Automation of Assigning Managed Metadata for GNOSIS Search Optimization

Ashwini Khambe
EXTC Department
Atharva College of engineering
Mumbai, India
khambeashwini-extc@atharvacoe.ac.in

Pooja Dubey
EXTC Department
Atharva College of engineering
Mumbai India
dubeypooja-extc@atharvacoe.ac.in

Surbhi Dhamankar
EXTC Department
Atharva College of engineering
Mumbai India
damankarsurbhi-extc@atharvacoe.ac.in

Payal Kheur
EXTC Department
Atharva College of engineering
Mumbai, India
kheurpayal-extc@atharvacoe.ac.in

Prof. Mahalaxmi Palinje
Assistant Professor, EXTC Department
Atharva College of engineering
Mumbai, India
mahalaxmipalinje@atharvacoe.ac.in

Abstract: "Automation of Assigning Managed Metadata for GNOSIS Search Optimization" is a project developed to address inefficiencies in document classification within GNOSIS, an enterprise document management system built on Microsoft SharePoint. Manual metadata tagging is time-consuming, error-prone, and leads to inconsistent data, which negatively impacts searchability and content organization. This project proposes an automated solution using C# scripting and the Microsoft Graph API to seamlessly interact with SharePoint Online.

The system leverages the Managed Term Store to access a predefined taxonomy. Based on document names or content logic, relevant metadata tags are applied automatically. Development was carried out in Visual Studio Code, using the .NET framework for building a lightweight, scalable, and maintainable solution. The Microsoft Graph API enables secure access to SharePoint document libraries, allowing the script to retrieve file properties and assign metadata programmatically.

This automation improves document discoverability, ensures consistent tagging, enhances collaboration, and reduces manual workload. The project highlights how modern development tools and APIs can be effectively utilized to optimize metadata management in enterprise environments.

Keywords- Managed Metadata, Gnosis Platform, Metadata Tagging, C# Scripting, Search Optimization, Microsoft 365 Integration, SharePoint Content Types, Automated Tagging Systems, Visual Studio Code (VS Code)

1 INTRODUCTION

In the modern enterprise ecosystem, the volume of digital documents created and managed continues to grow exponentially. Efficient document classification is essential for ensuring rapid access, effective collaboration, and regulatory compliance. GNOSIS, a document-centric platform built on SharePoint, faces challenges in maintaining structured metadata due to manual intervention. Manually tagging documents often leads to inconsistency, inaccuracy, and reduced productivity. This calls for a reliable and automated solution to manage metadata tagging efficiently.

2 NEED AND MOTIVATION

As digital content grows rapidly in organizations, managing documents efficiently has become increasingly important. The GNOSIS platform, which uses SharePoint for document storage, faces challenges due to manual metadata tagging. This process is often inconsistent, slow, and prone to human error. Users struggle to locate files quickly, leading to productivity loss and workflow delays.

The need for this project arises from the goal to streamline metadata tagging by automating the entire process. Automation reduces human effort, enhances accuracy, and ensures consistency in document classification. Using C# and Microsoft Graph API, the system integrates directly with SharePoint's Term Store. This approach ensures documents are properly tagged without user input. The motivation is to improve search efficiency, boost productivity, and create a more organized and scalable document management system.

3. LITERATURE SURVEY

1) Molnar, in the paper *"Automated Metadata Tagging for SharePoint and Office 365"*, presents a detailed approach to improving metadata quality through automation techniques in Microsoft SharePoint environments. The paper explores both rule-based and Natural Language Processing (NLP)-based tagging methods to enhance the consistency and usability of metadata across enterprise document libraries. A significant contribution of the work is the introduction of the Aqua-forest Searchlight Tagger tool, which supports large-scale, automated tagging by leveraging text zones, form fields, and term store mapping. The study emphasizes the importance of structured metadata in improving search accuracy, user experience, and long-term content governance in cloud-based document management systems.

2) Skluzacek et al., in their paper *"Automated Metadata Extraction: Challenges and Opportunities"*, discuss the practical difficulties of generalizing metadata extraction systems across diverse scientific data repositories. The authors present Xtract, a scalable metadata extraction tool designed for large-scale, heterogeneous file systems. The system uses a combination of lightweight extractors and statistical models to isolate file types, extract metadata, and rate its

quality using completeness, relevance, entropy, and readability as key metrics.

