Journal of Soft Computing Exploration
Homepage: htttps://shmpublisher.com/index.php/joscex
p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Implementation of digital human avatar virtual assistant with augmented generation retrieval technology in interactive systems for nutrition education

Genta Swarawisesa Erliarto Putra[1], Adang Suhendra[2], Achmad Benny Mutiara Q.N.[3], Asep Juarna[4]
[1,3] Department of Information Systems, Universitas Gunadarma, Indonesia
[2,4]Department of Informatics, Universitas Gunadarma, Indonesia

Article Info

Article history:

Received September 8, 2025
Revised September 24, 2025
Accepted November 20, 2025

Keywords:

Digital human
Retrieval augmented Generation
GiziAI
Chatbot
Website

ABSTRACT

In today's digital era, artificial intelligence (AI) based chatbots utilizing Large Language Models (LLM) have become a promising innovation for nutrition education. The integration of Natural Language Processing (NLP) technology with digital animation systems creates new opportunities in developing interactive applications in the context of Indonesian public health, with nutritional challenges in Indonesia showing 21.5% of toddlers experience stunting and 12.2% of adults face obesity, indicating an urgent need for accessible and comprehensive nutrition education. This research aims to develop the GiziAI website that integrates Retrieval Augmented Generation (RAG) technology with digital human avatars to provide nutrition education to Indonesian society. The research method implementing the Nusantara 2.7B Indo Chat large language model, ChromaDB as vector database, Three.js for 3D rendering, ElevenLabs for text-to-speech, and Rhubarb for lip synchronization, with React JS, Flask, MySQL, and LangChain frameworks. Evaluation was conducted using LangSmith to measure model response time, BERTScore to measure answer accuracy, and black box testing for website functionality. Research results show that the RAG system significantly improves model performance with precision increase of 71.5%, recall 60.6%, and F1-score 65.8%, while GPU usage accelerates response by 13.5% compared to CPU. Black box testing shows all website features function as expected.

Corresponding Author:

Genta Swarawisesa Erliarto Putra,
Department of Information Systems,
Gunadarma University,
Jl. Margonda Raya No.100, Depok 16424, West Java
Email: genta.swara@gmail.com
https://doi.org/10.52465/joscex.v6i3.619

1. INTRODUCTION

The development of Artificial Intelligence (AI) technology based on Large Language Models (LLM) has opened up innovative opportunities in the field of health education, particularly through the implementation of chatbots for the dissemination of nutritional information [1], [2]. The integration of Natural Language Processing (NLP) technology with digital animation systems creates new opportunities in the development of

interactive applications that can improve the accessibility of health information for the society [3]. However, the main challenges faced are the limitations of LLM in providing accurate and contextual information, as well as the tendency to generate hallucinations or information that appears reasonable but is in fact incorrect [4].

The public health context in Indonesia shows a high urgency for comprehensive and accessible nutrition education. Data from the Indonesian Ministry of Health in 2023 reveals an alarming situation in which 21.5 % of toddlers still suffer from stunting and 12.2% of adults face obesity problems [5]. This phenomenon of dual burden malnutrition shows that Indonesians need targeted nutrition education that is easy to understand and accessible at any time through familiar technology platforms.

Retrieval Augmented Generation (RAG) technology has emerged as a solution to overcome the fundamental limitations of LLMs by combining the generative capability of language models integrated with information retrieval systems from trusted external sources [6],[7]. Previous research by [8] demonstrated the effectiveness of RAG in improving the accuracy of AI systems for medical recommendations, while [9] successfully implemented RAG with the Mistral 7B LLM for an Indonesian medical herbal information system. Related study was undertaken by [10], who developed a Natural Language Processing system for stunting prevention, demonstrating the need for AI technology for stunting prevention in Indonesia. On the other hand, the implementation of RAG in the field of nutrition education integrated with digital human avatars has not been explored in depth, especially in the context of Indonesian culture and language.

The identified research gap shows that most RAG implementations in the health sector still rely on conventional text-based chatbot interfaces, especially for communities with varying levels of health literacy [6]. This study aims to develop an RAG system integrated with a digital human avatar on the GiziAI website, using the Nusantara 2.7B Indo Chat language model and ChromaDB as a vector database, to provide nutrition education to the Indonesian society through more natural and engaging interactions. The methods used in this study were selected taking into account the characteristics of the problem and the objectives to be achieved. The use of Retrieval Augmented Generation (RAG) is based on the fundamental weakness of LLM, which often produces inaccurate information (hallucinations) when faced with specific knowledge-based questions. With the integration of RAG, the system is able to combine the generative capabilities of language models with information retrieval mechanisms from trusted external sources, resulting in more relevant and accurate answers. Meanwhile, the use of digital human avatars was chosen because it can provide a more immersive and natural interaction experience, which is important in the context of nutrition education in Indonesia, where differences in public health literacy levels are quite high. This approach allows for the presentation of information not only through text, but also through visualizations and audio, supporting better user understanding. The scientific contribution of this research lies in the innovative combination of RAG technology, digital human avatar, and specific Indonesian nutrition domain knowledge, which can serve as a model for the development of interactive health education systems in the future.

2. METHOD

The stages of this study aim to provide an overview of the steps taken in creating a system using a modified waterfall approach with parallel development. Document dataset processing will use the langchain framework to facilitate RAG system development. The dataset used in this study consists of nutrition-related documents collected from various trusted sources, such as the 2023 Indonesian Health Profile published by the Indonesian Ministry of Health, the Balanced Nutrition guidelines from the Ministry of Health, and scientific articles and national publications on the issues of stunting, obesity, and food consumption patterns in Indonesia. The documents used are primarily PDF and DOCX file types, with content consisting of narrative text, nutritional recommendation tables, and public health indicators. Key features of the dataset include nutrition keywords such as stunting, obesity, balanced diet, numeric indicators including prevalence rates, consumption proportions, and descriptions of policies or recommendations. All documents were processed using a document loader within the LangChain framework to extract text content, then divided into several chunks of approximately 1,000 words with a 300-word overlap. This process aims to maintain contextual coherence in each chunk so that the model can produce more complete and accurate answers. Next, the divided text is converted into a vector representation using the SentenceTransformer model, then stored in ChromaDB as a vector database. This structured dataset is the knowledge base of the RAG system to answer user questions in a relevant and contextual manner.

