DESIGN AND IMPLEMENTATION OF AN NFT-BASED CERTIFICATE VERIFICATION SYSTEM

BY

AIGBOVO PROSPER OSAROBO

PSC2008107

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

FEBRUARY 2025

DESIGN AND IMPLEMENTATION OF AN NFT-BASED CERTIFICATE VERIFICATION SYSTEM

BY

AIGBOVO PROSPER OSAROBO

PSC2008107

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

FEBRUARY 2025

CERTIFICATION

This is to certify that this project titled "Design and Implementation of an NFT-Based Certificate Verification System" was carried out by **AIGBOVO PROSPER OSAROBO** with Matriculation Number **PSC2008107** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.sc) Degree in Computer Science of the University of Benin

DR. (MRS.) A.R. USIOBAIFO **DATE**

Project Supervisor

APPROVAL

This project work is hereby approved in partial fulfillment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

PROF. (MR.) G.O EKUOBASE	**DATE**
Head of Department	

DEDICATION

This project is dedicated to God Almighty for giving me the strength and wisdom to see it through to completion, and even throughout my stay in the University of Benin (UNIBEN). It is also dedicated to my parents Mr. and Mrs. AIGBOVO and friends for their unwavering support.

ACKNOWLEDGEMENT

My utmost acknowledgement goes to God Almighty for giving me the strength, wisdom and direction throughout my academic journey. I would like to express my gratitude to my project supervisor DR. (Mrs.) A.R. Usiobaifo for her consistent guidance towards ensuring the successful completion of this project.

I would also like to specially thank my project coordinator, and other lecturers in the Department of Computer Science who I have been opportune to cross paths with, and have impacted me immensely these past few years: Prof. G.O. Ekuobase, Dr.F.O. Oliha, Prof. K.C. Ukaoha, Prof. A.A. Imiavan, Prof. (Mrs.) F. Egbokhare, Prof. (Mrs.)

V.V.N. Akwukwuma, Prof. F.I. Amadin, Prof. (Mrs.) S. Konyeha, Prof. (Mrs.) V.I. Osubor, Dr. (Mrs.) Aziken, Dr. F.O. Chete, Dr. (Mrs) R.O. Osaseri, Dr. J.C. Obi, Mr. P. E.B. Imiefoh, Mr. I.E. Obasohan, Mr. S.O.P. Oliomogbe, Mr. K.O. Otokiti, Mr. I.E. obayagbonna, Mrs. R.I. Izevbizua, Mr. E.C. Igodan, Miss L.O.Usiosefe, Mr J. Okhuoya, Prof. F.A.U. Imouokhome, Mrs. J.I. Adun, Dr. E. Nweli and Mr. D.N. Idehen.

Finally, I also want to appreciate those who contributed to the success of this project. I would also like to thank my family and friends for their support, words of encouragement, and consistent guidance throughout this project.

TABLE OF CONTENT

LIST OF FIGURE

ABSTRACT

Implementing blockchain-based certification systems is pivotal for enhancing the security, transparency, and efficiency of credential verification in academic and professional sectors. This study explores the design and deployment of an NFT-based certificate verification system, leveraging Ethereum blockchain, smart contracts, and decentralized storage (IPFS) to issue, verify, and manage digital credentials. By applying blockchain technology, smart contract logic, and decentralized file systems, this research addresses challenges such as certificate forgery, inefficient manual verification, and limited global accessibility.

Key methodologies include the use of Solidity smart contracts for minting NFTs, Web3.js for blockchain interaction, and React.js for user interface development, alongside IPFS for secure off-chain certificate storage. The system is designed to reflect real-world use cases involving students, educational institutions, and verifiers, supporting operations such as certificate issuance, revocation, and verification.

Outcomes include instant, tamper-proof verification processes, reduced operational costs, and improved user trust in credential authenticity. By integrating emerging technologies like Web3 and decentralized identity (DID), this research establishes a foundation for scalable, secure, and adaptable certification systems capable of meeting the evolving needs of education and professional validation

CHAPTER ONE

INTRODUCTION

1.0 Background Study

In today's digital era, verification of academic and professional certifications remains a critical challenge due to issues like forgery, data breaches, and inefficient verification processes. Traditional certification methods often rely on centralized databases that are vulnerable to tampering and unauthorized access, leading to compromised data integrity. Institutions, employers, and regulatory bodies face difficulties in ensuring the authenticity of certificates, leading to trust issues in the verification process.

Blockchain technology, with its decentralized and immutable nature, offers a robust solution to these challenges. Non-Fungible Tokens (NFTs), built on blockchain platforms, provide unique, tamper-proof, and verifiable digital assets that can revolutionize certification systems. By tokenizing certificates as NFTs, institutions can create a secure, transparent, and easily accessible verification process that ensures authenticity and reduces the risk of forgery.

This project focuses on designing and implementing an NFT-Based Certification Verification System that leverages blockchain technology to provide an immutable and transparent platform for issuing and verifying certificates. The system ensures that each certificate is uniquely represented as an NFT, enabling verifiable ownership and authenticity through decentralized networks.

1.1 Motivation

The increasing cases of certificate forgery and the complex processes involved in verification across various institutions highlight the need for a more secure and efficient system. Traditional methods often require manual checks, leading to delays and the risk of human error. With the rise of blockchain technology, there is an opportunity to simplify and strengthen the verification process, making it transparent and immutable.

The motivation behind this project is to leverage the power of NFTs and blockchain to build a certification verification system that ensures data integrity, authenticity, and quick verification

without relying on centralized authorities. The aim is to reduce certificate fraud and provide a simple, scalable solution for institutions and employers.

1.2 Problem Definition

In today's digital world, the authenticity and integrity of academic and professional certificates remain a growing concern, especially with the increasing cases of certificate forgery and data manipulation. Traditional certification systems, which rely on paper-based documents or centralized databases, are vulnerable to forgery, counterfeiting, and unauthorized alterations, leading to fraudulent claims of qualifications. The manual processes involved in verifying certificates are often time-consuming, inefficient, and prone to human error, especially when dealing with large volumes of records. Furthermore, the lack of transparency and traceability in these systems means that verifiers frequently need to contact issuing institutions directly, resulting in delays and potential errors during verification. The reliance on centralized databases presents another major vulnerability, as these systems are susceptible to data breaches, hacking, and unauthorized access, which can compromise the integrity of stored certificates. In an era where digital transformation is accelerating, these limitations highlight the urgent need for a more secure, transparent, and efficient system that ensures the authenticity, traceability, and easy verification of issued certificates. This has created a strong case for the adoption of blockchain technology, specifically Non-Fungible Tokens (NFTs), to develop a decentralized certification verification system that addresses these persistent challenges.

1.3 Goal and Objectives

The primary goal of this project is to develop a secure, efficient, and transparent NFT-based certification verification system. The specific objectives are:

1. To design a decentralized system for issuing and verifying certificates using NFTs.
2. To ensure data integrity and security by leveraging blockchain technology.
3. To create an intuitive interface for both issuers and verifiers to interact with the system.
4. To minimize the risk of certificate forgery and tampering.
5. To provide a scalable solution that can be adopted by educational institutions, professional organizations, and other certifying bodies.

1.4 Scope of Research

This project focuses on the development of a decentralized certification verification system using NFT technology, aiming to enhance the security, transparency, and efficiency of certificate issuance and verification processes. It involves designing and implementing the system architecture using blockchain and NFT standards, such as Ethereum and ERC-721, to ensure the uniqueness and immutability of each certificate. A web interface will be developed to enable institutions to issue certificates and allow third parties to verify them seamlessly. Smart contracts will be implemented to manage the minting and verification processes, ensuring automation and trust in the system. Additionally, the project will evaluate the system's security, scalability, and usability to ensure it meets the needs of various stakeholders. While the project will leverage existing public blockchain platforms for NFT creation and verification, it will not involve the development of an entirely new blockchain network.

1.5 Research Methodology

The methodology employed in this project follows a systematic approach, combining both qualitative and quantitative research methods to ensure a comprehensive and effective development process. It begins with requirement analysis, where the functional and non-functional requirements of the certification system are identified through an extensive literature review and stakeholder interviews. This is followed by the system design phase, which utilizes Object-Oriented Analysis and Design (OOAD) principles, incorporating UML diagrams such as use case, class, and sequence diagrams to model system interactions and architecture accurately in the technology stack selection phase, appropriate blockchain platforms, smart contract languages (e.g., Solidity), and front-end technologies (e.g., React.js) are chosen to ensure a robust and scalable system. The implementation phase adopts agile methodologies, promoting iterative development and continuous testing to enhance flexibility and accommodate evolving project requirements. Finally, the system undergoes rigorous testing and evaluation, including functional, security, and usability testing, to ensure optimal performance, user experience, and adherence to the defined objectives. This structured approach ensures the development of a secure, efficient, and user-friendly NFT-based certification verification system.

1.6 Research Significance

The implementation of an NFT-Based Certification Verification System offers numerous benefits:

1. Enhanced Security: Blockchain's immutability ensures that once a certificate is issued, it cannot be altered or forged.
2. Transparency and Traceability: All issued certificates are recorded on a public ledger, enabling transparent and verifiable records.
3. Cost and Time Efficiency: Automated verification processes reduce manual efforts, saving time and resources for institutions and employers.
4. Decentralization: The elimination of central authorities reduces single points of failure, enhancing the system's resilience and reliability.
5. Universal Accessibility: The verification process can be conducted from anywhere in the world, promoting global accessibility.

CHAPTER TWO

LITRATURE REVIEW

The increasing rate of certificate forgery and the inefficiencies in traditional verification processes have highlighted the need for more secure and transparent certification systems. As blockchain technology continues to gain prominence, its application in the education sector, particularly for certification verification, has become a subject of significant interest. This chapter reviews existing literature on traditional and modern certification systems, explores the fundamentals of blockchain and Non-Fungible Tokens (NFTs), and examines how NFT technology can be applied to develop a decentralized certification verification system.

2.1 Overview of Certification Systems

Certification systems play a crucial role in validating academic achievements, professional qualifications, and skill proficiencies. Over the years, these systems have evolved from simple paper-based documents to sophisticated digital records, reflecting the technological advancements and the growing need for secure and efficient verification processes. This section provides an in-depth overview of certification systems, their historical context, existing challenges, and the emerging role of blockchain and NFTs in addressing these challenges.

Historical Background of Certification Systems

The concept of certification as proof of achievement or qualification dates back centuries. In ancient guilds during the Middle Ages, apprenticeships served as the foundation for skills training, culminating in certificates or licenses issued to craftsmen who mastered a trade. These early forms of certification validated skills and knowledge and were essential for individuals to practice their craft legally In the modern academic context, certificates became standardized as formal proof of education, such as diplomas, degrees, and professional licenses. Educational institutions, professional bodies, and government agencies adopted structured systems to issue certifications, creating a formalized process for recognizing academic and professional achievements. Over time, certification systems expanded to include not just academic degrees but also vocational training, professional development programs, and specialized industry credentials.

Traditional Certification Methods

For decades, traditional certification systems have relied on paper-based certificates and centralized databases to issue and manage records. While these methods have been widely accepted and used, they present several limitations:

i. **Paper-Based Certificates:** Institutions typically issue physical certificates that bear institutional seals, signatures, and watermarks to verify authenticity. However, paper certificates are susceptible to damage, loss, and, most notably, forgery.
ii. **Centralized Databases:** In an attempt to digitize records, many institutions have shifted towards centralized databases where certificate records are stored. Although this reduces the risk of physical damage or loss, it introduces new vulnerabilities, including data breaches, hacking, and unauthorized access.

Both methods still rely heavily on manual processes for verification. Employers, academic institutions, and other third parties often need to directly contact the issuing institution to confirm the authenticity of a certificate, which can be time-consuming and inefficient, especially when dealing with large volumes of records or international verifications.

Challenges in Traditional Certification Systems

Despite their widespread use, traditional certification systems face numerous challenges that undermine their effectiveness:

i. **Forgery and Counterfeiting:** Physical certificates can be easily forged using advanced printing technologies. Fake certificates, fake transcripts, and even counterfeit degrees have become common in many industries, leading to fraudulent claims of qualifications.
ii. **Inefficient Verification Processes:** The manual nature of verification processes slows down decision-making in critical situations, such as job recruitments or academic admissions. The need to contact issuing institutions and wait for confirmation can create delays.
iii. **Lack of Transparency and Traceability:** Centralized systems often do not provide transparent access to certificate records. Verifiers must rely on the issuing institution, making the process less transparent and susceptible to human error.

iv. **Centralized Vulnerabilities:** Centralized databases act as single points of failure. If compromised, all stored data can be at risk. Data breaches, hacking incidents, and unauthorized modifications have affected numerous institutions, leading to the loss or manipulation of sensitive data.

v. **Geographical and Institutional Barriers:** Verifying certificates across borders can be particularly challenging, especially in cases where different countries follow varying standards. Additionally, language barriers and differences in educational frameworks further complicate the verification process.

Digital Certification Systems: An Evolving Solution

In response to the limitations of traditional methods, institutions began adopting digital certification systems. These systems digitize certificate records, making them easier to store, manage, and share. Common digital certification methods include:

i. **Electronic Diplomas and Transcripts:** Many universities now issue e-diplomas and e-transcripts in PDF or other digital formats, which include digital signatures to verify authenticity.

ii. **Centralized Digital Databases:** Institutions host centralized online portals where employers or third parties can verify certificates after logging in with authorized credentials.

iii. **Cloud-Based Certification Platforms:** Cloud technology enables centralized databases to be hosted online, allowing for remote verification processes.

