DATA ANALYTICS AND APPLIED MATHEMATICS
E-ISSN: 2773-4854
VOLUME 7, ISSUE 1, 2026, 31 – 38
DOI: https://doi.org/10.15282/daam.v7i1.13380

RESEARCH ARTICLE

Development of a retrieval-augmented generation framework for automated proposal generation through Azure cloud

Aqilah Maisarah Azizi[1], Nor Azuana Ramli[1]*, Mohd Zaid Waqiyuddin Mohd Zulkifli[2]

[1]Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia
[2]Credence, 1 Jalan Damansara, Damansara Kim, 60000 W.P. Kuala Lumpur, Malaysia

Abstract - Artificial intelligence (AI) has been increasingly adopted to improve knowledge retrieval and decision-making in enterprise environments; however, proposal preparation in the pre-sales stage still relies heavily on manual searches across fragmented document repositories and heterogeneous file formats. This study addresses this gap by proposing a retrieval-augmented generation (RAG)-based system that automates the retrieval, summarization, and generation of proposal-related information from both internal repositories, such as OneDrive and SharePoint, and external web sources. The proposed system integrates Azure Blob Storage, Azure AI Search, Azure OpenAI, and Bing Search within a RAG framework, supported by a web-based interface developed using React/Next.js. Unlike conventional keyword-based search tools, the system interprets user intent and delivers consolidated, relevant information to support proposal drafting. Experimental evaluation in a pre-sales use case demonstrates a reduction in manual information retrieval effort and improved content relevance, while achieving an average response generation time of 102 milliseconds, enabling real-time interaction. Overall, the findings demonstrate how secure, enterprise-grade cloud integration and RAG-based conversational systems can transform pre-sales workflows by allowing professionals to shift their focus from manual information gathering to higher-value strategic content development.

Article History
Received : 3 November 2025
Revised : 5 February 2026
Accepted : 19 February 2026
Published : 31 March 2026

Keywords
Artificial intelligence
Proposal generation
Natural language processing
Retrieval-augmented generation
GPT-4o

1. Introduction

Artificial intelligence (AI) is reshaping enterprise operations by enabling automation, real-time analytics, and contextual decision-making. In large corporations that operate using a business-to-business (B2B) model, proposals are a key medium of communication. Proposal development is essential for establishing and maintaining professional partnerships, conducting contract negotiations, collaborating on projects, and addressing issues [1]. In the pre-sales stage, it is necessary to prepare high-quality proposals that align technical capabilities with client needs. However, current practices require the manual retrieval of information from distributed repositories, such as OneDrive and SharePoint, often in heterogeneous formats, including Word, PowerPoint, PDF, and Excel. This leads to prolonged preparation time, inconsistent proposal quality, and a higher risk of outdated or inaccurate information being included. Addressing these challenges, human intervention can be reduced through automated solutions that can identify and utilize data across multiple systems efficiently [2]. From the perspective of current technological advancements, AI-based modelling is recognized as a key driver of automated, intelligent systems, shaping the future of businesses by improving, accelerating, and enhancing the accuracy of processes [3]. This study proposes an AI-powered chatbot system that automates the extraction and retrieval of proposal-related information. The solution leverages Large Language Models (LLMs), Natural Language Processing (NLP), and Generative Pre-Trained Transformers (GPT) to assist users during proposal generation. It integrates Azure OpenAI for language understanding, Azure Blob Storage for secure data storage, Azure AI Search for semantic retrieval, Vercel AI for orchestration, and Bing Search for external knowledge supplementation. Front-end frameworks such as Next.js and React.js have gained significant popularity for developing modern applications [4], particularly for delivering high-performance, interactive, and dynamic chatbot interfaces [5]. Additionally, studies have shown that GPT-4o, when integrated with the Next.js App Router, can be used to develop highly effective, context-aware conversational chatbot systems [6]. For storing large volumes of unstructured data, Microsoft Azure Blob Storage is regarded as a suitable hybrid cloud solution due to its cost-effectiveness, security, and scalability [7].

