**ABSTRACT**

The Blockchain-Based Certificate Verification System is a secure and easy way to store and check educational and professional credentials. It uses blockchain technology to make sure certificates cannot be changed or faked. Users can save the digital fingerprints (hashes) of their certificates on a blockchain, which keeps a permanent record of issuance and verification. This system allows for quick and reliable checking by comparing the stored hashes with the ones provided. It also offers a public, verifiable record of when certificates were issued and any changes made. By simplifying the verification process, the system reduces fraud, improves efficiency, and builds trust in credential verification. At its core, the system uses cryptographic hashing to generate unique digital fingerprints for each certificate. These hashes are then securely stored on a blockchain, a decentralized and immutable digital ledger. Once recorded, these digital fingerprints cannot be altered or deleted, ensuring the authenticity and integrity of the stored credentials. Whenever a certificate needs verification, the system compares the hash of the presented certificate with the one stored on the blockchain. If they match, the certificate is confirmed as genuine.

**CONTENTS**

**CHAPTER 1: INTRODUCTION PAGE NO**

1.1 Overview 01

1.2 Literature Review 02-04

1.3 Problem Statement 04-05

1.4 Objectives 05

1.5 Scope of Project 06

**CHAPTER 2: METHODOLOGY**

2.1 Methods and Techniques 06-10

2.2 Tools and Techniques 10

2.3 Project Timeline 11-12

**CHAPTER 3: SYSTEM DESIGN AND IMPLEMENTATION**

3.1 System Architecture 13-15

3.2 Component Design 15-16

3.3 Implementation Details 16-21

**CHAPTER 4: RESULTS AND DISCUSSION**

4.1 Presentation of Results 22-26

4.2 Analysis and Result 27

4.3 Comparison with Expectations 28

**CHAPTER 5: CONCLUSION**

5.1 Summary of Findings 29

5.2 Limitations 29

5.3 Future Work 29-30

**REFERENCES**  31

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2.1.1 | Methodology Design | 07 |
| 2 | 2.1.2 | Use Case Diagram | 07 |
| 3 | 3.1.1 | Architecture | 13 |
| 4 | 4.1.1 | Dashboard | 22 |
| 5 | 4.1.2 | Upload Certificate | 22 |
| 6 | 4.1.3 | Verify Certificate | 23 |
| 7 | 4.1.4 | MetaMask Connect | 23 |
| 8 | 4.1.5 | Certificate Issued | 24 |
| 9  10  11  12  13  14  15  16 | 4.1.6  4.1.7  4.1.8  4.1.9  4.1.10  4.1.11  4.2.1  4.2.2 | Transaction Request  Transaction Confirmation  Transaction Preview in Ganache  Deployed Smart Contract  Verification of Valid Certificate  Verification of Invalid Certificate  Factors Over time  Global Forecast to 2029 | 24  25  25  26  26  26  27  27 |

**Sl. No Figure. No Description Page. No**

**CHAPTER 1**

**INTRODUCTION**

In today's digital age managing and verifying educational certificates and professional credentials in traditional systems present several technical and operational challenges. Conventional methods rely heavily on centralized databases, which are vulnerable to unauthorized access, data breaches, and tampering. Additionally, the verification process typically involves manual intervention, which is time-consuming, prone to errors, and lacks transparency. This reliance on centralized intermediaries often creates inefficiencies and raises questions about the trustworthiness of credentials.

To address these issues, blockchain technology offers a robust solution. Blockchain, with its decentralized and distributed ledger, ensures immutability, transparency, and enhanced security of stored data. By using cryptographic algorithms, sensitive information such as certificate details can be securely hashed and stored on the blockchain. This approach eliminates the risk of unauthorized modifications while ensuring that certificates are verifiable by any authorized party. The immutability of blockchain makes it an ideal choice for building a secure and reliable certificate management system, revolutionizing how credentials are stored and authenticated.

**1.1 Overview**

The primary goal of this project is to design and implement a blockchain-based platform for storing and verifying educational certificates and professional credentials. By leveraging blockchain technology, the system aims to ensure data integrity, security, and transparency, eliminating the risk of tampering or forgery. This solution not only streamlines the verification process for stakeholders such as employers and educational institutions but also empowers certificate holders with secure and easily accessible digital credentials.

The subsequent sections of this report provide a comprehensive overview of the system's architecture, design, and implementation. They highlight the technical decisions, challenges encountered, and solutions adopted during the development process. Through this project, we aim to deliver a robust and scalable solution for certificate management, fostering trust and efficiency in academic and professional ecosystems.

**1.2 Literature Review**

**1. Educational Certificate Verification System Using Blockchain (2020)** – Published in the International Journal of Scientific & Technology, this study by Dinesh Kumar K, Senthil P, and Manoj Kumar D.S. **[1]** focuses on using blockchain storage, smart contracts, and decentralization to ensure security in certificate verification. However, the proof-of-work systems used can be costly and energy-intensive.

**2. Certificate Validation Using Blockchain (2020**) – Published in IEEE, A. Gayathiri, J. Jayachitra, and S. Matilda **[2]** explore converting physical certificates into digital forms, generating hash values, and storing them on a blockchain. Challenges include enhancing user adoption and addressing interoperability issues with existing systems.

**3. Blockchain-Based Academic Certificate Verification System (2022)** – Published in Springer by Shivani Pathak, Vimal Gupta, Nitima Malsa, Ankush Ghosh, and R.N. Shaw, **[3]** the study highlights blockchain's decentralized and immutable nature for fraud prevention and process streamlining. However, it lacks a detailed discussion on how blockchain will interact with other existing systems.

**4. Smart and Secure Certificate Validation System through Blockchain (2020)** – In this IEEE publication, Padmavati E. Gundgurti and colleagues **[4]** emphasize the role of smart contracts in automating the verification process. The study identifies challenges in integrating blockchain technology with legacy systems.

**5. Blockchain-Based Certificate Verification System Management (2022)** – Authored by Qurotul Aini, Eka Purnama Harahap, Nuke Puji Lestari Santoso, Siti Nurindah Sari, and Po Abas Sunarya, **[5]** the paper discusses certificate structure extension, performance evaluation, and comparison with existing methods. Identified gaps include scalability, privacy, data security, network latency, and real-time response issues.

**6. Blockchain and Smart Contract for Digital Certificate (2018)** – Published in IEEE by Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen, **[6]** the paper explores generating digital versions of certificates, calculating hash values, and using QR codes for verification. Challenges include user training, ease of use, and long-term maintenance.

**7. Academic Certificate Fraud Detection System Framework Using Blockchain Technology (2022)** – Published in B-Front by Ninda Lutfiani, Desy Apriani, Efa Ayu Nabila, and Hega Lutfilah Juniar, **[7]** the study emphasizes using digital signatures and timestamps for verification. Gaps include the lack of practical implementation, legal and regulatory hurdles, and low user adoption rates. s

**8. Digital Certificate Authority with Cybersecurity in Education (2021)** – Published in IJCITSM by Giandari Maulani, Gunawan, Leli Nirmalasari, Efa Ayu Nabila, and Windy Yestina Sari, **[8]** the paper focuses on addressing cybersecurity threats and enhancing trust through blockchain-based databases. However, the study highlights cost implications for implementation as a significant challenge.

**9. Implementation of Blockchain for Academic Certificate Authentication (2020)** – Published in the Journal of Computer Networks by Charles Kim and Fatima Rahman, **[9]** this study combines QR code-based validation with blockchain to secure academic credentials. Key limitations include slow transaction speeds and insufficient privacy protection in data handling.

**10. Blockchain-Powered Verification of Educational Credentials (2021)** – Featured in the Computers & Security Journal by Peter Johnson and Wei Zhang, **[10]** the paper introduces a blockchain protocol utilizing smart contracts for automated validation. However, synchronization issues across nodes and high network maintenance costs pose significant challenges.

**11. Blockchain in Higher Education: Verification of Certificates (2022)** – Published in the Educational Technology Journal by Kevin Roberts and Rina Gupta, **[11]** the study proposes a hybrid approach .

**12. Blockchain-Based Certificates in Professional Development and Skill Recognition (2023)** – Published in the Journal of Blockchain Research and Development by Michael S. Robinson and Clara Hernandez, **[12]** this study focuses on leveraging smart contracts to automate certificate verification and provide verifiable proof of skills and qualifications. The research highlights the potential of blockchain technology in professional skill recognition systems. However, significant challenges remain, including the lack of infrastructure and universally accepted standards, which hinder widespread adoption and seamless integration across different platforms and industries

**Literature Summary**

The literature review covers various studies focused on blockchain-based certificate verification systems, highlighting the benefits and challenges of using blockchain for this purpose. Most studies leverage blockchain’s decentralized and immutable nature, along with smart contracts, to automate and secure the verification process. Key benefits include improved trust, prevention of fraud, and streamlined verification. However, common challenges persist, such as the integration of blockchain with legacy educational systems, scalability issues, and high energy costs associated with some consensus mechanisms like proof-of-work. Additionally, user adoption, awareness, and interoperability with existing systems are often cited as barriers. Legal and regulatory challenges, slow transaction speeds, and high network maintenance costs further complicate implementation. While newer approaches, like hybrid models using public and private blockchains, aim to balance security and performance, practical implementation and standardization remain significant gaps in the field.

