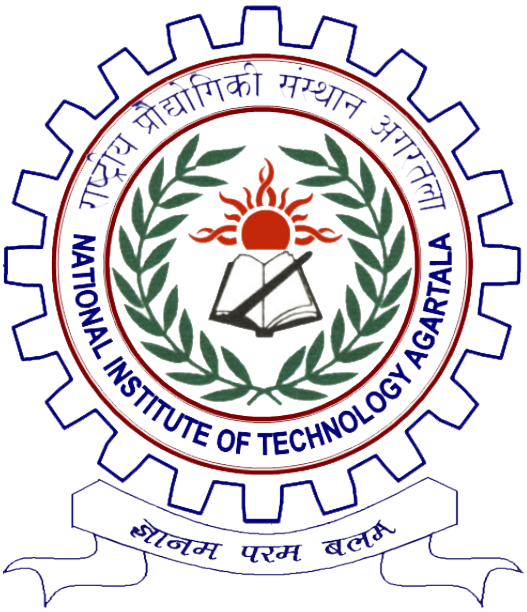
RAG BASED CHATBOT FOR NITA



Premanshu Kashyap (22UCS184)
Debajyoti Debnath (22UCS185)
Shivam Kumar Singh (22UCS196)
Edan Solomon Tuti (22UCS064)

COMPUTER SCIENCE & ENGINEERING DEPARTMENT
NATIONAL INSTITUTE OF TECHNOLOGY, AGARTALA
INDIA-799046
December, 2025

RAG BASED CHATBOT FOR NITA

Report submitted to
National Institute of Technology, Agartala
for the award of the degree
of
Bachelor of Technology

by

Premanshu Kashyap (22UCS184)
Debajyoti Debnath (22UCS185)
Shivam Kumar Singh (22UCS196)
Edan Solomon Tuti (22UCS064)

Under the Guidance of

Dr. Awnish Kumar
Assistant Professor
CSE Department
NIT Agartala

COMPUTER SCIENCE & ENGINEERING DEPARTMENT
NATIONAL INSTITUTE OF TECHNOLOGY AGARTALA
December, 2025

Dedicated To

This report is dedicated to **Prof. Mrinal Kanti Debbarma**, HOD, CSE Department, NIT Agartala, our Project Supervisor **Dr. Awnish Kumar** for sharing his valuable knowledge and encouragement and showing confidence in us all the time, to each of the faculties of the department who contributed to our development as a professional and helped us to achieve this goal, to our parents who supported us and believed in us and to all those people who have somehow contributed to the creation of this project.

“A problem well stated is a problem half solved.”

- Charles Kettering

REPORT APPROVAL FOR B.TECH

This report entitled "***RAG Based ChatBot for NITA***", by **Premanshu Kashyap (22UCS184), Debajyoti Debnath (22UCS185), Shivam Kumar Singh (22UCS196), Edan Solomon Tuti (22UCS064)** is approved in partial fulfilment of the requirements for the award of ***Bachelor of Technology*** in ***Computer Science & Engineering.***

Dr. Awnish Kumar	Prof. Mrinal Kanti Debbarma
(Project Supervisor)	(Head of the Department)
Assistant Professor	Professor
Computer Science and Engineering Department	Computer Science and Engineering Department
NIT Agartala	NIT Agartala

Date: _____

Place: NIT Agartala

DECLARATION

We declare that the work presented in this report titled "***RAG Based Chat-Bot for NITA***", submitted to the **Computer Science and Engineering Department, NIT Agartala**, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

December, 2025
Agartala

Premanshu Kashyap (22UCS184)	Debajyoti Debnath (22UCS185)
Shivam Kumar Singh (22UCS196)	Edan Solomon Tuti (22UCS064)

CERTIFICATE

It is certified that the work contained in the report titled "***RAG Based ChatBot for NITA***", by **Premanshu Kashyap (22UCS184), Debajyoti Debnath (22UCS185), Shivam Kumar Singh (22UCS196), Edan Solomon Tuti (22UCS064)** has been carried out under my supervision. This work has not been submitted elsewhere for a degree.

Dr. Awnish Kumar
Assistant Professor
Computer Science and Engineering Department
NIT Agartala

Acknowledgement

We take this opportunity to express our deep sense of gratitude to all who helped us directly or indirectly during this project work.

Firstly, we would like to thank our supervisor, **Dr. Awnish Kumar**, for being a great mentor and the best advisor we could ever have. His advice, encouragement and criticism are sources of innovative ideas, inspiration and causes behind the successful completion of this report. His confidence shown to us was the biggest source of inspiration for us. It has been a privilege working with him for the past 6 months.