Furthermore, Azure AI Search plays a crucial role in enhancing data retrieval performance, particularly through the integration of AI and NLP, which improves search quality and speed [8]. At the core of orchestrating these components is the Vercel AI SDK, an open-source library that enables conversational and streaming interactions within JavaScript and TypeScript applications [9]. Meanwhile, Azure OpenAI embedding models can be used to perform searches and query a knowledge base for the most relevant documents [10]. To complement internal data gaps, Bing Search, which currently holds a substantial portion of the global search engine market share [11], provides real-time external insights. Through the Bing Search API, developers can retrieve structured search results and integrate them directly into applications [12], making it an effective tool for enriching proposal generation with up-to-date information. Rather than treating proposal preparation as a conventional document search task, this study positions it as a knowledge-intensive pre-sales workflow that requires intent-aware retrieval and content synthesis across heterogeneous data sources. To address the limitations of existing keyword-based and repository-specific tools, this research proposes an end-to-end

***CORRESPONDING AUTHOR | Nor Azuana Ramli** | ✉ azuana@umpsa.edu.my

retrieval-augmented generation (RAG)-based conversational system that integrates enterprise documents and external knowledge sources within a unified interface. The significance of this work lies in demonstrating how a RAG-based approach can be operationalized in a real-world corporate setting to reduce manual information handling and support more efficient proposal development.

This study aims to identify suitable AI-driven models and cloud components to ensure accurate and efficient information extraction and retrieval within a RAG-based pre-sales proposal generation framework. Building upon this foundation, it seeks to design and develop a conversational system that integrates internal enterprise repositories with external search capabilities, thereby automating proposal-related information retrieval and content synthesis. Finally, the research involves implementing a responsive, interactive web-based interface using React or Next.js to facilitate real-time user interaction and seamless proposal generation. This paper is structured as follows: Section 2 reviews related work on enterprise search, conversational AI, and retrieval-augmented generation; Section 3 presents the methodology, including data processing and system implementation; Section 4 discusses the experimental results and analysis; and Section 5 concludes the paper with insights and directions for future research.

2. Literature Review

Artificial intelligence and machine learning have introduced new approaches to organizational knowledge management, particularly in tasks involving text classification, summarization, and content generation [13]. The development of generative AI and large language models has pushed natural language processing to higher levels of semantic understanding [14]. These advancements enable organizations to handle complex proposal documents using automated systems rather than manual review. Despite the advancements in NLP, LLMs, GPT models, cloud-based storage, augmentation and generation systems, and conversational UI frameworks, existing research and implementations have not focused on applying these technologies specifically to the pre-sales proposal development process. Current solutions either address general conversational assistance or generic document retrieval, without supporting the domain-specific, lengthy, and frequently updated proposal documents used. Additionally, the integration of internal organizational knowledge and external, real-time information sources remains limited in prior work. Therefore, this study addresses these gaps by integrating GPT-4o, Azure Blob Storage, Azure AI Search, Vercel AI SDK, React/Next.js, and Bing Search into a unified AI-powered chatbot system tailored to automate information extraction and retrieval during proposal generation. Table 1 summarizes previous studies that address the research gap in this study. Retrieval-Augmented Generation is an architectural paradigm that integrates information retrieval mechanisms with large language models to generate responses based on external knowledge sources. Rather than relying solely on static knowledge acquired during model pre-training, RAG systems retrieve relevant documents at query time and feed them into the generation process, enabling more accurate, context-sensitive, and up-to-date outputs. Previous studies have highlighted that this retrieval-generation coupling effectively mitigates common limitations of stand-alone LLMs, particularly hallucinations and stale responses, by generating responses adaptively to the evidence obtained rather than probabilistic inference alone [31].

The RAG paradigm is well suited for domain-specific and knowledge-intensive environments where information is distributed across internal repositories and is subject to frequent updates. Applied research shows that RAG improves reliability and efficiency in closed domain settings by enabling contextual intent retrieval and synthesis from organization-specific documents, reducing reliance on manual search and generic conversational assistance [32]. As a result, RAG provides a solid theoretical foundation for systems that aim to automate enterprise information extraction and content synthesis, particularly in workflows such as pre-sales proposal development that require accurate, timely domain-based responses.

Table 1. Summary of literature review and identified research gaps

Related Study	Focus Area / Key Findings	Research Gap	How the Current Study Addresses the Gap
Olutimehin et al. [15]	AI integration in business for efficiency and innovation	Lacks practical NLP application in proposal generation or retrieval systems	Applies GPT-4o and NLP for real-world proposal generation and automated retrieval
Huang [16]	Human-AI collaboration for coaching	Focuses on mentoring, not document automation	Uses adaptive LLM for contextual proposal generation
Lin and Jie ([17]	GPT-3.5 Turbo for translation and summarization	Limited to translation without UI or multi-source data integration	Integrates GPT-4o for retrieval, summarization, and generation in proposal context
George et al. [18]	GPT-4 for corporate communication	No integration with document storage or external search	Combines GPT-4o with Azure AI Search, Bing Search, and Blob Storage
Roumeliotis et al. [19]	LLMs for sentiment analysis in tourism	Focuses on sentiment tasks, not retrieval-based generation	Leverages GPT-4o for context-based proposal information generation
Agarwal and Prasad [20]	Azure Blob storage benchmarking	No AI or retrieval integration	Integrates Blob storage for secure data management and retrieval
Gnanasekaran et al. [21]	Azure Cognitive Search for data retrieval	Does not integrate generative AI	Combines Azure AI Search with GPT-4o for semantic retrieval and generation
Danushka [22]	LLM evaluation for historical text	Focuses on historical data, not proposal automation	Adapts GPT-4o for contextual proposal generation and retrieval