While digital certification systems have improved efficiency, they still face critical challenges. Centralization remains a major vulnerability, as cyberattacks and data breaches continue to compromise sensitive records. Additionally, digital documents such as PDFs can still be altered or forged using sophisticated editing tools.

The Emergence of Blockchain Technology in Certification

To overcome the limitations of centralized systems, institutions have started exploring blockchain technology as a solution for secure, transparent, and tamper-proof certification.

Blockchain, a decentralized ledger system, records data in immutable blocks across multiple nodes, making it resistant to tampering and unauthorized access.

Key Features of Blockchain for Certification Systems:

i. **Decentralization:** Records are stored across a distributed network, eliminating the risk of a single point of failure.
ii. **Immutability:** Once a certificate is added to the blockchain, it cannot be altered or deleted, ensuring data integrity.
iii. **Transparency and Traceability:** Blockchain allows stakeholders to track and verify certificates without relying on intermediaries, making the verification process faster and more transparent.
iv. **Security:** Blockchain uses cryptographic algorithms to secure data, making it highly resistant to hacking and unauthorized changes.

Universities, professional bodies, and even governments have begun to pilot blockchain-based certification projects. Institutions like the Massachusetts Institute of Technology (MIT) and University College London (UCL) have already adopted blockchain for issuing verifiable digital certificates, setting a precedent for others to follow.

Non-Fungible Tokens (NFTs) in Certification Verification

While blockchain offers a robust foundation for certification systems, the introduction of Non-Fungible Tokens (NFTs) has further revolutionized the process. NFTs are unique digital assets stored on the blockchain that represent ownership of a specific item or piece of data. Unlike cryptocurrencies, NFTs are indivisible and non-interchangeable, making them ideal for representing certificates.

Benefits of Using NFTs for Certification:

i. **Uniqueness and Authenticity:** Each certificate minted as an NFT carries a unique identifier, ensuring that no two certificates are identical.
ii. **Proof of Ownership:** NFTs are tied to specific blockchain addresses, enabling verifiable proof of certificate ownership.
iii. **Tamper-Proof Records:** Once a certificate is minted as an NFT, it cannot be altered or duplicated, safeguarding against forgery.

iv. **Global Accessibility:** NFT-based certificates can be verified from anywhere in the world using blockchain explorers or dedicated verification portals.
v. **Easy Transferability:** NFT certificates can be transferred between individuals or institutions without losing their authenticity, simplifying processes like academic transfers or job applications.

Several platforms now specialize in NFT-based certification. For example, Blockcerts enables institutions to issue verifiable credentials on the blockchain, while CertifyMe uses NFTs to provide tamper-proof digital certificates for professional development and skills training.

Real-World Applications of NFT-Based Certification Systems

Several educational institutions, professional bodies, and industries have started adopting NFT-based certification systems:

i. **Academic Institutions:** Universities are issuing diplomas and academic transcripts as NFTs, allowing graduates to share their credentials securely with employers.
ii. **Professional Certifications:** Organizations in fields like information technology, finance, and healthcare use NFT-based certifications to verify skills and qualifications.
iii. **Online Learning Platforms:** E-learning platforms like Coursera and Udemy are exploring NFT certifications to enhance the credibility and verifiability of their course completions.
iv. **Event and Conference Badges:** NFTs are also used to issue digital badges for conference participation, workshops, and hackathons, providing verifiable proof of attendance and achievement.

Future Trends in Certification Verification Systems

The adoption of blockchain and NFT technologies in certification is expected to grow, driven by the increasing need for secure, efficient, and globally accessible verification processes. Key future trends include:

i. **Interoperability Across Institutions:** Efforts are underway to create interoperable standards that allow different educational institutions and professional bodies to use compatible blockchain platforms for certification.

ii. **Decentralized Identity Systems:** Blockchain-based identity verification systems could integrate with NFT certificates, creating a unified digital identity for individuals.
iii. **Regulatory Compliance:** As the use of blockchain in certification becomes more widespread, regulatory bodies are working to establish guidelines and standards to ensure compliance with data privacy laws and educational regulations.
iv. **Enhanced User Interfaces:** To encourage widespread adoption, future certification platforms will focus on creating user-friendly interfaces that make it easier for both institutions and verifiers to interact with blockchain-based systems

2.2 Traditional Certification Verification Systems

Certification verification has long been a critical component in academic, professional, and skill-based qualification systems. Traditional certification verification systems have primarily relied on physical documents or centralized digital databases to authenticate credentials. While these methods have been widely accepted and used across educational institutions, corporate organizations, and government bodies, they present several challenges that compromise security, efficiency, and transparency. This section explores the structure of traditional certification systems, their inherent limitations, and the growing need for more advanced verification methods.

Structure of Traditional Certification Systems

Traditional certification verification systems generally operate using two primary methods: physical certificates and centralized digital records.

i. **Physical (Paper-Based) Certificates:** Physical certificates have long been the standard for representing academic degrees, professional qualifications, and training achievements. These documents typically feature signatures, institutional seals, and watermarks to signify authenticity. However, despite these security measures, physical certificates remain vulnerable to forgery and tampering.
ii. **Centralized Digital Records:** In recent years, many institutions have transitioned to maintaining centralized digital records. These records are stored in secure databases managed by the issuing institution, often accessible through dedicated online portals.

While digital records offer easier access and retrieval, they come with their own set of vulnerabilities, particularly concerning data security and system integrity. Both methods rely heavily on the issuing institution's authority, making the verification process dependent on manual checks and institutional cooperation.

Core Limitations of Traditional Certification Systems

Despite their widespread adoption, traditional certification verification systems face numerous limitations that hinder their effectiveness:

Forgery and Counterfeiting: One of the most significant challenges facing traditional certification systems is the ease with which physical certificates can be forged. Advanced printing technologies and graphic design tools have made it possible for counterfeiters to replicate certificates with high levels of accuracy. Fake diplomas, altered transcripts, and counterfeit degrees have become common in both academic and professional settings.

Key Issues:

i. **Lack of Real-Time Verification:** Employers and verifiers often rely on the visual inspection of certificates, which cannot confirm their authenticity.
ii. **Widespread Fraud:** Cases of certificate forgery have been reported globally, undermining the credibility of educational and professional qualifications.

Manual Verification Processes

The process of verifying traditional certificates often requires direct communication between the verifier and the issuing institution. In many cases, this involves sending requests via email or post, waiting for responses, and relying on institutional records to confirm authenticity. This manual process is time-consuming, inefficient, and prone to human error.

Challenges of Manual Verification:

i. **Time Delays:** Verification can take days or even weeks, especially when dealing with international institutions.

ii. **Administrative Burden:** Institutions must allocate staff to handle verification requests, leading to increased operational costs.
iii. **Human Error:** Mistakes in data entry, miscommunication, or failure to cross-reference records can result in incorrect verifications.

Case Studies of Issues in Traditional Certification Systems

i. **Diploma Mills:** Diploma mills are unaccredited institutions that sell fake degrees and certificates without requiring coursework or legitimate academic achievement. These fraudulent practices thrive because traditional verification systems lack transparency and standardization, making it difficult for employers to differentiate between legitimate and fake degrees.
ii. **International Verification Barriers:** In cross-border education and employment scenarios, verification becomes even more complex due to differences in language, educational standards, and institutional policies. Some institutions may not have clear processes for international verification, leading to delays and potential misinterpretations.
iii. **Data Breaches in Educational Institutions:** Educational institutions are increasingly targeted by cybercriminals due to the wealth of sensitive information they store. In cases where centralized databases are compromised, personal information, grades, and certificate data can be stolen or manipulated, raising concerns over data integrity and privacy.

2.3 Blockchain Technology in Certification Systems

The increasing need for secure, transparent, and tamper-proof certification verification systems has driven the adoption of in the academic and professional certification sectors. Originally designed as the underlying technology for cryptocurrencies like Bitcoin, blockchain has evolved into a versatile tool with applications across various industries, including finance, supply chain management, healthcare, and now, education and professional certification.In the context of certification systems, blockchain offers a decentralized and immutable framework that addresses the key challenges faced by traditional verification methods. Its ability to create

transparent, secure, and verifiable records makes it an ideal solution for issuing and managing academic degrees, professional licenses, training certificates, and other credentials.

Understanding Blockchain Technology

Blockchain is a decentralized, distributed ledger technology that records transactions across multiple nodes in a network. Each transaction is grouped into a block, which is cryptographically linked to the previous block, forming a chain of data — hence the name "blockchain." This structure ensures that once data is recorded, it cannot be altered or deleted without consensus from the entire network.

The blockchain operates on the following core principles:

i. **Decentralization:** Instead of storing data in a single, centralized database, blockchain distributes it across a network of nodes. Each node maintains a copy of the entire ledger, eliminating single points of failure and reducing the risk of data manipulation.
ii. **Immutability:** Data recorded on the blockchain is permanent and tamper-proof. Once a block is added to the chain, altering its contents requires approval from the majority of the network and re-computation of all subsequent blocks — a virtually impossible task.
iii. **Transparency:** Transactions on public blockchains are visible to all participants, allowing for independent verification and auditability. This transparency builds trust in the system and ensures accountability.
iv. **Security:** Blockchain uses cryptographic techniques, such as hashing and digital signatures, to secure data. Each block contains a unique hash and is linked to the previous block, ensuring that any attempt to tamper with data is immediately detectable.

How Blockchain Supports Certification Verification

Certification verification systems benefit significantly from blockchain's decentralized and immutable structure. By leveraging blockchain technology, institutions can issue digital certificates that are tamper-proof, easily verifiable, and globally accessible.

i. **Immutability:** One of the most critical features of blockchain is its immutability — once data is recorded, it cannot be changed or deleted. This is particularly valuable in certification systems, where the integrity of issued certificates is essential.

ii. **Tamper-Proof Records:** When a certificate is issued and recorded on the blockchain, it becomes a permanent entry that cannot be altered by anyone, including the issuing institution. This eliminates the risk of certificate forgery or unauthorized modifications.

iii. **Proof of Authenticity:** Every certificate on the blockchain carries a unique identifier and timestamp, enabling verifiers to confirm its authenticity instantly.

Decentralization

Traditional certification systems often rely on centralized databases, making them vulnerable to data breaches and single points of failure. Blockchain's decentralized architecture eliminates this risk.

- **No Single Point of Failure:** Since data is distributed across multiple nodes, there is no central authority that controls the information. Even if one node is compromised, the integrity of the entire blockchain remains intact.
- **Reduced Risk of Unauthorized Access:** Decentralization limits the possibility of unauthorized alterations, as changes to the data require consensus from the majority of nodes in the network.

Transparency

Transparency is a core principle of blockchain that enhances trust and accountability in certification systems.

- **Open Verification:** In public blockchains, anyone can verify the authenticity of a certificate by accessing the blockchain ledger. This eliminates the need for third-party intermediaries, reducing time and costs associated with manual verification.
- **Audit Trails:** Blockchain maintains a complete history of all transactions, enabling stakeholders to trace the issuance and verification of certificates over time.
- **Independent Validation:** Verifiers do not need to rely on the issuing institution to confirm the validity of a certificate they can independently verify it on the blockchain.

Security

Blockchain's security features protect certification data from tampering, hacking, and unauthorized access.

- **Cryptographic Hashing:** Each block in the blockchain is cryptographically secured using a unique hash. Any attempt to alter the data changes the hash, immediately flagging the inconsistency.
- **Consensus Mechanisms:** Blockchain networks use consensus algorithms (such as Proof of Work or Proof of Stake) to validate transactions. This prevents malicious actors from making unauthorized changes to the ledger.
- **Data Privacy:** While public blockchains provide transparency, private or permissioned blockchains can be used for sensitive information, ensuring that only authorized users have access to detailed certificate data.

2.4 Non-Fungible Tokens (NFTs) and Their Role in Certification

In recent years, Non-Fungible Tokens (NFTs) have emerged as one of the most transformative innovations in blockchain technology. Initially popularized through the digital art and collectibles market, NFTs have found practical applications across various industries, including gaming, real estate, identity verification, and notably, education and certification systems. Their inherent properties of uniqueness, immutability, and traceability make NFTs an ideal solution for creating secure and verifiable digital certificates. This section explores the concept of NFTs, their underlying technology, and how they can revolutionize certification verification systems by ensuring authenticity, ownership, and easy verification of academic and professional credentials.

Understanding Non-Fungible Tokens (NFTs)

A Non-Fungible Token (NFT) is a unique digital asset that exists on a blockchain. Unlike fungible tokens (such as Bitcoin or Ethereum), which are interchangeable and identical in value, NFTs represent distinct, one-of-a-kind assets. This uniqueness is what makes NFTs particularly valuable for applications requiring proof of ownership and authenticity.

Key Characteristics of NFTs:

i. Uniqueness: Every NFT has a unique identifier and metadata that distinguishes it from any other token.

ii. Indivisibility: NFTs cannot be divided into smaller units like cryptocurrencies. They exist as whole, individual assets.
iii. Verifiability: Since NFTs exist on the blockchain, anyone can verify the ownership history and authenticity of an NFT through blockchain explorers.
iv. Interoperability: NFTs can be created, transferred, and stored across different platforms that support the same blockchain standards (e.g., ERC-721 or ERC-1155 on Ethereum).