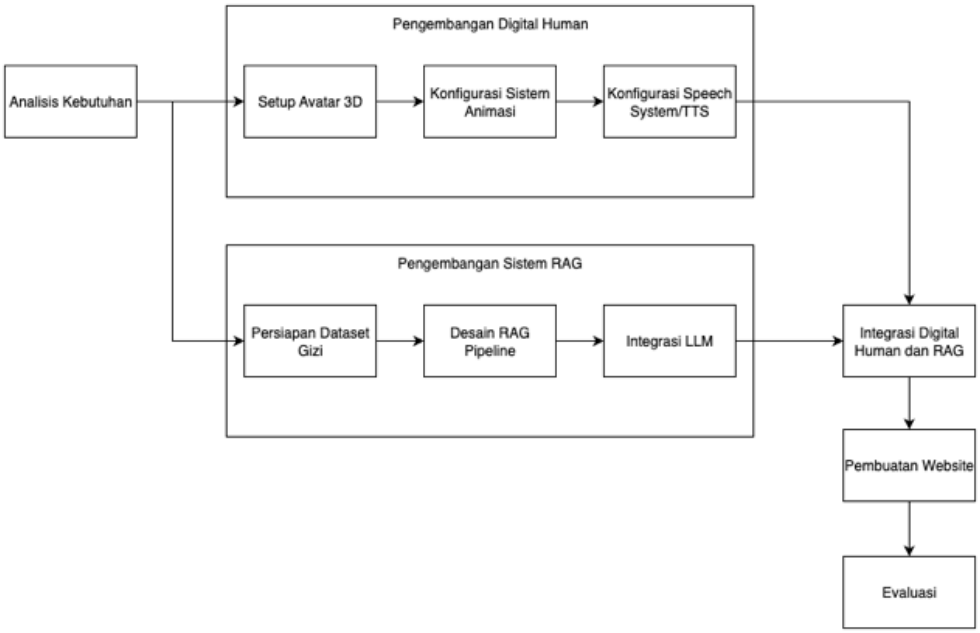

Figure 1. Research stages

This research phase was conducted through a series of systematic steps to ensure the developed system could function according to its nutritional education objectives. As shown in Figure 1, the process began with an analysis of hardware, software, and supporting library requirements to ensure the infrastructure was suitable for the processing capacity of a large language model. Following this, parallel development of a digital human avatar and a Retrieval Augmented Generation (RAG) system was conducted. The digital human avatar was created to provide more natural visual and audio interactions, while the RAG system was designed to enable the model to generate accurate answers by accessing a vector database containing trusted nutrition documents. The two were then integrated through an API on the backend so that user interactions with the avatar could be enriched with answers based on valid knowledge. The next phase was the development of an interactive website as the main platform, featuring a 3D avatar capable of speaking and responding to user questions about nutrition. To ensure system reliability, an evaluation phase was conducted that included testing answer accuracy with BERTScore, comparing CPU and GPU performance in terms of response speed, and black-box testing to ensure all website functions ran as expected. The evaluation section of this study is designed to assess system performance in terms of answer accuracy, response speed, and functional reliability. First, the answer accuracy evaluation is carried out using the BERTScore metric with three main indicators, namely precision, recall, and F1-score. Precision is used to measure the extent to which the generated answer is relevant to the reference answer, recall measures the completeness of the information captured by the model, while the F1-score provides an overview of the overall answer quality through a combination of the two metrics. Second, the system response speed test is carried out by comparing the time required for the model to generate an answer using the CPU and GPU, so that the efficiency of computing performance can be determined. Third, the website reliability evaluation is carried out using the black box testing method, which tests the main functions without viewing the internal code, including user interaction with digital avatars, input and output validity, the system's ability to display text, sound, and animation, and the stability of admin access.

3. RESULTS AND DISCUSSIONS

Creatiom Digital Human

This stage will showcase the results achieved in the creation of digital humans, ranging from the implementation of 3D avatar creation, the implementation of animation system configuration, and the implementation of speech system configuration.

3D Avatar Implementation

Figure 2 shows the results of the 3D avatar implementation, which has been customized to obtain the character visuals needed for the creation of this system. The customizations made include determining the gender to be male, body shape, hairstyle, and clothing.

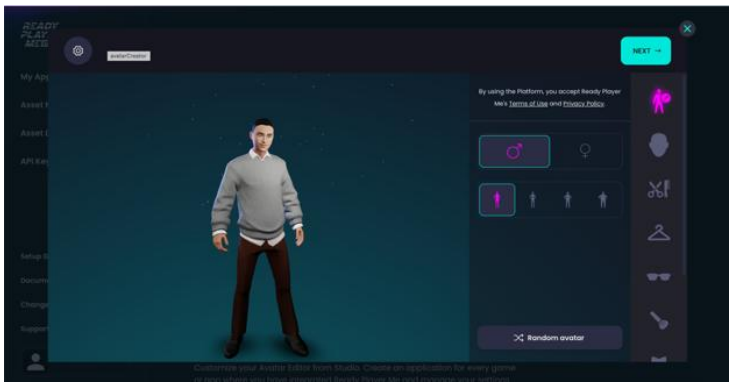

Figure 2. Result of 3D avatar creation

After the 3D avatar has been created, you will receive a link to download the avatar in .glb format. Then, convert the downloaded avatar in .glb format into a react component via the website https://gltf.pmnd.rs/.

Animation System Implementation

The next step in creating a digital human is to implement the animations that will be used by the 3D avatar using the Mixamo platform. In this study, there are 4 animations used: Idle, Sad Idle, Talking Two, and Talking One.

The Idle animation was chosen because it shows that this gesture is a natural standing posture that reflects attention and readiness to interact in accordance with the principles of communication in the context of health education. This animation will be visible when users first access the website.

