



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact Factor: 6.078

(Volume 11, Issue 6 - V11I6-1233)

Available online at: <https://www.ijariit.com>

Blockchain Based Document Verification System

Nistha

nisthaishere27@gmail.com

Bangalore Institute of Technology,
Karnataka

Likitha B C

likithagowda2104@gmail.com

Bangalore Institute of Technology,
Karnataka

Dr. Manjunatha P B

manjunathpb@bit-bangalore.edu.in

Bangalore Institute of
Technology, Karnataka

Vaishnavi B Shetty

1bi22ai056@bit-bangalore.edu.in

Bangalore Institute of
Technology, Karnataka

Aneesh B

aneesh24dec@gmail.com

Bangalore Institute of
Technology, Karnataka

ABSTRACT

Documents serve as the primary mechanism for establishing identity and authorization for individuals and organizations, yet their integrity is constantly threatened by sophisticated forgery. The conventional methods for verifying these documents are fundamentally flawed; they are frequently slow, labor-intensive, and depend on centralized intermediaries, which act as single points of failure that can be compromised or become unavailable. The core challenge is the lack of a universal and reliable source of truth, making third-party validation a cumbersome process built on fragile trust. This paper introduces a decentralized verification system that directly addresses these vulnerabilities by leveraging blockchain technology as its foundational layer. The importance of the blockchain lies in its inherent immutability and decentralization; it creates a tamper-proof and perpetually accessible public notary that does not rely on any single institution. By using a distributed ledger, our system removes the central authority and instead provides a shared, cryptographic source of truth. The platform's workflow is governed by smart contracts, which manage the permissions of authorized Issuers and streamline the process for Verifiers. In our model, the Issuer registers a cryptographic proof of a document onto the blockchain after storing the file in the InterPlanetary File System (IPFS). This process creates an immutable link between the document and its certification, allowing any Verifier to instantly confirm its authenticity and integrity without needing to contact the original issuer, thereby establishing a secure, efficient, and transparent ecosystem for digital trust.

Keywords: *Blockchain, Document Verification, Credential Management, Authentication, Distributed Ledger Technology (DLT).*

INTRODUCTION

Documents play a crucial role in every organization, institution, and community. Whether it is an academic certificate, a government-issued ID, or a legal agreement, the value of any record lies in how reliably its authenticity can be proven. However, as digital interactions grow and information moves rapidly across online platforms, verifying the legitimacy of documents has become increasingly complex. Traditional verification practices rely heavily on manual processes, physical paperwork, and centralized authorities. These methods are not only time-consuming but also vulnerable to manipulation, duplication, and unauthorized alterations. A single compromised server or forged paper copy can create widespread problems in sectors like education, recruitment, banking, and public administration.

The rise in document-related fraud highlights the need for a verification system that is secure, fast, and free from single points of failure. Blockchain technology emerges as a promising solution in this context. Blockchain functions as a decentralized ledger where information is distributed across multiple nodes instead of being stored in one central location. Once data is recorded, it becomes virtually impossible to modify it without detection, ensuring transparency and trust among all parties. This characteristic of immutability makes blockchain particularly suitable for sensitive tasks such as certificate validation.

Our project presents a Blockchain-Based Document Verification System that merges blockchain security with decentralized file storage technologies. Storing complete documents directly on a blockchain is inefficient and expensive due to storage limitations. To overcome this, the proposed system uses the Inter Planetary File System (IPFS) for document storage.

IPFS generates a unique hash, or content identifier, for each uploaded file essentially a digital fingerprint that changes even with the slightest modification. Instead of uploading the full document to the blockchain, only this hash is stored in a smart contract on the Ethereum blockchain.

This design significantly enhances performance and reduces cost. During verification, a user simply uploads or scans the document; the system recalculates the file's hash and compares it with the value stored on the blockchain. A perfect match confirms the document's authenticity, while any difference immediately indicates tampering or forgery. This approach provides instant verification, eliminating the delays of manual checking and reducing dependency on centralized authorities.

