Documentation management system and PDF Chat: An Integrated Approach

Vaibhav Devkate, Atharv Divekar, Prasad Hulmukhe(Corresponding Author), Om Randhir
B.E Student (Information Technology)
International Institute of Information Technology (I2IT)
Pune, India

Prachi Nilekar
Professor (Information Technology)
International Institute of Information Technology(I2IT)
Pune, India

Dr. Rajesh Chowdhary
Head - Research & Development, Consultancy and Collaboration | International Relations
Pralhad P. Chhabria Research Center,
International Institute of Information Technology (I2IT)
Pune, India

Sushrut Joshi
Senior Research Associate
Pralhad P. Chhabria Research Center,
International Institute of Information Technology (I2IT)
Pune, India

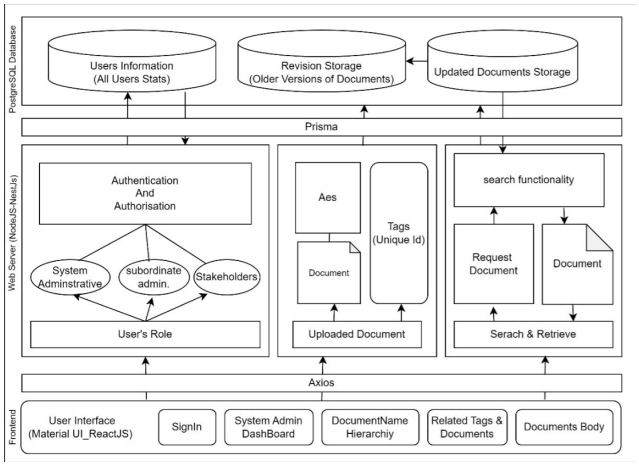
Abstract:**DMS represents the principle of software and hardware solutions that provide many capabilities including data storage, version control, metadata tagging, management and advanced search. These systems are essential tools for businesses, government agencies, schools, doctors, professionals, and individuals looking to improve their information management strategies. By facilitating collaboration in the business environment to ensures data integrity and public compliance management, DMS has proven to be relevant in many ways.DMS represents the principle of software and hardware solutions that provide many capabilities including data storage, version control, metadata tagging, management and advanced search. These are essential tools for businesses, government agencies, schools, doctors, professionals, and individuals looking to improve their datas management strategies. To Build this system we have used ReactJS for frontend, NestJS for Backend, for Database we used PostgreSQL and for the PDF chat we used Langchain in Python.**

Keywords—Documentaion Managemnet system, Access Control, Version Control, PDF Chat, Langchain, LLM, OpenAI, Embeddings

I. INTRODUCTION

A Documentation management system (DMS) in the industry is an important tool for the effective use of the crucial data. The system provides central storage of all project information to ensure security and auditability of electronic information. It is designed to manage big documents, thereby reducing the administrative effort associated with data collection management. DMS allows rapid retrieval of information using specified content and makes this information relevant. engineers, geology Scientists, stakeholders, and analysts can easily access data for remote operations. It is also the first of the assets to provide acces control and updating of information to ensure that only authorized people are allowed to see it, reducing overall data loss. For an industry, DMS can improve coordination, improve management information, provide instant access to information, and reduce risk and liability.

A. Basic Architecture Diagram OF DMS



II. LITERATURE SURVEY

A. ACCESS CONTROL METHODS:

Role-Based Access Control (RBAC) is a crucial tool in the industry, providing strict control over access control. It aligns with the organizational structure of the industry, ensuring that only necessary information is granted to employees. This reduces the risk of unauthorized access to sensitive data and maintains compliance with regulatory requirements. RBAC simplifies access control management, reducing administrative work and IT support required. It also improves operational efficiency by allowing users to perform their jobs more efficiently and autonomously, affecting project success and the company's bottom line. In summary, RBAC in a DMS for the oil and gas industry ensures sensitive information protection, compliance with regulations, and efficient documentation management.

Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role-Based Access Control (RBAC), and their applicability to cloud computing. DAC is traditional but difficult to maintain and scale well. MAC is more secure but requires careful planning and continuous monitoring. RBAC is central to secure processing needs of non-military systems. The paper introduces cloud-specific access control technologies like Attribute-Based Access Control (ABAC), distributed RBAC (dRBAC), and Cloud Optimized RBAC (coRBAC). The choice of access control method depends on the specific requirements of the cloud environment, such as system scale, centralized control, and data sensitivity. [2]

At Some Point the Discretionary Access Controls (DAC) are insufficient for commercial and civilian government organizations, and advocates for Role-Based Access Control (RBAC) as a more central approach to secure processing needs. DAC is suitable for commercial applications and civilian government security, while Mandatory Access Controls (MAC) are suitable for multilevel secure military applications. However, RBAC is more central to secure processing needs of non-military systems. The current set of security criteria, criteria interpretations, and guidelines have been developed by the Department of Defense over twenty years. The Trusted Computer System Evaluation Criteria (TCSEC) is the best-known U.S. computer security standard, designed to prevent unauthorized observation of classified information. RBAC is more appropriate for commercial and civilian government organizations because it supports higher-level organizational policy and centralizes control and maintenance of access rights. It also emphasizes the importance of control over transactions and the principles of least privilege and separation of duties. RBAC is flexible and useful for organizations with large personnel turnover, allowing easy management of access rights. [3]

B. VERSION CONTROL SYSTEMS:

Version control is an Important component of a Documentation Management System (DMS). It is a systematic approach to managing changes and revisions in documents, ensuring that the most up-to-date and accurate information is always available. This also make sure that the pre-existing file and the revision both should be there to keep up the record of changes.

I] Types of Version Control Systems:

There are Mainly two different version control systems as discussed in [3].

1) Centralized version-control system (CVCS) [3]:

There are few key points about Centralized version control system:

a) Repository: There is only one central repository which is the server.

b) Repository Access: Every user who needs to access the repository must be connected via network

c) In a CVCS, files are typically locked by a developer when they need to make changes. This prevents concurrent edits and potential conflicts.

d) CVCS systems maintain a complete history of all changes made to files and directories, including who made the changes and when.

2) Distributed Version-Control System (DVCS) [3]:

There are few key points about Distributed version control system:

a) Repository: Every user has a complete repository which is called local repository on their local computer

b) Repository Access: repository must be connected via network. DVCS allows every user to work completely offline. But user need a network to share their repositories with other users.

c) Version History: Every local copy in a DVCS contains the full version history of the project. This ensures that even if a central server is lost, individual copies can still be used to recover the project's history.

d) Flexibility: DVCS systems offer flexibility in defining workflows. Teams can choose a workflow that best suits their development process, whether it's centralized, feature-based, or topic-based branching.

II] Importance of Version-Control in DMS [2,3]:

a. *Accuracy and Consistency:* Version control helps maintain the accuracy and consistency of documents within an organization. It ensures that everyone is working with the same, current version of a document, reducing the risk of errors due to outdated information.
b. *Traceability:* Version-control provides a complete history of document revisions. This traceability is valuable for auditing, compliance, and accountability purposes, allowing organizations to track who made changes and when.
c. *Recovery from the Disaster:* Accidental deletions or data corruption can be disastrous for a DMS. Version control offers a safety net by allowing you to restore previous versions of documents, ensuring data integrity and continuity in the event of data loss.

III] Benefits

a. *Security:* Many industries require strict adherence to compliance regulations and data security standards. Version control helps organizations maintain compliance by ensuring that document revisions are tracked, audited, and securely stored.
b. *Helps in Decision Making:* Access to historical versions of documents aids in decision-making processes. Teams can review past iterations to understand the evolution of ideas, strategies, and decisions, leading to more informed choices.
c. *Improved productivity:* Version control eliminates the confusion and inefficiency associated with multiple document copies, email attachments, and conflicting edits. This results in increased productivity as team

members can focus on their tasks instead of managing document versions.

C. SECURITY

CryptoJS is like a toolbox for web developers. It's a special set of tools that helps them add strong security features to their websites. This toolbox is written in JavaScript, the language used for web development.