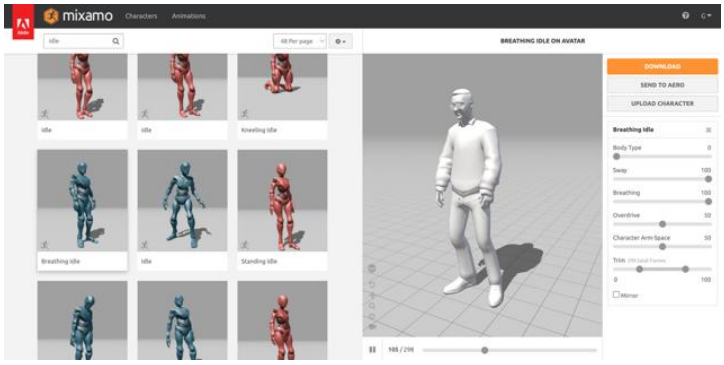

Figure 3. Idle animation

The Sad Idle animation is implemented based on body language indicators to show empathy and concern. The trigger for this gesture is based on emotional appropriateness when the system experiences a failure in providing an answer or when the answer to a user's question is not found in the model's basic knowledge.

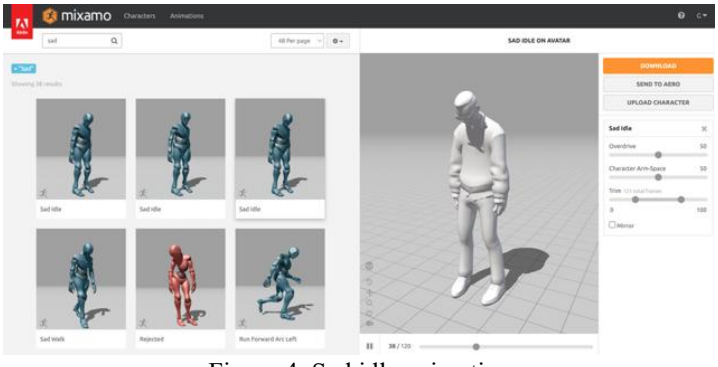

Figure 4. Sad idle animation

The next animations, Talking Two and Talking One, were selected based on the synchronization of natural speech gestures, where Talking One uses a single open hand gesture to convey trust and honesty, while Talking Two uses a two-handed gesture to convey complex concepts. Both are adapted to Indonesian communication patterns that prioritize politeness and proportional expression in an educational context.

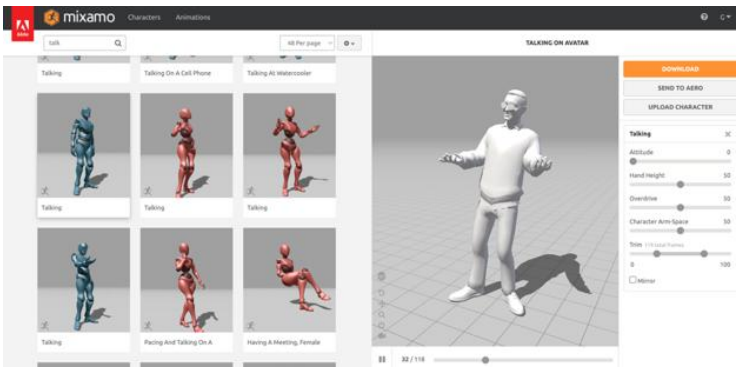

Figure 5. Talking two animation

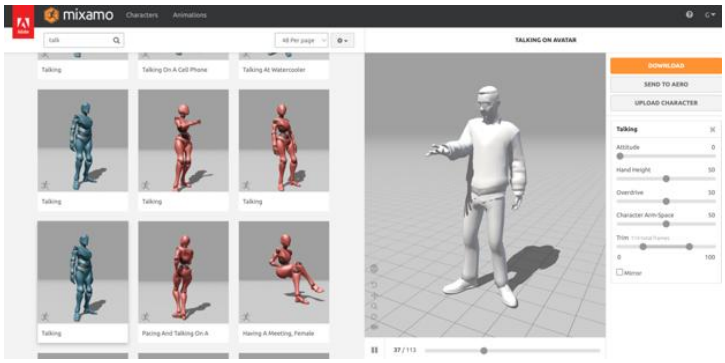

Figure 6. Talking one animation

After all the animations have been obtained, the next step is to combine the four animations into a single animation. These four animations are combined using the Blender application, as shown in Figure 8.

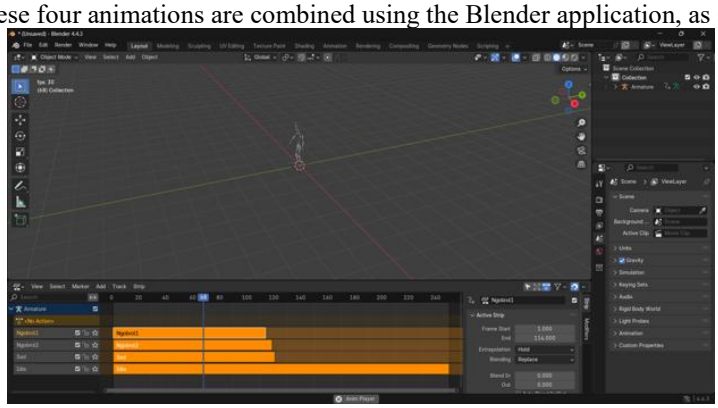

Figure 7. Combining animations using blender

Next, perform mapping to control animation selection based on the context of the system response. The animation mapping system implements two types of mapping. The first mapping is emotion_animations, which has basic emotion keys (default, smile, sad) linked to the previously prepared animations. Meanwhile, the second mapping is context_animations based on the topic or context of the conversation for the nutrition domain [11], [12].

Speech System Implementation

In this speech system implementation stage, we use the API from ElevenLabs for the voice generated in the Text to Speech system. The API is created by visiting the website https://elevenlabs.io, then searching for a voice that is suitable for the GiziAI application [13],[14].

