

“NLP-Based Intelligent Chatbot for Academic Query Resolution”

Shivam Parshuram Gupta¹, Mukund Suman Jha², Indresh Vidyasagar Kumar Gupta³, Dharmbeer Radheshyam Chauhan⁴, Amruta Sankhe⁵

¹ UG – Information Technology Engineering , Shree L.R Tiwari College of Engineering, Maharashtra

² UG – Information Technology Engineering , Shree L.R Tiwari College of Engineering, Maharashtra

³ UG – Information Technology Engineering , Shree L.R Tiwari College of Engineering, Maharashtra

⁴ UG – Information Technology Engineering , Shree L.R Tiwari College of Engineering, Maharashtra

⁵ Assistant Professor, Information Technology Engineering , Shree L.R Tiwari College of Engineering, Maharashtra

¹Shivam.p.gupta@slrtce.in, ²indreshkumar.v.gupta@slrtce.in, ³mukund.s.jha@slrtce.in,

⁴dharmbeer.r.chauhan@slrtce.in, ⁵amruta.sankhe@slrtce.in.

Abstract

With the growth of digital technology, students now expect quick and accurate access to academic information. Many students still face delays while getting details related to admissions, courses, fees, scholarships, examinations, and placements through traditional methods such as phone calls, emails, or office visits. To solve this problem, this project presents an AI-Powered Student Assistant developed for Rajasthan Technical Education.

The proposed system is an intelligent chatbot that uses Artificial Intelligence (AI), Natural Language Processing (NLP), and Retrieval-Augmented Generation (RAG) to understand student questions and provide meaningful responses in real time. The chatbot can answer common academic and administrative queries 24/7 through a user-friendly web platform.

The system is designed using React.js and Next.js for frontend development, Node.js for backend services, and a cloud database for storing records and user interactions. By automating repetitive queries, the platform reduces manual workload, saves student time, and improves communication efficiency.

This project aims to create a smart, scalable, and student-friendly digital assistant that improves access to technical education information across Rajasthan.

Keywords: AI Chatbot, NLP, Rajasthan Education, Student Support, RAG, Automation

1. INTRODUCTION

Technology has changed the way students learn, communicate, and access information. In the education sector, digital systems are becoming more important for improving student services and reducing manual work. One of the most useful innovations in recent years is the AI chatbot, which can answer user questions automatically and instantly.

Rajasthan has many engineering and polytechnic students who often need information related to admissions, counseling, courses, fee structure, scholarships, placements, and exam schedules. In many cases, students depend on websites, phone calls, emails, or physical visits to obtain this information. These methods may take time, create confusion, and increase workload for administrative staff.

To overcome these problems, an AI-based Student Assistant is proposed. This system acts as an

intelligent chatbot that provides students with quick and accurate responses through a web platform. It understands natural language questions asked by users and gives relevant answers in a simple format.

The chatbot uses NLP to understand the user query and Retrieval-Augmented Generation (RAG) to fetch accurate data from stored information sources. This helps in generating better and more useful responses.

The main goal of this project is to build a centralized smart assistant that improves communication between students and educational institutions. It saves time, increases accessibility, and supports students at any time of the day.

In addition to answering queries, the proposed system also helps students make better academic decisions by providing organized and updated information in one place. It can guide users during admission periods, help them compare colleges, understand eligibility criteria, and explore scholarship opportunities without depending on multiple sources.

2. PROPOSED ALGORITHM

The proposed project is a smart Student Assistant developed to help students of Rajasthan Technical Education get information quickly and easily. It is built in the form of an AI chatbot that can understand questions asked by users and provide instant replies. The system uses Artificial Intelligence, Natural Language Processing, Machine Learning, and Retrieval-Augmented Generation to improve the quality of answers. Its main purpose is to create a single platform where students can find academic and administrative details without depending on manual communication methods. Students can ask questions related to admission process, available courses, exam schedules, fees, scholarships, placements, and other college services. Instead of showing fixed or limited responses, the chatbot gives updated and relevant answers based on the user query. It studies the question, identifies what the student wants to know, and then provides a suitable response in simple language. One of the important features of the system is the use of RAG technology, where related information is first taken from trusted stored data and then combined with AI-generated output. This helps in giving more accurate and useful responses. The platform is easy to use and can run on web or mobile devices, so students can access it anytime and from any location. It also works 24/7 and supports multiple languages, making it useful for a wider group of users. For development, React.js and Next.js are used for the frontend, while Node.js is used for backend functions and API handling. OpenAI API and LangChain are used to improve chatbot communication, and the database stores important data and user activity. Overall, the system is a practical and student-friendly solution that saves time, improves communication, and reduces workload for staff.

