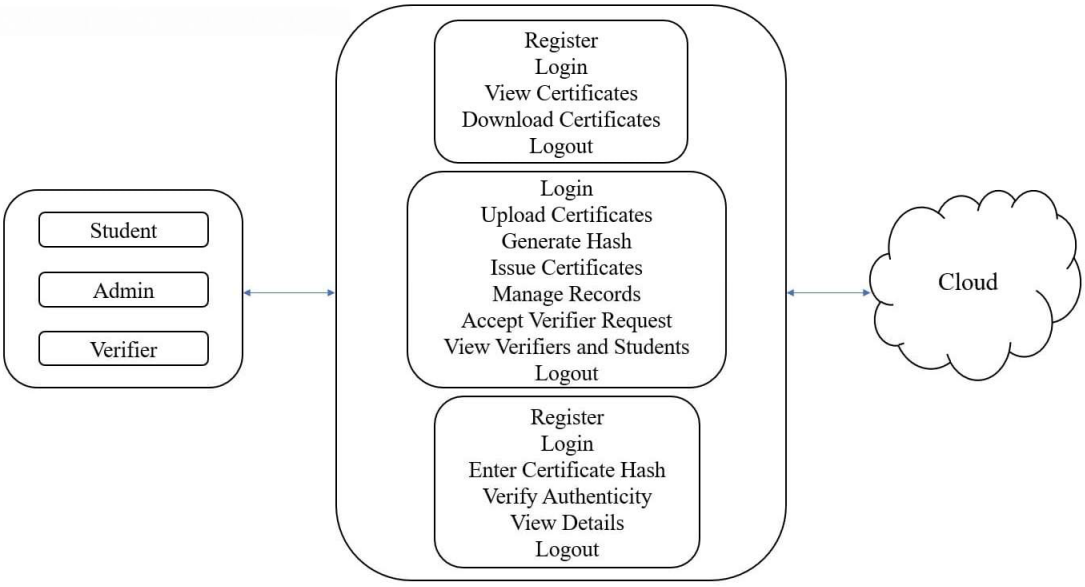

Fig.1 Software Architecture

Software Architecture

Technical Architecture

The technical architecture consists of frontend, backend, blockchain, and decentralized storage components. The frontend, developed using React and integrated with MetaMask, manages user interactions and sends requests to the backend. The backend processes requests, generates cryptographic hashes, and communicates with blockchain and IPFS services. The blockchain layer, implemented using Ethereum and smart contracts, securely stores certificate hash values. IPFS is used for decentralized storage of certificate files, where each file is assigned a unique content identifier. MetaMask is used to sign blockchain transactions securely. This architecture ensures decentralized storage, tamper-proof verification, and secure communication across system components.

Use Case Diagram

The use case diagram illustrates interactions between users and the system. It identifies different actors and the operations they perform. In the Secure-Docs system, the primary actors are administrator, student, and verifier. The administrator uploads certificates, generates hashes, and issues credentials to students. The student registers, logs in, and accesses issued certificates. The verifier validates certificate authenticity using a hash value or QR code. This diagram helps in understanding user responsibilities and ensures that all required functionalities are incorporated into the system design.

Class Diagram

The class diagram represents the static structure of the system and defines classes, attributes, and methods. In the Secure-Docs project, key classes

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

include User, Admin, Student, Verifier, and Certificate. The Certificate class stores information such as hash value, IPFS content identifier, and student details. The Admin class is responsible for issuing certificates, while the Verifier class performs validation operations. Relationships between classes define how data flows within the system. This diagram helps developers understand object relationships and supports structured implementation.

Sequence Diagram

The sequence diagram describes interactions between system components in chronological order. In the Secure-Docs system, it shows communication among user interfaces, backend services, IPFS, MetaMask, and blockchain. During certificate issuance, the administrator uploads a certificate, the backend generates a hash, the file is stored in IPFS, and the hash is recorded on the blockchain using MetaMask. During verification, the verifier submits a hash or QR code, and the system checks blockchain records to determine authenticity. The result is displayed as valid or invalid.

Activity Diagram

The activity diagram illustrates the workflow of certificate issuance and verification processes. In the issuance process, the administrator uploads a certificate, the system generates a cryptographic hash, stores the document in IPFS, and records the hash on the blockchain. In the verification process, the verifier enters a hash value or scans a QR code, and the system checks the blockchain to confirm authenticity. The output is displayed as either valid or invalid. This diagram provides a clear representation of the operational flow of the Secure-Docs system.