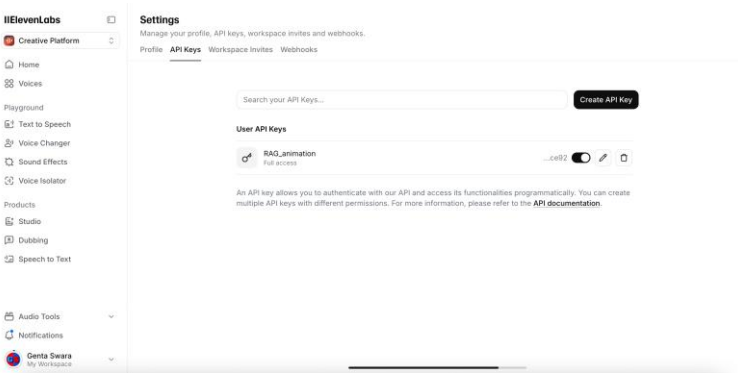

Figure 8. ElevenLabs API

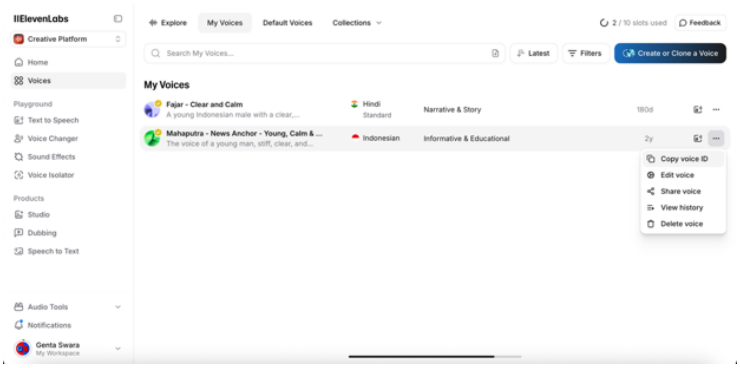

Figure 8. TTS voice

After the API has been created, the next step is to copy the voice ID from the selected voice, Mahaputra, to be entered into the program's .env configuration. Then, to create lip-sync movements from the avatar that move in accordance with the sound produced, an application called Rhubarb is needed [15].

RAG System Development

Next, this stage will display the results achieved during the creation of the RAG system based on the pipeline that has been created using the LangChain framework [16], [17].

Document Loader

The document loader is the stage where the system reads uploaded documents so that they can be processed in the next stage, which is the text splitter. This document reading process uses the PyPDFLoader function for .pdf files and Docx2txtLoader for .docx files from the langchain_community package with the document_loaders submodule.

Text Splitter

After the document has been successfully loaded, the next step is to divide the document into several chunks to make it easier for the LLM to process the document using the RecursiveCharacterTextSplitter function from the langchain package, namely langchain_text_splitters. This process will divide the document into chunks with a length of 1000 words each, and each new chunk will take the last 300 words from the previous chunk. This is done to ensure that the important information and context of each chunk is preserved, so that the answers generated will remain complete and accurate.

Embedding Model

This model embedding stage will use the SentenceTransformerEmbeddings function from the langchain_community package submodule sentence_transformer. The embedding model will be loaded first using the SentenceTransformerEmbeddings function with the model_name parameter to determine the model

to be used for the embedding process. The name of this embedding model is placed in the config file. Then, the embedding process is carried out by converting the document text into numerical vectors and adding them directly to the vector store [18], [19].

Vector Database

The steps after embedding are to insert the embedding vector into the vector database using ChromaDB. This process uses the Chroma library from the package langchain_chroma, with the parameters collection_name for the collection name in the vector database, persist_directory for the vector database storage directory taken from the config file, and embedding_function which comes from the previous process, namely the embedding model. This vector database enables similarity searches to find the most relevant documents to the user's query.

Retrieval

This retrieval stage will utilize the vectorstore from the previous stage as a retriever to search for documents relevant to the user's query. The vector store with the as_retriever method has parameters such as search_type using the mmr (maximal marginal relevance) type to reduce redundancy in search results, then search_kwargs or search parameters with k to retrieve the top 5 most relevant documents, fetch_k to retrieve 10 candidate documents, and lamda_mult to balance the relevance and diversity of results.

LLM Integration

Integrating LLM using the LlamaCpp library from the langchain_community package to load local models with the .gguf extension with parameter configurations including model_path for the model directory, n_gpu_layer of 20 layers to balance CPU and GPU usage given hardware limitations, temperature to regulate response creativity, top_p for word selection variation control, repeat_penalty to prevent word repetition, max_token limited to 256 tokens for responses, and n_ctx of 2048 tokens for context memory capacity. The system uses ChatPromptTemplate from langchain_core with a zero-shot prompting approach and answer_generation_chain to generate an integrated automated workflow.

Digital Human and RAG Integration

The process of integrating digital humans and RAG involves creating an API on the backend using the Flask framework to handle requests from the frontend when users ask questions. The API then returns a JSON response containing the text answer, facial expression, animation, audio, and lip-sync data generated, as shown in Figure 10.

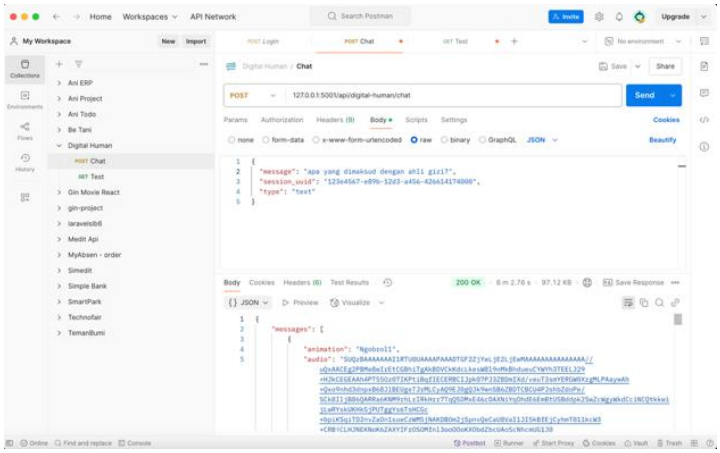

Figure 10. Chat API output

Once the response is obtained, it is returned to the frontend and processed to enable the digital human avatar to move its lips and display animations corresponding to the response generated by the backend API.