NFTs are created through a process known as minting, where a digital asset is tokenized and recorded on the blockchain. Once minted, the NFT contains unique metadata, which can include information such as the creator's identity, a timestamp, and ownership details.

NFTs in Certification Verification Systems

The core attributes of NFTs uniqueness, immutability, and verifiability align perfectly with the requirements of a secure certification verification system. By representing academic degrees, professional licenses, and training certificates as NFTs, institutions can issue tamper-proof and easily verifiable credentials.

How NFTs Enhance Certification Systems:

Tokenization of Certificates: Each certificate is minted as a unique NFT on a blockchain. The NFT's metadata includes critical information such as the recipient's name, course details, issuing institution, date of issuance, and a unique certificate ID.

i. Decentralized Ownership: The NFT is transferred to the recipient's digital wallet, establishing them as the rightful owner. This eliminates the need for centralized record-keeping by the issuing institution.
ii. Immutable Records: Once the certificate is minted and recorded on the blockchain, it becomes immutable, ensuring that the credential cannot be altered or tampered with.

iii. Global Verifiability: Verifiers (such as employers or other academic institutions) can independently verify the certificate by checking the NFT on the blockchain without needing to contact the issuing institution.

Advantages of NFTs in Certification Systems

NFTs offer several distinct advantages that address the limitations of traditional and centralized digital certification methods:

i. **Uniqueness and Authenticity:** Each NFT is inherently unique due to its metadata, ensuring that no two certificates are identical. This eliminates the risk of duplicate or counterfeit certificates. Example: A university issues two students with the same degree, but each degree is minted as a separate NFT with distinct identifiers. Even though the academic qualification is the same, the NFTs are unique, preventing duplication.

ii. **Proof of Ownership:** When a certificate is minted as an NFT and transferred to a recipient's blockchain wallet, it serves as irrefutable proof of ownership. The blockchain ledger maintains a transparent record of all transactions, allowing anyone to trace the certificate back to its original issuer. Example: In a job recruitment process, an employer can verify that the applicant is the rightful holder of a specific academic degree by reviewing the NFT's transaction history on the blockchain.

iii. **Tamper-Proof and Immutable Records:** Once a certificate is issued as an NFT, its details cannot be altered, deleted, or tampered with. The immutability of blockchain ensures that the certificate's authenticity remains intact over time. Example: Even if a university's internal database is compromised or deleted, the NFT certificate remains securely stored on the blockchain and verifiable by third parties.

iv. **Easy and Instant Verification:** Verifying a traditional certificate often requires contacting the issuing institution and waiting for confirmation. NFT-based certificates, however, can be verified instantly by checking the blockchain, significantly reducing verification time and effort. Example: An employer can use a blockchain explorer or a dedicated verification platform to confirm the authenticity of an applicant's certificate within seconds.

v. **Transparency and Traceability:** The transparent nature of blockchain ensures that all certificate issuance and ownership records are publicly accessible. Verifiers can trace

the entire history of a certificate, from issuance to its current holder. Example: In cases where an individual transfer their NFT certificate to another entity (for example, as part of a research grant application), the transaction history remains visible and verifiable.

vi. **Global Accessibility:** NFT-based certificates are not restricted by geographical boundaries. Verifiers worldwide can access and verify credentials using blockchain explorers, making NFT certification ideal for international education and employment.

2.5 Related Works on NFT-Based Certification Systems

Several research studies and real-world implementations have explored the use of blockchain and NFTs in certification systems. Below are some notable examples:

Kinkelin et al. (2020) proposed a system that enhances the security of X.509 certificate issuance by enforcing a policy-defined, multi-party validation and authorization workflow for certificate signing requests. The implementation leverages the distributed ledger and smart contract framework Hyperledger Fabric, ensuring tamper-resistant processes and accountability.

Szalachowski (2020) presented SmartCert, a novel approach utilizing smart contracts to improve digital certificates. The system conveys detailed information about the validation state of certificates, which changes according to specified smart contract code and individual domain policies. This design ensures transparency and accountability in certificate issuance and management. These studies collectively highlight the potential of integrating blockchain and NFT technologies to create secure, transparent, and efficient certificate verification systems, addressing the inherent challenges of traditional methods

Tumati (2021) introduced a certificate authentication system leveraging soulbound tokens on the blockchain. This approach aims to streamline the verification process for various documents, including educational certificates, by providing conclusive evidence of authenticity and reducing the time and human effort required.

Zhao and Si (2022) introduced NFTCert, a novel framework that leverages NFTs to establish a verifiable link between certificates and their owners on the blockchain. The system encompasses schema definition, minting, verification, and revocation processes. Notably,

NFTCert incorporates an online payment gateway, enabling users to engage without relying on cryptocurrencies, thereby enhancing accessibility and usability.

Wang et al. (2020) In their study on blockchain-based educational certificates, the authors highlighted the potential of NFTs in reducing certificate fraud and streamlining verification processes. They emphasized the importance of interoperability between institutions and blockchain platforms.

Summary Table of Related Works

Author(s)	Year	Objectives	Major Features
Kinkelin et al.	2020	To enhance the security of X.509 certificate issuance using Hyperledger Fabric.	Uses Hyperledger Fabric for multi-party validation and tamper-resistant processes.
Szalachowski	2020	To improve digital certificates using smart contracts for transparency and accountability.	Utilizes smart contracts to manage certificate validation states and domain policies.
Zhao and Si	2022	To establish a verifiable link between certificates and their owners using NFTCert.	Incorporates schema definition, minting, verification, revocation, and an online payment gateway.
Wang et al.	2020	To explore the use of NFTs in reducing certificate fraud and	Emphasizes interoperability between institutions

		enhancing interoperability.	and blockchain platforms.
Tumati	2021	To develop a certificate authentication system using soul bound tokens on the blockchain.	Leverages soul bound tokens for streamlined and conclusive certificate verification.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

This chapter provides an in-depth model of the development process while offering a detailed overview of the system analysis and design for the proposed NFT-Based Certification Verification System. It explores the methodologies used, analyzes the limitations of existing systems, and describes the proposed system's architecture, design tools, and implementation strategies.

3.1 System Analysis

System analysis is a vital stage in the software development life cycle (SDLC), focusing on understanding and documenting the system's requirements, components, and processes. It involves a detailed examination of the existing systems, user needs, and project objectives to create an efficient and scalable solution.

System Analysis Methodologies

There are several methods of system analysis and design, but the two major methodologies are:

Object-Oriented Analysis and Design (OOAD) Method: The OOAD approach focuses on modeling real-world entities into software objects, promoting modularity and reusability. It emphasizes the identification of objects, their attributes, behaviors, and interactions to create a robust system design. OOAD is used for this project due to its emphasis on scalability, maintainability, and adaptability.

Structured System Analysis and Design (SSAD) Method: The SSAD methodology uses process models like Data Flow Diagrams (DFDs) and flowcharts to break down systems into smaller, manageable components. It follows a top-down approach and is particularly useful for procedural programming but less adaptable for object-oriented solutions.

Why OOAD Was Chosen: The OOAD methodology was selected for this project as it aligns with the development of a decentralized, modular, and reusable system architecture. The focus on objects, such as certificates, users, and smart contracts, ensures better scalability, maintainability, and adaptability to evolving requirements.

3.2 Analysis of Existing System

Traditional certification systems predominantly rely on paper-based certificates or centralized digital databases. These methods, while familiar, have several inherent flaws that make them susceptible to security breaches, inefficiencies, and data manipulation.

Limitations of Traditional Certification Systems:

i. Forgery and Counterfeiting Paper certificates can be easily forged or altered, leading to fraudulent claims of qualifications.
ii. Manual Verification Process: Verifiers often need to contact issuing institutions directly, leading to time-consuming and error-prone processes.
iii. Data Vulnerabilities: Centralized databases present single points of failure, making them vulnerable to hacking, unauthorized modifications, and data loss.Limited Transparency: Verifiers cannot independently confirm the authenticity of certificates without the issuing institution, reducing trust and efficiency.

Example of Traditional Systems: Universities issuing physical diplomas and transcripts that require manual verification processes. Employers or other institutions must contact the issuing university, leading to delays and potential errors in verification.

3.3 Problems of the Existing System

Based on the analysis of traditional certification systems, the following problems have been identified:

i. Forgery and Fraudulent Certificates: The lack of tamper-proof mechanisms enables the proliferation of counterfeit documents.
ii. Time-Consuming Verification: Manual verification requires significant time and administrative resources, especially for international verifications.
iii. Data Integrity Risks: Centralized databases are prone to breaches, unauthorized access, and data loss.
iv. High Administrative Costs: Manual verification and certificate issuance processes lead to higher operational costs for institutions.
v. Lack of Global Accessibility: Cross-border verification is complicated by varying standards and requirements between institutions and countries.

3.4 Overview of Proposed System

The proposed NFT-Based Certification Verification System leverages blockchain and NFT technology to address the limitations of traditional systems. By issuing certificates as NFTs, the system ensures tamper-proof records, decentralized verification, and global accessibility.

Key Features of the Proposed System:

Decentralized Certificate Issuance: Institutions mint certificates as NFTs on the blockchain, ensuring transparency and immutability.

i. Secure Verification: Verifiers can instantly confirm the authenticity of certificates using blockchain explorers or the system's verification portal.
ii. Tamper-Proof Records: Once a certificate is minted as an NFT, it cannot be altered, ensuring data integrity.
iii. Ownership Proof: Each NFT certificate is tied to the recipient's digital wallet, providing verifiable proof of ownership.
iv. Revocation Mechanism: Institutions can revoke certificates if necessary, with the revocation recorded on the blockchain.
v. Global Accessibility: Certificates can be verified globally without needing to contact the issuing institution.

3.5 Proposed System Architecture and Interface

The system architecture follows a three-tier architecture consisting of the Presentation Layer, Application Layer, and Data Layer, supported by blockchain integration.

System Architecture Components:

i. Presentation Layer (Frontend): Provides a user-friendly web interface for institutions, students, and verifiers. Technologies: React.js, HTML5, CSS3, JavaScript.
ii. Application Layer (Backend): Handles business logic, user authentication, and interaction with the blockchain. Technologies: Node.js/Express.js, Web3.js, Solidity (for smart contracts).
iii. Data Layer: Blockchain Network (Ethereum): Stores certificate NFTs and related transaction data.
iv. IPFS (InterPlanetary File System): Stores large certificate files off-chain while keeping metadata on-chain.

v. MySQL Database: Stores user profiles, authentication data, and other metadata not stored on the blockchain.

User Interface Overview:

Students: View and download certificates, share certificates for verification.

Institutions: Mint NFT certificates. revoke certificates if necessary, view issued certificate records.

Verifiers: Verify certificates using a certificate ID, view certificate metadata and validation status.

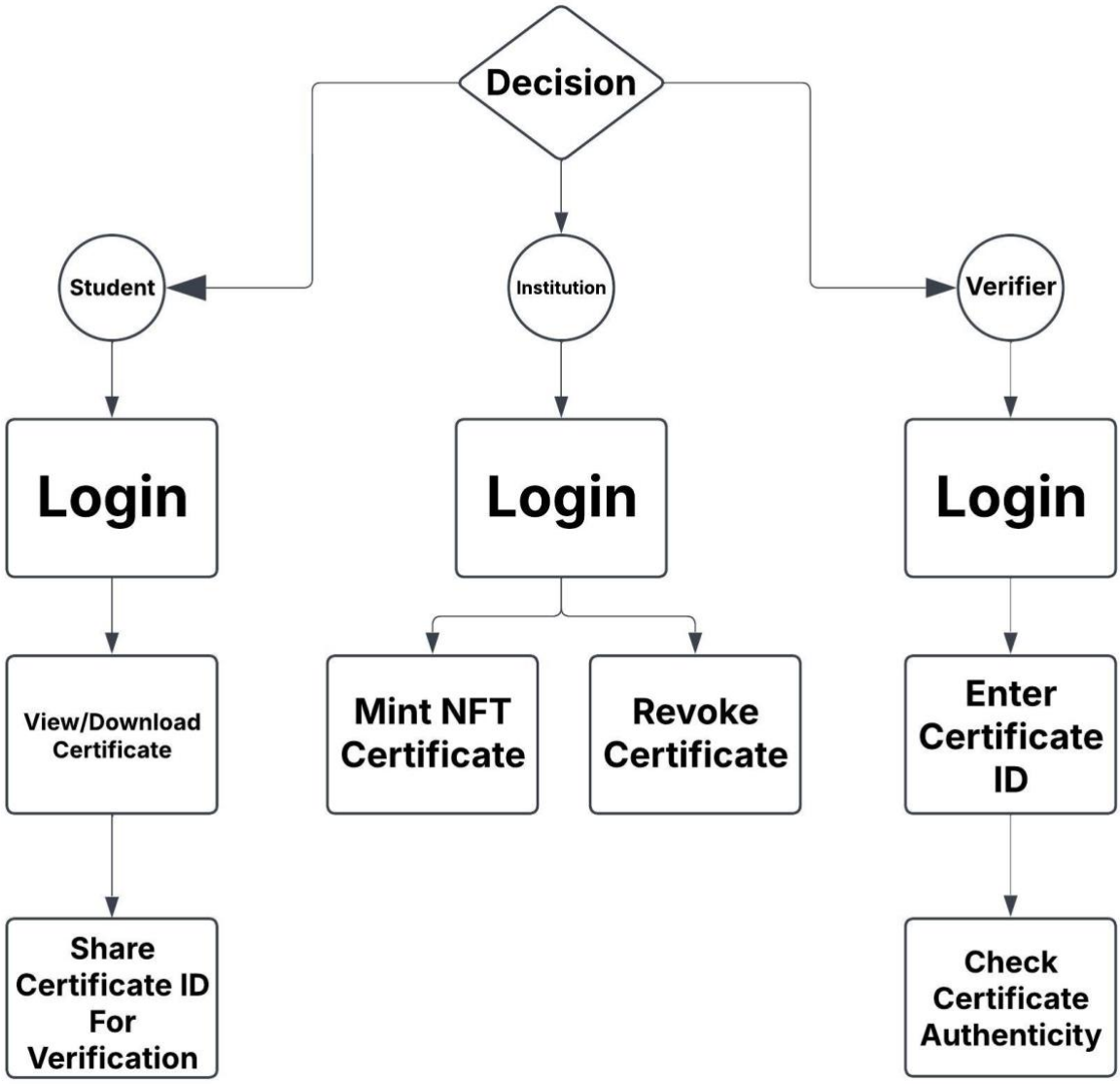

Figure 3.0 User Interface Flow for Certificate Management

This diagram illustrates the decision-making process and user interactions within the NFT-based certificate verification system. It showcases the three main user roles: students, institutions, and verifiers.