The system is also designed to improve the overall experience of students by reducing confusion during important academic processes. Many students face difficulty while searching for correct information from different websites or offices. This chatbot solves that problem by providing all major details in one place. It can guide students during admission periods, explain eligibility criteria, provide counseling updates, and help them understand scholarship opportunities. By offering quick and clear answers, the platform supports better decision-making and saves valuable time for students and parents.

Another advantage of the proposed project is its scalability and future growth potential. The platform can be upgraded regularly by adding new data, improved AI models, and additional services

according to student needs. Features such as voice interaction, personalized recommendations, document assistance, and mobile application support can be added in future versions. The system can also be adapted for use in colleges, universities, and other educational departments. This makes the project a long-term digital solution that can continue improving student support services over time.

3.OBJECTIVE

To create and implement an AI-powered student assistant that can deliver smart, real-time, and automatic responses to different types of student queries, helping improve the speed, quality, and effectiveness of communication between students and educational institutions.

1. To build a centralized digital platform that provides correct and updated academic as well as administrative information related to admissions, course details, examination schedules, fee structures, scholarships, placements, and other important student services, while reducing reliance on traditional communication methods.
2. To apply advanced Natural Language Processing (NLP) methods that allow the system to understand questions asked in normal human language, recognize user intent, and generate meaningful, relevant, and context-aware responses for a smoother and more user-friendly experience.
3. To improve the precision and dependability of responses by using Retrieval-Augmented Generation (RAG), where information is collected from a structured and verified knowledge base and combined with AI-generated outputs so that students receive accurate and useful answers.
4. To reduce the burden on administrative staff by automating common and repetitive student queries, allowing employees to focus on more important, complex, and decision-based tasks while increasing institutional productivity and work efficiency.
5. To provide an accessible and inclusive system with round-the-clock availability and multilingual support, ensuring that students from different language groups and geographical locations can use the platform anytime and from anywhere without difficulty.
6. To design a scalable and flexible architecture capable of handling many users at the same time, while also allowing future upgrades through new features, technologies, and additional data sources as educational requirements continue to change over time.

4. METHODOLOGY

The development of the AI Student Assistant for Rajasthan Technical Education is carried out through a well-planned and systematic methodology that combines data collection, Natural Language Processing (NLP), Machine Learning (ML), and Retrieval-Augmented Generation (RAG). The aim of this methodology is to create an intelligent chatbot that can understand user questions and provide fast, accurate, and useful responses related to academic and administrative services.

1. Data Collection

The first phase of the methodology involves gathering relevant and trusted information from official sources. This includes details regarding admissions, courses, examination schedules, fee structures, scholarships, placements, and other student services offered by the Department of Technical Education, Rajasthan. The required data is collected from government websites, institutional portals, notices, and verified resources to maintain correctness and reliability.

2. Data Preprocessing and Knowledge Base

After the data is collected, it is processed carefully to remove duplicate entries, errors, unnecessary text, and inconsistent records. The cleaned information is then arranged in a proper structured format and stored in a knowledge base. This knowledge base works as the main source of information for the chatbot and helps in quick retrieval of suitable content during query handling.

3. Natural Language Processing (NLP)

Natural Language Processing is an important component of the system because it allows the chatbot to understand questions written in normal human language. NLP methods are used to study the user input, identify keywords, detect user intent, and understand the meaning of the sentence. This enables the chatbot to answer effectively even when different users ask the same question in different styles.

4. Implementation of Retrieval-Augmented Generation (RAG)

The system uses Retrieval-Augmented Generation (RAG) to improve the quality and reliability of responses. In this method, the chatbot first searches the knowledge base and collects the most relevant information based on the user query. After that, the

retrieved data is combined with AI-generated content to produce clear, context-based, and accurate answers.

5. System Development

The complete platform is developed using modern web technologies for better speed, performance, and scalability. The frontend section is created using React.js and Next.js to provide a responsive and user-friendly interface. The backend is developed using Node.js, which manages API requests, data processing, and communication between different parts of the system. AI tools such as OpenAI API and LangChain are integrated to support advanced chatbot interaction.

6. Database Integration

A database is connected to the platform to store structured records, chatbot data, and user interaction history. It helps in managing information efficiently and supports fast retrieval whenever required. The database also assists in improving future system performance through stored usage records.