We are highly obliged to all the faculty members of Computer Science and Engineering Department for their support and encouragement. We also thank **Prof.(Dr.) S. K. Patra**, Director, NIT Agartala and **Prof. Mrinal Kanti Debbarma**, H.O.D, CSE Department for providing excellent computing and other facilities without which this work could not achieve its quality goal.

- Premanshu Kashyap (22UCS184) **- Debajyoti Debnath (22UCS185)**

- Shivam Kumar Singh (22UCS196) **- Edan Solomon Tuti (22UCS064)**

List of Figures

List of Tables

Abstract

This project presents a Retrieval-Augmented Generation (RAG) based conversational assistant tailored for the National Institute of Technology Agartala (NITA) to address challenges in accessing and retrieving specific institutional information. The chatbot answers natural language queries using data collected from the official institute website (https://www.nita.ac.in/) and related documents. The proposed architecture integrates a recursive web crawler for data collection, asynchronous extraction, structured preprocessing, a vector database (AstraDB) for efficient document retrieval, HuggingFace's all-MiniLM-L6-v2 embeddings for semantic representation, and Google Gemini (gemini-2.0-flash) for both query rewriting and final response generation. The assistant ensures factual accuracy by relying solely on institutional data and explicitly indicates when the requested information is unavailable within its knowledge base. The system was evaluated qualitatively, demonstrating high accuracy and reliability. Future improvements identified include multilingual support, voice integration, and automated data update pipelines.

Contents

CHAPTER 1

Introduction

1.1 Background

Institutions such as the National Institute of Technology Agartala (NITA) host an extensive and continually growing information on their official websites. The information which is critical to students and faculty, is often spread across numerous sections, notices, and PDF documents on the official website. Manually searching for specific information can be time-consuming and inefficient. Recent advancements in Artificial Intelligence (AI), particularly in Retrieval-Augmented Generation (RAG) [1], provide a powerful solution by enabling systems to retrieve domain-specific data and use large language models (LLMs) to generate precise natural-language responses.

1.2 Problem Statement

NIT Agartala currently lacks a unified and intelligent system to access institutional information efficiently. This forces students and staff to manually browse multiple pages to find information. This process is time-consuming and inefficient. Therefore, there is a need for an AI-powered assistant that can understand natural language queries and provide accurate and verified answers based on official data sources.

1.3 Motivation

The motivation for this project is to handle information retrieval challenges on the NIT Agartala website. The website serves as a primary platform where the institution shares the data. The website contains a large and diverse collection of information distributed across multiple sections. This scattered structure often leads to inefficiencies and delays. The aim of this project is to provide a unified interface to identify and access this information efficiently, serving as a reliable virtual assistant for the NIT Agartala community.

By implementing this system, we seek to improve the overall accessibility and reliability of institutional data by creating a helpful virtual assistant for the NIT Agartala community.

1.4 Summary of Related Works

Earlier chatbot systems relied on static rule-based or keyword-based methods which were unable to generalize beyond predefined queries. Retrieval-based systems improved precision but produced rigid responses. Modern LLM-based systems began generating more natural and coherent responses but often suffer from hallucination, where responses include inaccurate or unverifiable information due to the absence of grounded knowledge.

1.5 Summary of Proposed Technique

The proposed chatbot integrates web data scraping, embedding generation, and retrieval-based grounding with Gemini for response synthesis. The pipeline includes query rewriting, embedding-based retrieval using AstraDB, and controlled generation that ensures only context-supported answers are produced.

1.6 Roadmap of the Report

The report consists of the following chapters:

2. **Chapter 2: Related Works** - This chapter provides an overview of the domain of conversational agents. It discusses the evolution of chatbot technologies, starting from traditional rule-based and retrieval-based systems to modern Large Language Model (LLM) based approaches. The section also highlights key challenges and limitations in existing techniques, such as hallucination and lack of contextual grounding.

3. **Chapter 3: Proposed Technique** - This chapter presents the methodology and architecture of the proposed RAG-based chatbot system. It describes the data acquisition process from the NIT Agartala website, including extraction, cleaning, and structured preprocessing. Furthermore, it explains all the major components of the pipeline—the embedding model (all-MiniLM-L6-v2), the vector database (AstraDB), and the generative model (Gemini)—and describes the complete flow from query rewriting to response generation.