Table 1. continued

Related Study	Focus Area / Key Findings	Research Gap	How the Current Study Addresses the Gap
Sekhar Emmanni [23]	Full-stack freelance application	No AI or cloud-native scalability	Implements React/Next.js UI with Azure Functions for scalability
Harrison Oke Ekpobimi [24]	SPA framework comparison	Frontend-focused, not AI-integrated	Combines React/Next.js with AI for an interactive proposal system
Rathore et al. [25]	Next.js for high-performance web apps	No AI or retrieval system integration	Uses Next.js to build an intelligent AI-driven chatbot interface
Mohan et al. [26]	Next.js-based delivery management	Focused on the logistics domain	Extends Next.js for intelligent document workflow automation
Gangi Reddy et al. [27]	Collaborative search with summarization	No generative or Azure integration	Integrates Bing Search and GPT-4o for enhanced proposal insights
Patil [28]	Chatbot with Bing Search	Conversational only, lacks proposal generation	Integrates Bing Search with GPT-4o for proposal data retrieval and insights
Jovanić and Čarapina [29]	Real-time inventory using Vercel AI SDK	Focuses on stock management only	Uses Vercel AI SDK for real-time proposal chatbot generation
Workorb Blog [30]	AI in web content creation	Limited to content generation, no data retrieval integration	Combines Vercel AI SDK with Azure AI Search and Blob Storage

3. Methodology

The framework consists of tools designed to ease and automate information retrieval during proposal generation. It utilizes cloud-based services and artificial intelligence technologies to facilitate optimized data search and document access, while also driving AI-based insights. The sequence diagram in Figure 1 illustrates the end-to-end interaction between users and the proposed AI-powered solution development tool, designed to support the pre-sales process. This could help users streamline the extraction and retrieval of the relevant information through a conversational interface. The process begins when the users upload documents, such as technical specifications, templates, and brochures, which are stored securely in Azure Blob Storage, forming the system's internal data ingestion and knowledge base. Compared to traditional file management systems, this framework can transform raw documents into semantically searchable data by integrating with intuitive components.