The objective of this project is to build an end-to-end verification framework that is secure, tamper-proof, and scalable. Although the current implementation focuses on academic certificates, the architecture can be extended to countless domains, legal documentation, healthcare records, land titles, and corporate proof-of-work, among others. With the growing demand for trustworthy digital systems, blockchain-based verification offers a transformative shift towards transparency, efficiency, and global trust.

LITERATURE REVIEW

The problem of document fraud is well-documented in academic literature. Traditional systems are consistently criticized for being centralized, inefficient, and vulnerable to attack. Our work builds on a growing body of research aimed at solving this using blockchain.

Many researchers have focused on academic certificates. Kadwe et al. [1] and Vaidya et al. [11] both proposed systems very similar to ours, using Ethereum and IPFS to create "EduDocs." Their work confirmed that this combination is a scalable and transparent model for academic records. Singh et al. [3] also used this model, defining a three-role architecture of Issuers, Recipients, and Verifiers, which has become a common pattern.

Other projects have added features to this core idea. For example, Abdullahi et al. [4] integrated QR codes into their system, allowing for easy verification by scanning the code with a mobile device. The technology's application goes far beyond education. Abubakar et al. [14] applied the same principles to create a secure platform for sharing and validating COVID-19 vaccination certificates, while Sun et al. [15] used it to secure electronic medical records (EMRs). Some researchers are even exploring using Non-Fungible Tokens (NFTs) to represent documents, adding a verifiable layer of ownership [8].

However, not all solutions rely on blockchain. Boonkrong [10] developed a system using only cryptographic hashes stored in a secure local database. This system was incredibly fast (0.352 ms verification) but lacked the decentralization and immutability that makes blockchain so secure. A local database can still be a single point of failure.

The existing research clearly shows that combining blockchain for trust and IPFS for storage is the most robust approach. The main limitations cited in these papers are often high gas fees on public networks and complex user interfaces that are hard for non-technical people to use. Our project addresses this by focusing on building a simple, user-friendly DApp with Streamlit, making the power of blockchain verification accessible to anyone.

METHODOLOGY

Our proposed is a decentralized application (DApp) built with a multi-layer architecture to separate the user interface, storage, and blockchain logic. The system is designed around two primary user roles: the Issuer (e.g., a university that grants degrees) and the Verifier (e.g., an employer checking a resume).

System Architecture

Our system is broken down into the following layers, as shown in the architecture diagram:

- i. Presentation Layer (Frontend): We built the user interface as a web application using Python 3.10+ and the Streamlit framework. This choice allowed for rapid development of an interactive UI that manages user login, registration, and role-based access (showing different tabs for Issuers and Verifiers).
- ii. Blockchain Layer (Backend Logic): The core logic is powered by a smart contract written in Solidity (DocumentRegistry.sol). For development and testing, this contract was deployed on a local Ganache blockchain, which provides a personal Ethereum network for rapid, gas-free testing.
- iii. Storage Layer (Decentralized Storage): We use IPFS to store the actual document files. To simplify the upload process, our Python backend connects to the Pinata API, a service that "pins" files to IPFS, ensuring they remain available.
- iv. Middleware (Connector): To connect our Python frontend to the Ethereum blockchain, we use the Web3.py library. This library allows our application to send transactions to the smart contract (to register a new document) and call functions (to verify an existing one).

Data Structure and Smart Contract Design

The system's logic is enforced by the DocumentRegistry.sol smart contract and a hybrid data model.