7. System Testing

After development, the chatbot is tested with different types of user queries to evaluate response accuracy, processing speed, and overall performance. Multiple test cases are considered, including common student questions, difficult queries, and multilingual inputs. Testing ensures that the system works properly in real-world conditions and gives dependable results.

8. Deployment

Once testing and optimization are completed, the system is deployed on web and mobile platforms for student use. Deployment ensures that the chatbot remains available throughout the day so users can access services anytime and from any location.

9. Continuous Learning and Improvement

The AI Student Assistant is designed for continuous improvement through regular usage and feedback. User queries and interaction records are analyzed to improve understanding and answer quality over time. Machine Learning techniques are applied to refine the chatbot model, making it smarter, faster, and more effective in future operations.

5. RESULT

The developed DTE Rajasthan Student Assistant web application offers a smart and easy-to-use platform for students searching for information related to engineering and polytechnic education in Rajasthan. Based on the available interface screens, the system includes several useful modules such as AI chatbot support, college search, admission guidance, personalized recommendations, scholarship details, and contact services. The application shows an effective integration of frontend design, backend processing, and AI-driven user interaction.

The homepage of the platform displays an AI chatbot interface where students can ask questions related to admissions, leading colleges, fee details, scholarships, and counseling procedures. Suggested query cards improve user convenience by allowing quick access to commonly searched topics. The chatbot input area supports both text and voice options, which increases accessibility and makes interaction easier. This module reduces the need for manual counseling and delivers immediate responses to users.



Fig.1. Home page

The colleges module presents an organized list of engineering and polytechnic institutes available in Rajasthan. Each college card contains details such as institution image, location, offered branches, yearly fees, average salary package, and placement information. The search feature allows students to sort colleges by name, city, or code. Category options like All, Engineering, and Polytechnic make navigation easier and help users compare institutes in an efficient manner. This module increases transparency and saves time during the college selection process.

diploma courses. Important information such as registration dates, entrance examination schedule, and counseling timelines is clearly presented. Eligibility requirements, admission steps, and complete counseling procedures are arranged in a structured format for easy understanding. This section makes the admission process simpler for students and reduces confusion related to documents, schedules, and deadlines.



Fig.2. College list page

The Find College module works as a recommendation system where users can enter their rank, preferred branch, location, and type of college. Based on these inputs, the platform provides suitable college suggestions. This personalized filtering feature supports better decision-making and helps students identify the most appropriate institutions according to their academic background and preferences.



Fig.3. Admission page

The Contact Us module creates a direct communication link between students and the administration team. Users can submit their name, email address, phone number, and message through a responsive inquiry form. This feature helps provide support for unresolved queries and builds confidence in the platform. The simple and clean form design reflects a strong focus on usability and accessibility.

The left-side navigation panel allows smooth access to all major modules such as Chat, Colleges, Admissions, Scholarships, Find College, Predictor, and Contact. This consistent navigation structure

improves user experience and makes movement across different sections easy and convenient.



Fig.4.College Search Page

From a performance point of view, the application appears responsive, well-organized, and simple to use. Card-based layouts, categorized menus, modern icons, search options, and interactive buttons help create an attractive user interface. The combination of AI support with educational information makes the platform highly beneficial for students in both rural and urban areas.



Fig.5.Contact us page

Overall, the project successfully meets its goal of digitizing technical education guidance for students in Rajasthan. It helps reduce information gaps, offers personalized recommendations, and brings multiple educational services onto a single platform. The system can greatly improve student awareness, decision-making ability, and access to important admission resources.

6. DISCUSSION

The proposed system is a modern AI-based Student Assistant developed to improve learning support by delivering intelligent and real-time responses through technologies such as Large Language Models (LLM) and semantic search. In the initial stage, the system gathers course-related content from the Learning Management System (LMS). These

course documents are then processed for further analysis and response generation.

Through a data extraction module, useful text and important information are identified, filtered, and cleaned. This process ensures that only relevant academic content is selected for further use. After extraction, the content is divided into smaller sections called text chunks. This step is necessary because large documents are harder for AI models to process efficiently, while smaller chunks improve response accuracy and retrieval speed.

Each text chunk is then converted into vector representations known as embeddings. These embeddings capture the semantic meaning of the content instead of depending only on keywords. All generated embeddings are stored in a Knowledge Base, which works as an organized repository for quick information retrieval.

When a student asks a question, the system converts the query into a question embedding. Using semantic search, the query embedding is matched with stored embeddings in the Knowledge Base to identify the most relevant information. The selected results are then ranked according to relevance and sent to the LLM-based Generative AI module, which produces a clear and human-like response. This ensures that students receive structured, meaningful, and easy-to-understand answers instead of raw data.