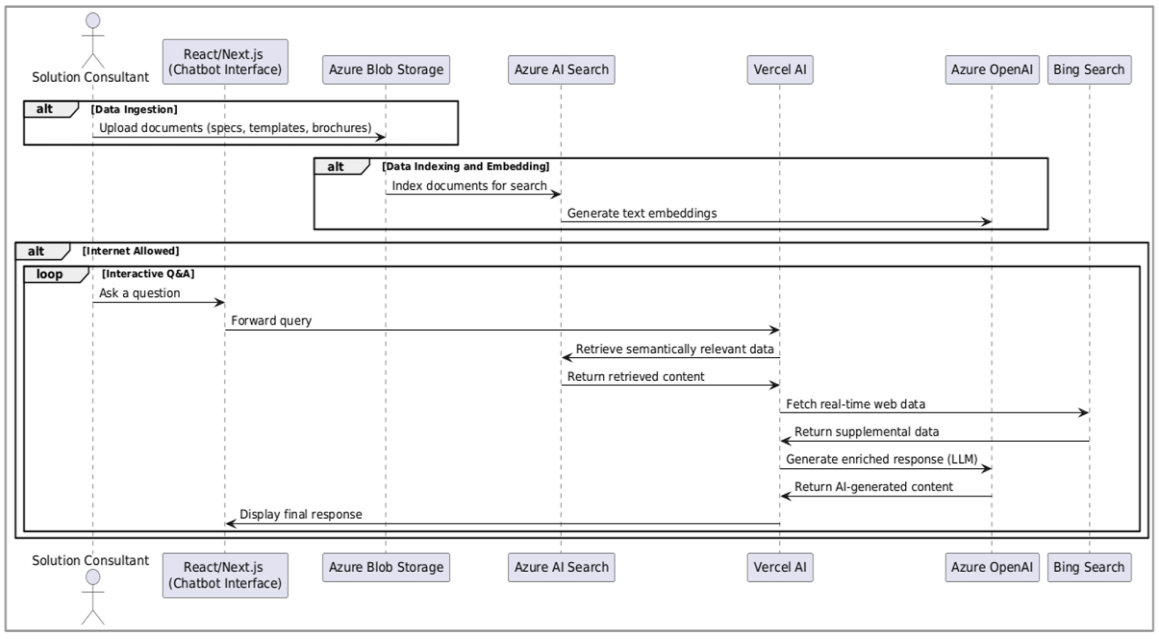

Figure 1. Sequence diagram of the AI-powered proposal generation workflow

To enable the content to be queryable, Azure AI Search then indexes and embeds the uploaded document. During this phase, Azure OpenAI's embedding model is utilized to convert the text into numerical vectors, enabling Azure AI Search to perform semantic matching rather than simple keyword lookup. Azure AI Search stores these embedded vectors, allowing users to receive context-aware and intelligent responses to their queries. After the data ingestion and vectorization process, the question-and-answer phase is also included in the framework. This interactive question-and-answer phase between users and the chatbot UI occurred on the front end using React and Next.js tools. React/Next.js is not utilized for UI; instead, it is integrated with the back end using Vercel AI to orchestrate the system workflow. This integration between the front-end and back-end enables users to chat on the UI and retrieve relevant and accurate answers from AI-driven tools that obtain responses from internal and external web-based data, thereby enhancing the chatbot's effectiveness.

For example, when a user queries the UI front-end, the query is sent to Vercel AI via OpenAI. Vercel AI calls Azure AI Search to gather the internal query-related information and then sends it back to Vercel AI. Azure AI Search returns embedded and indexed information that is stored in it. When the information is insufficient or outdated, Vercel AI will retrieve external query-related data from Bing Search to enrich the responses and access real-time web data. The responses are sent back to Azure OpenAI for advanced natural language generation. After uniting the information from internal and external sources, an AI-generated response is compiled. This extract is then presented back to the chatbot interface via Vercel AI. Overall, this framework demonstrates an intelligent and automated process that assists users in extracting essential information to generate proposals efficiently and effectively. The integration of multiple AI-driven tools enables the chatbot system to not only retrieve relevant data but also provide enriched, accurate, up-to-date, and contextually relevant content to support decisions made by users in pre-sales activities.

3.1 Data Collection

The study utilized company-supplied internal documentation. The data consisted only of internal documentation typically used during pre-sales activities. The data provided includes PowerPoint files, for example, containing service trial details and proposal content. Next, Excel files, for example, contain those technical configurations and schedules. Lastly, the CSV files contain structured operational data.

3.2 Data Preparation

This is a crucial phase of the data preparation process that can directly impact the performance and accuracy of the AI-driven proposal generation tool. All relevant data would be uploaded, ingested, embedded, and indexed appropriately to enable a seamless querying and data retrieval experience during the proposal generation process. Since Azure's system automates this process, it has eliminated the traditional processing step, making the system more effective.

3.3 Data Ingestion

The initial step, creating an Azure Blob Storage account, was done in a designated resource group, which served as the primary point for data ingestion. Five documents related to proposals, like past proposals, templates, and brochures, were uploaded in formats such as PowerPoint, Excel, and CSV. Once data has been stored, it has been ingested into the allocated container, which is well-structured and easily accessible for subsequent processes, such as indexing and embedding, to prepare the data for efficient retrieval during proposal generation.

3.4 Data Embedding

To enable semantic search and context-aware responses, each document chunk underwent vector embedding using the Azure OpenAI Embedding API. The embedding process converted textual content into high-dimensional numerical vectors representing meaning rather than keywords. These vectors were stored in Azure AI Search vector indexes, supporting similarity-based matching between user queries and document content. By using embedding models derived from GPT-4o, the system achieved a robust understanding of technical terminology and cross-document context.