- Students can log in, view/download their certificates, and share them for verification.

• Institutions are responsible for minting new certificates as NFTs and revoking invalid certificates.

• Verifiers can enter a certificate ID to check its authenticity.

This structured flow ensures a secure, transparent, and user-friendly approach to certificate management using blockchain technology.

3.7 System Design Tools

To ensure a structured and efficient system design, several industry-standard tools and methodologies were utilized:

1.Unified Modeling Language (UML): UML diagrams were used to visualize the system's architecture and user interactions. Key UML diagrams include:

i. Use Case Diagram: Illustrates the interactions between users (students, institutions, and verifiers) and the system.
ii. Class Diagram: Shows the structure of the system, highlighting classes, attributes, and relationships.
iii. Sequence Diagram: Maps out the flow of data and the sequence of interactions between the system components during key operations (e.g., certificate minting and verification).

2.Data Flow Diagram (DFD)

The DFD illustrates how data moves through the system, from user input to blockchain storage and retrieval from IPFS. It highlights:

Data input points (e.g., certificate upload).

Data processing steps (e.g., minting the NFT and storing the IPFS hash).

Data output (e.g., certificate verification results).

3. System Flowchart

The system flowchart provides a step-by-step representation of the processes within the system. It covers: The process of minting certificates.

The flow for verifying certificates using the token ID or QR code.

The steps involved in revoking a certificate and updating its status on the blockchain.

4. Entity-Relationship Diagram (ERD)

The ERD maps out the relationships between different data entities within the system, such as:

Students

Certificates

Institutions

Verifiers

It showcases the one-to-many and many-to-one relationships that exist between these entities, such as one institution issuing multiple certificates or one student holding multiple certificates.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Software Implementation Tools

The development of the system was executed using the following tools and technologies:

Technology/Tool	**Purpose**
Ethereum Blockchain	Secure and decentralized certificate storage
Solidity	Smart contract development
Remix IDE	Writing, compiling, and deploying smart contracts
IPFS (InterPlanetary File System)	Decentralized file storage for certificates
Web3.js	Integration between Ethereum and web frontend
React.js	Frontend development framework
MetaMask	Wallet integration for transactions
Visual Studio Code	Source code editor

4.2 System Architecture

The architecture is built on the Ethereum blockchain for certificate management, with IPFS handling file storage. The frontend is developed using React.js, integrated with Web3.js, and connected to MetaMask for user authentication and transactions.

4.2.1 System Architecture Diagram

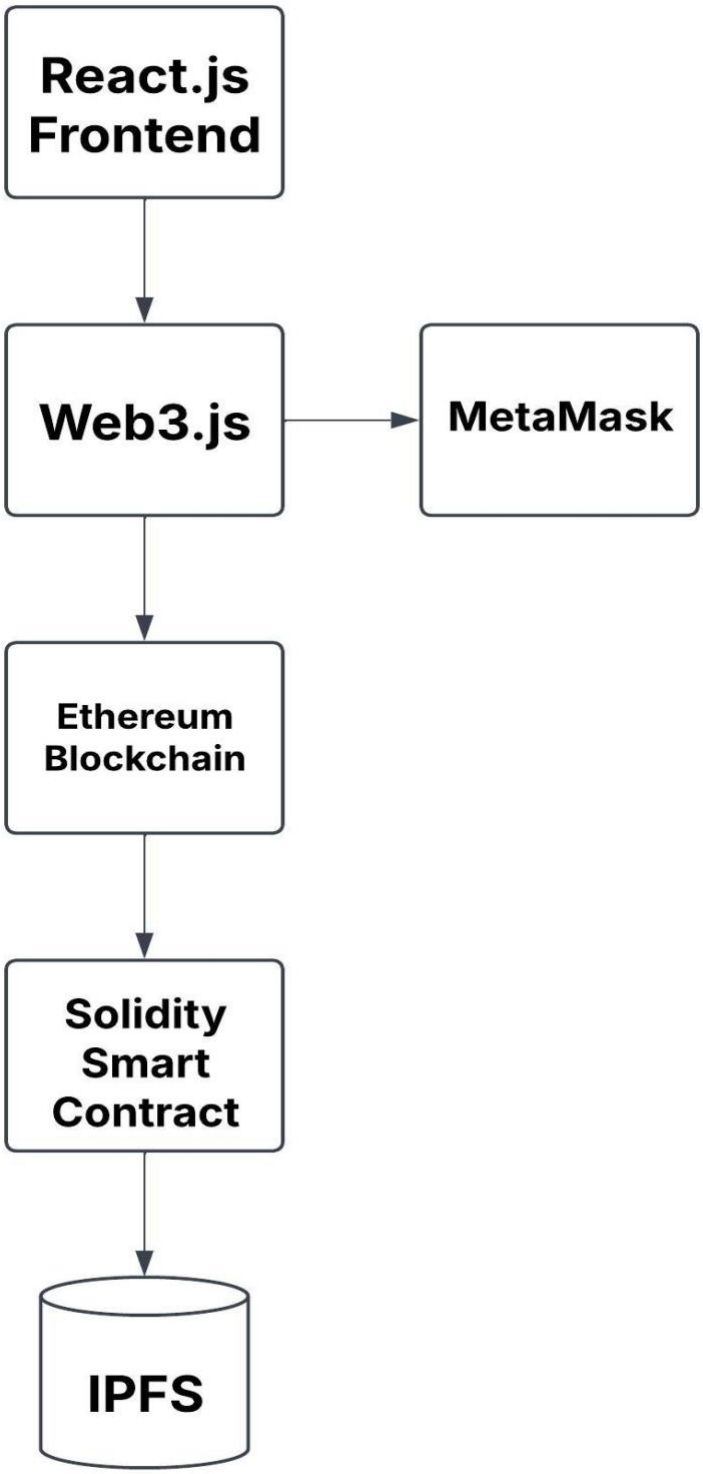

Figure 4.0 System Architecture Diagram

Key Components:

1. React.js Frontend→ Provides user interface for students, institutions, and verifiers.

2. Web3.js→ Connects the frontend to the Ethereum network.

3. Solidity Smart Contracts→ Manages the issuance, verification, and revocation of certificates.

4. IPFS→ Stores the actual certificate files, with hashes saved on the blockchain.

5. MetaMask→ Facilitates Ethereum transactions and wallet management.

4.3 Smart Contract Development

The core functionality of the NFT-Based Certification Verification System is built upon smart contracts, developed using Solidity and deployed on the Ethereum Blockchain through Remix IDE. Smart contracts serve as self-executing programs that facilitate the minting, verification, and revocation of certificates while ensuring transparency, security, and immutability. These contracts eliminate the need for intermediaries, streamline processes, and reduce the risk of certificate forgery. The smart contract acts as the backbone of the system by securely recording certificate data on the blockchain, managing certificate ownership, and enabling easy verification. By integrating IPFS (InterPlanetary File System) for off-chain storage of certificate files, the system ensures both cost efficiency and decentralization.

4.3.1 Smart Contract Functionalities:

The smart contract was designed to perform essential operations that support the certification lifecycle, including minting new certificates, verifying their authenticity, and revoking them when necessary. Below is an in-depth explanation of each core functionality:

Minting Certificates as NFTs:

- Purpose:

Convert certificates into unique, tamper-proof digital assets stored on the blockchain as Non-Fungible Tokens (NFTs) using the ERC-721 standard.

- Process:

- The issuing institution initiates the certificate creation through the frontend.

- The certificate data (e.g., student name, course, institution, issue date) is input into the system.

- The actual certificate file (PDF, image, etc.) is uploaded to IPFS, and the returned IPFS hash is stored in the smart contract.

- A new NFT is minted representing the certificate, assigned a unique token ID, and linked to the recipient's wallet address.
- Benefits:
- Each certificate is unique and traceable.
- Immutable records once minted, the data cannot be altered.
- Ownership is transparent and verifiable on the blockchain.
- Example Code Snippet:

```
function mintCertificate(
    string memory _studentName,
    string memory _courseName,
    string memory _institutionName,
    string memory _issueDate,
    string memory _ipfsHash
) public {
    certificateCount++;
    _mint(msg.sender, certificateCount);
    certificates[certificateCount] = Certificate(
        certificateCount,
        _studentName,
        _courseName,
        _institutionName,
        _issueDate,
        _ipfsHash,
        false
    );
}
```

Verifying Certificate Authenticity

- Purpose:

Allow third parties (e.g., employers, universities) to verify the authenticity of certificates without relying on the issuing institution.

- Process:
- The verifier scans the QR code or enters the certificate ID into the system.
- The system retrieves the certificate data from the blockchain.
- The corresponding certificate file is fetched from IPFS using the stored hash.
- The system checks the certificate's validity (i.e., whether it has been revoked or not).
- Benefits:
- Instant verification from anywhere in the world.
- Transparency verifiers can independently confirm authenticity.
- Tamper-proof records certificates cannot be altered once minted.
- Example Code Snippet:

```
function verifyCertificate(uint _certId) public view returns (
    string memory, string memory, string memory, string memory, bool
) {
    Certificate memory cert = certificates[_certId];
    return (
        cert.studentName,
        cert.courseName,
        cert.institutionName,
        cert.issueDate,
        cert.isRevoked
    );
```

}

Revoking Certificates if Invalidated:

Purpose:

Provide issuing institutions with the ability to revoke certificates in case of errors, policy violations, or fraud.

- Process:
- The institution selects the certificate to be revoked.
- The smart contract updates the certificate's status to "revoked".
- Any future verification attempts will show the certificate as invalid.
- Benefits:
- Maintains trust and integrity within the certification system.
- Allows for flexible management of certificate records.
- Example Code Snippet:

```
function revokeCertificate(uint _certId) public {
   require(_exists(_certId), "Certificate does not exist");
   certificates[_certId].isRevoked = true;
}
```

Storing IPFS Hashes for Document Retrieval:

Purpose:

To reduce gas costs and blockchain storage limitations, actual certificate files are stored on IPFS. The smart contract only stores the IPFS hash, acting as a reference to retrieve the document when needed.

- Process:
- When a certificate is minted, the document is uploaded to IPFS.

- IPFS returns a unique hash, which is saved in the smart contract.
- During verification, the system uses the stored IPFS hash to fetch the certificate.
- Benefits:
- Off-chain storage reduces blockchain congestion and gas fees.
- Maintains the decentralized nature of the system.
- Ensures easy and public access to the original certificate files.
- Example Code Snippet:

```
function getIPFSHash(uint _certId) public view returns (string memory) {
    require(_exists(_certId), "Certificate does not exist");
    return certificates[_certId].ipfsHash;
}
```

4.3.2 Smart Contract Code Example:

```
solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

contract CertificateNFT is ERC721 {
    uint public certificateCount = 0;

    struct Certificate {
        uint id;
        string studentName;
```

```
    string courseName;
    string institutionName;
    string issueDate;
    string ipfsHash;
    bool isRevoked;
}

mapping(uint => Certificate) public certificates;

constructor() ERC721("CertificateNFT", "CNFT") {}

function mintCertificate(
    string memory _studentName,
    string memory _courseName,
    string memory _institutionName,
    string memory _issueDate,
    string memory _ipfsHash
) public {
    certificateCount++;
    _mint(msg.sender, certificateCount);
    certificates[certificateCount] = Certificate(
        certificateCount,
        _studentName,
        _courseName,
        _institutionName,
        _issueDate,
        _ipfsHash,
```

```
            false
        );
    }

    function verifyCertificate(uint _certId) public view returns (
        string memory, string memory, string memory, string memory, bool
    ) {
        Certificate memory cert = certificates[_certId];
        return (
            cert.studentName,
            cert.courseName,
            cert.institutionName,
            cert.issueDate,
            cert.isRevoked
        );
    }

    function revokeCertificate(uint _certId) public {
        certificates[_certId].isRevoked = true;
    }
}
```

4.3.2 Smart Contract Deployment Using Remix IDE:

The Remix IDE is a powerful, browser-based development environment specifically designed for building, deploying, and testing Solidity smart contracts. It offers a suite of tools that simplify the process of writing, compiling, deploying, and interacting with smart contracts on the Ethereum blockchain. This section outlines the step-by-step process for deploying the NFT-Based Certification Verification Smart Contract using Remix IDE and integrating it with MetaMask for transactions.