3) Crossley et al**.** in their paper *"A Study on Metadata Tagging for Tracking Original File Information within the Cloud"* explore the role of metadata, particularly EXIF data, in cloud-based forensic investigations. They highlight how cloud storage can alter or erase metadata, making it harder to trace file origins. The study demonstrates methods and tools for extracting metadata from virtual environments. It emphasizes the need for collaboration between cloud service providers and forensic experts. The work supports the importance of metadata integrity in secure digital evidence management.

4) Khalilian et al.**,** in their paper *"Document Classification Methods"*, review various supervised and unsupervised techniques for organizing textual data. They explore clustering algorithms such as k-means and fuzzy c-means, along with hybrid methods like PSO-based models. The study emphasizes the role of feature selection techniques in improving classification accuracy. Their work contributes to the foundation of automated document organization systems.

5) Angus et al.**,** in their paper *"Document Classification with DocBERT"*, investigate deep learning models for classifying long legal documents such as U.S. Supreme Court decisions. The study evaluates multiple NLP models including DocBERT, fastText, CNNs, and LSTMs using various chunking techniques to handle lengthy text inputs. Results show that DocBERT achieves the highest classification accuracy but requires significantly more training time. This research supports the application of transformer-based models for metadata tagging and document organization in knowledge systems.

6) In the paper *"Introducing the Azure Services Platform"*, Chappell outlines Microsoft's early vision for cloud computing through key components such as Windows Azure, .NET Services, SQL Services, and Live Services. The work explains how these elements enable scalable application hosting, secure identity management, and cross-device data synchronization. It supports hybrid deployment models, integrating cloud and on-premises applications. This foundational study paved the way for modern cloud-based development and service-oriented architectures.

7) Leila Moeller provides practical insights into querying "SharePoint items using the Microsoft Graph API", emphasizing advanced data retrieval techniques. The paper examines OData query options like $filter, $expand, and $select to manage various field types, including lookup and managed metadata. It also addresses challenges in accessing complex SharePoint list structures programmatically. This work contributes to the efficient management of SharePoint data within Microsoft 365 environments.

8) Laurila, in his thesis *"Tunnistautuminen Opentuntiin Microsoft Graph APIn Avulla"* [2], implements an authentication system for a web application using Microsoft Graph API and Azure AD. He uses the OAuth 2.0 authorization code grant to enhance security and simplify user access. The study shows how centralized authentication can improve scalability and user experience in educational platforms.

9) A. Jacobsson, in his thesis *".NET Development for the Web using Microsoft Office SharePoint Server 2007 and ASP.NET"*, explores methods for integrating Excel-based desktop solutions with web technologies. He evaluates SharePoint 2007's Excel Services, highlighting its limitations with macros and external data. The study proposes using COM automation and UDFs as workarounds for unsupported features. His work offers a foundational perspective for web-enabling enterprise spreadsheet applications.

10) Graser and Burel, in their article "Metadata Automation: The Current Landscape and Future Developments", discuss evolving workflows for automating metadata in libraries using tools like Python, XSLT, and OpenRefine. They explore machine learning's role in automatic image annotation (AIA) and metadata enhancement. The paper highlights the growing need for technical skills among metadata librarians. Their work outlines key trends shaping future cataloging and semantic web integration.

4. OBJECTIVE

The objective of this project is to automate the assignment and management of metadata in SharePoint document libraries, thereby reducing manual effort and improving metadata consistency. The system aims to enhance GNOSIS search functionality by ensuring structured and accurate metadata for improved content discovery and faster document retrieval. By leveraging Microsoft Graph API, SharePoint, C#, and .NET, the project integrates Microsoft technologies to provide a seamless, scalable solution. Additionally, the system will incorporate secure authentication and permissions through Azure Active Directory to ensure safe and efficient document management, while optimizing search accuracy and accessibility for end users.