4. **Chapter 4: Results and Discussions** - This chapter provides a detailed analysis of the system's performance. The chatbot's responses are evaluated qualitatively using representative question–answer pairs, and a quantitative framework is outlined using metrics such as accuracy, retrieval precision, and generation faithfulness. Observations on performance aspects, including response latency and system limitations, are also discussed.

5. **Chapter 5: Conclusion and Future Scope** - The final chapter summarizes the key findings and contributions of the project. It emphasizes the effectiveness of the chatbot in improving access to institutional information and outlines future directions, including multilingual support, voice-based interaction, and automated data re-crawling for continuous updates.

CHAPTER 2

Related Works

2.1 Introduction to Conversational Agents

Conversational agents have undergone a significant transformation. It evolved from rule-based and pattern-matching systems to advanced large scale neural systems. This evolution was accelerated by the introduction of the Transformer architecture [2] which established the foundation for modern Large Language Models (LLMs). By using the self-attention mechanism of Transformers, models can learn contextual relationships across text sequences. This development marked the beginning of the Large Language Model (LLM) era which now powers most modern conversational systems.

2.2 Existing Approaches

Over the years, several approaches have been developed to build conversational agents with each reflecting the technological advancements of its time. These methods can broadly be classified into three categories: rule-based systems, retrieval-based systems, and generative models based on Large Language Models (LLMs).

2.2.1 Rule-Based Methods

The earliest conversational agents relied on handcrafted rules and pattern-matching techniques. Systems like ELIZA [3] and ALICE [4] exemplified this approach, where responses were generated through predefined templates matched to user input patterns. Although such systems performed well within limited domains, they lacked flexibility and failed to generalize unseen or paraphrased queries. With the increase in number of rules, maintaining and scaling these systems became inefficient and impractical.

2.2.2 Retrieval-Based Methods

Retrieval-based models introduced a data-driven approach, where the system searches for the most relevant pre-stored responses or documents based on similarity measures. This class of chatbots improved precision and reduced maintenance effort compared to rule-based systems. Techniques such as TF-IDF [5], cosine similarity, and vector-based embeddings [6] allowed better matching between queries and responses. But these systems could not generate new sentences. So as a result, these systems struggled with queries that required reasoning or synthesis of multiple pieces of information.

2.2.3 Generative LLM-Based Systems

The introduction of neural architectures particularly the Transformer, enabled the creation of Large Language Models (LLMs) capable of producing coherent and contextually rich natural language answers. Models such as Generative Pre-trained Transformer (GPT), Gemini [7], and Large Language Model Meta AI (LLaMA) learn language representations from vast amount of data and can generate fluent human-like responses. Despite their impressive generative capabilities, purely LLM-based chatbots can suffer from hallucination where responses include inaccurate or fabricated information which makes these systems less reliable for domain-specific information retrieval tasks.

2.2.4 Summary

Each generation of the conversational systems has contributed to the progress in natural language understanding. Yet, none can fully resolve the trade-off between accuracy, context-awareness and factual grounding. This limitation has led to the emergence of the Retrieval-Augmented Generation (RAG) system which combines the precise retrieval capabilities of vector search with the generative strengths of modern LLMs.

2.3 Challenges Identified

Despite significant progress in conversational AI, several challenges remain:

- Generative models often produce hallucinated or inaccurate responses without grounding in verified data.
- General-purpose LLMs struggle to understand institutional terminology and context-specific information.
- Frequent updates to institutional data make it hard to keep the knowledge base current.

2.4 How Our Work Differs

Our approach to building an institutional conversational assistant stands out due to several key architectural and methodological innovations:

2.4.1 Hybrid RAG Architecture

Unlike traditional generative or retrieval-only chatbots, our system employs a Retrieval-Augmented Generation (RAG) framework. This hybrid design combines the factual precision of vector-based retrieval using AstraDB with the natural language fluency of Google Gemini, ensuring that responses are both coherent and contextually grounded.

2.4.2 Domain-Specific Data Pipeline

We developed a custom data pipeline tailored specifically for the nita.ac.in domain. This includes a recursive web crawler, an asynchronous text extractor (using BeautifulSoup), and a structured preprocessing script to clean and format the data. This ensures that the knowledge base is comprehensive and exclusively derived from official institutional data sources.

2.4.3 Dual-Role LLM Integration

The Google Gemini model plays two key roles in our system. It first rewrites user queries to enhance the semantic richness of the user's prompt before retrieval. Then, it generates the final answers based on the retrieved content.