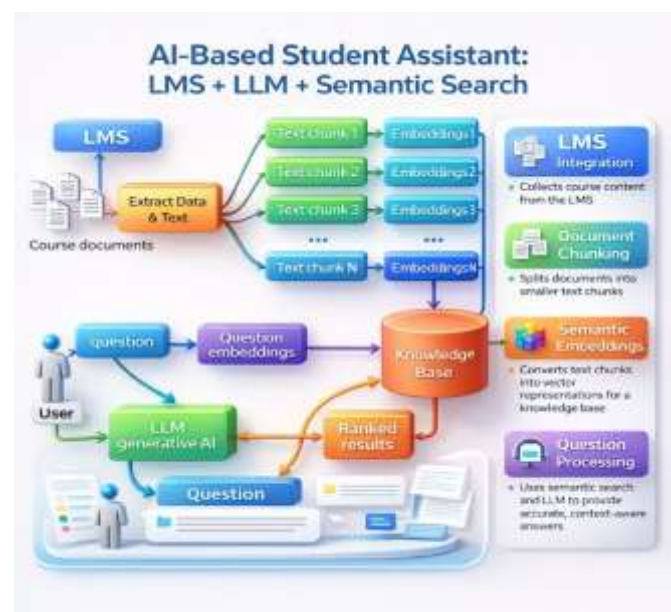


Fig.6. Data flow Diagram

Frontend (User Interface Layer)

The frontend of the system is developed using modern web technologies such as HTML, CSS, and JavaScript, with React.js used as the main framework to create a dynamic and responsive user interface. It offers an interactive chatbot-style environment where students can easily enter their questions and receive answers in real time. The user interface contains features such as input fields, chat bubbles, loading indicators, and response display areas, which make the system easy to use and visually attractive. Styling frameworks such as Tailwind CSS or Bootstrap are used to improve the overall design by providing a clean layout, device responsiveness, and a professional appearance suitable for educational platforms.

Backend (Application Logic Layer)

The backend of the system is developed using Node.js, which is well suited for creating scalable and real-time web applications. Next.js is used to manage both frontend and backend operations through API routes and server-side processing. The backend handles major functions such as receiving user queries, processing requests, connecting with AI services, retrieving relevant data, and generating useful responses. Supabase is used as the database to securely store user details, chat history, and application records. It also offers authentication and real-time features for smooth platform performance. The backend works as the central processing unit of the system, ensuring quick response time, seamless communication, and dependable integration of modern AI features.

The project follows a modular architecture where the system is divided into major functional components such as user interface, backend processing, database management, and AI support. This organized design improves the scalability, maintainability, and overall performance of the application.

Overall Architecture

The project is built on a three-layer architecture that includes the presentation layer, application layer, and data layer. The presentation layer is responsible for handling user interaction and displaying system features through the interface.

It is responsible for user interaction and contains all pages, forms, buttons, menus, and dashboards. The application layer manages business logic, request handling, AI chatbot responses, recommendation features, and communication between the frontend and database. The data layer stores important information such as colleges, branches, fee details, scholarships, and admission records.

Database Structure

The database stores all essential records needed by the system. A relational or cloud-based database such as Supabase, PostgreSQL, or MongoDB can be used for data management. Main tables or collections include Colleges, Courses, Admission Details, Scholarships, Student Queries, Contact Messages, and User Preferences. Each college record may contain fields such as college name, city, available branches, fees, package, placement details, ranking, and category. Proper indexing and a structured schema help improve the speed of data retrieval.

AI Chatbot Module

The chatbot module is one of the main components of the project. It is integrated using AI APIs and Natural Language Processing techniques. This module accepts student queries in text or voice format and provides meaningful responses related to admissions, colleges, fee structures, scholarships, and counseling procedures. Suggested prompts are also shown to improve user convenience. This feature reduces manual dependency and provides support 24/7.

College Recommendation Module

The Find College feature works as a recommendation engine for students. It accepts inputs such as rank, preferred branch, location, and type of college. Based on predefined logic or filtering methods, the system compares database records and suggests the most suitable colleges. This feature supports personalized decision-making and makes counseling preparation



Fig.7.Project Architecture

The DTE Rajasthan Student Assistant project is developed as a smart web-based platform that helps students obtain educational information related to engineering and polytechnic colleges in Rajasthan.

easier for students.