Blockchain Smart Contract

The blockchain smart contract defines the logic for storing certificate hashes and retrieving verification details. It includes functions for adding certificate data, validating hash values, and returning stored information. The smart contract ensures immutability and transparency, as data recorded on the blockchain cannot be modified. This component forms the core of the Secure-Docs verification mechanism and guarantees tamper-resistant document authentication.

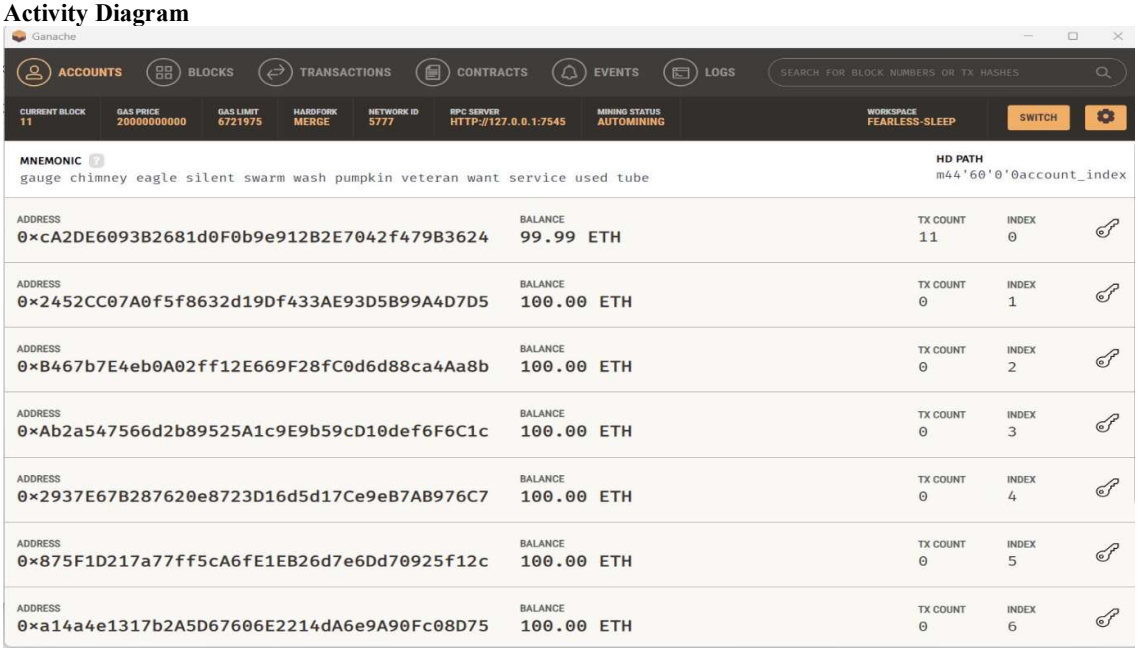

Fig 2Blockchain Smart Contract

IMPLEMENTATION

The Secure-Docs application is implemented using a combination of modern web technologies, blockchain frameworks, and decentralized storage solutions to provide a secure and transparent document verification platform. The system integrates frontend, backend, blockchain, and storage components to ensure tamper-resistant certificate management. By combining decentralized technologies with web-based interfaces, the application eliminates dependency on centralized authorities and enables reliable verification of academic credentials.

Front-End Technologies

The frontend of the Secure-Docs system is developed using React.js, which provides a flexible and component-based architecture for building dynamic user interfaces. React enables the creation of role-based dashboards for administrators, students, and verifiers, allowing each user to interact with system features relevant to their responsibilities. Its reusable components simplify development and maintenance, while the virtual

DOM mechanism improves performance by updating only necessary interface elements. React is integrated with Axios to handle communication with backend APIs and with Web3.js to enable interaction with the Ethereum blockchain. Additionally, QR code generation and scanning functionalities are incorporated to support certificate verification.MetaMask is used as a browser-based wallet that connects the frontend application to the Ethereum blockchain. It provides secure authentication and transaction signing without exposing private keys. Whenever an administrator issues a certificate or a verifier performs validation, MetaMask prompts the user to confirm the transaction. After approval, the transaction is signed and submitted to the blockchain network. This integration ensures that all blockchain interactions are authenticated and tamper-resistant, while also allowing users to manage multiple accounts conveniently.