One great thing about CryptoJS is that it gives developers a lot of options. They can use it to easily add strong encryption and decryption to their web applications. Encryption is like turning information into a secret code so that it can't be easily read by unauthorized people. It's handy when you want to keep sensitive data safe, like passwords or personal information.

CryptoJS supports different ways of encrypting information, using various algorithms. Think of algorithms as step-by-step instructions for encoding and decoding data. For example, it supports AES, DES, and Triple DES, which are well-known and widely used encryption methods.

One key feature of CryptoJS is that it works right in the web browser. This means that the encryption and decryption processes happen on the user's device, adding an extra layer of security. It's especially useful when you want to make sure that data is protected as it travels between the user's device and a server or when it's stored on the user's device.

Because it's open-source, CryptoJS is continuously improved by a community of developers. This means it's always getting better and more secure. Its flexibility and ease of use make it a popular choice for developers who want to make their web applications more secure by adding encryption features.

I] Encryption Algorithm:

CryptoJS offers support for different encryption algorithms, and the choice of which one to use depends on the security requirements of the Document Management System (DMS). A commonly used algorithm is the Advanced Encryption Standard (AES), which was proposed by NIST in 2001. It serves as a cryptographic algorithm and encryption standard designed to replace DES, offering robust security features.

AES provides flexibility with key lengths of 128, 192, or 256 bits, allowing for different levels of security. This symmetrical block cipher functions in multiple stages, completing 10, 12, or 14 rounds, effectively handling the encryption and decryption of data. The algorithm processes data in 128-bit blocks and utilizes cryptographic keys of 128, 192, or 256 bits. To ensure security in various applications within the DMS, NIST recommends five modes of operation, each with specific parameters. These parameters are crucial for maintaining the algorithm's security.

II] Encryption:

By successfully integrating CryptoJS, developers can initiate the encryption process within the Document Management System (DMS). Encryption serves as a fundamental security measure to protect sensitive data from unauthorized access. In the encryption process, plaintext data, which is the original and readable form, transforms into ciphertext, an unreadable and secure form. CryptoJS smooth this transformation by utilizing various encryption algorithms like Advanced Encryption Standard (AES), DES, and Triple DES. Developers have the flexibility to specify key lengths and algorithm parameters based on the DMS's security requirements. The encrypted data is protected by cryptographic keys and algorithms, ensuring that even if unauthorized entities gain access to the stored information, they won't be able to understand its contents without the appropriate decryption keys.

III] Decryption:

In the integrated Document Management System (DMS), CryptoJS smooths the decryption process, allowing authorized users to transform encrypted data back into its original, readable form. Decryption works by applying the inverse transformation to the ciphertext, using the same cryptographic key and algorithm parameters employed during encryption. CryptoJS seamlessly manages this process, ensuring that authorized users having the correct decryption keys can access and interpret sensitive information. This crucial step in the cryptographic lifecycle guarantees the confidentiality and integrity of data within the DMS, enabling secure storage and controlled access. Developers must exercise careful management and secure handling of decryption keys, as these keys play an important role in granting permission to the authorized uses to retrieve and use the protected data effectively.

IV] Secure Communication:

When a Document Management System (DMS) is used in client-server communication, ensuring the security of data transmission becomes crucial. CryptoJS becomes a valuable tool by encrypting data before it is sent. This encryption process adds an extra layer of security, protecting sensitive information from potential threats during transmission. Additionally, we can implement a secure communication protocol such as HTTPS for securing a chaneel. HTTPS encrypts the data during transmission, preventing eavesdropping attempts and reducing the risk of man-in-the-middle attacks. By integrating CryptoJS into the DMS, developers can create a robust framework that not only encrypts data before transmission but also ensures secure communication channels.

V] Comparison CryptoJs vs Bouncy Castle:

CryptoJS is great for web applications because it helps with encrypting things on the user's device. It works well in web development, supporting different encryption methods like AES, DES, and Triple DES. The people who work on CryptoJS actively update and support it, making it a good choice for web projects.

