

Admit Genie

Abdul Khader Rayif¹, Shana Santhosh², Gowthami³, M Pranamy⁴, Muhammad Hafil⁵

^{1,3,4,5}Student, Srinivas Institute of Technology

²Assistant Professor, Srinivas Institute of Technology

doi.org/10.64643/IJIRTV12I12-206761-459

Abstract—This project aims to develop a college enquiry Chabot that answers any queries post by students like college details, course-related questions, location of the college, fee structure etc. The College Enquiry Chatbot project is built using machine learning algorithms that analyze user's queries and understand the user's message. Chatbot systems have become increasingly popular for automating interactions with users and providing information in various domains, including college enquiries. This System is a web application that provides answers to the query. Any individual just has to query through the bot. The answers are appropriate to what the user queries. The User can query any college-related activities through the system. The user does not have to personally go to the college for enquiry. The System analyses the question and then answers to the user. Admission Enquiry Chatbot is an innovative artificial intelligence-powered solution designed to provide prospective students with instant and personalized responses to their admission-related queries. This chatbot leverages natural language processing and machine learning algorithms to understand and respond to user inputs, offering a user-friendly and efficient way to access admission information. By automating the admission enquiry process, the chatbot aims to enhance the overall applicant experience, reduce administrative burdens, and improve institutional responsiveness. The user can also give their suggestions through the suggestion box. The system replies using an effective Graphical User Interface which implies that as if a real person is talking to the user. The chatbot uses keyword matching to identify relevant information related to the user's query. The chatbot employs NLP to understand the context and intent behind the user's query. The chatbot has access to a knowledge graph that contains a vast amount of information related to the university, its programs, admission requirements, and more.

I. INTRODUCTION

The college admission process is often a challenging and time-consuming journey for students. It involves multiple steps such as understanding eligibility

criteria, gathering information about courses, meeting application deadlines, preparing necessary documentation, and staying updated with CET examination details and fee structures. Many students struggle to find accurate and timely information, which can lead to confusion and delays in the application process. Similarly, administrative staff often face a heavy workload, as they have to handle repetitive queries from a large number of applicants. To simplify this process and ensure efficient communication between students and institutions, Admit Genie was developed as an innovative solution to address these challenges. Admit Genie is an AI-powered intelligent chatbot specifically designed to guide students throughout the college admission process. It leverages the power of Natural Language Processing (NLP) and Machine Learning (ML) to understand, analyze, and respond to students' queries in a conversational manner, providing a more interactive and personalized experience. Unlike static FAQs or traditional websites, Admit Genie allows students to directly ask questions. In India, with KCET, JEE, MHT-CET, and other entrance exams, the admission landscape is particularly complex. Admit Genie specifically addresses this by related to admissions, such as eligibility criteria, application deadlines, required documents, college fee structures, scholarship opportunities, CET examination details, available courses, and more information about the institution. The chatbot continuously learns from user interactions, improving its accuracy and capability to handle complex queries over time. By simulating human-like conversations, it creates a user-friendly interface that makes obtaining information faster, simpler, and more engaging. With Admit Genie, students can easily obtain accurate and timely information by simply interacting with the chatbot, eliminating the need for manual inquiries. Additionally, by automating repetitive tasks, the

system significantly reduces the workload of administrative staff, allowing them to focus on more critical admission-related operations. This not only enhances the efficiency of the admission process but also improves the overall applicant experience, making the journey from inquiry to admission smoother and more convenient. In today's digital era, higher education institutions are increasingly adopting technology to streamline administrative processes and enhance communication with students. However, most existing systems still lack intelligent, real-time interaction capabilities, leaving students overwhelmed when navigating complex admission procedures. The growing number of applicants each year adds to the pressure on institutional resources.

II. LITERATURE REVIEW

The literature survey reviews existing research, tools, and technologies related to AI-powered chatbots and their applications in the admission process. Several studies highlight the use of Natural Language Processing (NLP) and Machine Learning (ML) to develop intelligent chatbots that provide quick and accurate responses to user queries.

[1] *"Retrieval-Polished Response Generation for Chatbot"* paper introduces a retrieval polished (RP) model for chatbot response generation, designed to overcome the limitations of retrieval-based and generation-based models. Retrieval-based methods ensure fluency but may lack contextual relevance, while generation-based models produce relevant replies but often generate "safe" or less diverse responses.