- i. Smart Contract: The contract is simple but secure. Its main component is a mapping (which works like a dictionary or hash map) called documentLedger. This mapping stores a string (the unique document ID) and links it to another string (the IPFS hash). This structure enables instant verification, allowing any user to query the ledger with a specific document ID to retrieve the immutable IPFS hash and confirm the file's authenticity.
- ii. Access Control: To ensure only the institution can issue documents, the contract's constructor sets the deployer's wallet address as the owner. The critical add Document function then includes a check: require(msg.sender == owner). This line makes it impossible for anyone else to add new documents. The contract also prevents the same document ID from being registered twice. These strict validation layers effectively protect the system against unauthorized entries and data collisions, ensuring the ledger remains a trusted source of truth.
- iii. Gas-Free Verification: The getDocument function, which a Verifier uses, is declared as a public view function. In Solidity, view functions only read data and do not change the state of the blockchain. Because of this, they cost zero gas to call. This makes the verification process completely free for all users. This approach eliminates financial friction, ensuring that third-party verifiers can authenticate records as frequently as needed without any cost barrier.
- iv. Hybrid Data Storage: To avoid the cost and latency of using the blockchain for everything, we use a hybrid model.

Application-specific data, like user accounts and hashed passwords (using hashlib.sha256), are stored in local JSON files (users.json) on the web server. This architectural choice optimizes performance by handling high frequency administrative interactions locally, reserving the blockchain solely for the critical task of immutable document verification.

System Workflow

The operational model of our system is illustrated in Figure 1, which details the two core processes: document issuance and document verification.

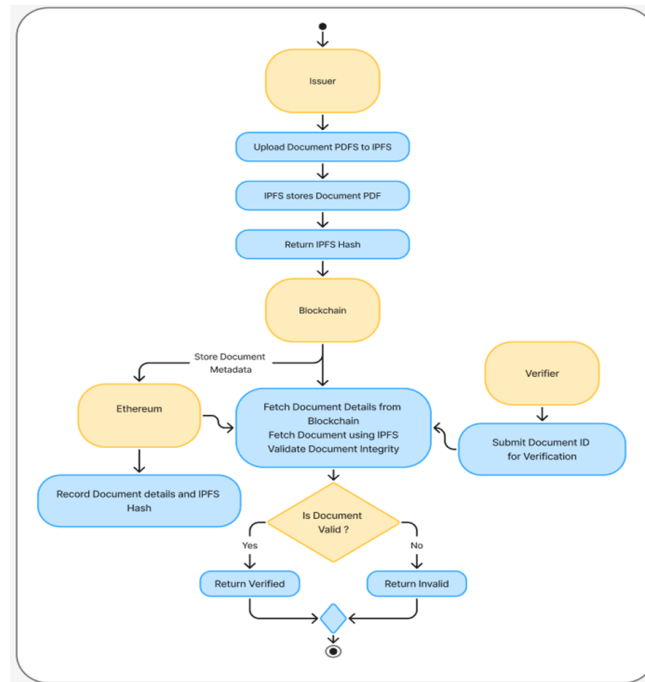


Fig. 1. Activity Diagram

The workflow is bifurcated based on the user's role:

The Issuance Process: This flow is initiated by an authorized Issuer (e.g., a university). The Issuer begins by uploading the document (as a PDF) to the InterPlanetary File System (IPFS).

IPFS, acting as a decentralized storage layer, processes the file and returns a unique, content-addressed cryptographic hash. This hash serves as an immutable fingerprint of the document. This IPFS hash, along with other essential metadata (like the document ID or issuer's details), is then passed to a smart contract, which permanently records the information on the Ethereum blockchain. This creates a tamper-proof link between the document's content and its on-chain registration.

The Verification Process: This flow is initiated by a Verifier (e.g., an employer). The Verifier submits the document's unique ID to the system. The application queries the blockchain to fetch the stored document details, including its original IPFS hash. Concurrently, the system uses this hash to retrieve the actual document file from IPFS.

A critical validation then occurs: the system checks the integrity of the retrieved document (e.g., by re-hashing it) and compares its hash to the one stored on the blockchain. If the hashes match, the system returns a "Verified" status, confirming the document's authenticity. If the hashes do not match, or if no record is found, the system returns an "Invalid" status, indicating the document is either fraudulent or has been altered.