On the other hand, Bouncy Castle is a Java-based tool that's good for more complex tasks, especially in server-side applications and Android development. It has a lot of capabilities for different encryption methods. People who use Java or C# find it reliable, and it's easy to include in their projects.

In short, CryptoJS is a better fit for web applications, especially when we need to do encryption directly in the user's web browser. It's easy to use with JavaScript, which is a common language for web development. CryptoJS is focused on making encryption simple for web developers.

III. METHODOLOGY

In Documentation Management systems, The Basic task is to organize the documents in such way where users feel more satisfied accessing documents of need. There are basic methodologies which need to be there in documentation management system are access control, version control, security of the document.

A. Access Control:

Role-Based Access Control (RBAC) is a security model that is ideal for the industry's Documentation Management System (DMS). It provides a structured approach to access control, aligning with the industry's organizational structure. RBAC assigns roles to users and grants permissions for specific actions, simplifying the management of complex permission structures. This hierarchical model can limit potential damage in case of a security breach by segmenting access and reducing the attack surface.

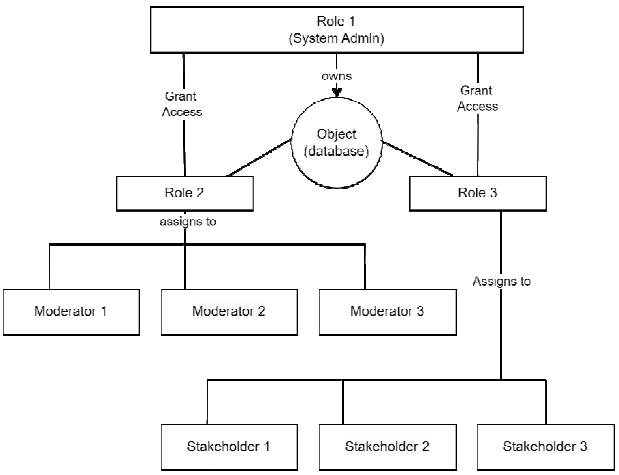
In the context of a DMS for industry, there are three roles: System Admin, Moderator, and Stakeholder.

- System Admin: Has the highest level of access and is responsible for managing the entire system, including user management, role assignment, and system configuration.
- Moderator: Can create, update, and read documents but does not have the authority to assign roles or manage users.
- Stakeholder: Can access the documentation of various entities, read documents. They do not have the authority to assign roles or manage users.

The authentication process ensures that only valid users can access the system, and the authorization process determines what each user can do within the system based on their assigned role. This ensures that each user has the appropriate level of access to perform their job duties, which is particularly important in the oil and gas industry where sensitive information must be protected.

RBAC is managed by the INCITS committee CS1, which oversees its updates and revisions. The model has been revised to incorporate features of attribute-based access control (ABAC) to provide more scalable and flexible access management.

The use of RBAC in a DMS for industry can help to ensure that sensitive information is protected, that compliance with regulations is maintained, and that the DMS is an efficient tool for managing documentation.



B. Version Control:

Version Control in context of documentation management system is nothing but storing all the revisions of a specific document over the time. Suppose we have documentation of an equipment then we should be having all the revised data to get knowledge about the updates which are done over the time.

For this System centralized version control system is more useful we need to have central storage so that the data integrity remains intact.

Version control helps maintain the accuracy and consistency of documents within an organization. It ensures that everyone is working with the same, current version of a document, reducing the risk of errors due to outdated information.

In our system we made use of centralized version control because the data is stored centrally and to maintain the Integrity and the security of the data.

C. Security:

Cryptographic techniques is used to ensure the security and integrity of sensitive data within a Document Management System (DMS). crypto-js emerges as a robust cryptographic library in JavaScript, offering a variety of algorithms for encryption and decryption.

