An IOMP

On

CERTIFICATE VALIDATION USING BLOCKCHAIN TECHNOLOGY

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

By

KALASHETTY LIKHITHA 227Y1A6220

GUGULOTHU SHILPA 227Y1A6244

Under the Guidance of

MRS – BHAVYA VARMA, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

MARRI LAXMAN REDDY

INSTITUTE OF TECHNOLOGY & MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

NAAC Accredited Institution with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

JUNE, 2025

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY & MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
NAAC Accredited Institution with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

Date: 30-June-2025

CERTIFICATE

This is to certify that the project work entitled **"Certificate Validation Using Blockchain Technology"** work done by **Likhitha(227Y1A6220), Shilpa(227Y1A6244)** students of Department of Computer Science and Engineering (Cyber Security), is a record of Bonafide work carried out by the member(s) during a period from Jan, 2025 to June, 2025 under the supervision of **Mrs. Bhavya Varma, Assistant Professor**. This project is done as a fulfilment of obtaining Bachelor of Technology Degree to be awarded by Jawaharlal Nehru Technological University Hyderabad, Hyderabad.

The matter embodied in this project report has not been submitted by me/us to any other university for the award of any other degree.

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

KALASHETTY LIKHITHA **GUGULOTHU SHILPA**

Date: (Mrs. Bhavya Varma)

The Viva-Voce Examination of above student(s), has been held on.....

Head of the Department **Principal/Director**

External Examiner

ACKNOWLEDGEMENTS

we would like to express my sincere gratitude to my guide **Mrs. Bhavya Varma, Assistant Professor**, Computer Science and Engineering (Cyber security), for his/her excellent guidance and invaluable support, which helped me accomplish the BTech (CSC) degree and prepared me to achieve more life goals in the future. His total support of my dissertation and countless contributions to my technical and professional development made for a truly enjoyable and fruitful experience. Special thanks are dedicated for the discussions we had on almost every working day during my project period and for reviewing my dissertation.

I/we am very much grateful to my Project Coordinator, **Dr. B Rebecca, Associate Professor,** Cyber Security, MLRITM, Dundigal, Hyderabad, who has not only shown utmost patience, but was fertile in suggestions, vigilant in directions of error and has been infinitely helpful.

I/we am extremely grateful to **Dr. M. Venkata Reddy, Associate Professor,** MLRITM, Dundigal, Hyderabad, for the moral support and encouragement given in completing my project work.

I/we wish to express deepest gratitude and thanks to **Dr. P. Sridhar, Director and Dr. R. Murali Prasad, Principal,** for their constant support and encouragement in providing all the facilities in the college to do the project work.

I/we would also like to thank all our faculties, administrative staff and management of MLRITM, who helped me to completing the mini project.

On a more personal note, I thank my **beloved parents and friends** for their moral support during the course of our project.

TABLE OF CONTENTS

				Page No.
Certificate				*ii*
Acknowledgements				*iii*
Table of Contents				*iv*
List of Figures				*vi*
List of Abbreviations				*vii*
List of Tables				*viii*
Abstract				*ix*
Chapter 1: Introduction				1-5
	1.1	Overview		1
	1.2	Purpose of the project		1
	1.3	Motivation		2
	1.4	Scope of the project		3
	1.5	Objectives		3
	1.6	Limitations		4
Chapter 2: Literature Survey				4-7
	2.1	Preferred Language		4
	2.2	Existing System		5
	2.3	Limitations of Existing Systems		6-7
Chapter 3: Proposed System				8-14
	3.1	Proposed System		8
	3.2	System requirements		9
		3.2.1	Software requirements	10
		3.2.2	Hardware requirements	10
		3.2.3	Functional requirements	11
		3.2.4	Non-Functional requirements	12
	3.3	Concepts used in the proposed system		13

	3.4	Data Sets used in proposed system		14
Chapter 4: System Design				15-20
	4.1	Components/ User in the proposed system		15
	4.2	Proposed System Architecture		16-17
	4.3	UML diagrams		18-20
		4.3.1	Use case diagram	18
		4.3.2	Sequence diagram	19
	4.4	Modular Diagram		20
Chapter 5: Implementation				21-26
	5.1	Source Code		26
Chapter 6: Implementation and Result				27-32
	6.1	Results		32
Chapter 7: Conclusion and Future Scope				33-34
	7.1	Conclusion		33
	7.2	Future Scope		34
Chapter 8: References				36
Abstract (Initial Signed Copy)				37-38
Yukti Innovation Repository Form				39-41
Publications (if any)				

LIST OF FIGURES

Figure No	Name Of the Figure	Pg.no
1.1	Centralized Ledger and Distributed Ledger	1
3.1	Design Process for Blockchain System	9
4.1	Certificate validation system overview	18
4.2	Use case diagram	18
4.3	Sequence diagram	19
4.4	Module diagram	20
6.1.1	Welcome page	27
6.1.2	Login page	27
6.1.3	Verification Form	28
6.1.4	Add certificate	28
6.1.5	Verifying QR code	29
6.1.6	Authentic Result	30
6.1.7	Verifying QR code	31
6.1.8	Verifying URL	32
6.1.9	Invalid certificate	32

LIST OF TABLES

Table No.	Name of the Table	Page No.
2.1	Comparison of various hashing mechanism	4
2.2	Summary feature of top5 blockchain platforms	6
3.2.1	Software Requirements of the Proposed System	10
3.2.2	Hardware Requirements of the Proposed System	10

ABSTRACT

As education becomes more diversified, decentralized and democratized, we still need to maintain reputation, trust in certification and proof of learning. Nowadays everyone has to show his/her Document and Certificate to any other person for some purpose/job. After seeing the document 3rd person cannot validate the originality of the certificate.

The same thing is applied for a land registry, PAN card, and Aadhar card verification. The increased focus on relevance and employability may also push us in this direction, as we also need more transparency. We can solve this problem or get trust by using blockchain technology. The digital currency Bitcoin is probably the best-known application of blockchain and is even better known than the Blockchain technology on which it is based. The blockchain is a chain of blocks and blocks are immutable in a distributed environment, which storage devices are not all connected to a common processor. It is a database of records/public ledger of all transactions /digital events that have been performed and information is shared within participating parties.

Each entry in the system is verified by common consent of the participants in the system. Once information is entered in blockchain it cannot be erased. It could provide a system that is transparent and secure. Blocks (Ordered Records) are added to blockchain with timestamp and a link to a previous block.

The system uses blockchain to securely issue and verify certificates. Certificate data is stored as hashes on-chain, while full details remain off-chain to protect privacy. Smart contracts and QR codes allow easy and tamper-proof verification, making the process faster, more secure, and trustworthy.

CHAPTER – 1
Introduction

1.1 OVERVIEW

Industrial Revolutions have been transforming industrial processes to facilitate mass production and large numbers of business transactions. New technologies are being introduced to disrupt previous ones to make the lives of people easier and more productive. We are witnessing the fourth industrial revolution, which primarily represents Industrial cyber physical systems loaded with technologies like Big Data, AI, IoT, and cloud computing. These systems have potential to bring disruptions for all traditional business models and hence there is an imperative need for a redesign and digitization of activities.

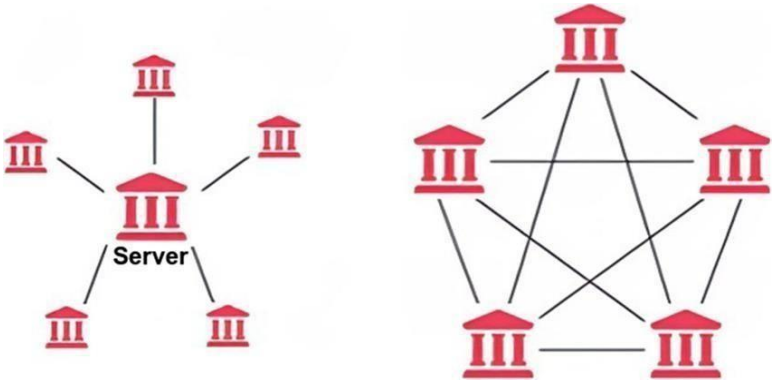


Figure 1.1: Centralized Ledger and Distributed Ledger

1.2PURPOSE OF THE PROJECT

The purpose of this project is to develop a secure, transparent, and efficient system for issuing and verifying

academic and professional certificates using blockchain technology. Traditional methods of certificate validation are often slow, manual, and vulnerable to fraud and forgery. This project aims to overcome these limitations by leveraging the decentralized and immutable nature of blockchain. By storing certificate data on a blockchain, the system ensures that each certificate is:

Tamper-proof – once issued, it cannot be altered or duplicated.

Easily verifiable – anyone can instantly check the authenticity of a certificate without relying on by issuing.

1.3 MOTIVATION

The increasing number of fake academic and professional certificates has become a significant concern for educational institutions, employers, and governments worldwide. Manual certificate verification is often time-consuming, costly, and error-prone, especially when dealing with a high volume of applicants or students from various institutions. At the same time, organizations are seeking trustworthy digital solutions that ensure authenticity and efficiency. Blockchain technology offers decentralization, transparency, immutability, and security—making it an ideal tool for solving these challenges. Additionally, with the global shift toward digitization and automation, there is a growing demand for secure systems that support real-time and cross-border verification of credentials. This project is motivated by the need to:

1. Eliminate certificate fraud.
2. Build trust in digital credentials.
3. Modernize traditional systems through emerging technologies like blockchain.
4. Empower individuals to own and share their credentials independently and securely.

1.4 SCOPE

Technical Scope:

- Development of a smart contract to securely issue and store certificate data on a blockchain (e.g., Ethereum, Hyperledger).
- Creation of a web or mobile interface for:
    - Institutions to issue certificates.
    - Users (students/employees) to access their credentials.
    - Third parties (employers/organizations) to verify them instantly.

Application Scope:

- Educational institutions: Schools, colleges, universities, and online learning platforms can issue blockchain-based degrees, diplomas, and transcripts.
- Corporate organizations: Issue verifiable training completion certificates or compliance credentials.
- Government agencies: Digitally issue licenses, identity proofs, and eligibility certificates.
- Employers and recruiters: Use the system to verify candidate credentials instantly and securely.

Scalability:

- The system can be scaled to support multiple institutions and multi-chain environments.
- Future integration with national education portals or global verification networks is possible.

1.5 OBJECTIVES

To design a secure system for issuing digital certificates using blockchain technology that ensures authenticity and integrity. To prevent certificate forgery by leveraging blockchain's immutable and tamper-proof nature. To create a decentralized verification mechanism that allows any third party (e.g., employers, institutions) to validate certificates in real time without relying on the issuing authority. To develop smart contracts that automate certificate generation, storage, and access, reducing manual processes and human errors. To build a user-friendly interface for:

Institutions to issue certificates.

Students or professionals to view/download their credentials.

Verifiers to check certificate authenticity via unique ID or QR code.

To ensure transparency and traceability, enabling all stakeholders to track the certificate issuance and validation process. To reduce the time, cost, and administrative burden associated with traditional certificate verification processes. To promote the adoption of emerging technologies (blockchain, IPFS, smart contracts) in the education and employment sectors.