**Literature Gaps**

• The gaps in blockchain-based certificate verification systems, as observed across various research projects, highlight several critical challenges that need to be addressed for widespread adoption and functionality.

• Key issues include the high cost and energy consumption of proof-of-work systems, as well as the difficulty in achieving interoperability with existing legacy systems.

 • Furthermore, scalability remains a persistent concern, particularly in terms of network latency, real-time response, and the ability to handle large volumes of certificates.

 • Privacy and data security pose significant risks, as blockchain’s transparency can inadvertently expose sensitive information.

 • Additionally, user adoption is hindered by a lack of awareness and training, while legal and regulatory frameworks remain underdeveloped, creating uncertainty around long term management and enforcement.

• Finally, the high maintenance costs and synchronization issues across multiple nodes further complicate the practical implementation of these systems.

• Addressing these gaps is essential for making blockchain-based certificate verification systems more efficient, secure, and accessible.

**1.3 Problem Statement**

The traditional way of verifying educational and professional certificates is slow and easy to tamper with, causing fraud and delays. This creates extra work for schools, employers, and individuals. We need a digital solution for secure, transparent, and efficient verification. Blockchain can provide unchangeable records, reducing fraud and improving efficiency.

**1.4 Objectives**

* To design and develop user interface for certificate holders, issuers, and verifiers .Enable easy input of certificate details during issuance.
* To develop and handle certificate data and verifications on the blockchain using the smart contract.
* To test and deploy the certificate verification system on the virtual blockchain platform.

**1.5 Scope of the Project**

1. For Certificate Holders:
2. Convenient access to securely stored certificates at any time.
3. Easy sharing of certificate details with employers or educational institutions for verification.
4. For Verifiers (Employers/Educational Institutions):
5. Real-time verification of certificate authenticity.
6. Reduced risk of fraudulent certificates through blockchain's immutable nature.
7. For Issuers (Educational Institutions/Certification Bodies):
8. Streamlined certificate issuance process with secure record-keeping.
9. Reduced administrative burden by eliminating manual verification requests.
10. Technology Integration:
11. Use of blockchain for immutable and secure data storage.
12. Scalability to accommodate future enhancements, such as integration with external systems.
13. Community Impact:
14. Increased trust and transparency in certificate verification processes.
15. Support for combating certificate fraud, fostering integrity in academic and professional sectors.

**CHAPTER 2**

**METHODOLOGY**

**2.1 Methods and Techniques**

The development of the **Blockchain-Based Certificate Verification System** will follow a structured and systematic approach to ensure efficient progress and the achievement of project goals. The project will be divided into several phases: **Planning and Requirement Gathering**, **Design**, **Development**, **Testing**, **Deployment**, and **Maintenance**. Below is a detailed outline of each phase:

**1. Planning and Requirement Gathering**

* **Stakeholder Meetings**: Conducted discussions with key stakeholders such as educational institutions, certification bodies, and employers to gather requirements and understand their needs for certificate verification.
* **Requirement Documentation**: Documented functional and non-functional requirements, outlining the essential features such as certificate upload, verification, and storage on the blockchain.
* **Project Planning**: Developed a detailed project plan, including timelines, milestones, and resource allocation to ensure smooth execution of the project.

**2. Design**

* **System Architecture**:
* A simple system architecture will be defined, consisting of users (issuers, holders, verifiers), a backend API, blockchain for storing certificate hashes, and the front-end interface for interacting with the system.

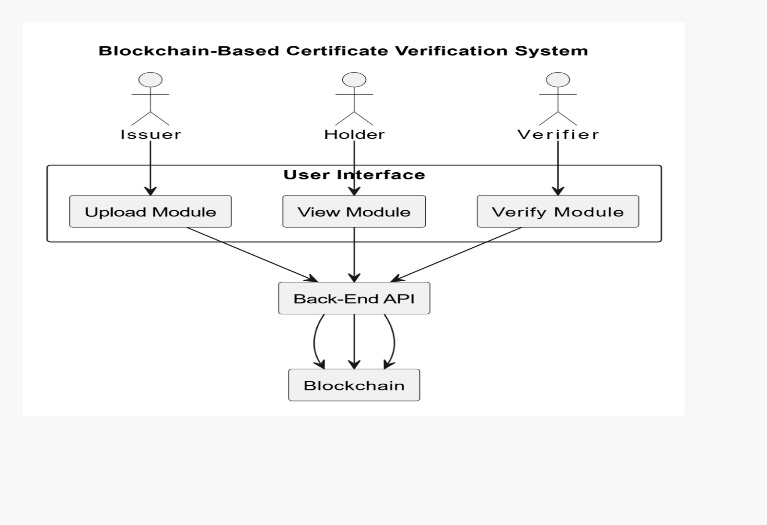


Fig.2.1.1 Methodology Design

**Use Case Diagram**: A  use  case  diagram will  outline the interactions  between  different system  components  such as users, certificates, blockchain.

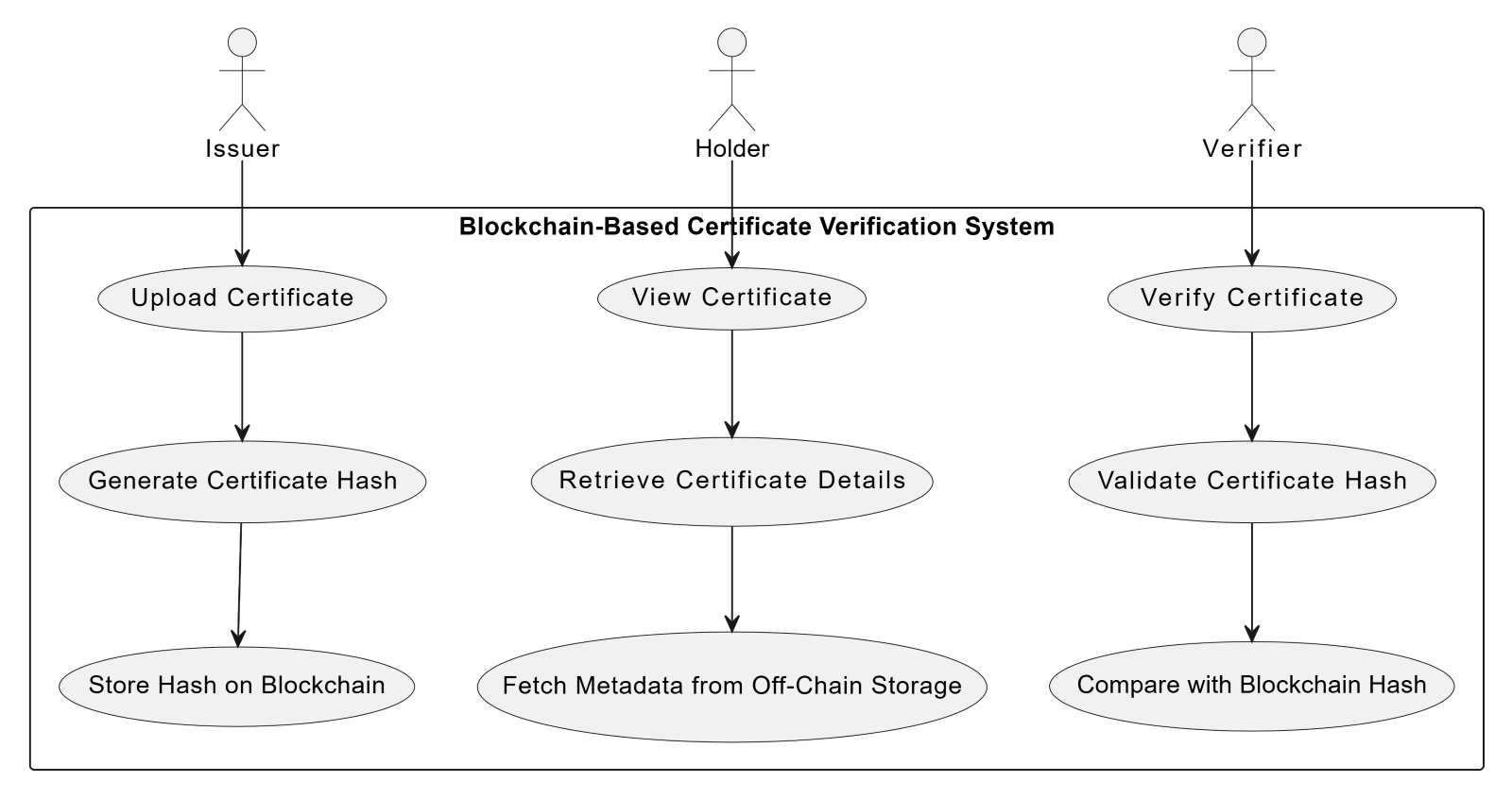


Fig.2.1.2 Use case diagram

* **Technical Specifications**:

Documenting the technologies, frameworks, and tools to be used, such as Solidity for smart contracts and Ethereum for blockchain.