1. Open Remix IDE

1. Navigate to the Remix IDE:

- Open your browser and go to Remix IDE.
- No installation is required as it runs entirely in your browser.

2. Set Up the Workspace:

- On the left sidebar, click on the “File Explorer” icon.
- Create a new file by clicking the “+” button and name it CertificateNFT.sol.

3. Activate Required Plugins:

- Go to the “Plugin Manager” (puzzle piece icon).
- Activate the following plugins:
- Solidity Compiler – For compiling the smart contract.
- Deploy & Run Transactions – For deploying and testing the contract.

2. Write & Compile Smart Contract

2.1. Write the Smart Contract

1. Paste the Smart Contract Code into CertificateNFT.sol:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

contract CertificateNFT is ERC721 {
    uint public certificateCount = 0;

    struct Certificate {
        uint id;
```

```
    string studentName;
    string courseName;
    string institutionName;
    string issueDate;
    string ipfsHash;
    bool isRevoked;
}

mapping(uint => Certificate) public certificates;

constructor() ERC721("CertificateNFT", "CNFT") {}

function mintCertificate(
    string memory _studentName,
    string memory _courseName,
    string memory _institutionName,
    string memory _issueDate,
    string memory _ipfsHash
) public {
    certificateCount++;
    _mint(msg.sender, certificateCount);
    certificates[certificateCount] = Certificate(
        certificateCount,
        _studentName,
        _courseName,
        _institutionName,
        _issueDate,
```

```
            _ipfsHash,
            false
        );
    }

    function verifyCertificate(uint _certId) public view returns (
        string memory, string memory, string memory, string memory, bool
    ) {
        Certificate memory cert = certificates[_certId];
        return (
            cert.studentName,
            cert.courseName,
            cert.institutionName,
            cert.issueDate,
            cert.isRevoked
        );
    }

    function revokeCertificate(uint _certId) public {
        require(_exists(_certId), "Certificate does not exist");
        certificates[_certId].isRevoked = true;
    }
}
```

2.2. Compile the Smart Contract

1. Select the Solidity Compiler:

- Click the “Solidity Compiler” tab on the left sidebar.
- Set the compiler version to ^0.8.0 to match the contract.

2. Compile the Contract:

- Press the “Compile CertificateNFT.sol” button.
- If the code has no errors, a green checkmark will appear.
- If there are errors, review the error messages and correct them.

3. Deploy the Contract

3.1. Set Up MetaMask

1. Install MetaMask Extension:

- Download and install the MetaMask browser extension.
- Set up your wallet and store your seed phrase securely.

2. Connect to Ethereum Testnet:

- Open MetaMask and switch to a test network (e.g., Goerli or Sepolia).
- Add Test ETH to your wallet using a Faucet:
- For Goerli: Goerli Faucet
- For Sepolia: Sepolia Faucet

3.2. Deploy the Smart Contract Using Remix

1. Go to the Deploy & Run Transactions Tab:

- In the left sidebar, click the Deploy & Run Transactions icon.

2. Set the Environment:

- Under the Environment dropdown, select injected Web3 (this connects Remix to MetaMask).
- MetaMask will prompt you to connect your wallet—click Connect.

3. Select the Contract:

- Under the Contract dropdown, select CertificateNFT.

4. Deploy the Contract:

- Click the Deploy button.
- MetaMask will pop up asking for confirmation and gas fee approval.
- Click Confirm to complete the deployment.

5. View the Deployed Contract:

- After deployment, the contract's address will appear in the Deployed Contracts section.
- Copy the contract address for future reference.

4. Verify the Contract

After deploying the smart contract, it's essential to verify and interact with it to ensure all functions work as expected.

4.1. Interact with the Contract in Remix

1. Expand the Deployed Contract in Remix.
2. Test the Functions:

- Mint Certificate:
- Input student details and an IPFS hash for the certificate file.
- Click transact and confirm in MetaMask.
- Verify Certificate:
- Enter the certificate ID and click call to check its authenticity.
- Revoke Certificate:
- Input the certificate ID and click "transact" to revoke it.

4.2. Verify Contract on Etherscan (Optional for Public Networks)

1. Go to Etherscan (Goerli Etherscan for Goerli Testnet).
2. Search for Your Contract Address.

3. Click Verify and Publish.

4. Select Compiler Version (e.g., 0.8.0) and copy-paste the Solidity code.

5. Verify after successful verification, your contract will be visible and transparent on the blockchain.

4.3.3 Smart Contract Workflow Diagram

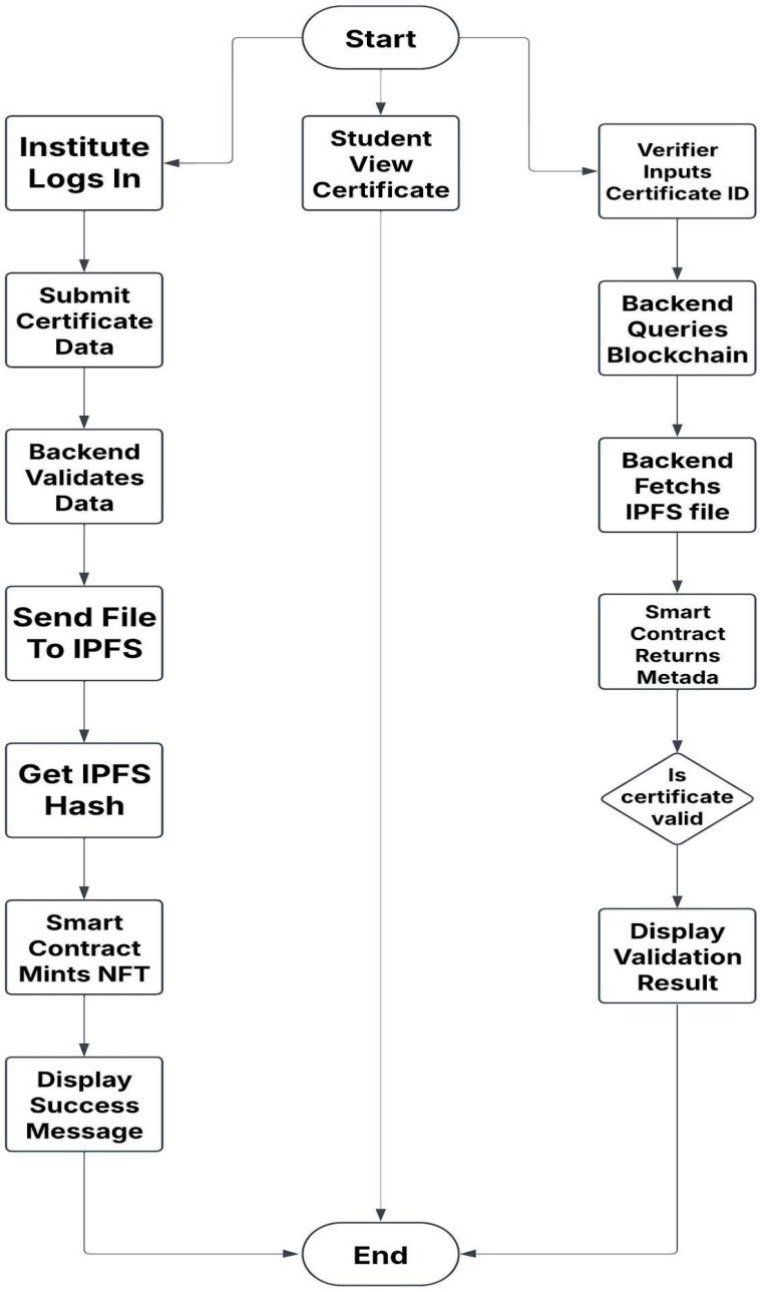

Figure 4.1 Smart Contract Workflow

4.4 Smart Contract Workflow

This diagram represents the workflow of the smart contract-based certificate issuance and verification process within the NFT-based certification system. The workflow is divided into three main interactions:

1. **Certificate Issuance (Left Side):**
    - The institution logs in and submits certificate details.
    - The backend validates the data and sends the certificate file to IPFS.
    - An IPFS hash is generated and stored in the blockchain.
    - A smart contract mints the certificate as an NFT and links it to the student's account.
    - The system displays a success message confirming the issuance.
2. **Student Certificate Access (Middle Path):**
    - Students can log in and view their certificates directly from the system.
3. **Certificate Verification (Right Side):**
    - Verifiers enter the certificate ID to check authenticity.
    - The system queries the blockchain and retrieves the IPFS file.
    - The smart contract returns certificate metadata for verification.
    - The system checks the validity status and displays the validation result.

This automated, decentralized, and tamper-proof approach enhances the security, transparency, and efficiency of certificate verification, eliminating the need for manual verification by institutions.

4.4.1 Key Frontend Features

The frontend of the system comprises several pages, each serving a distinct purpose for different users' students, institutions, and verifiers.

1. Student Dashboard

- Purpose: Enable students to manage their certificates.
- Key Functionalities:
- View Issued Certificates: Students can see a list of all certificates minted under their account.

- Download Certificates: Provides an option to download certificates in PDF format directly from IPFS.

- Check Verification Status: Real-time verification of certificate authenticity by fetching data from the blockchain.

- View NFT Metadata: Students can see the on-chain data linked to their certificates (e.g., token ID, issue date, etc.).

- Example UI Flow:

- Student logs in → Dashboard loads certificates → Select certificate → View/Download/Verify.

2. Institution Dashboard (Admin Panel)

- Purpose: Allow institutions or administrators to manage certificates.

- Key Functionalities:

- Mint New Certificates: Upload student details and mint new certificates as NFTs.

- Revoke Certificates: Select and revoke certificates if necessary, updating their status on the blockchain.

- Upload Certificate Files to IPFS: Store large files off-chain and link their IPFS hash to the smart contract.

- Admin Management: Control permissions, allowing only authorized personnel to mint or revoke certificates.

- Example UI Flow:

- Admin logs in → Upload certificate details → MetaMask transaction confirmation → NFT minted → Stored on Blockchain & IPFS.

3. Verifier Portal

- Purpose: Enable third parties (e.g., employers, academic institutions) to verify certificate authenticity.

- Key Functionalities:

- Verify Certificate by Token ID or QR Code: Verifiers can input a token ID or scan a QR code to check validity.

- View Certificate Metadata: Access essential details (e.g., student name, course, issue date, and status).

- Instant Status Feedback: Immediately confirms whether a certificate is valid, revoked, or not found.
- Fetch Certificate from IPFS: Retrieve the stored file for visual verification.
- Example UI Flow:
- Verifier visits portal → Enters Token ID or scans QR → Blockchain queried → Verification status displayed.

4.4.2 Core Pages of the Website

The system's frontend includes four primary pages to manage the complete lifecycle of certificates:

1. Upload Page (Mint Certificate)

- Purpose: Facilitate certificate issuance by uploading necessary details and files.
- Features:
- Form for Student & Course Details: Student name, course, issue date, etc.
- File Upload (IPFS Integration): Uploads the certificate file and returns an IPFS hash.
- MetaMask Transaction: After filling in the form, MetaMask prompts the admin to confirm the minting transaction.
- Real-Time Feedback: Users receive notifications for successful uploads and minting

2. Verify Page (Check Certificate Status)

- Purpose: Allows any user (student, verifier) to verify the authenticity of certificates.
- Features:
- Input Field for Token ID or QR Code Scanner: Fetches certificate details from the blockchain.
- Display of Certificate Data: Student name, issue date, status (valid/revoked), etc.
- Live Status: Immediately shows whether the certificate is valid or revoked.
- IPFS Integration: Provides a link to view the original certificate stored off-chain.

3. Delete Page (Revoke Certificate)

- Purpose: Enable institutions to revoke or delete certificates if they become invalid.
- Features:
- Search Bar to Find Certificate by ID.
- MetaMask Integration for Transaction Confirmation.
- Updates Blockchain State to Mark as Revoked.
- Real-Time Feedback: Notifies admins when a certificate has been successfully revoked.

4. Admin Page (Institution Dashboard)

- Purpose: Manage the entire certificate system.
- Features:
- List of All Minted Certificates: Token IDs, student names, issuance dates.
- Upload & Delete Options: Quick links for minting or revoking certificates.
- View Certificate Analytics: Number of valid/revoked certificates, activity logs.
- Add/Remove Admins: Manage who has permission to mint or revoke certificates.
- Security Features:
- JWT Authentication for secure admin access.
- MetaMask Integration for all transactions.

4.4.3 Web3.js Integration Code

Example

javascript

```
import Web3 from 'web3';
import CertificateNFT from './CertificateNFT.json';
```