1. Installing crypto-js

To integrate crypto-js into the DMS, the initial step involves installing the library. This can be accomplished through a package manager, such as npm, with the following command:

```
npm install crypto-js
```

2. Importing crypto-js in the DMS

Once installed, the library needs to be imported into the DMS codebase. This is typically achieved through the following import statement:

```
const CryptoJS = require('crypto-js');
```

3. Encryption and Decryption Functions

To secure documents stored in the DMS, encryption and decryption functions must be implemented. The crypto-js library provides various algorithms such as AES, DES, and TripleDES. For instance, utilizing AES encryption:

```
// Encryption function
function encryptDocument(document, secretKey) {
const encryptedDocument = CryptoJS.AES.encrypt(document, secretKey).toString();
return encryptedDocument;
}

// Decryption function
function decryptDocument(encryptedDocument, secretKey) {
const decryptedBytes = CryptoJS.AES.decrypt(encryptedDocument, secretKey);
const decryptedDocument = decryptedBytes.toString(CryptoJS.enc.Utf8);
return decryptedDocument;
}
```

4. Key Management

A crucial aspect of cryptographic operations is secure key management. The DMS should implement robust mechanisms for generating, storing, and rotating cryptographic keys. Key management practices should adhere to industry standards to mitigate security risks.

5. Integration with DMS Workflow

Integrating crypto-js into the DMS workflow involves invoking the encryption and decryption functions at appropriate stages. For instance, encrypting a document before storage and decrypting it when retrieved for authorized access.

The integration of crypto-js within the DMS serves as a foundational pillar for securing sensitive documents. The use of well-established cryptographic libraries enhances the overall security posture of the document management infrastructure, safeguarding against unauthorized access and data breaches.

IV. PDF CHAT

PDF chat is designed to extract and interactively retrieve information from PDF documents. The application leverages several state-of-the-art natural language processing (NLP) and document processing techniques to enable users to ask some questions according to requirement and receive answers directly from PDF content. The quality of the answers received by the user is enhanced with the use of artificial Intelligence.

A. Text Extraction and splitting:

The whole pdf is kind of a large data to handle and process at one go to minimize this and increase the accuracy we can divide the whole pdf into multiple chunks so the processing part will be easier, and the accuracy will be more.

For the first step we extract all the text present in the pdf.

For the second we fix the size of a chunk which is nothing but a smaller text paragraph.

We divide the pdf by giving the chunk size and some overlapping to not lose any data in the pdf file. For Example, we are making chunk of 800 and giving 200 as overlapping then first chunk will be from 0-800 and the second chunk will start from 600-1400 as we are giving 200 as overlapping. This ensures reduced data loss as some information is processed twice.

B. OpenAI Embeddings:

OpenAI embeddings typically refer to word embeddings, which are numerical representations of words in a format that can be used by machine learning models, such as neural networks. Word embeddings are a fundamental concept in natural language processing (NLP) and are used to convert words or text into vectors of real numbers. These vector representations capture semantic relationships between words and are useful in various NLP tasks, including text classification, language modeling, machine translation, and sentiment analysis.

These word embeddings are employed to transform the raw text data into numerical vectors, making it easier to perform semantic search and similarity calculations.

OpenAI embeddings play a critical role in performing semantic search within the document corpus. When a user submits a query, OpenAI embeddings are used to transform the query into a vector, and then a similarity search is conducted in the vector space to identify relevant data.

C. FAISS

FAISS (Facebook AI Similarity Search) is an open-source library formed by Facebook AI Research for precise similarity search and clustering of large datasets. It is primarily designed for searching for vectors in high-dimensional vector spaces, making it useful in various applications, including machine learning, computer vision, and natural language processing. FAISS is known for its speed and scalability in similarity search tasks.

The vectors generated by the embeddings are then indexed using FAISS. FAISS creates an index structure that allows for fast and efficient similarity searches. It organizes the vector data in a way that makes it optimized for nearest-neighbor searches, which is crucial for matching user queries with relevant document segments.

