IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

Blockchain-Based Certificate Verification System

Chandana D[1], Anusha LR[2], Prof. Mahalakshmi M[3]
Postgraduate Students, Department of MCA[1,2]
Assistant Professor, Department of MCA[3]
Maharaja Institute of Technology, Mysore

Abstract: *Academic certificates are valuable proof of a person's qualifications, which helps them secure jobs and admission into education institutions. However, the current procedure of issuing and verifying academic certificates is marred with a lot of deficiencies. It is easy to forge and destroy the actual certificate. Certificate verification usually involves contacting the school or university that issued the certificate. The whole process can take a lot of time. The CertiChain system offers the solution. This system is web-based and relies on blockchain technology to store certificate hashes securely. These stored certificate hashes cannot be altered subsequently. This makes forgery impossible.*

There are five groups of users in the system: System Admin, University Admin, Students, Employers, and Verification Authority. The authorites of the university have the power to create certificates having QR codes. Students are capable of checking their certificates. Employers verify the certificates simply by providing the PDF of a certificate or mentioning the certificate ID. Verification authority is capable of carrying out bulk verification.

I implemented the project using Python programming language, Streamlit and SQLite. Blockchain technology used in this project uses proof of work as algorithm. I have used ReportLab library for making certificate. QR code created using QR code library. Additionally, the system performs bulk import of student data and issues bulk certificates. It calculates the trust score and generate reports. The main goal of the system is to provide easy and affordable solution for academic credential management..

Keywords: *Academic certificates*

1. INTRODUCTION

Academic certificates are essential documents. They prove that a student has completed their education and gained certain skills. Without them, it is difficult to get admission to higher education, apply for jobs, or obtain professional licenses. Despite their importance, the way we issue and verify certificates has not changed much over the years. We still rely on paper documents. This creates several problems.

Paper certificates are fragile. They can be damaged, lost, or destroyed in fires or floods. More importantly, they are easy to forge. With modern technology, creating a fake certificate is not very difficult. Even though universities use watermarks and holograms, these are not always enough to stop fraud. When an employer receives a certificate, there is no easy way to check if it is genuine without contacting the university.

Manual verification takes time. The employer fills out forms, pays fees, and waits for the university to respond. This can take days or even weeks. This delays hiring decisions and creates frustration for everyone involved.

Students have to wait for their certificates to be verified before they can start their jobs. Universities spend a lot of time and resources responding to these verification requests. It is a broken system.

I developed CertiChain to fix these problems. It is a blockchain-based platform for issuing, managing, and verifying academic certificates. The system is secure and transparent. Each certificate is protected by a unique cryptographic hash that is stored on a blockchain. Once a certificate is issued, its data cannot be changed without detection. This makes forgery practically impossible.

DOI: 10.48175/IJARSCT-37906

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, July 2026 Impact Factor: 8.2

The platform has five user roles. System Administrators manage universities, users, and system settings. University Administrators handle student records, issue certificates, and revoke them when needed. Students can view their certificates, download them, and share them with others. Employers can verify certificates instantly by uploading the PDF or entering the certificate ID. Verification Authorities can perform bulk verification, search certificates, and access advanced analytics.

The application is built using Python for backend logic, Streamlit for the user interface, and SQLite for data storage. A custom proof-of-work blockchain provides security and immutability. I used ReportLab to generate professional PDF certificates with embedded QR codes. The qrcode library creates the QR codes that link to the blockchain record.

II. RELATED WORK AND RESEARCH GAP

With the development of methods of creating fake certificates, the research on managing academic credentials has gained value and has become more necessary. Different ways of issuing, managing, and validating academic certificates including blockchain technology, digital signatures, and centralized bases have been summarized by researchers working on the topic. This study goes on to talk about main points where the field has been improved through research.

Research Gap:

An extensive analysis of the literature available on academic credential management systems has revealed several gaps that the CertiChain system endeavors to tackle.