Back-End Technologies

The backend of the system is implemented using Node.js, which provides a server-side runtime environment for executing JavaScript. Node.js handles core functionalities such as certificate uploads, hash generation, and communication with IPFS and blockchain networks. Its event-driven and non-blocking architecture allows efficient handling of multiple requests, improving overall system performance. Node.js also provides modules for file handling, cryptographic operations, and API communication, making it suitable for developing scalable backend services.Express.js is used as a lightweight web application framework built on top of Node.js. It simplifies the creation of RESTful APIs and manages routing between client and server. In the Secure-Docs application, Express is responsible for handling authentication, request validation, logging, and error management. It also supports file uploads through middleware, enabling efficient processing of certificate documents. Express acts as an intermediary layer connecting frontend components with blockchain and IPFS services, ensuring smooth data flow within the system.

Pseudo Code Description

The implementation logic of the Secure-Docs system is divided into administrator, student, and verifier modules. The administrator module includes authentication, certificate upload, hash generation, and certificate issuance. It also allows approval of verifier requests, revocation of certificates, and secure logout. The student module supports user registration, authentication, viewing issued certificates, downloading documents from IPFS, and displaying QR codes for verification. The verifier module includes registration and login functionality, manual hash-based verification, QR code scanning, and displaying verification results. Each module interacts with backend APIs to perform operations and communicates with blockchain and IPFS components for secure data storage and retrieval. Secure logout functionality is implemented across all modules to ensure proper session management.

TESTING

Testing of the Secure-Docs system was performed to ensure that the application operates reliably, securely, and accurately. Since the system integrates blockchain and decentralized storage technologies, particular attention was given to validating data integrity, authentication mechanisms, and certificate verification functionality. The testing process was conducted in multiple stages, covering individual modules as well as the complete system workflow. This approach ensured that the application met functional requirements and delivered a consistent user experience for administrators, students, and verifiers.

Stages of Testing

Testing was carried out in different stages to validate each part of the system. Initially, unit testing was performed to examine individual modules such as administrator login, student registration, certificate upload, hash generation, and verification features. Each component was tested independently to confirm correct behavior. Integration testing was then conducted to verify that the frontend developed in React, the backend implemented using Node.js, the Ethereum blockchain, and IPFS storage interacted correctly. System testing followed, where the entire application workflow, including certificate issuance, decentralized storage, and verification, was evaluated as a complete system. Finally, acceptance testing was carried out with different user roles to confirm that the application satisfied project requirements and produced accurate results.

Phases of Testing

The testing process followed a structured approach consisting of requirement validation, test planning, test execution, bug identification and fixing, and final validation. During requirement validation, system specifications were reviewed to define testing objectives. Test planning involved preparing test scenarios and identifying expected outcomes. Test execution included running the application under different conditions and monitoring results. Bugs detected during execution were corrected, and the system was retested. Final validation ensured that all modules functioned correctly and met the intended performance criteria.

Test Cases

Test cases were prepared for each user role to validate system functionality. Administrator test cases included login validation, certificate upload, hash generation, issuing certificates, managing records, approving verifier requests, and logout operations. Both valid and invalid inputs were tested to ensure proper error handling and secure

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

authentication. The results confirmed that administrator functionalities operated correctly and produced expected outcomes.Verifier test cases evaluated registration, login, hash-based verification, QR code scanning, viewing certificate details, and logout features. The verification process accurately identified valid certificates by comparing hash values stored on the blockchain. Invalid inputs produced appropriate error messages, ensuring reliable validation. QR code scanning also functioned correctly and returned verification results in real time.