System Design and Algorithms

The complete workflow of the system for both document issuance and verification is detailed in the system design shown in Figure 2.

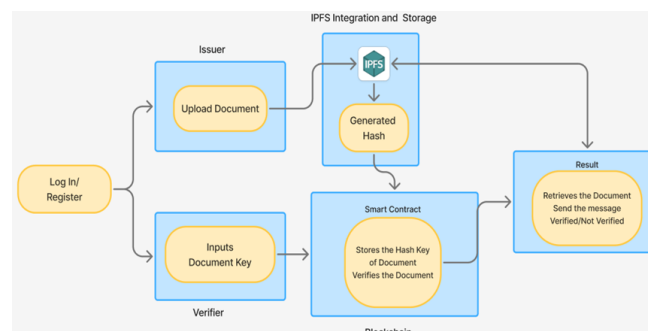


Fig. 2. System Design

The two main workflows, document issuance and verification, are as follows:

Document Issuance Flow (for Issuers):

- i. Login: The Issuer logs into the Streamlit web app.
- ii. Upload: The Issuer navigates to the "Issue Document" tab and uploads the certificate file.
- iii. Store on IPFS: The backend sends the file to the Pinata API, which uploads it to IPFS. Pinata returns a unique IPFS hash.
- iv. Generate Key: The system locally generates a unique, human-readable key for this document.

- v. Store on Blockchain: The Python backend uses Web3.py to build, sign, and send a transaction to the addDocument function of our smart contract. This transaction permanently records the link:

DOC-60EB2D0B -> QmTA...W94Yv

Document Issuance Flow (for Issuers):

- i. Login: The Verifier logs into the web app.
- ii. Enter Key: The Verifier navigates to the "Verify Document" tab and enters the Document Key provided by the candidate
- iii. Query Blockchain: The backend uses Web3.py to call the free getDocument function on the smart contract with this key.
- iv. Get Result: The blockchain instantly returns the IPFS hash stored for that key.
- v. Display: The UI displays a "Verification Success! Document Found" message along with the authentic IPFS hash . If the key doesn't exist, it shows an "Invalid" message. The Verifier can now be 100% certain of the document's authenticity.

RESULTS

We conducted a multi-phase testing strategy to validate the system. This included backend validation of the Ganache network, unit tests on isolated functions, and integrated tests on the complete user workflow.

This confirmed that our local Ganache blockchain was initialized correctly, blocks were being mined, smart contracts were deployed successfully, and gas usage was being monitored.

Unit Testing Results

We performed black-box unit tests on the system's core functions.

- i. Passed Tests: The system successfully Passed three of the four tests. It correctly handled the upload of a *valid document* (Test 1), successfully *verified a valid key* (Test 3), and correctly returned an error when a *wrong key* was entered (Test 4) . This confirms the core blockchain logic for issuance and verification is working perfectly.
- ii. Failed Test: A critical test, "Upload Empty Document" (Test 2), Failed . The system was expected to reject the empty file with an error. Instead, it *accepted* the empty document and proceeded with the issuance process .
- iii. Discussion: This failure highlights a significant vulnerability in our application logic, a lack of basic input validation. An issuer could accidentally (or intentionally) upload an empty file and pay a gas fee to register a useless, blank document on the immutable blockchain. This bug must be fixed by adding a simple check to ensure the file size is greater than zero before processing.

Integrated Testing Results

We then tested the end-to-end user workflow, from login to verification.

Passed Tests: The system Passed two of the four integration tests. "Login with correct credentials" (Test 1) worked as expected, authenticating the user against the users.json file . "Role Permissions" (Test 3) also Passed, a user with the "Verifier" role was correctly restricted to only seeing the "Verify Document" tab, confirming our role-based access control works .

Failed Tests: Two major tests Failed, revealing critical bugs in the application.