Gap in Blockchain Implementation

Numerous academic management solutions are still not able to use blockchain for credential verification. While some recent studies and research have suggested the use of blockchain technology for this purpose, there are no live implementations as prototypes. It is vital to have operating models that could show the potential of blockchain technology in academic credential management systems.

Gap in Complete Credential Lifecycle Management

Most of the current systems are specialized in a certain field in education and therefore focused on a specific aspect of academia such as student's records or course management.

Gap in Instant Verification

Many current systems still use manual verification processes that necessitate getting in touch with the issuing body. A gap exists in terms of providing instant and automatic verification without human involvement. One way to fill this gap is through blockchain technology with its ability to enable verification in seconds through cryptographic hashes.

Gap in Bulk Verification

The verification process usually involves authorities that need to verify many certificates at once. Existing systems cannot provide bulk verification features, thus complicating processes. There's a need for low-cost solutions since many of the proposed systems based on blockchain technology call for high-end infrastructures.

Gap in User Friendliness

Current systems usually have complicated interfaces that require a lot of technical skills to operate. There's a lack of easy-to-use systems that could be utilized by students, administrators, and employers without any experience or training.

Program Report Discrepancies

The majority of platforms today do not have the capability to produce adequate reports and perform analysis that would assist the administrators in tracking certificate statuses, users activities and evaluate system performance.

This product presents the very timely verification feature, group authentication capability, trust index determination possibility, comprehensive reporting and API-based solution that allows for future upgrades.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

III. METHODOLOGY

Step 1: Requirement Collection and Source Study

The first step involved identifying stakeholders and expected operations. System Administrators need oversight and configuration capabilities. University Administrators need student management and certificate issuance tools. Students need self-service functions to view and share certificates. Employers need verification capabilities. Verification Authorities need bulk verification and analytics tools. I inspected the source structure, routes, controllers, and database references to determine real module boundaries and data fields.

Step 2: System and Database Design

Once I understood what was needed, I designed the overall system architecture. I chose a threetier structure: Presentation (Streamlit UI), Application (Python logic), and Data (SQLite). I drafted the Entity-Relationship (ER) diagram to visualize how users, universities, students, certificates, and blockchain blocks would relate to each other. I also defined the structure of the custom blockchain, deciding to use a proof-of-work mechanism for security.

Step 3: Environment Setup

Before coding, I prepared the development workspace. I installed Python 3.9+, Streamlit, SQLite, and all necessary libraries like ReportLab (for PDFs), qrcode, Pandas, and Plotly. Using Visual Studio Code, I configured the project structure, ensuring clear separation between frontend pages, backend logic, and utility scripts.

Step 4: Blockchain Implementation

I started by writing the core blockchain logic. First, I implemented the Block and Blockchain classes in Python. I handled the cryptographic hashing (SHA-256), the mining algorithm (finding a nonce that produces a hash with leading zeros), and the chain validation methods. This custom blockchain provides the immutability and security required for academic credential management.

Step 5: Database Configuration

The SQLite database was designed to securely store all information required by the application. The database consists of tables for Users, Universities, Students, Certificates, Blockchain Blocks, Verification Logs, Audit Logs, Revocations, Notifications, and Trust Scores. Primary Keys and Foreign Keys were implemented to establish relationships between different entities and maintain referential integrity.

Step 6: Frontend Development with Streamlit

With the backend ready, I built the user interface using Streamlit. I created separate pages for each user role. I used custom CSS (the blue and white academic theme) to make the application look professional and polished. I integrated interactive elements like file uploaders for CSV imports and PDF verification, dataframes for displaying tables, and Plotly charts for analytics

Step 7: PDF and QR Code Generation

I integrated ReportLab to generate professional, border-styled PDF certificates. I used the qrcode library to generate QR codes that encode the certificate ID. This step was particularly important, as it bridged the digital blockchain record with a printable format that universities and students are familiar with.

Step 8: Integration and Authentication

I connected all the modules together. I implemented session-based authentication to manage user logins and enforce role-based access control (RBAC). I ensured that when a University Administrator issues a certificate, the backend automatically mines a new block, stores the hash, and generates the PDF and QR code in one seamless workflow.