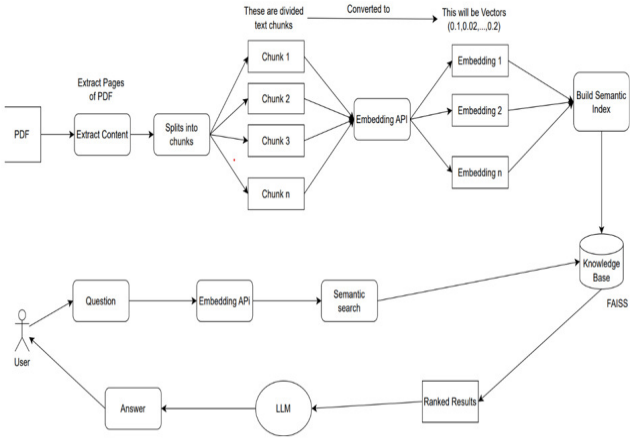
When a user submits a query through the web interface, the application uses FAISS to perform a similarity search. FAISS finds the most relevant text segments (chunks) based on the vector representations of the user's query. This allows the system to quickly identify which documents contain content relevant to the user's question.

After retrieving the most relevant text segments, the question answering chain is used to generate answers to the user's query. These answers are then presented to the user through the web interface.

D. Langchain:

Langchain is a newer framework for applications which are powered by language models. Large-language models (LLMs) can be said as core component of the Langchain framework. In this PDF Chat Model used the langchain’s large language model.

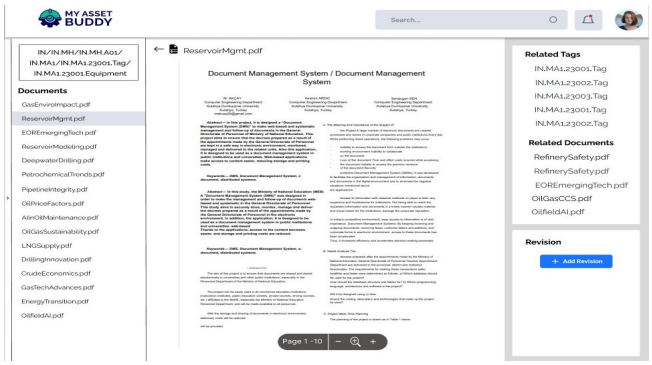
E. Architecture Diagram Of PDF CHAT:

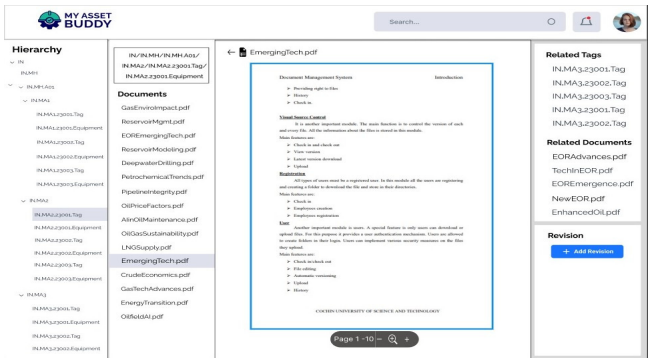


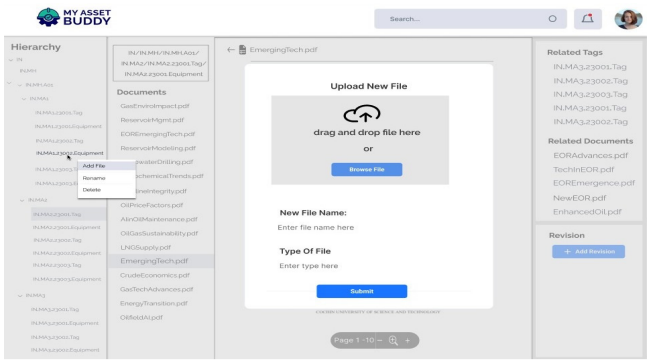
F. Implemented code:

```
def chatapp(query):
    os.environ["OPENAI_API_KEY"] = ""

    pdfreader = PdfReader('sample.pdf')

    # read text from pdf
    raw_text = ''
    for i, page in enumerate(pdfreader.pages):
        content = page.extract_text()
        if content:
            raw_text += content

    text_splitter = CharacterTextSplitter(
        separator = "\n",
        chunk_size = 800,
        chunk_overlap  = 200,
        length_function = len,
    )
    texts = text_splitter.split_text(raw_text)

    embeddings = OpenAIEmbeddings()

    document_search = FAISS.from_texts(texts, embeddings)

    chain = load_qa_chain(OpenAI(), chain_type="stuff")

    docs = document_search.similarity_search(query)
    return(chain.run(input_documents=docs, question=query))
```