7. CONCLUSION AND FUTURE WORK

The proposed AI-Based Student Assistant provides a major improvement over traditional learning systems by combining advanced technologies such as Large Language Models (LLMs), Natural Language Processing (NLP), and semantic search. These technologies enable students to receive accurate, context-aware, and real-time responses. Unlike conventional Learning Management Systems (LMS) that mainly depend on static content and manual searching, this system intelligently understands user queries and delivers meaningful answers, making the learning experience more interactive, efficient, and student-friendly.

The use of embeddings and vector databases plays an important role in improving system performance because it allows the platform to understand the semantic meaning of text instead of relying only on keyword matching. By converting course materials and user questions into vector representations, the system can perform efficient similarity searches to retrieve the most relevant and contextually suitable information. This process greatly increases the accuracy, speed, and reliability of responses provided to students.

The architecture of the system combines a user-friendly frontend interface, a powerful backend, and an optimized database layer to ensure strong scalability, flexibility, and performance.

This organized structure allows the system to manage large volumes of academic data and multiple user requests at the same time, making it suitable for use in colleges, universities, and online learning environments.

The system reduces the time and effort required by students to search for useful academic information, which improves productivity and encourages independent learning. By providing instant access to clear and structured answers, the AI assistant helps students understand difficult topics more effectively and supports better academic performance regardless of time or location.

In conclusion, this project represents an important step toward the development of intelligent and smart education systems. With future upgrades such as voice interaction, multilingual support, personalized recommendations, and mobile or cloud integration, the platform has the potential to transform digital education by making it more adaptive, inclusive, and

student-centered, while reducing the gap between

8. FUTURE WORK

The proposed AI-Based Student Assistant can be further improved by adding several advanced features to enhance functionality and user experience. Future development may include voice-based interaction, allowing students to ask questions through speech and receive spoken responses, making the platform more accessible and interactive. In addition, multilingual support can help students from different language backgrounds use the system more effectively. The platform can also be upgraded with personalized learning recommendations based on user behavior, previous queries, and performance analysis. Integration with mobile applications and cloud platforms can improve accessibility and scalability, allowing students to use the assistant anytime and from anywhere. Furthermore, offline AI models and stronger data security mechanisms can make the system more reliable and secure. These future improvements can transform the platform into a more intelligent, adaptive, and user-friendly educational assistant.

9. REFERENCE

- [1] Sharma, R.; Gupta, P.; Meena, S.; Verma, A. AI-Based Chatbot for Student Query Resolution in Technical Education. *Int. J. Comput. Appl.* 2024, *186*, 15–22. <https://doi.org/10.5120/ijca2024.18615>
- [2] Singh, V.; Joshi, M.; Kumar, R.; Jain, N. Machine Learning Approach for Student Performance Prediction in Higher Education. *Procedia Comput. Sci.* 2023, *218*, 145–152. <https://doi.org/10.1016/j.procs.2023.01.145>
- [3] Mehta, A.; Bansal, K.; Sharma, D. Natural Language Processing for Multilingual Learning Systems in India. *J. Educ. Technol. Syst.* 2024, *53*, 88–101. <https://doi.org/10.1177/0047239524123456>
- [4] Rajasthan Technical University. Smart Academic Management System Using Artificial Intelligence. *RTU Res. Rep.* 2024, Kota, India.
- [5] IIT Madras. NPTEL Personalized Learning Using AI-Based Recommendation Systems. *NPTEL Tech. Rep.* 2025, Chennai, India.
- [6] IIT Bombay. Virtual Labs for Technical

Education: AI Enabled Remote Learning Platform.
IIT Bombay Educ. Rep. 2024, Mumbai, India.

[7] Verma, S.; Choudhary, R.; Jain, P. Educational Data Mining for Student Retention Analysis in Engineering Colleges. *Int. J. Data Sci.* 2023, *11*, 34–42. <https://doi.org/10.1007/s41060-023-00342-1>

[8] Agarwal, P.; Sinha, R.; Tiwari, M. Deep Learning Based Recommendation System for E-Learning Platforms. *IEEE Access* 2024, *12*, 45671–45682.
<https://doi.org/10.1109/ACCESS.2024.3378912>

[9] Ministry of Education, Government of India. National Education Policy 2020. New Delhi, India, 2020.

[10] AICTE. AICTE Approval Process Handbook 2025–26. All India Council for Technical Education, New Delhi, India, 2025.

[11] BITS Pilani. Artificial Intelligence Applications in Smart Learning Environments. *BITS Pilani Res. Rep.* 2024, Pilani, Rajasthan, India.

[12] MNIT Jaipur. Digital Transformation in Technical Education Using AI Tools. *MNIT Jaipur Tech. Rep.* 2025, Jaipur, India.