```
const loadBlockchainData = async () => {
  const web3 = new Web3(window.ethereum);
  const accounts = await web3.eth.requestAccounts();
  const networkId = await web3.eth.net.getId();
  const networkData = CertificateNFT.networks[networkId];

  if (networkData) {
    const contract = new web3.eth.Contract(CertificateNFT.abi, networkData.address);
    console.log("Smart Contract Connected:", contract);
  } else {
    alert('Smart contract not deployed on this network');
  }
};

export default loadBlockchainData;
```

4.4.4 Frontend UI Screenshots:

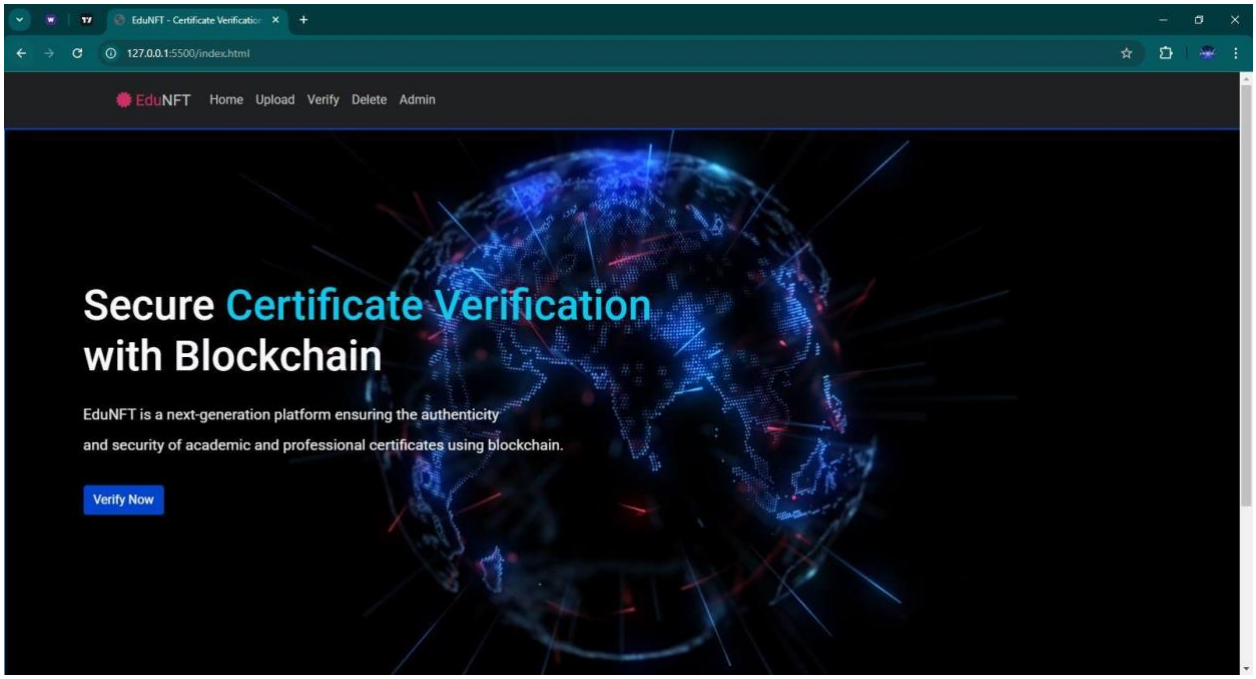

Figure 4.2 Home Page Interface

This page serves as the landing interface for users, providing access to certificate verification, issuance, and system navigation.

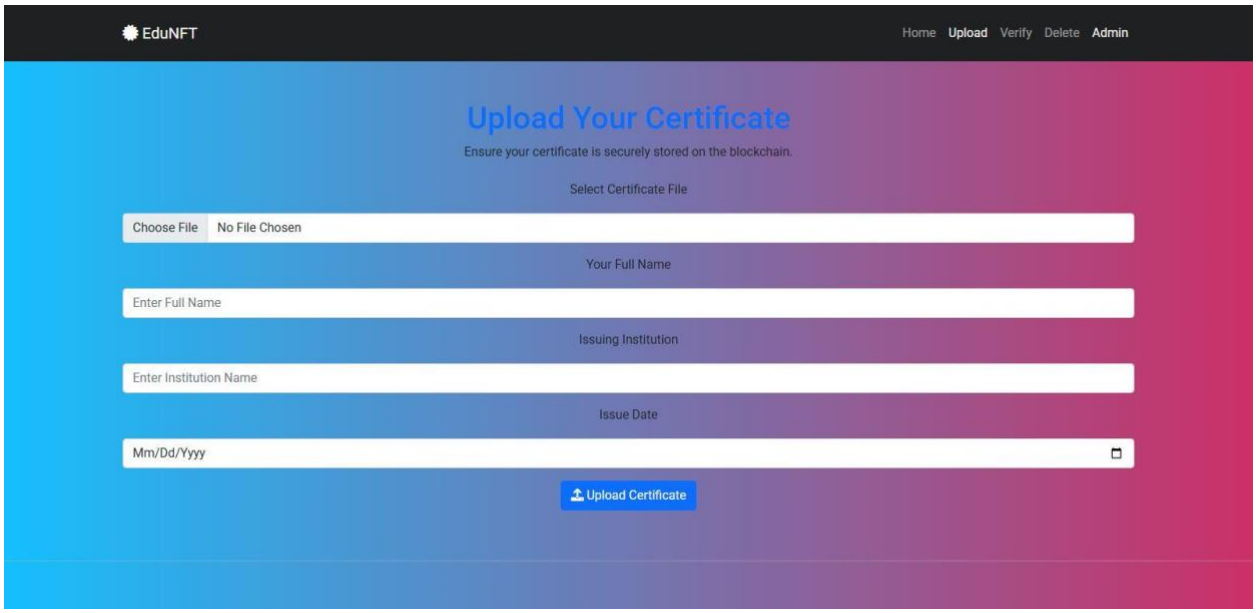

Figure 4.3 Certificate Upload Page

Institutions use this page to enter student details, upload certificate files, and mint NFTs via MetaMask integration.

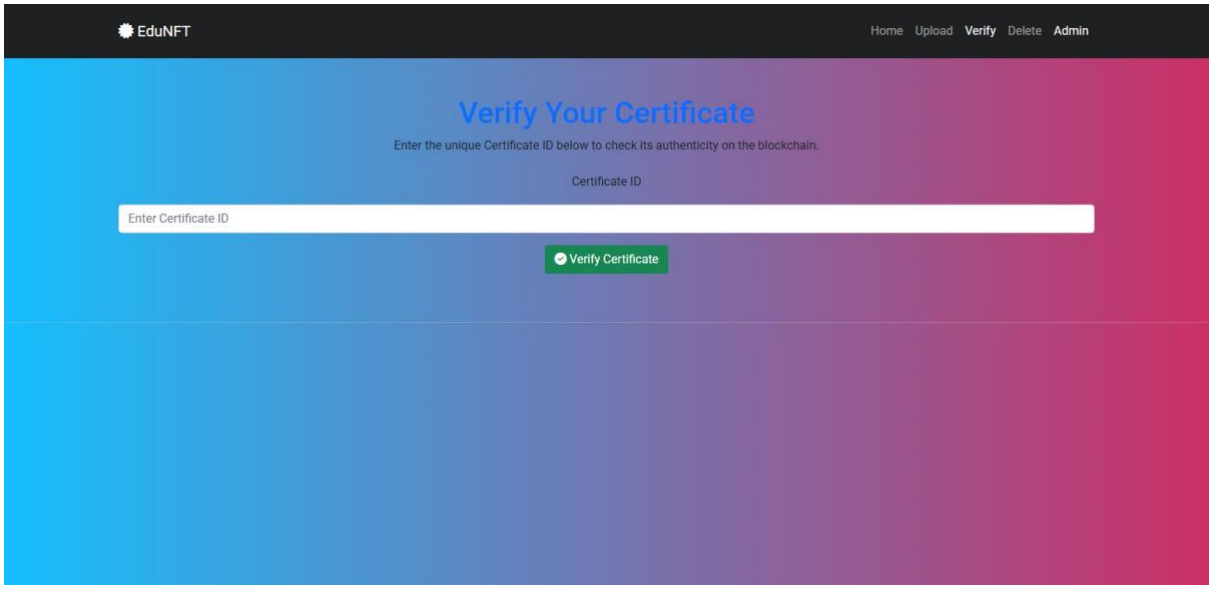

Figure 4.4 Certificate Verification Page

Verifiers can input a certificate ID or scan a QR code to check its authenticity. The system fetches details from the blockchain and IPFS.

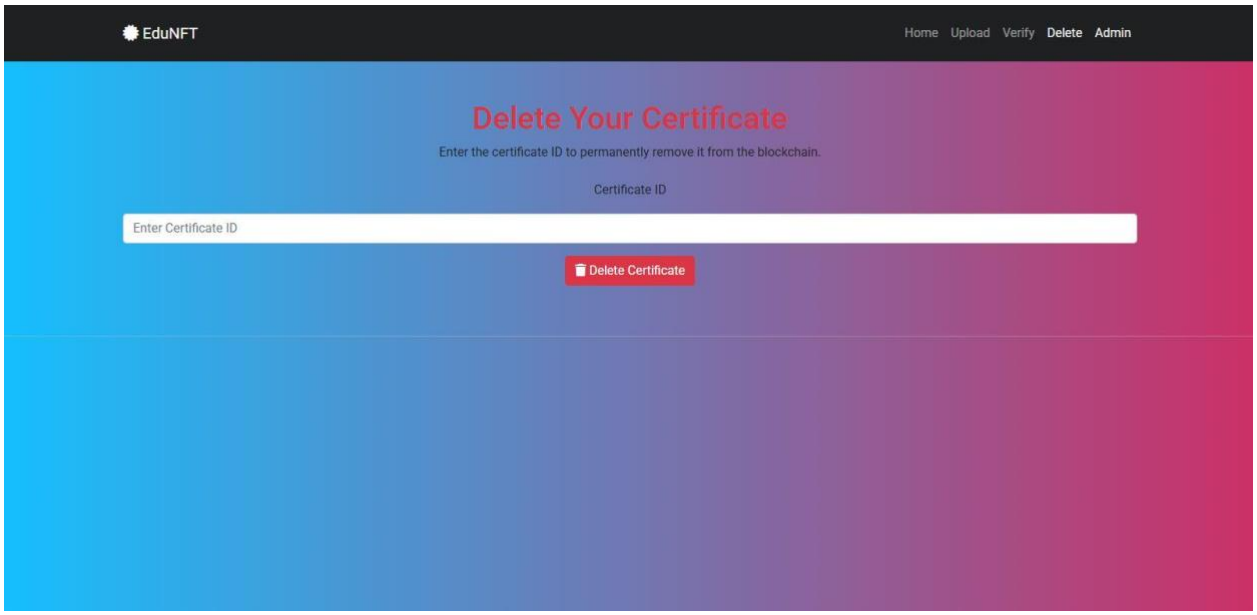

Figure 4.5 Delete or Revoke Certificate Page

Institutions can revoke previously issued certificates if necessary. This interface allows administrators to select a certificate, confirm its revocation, and update its status on the blockchain.

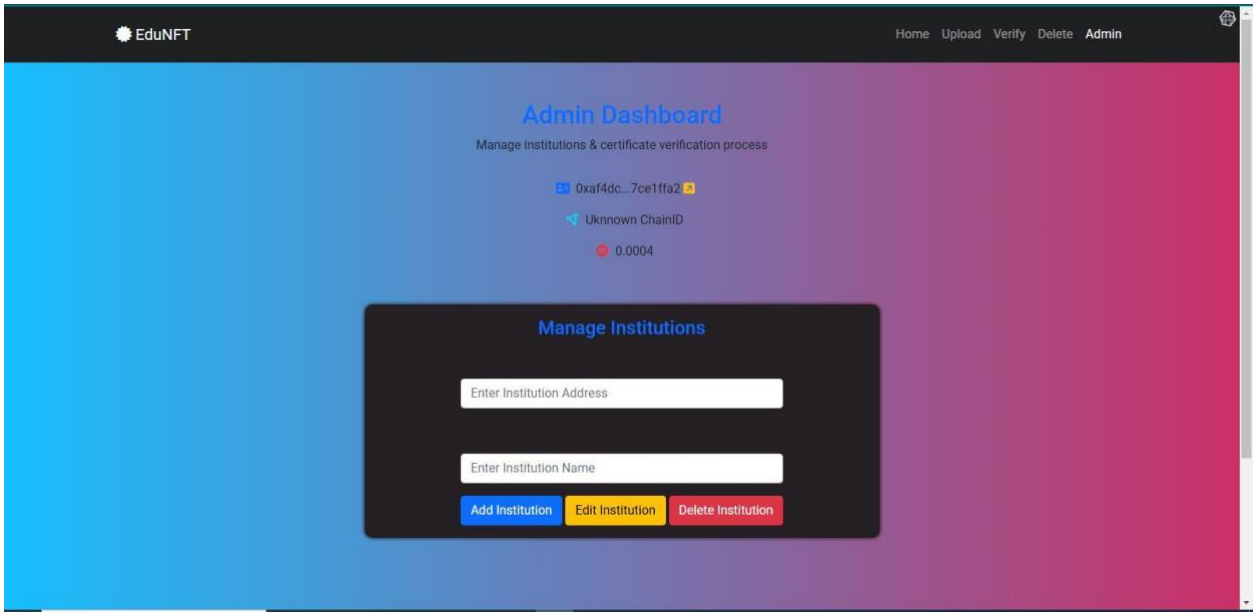

Figure 4.6 Admin Dashboard

This interface serves as the administrative control panel for managing institutions and overseeing the certificate verification process.

4.5 IPFS Integration for Off-Chain Storage

In the NFT-Based Certification Verification System, certificates are stored on the Interplanetary File System (IPFS) to ensure decentralized, secure, and efficient file storage. This architecture optimizes both blockchain performance and user accessibility while minimizing storage costs. Instead of saving large certificate files directly on the blockchain which would be costly and inefficient only the IPFS hash (a unique identifier for the file) is stored on the Ethereum blockchain. This method balances decentralization, security, and scalability.

What is IPFS?

The Interplanetary File System (IPFS) is a peer-to-peer distributed file system that enables decentralized file storage and sharing. Unlike traditional centralized servers, IPFS stores files across a network of nodes, ensuring resilience against data loss and censorship. It is particularly useful in blockchain applications, where minimizing on-chain storage is critical due to gas costs.

- Decentralized: Data is not stored on a single server but distributed across multiple nodes.
- Content-Addressed: Files are identified by a unique cryptographic hash rather than a location or URL.
- Tamper-Proof: Any modification to the file changes its hash, ensuring integrity.

- Permanent Links: As long as the file exists on the IPFS network, it can always be accessed using its unique hash.

4.5.1 Uploading Certificates to IPFS:

javascript