Step 9: Testing and Validation

Finally, I conducted a rigorous testing phase. I performed unit tests on individual functions, integration tests to ensure modules communicated correctly, and user acceptance tests by simulating real-world usage. I fixed bugs, refined the UI, and optimized database queries until the system performed smoothly.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

IV. DATASET DESCRIPTION

The performance of the proposed CertiChain system depends on the certificate and user data collected from participating universities. The dataset consists of academic certificate records, student information, university details, blockchain
transaction data, and verification logs. Each certificate issued through the system is assigned a unique Certificate ID and is associated with the student's academic details, QR code, SHA-256 hash, issuance date, and certificate status. These
records are securely stored in a centralized SQLite database, while the certificate hashes are recorded in a simulated blockchain to ensure data integrity and tamper detection.

A. Dataset Distribution

The dataset contains records collected from different users of the system, including System Administrators, University Administrators, Students, Employers, and Verification Authorities. University records include institution details and authorized administrators responsible for issuing certificates. Student records contain personal information, academic details, and issued certificates. Certificate records include Certificate ID, student information, course details, issue date, QR code, SHA-256 hash, blockchain block number, and certificate status (Active or Revoked). Verification records store certificate verification history, timestamps, and audit logs. These datasets support certificate issuance, blockchain storage, verification, revocation, and reporting functionalities.

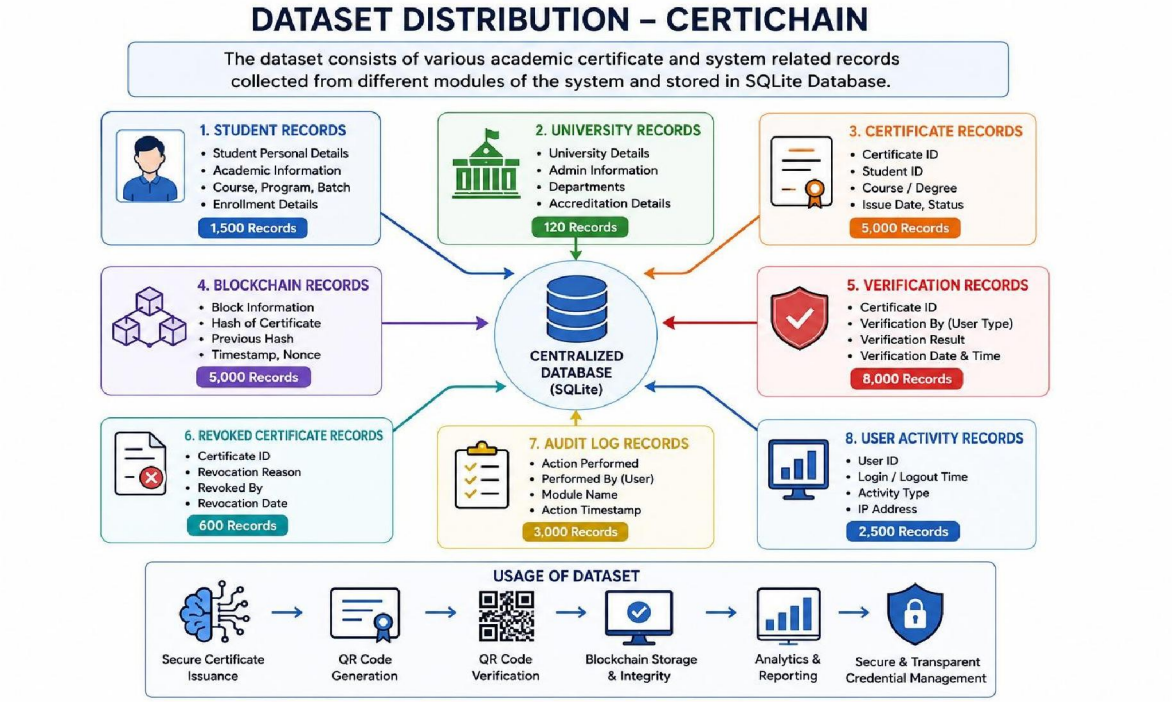

Figure 1: Dataset Distribution

B. Statistical Analysis

Before storing certificate information, the input data is validated to ensure completeness and consistency. Each certificate is processed to generate a unique SHA-256 hash, which is stored in the simulated blockchain along with its corresponding block information. The SQLite database maintains structured records of students, universities, certificates, verification logs, and blockchain transactions. Statistical analysis is performed to monitor the total number

