Journal of Engineering Research and Education
Volume 15, 2023 [17-26]

A Framework Design of Web-based Knowledge Management System with NoSQL database

R. J. Ong[1], R. A. A. Raof [1,2], S. Sudin [1, 2] and K. Y. Choong[1]

[1]Faculty of Electronic Engineering and Technology, Universiti Malaysia Perlis,Pauh Putra Campus, Malaysia
[2]Sports Engineering Research Centre, Universiti Malaysia Perlis, Pauh Putra Campus, Malaysia.

ABSTRACT

In today's dynamic and information-rich environment, effective knowledge management has emerged as a critical determinant for organizations seeking to take advantage of intellectual resources. Nevertheless, it is unrealistic to expect experts who lack a background in Information and Communications Technology (ICT) to possess an advanced level of technological proficiency or linguistic analytical skills. This study provides a comprehensive framework targeted at developing a Web-based Knowledge Management System (WKMS) coupled with chatbot application that is precisely tuned to meet the complex challenges of organizing, disseminating, and utilizing knowledge related to agriculture sector. This paper elaborates on a framework from both functional and technical perspectives, including the identification of knowledge origins, the establishment of mechanisms for capturing and sharing knowledge, and the facilitation of collaborative knowledge generation. This study employing a methodological approach and adopts a three-tier architecture for framework construction, coupled with text processing for data preparation. The findings emphasize that a successful WKMS should prioritize not only technology infrastructure but also strategies for building a culture favorable to information exchange. This paradigm lays the groundwork for enhanced discovery, innovation, and decision-making in a variety of professional fields by fusing theoretical insights and pragmatic considerations.

Keywords: Framework, knowledge management system, three-tier architecture

1. INTRODUCTION

A framework serves as a structured and systematic approach, providing developers with a foundational structure to simplify the creation of software applications, systems, or solutions [1]. By offering reusable code, established design patterns, and standardized practices, frameworks streamline the development process [2]. These frameworks can either be general-purpose, catering to a wide array of application types, or specialized for specific domains or tasks. The primary function is to abstract complex underlying processes, allowing developers to concentrate on implementing the unique features of the applications while bypassing the intricacies of lower-level technical details.

In the realm of framework design, the terms "functional framework" and "technical framework" frequently emerge in the literature [3], [4]. These terms pertain to distinct facets of software system design and implementation. The "functional framework" addresses inquiries such as "What are the system's intended achievements?" and "What are the core functionalities it needs to support?". Conversely, the "technical framework" delineates the technical dimensions of constructing, deploying, and sustaining the software system.

International Business Machines (IBM), the renowned technology giant, provides a wide range of services that include a mix of technology products, consulting services, and experience, all aimed at catalysing enterprises' digital transformation. When compared to single-tier and other multi-tier options, IBM's newest insights show that three-tier architecture is the most generally adopted multi-tier application architecture.

*Corresponding Author: suhizaz@unimap.edu.my

In terms of architectural designs, one-tier architecture combines all components into a single entity, whereas two-tier architecture comprises of display and data layers. Both models have a downside in the form of strong coupling between the client and server levels, which prevents components from being reused or modified autonomously. As [5] points out, switching to a different technology or platform within these designs can be a difficult task.

When entering the realm of multi-tier architectures, notably the n-tier architecture,structures with numerous tiers are encountered. However, setups with more than three layers are unusual due to limited benefits and significant downsides such as degraded performance, greater complexity, and increased operational expenses.

Emphasising the benefits of the three-tier design, its segmentation into display, businesslogic, and data tiers allows for faster development through parallel communication across various teams. Each layer benefits from being adapted to its individual requirements, which is supported by the most recent tools and languages [5]. This architecture not only accelerates development but also improves scalability by allowing independent scaling of tiers to satisfy varying demands. Furthermore, the three-tier configuration improves system resilience by confining failures to specific tiers, effectively limiting the extent of widespread disruptions.