[2] *"Static Quality Assurance for Chatbots"* article addresses the lack of quality assurance (QA) techniques in chatbot development, especially during early stages when bots are not fully functional. Unlike traditional software, chatbots are built around intents, training phrases, and conversation flows, which require specialized QA methods.

[3] *"The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review"* paper presents a systematic literature review (SLR) of 74 scholarly papers examining how chatbots contribute to digital business transformation (DBT)

across industries. The findings highlight that most research focuses on user perceptions, acceptance, communication styles, customer service, and learning applications.

[4] *"Retrieval-Polished Response Generation for Chatbot"* paper introduces a retrieval polished (RP) model for chatbot response generation, designed to overcome the limitations of retrieval-based and generation-based models. Retrieval-based methods ensure fluency but may lack contextual relevance, while generation-based models produce relevant replies but often generate "safe" or less diverse responses.

[5] *"Static Quality Assurance for Chatbots"* article addresses the lack of quality assurance (QA) techniques in chatbot development, especially during early stages when bots are not fully functional. Unlike traditional software, chatbots are built around intents, training phrases, and conversation flows, which require specialized QA methods.

[6] *"The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review"* paper presents a systematic literature review (SLR) of 74 scholarly papers examining how chatbots contribute to digital business transformation (DBT) across industries. The findings highlight that most research focuses on user perceptions, acceptance, communication styles, customer service, and learning applications.

[7] *"Chatbot for College Enquiry"* — In this paper, authors K.T Krishna Kumar, Kiran Mamidipaka proposes a Python Flask-based college chatbot that leverages NLP, CNN, and LSTM techniques to enhance query handling. It provides real-time.

[8] *"Admission Enquiry Chatbot"* — In this paper, authors Rithika Sharma, Threesha, Yashaswi C Shetty, Yashaswi K B, Prof. Shruthi V S develops an admission enquiry chatbot using a mix of rule-based logic and machine learning, integrated with NLP tools like Dialogflow.

[9] *"Index Delayed User Interfaces, Improving Responsiveness and Usability"* — In this paper, focuses on delayed UI responses that reduce usability

and user satisfaction in modern applications. The authors proposed an Index Delayed UI Model that prioritizes event scheduling, rendering, and task queues to minimize perceived latency.

III. REQUIREMENTS

Admit Genie is a smart platform designed to simplify the college admission process. It helps students easily find course information, check eligibility, understand fee structures, track application deadlines, and access CET exam updates. It also reduces the workload on college administration by answering repeated queries automatically.

I. HARDWARE REQUIREMENTS

The Admit Genie system requires basic hardware resources for smooth operation. At a minimum, the system can run on a dual-core 2.0 GHz processor with 4 GB of RAM and 250 GB of storage, along with a display resolution of 1024×768 and a stable internet connection of at least 1 Mbps. However, for better performance and scalability, it is recommended to use an Intel i5 or AMD Ryzen 5 processor, 8 GB or more of RAM, and a 500 GB SSD for faster data processing. A higher display resolution of 1366×768 or above, along with a high-speed internet connection of 10 Mbps or more, ensures smooth user interaction and efficient handling of multiple queries.

The system should support cross-platform operation, meaning it can run smoothly on Windows, Linux, and macOS machines. For deployment on servers, a multi-core processor and at least 16 GB RAM is recommended to handle large-scale queries during peak admission seasons. If the system is hosted on cloud platforms like AWS or Azure, scalable instance types (t2. medium or higher) ensure better performance. Devices accessing the chatbot (mobile or PC) should have updated browsers to support real-time interactions.

For institutions running the backend, a stable power supply and backup storage solution (external HDD or cloud backup) are necessary.

II. SOFTWARE REQUIREMENTS

Programming Languages: Node.js (for NLP & ML models), JavaScript (for frontend chatbot interface).

Frameworks & Libraries:

- Express.js (backend server)
- React.js (frontend development)
- NLTK (for NLP processing)
- Scikit-learn / TensorFlow / (for ML models)

Database:

MySQL for storing student queries and admission-related data.

Web Server: Apache / Nginx (for hosting the application)

Cloud Services (optional):

AWS / Azure / Google Cloud for deployment and scalability.