* **Database Design**:

The system does not require a traditional relational database but will focus on storing metadata off-chain using appropriate storage solutions like cloud storage for certificates, linked with blockchain hashes.

**3. Development**

* **Set Up Development Environment**:
* Install and configure tools such as **Ganache** (for local blockchain development), **Truffle Suite** (for smart contract deployment), **Metamask** (for user interaction with blockchain), and **VS Code** as the code editor.
* Set up version control using **Git** and host the project on **GitHub** for collaboration and version tracking.
* **Frontend Development**:
* **HTML/CSS/JavaScript**: Create an intuitive user interface that allows certificate upload, viewing, and verification.
* **React**: Implement the dynamic frontend using **React.js** for a responsive user experience.
* **Backend Development**:
* **Smart Contract Setup**: Develop and deploy smart contracts using **Solidity** on the Ethereum blockchain to store the cryptographic hash of certificates.
* **Blockchain Integration**: Integrate the blockchain network with the system to securely store certificate data.
* **User Authentication**: Implement secure authentication for users (issuers, holders, verifiers) using **Metamask** for blockchain interaction.
* **Certificate Upload and Verification**: Implement APIs to allow issuers to upload certificates and verifiers to validate them against the blockchain.

**4. Testing**

* **Unit Testing**:
* Write unit tests for individual components, including smart contracts, APIs, and UI elements to ensure they work as expected.
* **Integration Testing**:
* Perform tests to ensure that the interaction between the blockchain, backend, and frontend is seamless and secure.
* **System Testing**:
* Conduct end-to-end testing to validate the entire certificate verification process, from uploading to verification.
* **User Acceptance Testing**:
* Have real users (issuers, holders, verifiers) test the system to ensure it meets their requirements and functions as expected.

**5. Deployment**

* **Deploy Blockchain Network**:
* Deploy the smart contracts and the blockchain network on the **Ethereum testnet** (using **Ganache**) for initial testing and then on the **Ethereum mainnet** for production.
* **Deploy Frontend and Backend**:
* Deploy the frontend on a cloud platform like **Netlify** and backend APIs on **Heroku** or another suitable server.
* **Production Configuration**:
* Configure production settings for environment variables, database connections, and blockchain interaction.

**6. Maintenance**

* **Bug Fixes**:
* Continuously address any issues or bugs reported by users, especially regarding certificate upload, verification, or blockchain interactions.
* **Feature Enhancements**:
* Implement new features based on user feedback, such as adding support for new certificate types or enhancing the verification process.
* **Performance Optimization**:
* Optimize the system for better performance, ensuring it can handle large numbers of certificates and verifications efficiently.

**2.2 Tools and Technologies**

**Blockchain Platform:**

* Ethereum: A decentralized platform used for deploying smart contracts to ensure immutable and tamper-proof storage of certificate hashes.

**Smart Contract Development:**

* Solidity: A programming language used to write and deploy smart contracts on the Ethereum blockchain.

**Front-End Development:**

* HTML/CSS: For creating the structure and styling of the user interface.
* JavaScript: For building dynamic functionalities in the user interface.
* React.js: A JavaScript library for creating interactive and reusable UI components.

**Back-End Development:**

* Node.js: A JavaScript runtime for building scalable server-side applications.

**Blockchain Interaction:**

* Web3.js: A JavaScript library used to interact with the Ethereum blockchain and smart contracts.

**Development and Testing Tools:**

* Ganache: A personal Ethereum blockchain used for deploying and testing smart contracts in a local environment.
* MetaMask: A browser extension for interacting with the Ethereum blockchain, enabling transactions and contract interactions.

**2.3 Project Timeline**

**Week 1: Planning and Initial Setup**

Project Planning

* Define project scope and requirements.
* Collaborate with stakeholders to gather requirements.
* Document user stories and acceptance criteria.

Environment Setup

* Set up the development environment (e.g., install necessary tools and frameworks).
* Initialize the blockchain project and app.
* Configure settings and create initial database migrations.

Basic Models and Admin Interface

* Define models for certificates, users, and verification logs.
* Register models with the admin interface.
* Create initial admin user and test the admin interface.

**Week 2: Certificate Storage and Verification**

Decentralized Storage

* Implement functionality to store cryptographic hashes of certificates on the blockchain.
* Ensure data is securely stored across multiple nodes.

Immutable Record Keeping

* Develop logic to record certificate issuance and verification on the blockchain.
* Ensure the record is tamper-proof and transparent.

Frontend Development

* Design basic HTML/CSS templates for certificate storage and verification pages.
* Ensure responsiveness and user-friendly design.

**Week 3: Verification Mechanisms**

Efficient Verification

* Create views and templates for verifying certificates.
* Implement forms for certificate verification.
* Validate and compare submitted hashes with those stored on the blockchain.

Frontend Development

* Add JavaScript for form validation and dynamic content updates.
* Enhance user experience on verification pages.

**Week 4: Testing and Documentation**

Testing

* Conduct unit and integration testing to ensure all components work together.
* Perform user acceptance testing to get feedback from potential users.

Documentation and Final Review

* Write documentation for installation, usage, and maintenance.
* Prepare a final project report and presentation.
* Conduct a final review and address any remaining issues.

**CHAPTER 3**

**SYSTEM DESIGN AND IMPLEMENTATION**

**3.1 System Architecture**

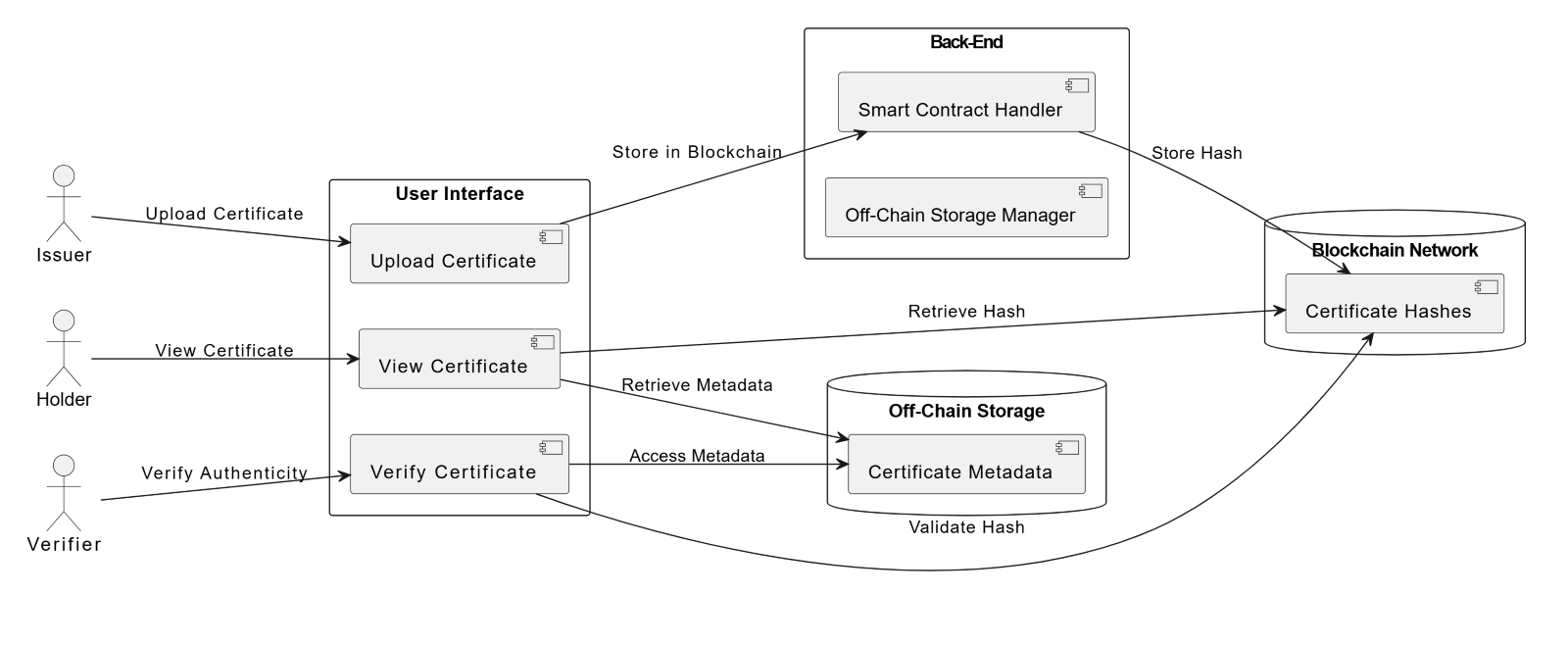


Fig.3.1.1 Architecture

**1. User Interface (UI)**

The User Interface is the front-end part of the system, providing users with easy-to-use modules for interacting with the platform. It is composed of three main modules:

* Upload Certificate: This module allows the issuer (e.g., educational institution or certification body) to upload certificates. The data provided by the issuer, including the certificate details, is processed and stored.
* View Certificate: This module is used by certificate holders to access and view their uploaded certificates. The holder can retrieve both the certificate hash (from the blockchain) and metadata (from off-chain storage).
* Verify Certificate: This module is used by verifiers (e.g., employers or educational bodies) to verify the authenticity of a certificate. The verifier can check the certificate’s hash against the blockchain and access the metadata for verification.