1.6 LIMITATIONS

1. **Technical Complexity:** Implementing blockchain requires specialized knowledge in smart contracts, cryptography, and distributed systems, which may be a barrier for smaller institutions.
2. **Initial Setup Costs**: Deployment on public blockchains can incur high gas fees and infrastructure costs. Private blockchains require server maintenance and configuration.
3. **Scalability Issues**: Public blockchain networks (e.g., Ethereum) may face latency and congestion during high traffic, affecting certificate issuance speed.
4. **Energy Consumption:** Some blockchain platforms (like those using Proof-of-Work) consume significant energy, which is not environmentally sustainable.
5. **Legal and Regulatory Uncertainty:** Any countries have not yet defined legal frameworks for blockchain-based documents, which may affect acceptance and trust.
6. **Data Privacy Concerns:** Storing personal or sensitive data on a public ledger can raise GDPR and privacy compliance issues unless off-chain or encrypted solutions are used.
7. **Dependency on Internet Access**: All users (issuers, students, verifiers) must have reliable internet access to interact with the blockchain, limiting use in remote areas.

CHAPTER - 2

Literature Survey

2.1 PREFERRED LANGUAGE

In blockchain-based certificate verification systems, selecting the right programming languages for both the backend and frontend is critical to ensure efficient, secure, and scalable application development. These systems aim to issue, store, and verify academic or professional certificates on a blockchain network in a tamper-proof manner. The backend and frontend layers are responsible for managing user interactions, business logic, and connectivity with the blockchain, and each layer requires specific languages and tools best suited to its tasks.

On the backend, the primary responsibility is to manage server-side operations such as receiving data from the user, generating cryptographic hashes of certificates, interacting with smart contracts, storing metadata in databases, and interfacing with blockchain nodes. One of the most popular backend languages used in blockchain certificate systems is JavaScript, particularly with the Node.js runtime. Node.js allows for asynchronous event handling, which is useful when dealing with blockchain networks that respond with delays due to block confirmations. JavaScript is often used with libraries like web3.js or ethers.js to interact with Ethereum smart contracts for certificate issuance and verification. Another widely used language is Python, which provides a clean and readable syntax and powerful cryptographic and data-processing libraries. Python, along with web3.py, is frequently used for generating hashes, integrating with IPFS (Inter-Planetary File System) for off-chain storage, and automating certificate-related tasks.

Algorithm	Hash Size	Message size	Block size	Word size	No. of steps
MD5	128	$<2^{64}$	5	3	64
SHA-1	160	$<2^{64}$	5	3	80
SHA-256	256	$<2^{64}$	5	3	64
SHA-384	384	$<2^{128}$	1024	6	80
SHA-512	512	$<2^{128}$	1024	6	80

Table 2.3. Comparison of various hashing mechanism

Table 2.1: comparison of various hashing mechanism

2.2 EXISTING SYSTEM

With the increasing demand for secure and tamper-proof credentials, several institutions and organizations around the world have implemented blockchain-based systems for certificate issuance and verification. Traditional systems rely heavily on centralized databases and manual processes for validating academic and professional certificates, which are often vulnerable to forgery, loss, and long verification delays. In contrast, blockchain technology offers decentralized, transparent, and immutable solutions, and several real-world applications and platforms have already demonstrated its potential.

One of the earliest and most well-known blockchain-based certificate verification systems is MIT's Block-certs. Developed by the Massachusetts Institute of Technology in collaboration with Learning Machine (now part of Hyland), Block-certs is an open standard for creating, issuing, viewing, and verifying blockchain-based certificates. Block-certs uses public blockchains like Bitcoin and Ethereum to anchor certificate data. Each certificate is issued with a unique hash that is permanently stored on the blockchain, making it easy for employers or institutions to verify its authenticity without needing to contact the issuing authority directly.

Another example is the University of Bahrain, which uses the Block-certs platform to issue digital diplomas on the blockchain. This move has allowed students to own and share their credentials easily and securely. Similarly, the National University of Singapore (NUS) and the Indian Institute of Technology (IIT) have piloted blockchain-based certificate solutions to improve trust and transparency in academic verification.

Private companies have also contributed to this space. TrueProfile.io, for example, uses blockchain to help professionals secure and verify their credentials, particularly in the healthcare sector. APPII is another blockchain-powered platform that enables individuals to build verified digital CVs using their education and career history, which is authenticated on the blockchain.

Governments have shown interest as well. The Government of Malta has implemented blockchain technology to issue educational certificates, while the Government of Andhra Pradesh in India has explored using blockchain for storing school and college records securely.

These systems typically use a combination of blockchain networks (public like Ethereum or private like Hyperledger), smart contracts for certificate lifecycle management, and decentralized storage systems such as IPFS to store actual certificate files. The result is a secure, efficient, and tamper-proof method of issuing and verifying certificates that reduces dependency on intermediaries, minimizes fraud, and speeds up the credential validation process**.**

2.3 LIMIATTIONS USED IN EXISTIONG SYSTEM

While blockchain technology offers numerous advantages for certificate verification, such as decentralization, security, and transparency, it also comes with several limitations that can hinder widespread adoption and implementation. One of the primary challenges is scalability. Public blockchains like Ethereum can experience slow transaction speeds and high costly. Additionally, storing complete certificate files directly on the blockchain
is impractical due to limited storage capacity and high data costs, necessitating the use of off-chain storage systems like IPFS, which introduces another layer of complexity.
Another limitation is technical complexity. Developing and maintaining blockchain-based systems requires specialized knowledge in smart contract development, cryptographic algorithms, and distributed ledger technologies. Most educational institutions or organizations may lack the technical expertise or financial resources to implement such systems effectively. Interoperability is also a concern, as different platforms may use different blockchain networks and data formats, making it difficult to integrate or transfer certificates across institutions or systems.

Summary of Features of top 5 Blockchain Platforms for Enterprises

	Ethereum	Hyperledger Fabric	R3 Corda	Ripple	Quorum
Industry-focus	Cross-industry	Cross-industry	Financial Services	Financial Services	Cross-industry
Governance	Ethereum developers	Linux Foundation	R3 Consortium	Ripple Labs	Ethereum developers & JP Morgan Chase
Ledger type	Permissionless	Permissioned	Permissioned	Permissioned	Permissioned
Cryptocurrency	Ether (ETH)	None	None	Ripple (XRP)	None
% providers with experience[1]	93%	93%	60%	33%	27%
% share of engagements[2]	52%	12%	13%	4%	10%
Coin Market Cap[3]	$91.5 B (18%)	Not applicable	Not Applicable	$43.9 B (9%)	Not Applicable
Consensus algorithm	Proof of Work (PoW)	Pluggable framework	Pluggable framework	Probabilistic voting	Majority voting
Smart contract functionality	Yes	Yes	Yes	No	Yes

1. *Based on responses from 15 leading blockchain service providers*
2. *Based on a random sample of set of 50 enterprise blockchain engagements across multiple industries*
3. *Coinmarketcap.com as of Feb 20, 2018, 6:20 PM UTC*

Source: HfS Research, 2018

HfS

Table 2.2: Summary Features of top5 blockchain platforms

Privacy concerns also arise, especially when handling sensitive user data. Although blockchain is secure, its immutable nature means that any incorrect or private data once uploaded cannot be modified or deleted, which may conflict with data protection regulations like GDPR. Furthermore, the user experience in blockchain applications is not always user-friendly; requiring digital wallets, handling cryptographic keys, or interacting with smart contracts can be intimidating for non-technical users.

Finally, legal and regulatory uncertainty poses a challenge. Many countries have yet to develop clear legal frameworks governing blockchain-based credentials, which affects their official recognition and validation. These limitations highlight the need for thoughtful implementation, improved tools, and supportive policies before blockchain can be fully adopted for certificate management at scale.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The proposed system introduces a blockchain-based approach for issuing and verifying academic and professional certificates, addressing the limitations of traditional certificate management systems. In conventional systems, certificates are stored in centralized databases or issued in physical form, making them vulnerable to tampering, forgery, loss, and time-consuming verification procedures. The proposed solution leverages the decentralized, transparent, and immutable nature of blockchain technology to enhance the security, reliability, and efficiency of certificate verification.

In this system, when an institution such as a university or training provider issues a certificate, it first generates a digital version of the certificate containing key data such as the recipient's name, course details, date of issue, and issuer's name. A cryptographic hash (unique fingerprint) of the certificate is then created and stored on the blockchain using a smart contract. This smart contract ensures that the certificate hash is recorded immutably and can be accessed for future verification. The full certificate file, due to its size, is stored off-chain using a decentralized file storage system like IPFS (Inter Planetary File System), which returns a content identifier (CID). The CID or file reference is also linked to the blockchain record.

Only authorized institutions are allowed to issue certificates by interacting with the smart contract, which can be controlled through role-based access or permissioned blockchain networks. Each issued certificate is associated with a unique certificate ID or transaction hash, which can be shared with the certificate holder. This ID can be embedded into a QR code printed on the certificate or shared as a link for easy access.

For verification, any employer, institution, or user can upload the certificate or enter its details on a dedicated web portal. The system then re-generates the hash from the uploaded certificate data and checks it against the hash stored on the blockchain. If the hashes match, the certificate is verified as genuine; otherwise, it is flagged as invalid or altered. Since blockchain data is immutable, once the certificate is registered, it cannot be modified, ensuring high data integrity.

The user interface is built using modern web technologies such as HTML, CSS, JavaScript, and React.js. For blockchain interaction, libraries such as web3.js or ethers.js are used. Wallet integration (e.g., MetaMask) is included to allow certificate issuers to securely sign transactions and interact with smart contracts. The backend is developed using Node.js or Python, which manages API calls, blockchain interactions, and file handling for IPFS.

This system drastically reduces manual work and delays involved in certificate validation and builds trust by making the verification process transparent and accessible worldwide. It eliminates the need for third-party verification services and enables users to control and share their credentials securely. The proposed system is scalable, tamper-proof, cost-efficient in the long run, and provides a reliable framework for digital credential management in academic, corporate, and governmental sectors.