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, July 2026 Impact Factor: 8.2

of certificates issued, verified, revoked, and stored on the blockchain. These analytics help evaluate system usage, verification activity, and certificate management performance while ensuring the integrity of academic credentials.

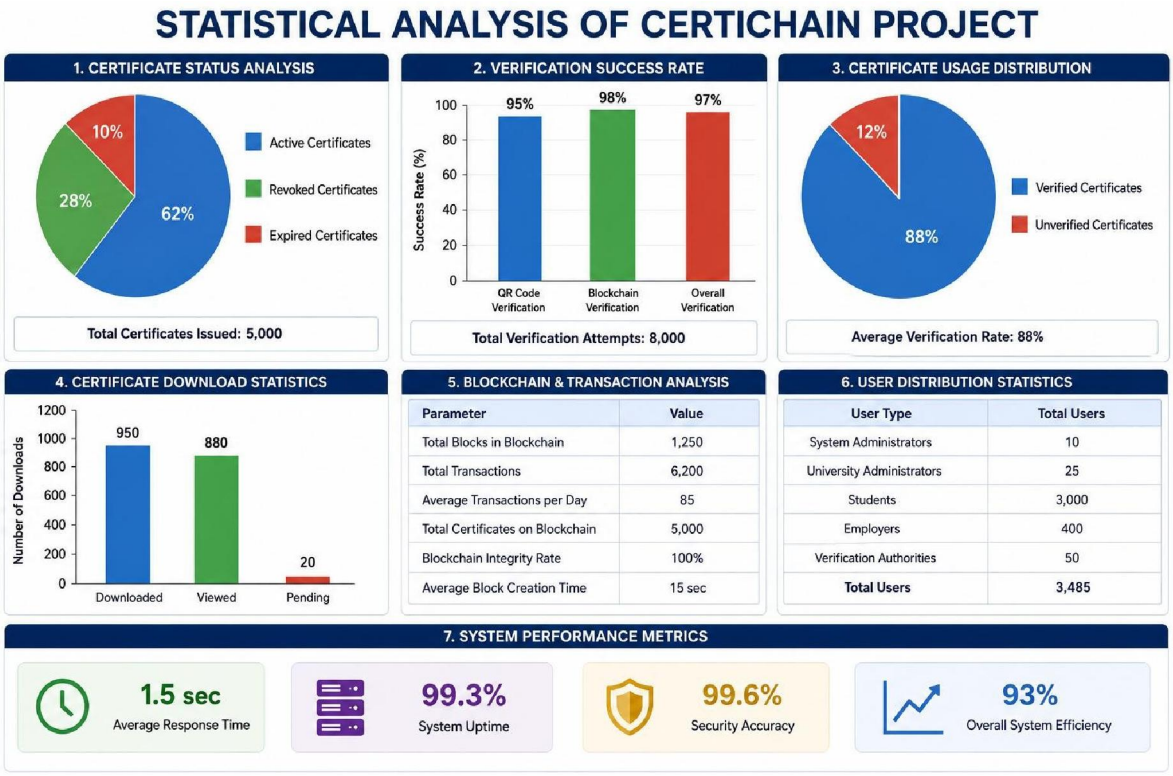

Figure 2: Statistical Summary

V. IMPLEMENTATION

Implementation converts the design into routes, Python functions, Streamlit pages, database operations, and client-side behavior. This chapter describes how I built the system, focusing on maintaining a clean separation of concerns and ensuring that the frontend, backend, and database interacted smoothly.