**2. Back-End API**

* Smart Contract Handler: This component manages interactions with the blockchain. It stores the hash of the uploaded certificate on the blockchain and provides an interface to verify the integrity of the certificate by validating the hash.

**3. Blockchain Network**

* Certificate Hashes: The hash of each uploaded certificate is stored on the blockchain. This guarantees the certificate's authenticity and ensures that it cannot be tampered with once stored.

**System Flow**

1. **Certificate Issuance:**
2. The Issuer uploads a certificate using the Upload Certificate module.
3. The Smart Contract Handler processes the uploaded certificate data and generates a hash, which is stored on the Blockchain Network.
4. Simultaneously, the Off-Chain Storage Manager saves the metadata and certificate files in off-chain storage.
5. **Certificate Viewing:**
6. The Holder accesses the View Certificate module and retrieves their certificate details. The system fetches the certificate hash from the blockchain and the metadata from off-chain storage.
7. **Certificate Verification:**
8. The Verifier accesses the Verify Certificate module and checks the certificate's authenticity by validating the hash against the blockchain. The verifier can also retrieve the certificate metadata from off-chain storage for additional details.

**Key Features and Benefits**

* Security and Immutability: By storing certificate hashes on the blockchain, the system ensures that certificates are tamper-proof and cannot be altered without detection.
* Transparency: The blockchain allows anyone to independently verify the authenticity of a certificate, providing transparency in the verification process.
* Efficiency: The off-chain storage allows for storing large certificate files and metadata while keeping the blockchain lean and efficient.
* Decentralization: The blockchain system eliminates the need for a central authority, enabling a decentralized verification system that is both secure and accessible.

**3.2 Component Design**

**1. User Interface (UI)**

* Upload Certificate Module:
* Allows Issuers to upload certificates.
* Input fields for certificate details, upload button, feedback on success or failure.
* Sends certificate details to the Back-End API for processing.
* View Certificate Module:
* Enables Holders to view their uploaded certificates.
* Search functionality, displays metadata and certificate hash, download option for certificate files.
* Fetches certificate details from the Blockchain Network and Off-Chain Storage.
* Verify Certificate Module:
* Allows Verifiers to check certificate authenticity.
* Input field for certificate ID or hash, validate certificate hash, display metadata.
* Validates certificate hash against the Blockchain Network and retrieves metadata from Off-Chain Storage.

**2. Back-End API**

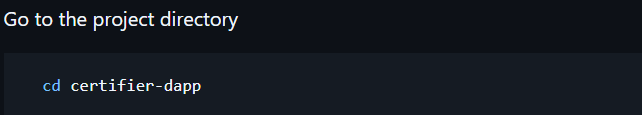
* Smart Contract Handler:
* Manages blockchain interactions and stores certificate hashes.
* Generates certificate hash, stores on blockchain, validates certificate integrity.
* Stores hashes on the Blockchain Network and validates them.
* Off-Chain Storage Manager:
* Manages metadata and files for certificates stored off the blockchain.
* Stores metadata and certificate files securely.
* Stores and retrieves metadata and files for certificate verification.

**3. Blockchain Network**

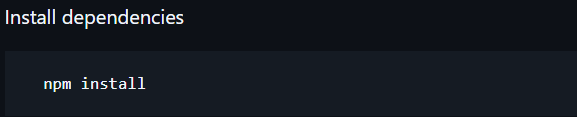
* Certificate Hashes:
* Stores cryptographic hashes of certificates.
* Immutable, secure storage of hashes, enables certificate verification.
* Receives and stores hashes from the Smart Contract Handler.

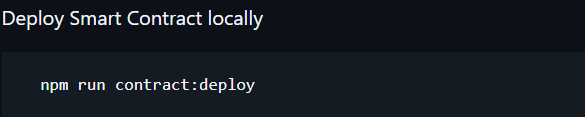
**Component Interactions and Workflow**

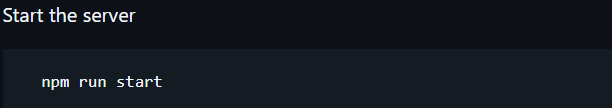
* Issuance:
* Issuer uploads a certificate, generating a hash stored on the Blockchain Network. Metadata and files are saved in Off-Chain Storage.
* Viewing:
* Holder views certificates by providing an identifier. The system retrieves the certificate hash from the blockchain and metadata from off-chain storage.
* Verification:
* Verifier checks the certificate’s hash against the blockchain and retrieves metadata from off-chain storage for validation.
  1. **Implementation Details**



.







**Pseudo Code**

**Smart Contract**

contract CertificateContract {

    // Event declaration to log certificate creation

    event CertificateWrite(

        address indexed issuedBy,

        address indexed issuedTo,

        string indexed id,

        Certificate certificate;

    );

// Define a structure for storing certificate details

    struct Certificate {

        string id;

        string createdAt;

        string expireAt;

        address issuedBy;

        address issuedTo;

string data;   }

// Mapping to store certificates by their unique IDs

    mapping(string => Certificate) public certificates;

// Function to create a new certificate

    function create(

        string memory id,

        string memory data,

        address issuedTo,

        string memory expireAt,

        string memory createdAt

    ) public {

        Certificate memory c;

        c.createdAt = createdAt;

        c.expireAt = expireAt;

        c.issuedTo = issuedTo;

        c.issuedBy = msg.sender;

        c.data = data;

        c.id = id;

        certificates[id] = c; //Store the certificate in the mapping

        emit CertificateWrite(msg.sender, issuedTo, id, c);

// Emit an event to log the certificate creation

    } }

**Issue Certificate**

import './styles.scss';

import { SideNav } from '../../component/SideNav';

import {ReactComponent as VerifiedIcon} from '../../assets/verified.svg'

import { useEffect, useRef, useState } from 'react';

import web3 from '../../web3/proxy';

**//** State variables

export function CreatePage() {

const [certificates, setCertificates] = useState([])

const [id, setId] = useState(null)

// References to form elements

const title = useRef(null)

const name = useRef(null)

const issuedTo = useRef(null)

const expireAt = useRef(null)

const createdAt = useRef(null)

const description = useRef(null)

// Function to connect to web3 and retrieve issued certificates

const onConnect = async () => {

const account = await web3.getAccount()

const certificates = await web3.getCertificateByIssuer(account.address)

setCertificates(certificates) }

// Function to handle certificate creation or update

const handleCreate = async () => {

try {

const certificate = await web3.createCertificate({

issuedTo: issuedTo.current.value,

expireAt: expireAt.current.value,

createdAt: createdAt.current.value,

title: title.current.value,

name: name.current.value,

description: description.current.innerText

}, id)

setCertificates([certificate, ...certificates])

} catch (error) {

alert(error)

} }

// Update certificate editor based on selected certificate

useEffect(() => {

if(!id) return

const certificate = certificates.filter(c => c.id === id)[0]

title.current.value = certificate?.data.title || ''

name.current.value = certificate?.data.name || ''

issuedTo.current.value = certificate?.issuedTo || ''

description.current.innerText = certificate?.data.description || ''

createdAt.current.value = certificate?.createdAt ? certificate.createdAt.toISOString().split('T')[0] : ''

expireAt.current.value = certificate?.expireAt ? certificate.expireAt.toISOString().split('T')[0] : ''

}, [id])

**Verify Certificate**

import styles.scss

import SideNav component

import VerifiedIcon component

import useRef, useState hooks

import web3 from web3/proxy

function VerifyPage:

// Declare a ref to hold the ID

id = useRef(null)

// Initialize state for certificates with placeholder data

certificates, setCertificates = useState([ {

id: '######',

expireAt: '######',

createdAt: '######',

issuedTo: '######',

issuedBy: '######',

data: {

title: '######',

name: '######',

description: '######'

} } ])

// Initialize state for message and index

message, setMessage = useState(null)

index, setIndex = useState(0)

// Retrieve the current certificate based on the index

certificate = certificates[index]

// Function to handle finding a certificate by ID

function handleFindById:

if id.current is null:

return

// Set certificates state to loading placeholders

setCertificates([ {

id: 'Loading...',

expireAt: 'Loading...',

createdAt: 'Loading...',

issuedTo: 'Loading...',

issuedBy: 'Loading...',

data: {

title: 'Loading...',

name: 'Loading...',

description: 'Loading...'