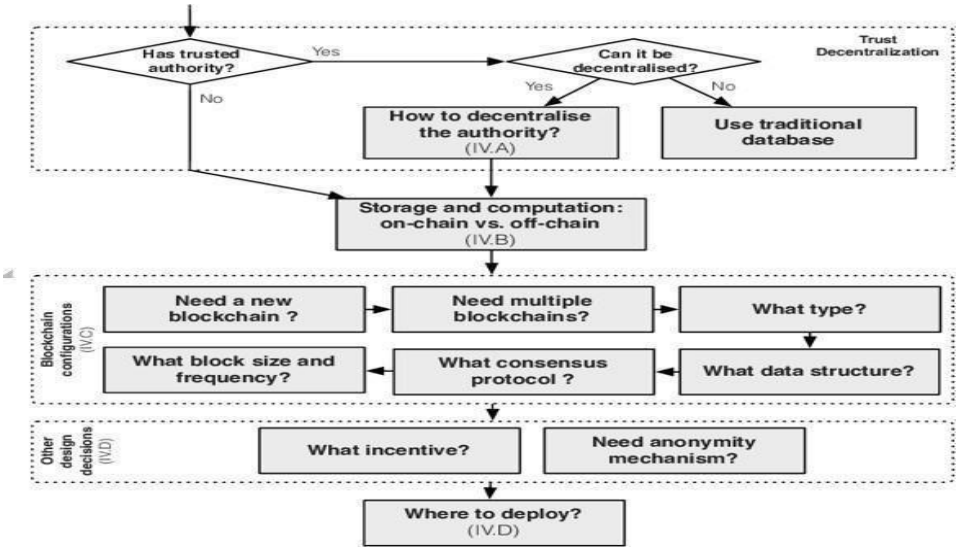


Fig 3.1: Design Process for Blockchain System

3.2 system requirements

The system requirements specify the hardware specify the certificate validation using blockchain technology efficiently:

3.2.1 software requirements

Component	Specification
Operating System	Windows 10 or higher / Linux / macOS
IDLE python	Python 3.11
Spring Boot Framework	Version 2.5+
IDE / Code Editor	IntelliJ IDEA / Eclipse / VS Code
Database Management System	MySQL 8.0 / PostgreSQL / H2 (for testing)
Frontend Technologies	HTML5, CSS3
Web Browser	Latest versions of Chrome / Firefox / Edge
Build Tool	Maven / Gradle
Version Control	Git
Postman / REST Client	Latest version

Table 3.2.1: Software Requirements of the Proposed System

3.2.2 Hardware requirements

Component	**Minimum Requirement**
Processor	Intel Core i3 or equivalent and above
RAM	Minimum 4 GB (8 GB recommended)
Storage	Minimum 100 GB HDD or SSD
Display	1366 x 768 resolution or higher
Network	Broadband Internet connection

Table 3.2.2: Hardware Requirements of the Proposed System

3.2.2 functional requirements

1. **User Registration and Authentication**: The system shall allow institutions to register and authenticate securely before issuing certificates.
2. **Certificate Issuance**: The system shall enable authorized institutions to issue digital certificates to recipients. A unique certificate ID and its hash shall be generated and stored on the blockchain.
3. **Certificate Storage**: The system shall store the certificate file (PDF/JSON) off-chain using a secure storage solution such as IPFS or a cloud server. Metadata like recipient name, course, and issue date shall be stored in a local database.
4. **Blockchain Record Creation**: The system shall record certificate hash, issuer address, issue date, and revocation status on the blockchain using smart contracts.
5. **Certificate Verification**: The system shall allow any third party (e.g., employers) to verify a certificate by entering its ID or scanning a QR code. Verification shall involve matching the certificate's hash with the blockchain record.
6. **Certificate Revocation**: The system shall allow issuers to revoke certificates and update their status on the blockchain. The revocation date shall be recorded and displayed during verification.
7. **Certificate Status Check**: The system shall display whether a certificate is valid or revoked based on the blockchain data.
8. **Smart Contract Execution**: The system shall use smart contracts to handle issuance, verification, and revocation logic in a decentralized and secure manner.
9. **Audit Trail & Transparency**: The system shall maintain an immutable and transparent history of all certificate transactions on the blockchain.
10. **Search and Filter**: The system shall allow administrators to search and filter issued certificates by name, course, or status.

3.2.4 non-functional requirements

1. **Security**: The system shall ensure the integrity of certificates using cryptographic hashing and blockchain immutability. User data and certificates stored off-chain shall be protected through encryption and secure access control.
2. **Privacy**: Personal information (e.g., student names, IDs) shall not be stored on the blockchain to comply with data protection standards (e.g., GDPR). Only non-sensitive metadata and hashes shall be stored on-chain.
3. **Performance**: Certificate verification shall be completed within a few seconds, even under concurrent requests. Smart contract execution and hash comparison shall be optimized for speed and low gas usage (in public chains).
4. **Scalability**: The system shall be able to handle increasing numbers of certificates and users without performance degradation. Off-chain storage and indexing mechanisms shall support efficient data retrieval.
5. **Availability**: The system shall be accessible 24/7 with minimal downtime, ensuring users can verify certificates at any time.
6. **Usability**: The system shall provide a user-friendly web interface for certificate issuance, verification, and revocation. Verification should require minimal input, such as scanning a QR code or entering a certificate ID.
7. **Interoperability**: The system shall use standard data formats (e.g., JSON, PDF) to allow integration with other academic or professional verification systems.
8. **Maintainability**: The software shall be modular, allowing future updates to smart contracts, APIs, and user interfaces with minimal disruption.
9. **Reliability**: The system shall ensure reliable operation and prevent data loss through regular backups of off-chain data and redundant node configuration.
10. **Transparency**: All certificate-related events (issuance, revocation) recorded on the blockchain shall be publicly verifiable and tamper-proof.

3.3 concepts used in proposed system

3.4 Block chain Technology: Blockchain is a decentralized and immutable ledger system that ensures data integrity and transparency. In the proposed system, blockchain is used to store certificate hashes and verification metadata, making the certificates tamper-proof and publicly verifiable.

1. **Smart Contracts**: Smart contracts are self-executing programs deployed on the blockchain. They automate the issuance, verification, and revocation of certificates. Once deployed, they operate transparently and cannot be altered, ensuring trust in the process.
2. **Cryptographic Hashing (SHA-256)**: Each certificate file is hashed using a secure hashing algorithm (like SHA-256). The hash is stored on the blockchain and used during verification to ensure that the certificate has not been altered.
3. **Digital Signature / Public Key Infrastructure (PKI)**: Issuers sign certificates using their private keys. Verifiers can confirm authenticity using the issuer's public key, ensuring the certificate comes from a trusted source.
4. **Decentralized Storage (e.g., IPFS)**: To avoid storing bulky files on-chain, the actual certificate documents (PDF or JSON) are stored off-chain using decentralized storage like IPFS (Inter- Planetary File System), which provides a content-addressable, distributed way to store files securely.
5. **Certificate Revocation Mechanism**: The system includes a revocation function within smart contracts, allowing issuers to mark a certificate as revoked. This status is recorded on the blockchain and displayed during verification.
6. **QR Code Integration**: Each issued certificate contains a QR code that encodes the certificate ID or blockchain verification URL. This allows quick and easy verification by scanning.
7. **Web-Based Frontend Interface**: A user-friendly web application is used by issuers to upload and issue certificates, and by verifiers to validate them. This interface communicates with blockchain smart contracts and the off-chain database via APIs.

3.4 data sets used in proposed system

In the proposed certificate validation system using blockchain technology, the dataset plays a crucial role in managing both on-chain and off-chain information related to issued certificates. The core dataset consists of structured records containing fields such as certificate ID, recipient name, recipient ID (e.g., email or student number), course or program title, issue date, and certificate file path. A cryptographic hash of the certificate file is generated (using algorithms like SHA-256) and stored on the blockchain to ensure authenticity and integrity. This hash acts as a unique digital fingerprint for each certificate.

The dataset also includes the issuer's blockchain address, which serves as a trusted identifier, and a revocation status field to indicate whether the certificate remains valid. Additional metadata like revocation date, certificate expiry date (if applicable), and issuer institution name are stored in a secure off-chain database or decentralized storage such as IPFS. By separating sensitive or large data (like recipient details and actual certificate files) from critical verification data (like hashes and statuses), the system maintains both privacy and performance.

CHAPTER 4

SYSTEM DESIGN

4.1 COMPONENT IN USER/USED IN THE PROPOSED SYSTEM

The proposed system architecture for blockchain-based certificate verification is designed to provide a secure, transparent, and decentralized platform that can be used by different types of users, including institutions, students, employers, and verifiers. The system eliminates the problems of certificate forgery, slow manual verification, and centralized data control by leveraging blockchain's key features—immutability, decentralization, and cryptographic security. The architecture consists of multiple layers and user roles that work together to create a seamless, tamper-proof certificate issuance and verification environment.

Users in the Proposed System

Issuing Authority: These are the verified organizations, such as universities, training centers, or certifying agencies, responsible for issuing digital certificates. They log into the platform, input student or employee details, generate the digital certificate, and submit it to the blockchain using a connected wallet like MetaMask. They are the only users authorized to execute the smart contract functions that store certificate data.

certificate holder student /professor: The individual who receives the certificate from the issuing authority. They can download the certificate, access the verification link or QR code, and share it with potential employers or academic institutions. They do not alter or upload the certificate themselves, but they own the credential.

1. **Verifier**

   This role is played by organizations or individuals who need to verify the authenticity of a certificate. They use the platform to enter certificate details or scan a QR code. The system checks the certificate's cryptographic hash stored on the blockchain and returns the verification result instantly.

2. **System administrator**

   Responsible for managing platform access, onboarding new issuing institutions, and maintaining smart contract deployment and system updates. This role ensures smooth operation and governance of the system.

3. **Proposed System Architecture Overview**

   The architecture is composed of four layers, each with defined responsibilities and tightly integrated workflows:

4. **Layer (Front End API)**

Built using HTML, CSS, JavaScript, and frameworks like React.js, this layer allows users to interact with system. It includes:

a. Certificate issuance forms (for institutions)
b. Verification tools (for employers or universities)
c. Wallet integration (e.g., MetaMask) for secure smart contract interaction
d. User-friendly dashboards for different roles

5. **Application Layer (API)**

The backend, typically developed in Node.js or Python, handles:

a. Hash generation of certificate data (e.g., SHA-256)
b. Communication with IPFS for file storage and retrieval
c. Blockchain interaction via Web3.js or Ethers.js
d. RESTful APIs that connect the frontend with the blockchain and storage layers

6. **Blockchain-Layer**

This layer acts as the immutable ledger of trust. It stores:

a. Certificate hashes and metadata (issuer, date, etc.)
b. Smart contracts that validate and store the issued credentials
c. Permission logic to allow only approved issuers to create entries

7. **Off Chain Storage Layer**

Since storing large files on the blockchain is costly and inefficient, the actual certificate files are stored in a decentralized file system like IPFS. The system records the CID (Content Identifier) of each file on the blockchain so it can be retrieved when needed.

4.2 PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture for certificate verification using blockchain technology is designed to ensure a decentralized, secure, transparent, and scalable platform for issuing and verifying digital certificates. This architecture replaces traditional, centralized verification systems with a robust framework that leverages the power of blockchain, smart contracts, cryptographic hashing, and decentralized storage. The architecture is modular and layered, supporting seamless interaction between users and the underlying blockchain network while ensuring data integrity and Privacy.

The architecture is divided into four main layers: the Frontend (Presentation Layer), the Backend (Application Layer), the Blockchain Layer, and the Off-Chain Storage Layer. These layers work in unison to provide a complete solution for institutions issuing certificates and for verifiers checking their authenticity.

1. **Frontend Layer (User- Interface)**

   This is the interactive part of the system that users—such as institutions, students, and verifiers—engage with. Built using modern web technologies like HTML, CSS, JavaScript, and React.js, this layer includes features such as certificate issuance forms, verification portals, and QR code generators. For security, wallet integration (e.g., MetaMask) is used to allow issuers to authenticate and authorize transactions securely on the blockchain. This layer ensures user-friendly access to blockchain functionality.

2. **Back End Layer (Application Layer)**

   The backend is developed using technologies such as Node.js or Python and serves as the middleware connecting the frontend to the blockchain and storage systems. It processes certificate issuance requests, validates input, generates cryptographic hashes (e.g., using SHA-256), and interacts with the blockchain through libraries like Web3.js or Ethers.js. The backend also handles file uploads to off-chain storage (such as IPFS) and returns the content identifier (CID), which is recorded on the blockchain along with the certificate metadata.

