IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

EDUCATIONBOT: An AI-Powered Multi-Input Chatbot for Student Assistance

Tushar Navale[1], Chetan Gabhane[2], Onkar Wagh[3], Aniket Shinde[4], Prof. P. V. Nagare[5], Prof. H. P. Bhabad[6]
[1234] Student, [56] Faculty, Department of Computer Engineering
Loknete Gopinathji Munde Institute of Engineering Education and Research, Nashik, India
[1] tusharnavale733@gmail.com [2] chetangabhane9604@gmail.com
[3] onkarwagh1519@gmail.com [4] aniketshinde87067@gmail.com
[5] nagarepravin1189@gmail.com [6] harish.bhabad@logmieer.edu.in

Abstract: *The quick growth of content and unstructured data in schools has created a big need for smart systems that can find information fast. Traditional search methods that use keywords often do not understand what students are really asking for which leads to search results and frustration for the students. This paper talks about EDUBOT, a system that uses methods to help students find information by talking or typing and it is made for universities. The system uses a FastAPI backend and other tools like Facebook AI Similarity Search and Groqs Large Language Model to understand what students are asking for and to answer them quickly. It also uses Whisper to turn speech into text. Google Text-to-Speech to create audio answers. By breaking down text into pieces using special codes to represent words and processing information quickly the system reduces mistakes and keeps student information private. This paper explains how the system works from start to finish looks at how it can process information and presents a way to make it work for many students at the same time so they can get help with their questions, in real time. The EDUBOT system is designed to help students find information quickly and accurately. It uses the latest technology to make this happen including EDUBOT and its retrieval system. The goal of the EDUBOT system is to make it easy for students to find what they need and to make sure that the information they get is relevant and useful using the system and its features.*

Keywords: EDUCATIONBOT: An AI-Powered Multi-Input Chatbot for Student Assistance

I. INTRODUCTION

A. Background and Significance

The way Information Retrieval and Artificial Intelligence work together has changed how people use sets of data. [1] In colleges there are amounts of text that are not organized like class schedules, student lists, exam schedules and course documents. [2] Because there is much digital content now people need systems that can find the right information quickly. [3] The old way of searching using keywords has problems because it can get confused when words have meanings or are used in different contexts. [4] This can lead to search results. [5] New Natural Language Processing techniques are trying to solve this problem by using vector-based search to make searches more accurate and relevant. [6] Information Retrieval systems that use these techniques can understand the meaning of words. [7] The use of Large Language Models has also improved Information Retrieval by allowing for real-time classification of documents expansion of search queries and generation of responses based on content. [8] Voice assistants make it possible for people to use instructions and automate tasks, which makes it easier for people to use these systems. [9]

B. The Problem of Hallucination and Domain-Specificity General-purpose Large Language Models are good at having conversations. [11] They are not as accurate when it comes to specific areas of knowledge. [12] This is because they

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

are limited by the data they were trained on. [13] In education it is crucial to have information and if a model provides incorrect information it can cause serious problems. [14] For example if a model gives the exam date it can cause operational problems. [15] It is not practical to retrain these models every time there is an update because it would be too expensive. [16] Retrieval-Augmented Generation is a technique that can solve this problem by using sources of information to improve the accuracy of the model. [17] However it is still a challenge to handle pieces of text and keep the context clear. [18] Traditional methods of breaking up text often lose information, which reduces the effectiveness of the search. [19]

C. Proposed Contribution

This research presents a working voice-enabled pipeline that uses Retrieval-Augmented Generation. [22] Is deployed using a concurrent API architecture. [23] The main contributions of this paper are: Multimodal Integration: This allows people to interact with the system using voice commands making it more accessible. [24] Adaptive Document Processing: This is a technique that preserves the meaning of documents by breaking them up into smaller chunks. [25] Low-Latency Dense Retrieval: This uses a technique called FAISS indexing to make searches faster and more accurate. [1] Privacy-, by-Design Architecture: This is a way of designing the system that prioritizes data security and makes sure that data is deleted from memory immediately after it is used. [2]

II. LITERATURE REVIEW

A. Evolution of Semantic Search and Vector Databases. [7] The way we search for things on the internet is changing. [8] We used to type in keywords but now we can use semantic search, which understands the context of what we are looking for. [9] This is made possible by something called vector databases. [10] Traditional databases are not very good at handling data so vector databases were created to support fast and accurate searches. [11] There are frameworks like FAISS that help with this. [12] They use things like Inverted File and Hierarchical Navigable Small World graphs to make searches faster and more accurate. [13] These algorithms help us find a balance between how we want our searches to be and how accurate we want them to be. [14] One of these algorithms called HNSW is very good at finding things and is often used in real-time applications. [15]