Version Control:

Git & GitHub for source code management

Package Manager:

NPM (for frontend) and pip (for Node.js libraries)

IV. METHODOLOGY

I. SYSTEM ARCHITECTURE

Admit Genie employs a three-tier PERN stack architecture to support real-time NLP-driven functionalities like query classification and personalized responses, prioritizing security, scalability, and user accessibility.

The frontend (React.js) provides a responsive chat interface for voice/text inputs without handling sensitive data like API keys or student profiles, relying on RESTful API calls to the backend. The Express.js/Node.js backend manages authentication, NLP preprocessing, model routing, and integrations with external services, while PostgreSQL stores anonymized session logs, eligibility rules, and college databases.

II. MODULE LEVEL METHODOLOGY

Admit Genie divides functionality into focused modules—Chat Assistant, Eligibility Checker, College Recommender, and Query Analyzer—each following a uniform workflow: input capture, validation, backend NLP processing, model inference, and structured JSON output. The Chat Assistant module parses natural language queries using tokenization and intent recognition, forwarding to ML

models for response generation. Eligibility Checker extracts user metrics (e.g., CET scores) via named entity recognition (NER), cross-references college cutoffs, and scores matches; modules like College Recommender use similarity algorithms on profiles.

III. FLOW OF THE SYSTEM

User flow starts at the React frontend where students input queries (e.g., "Am I eligible for engineering colleges with 85% CET?"); data routes via HTTP POST to the backend for authentication, sanitization, and NLP pipeline activation. Backend applies tokenization, intent classification, and entity extraction before model invocation; outputs undergo post-processing like formatting eligibility tables or college lists.

Refined JSON returns to frontend for display, with logs stored for analytics, ensuring sub-second latency during peak seasons.

IV. AI MODEL INTEGRATION FLOW

Backend centralizes NLP/ML integration (e.g., BERT for classification, GPT variants for generation, custom fine-tuned models) via a handler abstracting provider differences like.js or Angular, providing students and administrators with intuitive chat interfaces that support both text and voice interaction. All user inputs, queries, and system updates are transmitted to the backend using secure RESTful APIs, ensuring privacy and data integrity while keeping all processing and sensitive admission content server-side.

The backend serves as the intelligence center of Admit Genie, orchestrating natural language processing, business logic, and data handling. User requests are received and Ollama or OpenRouter APIs. Requests standardize prompts (e.g., "Classify admission query and extract CET score"), with rate limiting, retries, and error fallbacks to the architecture of the Admit Genie college admission chatbot is based on a modular, layered design utilizing modern web technologies for reliability and scalability.

The frontend is built with a responsive JavaScript framework such as React, parsed by the NLP Processor, which interprets questions using intent detection and entity extraction models built with libraries like SpaCy or Transformer-based frameworks.

V. API COMMUNICATION FLOW

Frontend-backend interactions use RESTful JSON APIs with middleware (e.g., multer for file uploads like transcripts); POST /query endpoints handle inputs securely via environment variables. Backend proxies to NLP services, catches errors (e.g., model timeouts), and returns user-friendly messages; CORS and JWT ensure authenticated, stateless sessions. This isolates sensitive operations, supporting high concurrency without frontend exposure.

VI. DATA PROCESSING STEPS

NLP pipeline tokenizes queries, applies NER for entities (e.g., scores, courses), normalizes data against college DB, and feeds cleaned inputs to models. Post-inference, backend removes hallucinations, structures outputs (e.g., ranked college lists), and anonymizes logs; voice inputs convert via speech-to-text before processing. Steps ensure GDPR compliance, accuracy, and readability for features like eligibility scoring.

Admit Genie's PERN-based methodology deliver a modular, secure NLP platform where frontend chat flows seamlessly to backend processing, NLP models, and database persistence, automating inquiries with low latency and high accuracy. Uniform module workflows, standardized integrations, REST APIs, and robust pipelines ensure maintainability, scalability, and privacy for evolving admission needs.