3. **Block chain Layer**

   This is the core layer where the actual verification data is stored securely and immutably. Smart contracts, written in Solidity (for Ethereum) or Go (for Hyperledger Fabric), are deployed on the blockchain to manage the certificate lifecycle. These smart contracts store certificate metadata, issuer information, timestamps, and the hash of the original certificate file. They also ensure that only authorized institutions can issue certificates and that data cannot be altered once submitted, preserving the integrity of the system.

4. **Off Chain Storage Layer**

   (IPFS)Since blockchains are not optimized for storing large files, the actual certificate documents (such as PDFs or images) are stored in a decentralized storage network like IPFS (Inter-Planetary File System). When a certificate is uploaded, IPFS generates a unique CID for the file, which is stored on the blockchain alongside the hash of the certificate content. This enables easy and reliable access to the original file during verification, without overloading the blockchain.

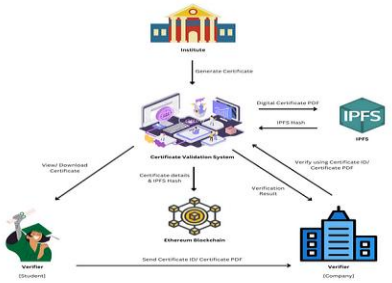


Fig 4.1 Certificate Validation System Overview

4.3 UML DIAGRAM

4.3.1 use case diagram

A use case diagram for a blockchain-based certificate verification system illustrates interactions between users and the system. Key actors include Admin, Student, Employer/Verifier, and Blockchain Network. Admin issues and manages certificates, students view and share them, and verifiers check authenticity. The blockchain securely stores certificate data using cryptographic hashes, ensuring integrity and trust. This diagram helps visualize system functionality, user roles, and the secure certificate validation process in a decentralized environment.

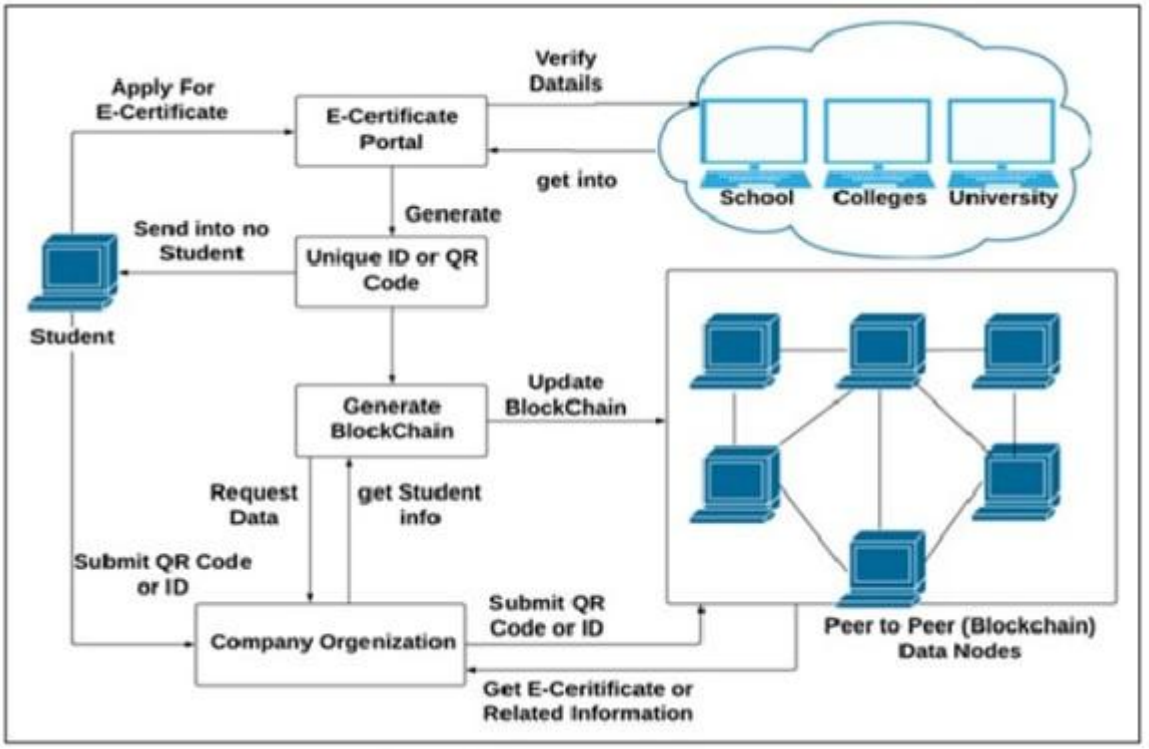


Fig 4.2: Use case diagram

4.3.2 Sequence Diagram

A sequence diagram illustrates the flow of messages or interactions between various system components and users over time. In a Blockchain-based Certificate Verification System, the process begins with the admin issuing a certificate. The system generates a cryptographic hash and stores it on the Blockchain. The student accesses the certificate and shares it with an Employer/Verifier. The Verifier submits a verification request, and the system retrieves the hash from the blockchain to validate authenticity. This sequential flow ensures integrity, transparency, and traceability. Sequence diagrams help visualize communication between actors and modules, making them essential for system design and debugging.

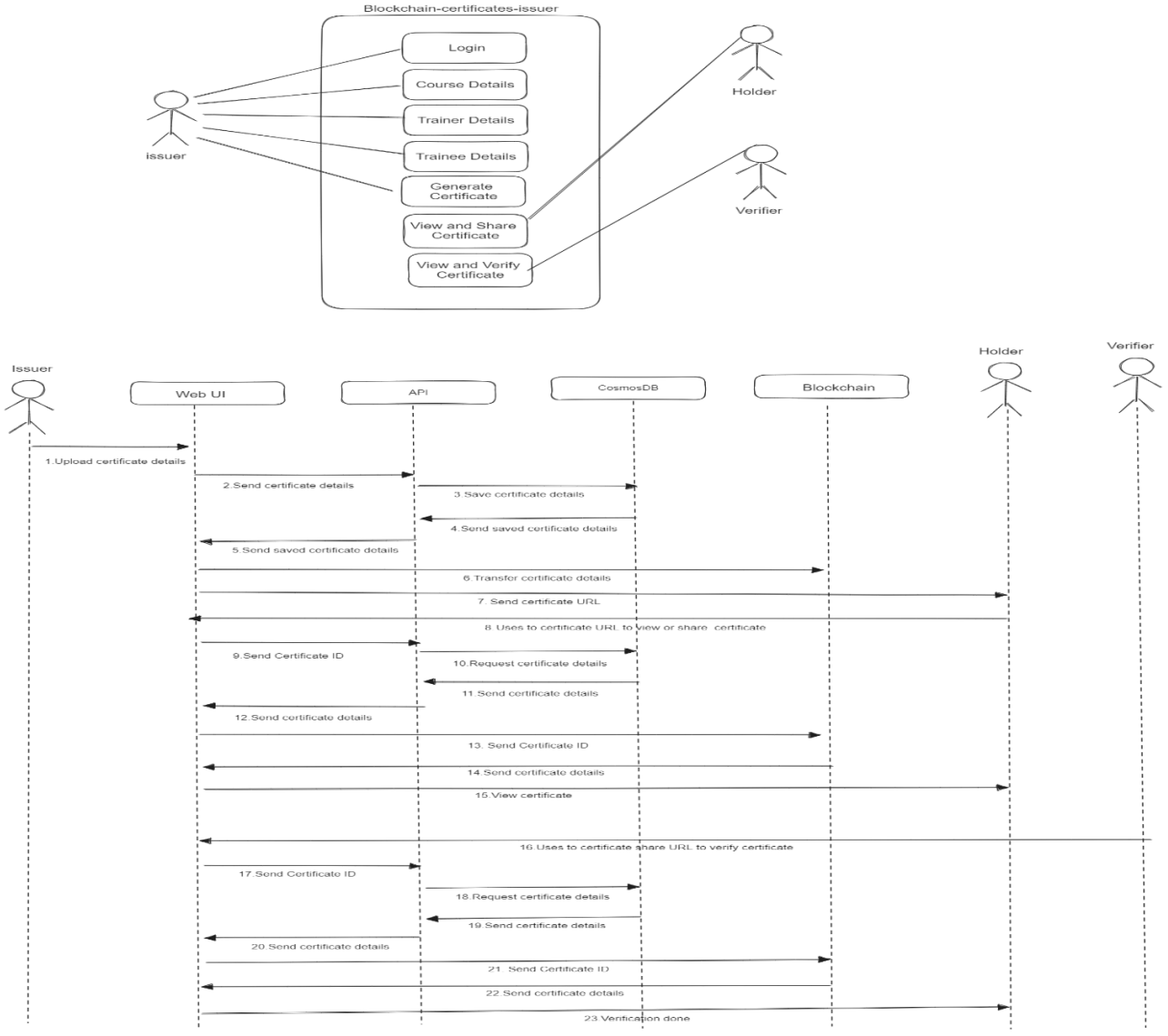


Fig 4.3: Sequence Diagram

4.3.3 Module Diagram

A module diagram theoretically represents the structural organization of a software system by dividing it into interconnected, functional components called modules. Each module performs a specific role within the system. In a Blockchain-based Certificate Verification System, the system is divided into modules such as User Management, Certificate Handling, Blockchain Interface, Verification, and User Interface. These modules interact to ensure smooth and secure operations. For example, the blockchain module handles data immutability, while the certificate module manages issuance and storage. This modular approach enhances system scalability, security, and maintainability by separating responsibilities and simplifying system updates and troubleshooting.