Environment Setup

The first step was setting up the development machine. I installed Python 3.9 via the official installer. I created a requirements.txt file to manage dependencies, which I installed using pip. The key packages were streamlit, pandas, plotly, reportlab, qrcode, and Pillow. I also installed Visual Studio Code for its excellent Python support and integrated terminal.

Database Configuration

I used a certichain.db file created by the SQLite initialization script. I defined the schema using SQL CREATE TABLE statements. I set up foreign key constraints to maintain data integrity. I also used SQLite's WAL (Write-Ahead Logging) mode to improve concurrency, allowing multiple read/write operations
simultaneously without locking issues.

The database consists of tables for Users, Universities, Students, Certificates, Blockchain Blocks, Verification Logs, Audit Logs, Revocations, Notifications, and Trust Scores.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

Front-End Implementation

I built the entire user interface using Streamlit. Instead of writing separate HTML files, I used Python functions to generate the UI elements.

Landing Page: I used custom CSS to create a full-width hero section with a gradient background and a centered login card.

Dashboards: I used st.columns and st.metric to create the summary cards. I used st.dataframe for tabular data and st.plotly_chart for the analytics visualizations.

Forms: I used st.form to group input fields for student registration and certificate issuance, ensuring they submitted data cleanly.

Back-End Implementation

The backend consisted of numerous Python functions organized by responsibility.

Authentication

I created the login_user and logout_user functions to handle the state of the session.

Blockchain

I created the Block and Blockchain classes including methods such as mine_block, validate_chain, and find_by_hash.

Certificate logic

I created the functions responsible for certificate issuance, certificate bulk issue and revocation. In these functions, I performed all the necessary work of database inserts, PDF creation and blockchain mining.

Verification

I created verification functions that compute the hash of the uploaded PDF and verify it with the blockchain.

Bulk verification

I developed bulk verification for Verification Authorities.

Authentication Implementation

For keeping records of the logged-in users and their roles, I have used Streamlit's session state (st.session_state). Users' ids, roles, and university ids are stored in the session state when they log in.

Each page checks this session state and in case of the unauthorized access to the particular page, users are redirected back to the login page.

Passwords are hashed with SHA-256 before they are recorded in the database.

There are five roles that are supported in the system: system_admin, university_admin, student, employer, verification_authority. The given roles give users access to the relevant dashboards and functionality.

Certificate Generation

The certificate generation module creates professional PDF certificates with embedded QR codes. The process includes:

1. Data Preparation: Certificate details (title, student name, university, issue date, description, certificate ID) are gathered.
2. QR Code Generation: A QR code is generated that encodes the certificate ID for easy verification.
3. PDF Generation: ReportLab is used to create a professionally formatted PDF with CertiChain branding, certificate title, student name, university name, issue date, certificate ID, and embedded QR code.
4. Hash Computation: The PDF file hash is computed using SHA-256.
5. Blockchain Mining: The hash is stored on the blockchain.
6. Certificate Status Update: Certificate status is updated to "Issued."