The MVT design pattern is used in web frameworks, particularly Django, a Python framework. Table 1 highlights a comparison of the MVC (Model-View-Controller) and MVT (Model-View-Template) patterns, indicating distinct ways to structuring software programmes as proven by several research [6]–[8]. The controller orchestrates interactions between the model and view in the MVC paradigm, whereas views manage HTTP requests and answers in MVT. MVC requires explicit code for control-related tasks, whereas MVT outsource controller responsibilities to the framework, resulting in a more loosely connected architecture. Because of this decoupling, modifications are difficult in the MVC paradigm but easy in the MVT model. Modifications are challenging within the MVC model but seamless within MVT due to this decoupling. MVC is well-suited for large applications, while MVT proves adaptable for projects of varying scopes. MVC's clearly defined flow contrasts with the occasional intricacy associated with understanding MVT. Notably, unlike MVT, MVC lacks URL mapping. Examples of MVC include ASP.NET MVC and Spring MVC, whereas the MVT paradigm finds its embodiment in Django.

Table 1 Comparison between MVC and MVT

Model View Controller (MVC)	Model View Template (MVT)
MVC has controller that drives both Model and View.	MVT has Views for receiving HTTP request and returning HTTP response.
View tells how the user data will be presented.	Templates are used in MVT for that purpose.
In MVC, must write all the control specific code.	Controller part is managed by the framework itself.
Highly coupled	Loosely coupled
Modifications are difficult	Modifications are easy
Suitable for development of large applications but not for small applications.	Suitable both small and large applications.
Flow is clearly defined thus easy to understand.	Flow is sometimes harder to understand as compared to MVC.
Does not involve mapping of URLs.	URL pattern mapping takes place.
Examples are ASP.NET MVC, Spring MVC etc.	Django uses MVT pattern.

Traditional SQL databases or relational database management systems (RDBMS) face limitations when dealing with big data and evolving data structures due to schema constraints. The most significant drawback of RDBMS is the requirement for stored values in relational tuplesto be straightforward. These values cannot include any structures, such as nested lists, as they may cause impedance mismatches [9]. Therefore, the development of a framework that facilitates the creation of enhanced knowledge management systems capable of adapting to new demands and leveraging prior knowledge for lifelong learning becomes imperative.

The research aims to create a comprehensive framework that addresses challenges in agricultural knowledge management by leveraging a Web-based Knowledge Management System (WKMS) integrated with a chatbot application, with a specific focus on the agriculture sector.

The research's objective is to design a framework for a Web-Based Knowledge Management System (WKMS) with a primary focus on vegetable-related information. This framework seamlessly incorporates a rule-based chatbot application, trained in Bahasa Malaysia. The system has a dual purpose, catering to two specific user groups: normal users seeking information and admin users responsible for content creation and management. In addition, the research introduces a user-friendly chatbot training platform specially designed for non-technical domain experts (admin users). This platform simplifies the process of chatbot training and establishes a streamlined question-answering system, which is seamlessly extended to normal users through the proposed chatbot application.

This system serves as a digital platform for managing and organizing vegetable-related information. A database or repository included where information is stored, indexed, and easily retrievable. One of the innovations in this research is the introduction of a user-friendly training platform. This platform is designed for non-technical domain experts (admin users) who may not have a technical background. It simplifies the process of training the chatbot, likely by allowing admin users to input questions and corresponding answers, which then become part of the chatbot's knowledge base.

The goal is to create a knowledge management system that can effectively provide answer tousers. Unlike AI-driven chatbots, which use machine learning to generate responses, a rule-basedchatbot relies on predefined rules and patterns to provide answers. This approach can be effective for specific domains like vegetable-related information. The training platform is designed to ensure that the chatbot has a solid understanding of the domain (vegetable-related information) and can provide accurate and relevant responses. The chatbot application should seamlessly integrate into the WKMS, offering a user-friendly interface for both normal and admin users. This integration ensures that users can access the chatbot's knowledge base without friction.

2. LITERATURE REVIEW

Approximately thirty years ago, Knowledge Management Systems (KMS) underwent a transition from being a nascent concept to becoming an integral part of various domains. Over the past decade, KMS has evolved from academic theory to practical necessity, playing a pivotal role in everyday life [10]. The concept of knowledge management involves creating, disseminating, utilizing, and managing information and knowledge, contributing to efficient decision-making and innovation.

Knowledge transfer (KT) and knowledge sharing (KS) are fundamental processes withinthe realm of knowledge management. While often used interchangeably, Paulin and Suneson (2011) clarify that KT and KS represent distinct concepts. KT involves transmitting information from one group to another, analogous to a teacher sharing expertise with students [11]. Conversely, KS refers to the process of information sharing among members of the same group.