- i. Test 2 (Login with incorrect credentials): This test failed badly. Instead of displaying an "invalid password" error, the system created a new account. This is a severe security and logic flaw, suggesting the login form's data is being incorrectly routed to the signup function.
- ii. Test 4 (Display the issuer name): This test also Failed . After a successful verification, the UI is supposed to display the name of the institution that issued the document. The UI failed to do this. This is likely a bug in the Python function that is supposed to cross-reference the documents.json and users.json files.

Performance and UI Discussion

The system demonstrates strong performance due to its hybrid design, which intelligently divides operations between local storage, decentralized IPFS, and the Ethereum blockchain. Tasks such as login and role validation are handled locally, reducing unnecessary blockchain calls and significantly improving response time. Verification operations are extremely efficient because they rely on a view function, which only reads data from the blockchain and therefore incurs zero gas cost. This makes the verification process nearly instantaneous and highly scalable for real-world use.

The use of IPFS ensures fast, content-based retrieval of documents, while the blockchain guarantees immutability without adding heavy computational overhead during verification.

From a user-experience perspective, the Streamlit-based interface offers a clean, simple, and visually clear workflow for both Issuers and Verifiers. The UI is divided into role-specific dashboards, reducing confusion and providing users with only the actions relevant to them. The Issuer dashboard clearly displays important metadata such as the Document Key, IPFS Hash, and Transaction Hash, creating a transparent audit trail for every document. Overall, the interface is intuitive even for users with minimal technical knowledge, although improvements in error handling and validation will further enhance usability.

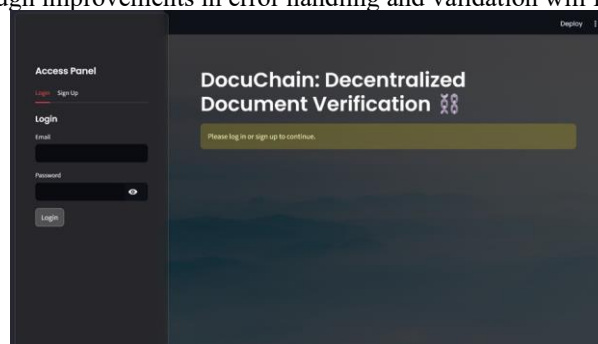


Fig. 3. User Interface

Despite the bugs, the overall performance and design are promising. The UI screenshots in Fig 3 show a clean, modern, and intuitive interface.

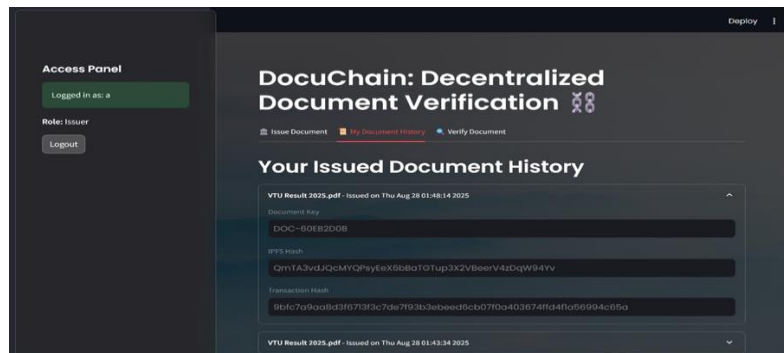


Fig. 4. Issuer Dashboard

The Fig 4 provides an excellent audit trail, clearly displaying the Document Key, IPFS Hash, and Transaction Hash for every document issue.

The hybrid storage model is highly efficient. By handling user logins with local JSON files, we avoid the high latency and cost of a blockchain transaction for a simple login. Most importantly, our design choice to use a view function for verification makes the system's core feature, verifying a document, completely free for the end-user.