V. SYSTEM DESIGN







VI. CONCLUSION

In the above paper we've mentioned various methodologies about the document-management system like how we are able to practice access control to the system, how can we add version control to the document, to store the revisions and how the security algorithm we can implement. As for access control we are using Role Based access control, like we are having different roles with different permissions. For Version control we are making use of Centralized version control system and for security we are making use of CryptoJS. We have also seen the pdf chat model where if we want some particular information about something in the document, then we are able to simply ask the question and the model will give the answer correctly. As we are using langchain's LLM the accuracy of the Langchain-model is nearly 92.5% so we simply can clearly say that our model is also correct in giving the right solution to the question asked.

VII. REFERENCES

[1] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[2] Dmitriev, Sviatoslav O., Dmitrii A. Valter, and Artemii M. Kontsov. "System for Efficient Storage and Version Control of Arbitrary File Collections." 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2020.

[3] Zolkifli, Nazatul Nurlisa, Amir Ngah, and Aziz Deraman. "Version control system: A review." Procedia Computer Science 135 (2018): 408-415.

[4] Xu, D. and Zhang, Y., 2014, June. Specification and analysis of attribute-based access control policies:An overview. In 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion (pp. 41-49). IEEE.Zolkifli, Nazatul Nurlisa, Amir Ngah, and Aziz Deraman. "Version control system: A review." Procedia Computer Science 135 (2018): 408-415.

[5] Liddell Henry George, Scott Robert, Jones Henry Stuart, McKenzie Roderick (1984). A Greek-English Lexicon. Oxford University Press. page 827.

[6] Agrawal, Monika (2012). A Comparative Survey on Symmetric Key Encryption Techniques. International Journal on Computer Science and Engineering. 4: pages 877–882. CiteSeerX 10.1.1.433.2037

[7] El Sibai, R., Gemayel, N., Bou Abdo, J., & Demerjian, J. (2020). A survey on access control mechanisms for cloud computing. Transactions on Emerging Telecommunications Technologies, 31(2), e3720.

[8] Cho, V. (2008). A study of the effectiveness of electronic document management systems. International journal of information technology and management, 7(3), 327-352.

[9] Alade, S. "Design and Implementation of a Web-based Document Management System." Information Technology and Computer Science 2 (2023): 35-53.

[10] Livina, I. S. I., Chukwuemeka, A. E., Nnanna, E., Obi, N., Ikechukwu, I. S., & Ogonnaya, A. B. (2020). A Web Based Document Encryption Application Software for Information Security in Tertiary Institutions. Journal of Cybersecurity and Information Management (JCIM) Vol, 4(1), 26-35.

[11] Simarmata, J., Limbong, T., Ginting, M.B., Damanik, R., Nasution, M.I.P., Hasugian, A.H., Mesran, M., Sembiring, A.S., Hutahaean, H.D., Taufik, I. and Hasugian, P.M., 2018. Implementation of AES Algorithm for information security of web-based application. Int. J. Eng. Technol, 7(3.4).

[12] Sambetbayeva, Madina, Inkarzhan Kuspanova, Aigerim Yerimbetova, Sandugash Serikbayeva, and Shynar Bauyrzhanova. "Development of Intelligent Electronic Document Management System Model Based on Machine Learning Methods." Eastern-European Journal of Enterprise Technologies 1, no. 2 (2022): 115.

[13] Abidin, S.S.Z. and Husin, M.H., 2018. Improving accessibility and security on document management system: A Malaysian case study. Applied Computing and Informatics, 16 (1–2), 137–154.

[14] Ismael, Arkan, and Ibrahim Okumus. "Design and implementation of an electronic document management system." Mehmet Akif Ersoy Üniversitesi Uygulamalı Bilimler Dergisi 1.1 (2017): 9-17.