However semantic search is not perfect. [16] Sometimes it can struggle to find matches especially when it comes to specific words or phrases. [17] This is why some people are using something called Hybrid Retrieval, which combines search with other methods. [18] This approach helps to improve the accuracy of searches and makes them more robust. [19] Studies have shown that Hybrid Retrieval can even help to reduce errors in language models. [20]

B. Retrieval-Augmented Generation and Multimodality There is a way of generating text called Retrieval-Augmented Generation or RAG. [22] It uses language models to create text. [23] It also uses a search function to find relevant information. [24] This makes the generated text more accurate and helpful. [25] RAG systems can search through amounts of data to find the information they need. [1] They can even use things like images and audio to help with the search. [2] This is called multimodality. [3] Multimodal RAG is very useful for things like education, where there are a lot of types of data. [4] It can help to create engaging and interactive learning materials. [5] The system can use things like diagrams and videos to help explain concepts. [6] This makes it easier for students to understand the material. [7] The RAG system can even use sources of data at the same time. [8] This is called -modal retrieval. [9] It allows the system to find relationships between types of data like text and images. [10] This makes the generated text more accurate and helpful. [11]

C. Graph-Enhanced RAG and Multi-Hop Reasoning

Traditional RAG systems have some limitations. [14] They can struggle to understand relationships between pieces of data. [15] This is where Graph-Enhanced RAG comes in. [16] It uses a graph to represent the relationships between pieces of data. [17] This allows the system to understand how different pieces of data are connected. [18] Graph-Enhanced RAG can use something called -hop reasoning to find relationships between pieces of data. [19] This is like a

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

game of six degrees of separation, where the system can find connections between pieces of data that are not directly related. [20] This makes the system very powerful for things like education, where complex relationships between conceptsre common. [21] The system can even use things like community detection to find groups of data. [22] This helps to create an organized and structured representation of the data. [23] The system can then use this representation to generate accurate and helpful text. [24]

D. Long-Context Models vs Retrieval Architectures

There are some language models that can handle very long pieces of text. [2] These are called long-context models. [3] They are very good at understanding relationships between pieces of data. [4] However they can be slow and expensive to use. [5] This is where retrieval architectures come in. [6] They use a search function to find information rather than trying to process the entire piece of text at once. [7] This makes them faster and more efficient. [8] They can even use things like RAG to generate text based on the retrieved information. [9] Some people are using a combination of both approaches. [10] They use long-context models for tasks that require a lot of context and retrieval architectures for tasks that require more efficient processing. [11] This allows them to balance the trade-offs between accuracy and speed. [12]

E. Agentic RAG and Evaluation Frameworks

RAG systems are becoming more advanced. [15] They can now use agents to work together to generate text. [16] These agents can communicate with each other. [17] Work together to find the best information. [18] This is called RAG. [19] It allows the system to be more flexible and adaptive. [20] The agents can even use things like search to find relevant information. [21] To evaluate these systems we need to use frameworks. [22] These frameworks can measure the accuracy and effectiveness of the generated text. [23] They can even use things like language models to judge the quality of the text. [24] This helps to ensure that the system is generating high-quality text that's relevant and helpful. [25] It also helps to prevent the system from generating misleading information.

F. The Throughput-Latency Tradeoff and Edge DeploymentsThere is a trade-off between how a system can process information and how long it takes to get a response. [3] This is called the throughput-latency tradeoff. [4] For AI this trade-off is very important. [5] We want the system to be able to respond but we also want it to be able to process a lot of information. [6] To solve this problem some people are using something called edge deployments. [7] This means that the system is deployed on a device rather than in the cloud. [8] This can help to reduce the latency and make the system more responsive. [9] We can even use things like language models to make the system more efficient. [10] These models are smaller and faster than language models but they are still very effective. [11] By using edge deployments and small language models we can create AI systems that are fast responsive and accurate. [12]