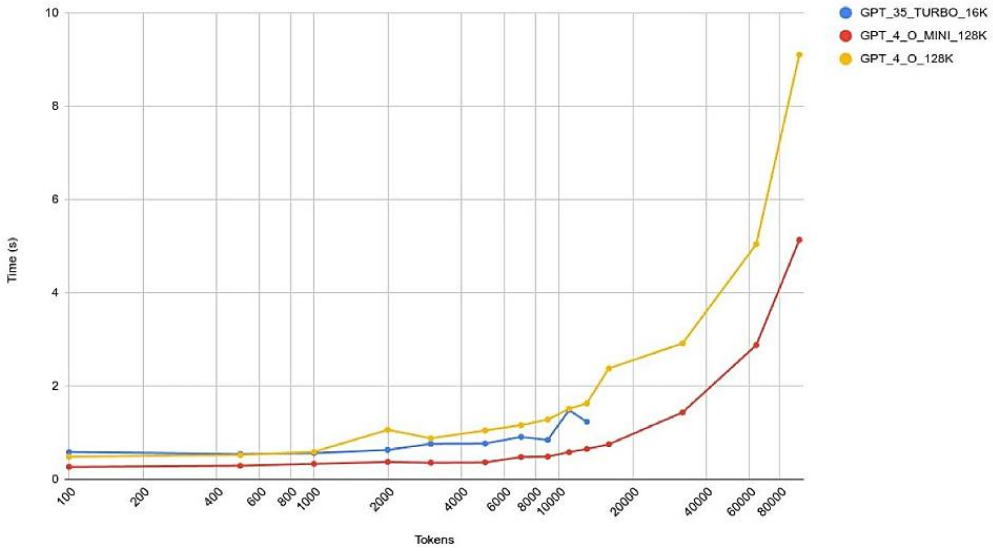

Figure 2. API response times compared with GPT-3.5 turbo, GPT-4o mini and GPT 4o [31]

3.5 Data Indexing

Following embedding, data indexing structured the information into retrievable clusters. Indexes included metadata such as document title, author, source type, and timestamp. Each index entry was linked to its corresponding embedding vector to support hybrid (keyword + semantic) queries. Azure AI Search automatically refreshes indexes when documents are

updated in Blob Storage. This ensured the chatbot could consistently retrieve the latest validated content, reduce manual curation, and ensure information accuracy.

3.6 Implementation

After reviewing multiple transformer architectures, GPT-4o was selected as the optimal model due to its multimodal capability, high context window, and efficiency in corporate environments. Compared to GPT-3.5 Turbo, GPT-4o provided faster inference and improved comprehension of mixed technical and natural-language queries, also supporting multilingual interaction. The model's integration was fine-tuned through Azure OpenAI endpoints. Figure 2 shows the API response times compared with GPT-3.5 turbo, GPT-4o mini and GPT 4o.

The React/Next.js framework was adopted to develop the chatbot interface due to its modularity, performance, and server-side rendering capabilities. The interface comprises chat input, response display, and message-stream components linked to Vercel AI's API routes. React hooks manage state synchronization, while Tailwind CSS ensures responsive design. Next.js supports server-side rendering, which enables faster load times and enhances SEO, making the solution scalable for enterprise use. The project integrated several AI-driven tools:

i) Azure Blob Storage: central data repository enabling secure storage and fast retrieval.

ii) Azure AI Search: vector-based semantic retrieval engine.

iii) Azure OpenAI: embeddings model and generative responses.

iv) Vercel AI SDK: orchestrator connecting front-end and back-end processes.

v) Bing Search API: supplements internal data with verified external information and real-time industry insights.

Together, these tools form a unified ecosystem for intelligent information extraction and retrieval in proposal development.

3.7 Deployment

A RAG-based proposal-generation system was deployed to users via a web application with a custom domain. The front-end of the web application was built with React and Next.js, enabling users to experience a responsive, dynamic interface for information extraction and retrieval during the proposal generation process. Cloud infrastructure was used to deploy this chatbot, ensuring scalability, availability, and secure data management. Then, for security concerns, users have secure access to the system, which only authorized users can access sensitive proposal data and interact with the AI-driven tools. They are required to log in through Azure Entra before accessing the chatbot. The system's cloud-based architecture not only ensures it can meet users' needs by interacting with multiple tools simultaneously but also maintains data security and performance.

4. Results and Discussion

The implementation phase transformed the research framework into a functional RAG-based system for pre-sales proposal support. The system was designed to automate information retrieval, augmentation, and generation by integrating a domain-specific AI model, a responsive front-end interface, and cloud-based services. GPT-4o was selected as the core AI model due to its advanced natural language understanding and generation capabilities. Internal proposal documents are securely stored in Azure Blob Storage and indexed using Azure AI Search, which generates semantic embeddings for intent-aware retrieval. External knowledge is accessed through Bing Search to complement internal data, ensuring up-to-date information. The system follows a retrieval-augmentation-generation pipeline. In the retrieval stage, relevant internal and external documents are identified based on semantic similarity to the user query. During augmentation, retrieved documents are combined with the query to form a context-enriched prompt. Finally, GPT-4o generates coherent outputs tailored for proposal drafting. The entire workflow is orchestrated via the Vercel AI SDK, which facilitates seamless communication between the front-end and backend services.