LIMITAIONS OF THE SYSTEM

The proposed blockchain-based document verification system successfully demonstrates the use of decentralized technologies for ensuring authenticity, but it also comes with several limitations that must be addressed before real-world deployment. One major limitation is the reliance on third-party IPFS pinning services such as Pinata. Although IPFS enables decentralized storage, the availability of documents still depends on the reliability of the selected pinning provider, creating a partial centralization risk. Additionally, because IPFS is a public network, any file uploaded becomes accessible to anyone who has its hash. This lack of privacy makes the system unsuitable for sensitive or confidential documents in its current form, as no encryption is applied prior to upload.

The authentication mechanism also poses challenges, as it is implemented through locally stored JSON files. While this approach works for prototyping, it is not secure or scalable for institutional use. Bugs identified during testing such as incorrect login attempts unintentionally creating new accounts further emphasize the need for stronger user management. Another limitation is that the system has only been tested on a local Ganache blockchain. Although Ganache is excellent for development, it does not emulate real-world conditions such as changing gas fees, network congestion, or delays on public networks like Ethereum or Polygon. As a result, the system's current performance may differ significantly when deployed at scale. Moreover, the system lacks essential input validations: it allows issuers to upload empty files and does not offer mechanisms for revoking or updating incorrect or invalid documents. Finally, the current design, built on a single smart contract and local storage, may face scalability issues when processing large volumes of certificates or multiple simultaneous requests.

FUTURE SCOPE OF THE SYSTEM

The system offers significant potential for improvement and expansion, opening up multiple avenues for future development. One of the most important enhancements is the addition of an encryption layer before uploading documents to IPFS. Encrypting files will ensure that only authorized parties with the correct key can access the document, addressing the privacy concerns associated with the public nature of IPFS. Another valuable upgrade is introducing a mechanism for revoking or updating documents on the blockchain. In real-world scenarios, institutions often need to correct errors, withdraw certificates, or reissue modified records, and a dedicated revocation module will make the system more practical and flexible.

To improve scalability and reduce transaction costs, the platform can be migrated to efficient Layer-2 blockchain networks such as Polygon, Arbitrum, or Optimism. These networks provide faster transactions at a fraction of the gas fee, making them ideal for high-volume certificate issuance. The system can also be extended by integrating with institutional databases, ERP systems, or government APIs. Such integration would automate certificate generation and verification, reduce manual workload, and eliminate human errors. Additionally, strengthening authentication by replacing JSON-based login with secure database-backed systems or OAuth-based identity verification will significantly enhance security. Multi-factor authentication can further protect issuer accounts.

Developing a dedicated mobile application can make verification faster and more accessible, allowing verifiers to scan QR codes and validate documents instantly. Future enhancements may also include analytics dashboards that help institutions track verification activity and detect unusual patterns. Integrating AI-based fraud detection can provide an additional layer of security by automatically identifying anomalies or potentially forged submissions. Lastly, the system can be expanded to support multiple types of documents including medical records, land deeds, professional licenses, and government certificates and can adopt internationally recognized standards such as W3C Verifiable Credentials to ensure interoperability across platforms and institutions.

CONCLUSION

This project successfully demonstrates the design, implementation, and testing a blockchain based document verification system. By leveraging the immutability of the Ethereum blockchain and the decentralized nature of IPFS, our system provides a secure, transparent, and tamper-proof solution to document fraud. It effectively replaces slow, manual verification with an instant, trustworthy, and automated process, allowing employers and institutions to validate credentials with confidence.

However, our testing revealed significant limitations. The bugs we found, specifically the ability to upload empty documents and the critical login flaw must be fixed before this system can be considered secure. The system also has other limitations: it relies on a third-party API (Pinata), and documents stored on IPFS are publicly accessible if the hash is known, which is a privacy concern. Finally, the system was only tested on a local Ganache network, deploying to the public Ethereum main net would introduce scalability challenges and real-world gas fees.