G. Privacy Risks and Governance in Enterprise RAG as RAG systems become more common there are concerns about privacy. [14] We need to make sure that the system is handling information in a responsible way. [15] This is where governance comes in. [16] We need to have rules and regulations in place to ensure that the system is being used in a way that's fair and transparent. [17] We can use things, like context- indexing to help with this. [18] This means that the system can understand the context in which the information is being used and make sure that it is being handled appropriately. [19] By using governance and context-aware indexing we can create RAG systems that're not only effective, but also responsible and trustworthy. [20]

scores This testing is important to make sure the system is good enough to be used in time, for academic or clinical support. [22]

Scalable Deployment Architecture :We need to make a Deployment Architecture for the system. [23] This Scalable Deployment Architecture is made using microservices. [24] The main goal of this Scalable Deployment Architecture is to handle a lot of users at the time. [25] We also want this Scalable Deployment Architecture to be deployed on cloud

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, June 2026 Impact Factor: 8.2

infrastructure. [1] This way the system will be good for educational institutions. [2] The Scalable Deployment Architecture will make sure the system is sustainable. [3] This is important, for educational institutions that use the system. [4]

III. METHODOLOGY

A. System. Pipeline

The EDUBOT architecture is made up of a system that processes data in stages. [10] This system is designed to handle a lot of questions from institutions. [11] First it takes in data from school syllabi and schedules. [12] This data is not organized so the system cleans it up. [13] The text is then broken down into pieces using an algorithm that respects how people naturally think. [14] These pieces are then changed into codes using a helps the system understand pre-trained model. [15] This what the text is about. [16] These codes are stored in a database called FAISS. [17] This database is on a computer so it can be accessed quickly. [18]

B. Multimodal Integration and Intent Parsing

When someone talks to the system it records what they say and turns it into text. [21] This is done using a model called Whisper. [22] Once the system has the text it tries to understand what the person means. [23] This is called extraction. [24] It makes sure the system understands the question correctly even if it is complex. [25] If someone types a question of speaking the system skips the part where it turns speech into text. [1] It just tries to understand what they mean. [2]

C. Retrieval and Autoregressive Generation

When the system is trying to answer a question it turns the question into a code. [5] This code is compared to the codes in the FAISS database to find the relevant information. [6] The system then takes the information and adds it to a template. [7] This template also includes the question. [8] The system then uses a language model to generate an answer. [9] This model is limited to using the information it found so it does not make things up. [10] Finally the system can turn the answer into speech using Google Text-, to-Speech so the person can hear the answer. [11] This completes the loop of the system talking to the person. [12] The EDUBOT architecture is designed to make this process smooth and accurate. [13] The system uses the architecture to process data and generate answers. [14]

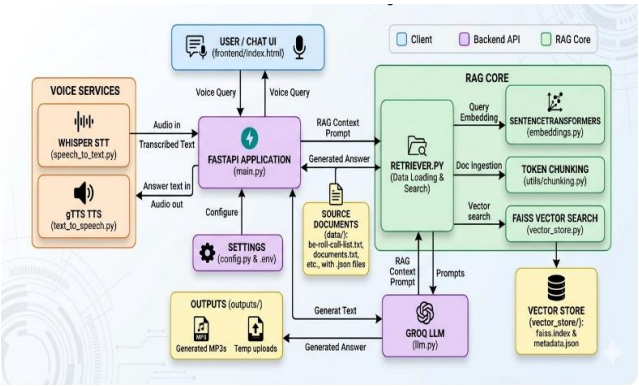

Fig.1 : architecture diagram

IV. ADVANCED ACOUSTIC MODELING AND STT OPTIMIZATION

A. Mel-Frequency Cepstral Coefficients Extraction [

When we want to use voice capabilities we need to know how to turn audio into something a machine can understand. [19] So when a student uploads a file like a wav or mp3 file we first change the sound to make it clearer. [20] The Whisper model then breaks this sound into bits that are 25 milliseconds long and it does this every 10 milliseconds. [21] For each of these bits the system figures out the Mel-Frequency Cepstral Coefficients, which's like a map of the

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

sound. [22] This map shows what the sound looks like and it helps us understand what the person is saying. [23] We make this map by looking at the sound closely and then we use some math to make it clearer. [24] This helps us focus on the persons voice and ignore any background noise. [25]

B. Mitigating Transcription Latency

One big problem with using speech-, to-text models in time is that it can take a while to get the words written down. [3] To make it faster we can use a version of the Whisper model. [4] This simpler version uses complicated math, which means it can work faster. [5] We do this by changing the way the model thinks about numbers so it uses numbers that are easier to work with. [6] This makes the model smaller and faster so it can write down what the person is saying as soon as they say it. [7] Mel-Frequency Cepstral Coefficients are still important here because they help the model understand what the person is saying. [8] By using Mel-Frequency Cepstral Coefficients and a simpler model we can make the whole process faster and more accurate. [9]