The front-end interface, developed with React and Next.js, enables conversational interaction and real-time query handling. Azure Entra authentication ensures secure, role-based access, while audit trails maintain compliance with enterprise data governance standards. The system architecture supports scalability, modularity, and secure deployment as a centralized web application, providing a robust platform for automating the retrieval and synthesis of proposal-related information. Upon completion of the implementation phase, the RAG-based system was fully deployed as a secure web application, enabling pre-sales professionals to interact with internal and external data sources via a responsive conversational interface. With the system architecture and retrieval-augmentation-generation workflow established, its performance, effectiveness, and impact on the proposal generation process were evaluated in a real-world use case.

The deployed system successfully integrates internal enterprise documents with external web-based knowledge, providing a comprehensive platform for automated proposal support. User queries submitted through the React/Next.js interface is processed in real time by GPT-4o, which synthesizes retrieved content into coherent, contextually relevant outputs. Azure Entra authentication ensures secure access, while role-based permissions and audit trails maintain enterprise compliance. Unlike general-purpose chatbots, which rely solely on public datasets, the proposed system combines internal vector databases with external retrieval, ensuring both accuracy and currency of information in a domain-specific, enterprise-ready environment.

Performance evaluation demonstrates an average response generation time of 102 milliseconds, as shown in Figure 3, confirming the low-latency capability of the RAG pipeline and supporting real-time interaction suitable for dynamic pre-

sales workflows. The semantic embeddings generated through Azure AI Search enable intent-aware retrieval of internal documents, while augmentation incorporates external information to enrich query context before generation by GPT-4o. This workflow ensures that outputs are accurate, relevant, and structured for direct inclusion in proposal drafts, significantly reducing the manual effort and inconsistency associated with conventional document searches.

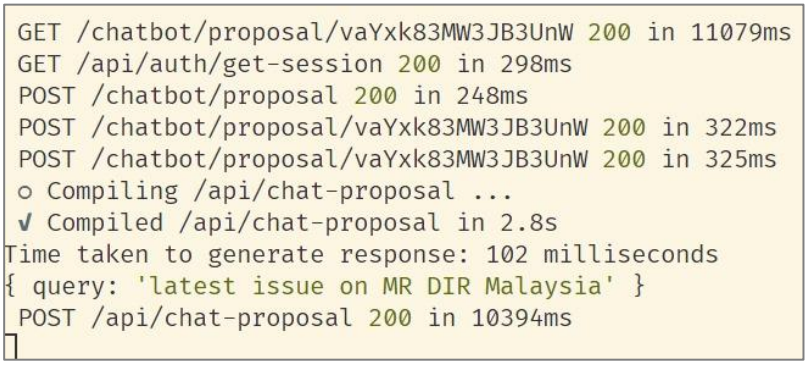

Figure 3. Terminal output while the query is processed in real time

By consolidating heterogeneous sources and automating information synthesis, the system allows pre-sales professionals to focus on strategic content development, rather than repetitive data gathering. The hybrid architecture, combining retrieval, augmentation, and generation, represents a novel contribution to enterprise knowledge management, delivering a secure, scalable, and context-aware solution tailored for proposal preparation. Overall, the results demonstrate that integrating cloud-based AI services within an RAG framework can significantly enhance efficiency, accuracy, and productivity in pre-sales operations, providing a practical and innovative tool for corporate decision-making.

5. Conclusions

This study successfully developed a retrieval-augmented generation-based system to automate the retrieval, augmentation, and generation of proposal-related information in the pre-sales process. The system integrates multiple Azure cloud services, including Azure Blob Storage for document storage, Azure AI Search for semantic retrieval, Azure OpenAI for content generation, and Bing Search for external knowledge enrichment, enabling efficient access to both internal and external data sources. A web-based conversational interface built with React/Next.js provides a responsive platform for users to submit queries and receive real-time, context-aware responses.

A key contribution of this work lies in the augmentation stage, where retrieved documents are dynamically combined with user queries to construct enriched prompts that preserve contextual relevance before response generation. The generation stage, powered by GPT-4o, synthesizes the augmented information into coherent and task-specific outputs, supporting proposal drafting rather than simple document retrieval. The adoption of the Vercel AI SDK further facilitates real-time communication between the front-end and back-end, ensuring low-latency interaction and efficient data processing. Overall, the proposed RAG-based system streamlines proposal generation by reducing manual information handling and enabling users to focus on strategic content creation, demonstrating the effectiveness of combining cloud-based AI services with modern web technologies in enterprise pre-sales workflows.