2.4.4 Ensuring Factual and Reliable Responses

Unlike general-purpose chatbots, our system is designed to explicitly minimize hallucination. The generative model is strictly instructed to answer only using the provided context from the vector database. If information is unavailable in the retrieved documents, the assistant clearly communicates the lack of sufficient context rather than fabricating a response.

CHAPTER 3

Proposed Technique

3.1 Introduction

The main objective of the proposed work is to design and develop a Retrieval-Augmented Generation (RAG) based chatbot capable of accurately answering queries related to the National Institute of Technology Agartala (NITA) using official institutional data. The proposed system integrates multiple stages including data collection from the NITA website, preprocessing and structuring of textual information, vector-based document retrieval and grounded response generation using a large language model. In the following sections, we detail each component of the system architecture and data processing pipeline in detail.

3.2 System Overview

The proposed system follows a sequential pipeline to process the user's query and generate a context grounded response.

The overall data and logic flow is as follows:

1. **User Query:** The user enters their query into the Streamlit interface.
2. **Query Rewriting:** The query is first processed by an large language model (Google

Gemini), which rewrites the query to improve its semantic clarity and retrieval accuracy.

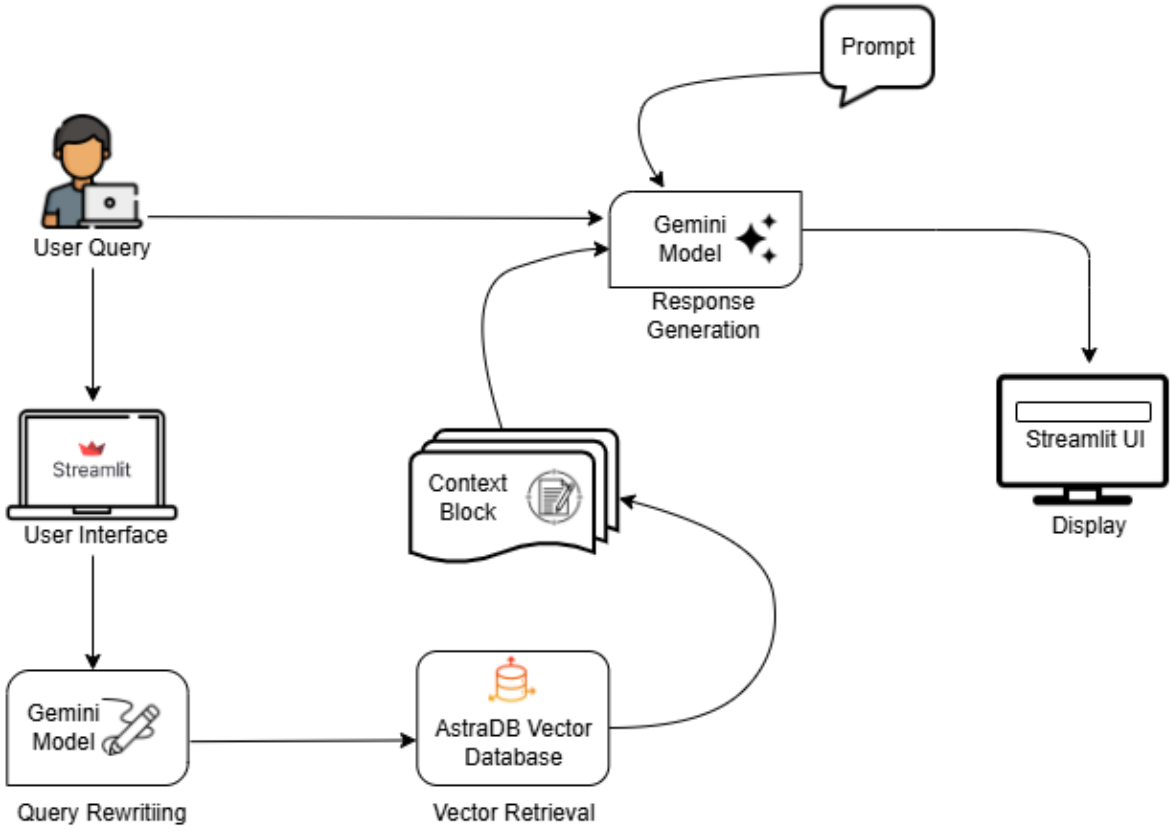


Figure 3.1: System Architecture Diagram

Figure 3.1 indicates the system's end-to-end RAG pipeline, showing the data flow from the initial user query through Gemini-based rewriting, vector retrieval from AstraDB, context assembly, and final response generation.