SCREENSHOTS

Displaying the Title Page of Secure-Docs

Fig 6.1 Home Page

Signing in with sign in credentials

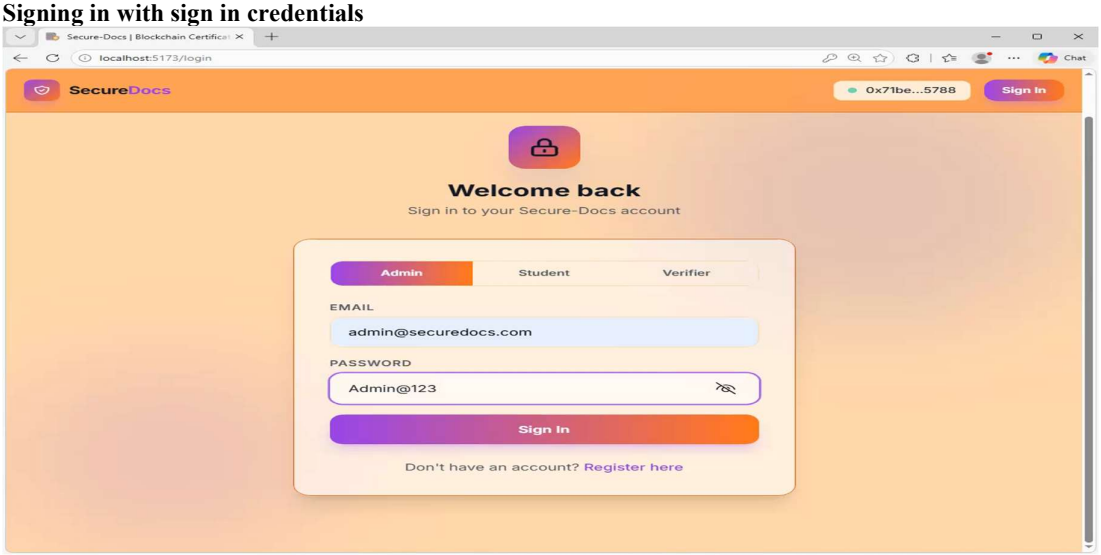

Fig 6.4 Admin Sign-In

Displaying the Certificate to Download

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

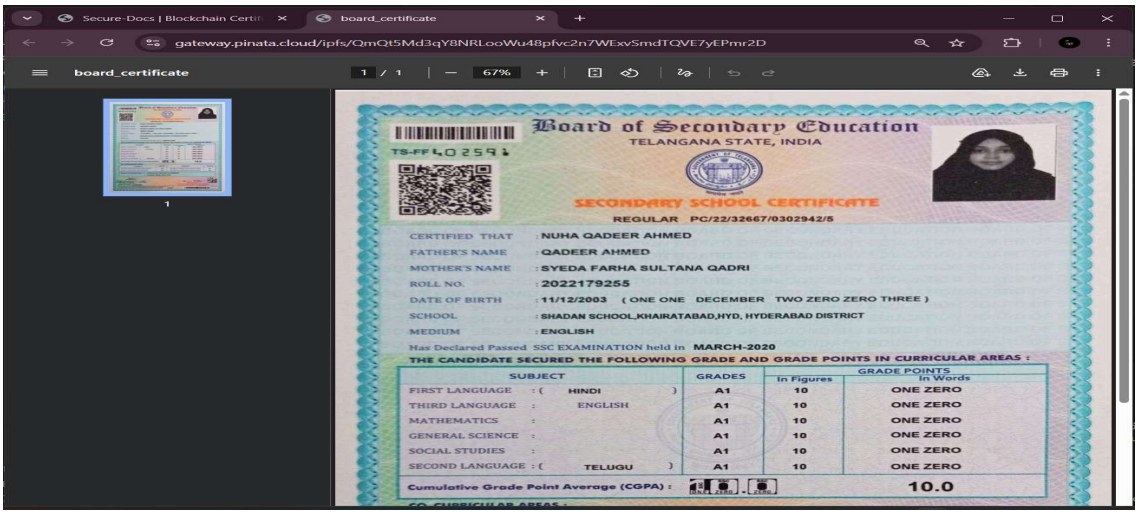

Fig 6.16 Download the Certificate

Displaying QR Code of Certificate

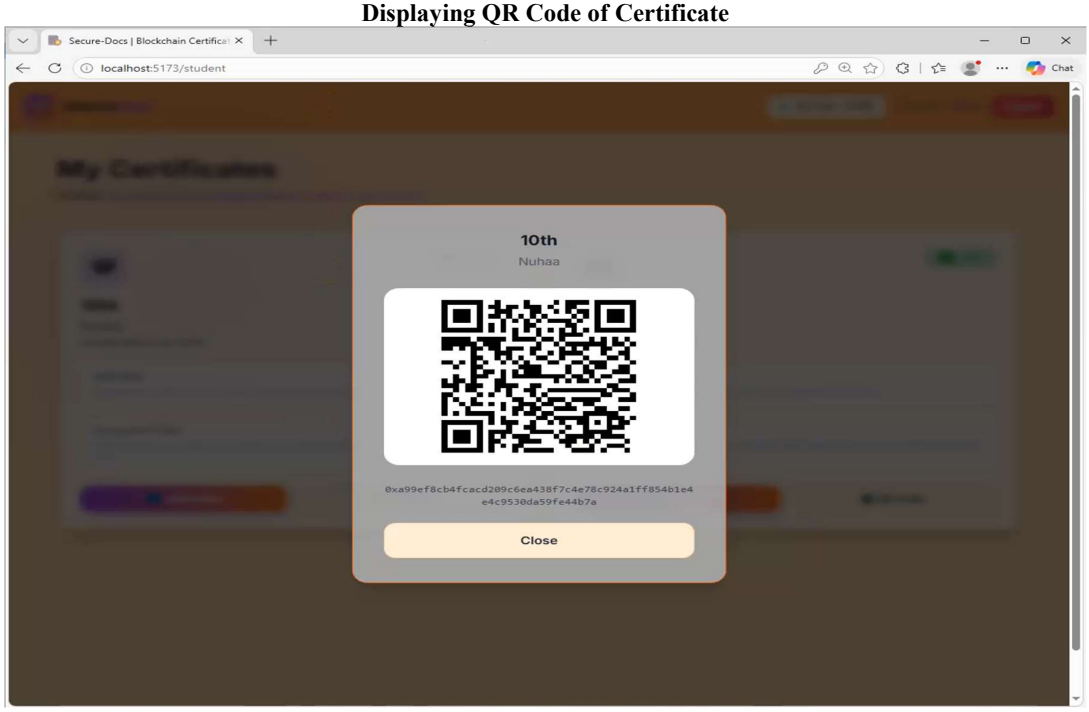

Fig 6.17 QR Code of Certificate

Showing that the QR Code is Valid and the Verification Successful

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

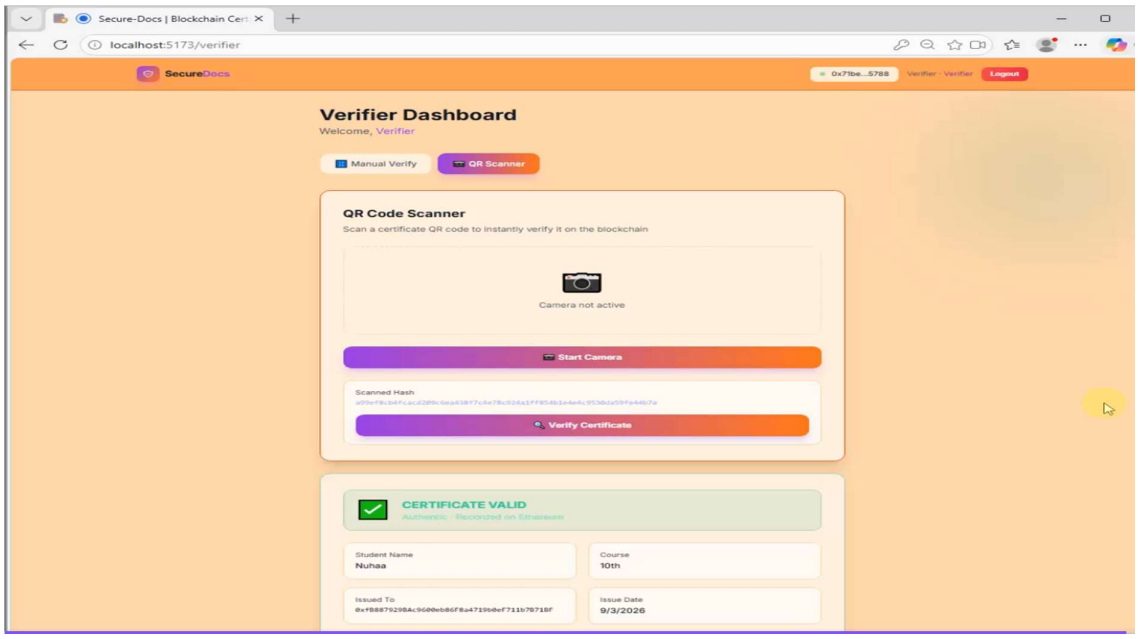

Fig 6.26 Certificate Verified

IPFS

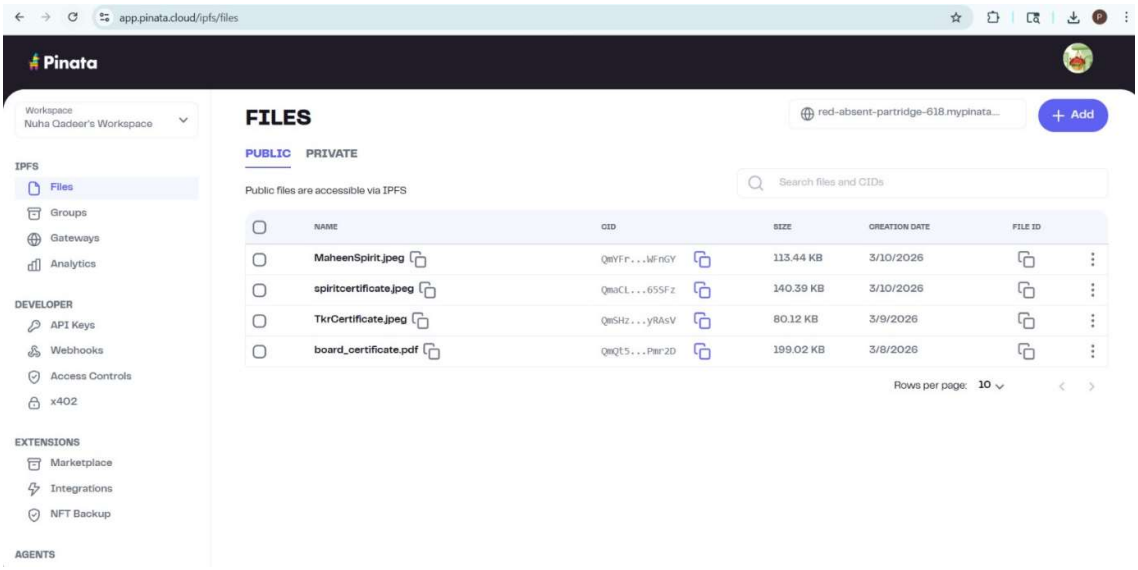

Fig 6.3 IPFS

Conclusion

The Secure-Docs system demonstrates an efficient and reliable approach for storing and verifying academic certificates using Blockchain technology integrated with IPFS. By generating unique cryptographic hashes and maintaining records on a decentralized ledger, the system significantly reduces the possibility of document tampering and forgery. The use of IPFS ensures distributed storage, improving data availability and integrity.