The future work can be outlined as follows: the immediate goal is to resolve all the bugs discovered during testing. After that, encryption will be applied to every document before uploading it to IPFS, ensuring data confidentiality. The smart contract will also be upgraded to include document revocation functionality, which is vital for managing cases where a certificate needs to be withdrawn. To enhance scalability and minimize transaction costs, deployment on a Layer 2 network such as Polygon will be explored. Lastly, developing a dedicated mobile application could make the verification process more convenient and accessible for users around the globe.

REFERENCES

- [1] S. Kadwe .et al, "EduDocs: Document Verification using Blockchain," 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), 2024.
- [2] A. S. Rajasekaran et al., "Blockchain-Based Document Verification Scheme for Enhanced Security and Fraud Control", 2024 International Conference on Emerging Research in Computational Science (ICERCS), IEEE, 2024.
- [3] A. Singh et al., "Blockchain Based Verification of Educational and Professional Certificates," 2023 2nd International Conference on Computational Systems and Communication (ICCS), 2023.
- [4] M. U. Abdullahi et al., "Certificate Generation and Verification System Using Blockchain Technology and Quick Response Code," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 24, no. 1, pp. 37-47, 2022.
- [5] J. G. Dongre et al., "Education Degree Fraud Detection and Student Certificate Verification using Blockchain," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 7, pp. 300-303, 2020.
- [6] A. Deshpande et al., "Blockchain-based Skill Verification System," Proceedings of the International Conference on Sustainable Computing and Smart Systems (ICSCSS 2023), IEEE, 2023.
- [7] D. Uikay et al., "A Blockchain-Based Digital Notary System Provides Reliable and Tamper-Proof Timestamping and Verification Services for Digital Documents: A Review," International Journal for Multidisciplinary Research (IJFMR), vol. 6, no. 2, pp. 1-9, 2024.
- [8] N. Kumawat et al., "Utilizing NFTs to Revolutionize Document Verification and Authentication through Blockchain", 15th ICCNT IEEE Conference, IIT Mandi, IEEE, 2024.
- [9] A. Shende et al., "Enhancing Document Verification Systems: A Review of Techniques, Challenges, and Practical Implementations," International Journal of Computer Engineering & Technology (IJCET), 2024.
- [10] S. Boonkrong, "Design of an Academic Document Forgery Detection System," International Journal of Information Technology, 2024.
- [11] R. Vaidya et al., "Blockchain-Powered Certificate Authentication: Enhancing Trust and Transparency", IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), IEEE, 2024.
- [12] A. Pandey et al., "Blockchain-Based Digital Multimedia Content Authentication System: Using IPFS and Ethereum," 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), 2024.
- [13] A. K. C. et al., "Detection of Fake Physical Certificates Using a Blockchain-Based Certificate Verification and Issuer Validation System," 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), 2024.
- [14] M. Abubakar et al., "Blockchain-Based Platform for Secure Sharing and Validation of Vaccination Certificates," 2021 IEEE International Conference on Security of Information and Networks (SIN), 2021.
- [15] Sun et al. (2020), "Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS," IEEE Access, 2020.
- [16] F. I. Louis et al., "Blockchain-Based Public Key Infrastructure Using Smart Contracts", IEEE International Conference on Data and Software Engineering (ICoDSE), IEEE, 2024.
- [17] A. Vijaya Kumar et al., "A Blockchain-Based Document Verification Model in Freshers Hiring Process", IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2022.
- [18] A. Patel, R. Shivani and M. P B, "Developing a Virtual Diagnosis and Health Assistant Chatbot Leveraging LLaMA3," 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/CSITSS64042.2024.10817034. keywords: {Accuracy;Computational modeling;Documentation;Chatbots;Multilingual;Reliability;Medical diagnosis;Information technology;Medical diagnostic imaging;Testing;API Integration;Health Assistant Chatbot;LlaMA3 Model;Medical NLP;Virtual Diagnosis},
- [19] Reddy, M. P. B., & Narayanappa, S. S. (2024). *Deep reinforcement learning based quality of experience aware for multimedia video streaming*. International Journal of Electrical & Computer Engineering, 14(5), 5209–5220.