V. DESIGN

1. ARCHITECTURE

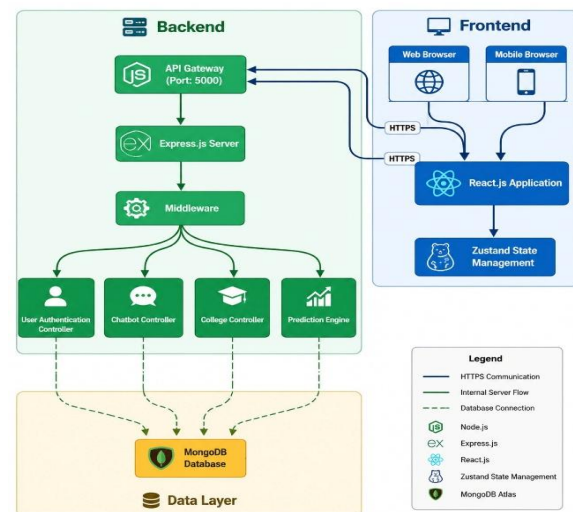


Fig1: System Architecture

II. BLOCK DIAGRAM DESIGN



Fig2: Block Diagram Design

The block diagram design of the Admit Genie chatbot system presents a clear and modular view of how each primary component delivers admission guidance to students and administrators. The process starts at the user interface, where users enter their questions via text or voice. This input is immediately sent to the messaging layer, which ensures that every query, regardless of format, is standardized and routed to the backend for processing.

III. SEQUENCE DIAGRAM

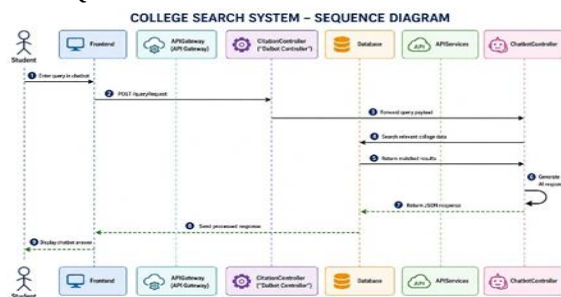


Fig3: Sequence Diagram

The sequence diagram describes the end-to-end flow of a student's query in the Admit Genie system, detailing how information moves across system components to produce a chatbot response. The process begins when the student submits a query via the frontend (web or mobile interface). The Frontend captures this input and sends it as a POST request to the API Gateway, which manages backend communication and routing. The API Gateway forwards the query to the Chatbot Controller, which is responsible for processing natural language queries and determining the required action.

The Chatbot Controller queries the Database to fetch relevant admission, college, or profile information as needed by the query. Results from the database are returned to the Chatbot Controller, which compiles and generates an appropriate JSON response. This response is sent back through the API Gateway to the Frontend. Finally, the Frontend displays the answer to the student, completing the loop. This sequence ensures efficient, secure, and accurate handling of student queries, allowing users to interact conversationally with Admit Genie and receive real-time admission support. The sequence diagram of the Admit Genie system illustrates a complete, step-by-step interaction flow from the moment a student submits a query to the delivery of the final response. The process begins when the student enters a question on the web or mobile interface, which the frontend instantly captures and packages into a structured POST request. This request is sent to the API Gateway, which acts as the central router for all communication, ensuring security, rate-limiting, and proper request validation. Once validated, the API Gateway forwards the query to the Chatbot Controller—the main processing unit responsible for interpreting and handling user queries.

If the query demands official admission details, eligibility information, fee structure, or course data, the controller fetches the necessary records from the database through a secure query. The database returns the relevant results, which the controller formats into a clean and meaningful JSON response. This response travels back through the API Gateway to the frontend interface, where it is presented to the student in a clear and conversational format. This well-orchestrated flow ensures that every query is handled efficiently,

securely, and accurately, allowing Admit Genie to deliver instant, reliable, and context-aware admission guidance.

VI. RESULTS

The Admit Genie project successfully delivers an intuitive and AI-powered college admission assistance platform. The landing page clearly communicates the core purpose of the system — guiding students through the Indian college admission process using artificial intelligence. The homepage design is clean, modern, and user-friendly, making it easy for students to begin interacting with the platform.

Another important criterion is usability and user experience. Testing in this phase involves evaluating the chatbot's interface design, accessibility, and clarity of responses. Students should find it easy to interact with the chatbot through Streamlit's input box and receive well formatted, readable outputs. The Admit Genie project successfully delivers an intuitive and AI-powered college admission assistance platform. The homepage design is clean, modern, and user-friendly, making it easy for students to begin interacting with the platform.