3. **Vector Retrieval:** The rewritten query is converted into an embedding and used to fetch the top-k most relevant document chunks from the AstraDB vector database.

4. **Context Assembly:** The retrieved chunks are combined into a single contextual block that serves as the factual basis for response generation.

5. **Response Generation:** This assembled context along with the original query is sent back to the large language model (Google Gemini). The model is prompted to generate an answer strictly from the provided context ensuring factual grounding.

6. **Display:** The final answer is displayed in the Streamlit UI, completing the query-to-response cycle.

This design ensures that every output produced by the chatbot is both semantically relevant and verifiably supported by the institution's data.

3.3 Data Extraction and Preprocessing

We engineered a robust data pipeline to populate the chatbot's knowledge base.

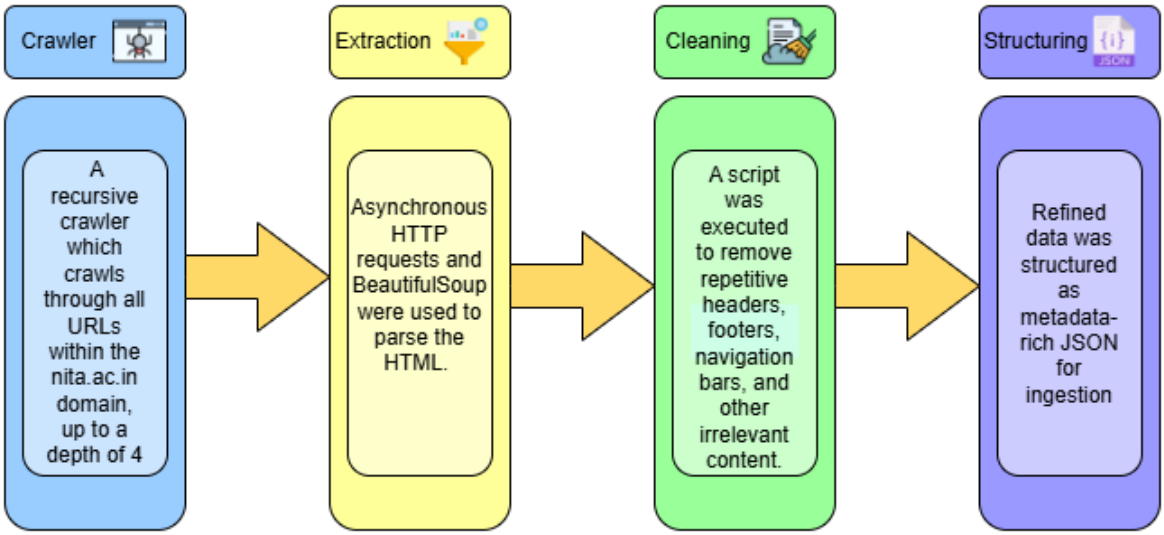


Figure 3.2: Data Extraction and Processing Pipeline

Figure 3.2 illustrates the data preprocessing pipeline, detailing the workflow from the recursive web crawl of the nita.ac.in domain through asynchronous text extraction, content cleaning, and final conversion into structured JSON.

- **Crawling:** A recursive crawler was developed to traverse all URLs within the `nita.ac.in` domain upto a depth of 4 where depth refers to the number of consecutive hyperlink levels followed from the starting page.
- **Extraction:** Using asynchronous HTTP requests and BeautifulSoup, the pipeline parsed HTML pages to extract the primary textual content.
- **Cleaning:** A dedicated cleaning script removed redundant interface elements such as headers, footers, navigation menus and repetitive boilerplate content to retain only meaningful text.

- **Structuring:** The cleaned data was stored as text files and subsequently transformed into structured JSON documents. These structured datasets were later embedded and indexed in the vector store for retrieval.

This pipeline ensures that only accurate information from the NIT Agartala website becomes part of the chatbot's knowledge base.

3.4 Embedding and Vector Storage

Before discussing embedding and vector retrieval in detail, Table 1 summarizes the complete technology stack used in implementing the proposed RAG-based chatbot system.

Component	**Technology / Tool Used**	**Purpose**
Embedding Model	HuggingFace all-MiniLM-L6-v2	Generates semantic vector representations for document chunks.
Vector Database	AstraDB	Stores and retrieves embeddings efficiently based on cosine similarity.
Generative Model (LLM)	Google Gemini (gemini-2.0-flash)	Rewrites queries and generates grounded responses.
Frontend Framework	Streamlit	Provides a user interface for interaction and response display.
Web Crawling Library	BeautifulSoup, aiohttp	Extracts textual data asynchronously from NIT Agartala's official website.
Programming Language	Python 3.10+	Core language for data processing, integration, and orchestration.