```
import { create } from 'ipfs-http-client';


const ipfs = create('https://ipfs.infura.io:5001');


const uploadFileToIPFS = async (file) => {
  try {
    const result = await ipfs.add(file);
    console.log('IPFS Hash:', result.path);
    return result.path;
  } catch (error) {
    console.error('IPFS Upload Error:', error);
  }
};
```

4.5.2 IPFS Storage Diagram

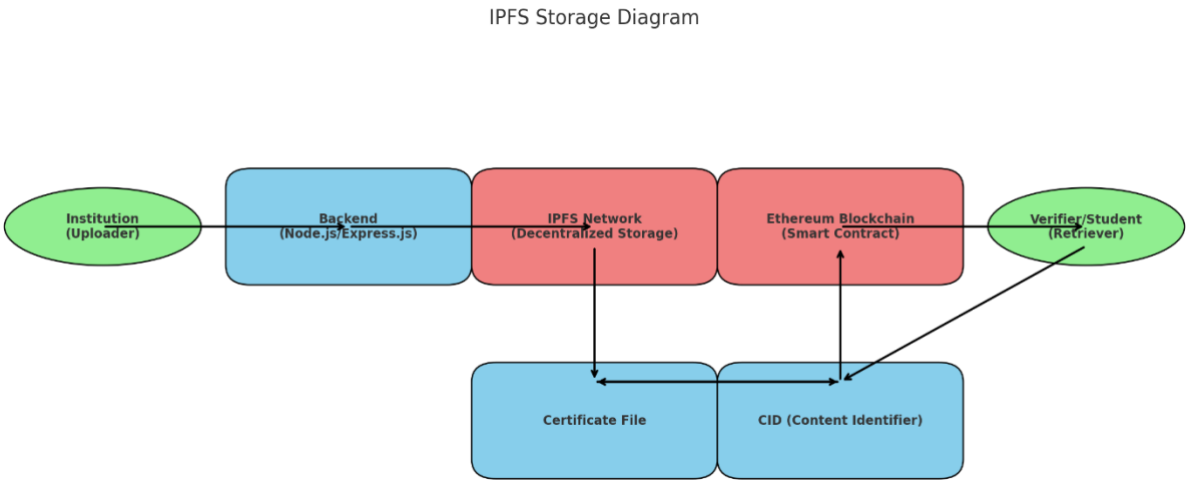

Figure 4.7 IPFS Storage Flow Diagram

This diagram illustrates the decentralized storage process used in the NFT-based certificate verification system. Instead of storing certificate files directly on the blockchain, the system utilizes InterPlanetary File System (IPFS) for off-chain storage while maintaining blockchain immutability.

Process Flow:

1. Institution (Uploader): The institution submits certificate data through the system backend.
2. Backend Processing: The backend (Node.js/Express.js) processes and sends the certificate file to IPFS for decentralized storage.
3. IPFS Network: The certificate file is stored on IPFS, generating a unique Content Identifier (CID).
4. Blockchain Storage: The CID is stored on the Ethereum blockchain via a smart contract, ensuring data integrity and tamper resistance.

5. Verification Process: When a student or verifier queries the system, the CID is fetched from the blockchain and used to retrieve the original certificate from IPFS.

4.6 Deployment

The deployment phase is a critical component of the NFT-Based Certification Verification System, ensuring that all system components including the frontend, backend, blockchain network, and decentralized storage (IPFS) work seamlessly together in a live environment. This system was designed for secure, scalable, and user-friendly deployment, integrating MetaMask for authentication, Remix IDE for smart contract deployment, and platforms like Netlify or Vercel for hosting the frontend. The deployment process covers the following major steps:

1. Smart Contract Deployment on Ethereum Blockchain
2. Frontend Hosting on Platforms like Netlify or Vercel
3. Integration with MetaMask for Wallet Authentication
4. IPFS for Decentralized File Storage

4.6.1 Deployment Architecture Diagram:

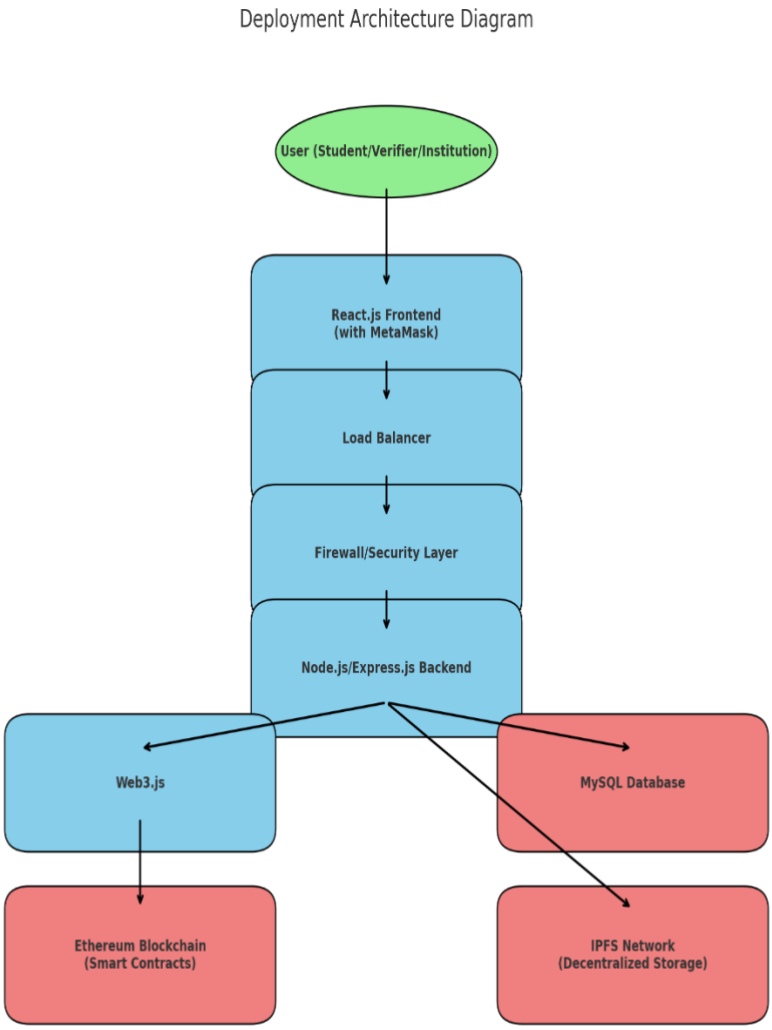

Figure 4.8 Deployment Architecture

This diagram illustrates the deployment structure of the NFT-based certificate verification system. Users (students, verifiers, institutions) interact with the React.js frontend, which connects to the Node.js/Express.js backend via a load balancer and security firewall.

The backend handles requests using Web3.js to interact with the Ethereum blockchain for smart contract operations and MySQL/IPFS for data storage. This architecture ensures scalability, security, and efficient decentralized certificate management.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

With the rapid advancement in technology, traditional methods of issuing and verifying certificates such as paper-based documents, centralized databases, and manual verification processes have proven to be inefficient, highly susceptible to forgery, time-consuming, and resource-intensive. These traditional approaches require significant manual effort for verification and are prone to errors and security breaches. As the demand for transparent and efficient verification systems increases, it has become imperative to explore modern approaches that address these challenges while ensuring authenticity, security, and scalability. To this end, the NFT-Based Certification Verification System was developed to leverage blockchain technology, Non-Fungible Tokens (NFTs), and decentralized file storage to provide a modern, tamper-proof, and transparent approach to certificate issuance and verification. The goal of this project is to provide institutions, students, and third-party verifiers with a decentralized platform that ensures the authenticity of certificates while simplifying the verification process.

The system employs the Object-Oriented Analysis and Design (OOAD) approach for system analysis and design, ensuring a modular and scalable architecture. The entire system was designed as a full web application, incorporating technologies like Ethereum Blockchain, Solidity Smart Contracts, IPFS (InterPlanetary File System) for decentralized storage, React.js for the frontend, and Web3.js for blockchain integration. The process begins with institutions logging into the system using MetaMask for secure authentication. Once authenticated, the institution can mint new certificates by uploading student details and the certificate file. The certificate file is stored on IPFS, and the generated IPFS hash is stored on the Ethereum blockchain through a smart contract. This process ensures that the certificate cannot be altered or forged.

Students can log in to the platform using their wallet addresses via MetaMask and view their issued certificates. Each certificate is represented as an NFT, uniquely tied to the student, and can be downloaded directly from IPFS. The student’s dashboard provides a simple interface for managing and viewing all issued certificates.

For third-party verifiers (e.g., employers or other institutions), the system provides a Verifier Portal. Verifiers can input the unique certificate ID or scan a QR code to instantly verify the certificate's authenticity. The system retrieves the IPFS hash from the blockchain and fetches the actual certificate from IPFS, allowing the verifier to cross-check the document. This process eliminates the need to contact the issuing institution directly, streamlining the verification process. The system also provides administrative controls, allowing authorized personnel to revoke certificates if they become invalidated. Revoked certificates are flagged on the blockchain, and any future verification attempts will display the revoked status to verifiers.

This project successfully bridges the gap between modern blockchain technology and the need for secure, efficient certificate management systems. By combining NFTs, IPFS, and Ethereum Smart Contracts, the platform offers a tamper-proof, transparent, and easily verifiable system for credential issuance and management.

5.2 Conclusion

This project addresses the critical challenges faced in traditional certificate issuance and verification systems by introducing a decentralized, blockchain-based solution. It analyzes the limitations of paper-based certificates, centralized databases, and manual verification processes highlighting issues such as forgery, inefficiency, and data vulnerability. The NFT-Based Certification Verification System provides a robust solution to these challenges by integrating blockchain and NFT technologies with decentralized file storage. The use of Ethereum Blockchain ensures data immutability and transparency, while IPFS reduces on-chain storage costs and provides permanent access to certificate files. The incorporation of smart contracts guarantees that certificates are tamper-proof and verifiable at any time.

The system empowers institutions to issue certificates securely, enables students to manage their academic records with confidence, and provides third-party verifiers with a simple and reliable way to authenticate certificates. The use of NFTs ensures that each certificate is unique and traceable, reducing the risk of forgery and unauthorized duplication. This solution has the potential to transform how certificates are issued and verified globally, offering a transparent and scalable approach to credential management. It not only eliminates the inefficiencies of traditional methods but also provides a more secure and user-friendly experience for all stakeholders involved.

5.2 Recommendations

To enhance the adoption and scalability of the NFT-based certificate verification system, institutions can deploy the system on Layer-2 solutions like Polygon, Arbitrum, or Optimism to reduce Ethereum's high gas fees. Integrating multi-chain support with networks like Binance Smart Chain (BSC), Solana, or Hyperledger Fabric will improve accessibility and flexibility. Ensuring regulatory compliance through collaboration with government agencies and accrediting bodies will help establish legal recognition and standardization.

Raising institutional awareness through workshops and training programs is essential for driving adoption among universities and employers. Enhancing the user experience by simplifying wallet integration, mobile support, and QR-based verification will make the system more accessible. Integrating Decentralized Identity (DID) solutions will further improve security by linking certificates to verifiable digital identities.

To maintain trust and accuracy, automated certificate revocation and expiry mechanisms should be implemented via smart contracts. Building partnerships with universities and employers will encourage real-world adoption. Security can be strengthened with AI-driven fraud detection to prevent certificate forgery. Finally, scalability optimization through batch processing and energy-efficient consensus mechanisms will support large-scale implementation while reducing costs.

By addressing these areas, the system will achieve broader adoption, regulatory recognition, and long-term sustainability in the education and credentialing sectors.

REFERENCE

Abdullahi, M. B., Nura, Z. M., & Jiya, L. M. (2018). Examination Eligibility Verification and Attendance System Using Quick Response Code. *i-manager's Journal on Digital Signal Processing (JDSP)*, 1-9.

Ahmed, M. (2021, January 5). Automatic Attendance System - Advantages Of Automated Attendance System. *MasterSoft*. Retrieved from https://www.iitms.co.in/blog/automated-student-attendance-management-system-for-schools-and-colleges.html

Aisha Talib, B. A.-S., Khadija Abdullah AL-Saidi, & Justin Rajasekaran. (2018, October). Effect of implementing 'QR Code Attendance System' during Examination in SHCT-Limits and Possibilities. *Research Gate*. Retrieved from https://www.researchgate.net/publication/328306138_Effect_of_implementing_'QR_Code_Attendance_System'_during_Examination_in_SHCT-Limits_and_Possibilities

Aishwarya, R. (2023, January 25). Who Invented Exams? - Origin of Exams, History. *Get My Uni*. Retrieved from https://www.getmyuni.com/articles/who-invented-exams

Bawar Ali Abdalkarim, & Dakgun Sakarya. (2022, July). A Literature Review on Smart Attendance Systems. *Research Gate*. Retrieved from https://www.researchgate.net/publication/362405196_A_Literature_Review_on_Smart_Attendance_Systems

Ebenezer Oyebode, & Taiwo Oyedepo. (2021). QR–code attendance system for Ajayi Crowther University. *Computer Science Series (19)*, 23-25.

Fadi Almasalha, & Nael Hirzallah. (2014, January). A Students Attendance System Using QR Code. *Research Gate*. Retrieved from https://www.researchgate.net/publication/275605455_A_Students_Attendance_System_Using_QR_Code

Frederiks, E. (2021). Academic Success: Types of Exams. *University of Southern Queensland*. Retrieved from https://usq.pressbooks.pub/academicsuccess/chapter/types-of-exams/

Hamdan, M. S., Mohd Yasin, M. I., Wan Mohd Nor, W. N., & Abdul Rashid, M. R. (2021, September). QR Code-Based Student Attendance System. *IEEE Xplore*. Retrieved from https://ieeexplore.ieee.org/document/9681472/

Kinkelin, P., Hett, T., Hellebrand, S., & Hellebrand, S. (2020). Securing X.509 Certificates Using Hyperledger Fabric. *IEEE Access*, 8, 140762-140776. DOI: 10.1109/ACCESS.2020.3013483

Szalachowski, P. (2020). SmartCert: An Approach to Improve Digital Certificates Using Smart Contracts. *Proceedings of the 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 29-37. DOI: 10.1109/EuroSPW51379.2020.00012

Tumati, P. (2021). Soulbound Tokens for Secure Certificate Authentication. *International Journal of Blockchain Applications*, 5(3), 45-52.

Zhao, Y., & Si, Z. (2022). NFTCert: NFT-Based Certificates with Online Payment Gateway. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/358763243_NFTCert_NFT-Based_Certificates_With_Online_Payment_Gateway

Wang, L., Zhang, T., & Liu, Y. (2020). Blockchain-Based Educational Certificates: Using NFTs to Streamline Verification. *IEEE Transactions on Learning Technologies*, 13(4), 789-797. DOI: 10.1109/TLT.2020.2998501

APPENDIX

SOURCE CODE

Home Page