Website Development

The process of integrating digital humans and RAG involves creating an API on the backend using the Flask framework to handle

Login Page Implementation

Admins can log in to access the admin dashboard by entering their username and password registered in the database. The implementation of the login page is shown in Figure 11.

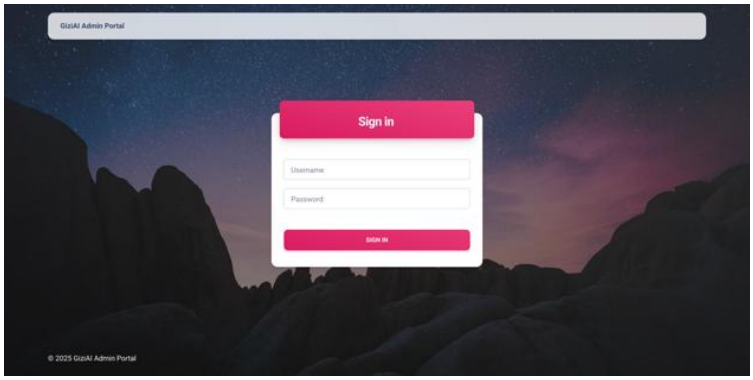

Figure 11. Login page

Digital Human Chat Page Implementation

The digital human chat page is the page that users will see when they access the GiziAI website. On this page, users can ask questions related to nutrition by filling out the input form and then sending the question to the system. Once the question has been processed, users will receive an answer to their question in the form of an animation from a 3D avatar, text, and voice.

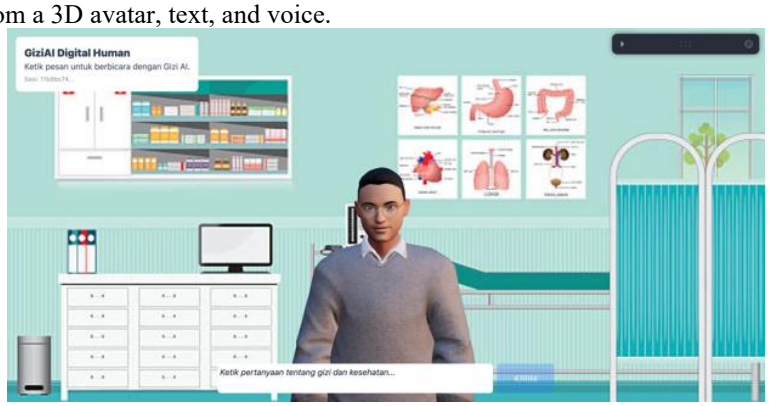

Figure 12. Digital human chat page

Admin Dashboard Page Implementation

This admin dashboard page is the main page for administrators after logging in. On this page, administrators can see the number of document files that have been uploaded, the number of conversation sessions based on session_uuid, the number of conversations in the system, and the status of the system. The implementation of the admin dashboard page is shown in Figure 13.

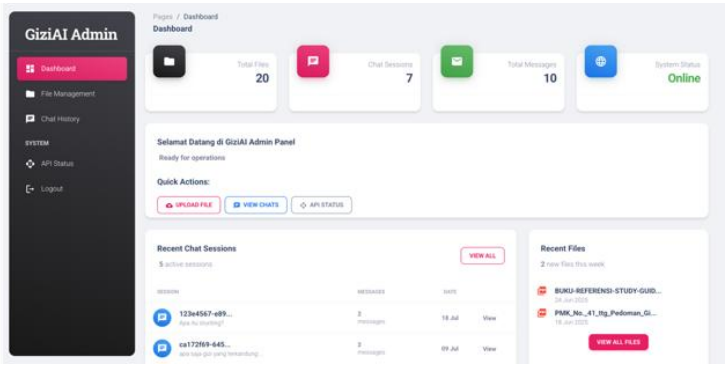

Figure 13. Admin dashboard page

.

Chat History Page Implementation

On this file management page, the admin can manage document files that will be used for the RAG system. The admin can upload files by pressing the upload file button, view all uploaded document files, and delete files that are no longer relevant by pressing the delete button, as shown in Figure 14.

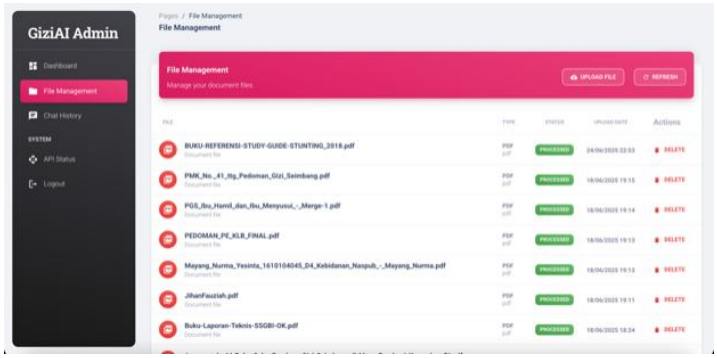

Figure 14. File Management page

Admin Dashboard Page Implementation

On this chat history page, administrators can view information related to the conversation history in the system, such as the total number of sessions, the number of chats on that day, the average number of messages per session, the time of the last conversation activity, and all chat sessions on the system's. Administrators can also view all messages in a selected session by pressing the view button, which will display a pop-up as shown in Figure 16.