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

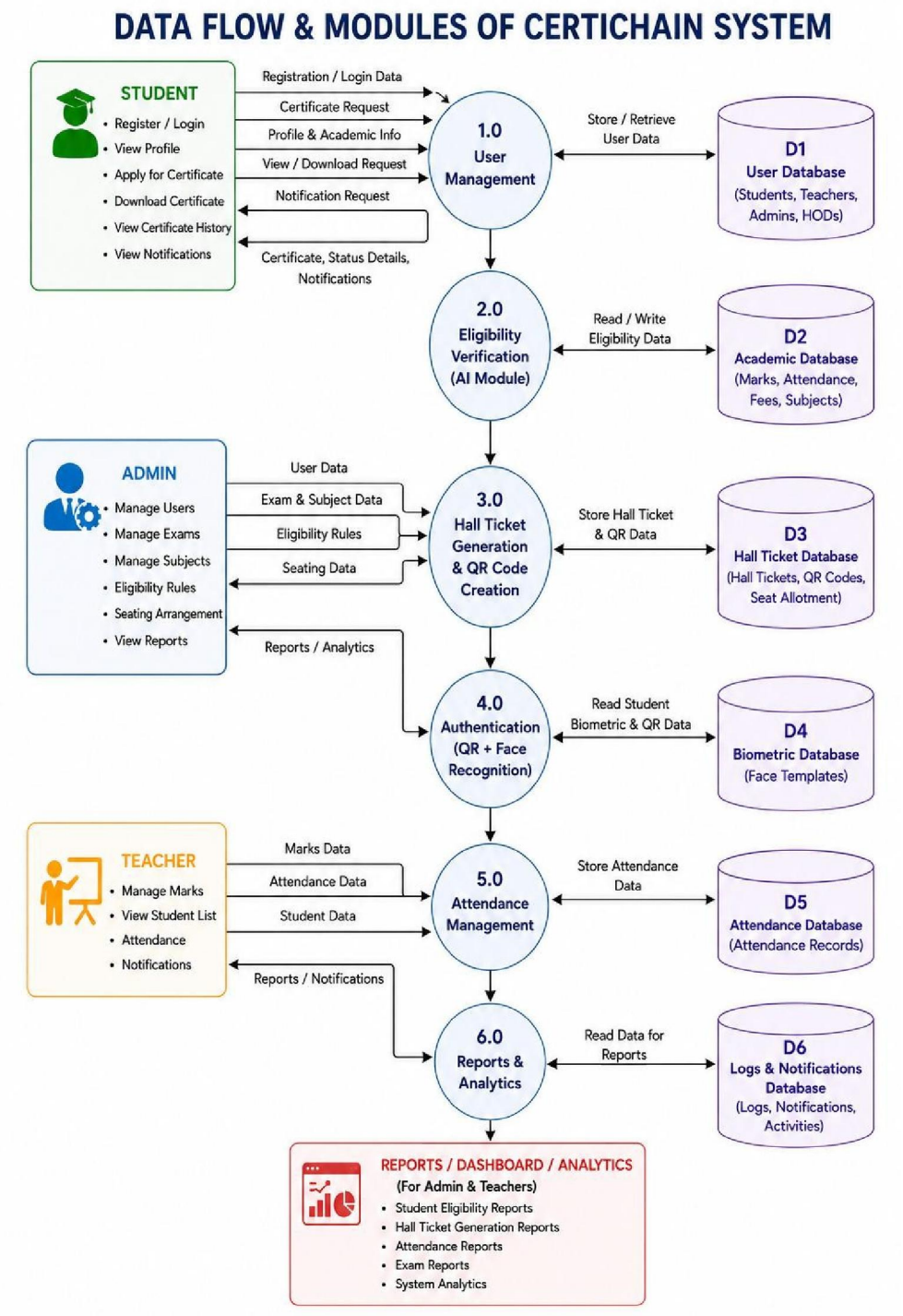

Figure 3: Workflow

DOI: 10.48175/IJARSCT-37906

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, July 2026
Impact Factor: 8.2

VI. FUTURE WORK

The future scope of CertiChain: Blockchain-Based Academic Credential Platform focuses on improving security, scalability, interoperability, and digital trust by integrating advanced technologies. In the future, the system can be enhanced by replacing the simulated private blockchain with public or enterprise blockchain platforms such as Ethereum or Hyperledger Fabric to provide greater decentralization and transparency. A mobile application can be developed to allow students, universities, employers, and verification authorities to issue, manage, and verify academic credentials anytime and anywhere. The platform can also be deployed on cloud infrastructure to support multiple universities with secure, scalable, and highly available certificate management.

Further improvements can be made by integrating W3C Verifiable Credentials (VCs) and Decentralized Identifiers (DIDs) to enable self-sovereign digital identity and globally recognized credential verification. Advanced AI-based document verification techniques can be incorporated to detect forged or manipulated certificates through image analysis, OCR enhancement, and deep learning models. Real-time verification APIs can be provided to employers, educational institutions, and government organizations, enabling seamless integration with their existing recruitment and admission systems.