V. SYSTEM ARCHITECTURE AND FASTAPI CONCURRENCY

The EDUBOT system uses a design with many small services working together. [12] This design is managed by FastAPI. [13] A key requirement for college administration systems is handling visitors at the same time. [14]

A. [15] -Blocking Network Operations [16]

Traditional frameworks use a separate thread for each request. [17] If a request is waiting for a response the thread is. [18] Can't do anything else. [19] This quickly uses up server resources. [20] FastAPI uses non-blocking I/O with Pythons event loop. [21] When the /ask endpoint makes a request control is given back to the event loop. [22] This lets one worker process handle student requests at the same time. [23]

B. Real-Time Inference Pipeline The inference pipeline works in a sequence:

- Multimodal Input Reception: The /ask endpoint takes either text or an audio file. [1] If its a file Automatic Speech Recognition (ASR) converts spoken language into text. [2] The temporary file is deleted away. [3]
- Context Retrieval: The transcribed text is embedded. [4] The RetrieverService searches the FAISS index for chunks. [5]
- LLM Answer Generation: The retrieved context and users query are put into a template. [6] The GroqLLMService uses this to generate a response. [7]
- Text-to-Speech Synthesis: Text-to-Speech (TTS) converts text responses back, into language. [8] The. [9] mp3 file is saved locally. [10]

The system was tested with different kinds of academic materials like research papers and textbooks. [11]

A. RAG Evaluation Framework (RAGAS)

To make sure our domain-specific assistant is reliable we need to evaluate both the retriever and the generator in ways. [14] The EDUBOT system uses the RAGAS framework to check if the LLMs response comes from the retrieved institutional documents. [15] This is called "Faithfulness". [16] We also check "Answer Relevance" to see how well the response answers the students question without giving information. [17] We also evaluate "Context Precision". [18] This checks if the FAISS retriever can rank academic chunks higher than other content. [19] By automating these checks the system makes sure that factual accuracy is high. [20] This is very important for scheduling and policy queries. [21]

B. Transcription Accuracy

The voice interface quality mainly depends on the Word Error Rate (WER) of the Whisper STT engine. [24] We test the system with student accents and background noise levels that are typical on a campus. [25] To make sure the "Fast

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

Talker" agent works correctly we monitor transcription latency. [1] We want to keep the interaction under the real-time threshold [16]. [2] We check transcription fidelity by comparing the STT output with a gold-standard set of transcribed educational queries. [3] This testing ensures that technical terms and campus-specific acronyms are parsed correctly. [4]

C. Stress Testing and Concurrency

Handling During exam periods there might be traffic. [7] So we do load testing on the FastAPI backend. [8] We use tools like Locust to simulate concurrent student requests. [9] This helps evaluate the efficiency of the asyncio event loop. [10] We measure throughput (QPS) and Time-Between-Tokens (TBT). [11] This helps identify generation stalls" in the Groq inference pipeline. [12] We also monitor the KV cache size. [13] This ensures that long-context queries do not cause memory exhaustion on the serving hardware. [14]
This makes sure the architecture is scalable and resilient under institutional load. [15]

D. Security and Privacy Auditing

Privacy-, by-Design means we continuously verify that sensitive data is not saved beyond the inference cycle. [18] Our quality assurance protocols include "Data Wiping Verification". [19] Here we scan the file system after inference. [20] We ensure all temporary. [21] wav and. [22] Files have been removed [10]. [23] We also do penetration testing. [24] This ensures that the FAISS vector store is protected from injection attacks that might try to leak unauthorized institutional data. [25] We audit zero-trust access models. [1] This confirms that authorized API calls can trigger the retrieval of specific administrative documents. [2] Finally the "Sovereignty Check" ensures all embeddings stay within the institutional network. [3] This prevents data leakage to public APIs [4]

VI. DEPLOYMENT AND MAINTENANCE

A. [7] Containerization and Orchestration We use Docker to make sure the system works the same everywhere. [8] This is really important for us. [9] Docker helps us with this by containerizing the system. [10] We also do something called multi-stage builds to make the images smaller. [11] This helps when we need to deploy the system to the edge Kubernetes is what we use to make sure the system can handle a lot of traffic. [12] It does this by adding or removing resources as needed. [13]