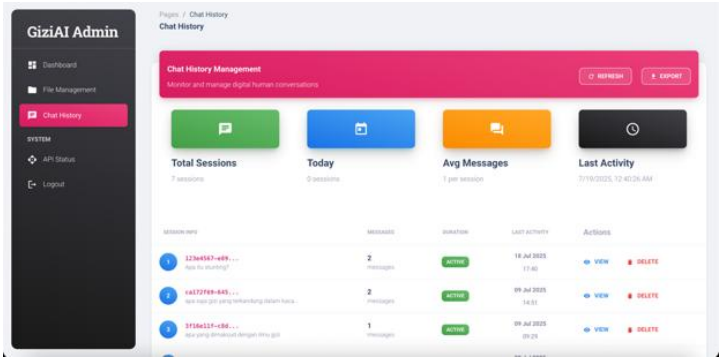

Figure 15. Chat history page

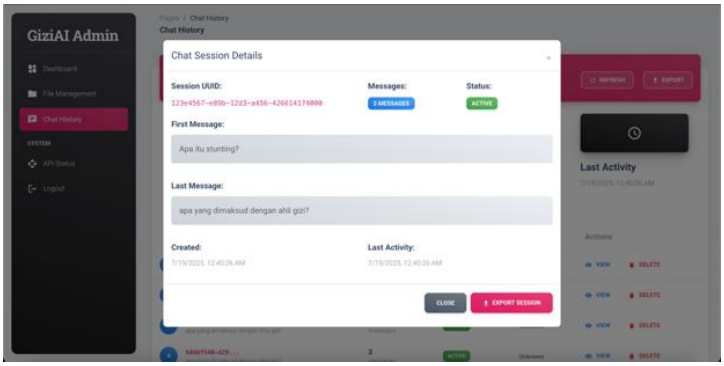

Figure 16. Chat history popup

API Status Page Implementation

When clicking the API Status menu in the sidebar, the admin will be redirected to a new tab displaying the JSON of the currently running API status.

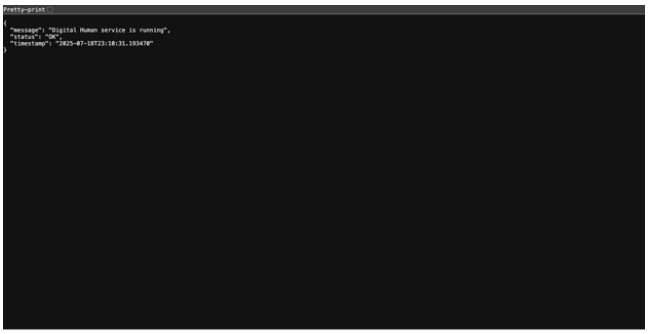

Figure 17. API status page

Deployment

During this deployment stage, the GiziAI website will use a Virtual Private Server (VPS) from one of the hosting service providers, namely https://www.niagahoster.co.id/. The website can be accessed via the URL https://giziai.anitech.id/ for the user page and https://admingiziai.anitech.id/ for the admin page.

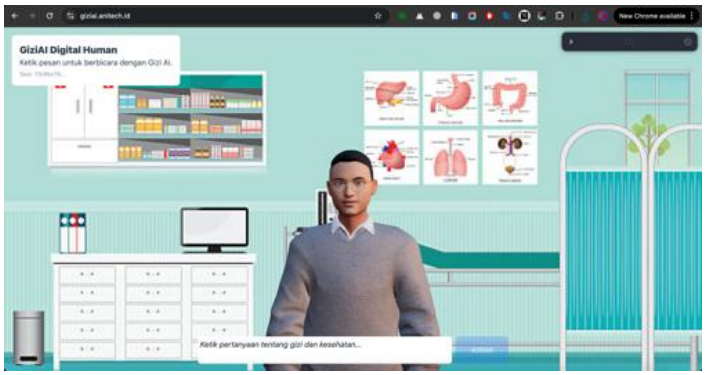

Figure 18. User page website successfully launched

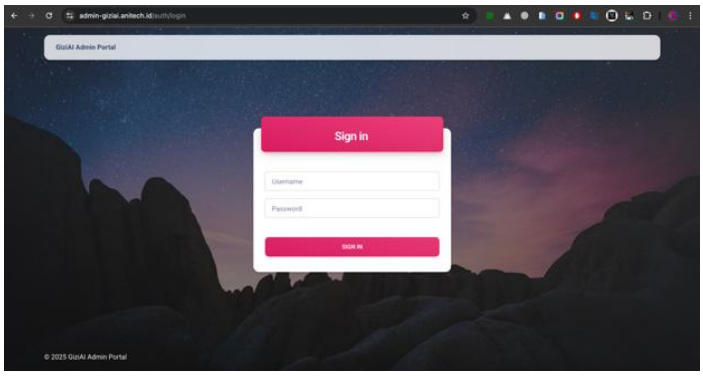

Figure 19. Admin page website successfully launched

Evaluation Results

The evaluation stage was conducted to measure the performance of the large language model and test the GiziAI website to determine whether the web application could run as expected. Model performance testing will include comparing the time taken by the model to generate answers using LangSmith and comparing the accuracy of the generated answers with the reference answers before and after implementing the RAG system using BERTScore. Meanwhile, website testing will use Black Box Testing. Table 1 shows the questions used to test the large language model.

Table 1. Evaluation questions

No	Questions
1	What is meant by the pillar of food diversity?
2	How many servings of vegetables are recommended per day?
3	Why is clean living important in nutrition?
4	What is the function of physical activity in nutrition?
5	How can you maintain a normal weight?
6	What are the indicators of poor nutritional status in toddlers?
7	Why is exclusive breastfeeding important for babies aged 0–6 months?
8	What are the risks of excess sugar, salt, and fat?
9	What are the recommendations for drinking water?
10	What is the purpose of the Balanced Nutrition Guidelines?

Comparison of CPU and GPU Latency

Based on the questions in Table 1, testing was conducted to determine the time required by the model to generate answers. This test compares the time required by the model to generate answers using a CPU and a GPU. Table 2 shows the time required when using a CPU.

Table 2. Comparison of CPU and GPU time

Questions	CPU	GPU
1	14.40s	9.47s
2	4.83s	8.27s
3	17.93s	9.45s
4	9.77s	8.28s
5	10.04s	9.08s
6	12.64s	14.43s
7	12.55s	10.30s
8	20.96s	23.33s
9	14.21s	9.04s
10	10.97s	9.33s
Average	12.83s	11.098s
Performance Improvement	1.16x	
Percentage Improvement	13.5%	
Time Reduction	1.732s	

From the time comparison results in Table 2, the use of a GPU can increase the speed of the model to produce answers 1.16 times faster or an increase of 13.5% compared to a CPU. The average time obtained by the CPU from a total of 10 questions was 12.83 seconds, which is 1.732 seconds slower than the average time obtained by the GPU, which was 11.098 seconds.