```
<div class="loader-wraper">
   <div class="lds-roller">
     <div></div>
     <div></div>
     <div></div>
     <div></div>
     <div></div>
     <div></div>
     <div></div>
     <div></div>
   </div>
 </div>


 <!-- Navbar -->
 <nav class="navbar navbar-expand-lg navbar-light bg-light py-3 navbar-dark">
   <div class="container">
     <a class="navbar-brand" href="index.html">
       <i class="fa-solid fa-certificate home_text"></i>
       <span class="home_text">Edu</span>NFT
     </a>
     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent">
```

```
    <span class="navbar-toggler-icon"></span>
   </button>
   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
     <li class="nav-item">
      <a class="nav-link active" aria-current="page" href="index.html">Home</a>
     </li>
     <li class="nav-item">
      <a class="nav-link " href="upload.html">Upload </a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="verify.html">Verify </a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="delete.html">Delete </a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="admin.html">Admin</a>
    </ul>
   </div>
  </div>
</nav>

<!-- Home Section -->
<section class="home py-5 d-flex align-items-center" id="header">
```

```
  <video autoplay muted loop id="myVideo">
    <source src="./files/earth.mp4" type="video/mp4">
  </video>
  <div class="container text-light py-5" data-aos="fade-right" style="padding-left: 70px;">
    <h1 class="headline text">
      Secure <span class="text-info">Certificate Verification</span> <br>
      with Blockchain
    </h1>
    <p class="para py-3">
      EduNFT is a next-generation platform ensuring the authenticity <br>
      and security of academic and professional certificates using blockchain.
    </p>
    <div class="my-3">
      <a class="btn btn-primary" href="verify.html">Verify Now</a>
    </div>
  </div>
</section>



<!-- Footer -->
<footer class="footer-dark">
  <div class="container">
    <div class="row">
      <div class="col-sm-6 col-md-3 item">
        <h3>Services</h3>
```

```
        <ul>
          <li><a href="#">Ethereum Blockchain</a></li>
          <li><a href="#">Smart Contracts</a></li>
          <li><a href="#">NFT-based Authentication</a></li>
        </ul>
      </div>
      <div class="col-md-6 item text">
        <h3><i class="fa-solid fa-certificate"></i> EduNFT</h3>
        <p>EduNFT is a trusted platform for verifying academic documents securely using blockchain technology.</p>
      </div>
    </div>
    <p class="copyright">EduNFT © 2025</p>
  </div>
</footer>
```

Upload Page

```
<!-- Upload Section -->
<section class="upload-section py-5">
  <div class="container text-center">
    <h1 class="text-primary">Upload Your Certificate</h1>
    <p class="text-muted">Ensure your certificate is securely stored on the blockchain.</p>
```

```
<form id="uploadForm" class="upload-form mt-4">
    <div class="mb-3">
        <label for="certificateFile" class="form-label">Select Certificate File</label>
        <input type="file" class="form-control" id="certificateFile" accept=".pdf,.png,.jpg,.jpeg" required>
    </div>

    <div class="mb-3">
        <label for="fullName" class="form-label">Your Full Name</label>
        <input type="text" class="form-control" id="fullName" placeholder="Enter full name" required>
    </div>

    <div class="mb-3">
        <label for="institution" class="form-label">Issuing Institution</label>
        <input type="text" class="form-control" id="institution" placeholder="Enter institution name"
            required>
    </div>

    <div class="mb-3">
        <label for="issueDate" class="form-label">Issue Date</label>
        <input type="date" class="form-control" id="issueDate" required>
    </div>

    <button type="submit" class="btn btn-primary">
```

```
            <i class="fa-solid fa-upload"></i> Upload Certificate
        </button>
    </form>

    <div id="uploadStatus" class="mt-3"></div>
  </div>
</section>
```

Verify Page

```
<!-- Verification Section -->
<section class="verify-section py-5">
  <div class="container text-center">
    <h1 class="text-primary">Verify Your Certificate</h1>
    <p class="text-muted">Enter the unique Certificate ID below to check its authenticity on the blockchain.</p>

    <form id="verifyForm" class="verify-form mt-4">
      <div class="mb-3">
        <label for="certificateID" class="form-label">Certificate ID</label>
```

```
            <input type="text" class="form-control" id="certificateID" placeholder="Enter certificate ID"
                required>
        </div>

        <button type="submit" class="btn btn-success">
            <i class="fa-solid fa-check-circle"></i> Verify Certificate
        </button>
    </form>

    <div id="verifyStatus" class="mt-3"></div>
  </div>
</section>
```

Delete Page

```
<!-- Delete Certificate Section -->
<section class="delete-section py-5">
  <div class="container text-center">
    <h1 class="text-danger">Delete Your Certificate</h1>
    <p class="text-muted">Enter the certificate ID to permanently remove it from the blockchain.</p>

    <form id="deleteForm" class="delete-form mt-4">
      <div class="mb-3">
        <label for="certificateID" class="form-label">Certificate ID</label>
        <input type="text" class="form-control" id="certificateID" placeholder="Enter certificate ID"
```

```
                required>
            </div>

            <button type="submit" class="btn btn-danger">
                <i class="fa-solid fa-trash"></i> Delete Certificate
            </button>
        </form>

        <div id="deleteStatus" class="mt-3"></div>
    </div>
</section>
```

Admin Page

```
<!-- Admin Dashboard -->
<div class="container py-5">
    <h2 class="text-center text-primary">Admin Dashboard</h2>
    <p class="text-center text-muted">Manage institutions & certificate verification process</p>

    <!-- Blockchain Status -->
    <div class="row content m-1">
        <div class="col-lg-12 wallet-status py-3 text-center">
            <i class="fa-brands fa-connectdevelop icon"></i>
            <p id="userAddress">Address: <span class="text-warning">n/a</span></p>
            <p id="network">Network: <span class="text-warning">n/a</span></p>
            <p id="userBalance">Balance: <span class="text-warning">n/a</span></p>
```

```
    </div>
  </div>

  <!-- Admin Actions -->
  <div class="row justify-content-center mt-4">
    <div class="col-lg-6 py-3 data-upload">
      <h4 class="text-center text-primary">Manage Institutions</h4>
      <form id="adminForm">
        <div class="mb-3">
          <label for="Exporter-address" class="form-label">Institution Address</label>
          <input id="Exporter-address" type="text" class="form-control"
            placeholder="Enter Institution Address" required>
        </div>
        <div class="mb-3">
          <label for="info" class="form-label">Institution Name</label>
          <input type="text" id="info" class="form-control" placeholder="Enter Institution Name" required>
        </div>
        <p id="note" class="text-danger"></p>

        <div class="text-center mt-3">
          <button type="button" onclick="addExporter()" class="btn btn-primary">Add Institution</button>
          <button type="button" onclick="editExporter()" class="btn btn-warning">Edit Institution</button>
          <button type="button" onclick="deleteExporter()" class="btn btn-danger">Delete
```

```
                    Institution</button>
                </div>
            </form>
        </div>
    </div>
</div>
```

Blockchain Integration and Smart Contract Interaction

Web3.js Setup (Connecting Frontend to Blockchain)

```
// web3Setup.js
import Web3 from 'web3';

let web3;
if (window.ethereum) {
    web3 = new Web3(window.ethereum);
    try {
        // Request account access if needed
        await window.ethereum.enable();
    } catch (error) {
        console.error("User denied account access");
    }
```

```
} else if (window.web3) {
    web3 = new Web3(window.web3.currentProvider);
} else {
    console.log("No Ethereum browser detected. Try MetaMask!");
}

export default web3;
```

MetaMask Authentication

```
const connectMetaMask = async () => {
    if (window.ethereum) {
        try {
            await window.ethereum.request({ method: 'eth_requestAccounts' });
            console.log('Connected to MetaMask');
        } catch (err) {
            console.error('User denied MetaMask connection:', err);
        }
    } else {
        console.error('Please install MetaMask');
    }
};
```

Upload Certificate (Front End Code)

```
// upload.js
```

```
import web3 from './web3Setup';
import ipfs from './ipfsSetup';
import contractABI from './contractABI.json';

const contractAddress = ' 0x97a7F5AE4BEdEb1349e5CFf76ec0fD90176adFb9';
const contract = new web3.eth.Contract(contractABI, contractAddress);

const uploadCertificate = async (file, userName, institution) => {
  try {
    const accounts = await web3.eth.getAccounts();

    // Upload file to IPFS
    const added = await ipfs.add(file);
    const ipfsHash = added.path;

    // Hash file using SHA-3
    const reader = new FileReader();
    reader.onloadend = async () => {
      const fileHash = web3.utils.sha3(reader.result);

      // Store hash and IPFS link on the blockchain
      await contract.methods.addDocHash(fileHash, ipfsHash)
        .send({ from: accounts[0] });

      console.log('Certificate uploaded successfully:', ipfsHash);
```

```
        };
        reader.readAsArrayBuffer(file);
    } catch (err) {
        console.error('Upload failed:', err);
    }
};

// Event Listener
document.getElementById('uploadForm').addEventListener('submit', async (e) => {
    e.preventDefault();
    const file = document.getElementById('certificateFile').files[0];
    const userName = document.getElementById('fullName').value;
    const institution = document.getElementById('institution').value;
    await uploadCertificate(file, userName, institution);
});
```

Verify Certificate (Front End Code)

```
// verify.js
import web3 from './web3Setup';
import contractABI from './contractABI.json';

const contractAddress = ' 0x97a7F5AE4BEdEb1349e5CFf76ec0fD90176adFb9';
const contract = new web3.eth.Contract(contractABI, contractAddress);
```

```
const verifyCertificate = async (certificateID) => {
    try {
        const certData = await contract.methods.findDocHash(certificateID).call();

        if (certData.blockNumber !== '0') {
            console.log('Certificate is valid!');
            console.log(`Issuer: ${certData.info}`);
            console.log(`Issued At: ${new Date(certData.minetime * 1000)}`);
            console.log(`IPFS Link: https://ipfs.io/ipfs/${certData.ipfs_hash}`);
        } else {
            console.log('Invalid or Non-existent Certificate');
        }
    } catch (err) {
        console.error('Verification failed:', err);
    }
};

// Event Listener
document.getElementById('verifyForm').addEventListener('submit', async (e) => {
    e.preventDefault();
    const certID = document.getElementById('certificateID').value;
    await verifyCertificate(certID);
});
```

Delete Certificate (Front End Code)

```
// delete.js
import web3 from './web3Setup';
import contractABI from './contractABI.json';

const contractAddress = ' 0x97a7F5AE4BEdEb1349e5CFf76ec0fD90176adFb9';
const contract = new web3.eth.Contract(contractABI, contractAddress);

const deleteCertificate = async (certificateID) => {
    try {
        const accounts = await web3.eth.getAccounts();
        await contract.methods.deleteHash(certificateID)
            .send({ from: accounts[0] });
        console.log('Certificate deleted successfully');
    } catch (err) {
        console.error('Deletion failed:', err);
    }
};

// Event Listener
document.getElementById('deleteForm').addEventListener('submit', async (e) => {
```

```
    e.preventDefault();
    const certID = document.getElementById('certificateID').value;
    await deleteCertificate(certID);
});
```