These processes play a crucial role in enhancing collaboration and knowledge flow within organizations and sectors. The existence of knowledge barriers (KB) poses challenges in understanding between learners, subjects of study, and the targeted environments. KB can hinder effective cooperation, particularly in cases of generational knowledge gaps or when new members join a group. Knowledge Management Systems (KMS) emerge as essential platforms for bridging these gaps, enabling the sharing of information and facilitating its application, thus enhancing the effectiveness of collaborative endeavours.

Web-based information systems have emerged as powerful tools for efficient information management and professional development. These systems employ architectural strategies to streamline record management and provide users with flexibility, diverse resources, and cost-effective solutions. Initiatives like the Future Ready Schools Initiatives in the United States demonstrate the potential of web-based platforms in offering one-stop resource centers for ongoing professional development [12]. These platforms align with the goal of research in creating a Web-based Knowledge Management System (WKMS) that empowers users with comprehensive, accessible, and up-to-date information.

RDBMS limitations include the inability to store diverse data types and the traditional method of scaling by adding resources becomes impractical due to rising costs and capacity constraints as data expands rapidly. In response to the limitations of RDBMS, NoSQL databases emerged as a viable solution to handle the demands of big data. NoSQL, meaning "Not Only SQL,"represents a departure from the traditional principles of RDBMS and offers more flexible data models, such as document, key-value, column-family, and graph [13].The design philosophy of NoSQL databases is rooted in agility, making them suitable for managing unstructured, semi- structured, and rapidly changing data. Several studies have compared the performance of SQL and NoSQL databases and found that NoSQL databases perform better in operations like data creation, deletion, and reading [14]–[17].

Overall, the literature highlights the importance of knowledge management, knowledgetransfer, sharing, and overcoming barriers, along with the potential of web-based systems for information dissemination and professional growth. These concepts form the basis in crafting a specialized WKMS for agriculture. The emergence of NoSQL databases for handling big data directly aligns with the aim of the study on developing a Web-based Knowledge Management System integrated with a NoSQL database to enhance data management and accessibility.

3. METHODOLOGY

Figure 1 depicts the proposed framework for a Web-based Knowledge Management System (WKMS), which includes two unique user categories: normal user and admin user. Normal user can utilise the chatbot application to do a variety of tasks such as information retrieval, query formulation, response acquisition, and communication with the admin user. Admin user, on the other hand, have the power to execute create, read, update, and delete (CRUD) activities on various content types within both the WKMS and chatbot application.

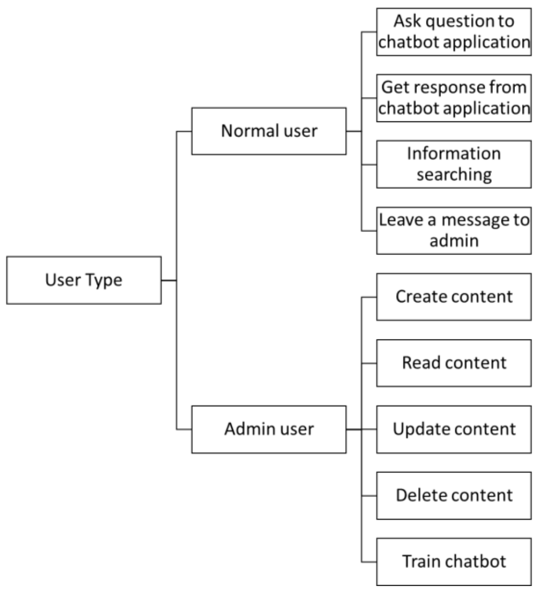


Figure 1. Functional framework structure of proposed framework

A knowledge management system has been built in accordance with the MVT design pattern, as shown in Figure 2. The web page that serves as the front-end user interface was painstakingly created using HTML, CSS, and JavaScript. Both normal and admin users are able toaccess the system by provide specific Uniform Resource Locator (URL). The Python-based business logic layer (Views) activates a designated function to carry out duties related to the request as soon as the system server receives an HTTP request. The data access layer (Models), in turn, allows the business logic layer to access data from the database. The business logic layerreceives an HTTP response from the presentation layer along with an HTML document (Templates).