BERTScore

BERTScore testing was conducted based on the questions in Table 1 to determine the accuracy of the answers generated by the large language model using three main metrics, namely precision, recall, and F1, which have values between 0 and 1. The higher the value obtained, the better the quality of the answers generated by the model.

Table 3. BERTScore nusantara 2.7B indo chat non RAG

Questions	Precision	Recall	F1
1	0.437459	0.470109	0.453197
2	0.545308	0.490097	0.516230
3	0.424433	0.547126	0.478032
4	0.460617	0.614621	0.526590
5	0.379830	0.512522	0.436310
6	0.375450	0.504351	0.430458
7	0.471776	0.445372	0.458194
8	0.544553	0.644454	0.590307
9	0.408586	0.479198	0.441084
10	0.437703	0.545922	0.485860

Table 3 shows the results of BERTScore testing for the Nusantara 2.7B Indo Chat model without the RAG system. From this table, we can see that the precision, recall, and f1 scores are still not very good. This indicates that the Nusantara 2.7B Indo Chat model without the RAG system has limited performance when it comes to specific knowledge about nutrition.

Table 4. BERTScore nusantara 2.7B indo chat RAG

Questions	Precision	Recall	F1
1	0.465985	0.640488	0.539476
2	0.650439	0.532076	0.585334
3	0.696732	0.945419	0.802245
4	1.000000	1.000000	1.000000
5	1.000000	1.000000	1.000000
6	1.000000	1.000000	1.000000
7	1.000000	1.000000	1.000000
8	0.537302	0.858450	0.660930
9	0.504279	0.527055	0.515415
10	0.837579	0.933755	0.883056

The results of the BERTScore testing for the Nusantara 2.7B Indo Chat model with the RAG system are shown in Table 4. From these results, there was an increase in precision, recall, and f1 scores compared to the model without the RAG system. This indicates that the performance of the Nusantara 2.7B Indo Chat model, when enhanced with the RAG system, will improve for specific knowledge about nutrition.

Table 5. Average BERTScore

Model	Precision	Recall	F1
Nusantara 2.7B Indo Chat Non RAG	0.4486	0.5254	0.4816
Nusantara 2.7B Indo Chat RAG	0.7692	0.8437	0.7986

From the results obtained in Table 3 and Table 4, the average for each metric shows a significant improvement when the Nusantara 2.7B Indo Chat model is enhanced with the RAG system. The improvement in the precision metric from 0.4486 to 0.7692 or an increase of 71.5% shows that the RAG system can help the model generate accurate and relevant answers between the results and the reference answers.

The improvement in the recall metric from 0.5254 to 0.8437, or an increase of 60.6%, shows that the RAG system can help the model generate complete answers and capture important information contained in the reference answers in accordance with the base knowledge. The improvement in the last metric, f1, from 0.4816 to 0.7986, or an increase of 65.8%, shows an improvement in the overall quality of the answers generated by the model.

Black Box Testing

The GiziAI website will use black box testing to verify that the website's functionality works correctly as expected [20].

Table 6. Black box testing

No	Scenario	Expected Results	Test Results	Conclusion
1	Users access the main page and input questions related to nutrition	The system displays a 3D avatar, receives and processes input from users	As Expected	Success
2	Users receive answers to their questions	The system generates and displays answers in the form of text, voice, and appropriate animations	As Expected	Success
3	The administrator logs in with the correct username and password	The system directs to the admin dashboard	As Expected	Success
4	The admin accesses the "File Management" page	Displays a list of uploaded files	As Expected	Success
5	The admin uploads a document file in .pdf or .docx format	File successfully uploaded and appears in the file management list	As Expected	Success
6	The administrator clicks the option to delete the uploaded file	The file was successfully deleted and disappeared from the file management list	As Expected	Success
7	The administrator accesses the chat history page	Displaying a list of conversation sessions in the system	As Expected	Success
8	The administrator clicks the "View" button on one of the chat sessions	The system displays a popup with the conversation details in that session	As Expected	Success
9	The admin clicks the "Delete" button on one of the chat sessions	The chat session was successfully deleted and removed from the list	As Expected	Success

No	Scenario	Expected Results	Test Results	Conclusion
10	The admin clicks the button to export the chat history to CSV	The conversation data was successfully downloaded in CSV format	As Expected	Success
11	The admin accessed the API status page	Opens a new tab and displays the API status in JSON format	As Expected	Success

Comparison Of Research Results With Previous Research

The comparison highlights that while previous studies have demonstrated the potential of advanced AI methods such as RAG and NLP in healthcare and stunting prevention, their applications remained limited to either clinical recommendations or expert systems with text-based interfaces. Yang et al. [5] showed that RAG reduces bias and enhances transparency in healthcare recommendations, Yusuf et al. (2025) [10] achieved high accuracy in stunting prevention through NLP, and Firdaus et al. (2024) [9] validated RAG's effectiveness in the domain of Indonesian herbal medicine. Building on these foundations, this study advances the field by applying RAG specifically to nutrition education in Indonesia and integrating it with digital human avatars, thereby not only ensuring accuracy and reliability but also enhancing user engagement and accessibility through immersive interaction.

Table 7. Comparison of research results with previous research

No	Researchers	Focus & Key Findings of Previous Studies	Relevance & Comparison with This Study
1	Yang et al. (2025) [5]	Demonstrated that RAG systems in healthcare can reduce bias, improve source transparency, and provide personalized recommendations based on patient data.	This study aligns with those findings, as evidenced by significant performance improvements with precision increasing by 71.5%, recall by 60.6%, and F1-score by 65.8% in the domain of nutrition education. The novelty lies in applying RAG for public nutrition education rather than individual clinical recommendations.
2	Yusuf et al. (2025) [10]	Developed an expert system for stunting prevention using NLP and forward chaining, achieving 97% accuracy and web-based accessibility.	This study extends the approach from NLP to a RAG-based system with digital human avatars. The advantage is not only accuracy but also immersive interaction that facilitates laypeople's understanding of nutritional information.
3	Firdaus et al. (2024) [9]	Integrated RAG with Mistral 7B for Indonesian medical herbs, achieving a METEOR score of 0.22%, outperforming Llama2 7B (0.14%), and validated results with nine academic journals.	This study similarly confirms the validity of RAG but broadens its application to nutrition education. The distinct contribution is the integration with digital human avatars, making the system more practical and engaging for public health literacy in Indonesia.