I. HOME PAGE

The Home Page constitutes the primary entry interface for the Admit Genie platform, designed to provide students with immediate and intuitive access to all core functional modules. As shown in the screenshot, the interface uses a clean layout with a prominent navigation bar at the top that contains direct links to the Dashboard, Colleges, Rank Predictor, Chat Assistant and Logout options. This ensures that users can quickly move between major sections of the system without confusion or unnecessary clicks.

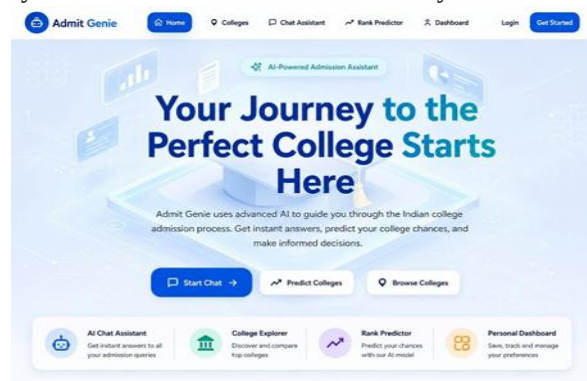


Fig1: Home Page

II. LOGIN PAGE

The login page of Admit Genie typically prompts users to enter their email and password in dedicated fields to access their accounts securely. Additionally, the page may request other essential information such as the user's full name, a contact number, selection of the user type (student or parent), and acceptance of terms and conditions.

These fields help in verifying user identity, ensuring account security, and personalizing the platform experience.

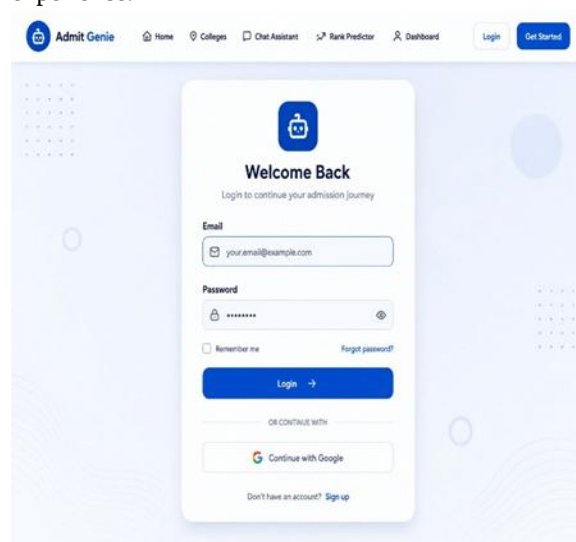


Fig 2:login page

III. COLLEGES

The displayed page from the Admit Genie platform shows the Find Your Perfect College section, where users can browse a comprehensive database of colleges and universities. The interface includes a search bar that allows filtering colleges by name, location, or description, along with a prominent “Explore Colleges” button.

Below, the page showcases top engineering institutes in India—such as IIT Bombay, IIT Delhi, IIT Madras, IIT Kanpur, IIT Kharagpur, and NIT Trichy—each presented in card format with details like location, accreditation, brief descriptions, and a “View Details” option.

This organized and user-friendly layout helps students easily compare top institutions and navigate towards selecting the best college for their academic goals.

The "Find Your Perfect College" section on the Admit Genie platform is designed with a keen focus on user-centricity, ensuring effortless navigation and informed decision-making for prospective students.

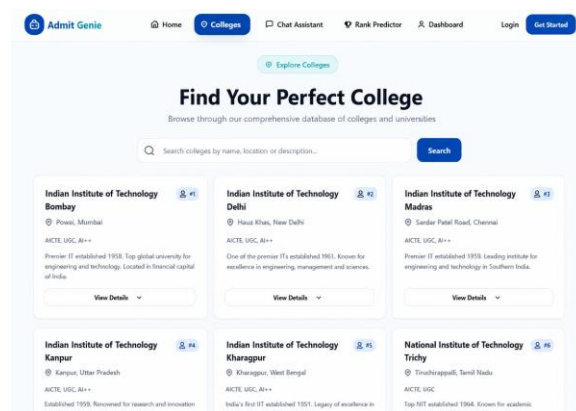


Fig3: Colleges

IV. CHAT ASSISTANT

The chat assistant in the Admit Genie project acts as an intelligent, interactive guide for students navigating the college admission process. Accessible directly from the home page, this chatbot uses natural language processing and machine learning to understand and respond accurately to queries about eligibility, fee structures, college details, admission cutoffs, and placement statistics. Students can type or use voice input to ask questions such as, “Which colleges am I eligible for with my CET rank?”, “What is the average fee for Computer Science in top colleges?”, or “What are the placement rates at XYZ College?” The chatbot automatically detects the user's intent, consults the relevant database tables using backend algorithms, and generates a clear, personalized reply within seconds. Every query and its response are logged, allowing for continuous improvement and user-specific support.