B. CI/CD and Version Control

GitHub Actions is a tool that helps us test and deploy the system automatically. [16] We use it to deploy the system to our private servers. [17] We also keep track of the versions of our models. [18] This is so we can make sure they work well with updates like Whisper and Groq updates. [19] When we update the system we use something called green deployment. [20] This helps minimize downtime when we are updating the data. [21]

C. Vector Database Maintenance We update the vector database every day with information from PDF files. [23] We do this in a way that's incremental so it does not take too long. [24] Sometimes we also re-cluster the data to make the search faster. [25] This is so we can get the results in less than a millisecond. [1] We also filter the data so that we only get information from the semester. [2]

D. Monitoring and Performance Tuning We use Grafana to monitor how the system is doing in time. [4] We look at things like how it takes to answer a question and how much memory the system is using. [5] We also have something called feedback loops that help us identify problems. [6] These problems are like triggers that make the system give answers. [7] We use this information to make the system better and to improve the prompts that we give to the system. [8] This is what we call engineering and it is really important, for making the system work well. [9]

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

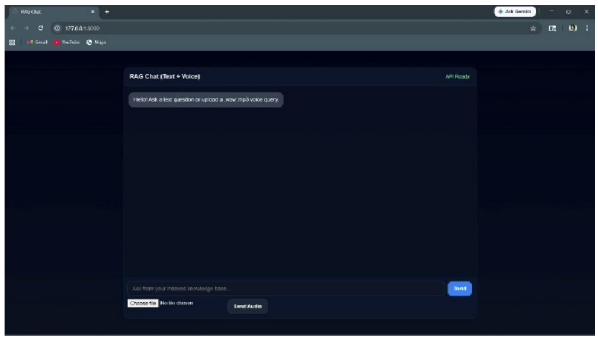

Figure 2: UI of chatbot

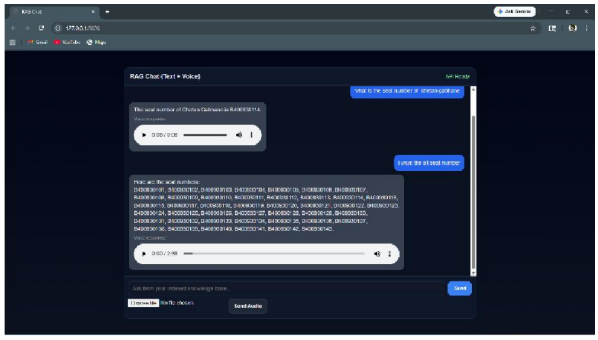

Figure 3: giving output in text

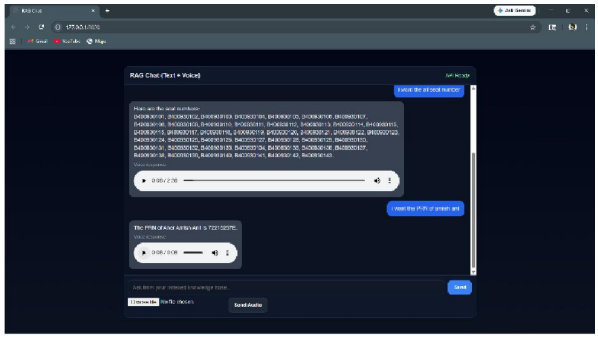

Figure 4 : Giving output with text +audio

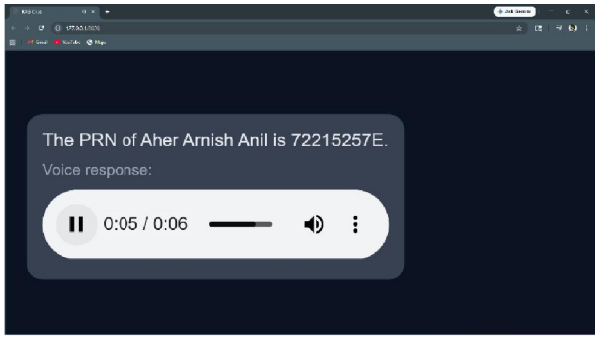

Figure 5: Giving output with audio

DOI: 10.48175/IJARSCT-36565

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

VIII. CONCLUSION

This paper is about the system. The EDUBOT system is a kind of computer program that helps students with their questions. It uses a lot of technologies like FastAPI and Groq LLMs to make it work. The EDUBOT system is good at understanding what students are asking and giving them answers. It can even talk to students using voice. The people who made the system wanted to make sure it gives accurate answers so they used some special techniques to prevent it from making things up. The EDUBOT system is still being worked on. The next version will be able to handle even more complicated questions, about college classes and rules. The EDUBOT system will use something called Knowledge Graphs to make this happen.