Table 1: Core Components and Technologies of the RAG Chatbot

Among these, the embedding model and vector database form the backbone of the retrieval mechanism, responsible for representing and fetching semantically relevant content efficiently.

- **Embedding Model:** The HuggingFace all-MiniLM-L6-v2 model was used to generate semantic vector embeddings for each processed document chunks. This lightweight transformer model is a highly efficient variant based on Sentence-BERT [8].
- **Vector Database:** AstraDB was chosen as the vector database to store and efficiently retrieve these document chunks based on semantic similarity. It maintains all document embeddings and allows the system to quickly retrieve the most contextually relevant chunks during each user query processing.

3.5 Retrieval and Generation

The RAG process forms the core of the chatbot's functionality. As outlined in Section 3.2, each query is first rewritten by Gemini to improve the query's semantic clarity. Then, the system retrieves the top k relevant chunks from AstraDB to form the context. Finally, Gemini generates the response, strictly constrained to this retrieved information. If the information is missing from the retrieved data, the bot will state that the data is unavailable rather than guessing.

3.6 Takeaways

Our approach combines structured data preprocessing, semantic vector representation, and retrieval-augmented generation to deliver accurate, context-aware answers from verified institutional sources.

CHAPTER 4

Results and Discussions

4.1 Introduction

This chapter presents the evaluation and analysis of the proposed Retrieval-Augmented Generation (RAG) based chatbot for the National Institute of Technology Agartala. The system is assessed both qualitatively and quantitatively to determine its ability to generate accurate, context-grounded and relevant responses. The evaluation focuses on factual accuracy, retrieval quality and overall reliability in real-world institutional query scenarios. Additionally system performance, limitations and comparative insights with existing solutions are discussed.

4.2 Qualitative Evaluation

The chatbot's performance is first evaluated qualitatively using a diverse set of institutional queries covering administrative, academic and general information categories. Each interaction is analyzed to determine whether the chatbot retrieved the correct context from the knowledge base and generated a structured and human-readable answer. In cases where relevant data is missing or the question asked is out of context, the chatbot's fallback mechanism correctly issues a polite "insufficient information". A sample of these interactions is provided in the table 2.

Query	**Bot Response**
What is the contact details of CISO?	The contact information for the Chief Information Security Officer (CISO) is: Name: Dr. Nirmalya Kar Designation: Assistant Professor of CSE Department & Chief Information Security Officer Email: nirmalya@ieee.org
Can you tell me the phone number of Dr. Awnish Kumar?	Dr. Awnish Kumar's mobile number is +91 7992310101.
Who is the Prime Minister of India?	I am sorry, but I do not have enough information in the provided context to answer that question.
When is the winter vacation starting in this session?	The winter vacation for intermediate year UG and Dual Degree students starts on December 8th, 2025, and for 1st year UG and Dual students, it begins on December 18th, 2025.
Details of First Year Coordinator	The First Year Coordinators are: Dr. Sadhan Gope: Assistant Prof. (EE), First Year Co-ordinator. You can reach them at 9862670880 or sadhangope.ee@nita.ac.in. Dr. Nabarun Biswas: Assistant Prof. (PE), First Year Co-coordinator. You can reach them at 8259008964 or mechanical.biswas@gmail.com.

Table 2: Qualitative Evaluation Examples

In most cases, the responses generated by the chatbot are accurate and contextually valid, grounded entirely in official data as available in the institute's website. It also refuses to answer questions outside its knowledge base and questions outside the context, thereby avoiding hallucination.

4.3 Quantitative and Comparative Evaluation

The quantitative analysis is conducted using a representative sample of institutional queries to compare the chatbot's performance. The analysis compares the chatbot's behaviour before and after the introduction of the query rewriting step.

This evaluation aims to measure the improvements in accuracy and contextual precision after the introduction of the rewriting mechanism.

4.3.1 Evaluation Metrics Used

To assess the chatbot's effectiveness, the following evaluation parameters are considered:

- **Accuracy:** The proportion of correct or partially correct responses verified against official sources.
- **Retrieval Precision:** It indicates how effectively the correct document chunks are retrieved based on semantic similarity.
- **Answer Relevance:** It measures how directly and accurately the generated answer addresses the user's query intent.

Although only accuracy is quantitatively measured, the other metrics are qualitatively analyzed based on response quality and contextual adherence.