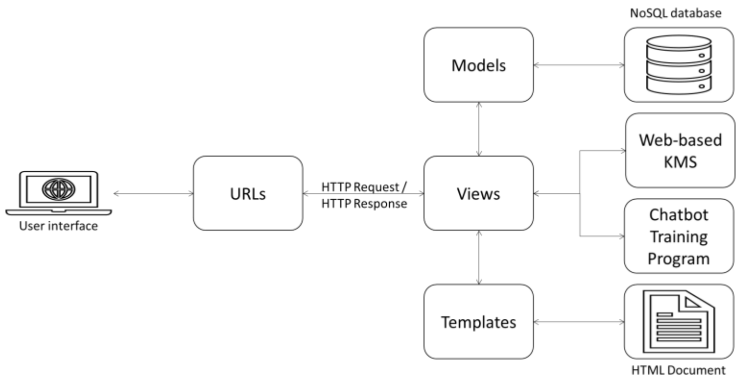


Figure 2. Technical framework structure of proposed framework

3.1 Web-based Knowledge Management System

The transition to web-based systems has revolutionized the way corporations conceive, construct, manage, and safeguard web information. These shifts have ushered in heightened interactivity, personalization, and real-time updates. In conventional web development, the norm entailed crafting static systems that remained unaltered until manual updates were performed. Conversely, within dynamic systems, the development emphasis lies in constructing interactive applications that can be instantaneously updated, achieved through the utilization of JavaScript and server-side scripting languages (such as Python).

A data model is a blueprint for a data pipeline that enables the seamless and efficient transfer of data from source to the destination. As depicted in Figure 3, the envisioned WKMS willadvance through the phases of knowledge chaos, knowledge management, and knowledge integration. The WKMS proposes data models that evolve from a state of knowledge chaos to knowledge management and eventually to knowledge integration. Once the training corpus has been converted into a machine-readable format, the developed system will serve as the foundation for data visualization.

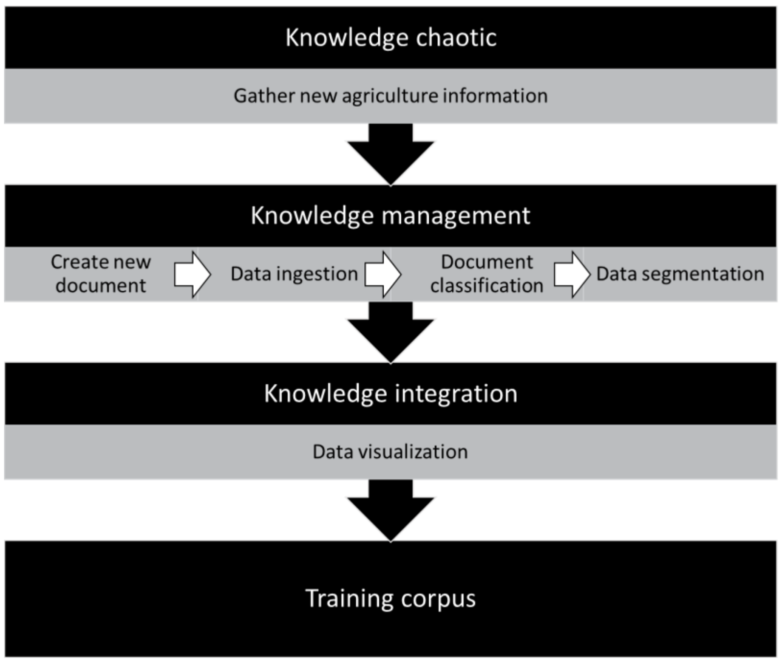


Figure 3. Working principle framework of WKMS

When new data is collected, a fresh document is generated and incorporated into a predefined HTML template. Following this, these documents are systematically organized and stored within the appropriate folder. The segmentation of data stands as a pivotal phase in the data visualization process, achievable through two methods: the Document Object Model (DOM)and HTML parsing. Consequently, the availability of a predefined HTML template within a chatbot training program offers significant advantages for both data visualization and web scraping. Any forthcoming content updates can be seamlessly executed by modifying the HTML document, demanding minimal effort from the developer.

DOM represents a language-neutral interface that empowers programs and scripts todynamically access and update the content, structure, and style of a document. This object model equips JavaScript with all the necessary capabilities to generate dynamic HTML, encompassing the ability to alter HTML elements, attributes, and CSS styles on the page.

HTML parsing entails the processes of tokenization and tree construction. HTML tokens encompass initiation and termination tags, alongside attribute names and values. The parsing procedure is uncomplicated and expedited when the document is well-formed. The parser takes tokenized input and assembles the document tree, effectively mirroring the document's structure.