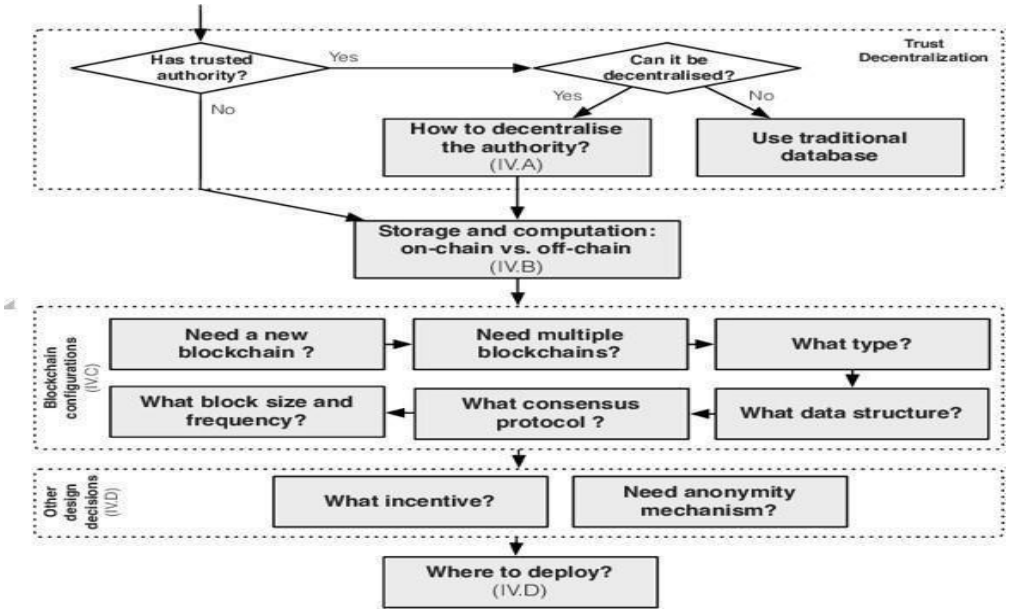
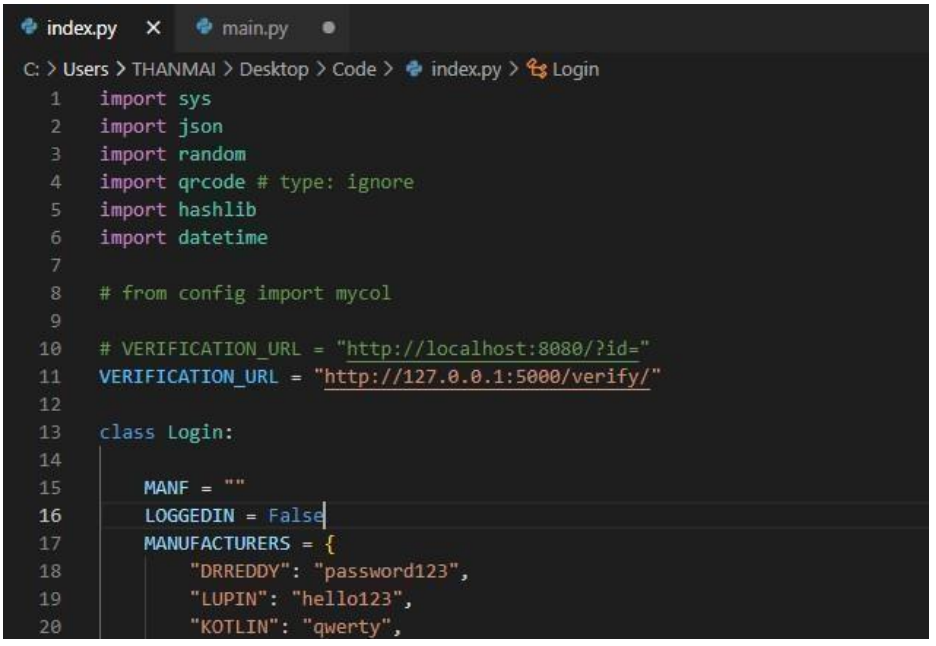


Fig 4.4: Module Diagram

CHAPTER 5
IMPLEMENTATION

SOURCE CODE
INDEX.PY



```
            self.end_date = ""
            self.mark = ""
            self.certfile = ""
            self.personality = ""


        def actions(self):
            choice = input("Enter 1 to ADD item or 2 to Verify BlockChain\n")

            if choice == "1":
                self.department = input("Enter the department:\n")
                self.student_name = input("Enter student name:\n")
                self.academic_year = input("Enter academic year:\n")
                self.regnum = input("Enter reg number:\n")
                self.joining_date= input("Enter joining date:\n")
                self.end_date = input("Enter end date:\n")
                self.mark= input("Enter student mark:\n")
                self.certfile = input("uploed certificate file:\n")
                self.personality= input("Enter student infomation:\n")
                self.newCertificate()

            elif choice == "2":
                if self.isBlockchainValid():
                    sys.exit("BlockChain is valid")
                else:
                    sys.exit("BlockChain is invalid")

            else:
                sys.exit("Logged out successfully")
```

```
    "ADMIN": "qwerty"
}

def main(self):
    loginid = input("Enter your login id:\t")
    password = input("Enter your password:\t")

    if loginid in self.MANUFACTURERS.keys():
        if self.MANUFACTURERS[loginid] == password:
            self.LOGGEDIN = True
            self.MANF = loginid

def isLoggedIn(self):
    if self.LOGGEDIN:
        print("\nWelcome to the blockchain world\n")
    else:
        sys.exit("Please login to experience the blockchain world")

def getManf(self):
    return self.MANF


class BlockChain:

    def __init__(self):
        self.department = ""
        self.student_name = ""
        self.academic_year = ""
        self.regnum = ""
        self.joining_date = ""
```

```
    def newCertificate(self):
        data = {
        "Department": self.department ,
        "Studentname": self.student_name ,
        "AcademicYear": self.academic_year,
        "RegNum": self.regnum,
        "JoiningDate": self.joining_date,
        "EndDate": self.end_date ,
        "Mark": self.mark,
        "CertificateFile": self.certfile ,
        "Personality": self.personality
        }

        proHash = hashlib.sha512(str(data).encode()).hexdigest()
        print(proHash)
        data["hash"] = proHash

        # x = mycol.insert_one(data)

        self.createBlock(data)

        imgName  = self.imgNameFormatting()
        self.createQR(proHash, imgName)

    def addCertificate(
        self,
        department,
        student_name,
        academic_year,
        regnum,
```

```
        joining_date,
        end_date,
        mark,
        certfile,
        personality
    ):
        self.student_name = student_name
        data = {
            "Department":department ,
            "Studentname":student_name ,
            "AcademicYear":academic_year,
            "RegNum":regnum,
            "JoiningDate":joining_date,
            "EndDate":end_date ,
            "Mark":mark,
            "CertificateFile":certfile ,
            "Personality":personality
        }

        proHash = hashlib.sha512(str(data).encode()).hexdigest()
        print(proHash)
        data["hash"] = proHash

        # x = mycol.insert_one(data)

        self.createBlock(data)

        imgName  = self.imgNameFormatting()
        self.createQR(proHash, imgName)
```

```
143       def createBlock(self, data):
145           if self.isBlockchainValid():
146               blocks = []
147               for block in open('./NODES/N1/blockchain.json', 'r'):
148                   blocks.append(block)
149               print(blocks[-1], "jsdata==========")
150
151               preBlock = json.loads(blocks[-1])
152
153               index = preBlock["index"] + 1
154               preHash = hashlib.sha512(str(preBlock).encode()).hexdigest()
155
156           transaction = {
157               'index': index,
158               'proof': random.randint(1, 1000),
159               'previous_hash': preHash,
160               # 'hash': proHash,
161               'timestamp': str(datetime.datetime.now()),
162               'data': str(data),
163           }
164
165           with open("./NODES/N1/blockchain.json", "a") as file:
166               file.write("\n" + json.dumps(transaction))
167           with open("./NODES/N2/blockchain.json", "a") as file:
168               file.write("\n" + json.dumps(transaction))
169           with open("./NODES/N3/blockchain.json", "a") as file:
170               file.write("\n" + json.dumps(transaction))
171           with open("./NODES/N4/blockchain.json", "a") as file:
172               file.write("\n" + json.dumps(transaction))
```

```
  def createQR(self, hashc, imgName):
          img.save("./QRcodes/" + imgName)
          return
      def imgNameFormatting(self):
          dt = str(datetime.datetime.now())
          dt = dt.replace(" ", "_").replace("-", "_").replace(":", "_")
          return self.student_name + "_" + dt + ".png"
      def isBlockchainValid(self):
          with open("./NODES/N1/blockchain.json", "r") as file:
              n1_hash = hashlib.sha512(str(file.read()).encode()).hexdigest()
              print(n1_hash)
          with open("./NODES/N2/blockchain.json", "r") as file:
              n2_hash = hashlib.sha512(str(file.read()).encode()).hexdigest()
              print(n2_hash)
          with open("./NODES/N3/blockchain.json", "r") as file:
              n3_hash = hashlib.sha512(str(file.read()).encode()).hexdigest()
              print(n3_hash)
          with open("./NODES/N4/blockchain.json", "r") as file:
              n4_hash = hashlib.sha512(str(file.read()).encode()).hexdigest()
              print(n4_hash)
          if n1_hash == n2_hash == n3_hash == n4_hash:
              return True
          else:
              return False
  if __name__ == "__main__":
      lof = Login()
      lof.main()
      lof.isLoggedIn()
      LOGGEDINUSER = lof.getManf()
      bc = BlockChain()
      bc.actions()
```

MAIN.PY

index.py 2 ● main.py ●

C: > Users > THANMAI > Desktop > Code > main.py > success

```python
import base64
from io import BytesIO
from flask import Flask, render_template, request, redirect, url_for, session, flash, Response, send_file
from index import BlockChain
import json
from tkinter import *
from werkzeug.wsgi import wrap_file
app = Flask(__name__)
app.secret_key = "alkdjfalkdjf"
@app.route("/")
def welcome():
        return render_template('welcome.html')
@app.route("/home")
def home():
    if session.get("user"):
        return render_template('home.html')
    else:
        flash("Please login to access Verifier")
        return redirect(url_for('login'))
@app.route("/login", methods=["POST", "GET"])
def login():
    if request.method == "POST":
        user = request.form["username"]
        pswd = request.form["password"]
        if user == "Admin":
            if pswd == "password":
                session["user"] = "Admin"
                return redirect(url_for("admin"))
        else:
            flash("Invalid Login details")
            return redirect(url_for('login'))
    else:
```

```python
        return render_template('login.html')
@app.route("/verify/<kid>", methods=["GET"])
def verify(kid):
        return render_template('verify.html', keyId=kid)
def getBlockByKey(key):
    with open('./NODES/N1/blockchain.json', 'r') as bfile:
        n1_data = str(bfile.read())
    with open('./NODES/N2/blockchain.json', 'r') as bfile:
        n2_data = str(bfile.read())
    with open('./NODES/N3/blockchain.json', 'r') as bfile:
        n3_data = str(bfile.read())
    with open('./NODES/N4/blockchain.json', 'r') as bfile:
        n4_data = str(bfile.read())
    pd = str(key)
    if (pd in n1_data) and (pd in n2_data) and (pd in n3_data) and (pd in n4_data):
        with open('./NODES/N1/blockchain.json', 'r') as bfile:
            for x in bfile:
                if pd in x:
                    a = json.loads(x)["data"]
                    b = a.replace("'", "\"")
                    data = json.loads(b)
                    department = data["Department"]
                    student_name = data["Studentname"]
                    academic_year = data["AcademicYear"]
                    regnum = data["RegNum"]
                    joining_date = data["JoiningDate"]
                    end_date = data["EndDate"]
                    mark = data["Mark"]
                    certfile = data["CertificateFile"]
                    personality = data["Personality"]
        return {
```

```
            "department": department,
            "student_name": student_name,
            "academic_year": academic_year,
            "regnum": regnum,
            "joining_date": joining_date,
            "end_date": end_date,
            "mark": mark,
            "certfile": certfile,
            "personality": personality,
        }
app.route("/verify", methods=["POST"])
ef success():
    key = request.form["keyId"]
    block = getBlockByKey(key=key)
    if block:
        return render_template('success.html',keyId=key, department=block["department"], studentname=block["student_name"],
    else:
        return render_template('fraud.html')

app.route("/addcertificate", methods=["POST", "GET"])
ef addcertificate():
    if request.method == "POST":
        file = request.files['certfile']
        if not file.filename.endswith('.pdf'):
            flash("only pdf are supported")
            return redirect(url_for('certificate'))
        department   = request.form["department"]
        studentname  = request.form["studentname"]
        academicyear = request.form["academicyear"]
```

```
        regnum = request.form["regnum"]
        joiningdate = request.form["joiningdate"]
        enddate= request.form["enddate"]
        mark= request.form["mark"]

        certfile = base64.b64encode(file.read()).decode()

        personality = request.form["personality"]
        print(department, studentname, academicyear, regnum, joiningdate, enddate, mark, certfile, personality)
        bc = BlockChain()
        bc.addCertificate(department, studentname, academicyear, regnum, joiningdate, enddate, mark, certfile, personality)

        flash("Certificate added successfully to the Blockchain")
        # return render_template('home.html')
        return redirect(url_for('home'))
    else:
        # return render_template('home.html')
        return redirect(url_for('home'))
app.route("/admin")
ef admin():
    if session["user"] == "Admin":
        return render_template('admin.html')
    else:
        return redirect(url_for('login'))
app.route("/verifyNodes")
ef verifyNodes():
    bc = BlockChain()
    isBV = bc.isBlockchainValid()
    if isBV:
```