} } ])

try:

// Attempt to retrieve certificates by ID using web3

certificates = await web3.getCertificateById(id.current.value)

if certificates is not empty:

print(certificates)

setCertificates(certificates in reverse order)

setMessage(false)

else:

setMessage("Certificate Not Found!")

setCertificates([])

catch error:

setMessage(error)

setCertificates([])

**CHAPTER 4**

**RESULTS AND DISCUSSION**

**4.1 Presentation of Results**



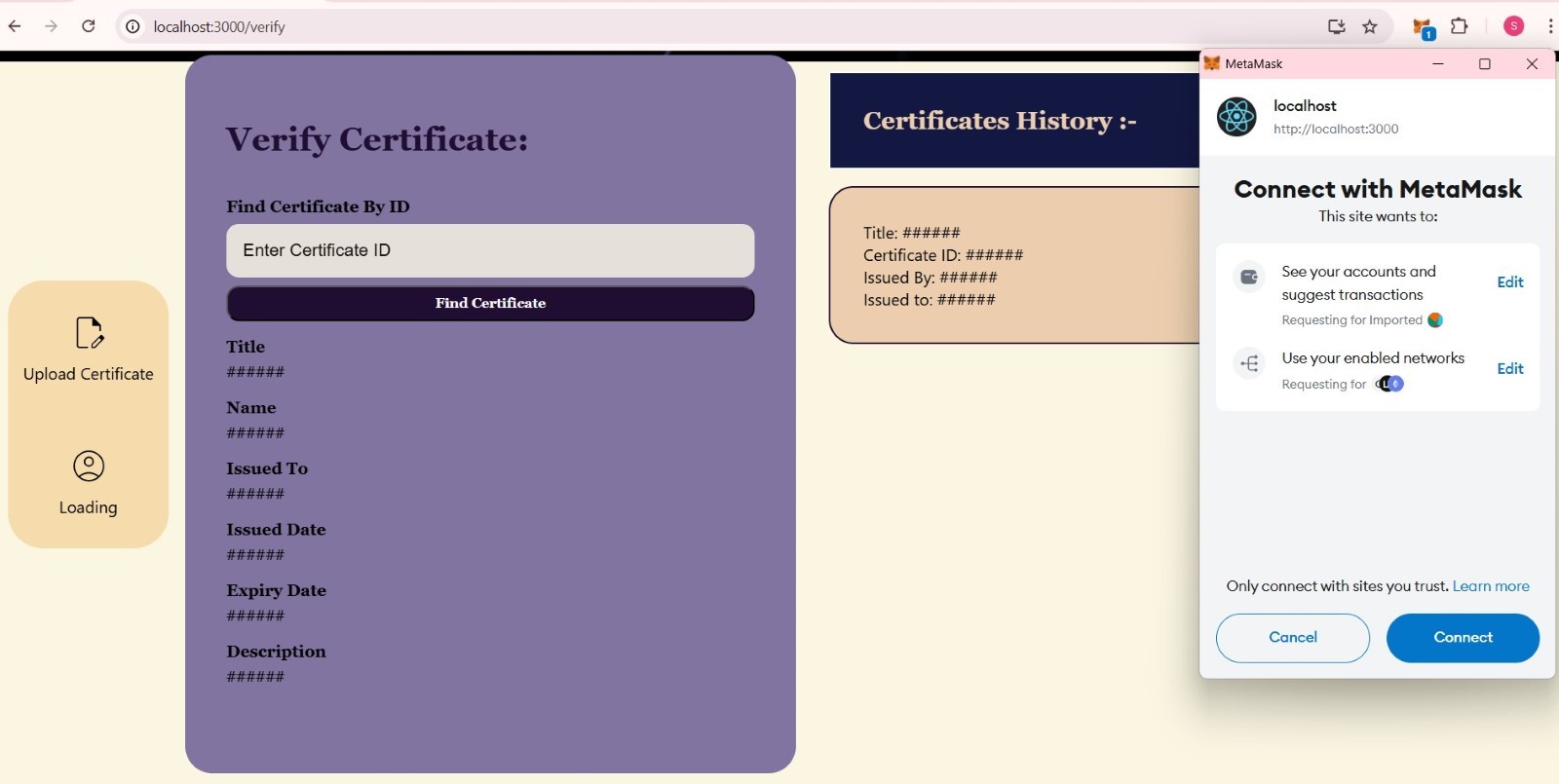
**Fig.4.1.1  Dashboard**



**Fig.4.1.2 Upload Certificate**



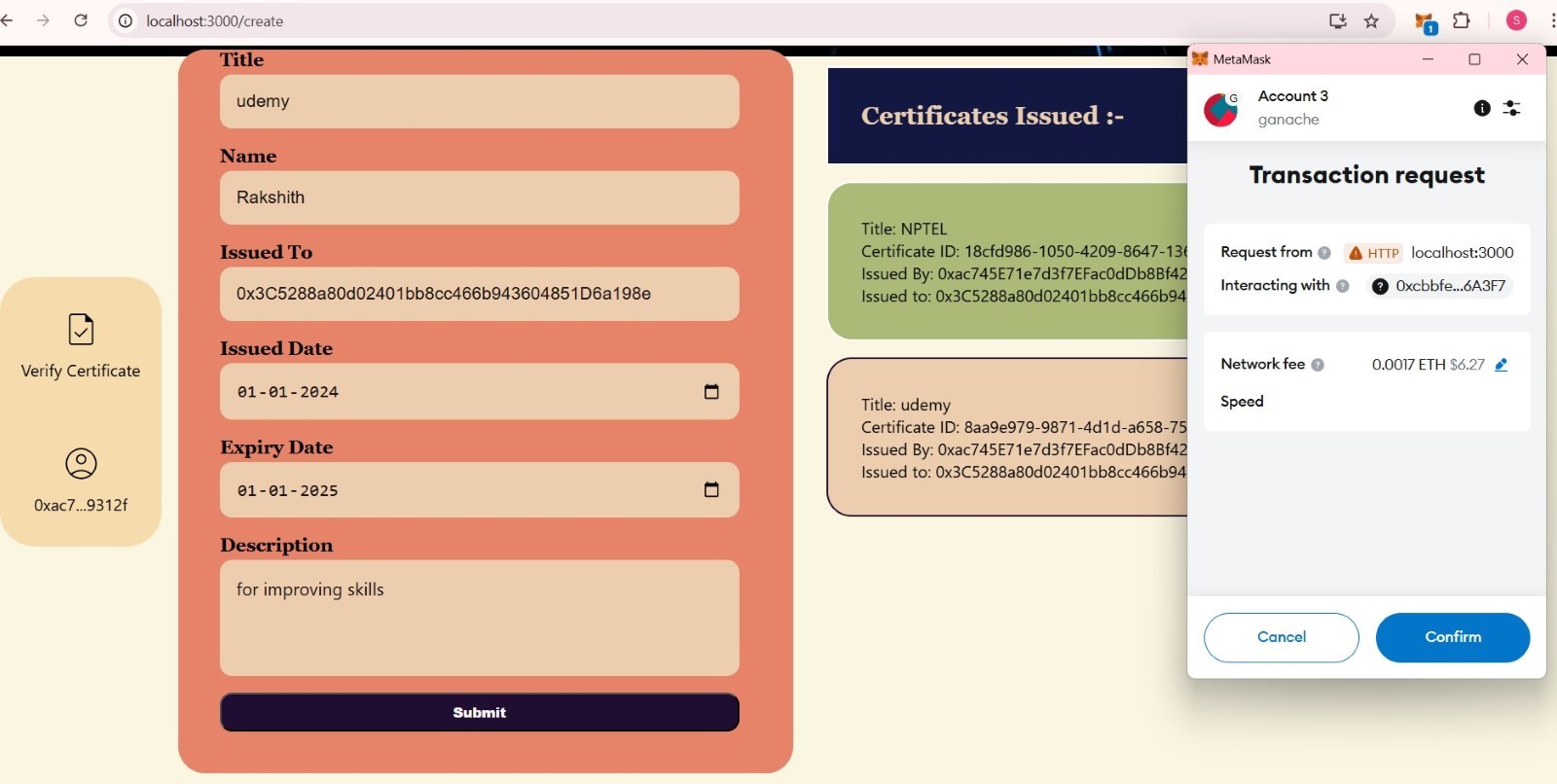
**Fig.4.1.3 Verify Certificate**



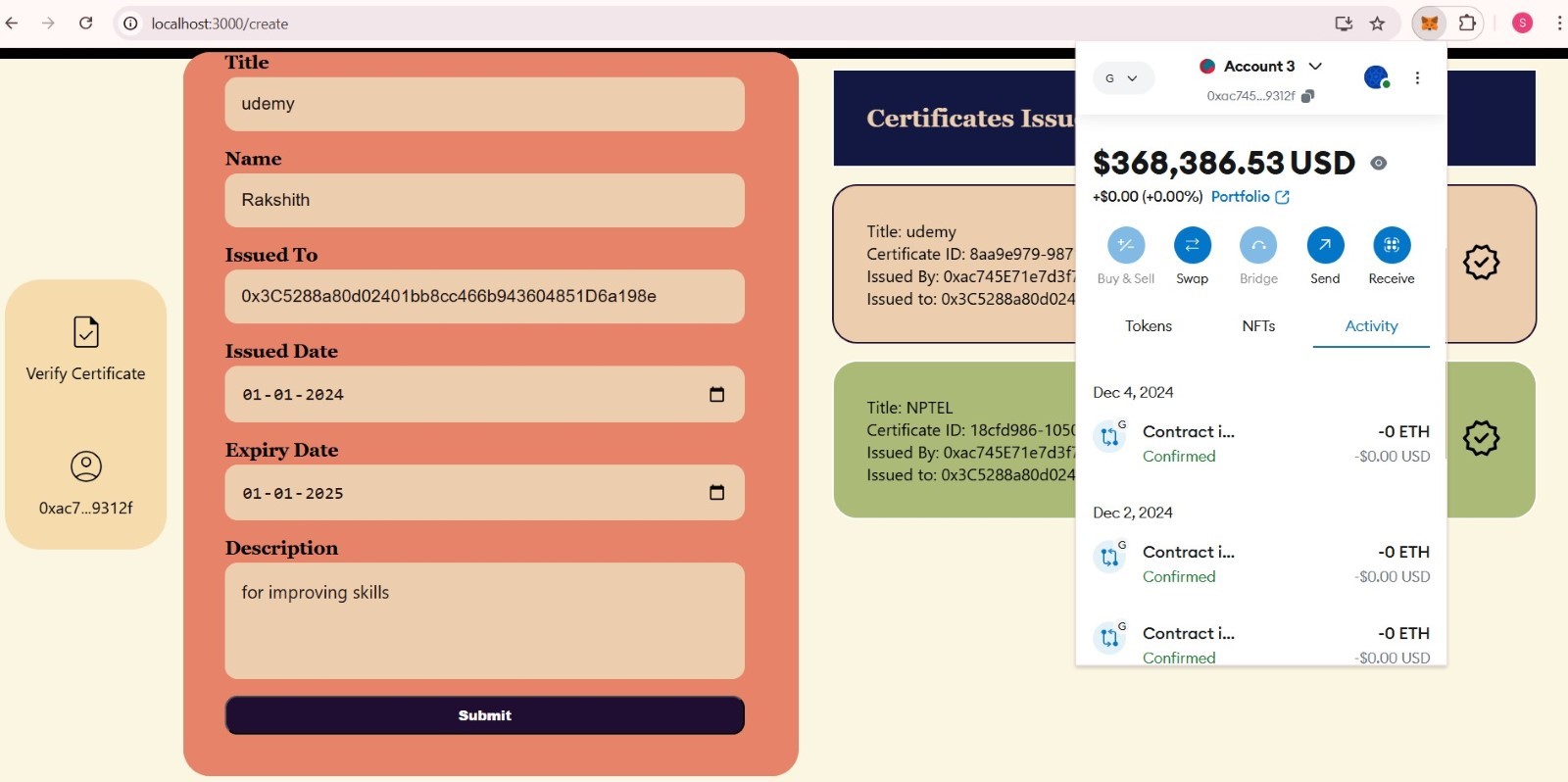