5. METHODOLOGY

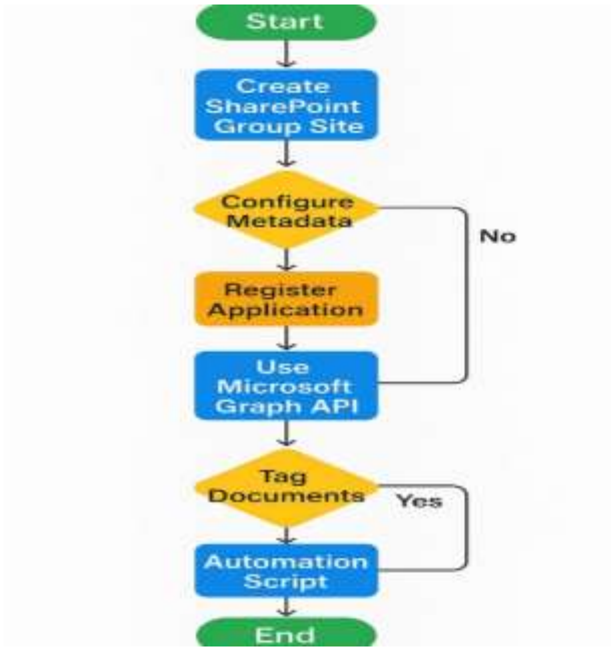
The methodology of this project involves the development of an automated metadata tagging system for SharePoint using Microsoft Graph API and C#. Initially, the functional requirements were gathered to identify the need for automating the tagging process and enhancing the GNOSIS search accuracy. The system was designed to be scalable and secure, using a modular architecture that integrates with SharePoint through Microsoft Graph API. Azure Active Directory was implemented for authentication, ensuring that only authorized access is permitted. The automation logic was built on predefined rules, where document names trigger specific metadata tags. This logic helps in maintaining consistency and reducing manual efforts. The solution was implemented using the .NET platform and tested across different phases including unit, integration, and system testing. A basic user interface was also developed for administrative control and monitoring. The deployment phase included performance optimization and user feedback collection. Future improvements are planned, such as adding AI-based metadata suggestions and expanding the system for broader enterprise use.

6. IMPLEMENTATION

The system was developed using C# and the .NET framework, leveraging Microsoft Graph API to interact with SharePoint document libraries. Azure Active Directory was used for secure authentication via client credentials, ensuring that only authorized access was granted. The application retrieves documents from a specified library and applies metadata tags based on predefined file-naming conventions. Logic was implemented to identify keywords within file names and assign corresponding metadata values automatically. The tagging process was executed using Graph API's update methods, which modify list item fields directly. The solution was modularized for clarity and ease of maintenance, with separate components for authentication, rule processing, and logging. Rigorous testing was carried out to verify accuracy, performance, and integration reliability. Finally, the system was deployed in a SharePoint environment and validated with real-time document sets to ensure its effectiveness in streamlining metadata management.

Fig.1. Block Diagram of Metadata Tagging Workflow

This block diagram outlines an automated workflow for managing and tagging documents within a SharePoint environment. It begins by creating a SharePoint group site and configuring the necessary metadata. After registering an application in Microsoft Azure, the workflow integrates with the Microsoft Graph API for document management. Documents are then evaluated for appropriate metadata tagging, with automation scripts applied for bulk tagging. The process ensures consistent categorization and enhanced document management. The workflow concludes once the automation scripts complete their execution, resulting in a well-structured document repository.



1. **Initiation of Workflow**: The process begins with the initialization of a document management workflow, which is designed to support automated tagging and handling of documents within a collaborative environment.
2. **SharePoint Group Site Creation:** A dedicated SharePoint group site is established to serve as a centralized repository for document storage and collaboration. This site provides the foundational infrastructure for content management and access control.
3. **Metadata Configuration**: Metadata properties are configured to enable document categorization and efficient retrieval. This step ensures that the necessary classification attributes are defined prior to document ingestion or processing.
4. **Application Registration**: The next phase involves registering an application with Microsoft Azure Active Directory. This is required to obtain authentication credentials and permissions to access SharePoint resources through Microsoft Graph API.
5. **Integration with Microsoft Graph API**: Using the acquired credentials, the workflow integrates with Microsoft Graph API to facilitate programmatic access to the SharePoint site. This API enables the automation of document operations such as retrieval, updates, and tagging.
6. **Document Tagging:** At this stage, documents are evaluated for the presence of appropriate metadata tags. If the tagging is incomplete or missing, the workflow redirects to reprocess or update metadata configurations**.**
7. **Automation Scripts:** Upon successful tagging, automation scripts are executed to apply the metadata consistently across documents. Automation scripts developed in C# using the .NET framework in Visual Studio Code are executed to apply metadata uniformly. These scripts manage bulk tagging operations, ensuring consistency across documents and aiding efficient document lifecycle management.
8. **Termination of Process:** The workflow concludes after the automation script completes execution, resulting in a structured, metadata-enriched document repository that supports enhanced searchability and governance.