CHAPTER -6

IMPLAMENTATION ANRESULT

6.1 RESULTS

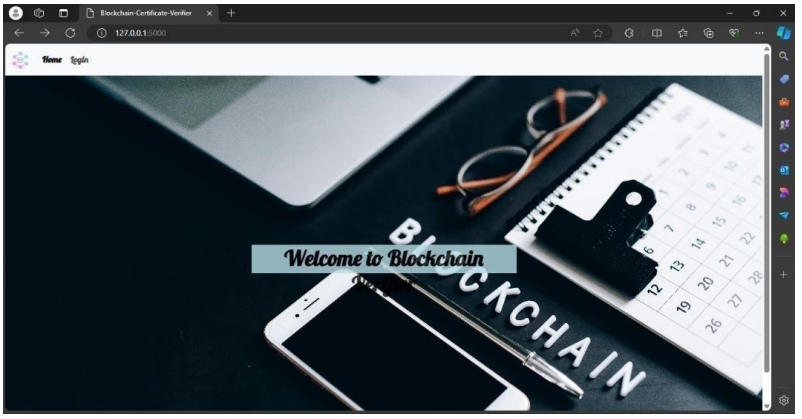


Fig:6.1.1 Welcome page

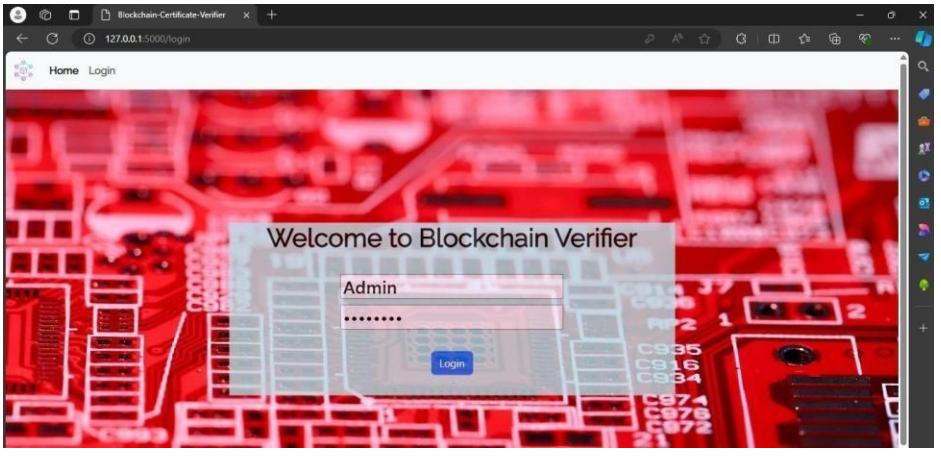


Fig: 6.1.2 login page

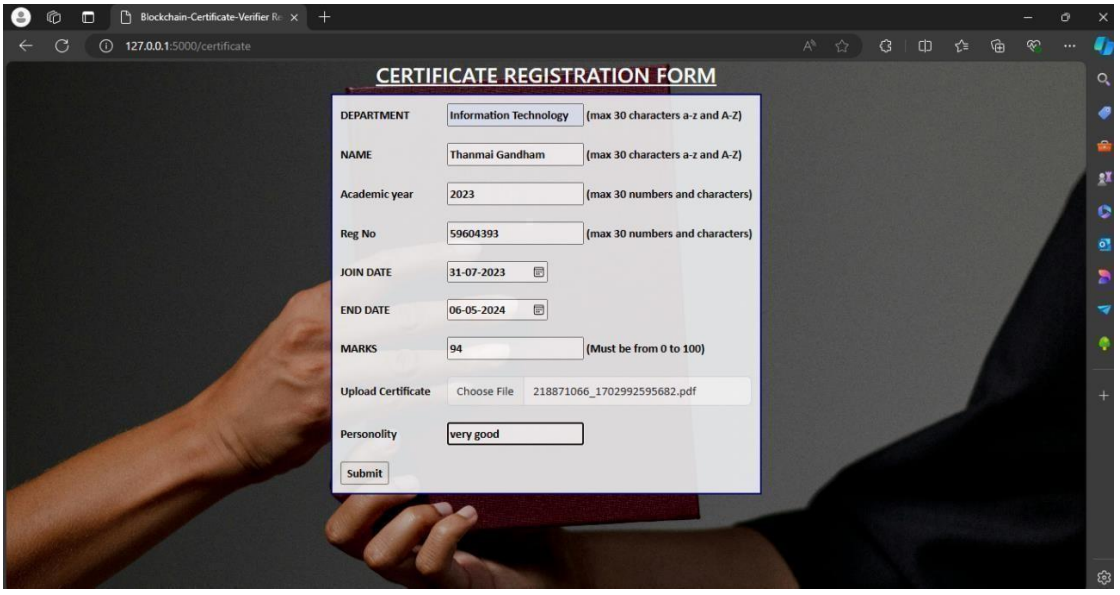


FIG:6.1.3 Verification Form

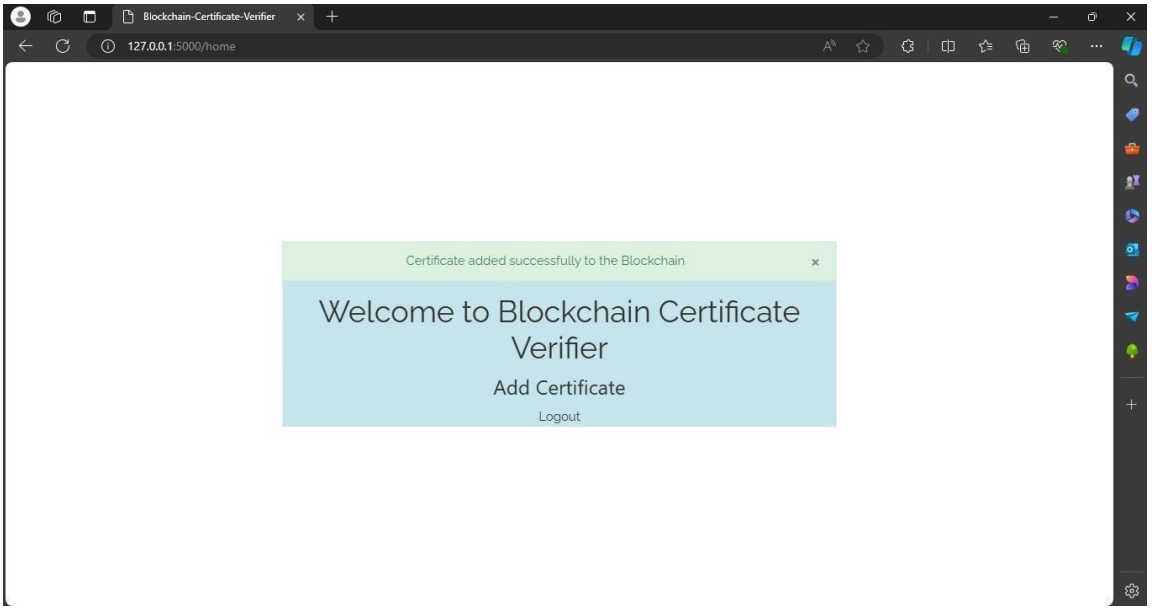


Fig 6.1.4: add certificate

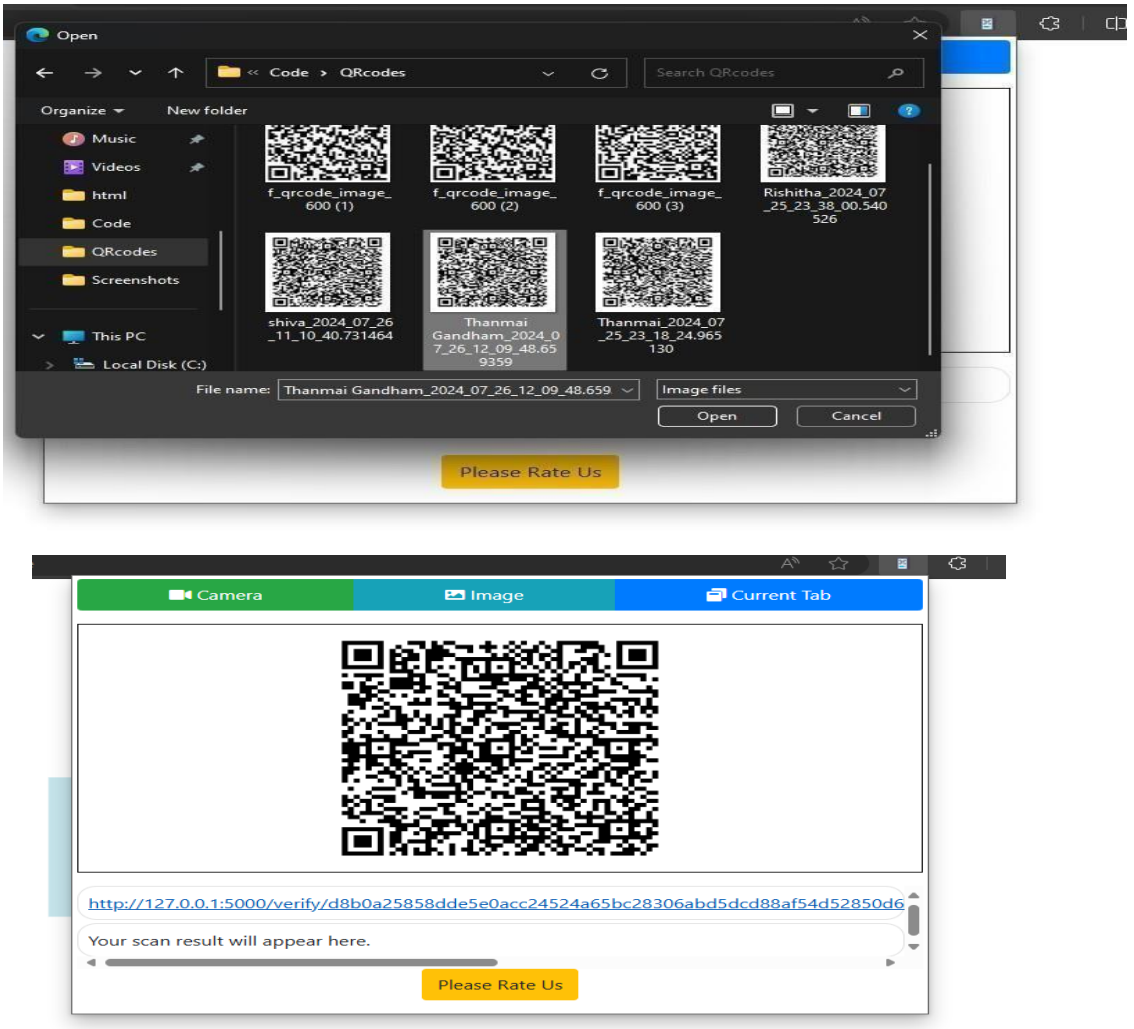


Fig6.1.5VerifyingQRcode

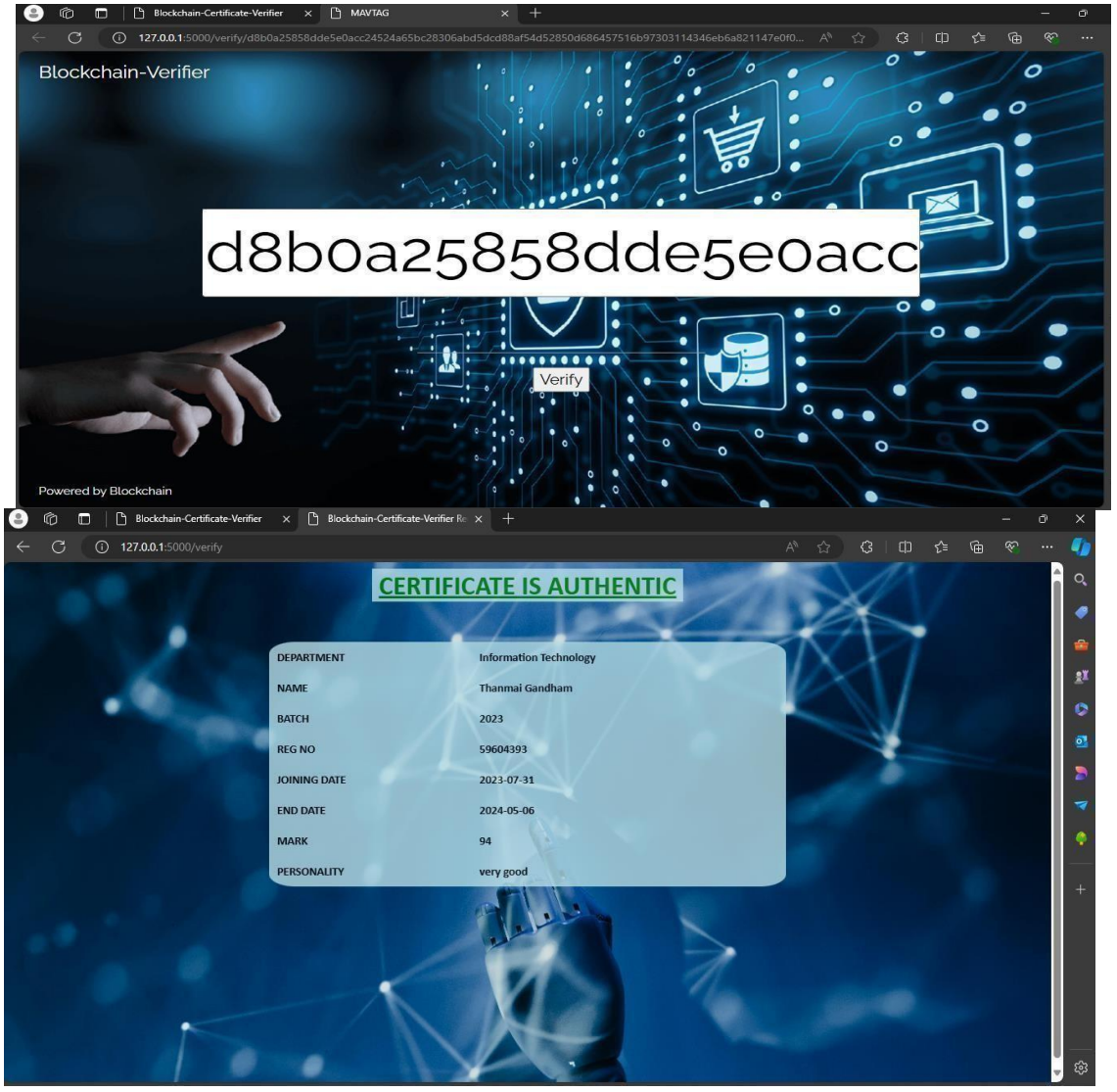


Fig 6.1.6 Authentic Result

WHEN WE UPLOAD FAKE CERTIFICATES:

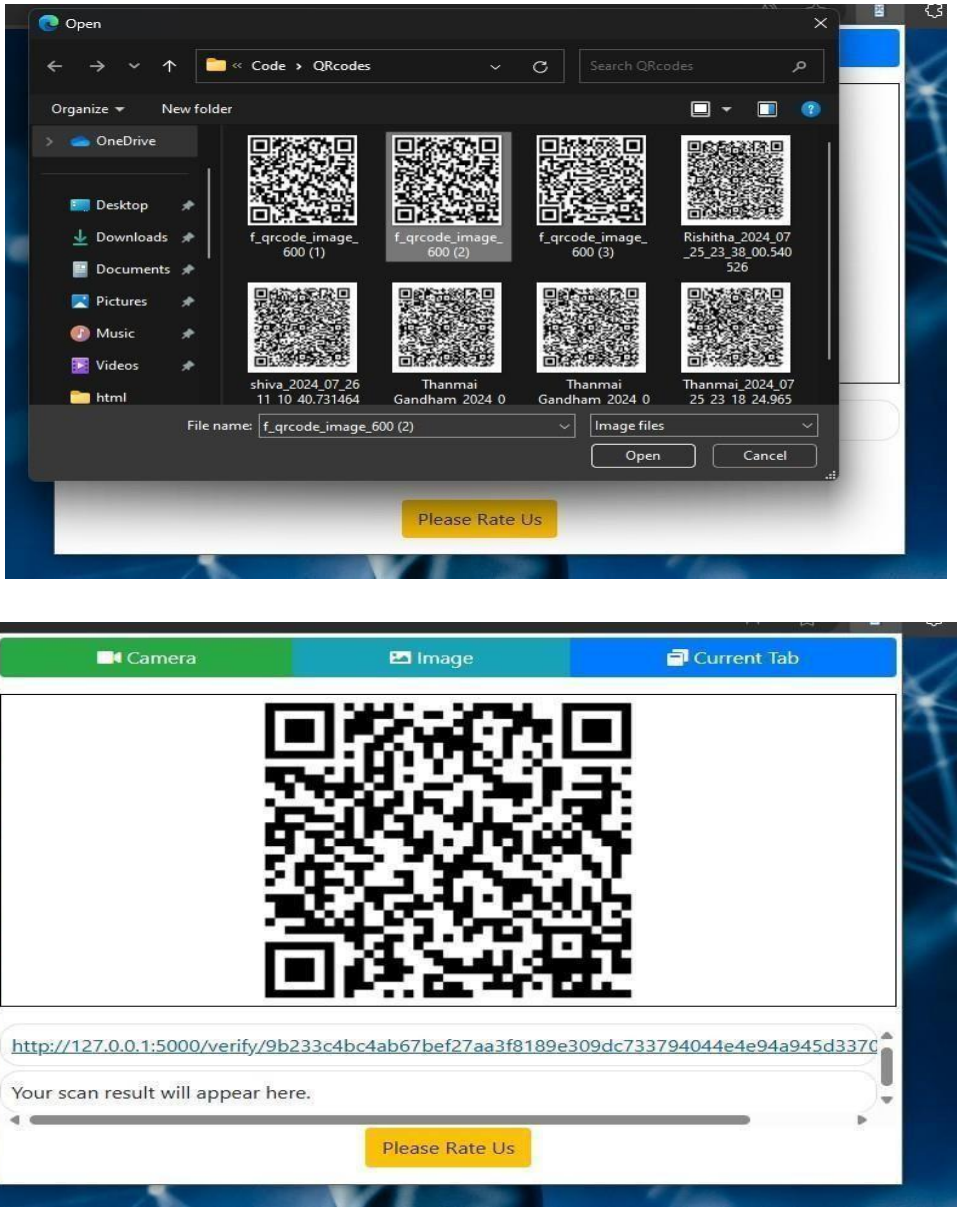


Fig 6.1.7 verifying QR code

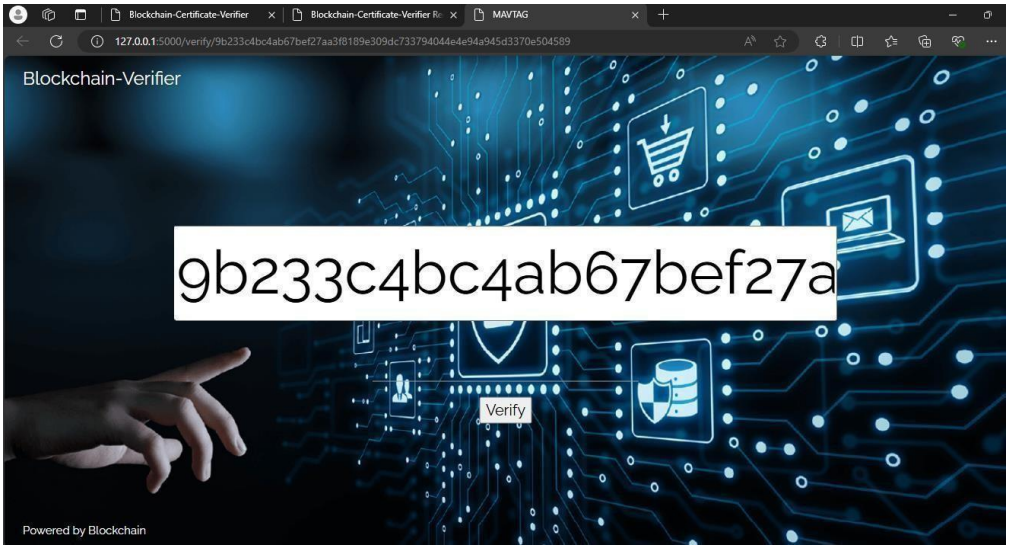


Fig 6.1.8 Verifying URL

Fig 6.1.9 Invalid certificate

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

As of now we are using internet (which is decentralized online platform) to sharing information. But when we transfer money; we are using old-fashioned, centralized financial establishments like banks. In other areas we are also using centralized system to share information (like education- where university has full control).

Blockchain technology provides a way to eliminate this "middleman/central authority. It does this by filling three important roles – recording transactions, establishing identity and establishing contracts. Information security is one of the most important features of Blockchain.

Blockchain can be used to store any type of digital information (e.g. computer code) rather than cryptocurrency usages. Previous work in the field of the blockchain, which is mainly focused on the cryptocurrency and it's mining. In 2017, the blockchain rose to a high level, most of the attention has been on cryptocurrencies such as Bitcoin and Ethereum as investors try to catch the next wave. Now it is going to different sector- Education, Land registry, Banking Share marking….