4. CONCLUSION

The GiziAI website was successfully developed by integrating Retrieval Augmented Generation (RAG) technology and digital human avatars using the Nusantara 2.7B Indo Chat model, ElevenLabs for Text to Speech, and Rhubarb for lip sync synchronization, which can be accessed via https://giziai.anitech.id/. Evaluation results show that GPU accelerates model response by 13.5% compared to CPU, the RAG system significantly improves performance with precision increasing by 71.5%, recall by 60.6%, and F1-score by 65.8%, while black box testing confirms that all website functions are running as expected.

Given the hardware and server limitations in this study, the GiziAI website can be improved through the use of larger parameter language models for more complex responses, the addition of Speech to Text voice recognition features for more interactive interactions, and the implementation of conversation history on the user display to facilitate access to previous conversations.

REFERENCES

[1] C. Rozali, A. Zein, dan E. S. Eriana, "Artificial Intelligence (AI) Dimasa Depan : Tantangan Dan Peluang," *Jitu J. Inform. Utama*, vol. 2, hal. 66–71, 2024, doi: https://doi.org/10.55903/jitu.v2i1.177.

[2] A. Ahmad, "Mengenal Artificial Intelligence , Machine Learning , Neural Network , dan Deep Learning," *J. Teknol. Indones.*, 2017.

[3] R. C. Tarumingkeng, "Natural Language Processing (NLP)," in *RUDYCT e-PRESS*, no. November, 2024.

[4] V. Gumma, A. Raghunath, M. Jain, dan S. Sitaram, "HEALTH-PARIKSHA: Assessing RAG Models for Health Chatbots in Real-World Multilingual Settings," *arXiv*, 2024, doi: https://doi.org/10.48550/arXiv.2410.13671.

[5] R. Yang *et al.*, "Retrieval-augmented generation for generative arti fi cial intelligence in health care," *npj Heal. Syst.*, 2025, doi: 10.1038/s44401-024-00004-1.

[6] K. N. L. P. Tasks *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv*, 2020, doi: https://doi.org/10.48550/arXiv.2005.11401.

[7] M. A. Hasbi, R. Imanda, dan M. F. Fauzan, "Implementasi Chatbot Berbasis Large Language Model Untuk Pencarian Skripsi Mahasiswa Terintegrasi dengan Whatsapp," *J. Comput. Sci. Artif. Intell.*, vol. 5, no. 1, hal. 148–167, 2025, doi: https://dx.doi.org/10.29240/arcitech.v5i1.13974 Implementasi.

[8] P. K. Indonesia, *PROFIL KESEHATAN INDONESIA 2023*. 2023.

[9] D. Firdaus, I. Sumardi, dan Y. Kulsum, "Integrating Retrieval-Augmented Generation with Large Language Model Mistral 7b for Indonesian Medical Herb," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 9, no. 3, hal. 230–243, 2024, doi:

https://doi.org/10.14421/jiska.2024.9.3.230-243.
[10] M. Yusuf, I. P. Sari, dan V. Kristy, "Sistem Pakar Mencegah Stunting dengan Menentukan Gizi Anak Menggunakan Natural Language Processing (NLP)," *J. JTIK (J. Teknol. Inf. dan Komun.)*, vol. 9, no. September, hal. 924–934, 2025, doi: https://doi.org/10.35870/jtik.v9i3.3614.
[11] M. Korban dan X. Li, "A Survey on Applications of Digital Human Avatars toward Virtual Co-presence," *arXiv*, no. May, 2021, doi: https://doi.org/10.48550/arXiv.2201.04168.
[12] J. Sanchez-riera, A. Civit, dan M. Altarriba, "AVATAR : Blender add-on for fast creation of 3D human models," *arXiv*, hal. 1–7, 2020, doi: https://doi.org/10.48550/arXiv.2103.14507.
[13] M. H. Mubarak dan A. B. Santoso, "Persepsi Mahasiswa dalam Penggunaan Aplikasi Berbasis Text to Speech Pada Mata Kuliah Teknologi Pembelajaran Bahasa Arab," *J. Ilm. Iqra*, vol. 17, hal. 73–84, 2023, doi: https://doi.org/10.30984/jii.v17i1.2376.
[14] A. Fauzan dan S. Hartati, "Text to Speech untuk Bahasa Arab Menggunakan Perangkaian Diphone (Text to Speech for Arabic Using Diphone Concatenation)," *J. UMP*, vol. VI, hal. 9–14, 2018.
[15] M. B. Nendya *et al.*, "AUTO LIP-SYNC PADA KARAKTER VIRTUAL 3 DIMENSI," *REKAM J. Fotogr. Telev. dan Animasi*, vol. 11, no. 2, hal. 137–144, 2015, doi: https://doi.org/10.24821/rekam.v11i2.1299.
[16] K. el Haddad, F. Zajega, dan T. Dutoit, "An Open-Source Avatar for Real-Time Human-Agent Interaction Applications," *IEEE*, 2019, doi: https://doi.org/10.1109/ACIIW.2019.8925115.
[17] A. Vaswani, "Attention Is All You Need," *arXiv*, no. Nips, 2017, doi: https://doi.org/10.48550/arXiv.1706.03762.
[18] J. Camacho-collados, "Embeddings in natural language processing: Theory and advances in vector representations of meaning," 2020.
[19] L. Wang, N. Yang, X. Huang, dan B. Jiao, "Text Embeddings by Weakly-Supervised," *arXiv*, hal. 1–17, 2022, doi: https://doi.org/10.48550/arXiv.2212.03533.
[20] M. N. Ichsanudin dan M. Yusuf, "PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, hal. 1–8, 2022, doi: https://doi.org/10.55123/storage.v1i2.270.

.