7. SOFTWARE TOOLS

1. **Visual Studio Code:**

   A Visual Studio Code is a lightweight and versatile code editor that supports multiple programming languages, including C#. It was used as the primary development environment for writing and managing the automation scripts in this project.

2. **NET SDK (Core/9.0):**

   The .NET SDK provides the libraries, runtime, and tools necessary to build and execute C# applications. It enabled the development of cross-platform scripts that interact with the Microsoft ecosystem.

3. **C# Programming Language:**

   C# was chosen for implementing the automation logic due to its compatibility with the .NET platform and strong support for Microsoft APIs. Its object-oriented features made it suitable for building structured, maintainable code.

4. **Microsoft Graph API:**

The Microsoft Graph API served as the main communication channel between the automation script and SharePoint. It allowed secure access to files and metadata within the document library, enabling dynamic tagging operations.

5. **Azure Active Directory (Azure AD):**

   Azure AD was used to handle secure authentication for the application. It provided access tokens and credentials necessary for calling Graph API endpoints safely within the organization's network.

6. **SharePoint Online:**

   SharePoint Online acted as the central repository for all documents involved in this project. The automation system worked on files stored in its document libraries by reading and updating their metadata.

7. **NuGet Package Manager:**

   NuGet was used to install and manage essential .NET libraries like Microsoft Graph and Azure.Identity. These packages enabled connectivity with Microsoft services and simplified integration.

8. FUTURESCOPE AND APPLICATION

A. Future scope

Another promising area of future development involves the implementation of **predictive tagging models** that learn from user behavior and previous tagging decisions. These models can adapt over time and improve the system's ability to assign metadata accurately, especially in dynamic environments where document types and structures frequently change. Moreover, incorporating **workflow automation tools**, such as Microsoft Power Automate, could enable metadata tagging to trigger additional actions like document approvals, archival, or sharing, thereby streamlining entire document lifecycle processes.

The system could also benefit from **integration with data compliance tools**, helping organizations meet regulatory requirements by tagging sensitive or confidential data appropriately. In highly regulated industries, automated tagging aligned with compliance standards can significantly reduce risk and audit efforts. Additionally, enabling **cross-language support** would allow the automation engine to interpret and tag documents in multiple languages, making the system suitable for global organizations with multilingual content repositories.

To further support enterprise-level adoption, **load balancing and distributed processing** capabilities could be implemented to handle large volumes of documents simultaneously without performance degradation. In future iterations, **blockchain-based tracking** could be explored for maintaining an immutable record of tagging actions, increasing transparency and trust in document handling processes. Lastly, collaboration with knowledge management systems and enterprise search platforms can maximize the impact of metadata by improving how organizational knowledge is stored, retrieved, and reused across departments. Future iterations of the system could incorporate user feedback loops, allowing the tagging algorithm to refine its accuracy based on user interactions. Additionally, support for real-time metadata synchronization across collaborative platforms can ensure consistency during simultaneous edits. Incorporating audit trails for all metadata changes can also enhance transparency and accountability within document workflows.