3.2 Chatbot Training

As depicted in Figure 4, an outline for a chatbot training program has been conceptualized through the amalgamation of diverse approaches and algorithms. Within this program, the initial stage involves utilizing the Django session framework to designate the topic under which the chatbot will undergo training. Much like HTTP cookies, the web server generates concise data packets during Django sessions, expediting the process of information dissemination.

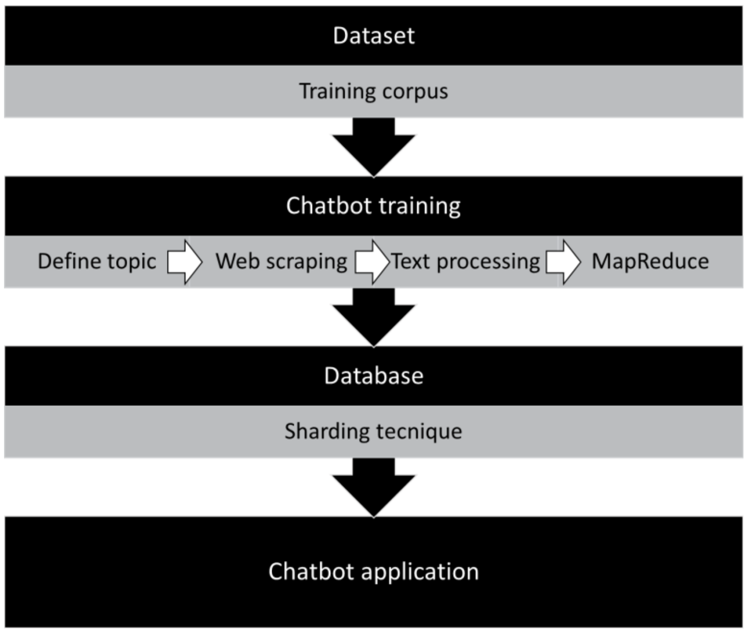


Figure 4. Working principle framework of chatbot training program

The subsequent stage entails the utilization of web scraping techniques to extract data from an HTML document crafted during an earlier phase of web development. This is accomplished by furnishing a designated URL.

A method termed tokenization, derived from text processing, is then employed to eliminate extraneous punctuation and stop words (such as "the," "is," "a," and "an") that lack significant contributions to the communication. The words obtained through tokenization from the original text are subsequently employed to generate a "bag of words" within the program. This "bag of words" constitutes a textual representation of word occurrences within the document.

Furthermore, with the intention of amplifying the efficacy of the chatbot trainer's activities, the program also subjects the original text to sentence-level tokenization. The MapReduce algorithm leverages the JSON file format to segment the dataset into smaller subsets predicated on utterances, intents, and entities. These subsets are subsequently autonomously stored utilizing the sharding technique in dedicated databases associated with the pertinent topics.

4. RESULTS AND DISCUSSION

The depicted architecture of the proposed framework is illustrated in Figure 5. The architecture appears to have a clear separation between user roles and respective access levels. Normal users primarily interact with the frontend and chatbot, while admin users have backend access and control over data and training processes. This kind of architecture is common in systems where there is a need to manage user access, data manipulation, and the improvement of WKMS. Within this architecture, normal users are empowered to interact with the chatbot and explore agriculture-related information using an intuitively designed interface.

In contrast, administrative users possess the ability to input new data into the system and generate a noveltraining corpus for the chatbot application.