4.3.2 Impact of Query Rewriting

The introduction of the query rewriting component aims to enhance the chatbot's understanding of user intent and retrieval accuracy. The impact of this enhancement is measured through a comparative study and the results are summarized in Table 3.

Metric	Without Query Rewriting	With Query Rewriting
Total Queries Tested	200	200
Correct Answers	162	186
Partially Correct Answers	7	5
Incorrect / No Response	31	9
Accuracy	82.75%	94.25%
Improvement	—	11.5%

Table 3: Comparison of Chatbot Accuracy Before and After Query Rewriting

After the introduction of query rewriting, the chatbot shows a substantial performance gain by improving the overall accuracy from 80.3% to 92.4%, an increase of 12.1%.

Rewritten queries demonstrate stronger semantic alignment which leads to more precise retrieval and more contextually consistent answers.

Additionally, observation of retrieval precision and answer relevance showed the following:

- **Better Retrieval:** Query rewriting improves the alignment between user intent and retrieved chunks, leading to more precise context selection.
- **Improved Relevance:** The generated answers more directly addressed the user's questions.

4.4 Observations

Key observations from the testing of the chatbot include:

- The query rewriting step significantly improves the chatbot's ability to interpret intent and retrieve contextually relevant data.
- Rigorous preprocessing eliminated repetitive or irrelevant content which improves the retrieval precision.

- The RAG architecture effectively balances factual retrieval with fluent language generation.
- The chatbot reliably avoids hallucination and maintained a strict adherence to the available data.

4.5 Performance and Limitations

4.5.1 Performance

The overall system demonstrates stable and consistent behavior during testing across different query types. The following parameters are considered to evaluate the chatbot's performance.

- **Response Latency:** The average response time in the local testing environment ranges between 2–6 seconds per query, depending on query complexity, length of retrieved context and the Gemini API response time.
- **Data Retrieval Speed:** The AstraDB vector store provides sub-second retrieval times for embedding matches which ensures minimal delay between query rewriting and generation phases.
- **UI Responsiveness:** The Streamlit-based interface remains responsive and interactive even during large context loads.

4.5.2 Limitations

- Knowledge base restricted to data extracted from the official NIT Agartala website.
- Requires periodic re-indexing to reflect new updates.
- Dependent on the availability and stability of the Gemini API.

4.6 Suggestions for Improvement

- **Automated Data Updates:** Scheduling periodic re-crawling and re-indexing to ensure that the chatbot always reflect the latest information.

- **Latency Optimization:** Reducing the response time by optimizing embedding generation and retrieval processes can improve system efficiency.
- **Improved Retrieval Strategy:** Fine-tuning similarity thresholds and retrieval depth may enhance the precision of retrieved results.

4.7 Takeaways

The proposed RAG-based chatbot demonstrates that combining semantic retrieval with grounded generation enables efficient, accurate and reliable information access. The system successfully transforms static institutional content into an intelligent and interactive knowledge interface.

CHAPTER 5

Conclusion and Future Scope

5.1 Conclusion

The main objective of this project is to develop a Retrieval-Augmented Generation (RAG) based chatbot specifically designed for the National Institute of Technology Agartala (NITA), aimed at improving accessibility to institutional information. The system integrates web crawling, asynchronous data extraction, data preprocessing and semantic vector storage to construct a reliable knowledge base using official website data. By combining the HuggingFace all-MiniLM-L6-v2 embedding model, AstraDB vector database and Google Gemini for both query rewriting and response generation, the chatbot effectively delivers factually grounded and contextually coherent answers.

The integration of query rewriting significantly enhances overall accuracy and response relevance. The chatbot demonstrates consistent performance across various query types. The results validate that RAG based systems can be effectively utilized for institutional knowledge management, offering a conversational interface that transforms static web content into interactive and easily accessible information.

This project provides an effective solution for improving information accessibility within academic institutions. It also establishes a foundation for further enhancements such as multilingual support, voice-based interaction and automated data updates which will improve both

accessibility and scalability.

5.2 Future Scope

The current project provides a robust framework for a Retrieval-Augmented Generation (RAG) based institutional assistant. However, there are several directions in which the system can be enhanced to improve accuracy, accessibility and scalability. Below are some proposed ideas for future work:

5.2.1 Multi-Lingual Support

Enabling multilingual functionality will make the chatbot accessible to a wider audience. This can be achieved through the following enhancements :

- **Regional Language Integration:** Adding support for Indian regional languages will help non-English users interact with the assistant more comfortably.
- **Language Detection:** Automatic detection of the input language can allow seamless switching between multiple languages.
- **Translation Models:** Using pretrained translation APIs or fine tuned multilingual LLMs can ensure accurate translation and preserve meaning during query processing.