B. Application.

Applications of IoT based Health Monitoring System using Arduino and generic ESP8266 are as follows

1. **Enterprise Content Management:** The automated tagging system can significantly enhance document classification and retrieval in large organizations by ensuring consistent metadata across repositories, reducing time spent on manual indexing.
2. **Improved Knowledge Management Systems:** By organizing files with standardized tags, the system supports better knowledge discovery and sharing, which is crucial in collaborative work environments such as intranets and digital workplaces.
3. **Compliance and Information Governance:** Accurate and automated metadata assignment supports compliance with data classification standards and regulatory policies by ensuring that documents are properly categorized based on their content.
4. **Enhanced Search Capabilities in SharePoint:** Structured metadata enables more accurate and faster search results in platforms like SharePoint and GNOSIS, improving user experience and reducing the overhead of information retrieval.
5. **Document Lifecycle Automation:** The system can assist in automating various stages of the document lifecycle—such as retention, archival, and disposal—by tagging documents according to their type and business relevance.
6. **Scalable Metadata Management:** This solution can be scaled across departments or business units, handling large volumes of documents efficiently without the need for manual input, which is beneficial for growing organizations.
7. **Integration with Business Intelligence Tools:** Tagged data can be further integrated with reporting or analytics platforms, allowing organizations to derive insights based on document categories, usage patterns, or departmental activity.
8. **Support for Multi-Domain Use Cases:** The solution is flexible and can be customized for industries such as finance, healthcare, education, or construction, where metadata is critical for organizing compliance documents, research papers, or project files.

9. CONCLUSION

The project successfully addresses the challenge of manual metadata management in SharePoint by introducing an automated solution using Microsoft Graph API and C#. By integrating rule-based tagging and secure access through Azure Active Directory, the system ensures consistent and efficient document classification. This automation not only reduces human error and administrative overhead but also significantly improves the accuracy and speed of information retrieval in enterprise environments. The flexibility of the solution allows it to scale across various departments and domains, making it a valuable asset for organizations aiming to streamline content organization and enhance search capabilities. Overall, this approach demonstrates a practical application of automation in enterprise content management, laying a strong foundation for future enhancements such as intelligent tagging using AI and broader integration across Microsoft 365 services.

Furthermore, the use of standard APIs and secure authentication mechanisms enhances system interoperability and data security. The modular structure of the implementation allows future extensions with minimal effort. As organizations continue to handle growing volumes of digital data, solutions like this offer long-term benefits in terms of scalability and performance. Real-time tagging not only accelerates workflows but also ensures that documents remain compliant with classification policies. The successful deployment of this solution reflects its practical viability and technical soundness. With ongoing advancements in AI and natural language processing, the system can evolve to support semantic tagging. Future work can also explore integration with external document management platforms and cloud storage services.

10. REFERENCE

1. **Molnar, D. (n.d.)**. Automated metadata tagging for SharePoint and Office 365. *International Journal of Cloud Applications and Services*, 6(1), 22–30.
2. **Skluzacek, B., Chard, A., & Foster, I. (2017).** Automated metadata extraction: Challenges and opportunities. Proceedings of the IEEE International Conference on e-Science, 13(1), 176–185.
3. **Crossley, S., James, J., & Zheng, H. (2020**). A study on metadata tagging for tracking original file information within the cloud. Journal of Digital Forensics, Security and Law, 15(3), 1–14.
4. **Khalilian, J., Mustapha, M., & Sulaiman, M. N. (2010).** Document classification methods. International Conference on Computer Engineering and Applications, 2(1), 233–237.
5. **Angus, D., Smith, M., & Nankervis, S. (2021).** Document classification with DocBERT. Australasian Language Technology Association Workshop, 19(1), 45–54.
6. **Chappell, D. (2008).** Introducing the Azure Services Platform. Microsoft Architecture Journal, 21(4), 10–18.
7. **Moeller, L. (2020).** Querying SharePoint items using the Microsoft Graph API. Journal of Information Systems and Technology Management, 17(2), 101–108.
8. **Laurila, A. (2018).** Tunnistautuminen Opentuntiin Microsoft Graph APIn Avulla. Bachelor’s Thesis, Turku University of Applied Sciences, Turku, Finland.
9. **Jacobsson, A. (2009).** .NET Development for the Web using Microsoft Office SharePoint Server 2007 and ASP.NET. Master’s Thesis, Chalmers University of Technology, Gothenburg, Sweden.
10. **Graser, M., & Burel, M. (2019).** Metadata automation: The current landscape and future developments. VRA Bulletin, 45(2), 34–47.