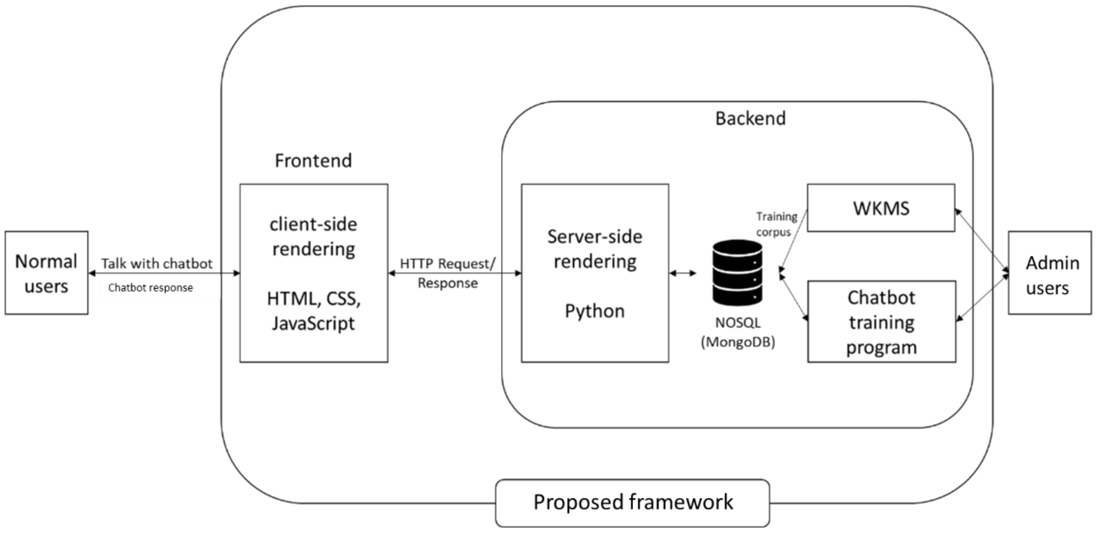
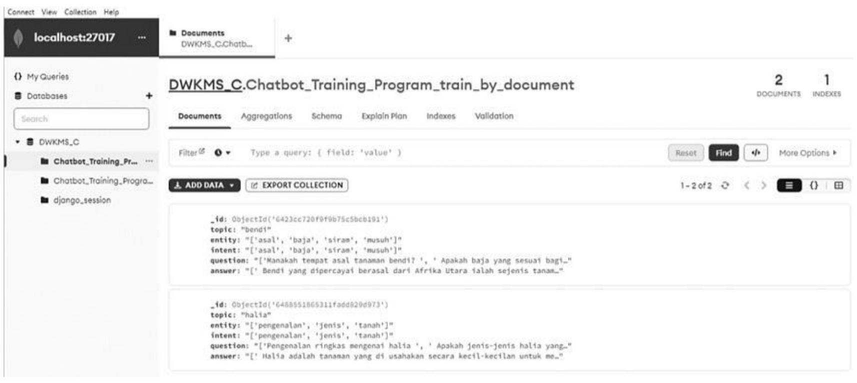


Figure 5. The proposed framework

Front-end development, or client-side rendering, involves coding using a blend of HTML, CSS,and JavaScript. On the other hand, back-end development, or server-side rendering, is accomplished using Python. The proposed WKMS generates a training corpus in a machine- readable format, derived from the data assimilated by system users. This training corpus can be retrieved by users from the database, subsequently serving as input for the chatbot training program, thereby facilitating the chatbot's training process. Training corpus refers to the collection of data used to train the chatbot application. Admin users can generate a new training corpus, which influence how the chatbot learns and improves over time.

The process involves storing the dataset within a document-oriented NoSQL database, whichis designed to handle data in the form of documents, typically arranged in hierarchical structures. Following the data refinement phase, the dataset finds its place within the database, where it is systematically organized into individual documents and further categorized into collections through the application of a sharding technique. To provide a visual representation, consider Figure 6, where an illustrative collection named "Chatbot_Training_Program_train_by_document" encompasses distinct documents labeled "bendi" and "halia". This architectural approach offers a dynamic and adaptable means of storing and managing data, well-suited for scenarios marked by evolving data structures or the need for efficient handling of intricate data relationships.



```
_id: ObjectId('6423cc720f9f9b75c5bcb191')
topic: "bendi"
entity: "['asal', 'baja', 'siram', 'musuh']"
intent: "['asal', 'baja', 'siram', 'musuh']"
question: "['Manakah tempat asal tanaman bendi? ', ' Apakah baja yang sesuai bagi…"
answer: "[' Bendi yang dipercayai berasal dari Afrika Utara ialah sejenis tanam…"

_id: ObjectId('6488551865311fadd829d973')
topic: "halia"
entity: "['pengenalan', 'jenis', 'tanah']"
intent: "['pengenalan', 'jenis', 'tanah']"
question: "['Pengenalan ringkas mengenai halia ', ' Apakah jenis-jenis halia yang…"
answer: "[' Halia adalah tanaman yang di usahakan secara kecil-kecilan untuk me…"
```

Figure 6. Document-oriented NoSQL database

5. CONCLUSION

This research focuses on creating a comprehensive WKMS for the field of agriculture, aiming to address the lack of accessible and accurate information. The WKMS employs a user-friendly design, utilizing HTML, CSS, and JavaScript for the front-end, and Python (Django) with a NoSQL database (MongoDB) for the back-end. Agricultural data undergoes stages of organization to form a machine-readable training corpus. Another objective is the development of a rule-based chatbot that efficiently handles short sessions and specific domain conversations. The chatbot's training involves defining topics, web scraping, text processing, and partitioning for effective data storage.