The developed platform simplifies certificate verification for institutions, employers, and students by providing a transparent and trustworthy validation mechanism. Additionally, the system enhances security, minimizes manual verification efforts, and improves operational efficiency. Overall, the proposed solution offers a user-friendly, scalable, and dependable method for managing academic credentials in a secure digital environment.

Future Scope

The Secure-Docs platform can be further enhanced by integrating it with university management systems and government databases to enable real-time certificate issuance and verification. Advanced encryption techniques may be incorporated to strengthen data privacy, while multi-factor authentication can improve access control and prevent unauthorized usage.Future development

may also include the implementation of smart contracts on the Ethereum blockchain to automate certificate issuance, revocation, and verification processes. The system can be extended to mobile applications for wider accessibility and ease of use. Furthermore, the solution can be adapted for other sectors such as healthcare, professional licensing, and corporate training, where secure document validation is essential.

REFERENCES

1) [1] N. Vikhankar et al., "E-Certificate Verification Using Blockchain," *International Journal of Engineering Research & Technology (IJERT)*, vol. 13, no. 5, pp. 1–4, May 2024.
2) [2] T. Ifeyemi et al., "Blockchain-Based Digital Educational Certificate Verification System," *ITEGAM Journal of Engineering and Technology for Industrial Applications*, vol. 10, no. 49, pp. 25–31, Sep. 2024.
3) [3] S. Gangwar and A. Chaurasia, "Blockchain-Based Authentication and Verification System for Academic Certificate," *International Journal of Computer Applications*, vol. 186, no. 26, pp. 12–16, Jul. 2024.
4) [4] S. Nayak et al., "Engineering Degree Certification Verification Using Blockchain Technology," *International Journal of Engineering Research & Technology (IJERT)*, vol. 14, no. 5, pp. 20–25, May 2025.
5) [5] M. Al Hemairy et al., "Blockchain Framework for Validation and Authentication of Academic Certification," *Education and Information Technologies*, Mar. 2024.
6) [6] R. R. S. et al., "Secure Academic Credential Management Using Blockchain-Based Self-Sovereign Identity," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 13, no. 6, pp. 75–80, 2025.