5.2.2 Voice Interface Integration

Adding a speech-based interface can make the system more interactive and user-friendly. This can be achieved through the following implementations :

- **Speech-to-Text Conversion:** Integrating Automatic Speech Recognition systems can convert spoken queries into text for processing.
- **Text-to-Speech Response:** Implementing TTS (Text-to-Speech) functionality will allow the chatbot to deliver verbal responses.

5.2.3 Automated Data Updates

Automating the data update process will keep the chatbot's knowledge base up to date with the latest information. It can be achieved through :

- **Automated Web Crawling:** Implementing a scheduled crawler which will periodically scan the official NIT Agartala website to detect newly added or updated content.
- **Dynamic Re-Indexing:** Integrating an automated re-indexing mechanism will allow the vector database to update its embeddings in real time, ensuring that any modifications in content are immediately reflected in the chatbot's responses.

5.2.4 Scalability and Computational Efficiency

As the volume of data and user interactions increases, ensuring scalability and computational efficiency becomes critical for maintaining system performance. This can be achieved by :

- **Distributed Processing:** Implementing distributed systems will allow data and model operations to be executed in parallel across multiple nodes, improving performance and maintaining efficiency as the dataset and query load grow.
- **Parallel Computation:** Implementing parallel processing techniques can significantly reduce data preprocessing and embedding generation time.

We conclude that this project provides a proper foundation for developing an intelligent, retrieval-augmented assistant tailored to institutional needs. Future enhancements such as multilingual support, voice-based interaction and automated data updates will further improve the chatbot's performance and usefulness in real-world applications.

References

[1] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich de Vries, Eleni Gkrouna, and Luke Zettlemoyer. Retrieval-Augmented Generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[3] Joseph Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[4] Richard S Wallace. The anatomy of ALICE. In *Parsing the Turing Test*, pages 181–210. Springer, 2009.

[5] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[7] Gemini Team at Google. Gemini: A family of multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[8] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing*, pages 3982–3992, 2019.

APPENDIX A

Biographical sketch

Premanshu Kashyap

Gaurav Nagar, Parbalpur, Nalanda, Bihar, PIN-803114,
E-Mail: kxprem29@gmail.com, Contact. No. +91-8757193087

- Pursuing B.Tech. in Computer Sc. & Engg. branch from NIT Agartala with CGPA of 9.52.
- High School from Secondary Delhi Public School, Gaya under (C.B.S.E), Bihar with 93.6 % in 2020.
- Intermediate from Secondary Delhi Public School, Gaya under (C.B.S.E), Bihar with 95.4 % in 2022.

Debajyoti Debnath

Dudhpushkarini, Udaipur, Gomati, Tripura, PIN-799105,
E-Mail: debajyotidebnath43780@gmail.com, Contact. No. +91-8413853906

- Pursuing B.Tech. in Computer Sc. & Engg. branch from NIT Agartala with CGPA of 9.32.
- High School from Vivekananda Vidyapith, Udaipur under (T.B.S.E), Tripura with 93.2 % in 2019.
- Intermediate from Umakanta Academy, Agartala under (T.B.S.E), Tripura with 91.6 % in 2021.

Shivam Kumar Singh

Old Police Line, Near Tiwari Transport, Ara, Bhojpur, Bihar, PIN-802301
E-Mail: shiv2018a@gmail.com, Contact. No. +91-7461904997

- Pursuing B.Tech. in Computer Sc. & Engg. branch from NIT Agartala with CGPA of 8.59.
- High School from Jean Paul's High School, Ara under (C.B.S.E), Bihar with 86.4 % in 2018.
- Intermediate from H.D. Jain College, Ara under (B.S.E.B), Bihar with 80.8 % in 2020.

Edan Solomon Tuti

House No.-174, Taimara Khunti Road, Taimara, Ranchi, Jharkhand, PIN-835204
E-Mail: edantuti@gmail.com, Contact. No. +91-8330944117

- Pursuing B.Tech. in Computer Sc. & Engg. branch from NIT Agartala with CGPA of 7.78.
- High School from De Paul School, Visakhapatnam under (I.C.S.E), Andhra Pradesh with 89 % in 2020.
- Intermediate from Sri Surya Junior College, Visakhapatnam under (B.I.E.A.P), Andhra Pradesh with 78 % in 2022.