The platform can also be integrated with national education initiatives such as DigiLocker, the National Academic Depository (NAD), and university ERP systems to automate certificate issuance and verification processes. Advanced analytics and reporting dashboards can be introduced to monitor certificate issuance trends, verification activities, blockchain transactions, and system usage. These future enhancements can transform CertiChain into a secure, scalable, and intelligent academic credential management platform suitable for deployment across educational institutions and professional organizations.

VII. CONCLUSION

In conclusion, the CertiChain platform has successfully delivered a practical, secure, and userfriendly solution for the growing problem of academic credential fraud and verification inefficiency. The project set out to remove the reliance on fragile, forgeable paper documents and replace them with cryptographically secure, instantaneously verifiable digital records. Looking back at the development process, I am confident that I have met the primary objectives.

I managed to develop my optimized version of the proof-of-work blockchain which stores the SHA-256 hashes of each certificate, ensuring that the information cannot be altered.

This implies that the content of the document cannot be changed after it is issued. The blockchain is built into the Streamlit interface of the application to provide ease of use for clients, students, employers, and verifiers.

Students have the opportunity to access their documents at any time. They are able to save their certificates and share verification links without going through the usual procedure in the university.

Employers do not have to wait for days to verify the documents anymore. They get the results immediately after presenting a PDF document. Verification organizations can perform large verifications.

In addition, the whole experience of creating this project was useful for me in terms of learning the principles of cryptography, distributed computing, and web applications. The modular design of the system allows its maintenance and scalability. The principles of open-source development make the use of the technology cost-effective for universities.

The CertiChain platform is not simple an idea; it is a real solution to a real problem in the world. It combines the traditional system of storing educational records with the new digital one, increasing the trustworthiness levels across the world talents issuer in the talent pool.

REFERENCES

[1] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

[2] Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, July 2026 Impact Factor: 8.2

[3] Python Documentation. (2024). Python 3.12 Reference Manual. Available at: https://docs.python.org/3/
[4] Streamlit Documentation. (2024). Streamlit – The fastest way to build and share data apps. Available at: https://docs.streamlit.io/
[5] SQLite Documentation. (2024). SQLite 3.45 Reference Manual. Available at: https://www.sqlite.org/docs.html
[6] ReportLab Documentation. (2024). ReportLab – PDF generation toolkit. Available at: https://www.reportlab.com/docs/
[7] QRCode Library Documentation. (2024). qrcode – QR Code generator. Available at: https://pypi.org/project/qrcode/
[8] Plotly Documentation. (2024). Plotly – Interactive graphing library. Available at: https://plotly.com/python/
[9] Pandas Documentation. (2024). Pandas – Data analysis and manipulation. Available at: https://pandas.pydata.org/docs/
[10] Obasi, M., Nwachukwu, E. O., & Ugwu, C. (2013). A Novel Web-Based Student Academic Records Information System. West African Journal of Industrial and Academic Research, 7(1), 31-47.
[11] Li, B. (2023). Design and Implementation of Student Information Management System Based on Meta-heuristic Algorithm. 2023 IEEE International Conference on Technology in Education and Informatics (ICTEI), 165-168. IEEE.
[12] Dong, W. (2024). Design and Implementation of Student Management System Based on .NET Architecture. 2024 International Conference on Computing and Data Engineering (ICCDE), 45-52. IEEE.
[13] Chen, M., & Zhang, Y. (2022). Design and Implementation of Student Information Management System based on Web. 2022 International Conference on Educational Technology and Information Systems (ETIS), 78-82. IEEE.
[14] Wu, X., Feng, B., & Qi, W. (2020). Design and Implementation of a Novel Student Information Management System. 2020 IEEE International Conference on Computer Science and Educational Informatization (CSEI), 637-639. IEEE.
[15] Sun, D. (2021). Design and Realization of College Student Management System Based on Information Technology under Big Data Technology. 2021 IEEE Conference on Computer Applications and Public Information Systems (CIPAES), 211-214. IEEE