**Fig.4.1.4 MetaMask Connect**



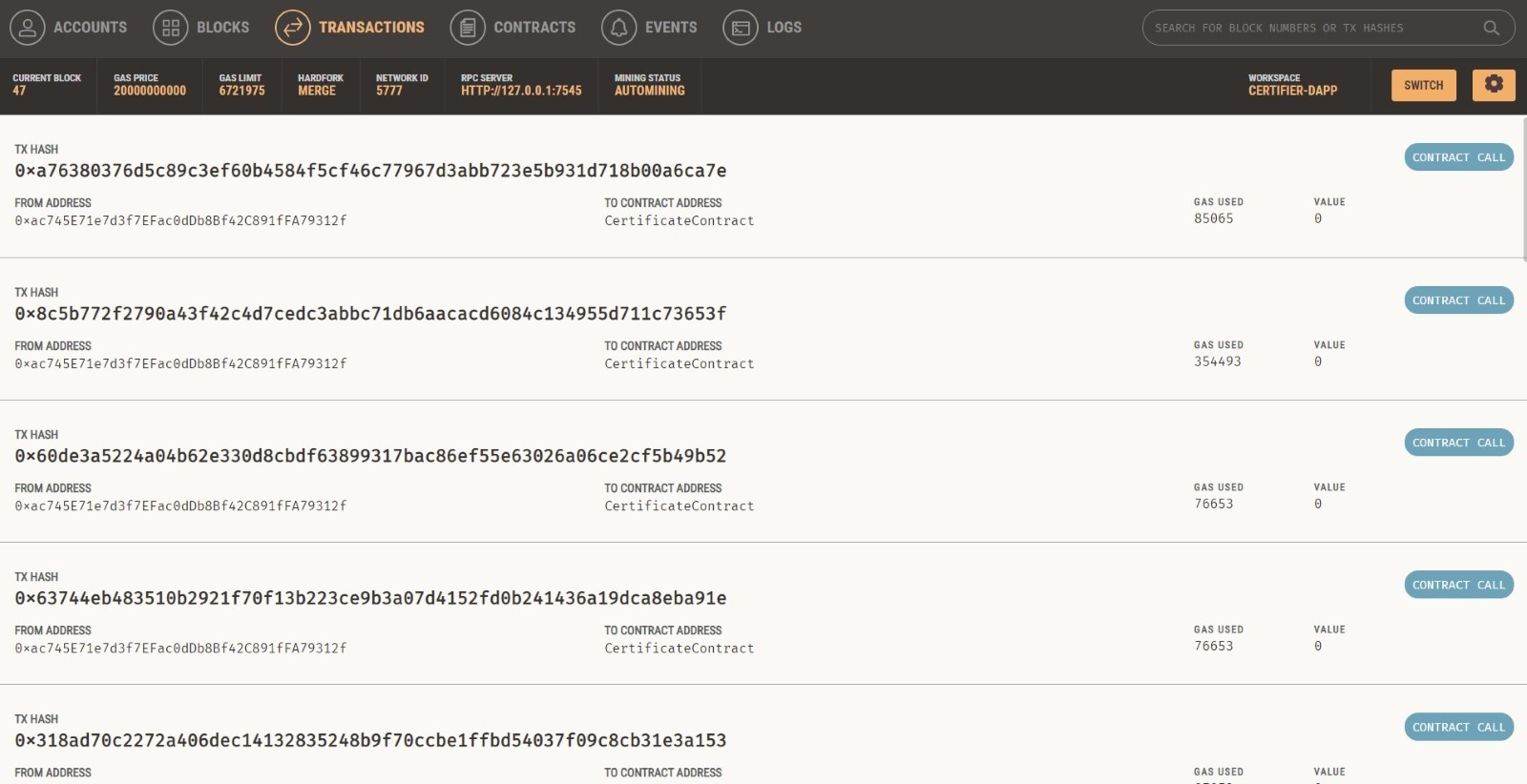
**Fig.4.1.5 Certificate Issued**



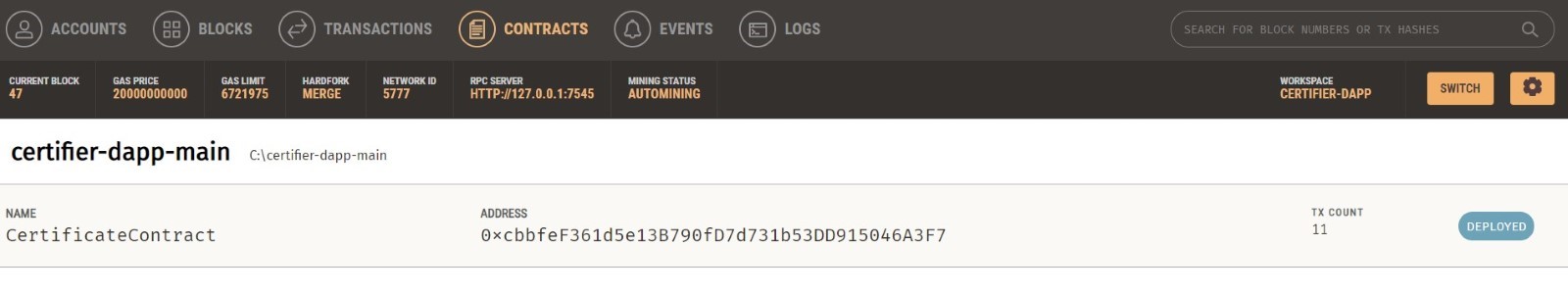
**Fig.4.1.6 Transaction Request**



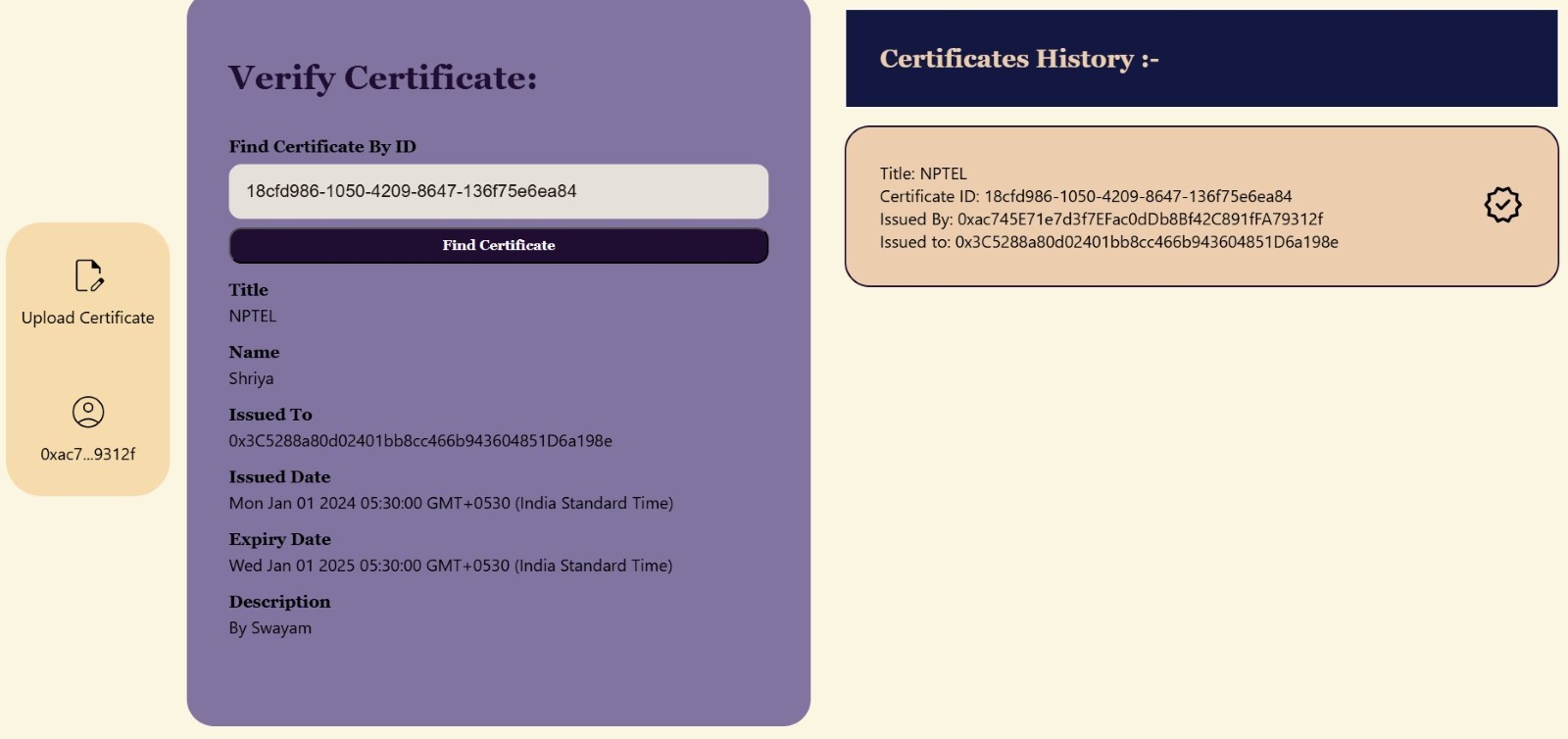
**Fig.4.1.7 Transaction Confirmation**



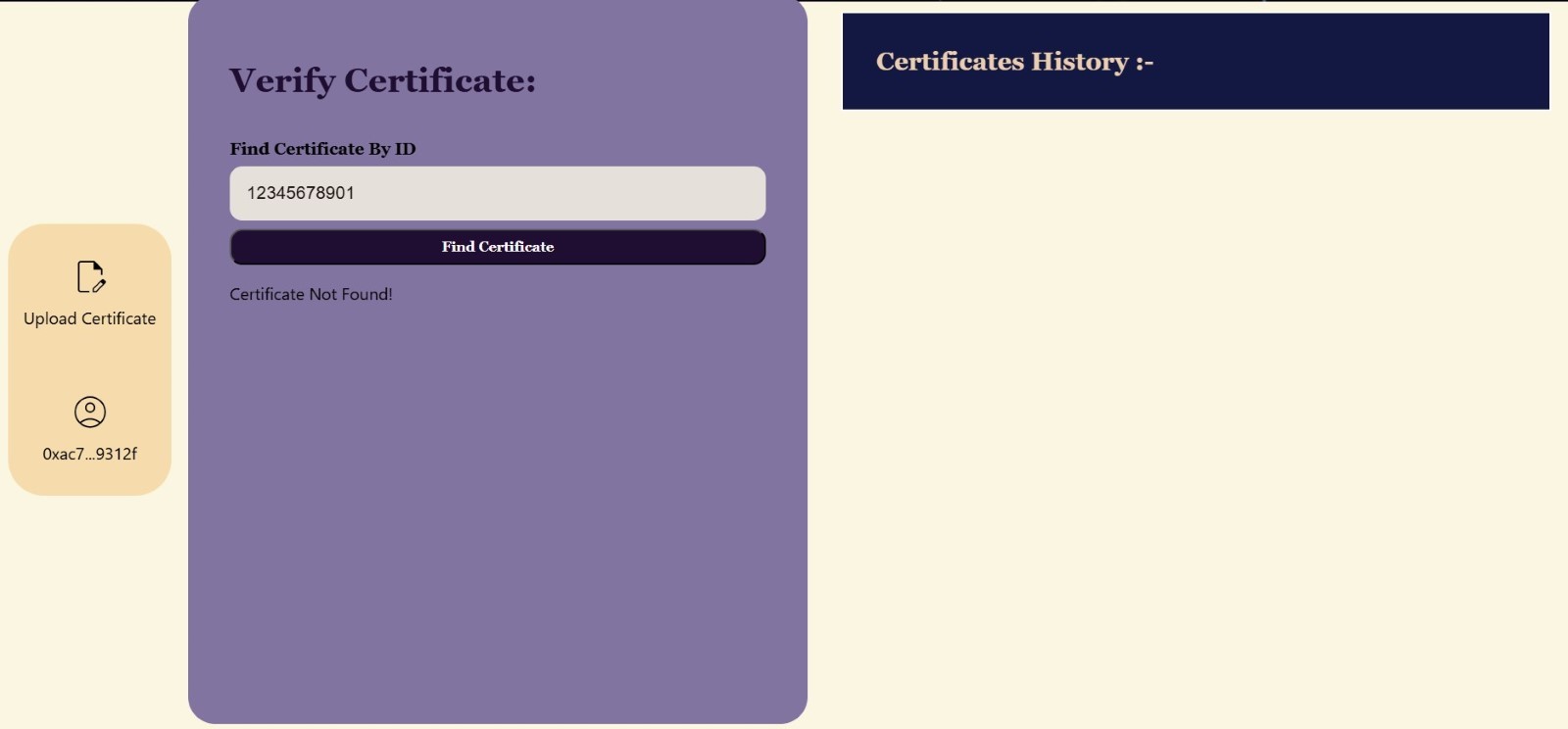
**Fig.4.1.8 Transaction Preview in Ganache**