Future research will focus on extending the proposed system through automatic scaling and event-driven processing by integrating Azure Functions, thereby enabling greater flexibility, resilience, and resource efficiency across varying user workloads. In addition, multimodal data handling will be explored to enable the chatbot to process and reason over images, tables, and graphical content alongside textual data, thereby supporting richer, more comprehensive proposal generation. These enhancements are expected to improve system scalability and broaden its applicability to more complex enterprise documents and real-world pre-sales scenarios.

Acknowledgement

The authors would like to express their gratitude to Universiti Malaysia Pahang Al-Sultan Abdulah (UMPSA) for the support/ facilities.

Funding

This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sectors.

Declaration of Competing Interest

The authors declare no conflict of interest.

CRediT Authorship Contribution Statement

Aqilah Maisarah Azizi (Formal analysis; Visualization; Methodology; Writing - original draft; Resources)
Nor Azuana Ramli (Conceptualisation; Writing - review & editing)
Mohd Zaid Waqiyuddin Mohd Zulkifli (Supervision; Methodology; Validation; Data curation)

Availability of the Data and Materials

The data used to support the findings of this study are included within the article.

Ethical Declarations

No artificial intelligence tools were used in the preparation of this manuscript. All content was developed manually by the authors. This study did not involve human participants or animals. Ethical approval was therefore not required.

Generative Artificial Intelligence Declarations

The authors claim that artificially intelligent-assisted technologies in the form of generative AI were not used to generate content, ideas, or theories. We have just utilized AI to enhance readability and refine the language. This was used with extreme human control and oversight. The authors take full responsibility for reviewing and approving the content.

References

[1] Pinky. Understanding Business-to-Business (B2B) business model [Internet]. NEXEA; 2023 Feb 22 [cited 2025 Jan 20]. Available from: https://www.nexea.co/understanding-business-to-business-b2b-business-model/

[2] Alshehri I, Alshehri A, Almalki A, Bamardouf M, Akbar A. BreachSeek: a multi-agent automated penetration tester [Internet]. arXiv [Preprint]. 2024 Aug 31 [cited 2025 Jan 20]. Available from: http://arxiv.org/abs/2409.03789

[3] Sarker IH. AI-based modeling: techniques, applications, and research issues towards automation, intelligent and smart systems. SN Computer Science. 2022;3(2):158.

[4] MedCode. Next.js & React.js: useful blogs & articles [Internet]. DEV Community; 2024 Jan 14 [cited 2025 Jan 20]. Available from: https://dev.to/med_code/nextjs-reactjs-useful-blogs-articles-4489

[5] Hasan MM. Build an AI chatbot frontend with React, Next.js, and FastAPI powered by Ollama & DeepSeek-R1 [Internet]. Medium; 2025 Mar 3 [cited 2026 Mar 25]. Available from: https://medium.com/@rabbi.cse.sust.bd/build-an-ai-chatbot-frontend-with-react-next-js-and-fastapi-powered-by-ollama-deepseek-r1-9a7adc600804

[6] Trivedi L. Building your custom chatbot with OpenAI GPT-4.0 and Next.js (App Router) [Internet]. Medium; 2025 Jan 16 [cited 2026 Mar 25]. Available from: https://medium.com/zestgeek/building-your-custom-chatbot-with-openai-gpt-4-0-and-next-js-app-router-909914e97125

[7] Rootstack. Benefits of Microsoft Azure Storage Explorer for business [Internet]. Rootstack; [date unknown] [cited 2026 Mar 25]. Available from: https://rootstack.com/en/blog/benefits-microsoft-azure-storage-explorer-business

[8] Parwani K, Das S, Mittal S, Raj R. Enhancing information retrieval in Azure Cognitive Search. Journal of Analysis and Computation (JAC). 2023;11(3):22-30.

[9] Palmer J, Ding S, Leiter M. Introducing the Vercel AI SDK [Internet]. Vercel Blog; 2023 Jun 15 [cited 2026 Mar 25]. Available from: https://vercel.com/blog/introducing-the-vercel-ai-sdk

[10] Mrbullwinkle. What is Azure OpenAI in Azure AI Foundry models? [Internet]. Microsoft Learn; 2025 Jan 20 [cited 2026 Mar 25]. Available from: https://learn.microsoft.com/en-us/azure/ai-services/openai/overview#comparing-azure-openai-and-openai

[11] EBSCO Information Services. Bing (search engine) | EBSCO [Internet]. EBSCO; 2025 [cited 2026 Mar 25]. Available from: https://www.ebsco.com/research-starters/computer-science/bing-search-engine

[12] Shinde M. A guide to using the Bing Search API: retrieving results with an API key [Internet]. Medium; 2024 Mar 31 [cited 2026 Mar 25]. Available from: https://mayurashinde.medium.com/a-guide-to-using-the-bing-search-api-retrieving-results-with-an-api-key-398ec3a1a0fc

[13] Andrade FA. Azure machine-learning service and AI-driven application for content management [Internet]. 2024 [cited 2026 Mar 25].