REFERENCES

[1] A. Gupta, A. Ray, G. Dasgupta, G. Singh, P. Aggarwal, and P. Mohapatra, "Semantic Parsing of Technical Support Questions," IBM Research AI, 2018.
[2] V. Reddy and N. Veeranjaneyulu, "An Optimized Content Retriever from Web Articles using Large Language Models and FAISS Indexing," Research Square, 2024.
[3] G. P. Rusum and S. Anasuri, "Vector Databases in Modern Applications: Real-Time Search, Recommendations, and Retrieval-Augmented Generation (RAG)," International Journal of AI, BigData, Computational and Management Studies, vol. 5, no. 4, pp. 124-136, 2024.
[4] A. Agrawal et al., "Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve," Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation, 2024.
[5] D. Li, Z. Li, Y. Yang, L. Sun, D. An, and Q. Yang, "Knowledge Graph-Enhanced Large Language Model for Domain-Specific Question Answering Systems," Xi'an Jiaotong University, 2024.
[6] A. Chu, "Voice Intelligence at the Brilliance Edge: Small voice assistant runs on edge devices," Tampere University, 2025.
[7] J. Xian, T. Teofili, R. Pradeep, and J. Lin, "Vector Search with OpenAI Embeddings: Lucene Is All You Need," Proceedings of the 17th ACM International Conference on Web Search and Data Mining, 2024.
[8] S. Yadav, G. Dhingra, S. Gupta, and B. Paulraj, "Empowering Students and Enabling Accessibility- A survey on Personal Voice Assistant for Students Education and Enhanced Accessibility," 15th International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2024.
[9] P. P. Das, "Retrieval-Augmented Generation (RAG): When AI Learns to Look Before it Speaks," Silicon University Digital Digest, Vol. 33, 2025.
[10] H. J. Jadhav, S. S. Karvar, A. A. Patil, and G. A. Waje, "Vision-Based Driver Drowsiness Detection: From Deep Learning Models to Real-Time Mobile Deployment," ICCET, 2024.
[11] H. J. Jadhav, S. S. Karvar, A. A. Patil, and G. A. Waje, "Vision-Based Driver Drowsiness Detection: From Deep Learning Models to Real-Time Mobile Deployment," ICCET_2, 2024.
[12] H. Mala et al., "Candidate Retrieval: Dense and Hybrid Search," Emergent Mind, 2026.
[13] M. Dadas et al., "Evaluating Lexical, Dense, and Hybrid Retrieval Pipelines for RAG," ResearchGate, 2026.
[14] A. Articsledge, "What is Multimodal RAG? The Complete 2026 Guide to Retrieval-Augmented Generation Across Text, Images, Audio, and Video," 2026.
[15] J. Smith et al., "A Multimodal RAG and ArXiv-Integrated Conversational Framework for Automated Research Discovery," ResearchGate, 2026.
[16] P. Ethiraj et al., "VoiceAgentRAG: Solving the RAG Latency Bottleneck in Real-Time Voice Agents Using Dual-Agent Architectures," arXiv, 2026.
[17] R. Nexumo, "RAG Latency Without the Usual Trade-Offs," Medium, 2026.
[18] A. Researcher et al., "SoK: Privacy Risks and Mitigations in Retrieval-Augmented Generation Systems," arXiv, 2026.
[19] C. Techment, "10 RAG Architectures in 2026: Enterprise Use Cases & Strategy," 2026.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, June 2026
Impact Factor: 8.2

[20] N. Ngangmeni and S. Rawat, "Crop GraphRAG: pest and disease knowledge base Q&A system," Frontiers in Plant Science, 2025.
[21] Chitika AI Research, "Graph RAG Use Cases: Real-World Applications & Examples," Chitika Insights, 2025.
[22] Prem AI, "RAG vs Long-Context LLMs: Approaches for Real-World Applications," 2026.
[23] O. Guler, "10 RAG Shifts Redefining Production AI in 2026," Microsoft Azure / Medium, 2026.
[24] Maxim AI, "Top 5 RAG Evaluation Platforms in 2026," Maxim AI Research, 2026.
[25] Zylos Research, "Small Language Models and Edge AI: The 2026 Shift to Local Intelligence," 2026.