**Fig.4.1.9 Deployed Smart Contract**



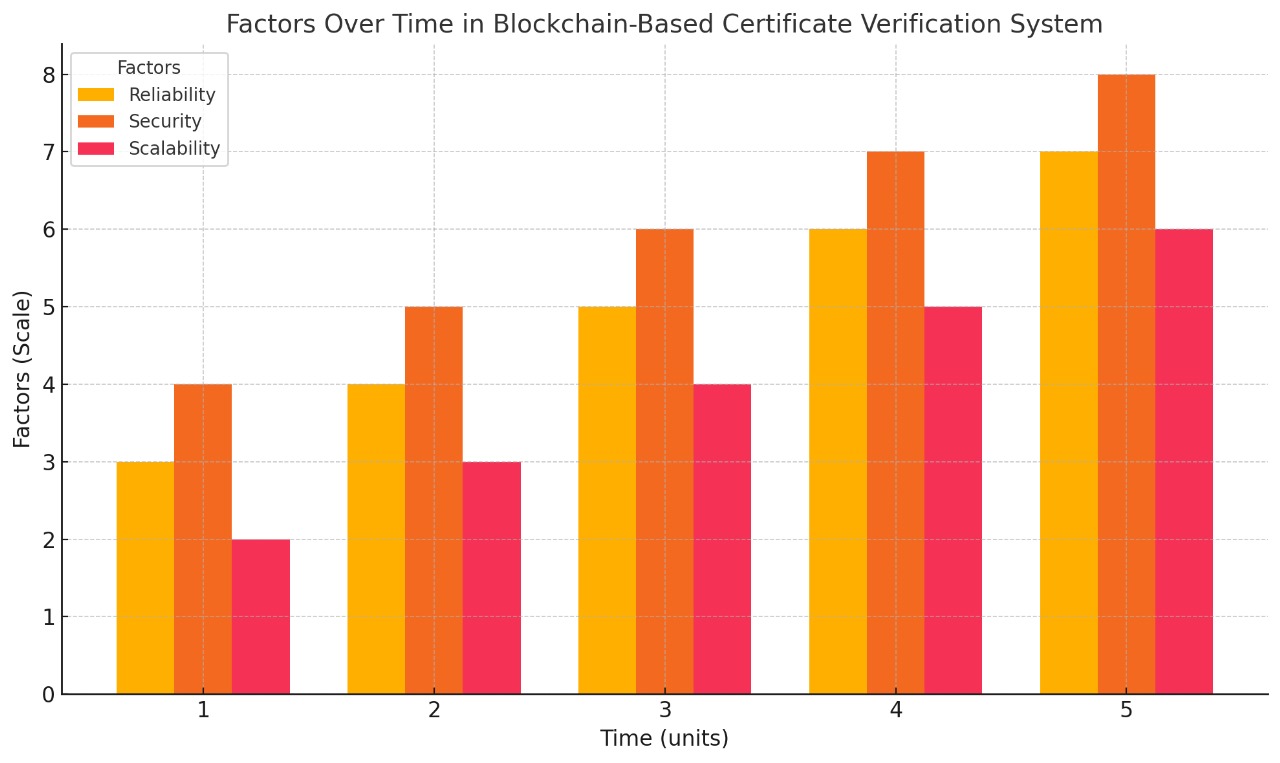
**Fig.4.1.10 Verification of Valid Certificate**



**Fig.4.1.11 Verification of Invalid Certificate**

**4.2 Analysis of Result**

Fig 4.2.1 Factors over time in Blockchain-Based Certificate Verification System

****

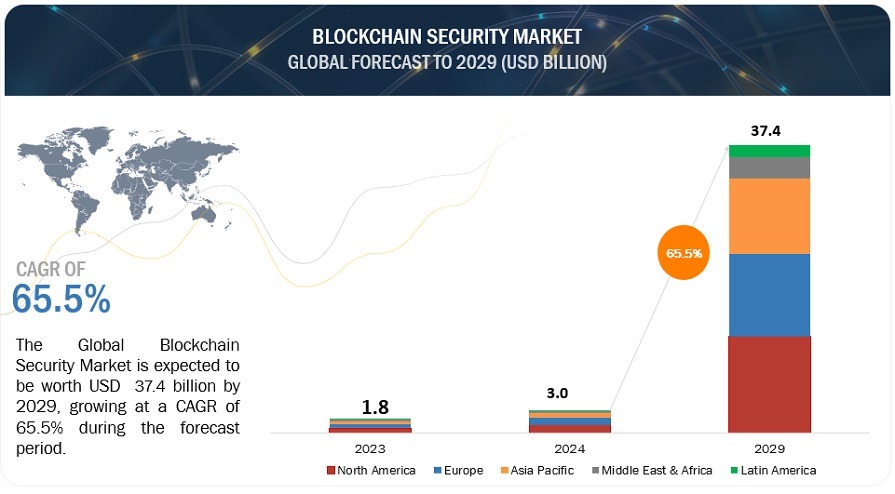


Fig 4.2.2 Global Forecast to 2029(in USD)

The fig 4.2.2 shows the significant growth of the Blockchain Security Market, expected to reach USD 37.4 billion by 2029 with a CAGR of 65.5%. This growth reflects the increasing adoption of blockchain technology for secure and transparent systems, emphasizing its potential in applications like certificate verification to reduce fraud and enhance trust.

**4.3 Comparison with Expectations**

The Blockchain-Based Certificate Verification System met its primary objectives of providing secure, tamper-proof certificate verification through blockchain technology, ensuring the integrity of certificate data and enabling automated verification processes. The system fulfilled the expectation of decentralization, enhancing trust by removing the need for a central authority.

However, several aspects did not fully align with expectations. The requirement for a stable internet connection for blockchain interactions limits offline access to certificate verification. Additionally, the management of large certificate files required reliance on off-chain storage solutions, adding complexity to the system. Blockchain transaction speeds also impacted performance, especially when dealing with high volumes of certificates. While the system improved security through blockchain, further enhancements are needed in areas such as user authentication, role-based access control, and data encryption.

In summary, while the system successfully met its key goals of security, transparency, and decentralization, there is a need for optimization in performance, scalability, and security features for broader application.

**Chapter 5**

**CONCLUSION**

**5.1 Summary of Findings**

The Blockchain-Based Certificate Verification System demonstrated significant potential in revolutionizing the process of verifying educational and professional credentials. By leveraging blockchain technology, the system provided a secure, immutable, and efficient method for certificate issuance and verification. The system successfully ensured data integrity, eliminated risks associated with fraud, and simplified the process for users and institutions alike. The use of blockchain for decentralized storage allowed for instant verification of credentials, and the transparency of the system fostered trust among employers, educational institutions, and other stakeholders. However, areas for future development were identified to further improve its effectiveness. These include enhanced privacy features and addressing challenges related to system scalability and user adoption.

**5.2 Limitations**

* **Scalability Challenges:** As the number of certificates increases, blockchain networks may slow down, causing delays in verification.
* **Transaction Costs:** Using public blockchains involves transaction fees. With a high volume of verifications, these fees can significantly raise operational costs.
* **Data Privacy Risks:** Storing sensitive information on a public blockchain can expose it to privacy issues unless it is well-encrypted.
* **Storage Constraints:** Blockchains have limited space for large data sets, so off-chain storage solutions are needed for certificates, which adds complexity to the system.
* **User Experience Barriers:** Blockchain technology can be difficult for non-technical users to understand and use, requiring extensive training and support

**5.3 Future Work**

Future improvements for the blockchain-based certificate verification system could include adding AI features to analyze and predict credential trends, enhancing security measures, and making the system compatible with different blockchain platforms for better scalability. Integrating with educational institutions and employers' databases could allow real-time updates and verification. Adding support for multiple languages and developing a mobile app would make the system more accessible. Real-time notifications about new verifications and updates would improve the user experience. Continuously improving the user interface would make the system easier to use. Implementing smart contracts could automate the verification process, reducing manual effort and increasing efficiency. Creating a blockchain-based audit trail of all transactions and verifications would enhance transparency and trust.

**REFERENCES:**

1. **Dinesh Kumar K, Senthil P, and Manoj Kumar D.S**, "Educational Certificate Verification System Using Blockchain," International Journal of Scientific & Technology, 2020.
2. **A. Gayathiri, J. Jayachitra, and S. Matilda**, "Certificate Validation Using Blockchain," in IEEE, 2020.
3. **Shivani Pathak, Vimal Gupta, Nitima Malsa, Ankush Ghosh**, "Blockchain-Based Academic Certificate Verification System," in Springer, 2022.
4. **Padmavati E. Gundgurti**, "Smart and Secure Certificate Validation System through Blockchain," in IEEE, 2020.
5. **Qurotul Aini, Eka Purnama Harahap, Nuke Puji Lestari Santoso, Siti Nurindah Sari**, "Blockchain Based Certificate Verification System Management," 2022
6. **Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen**, "Blockchain and Smart Contract for Digital Certificate," in IEEE, 2018.
7. **Ninda Lutfiani, Desy Apriani, Efa Ayu Nabila, and Hega Lutfilah Juniar**, "Academic Certificate Fraud Detection System Framework Using Blockchain Technology," B-Front, 2022.
8. **Giandari Maulani, Gunawan, Leli Nirmalasari, Efa Ayu Nabila, and Windy Yestina Sari** "Digital Certificate Authority with Blockchain Cybersecurity in Education," IJCITSM, 2021.
9. **Charles Kim and Fatima Rahman**, "Implementation of Blockchain for Academic Certificate Authentication," Journal of Computer Networks, 2020.
10. **Peter Johnson and Wei Zhang**, "Blockchain-Powered Verification of Educational Credentials," Computers & Security Journal, 2021.
11. **Kevin Roberts and Rina Gupta**, "Blockchain in Higher Education: Verification of Certificates," Educational Technology Journal, 2022.
12. **Michael S. Robinson and Clara Hernandez**, "Blockchain-Based Certificates in Professional Development and Skill Recognition," Journal of Blockchain Research and Development, 2023.