For truly digitization process in Banking and other sectors, we can use Blockchain technology as a base. It will build trust and provide a way that someone can verify the other person documents in less time and validate the originality.

If we use blockchain in Education/Land Registry/ID card verification/Banking sector, then it will be a "1st step towards corruption free country."

7.2 FUTURE SCOPE:

1.Global Standardization of Credentials:

Blockchain has the potential to become a global standard for issuing and verifying certificates across universities, institutions, and countries.

2. Digital Identity and Credential Wallets:

Individuals may use digital wallets to store all verified academic and professional credentials in one place, enabling easy access and sharing.

3.Integration with Decentralized Identity (DID):

DID frameworks will allow secure, privacy-respecting identity management linked to verified certificates, reducing reliance on centralized databases.

4 Automation through Smart Contracts:

Verification processes will become faster and more reliable through smart contracts that automatically

validate credentials without manual intervention.

4 Enhanced Fraud Detection and Data Integrity:

With real-time blockchain tracking and immutable records, institutions can easily detect and prevent certificate fraud.

5 Government and Policy Adoption:

More governments are likely to support blockchain-based credentialing for public education systems and official documentation.

6 Cross-border Education and Employment:

Blockchain can simplify international admissions, credit transfers, and job applications by providing verifiable, tamper-proof records.

7 Cost and Time Efficiency:

Automated, transparent systems will reduce verification time and administrative costs for both issuers and verifiers.

8 Environmentally Sustainable Blockchains:

Emerging blockchain technologies like Proof of Stake (POS) will offer energy-efficient Solutions making large-scale deployment more feasible.

CHAPTER 8
REFERENCES:

[1] Lyndon Lyons and Andreas Bachmann, "The Blockchain (R)evolution The Swiss Perspective," Switzerland, 2017.

[2] Don Tapscott and Alex Tapscott, "Realizing the Potential of Blockchain-A Multistakeholder Approach to the Stewardship of Blockchain and Cryptocurrencies," in World Economic Forum, 2017.

[3] Alex Tapscott, BLOCKCHAIN REVOLUTION: Understanding the 2nd Generation of The Internet and the New Economy, 2017.

[4] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, White Paper.

[5] George F. Hurlburt and Irena Bojanova "Bitcoin: Benefit or Curse?" in IEEE, 2014.

[6] Nicola Dimitri, The Blockchain Technology: Some Theory and Applications, 2017, MSM-Working Paper No. 2017/03.

[7] Deokyoon Ko, Sujin Choi, Sooyong Park, Kari Smolander Jesse Yli-Huumo, "Where Is Current Research on Blockchain Technology—A Systematic Review," October 2016

[8] Nirmala Singh and Sachchidanand Singh, "Blockchain: Future of financial and cyber security," in IEEE, Noida, 2016.

[9] Engin Zeydan and Suayb Sb Arslan Gültekin Berahan Mermer, "An overview of blockchain technologies: Principles, opportunities and challenges," in IEEE, Turkey, 2018.

[10] Narn-Yih Lee, Chien Chi and Yi-Hua Chen Jin-Chiou Cheng, "Blockchain and smart contract for digital certificate," in IEEE, Japan, 2018.

[11] Henrique Roch, Marcus Denker and Stephane Ducasse Santiago Bragagnolo, "Smart-Inspect: solidity smart contract inspector," in IEEE, Italy, p. 2018.

[12] GWYN D'MELLO. (2017, Dec.) https://www.indiatimes.com/technology/news. [Online].https://www.indiatimes.com/technology/news/bitcoin-miners-are-using-more electricity-than-

Ireland- other-159-countries-no-kidding-335114.html

[13] Abdul Wadud Chowdhury. (2017, Nov.) https://medium.com. [Online].

https://medium.com/oceanize-geeks/blockchain-and-the-future-of-digital-trust-354acc279acc

Abstract Proforma

FBP / IOMP / MAJOR PROJECT

Year & Branch:	**Section:**	**Batch No.:**
Academic Year:		**Regulation:**
Student Registration Details	Name	Roll Number
	1. KALASHETTY LIKHITHA 2. GUGULOTHU SHILPA	227Y1A6220 227Y1A6244
Name of the Guide & Designation	Mrs. Bhavya Varma	
Area (Domain) of the Project	INDUSTRY ORIENTED MINI PROJECT	
Title of the Project	CERTIFICATE VALIDATION USING BLOCKCHAIN TECHNOLOGY	
Tools Required	Python, CSS, Html	

Abstract

- **Background/Introduction:** Provide context or background information on why the project is important.
- **Objectives:** State the main goals or aims of your project.
- **Methodology:** Briefly describe the approach, methods, or techniques you will use to achieve your objectives.
- **Expected Results/Outcomes**: Summarize the anticipated results or outcomes of the project.
- **Significance/Impact**: Explain the potential impact or significance of the project.

Key Words:

Guide **Project Coordinator** **HOD**

Guidelines for a Strong Title:

1. **Be Specific:** The title should clearly indicate the focus of the project. Avoid vague or overly broad terms.
2. **Include Key Elements:** Mention the main components or technology used, the problem addressed, or the expected outcome.
3. **Be Concise:** Aim for a title that is succinct yet descriptive. Typically, a title should be between 10-15 words.
4. **Use Keywords:** Include important keywords that reflect the core of your project. This helps in making the title more searchable and relevant.

Example Title Components:

1. **Technology or Approach:** Mention if your project involves specific technologies (e.g., IoT, AI, machine learning).
2. **Application Area:** Indicate the field or area where the project is applied (e.g., agriculture, healthcare, education).
3. **Purpose or Goal:** Highlight the main objective or problem being addressed (e.g., optimization, enhancement, reduction).

Example Titles:

1. **Developing an IoT-Based Smart Irrigation System for Efficient Water Usage in Agriculture**
2. **AI-Driven Healthcare Monitoring System for Early Disease Detection**
3. **A Machine Learning Approach to Predictive Maintenance in Manufacturing Industries**
4. **Renewable Energy Solutions for Sustainable Urban Development**
5. **Designing an Educational Platform for Personalized Learning Using Adaptive Algorithms**

Crafting a Title for the Provided Example:

If we consider the earlier example of the smart irrigation system, a suitable title could be:

"IoT-Based Smart Irrigation System for Optimized Water Usage in Sustainable Agriculture"

This title clearly mentions:

- The technology used (IoT-Based)
- The main focus (Smart Irrigation System)
- The goal (Optimized Water Usage)
- The application area (Sustainable Agriculture)

By following these guidelines, you can create a title that is informative, specific, and engaging for your project abstract.

Ministry of Education | MoE's INNOVATION CELL (GOVERNMENT OF INDIA)

APF/YUKTI - National Innovation Repository

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT, (AICTE PID : 1-2506936)

Submit your innovations for the AICTE Productization Fellowship(APF) and YUKTI Innovation

HI KALASHETTY LIKHITHA,

Edit Idea/PoC Details

*Title	Title / Name (20 Words Max) *: Blockchain-Based System for Secure Academic Certificate Validation Total Number of words: 0 / 20
*Developed as part of	Academic Requirement/Study Project
*Choose the Financial Year, during the Idea-PoC/Innovation Developed	2024-25
*Sector / Domain	Theme *: ICT, cyber-physical systems, Blockchain, Cognitive computing, Cloud comp...
*Innovation Type	Product, Process
*Development Stage - Technology Maturity of the Solution/Innovation in terms of Technology Readiness Level TRL (if applicable (Refer TRL Stages)	TRL 3 : Applied research. First laboratory tests completed; proof of concept
*Define the problem and its relevance to today's market / sociaty / industry need (Max: 100 Words)	Define the problem and its relevance to today's market / sociaty / industry need *: Academic certificate forgery and verification delays are growing concerns in recruitment and higher education. Traditional systems involve manual verification and central authorities, leading to inefficiencies, fraud, and lack of transparency. In today's fast-paced digital environment, there's a need for secure, tamper-proof, and easily verifiable credentials. A blockchain- Total Number of words: 0 / 100

***Describe the Solution / Proposed / Developed (Max: 100 Words)**	Describe the Solution / Proposed / Developed * The solution involves issuing digital certificates stored on a blockchain platform (Ethereum) and accessed via a QR code. Each certificate is hashed and stored using smart contracts, ensuring immutability and tamper-proof verification. The blockchain ledger maintains transparency, and the QR code enables quick validation through a web-based interface. The system is **Total Number of words: 0 / 100**
***Explain the uniqueness and distinctive features of the (product / process / service) solution (Max: 100 Words)**	Explain the uniqueness and distinctive features of the (product / process / service) solution * The solution uniquely combines blockchain immutability with QR code accessibility, offering instant verification from any location. By using smart contracts, it ensures automated, transparent, and decentralized validation. Unlike traditional systems, it removes intermediaries and reduces the risk of forgery. The inclusion of hashing, secure identity, and timestamping **Total Number of words: 0 / 100**
***How your proposed / developed (product / process / service) solution is different from similiar kind of product by the competitors if any (Max: 100 Words)**	How your proposed / developed (product / process / service) solution is different from similiar kind of ... Unlike conventional centralized platforms or private permissioned chains like Hyperledger, this system uses a public blockchain (Ethereum), enhancing scalability, transparency, and trust. Unlike existing solutions, which store certificates off-chain or require manual validation, this solution stores hashed certificate data directly on-chain and automates validation **Total Number of words: 0 / 100**
***Is there any IP or Patentable Component associated with the Solution?**	No
***Has the Solution Received any Innovation Grant/Seefund Support?**	No
***Are there any Recognitions (State/National/International) Obtained by the Solution?**	No
***Is the Solution Commercialized either through Technology Transfer or Enterprise Development/Startup?**	No
***Had the Solution Received any Pre-Incubation/Incubation Support?**	No
Video URL	Video URL null
Upload Photograph: : (JPG, PNG, PDF max 2 MB)	Choose file Browse
	View File

Update

CONTACT

MoE's Innovation Cell

APF/YUKTI - National Innovation Repository

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT, (AICTE PID : 1-2506936)

Submit your innovations for the AICTE Productization Fellowship(APF) and YUKTI Innovation

HI KALASHETTY LIKHITHA,

Repository I

Building I&E Attitude

Enhancing I&E Ability

Innovation Re

Achieving I&E Aspirations

Add Team Mem

act Your Institute Expert Sessions

VIEW PROFILE

UPDATE INNOVATION

RESET PASSWORD

LOGOUT

Team Member Details

Name	Email	Phone	Designation	Gender	Caste	Action
GUGULOTHU SHILPA	227y1a6244@mlritm.com	7416381140	Student	Female	ST	Edit Delete

Add Mentor Details

Team Mentor Details

Name	Email	Phone	Designation	Organization	Type	Action
Mrs.Bhavya Varma	bhavyavarma@gmail.com	9963323459	Associate Professor	marri laxman reddy institute of technology and management	Internal to Institute	Edit Delete

Back

CONTACT

MoE's Innovation Cell

All India Council for Technical Education (AICTE), Nelson Mandela Marg, VasantKunj, New Delhi-110070.