[14] Hagos DH, Battle R, Rawat DB. Recent advances in generative AI and large language models: Current status, challenges, and perspectives. IEEE Transactions on Artificial Intelligence. 2024.

[15] Olutimehin OA, Ajayi TO, Adebayo HA, Olanrewaju I. Leveraging generative AI for proposal automation. International Journal of Artificial Intelligence Research. 2024;14(2):133-145.

[16] Huang EJ. Human-AI collaboration for entrepreneurship coaching [Internet]. 2024 [cited 2026 Mar 25].

[17] Lin X, Jie L. Integration of AI search and cloud platforms for corporate applications. Cloud Computing Journal. 2024;15(1):76-84.

[18] George D, George A, Martin ASG. Revolutionizing business communication: exploring the potential of GPT-4 in corporate settings. Partners Universal International Research Journal. 2023.

[19] Roumeliotis KI, Tselikas ND, Nasiopoulos DK. Leveraging large language models in tourism: a comparative study of the latest GPT Omni models and BERT NLP for customer review classification and sentiment analysis. Information. 2024;15(12):792.

[20] Agarwal D, Prasad SK. AzureBench: benchmarking the storage services of the Azure cloud platform. In: Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops (IPDPSW); 2012 May 21-25; Shanghai, China. New York: IEEE; 2012. p. 1048-1057.

[21] Gnanasekaran T, Prakash K, Babu P, Kumar S. AI-integrated cloud systems for document intelligence. IEEE Access. 2024;12:23345-23358.

[22] Danushka FA. Design a freelance application evaluating a full stack for scalable and user-centric development [Internet]. 2025 [cited 2026 Mar 25].

[23] Sekhar Emmanni P. Comparative analysis of Angular, React, and Vue.js in single page application development. International Journal of Science and Research (IJSR). 2023;12(6):2971-2974.

[24] Harrison OE. Building high-performance web applications with NextJS. Computer Science & IT Research Journal. 2024;5(8):1963-1977.

[25] Rathore SPS, Kaushik P, Poonia M, Sikarwar SS, Singh D, Jain D. Ease delivery: a next-gen delivery management solution. In: 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI); 2024 Mar 14-16; Gwalior, India. New York: IEEE; 2024. p. 1-6.

[26] Mohan V, Sivaganesan D, Balasubramaniam S, Vikhas SG, Dhinesh SP. Collaborative search with knowledge sharing and summarization. In: 4th International Conference on Sustainable Expert Systems (ICSES); 2024 Sep 26-27; Coimbatore, India. New York: Springer; 2024. p. 582-590.

[27] Gangi Reddy R, Chandra S, Bai H, Yao W, Sidhu M, et al. CharmBana: progressive responses with real-time internet search for knowledge-powered conversations. In: WSDM '24: Proceedings of the 17th ACM International Conference on Web Search and Data Mining; 2024 Mar 4-8; Merida, Mexico. New York: ACM; 2024. p. 1050-1053.

[28] Patil O. Real time inventory management system powered by generative user interface. International Journal of Scientific Research in Engineering and Management. 2024;8(4):1-5.

[29] Jovanić M, Čarapina M. Application of artificial intelligence in the creation of web content. In: 2024 47th ICT and Electronics Convention (MIPRO); 2024 May 20-24; Opatija, Croatia. New York: IEEE; 2024. p. 2063-2068.

[30] Workorb Blog. Comparing latency of GPT-4o vs. GPT-4o mini [Internet]. Workorb; 2024 Aug 29 [cited 2026 Mar 25]. Available from: https://www.workorb.com/blog/comparing-latency-of-gpt-4o-vs-gpt-4o-mini

[31] Swacha J, Gracel M. Retrieval-augmented generation (RAG) chatbots for education: a survey of applications. Applied Sciences. 2025;15(8):4234.

[32] Aurora C, Mauritsius T. Creation of RAG chatbot in answering queries related to banking terms using Microsoft Azure. International Journal of Cyber and IT Service Management. 2025;5(2):144-155.