ACKNOWLEDGEMENTS

This research was funded by a grant from Ministry of Higher Education of Malaysia (FRGS/1/2020/ICT06/UNIMAP/02/1).

REFERENCES

[1] V. Stanojevic, S. Vlajic, M. Milic, and M. Ognjanovic, "Guidelines for framework development process," in *2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR)*, IEEE, Oct. 2011, pp. 1–9. doi: 10.1109/CEE-SECR.2011.6188465.

[2] M. S. F. Soberi, R. Ahmad, M. I. Hussain, and N. Muhammad, "Setup Time Reduction for 5-Axis Composite Material's Trimming Machine by Using a New Framework of SMED Integrated with ECSC Concept," *Journal of Engineering Research and Education*, vol. 9, pp. 29–40, 2017.

[3] R. Li, "Framework Structure Design on College English Assisted Instruction System," *Adv Mat Res*, vol. 846–847, pp. 1776–1779, Nov. 2013, doi: 10.4028/www.scientific.net/AMR.846-847.1776.

[4] Y. M. Li and L. F. Wang, "Design on Framework Structure of College English Learning Management System Based on Struts2," *Adv Mat Res*, vol. 846–847, pp. 1558–1561, Nov. 2013, doi: 10.4028/www.scientific.net/AMR.846-847.1558.

[5] IBM, "What is three-tier architecture?," *IBM*, 2023.

[6] S. I. Ahmad, T. Rana, and A. Maqbool, "A Model-Driven Framework for the Development of MVC-Based (Web) Application," *Arab J Sci Eng*, vol. 47, no. 2, pp. 1733–1747, Feb. 2022, doi: 10.1007/s13369-021-06087-4.

[7] H. Mu and S. Jiang, "Design patterns in software development," in *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, IEEE, Jul. 2011, pp. 322–325. doi: 10.1109/ICSESS.2011.5982228.

[8] K. Zhou, "Research and Implementation of MVC Design Pattern on J2EE Platform," 2022, pp. 1122–1127. doi: 10.1007/978-981-19-4132-0_153.

[9] P. Bhatia, “Big Data and NoSQL,” in *Data Mining and Data Warehousing*, Cambridge University Press, 2019, pp. 442–466. doi: 10.1017/9781108635592.016.

[10] J. Girard and J. Girard, “Defining knowledge management: Toward an applied compendium,” *Online Journal of Applied Knowledge Management*, vol. 3, no. 1, pp. 1–20, 2015.

[11] D. Paulin and K. Suneson, “Knowledge transfer, knowledge sharing and knowledge barriers-three blurry terms in KM,” *Proceedings of the European Conference on Knowledge Management, ECKM*, vol. 2, no. 1, pp. 752–760, 2011.

[12] D. Hu, B. Yuan, J. Luo, and M. Wang, “A review of empirical research on ICT applications in teacher professional development and teaching practice,” *Knowledge Management and E-Learning*, vol. 13, no. 1, pp. 1–20, Mar. 2021, doi: 10.34105/j.kmel.2021.13.001.

[13] E. Khashan, A. Eldesouky, and S. Elghamrawy, “An adaptive spark-based framework for querying large-scale NoSQL and relational databases,” *PLoS One*, vol. 16, no. 8 August, Aug. 2021, doi: 10.1371/journal.pone.0255562.

[14] R. Aghi, S. Mehta, R. Chauhan, S. Chaudhary, and N. Bohra, “A comprehensive comparison of SQL and MongoDB databases,” *International Journal of Scientific and Research Publications*, vol. 5, no. 1, 2015.

[15] D. Damodaran B, S. Salim, and S. M. Vargese, “Performance Evaluation of MySQL and MongoDB Databases,” *International Journal on Cybernetics & Informatics*, vol. 5, no. 2, 2016, doi: 10.5121/ijci.2016.5241.

[16] C. Gyorodi, R. Gyorodi, G. Pecherle, and A. Olah, “A comparative study: MongoDB vs. MySQL,” in *2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015*, 2015. doi: 10.1109/EMES.2015.7158433.

[17] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, “A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages,” in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, 2017. doi: 10.1109/I-SMAC.2017.8058365.