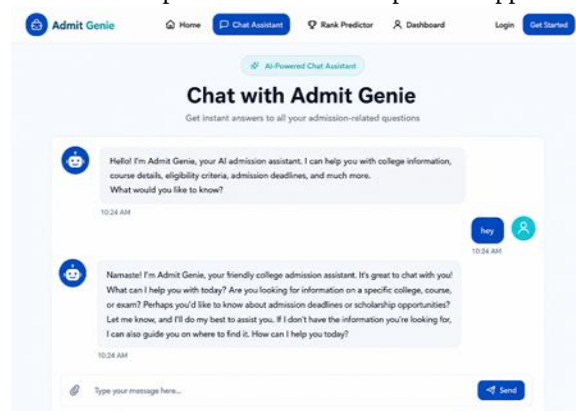


Fig4: Chat Assistant

V.COLLEGE RANK PREDICTOR

The College Rank Predictor is a key decision-support feature that helps students estimate their chances of

admission based on their entrance exam rank and reservation category. The interface is divided into an input section and a result section, making it easy for users to understand the flow of information. In the input area, students first select the relevant examination, such as KCET, JEE or other supported tests, from a dropdown list. They then enter their obtained rank in a numeric field and choose their category (General, OBC, SC, ST, etc.) from a set of options. An optional field is provided to specify the preferred branch, for example CSE, ECE or Mechanical, so that the output can be filtered according to individual interest.

Once the student submits the form, the system validates the entered values to ensure that the rank is within an acceptable range and that required fields are not left empty. After validation, the backend component retrieves historical cutoff data from the cutoff_data table in the MySQL database. This data includes college-wise, branch-wise and category-wise cutoff ranks for previous admission years. The predictor then compares the user's rank with these stored values and applies simple probability rules to classify each college-branch combination into safe, moderate or difficult categories.

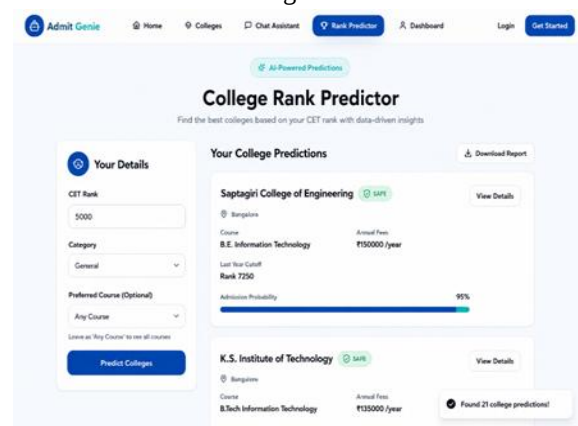


Fig5: College Rank Predictor

REFERENCES

- [1] V. S. Thrisha, P. R. Bhat, S. Pachori, S. Mishra, and S. Majee, "College Admission Enquiry Chatbot," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 5, pp. M28–M34, May 2024.
- [2] R. Gokul and S. Kalaivani, "AI College Enquiry Chatbot System," *Journal of Emerging Technologies and Innovative Research (JETIR)*,

- vol. 10, no. 2, pp. D446–D450, Feb. 2023.
- [3] K. T. Krishna Kumar and K. Mamidipaka, "Chatbot for College Enquiry," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 7, pp. D122–D127, Jul. 2024.
 - [4] R. Sharma, Threesha, Y. C. Shetty, Y. K. B., and S. V. S. Shruthi, "Admission Enquiry Chatbot," *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 12, no. 7, Jul. 2025.
 - [5] J. S. Cuadrado et al., "Integrating Static Quality Assurance in CI Chatbot Development Workflows," *IEEE Software*, vol. 41, no. 5, pp. 60–69, 2024.
 - [6] L. Zhang, Y. Yang, J. Zhou, C. Chen, and L. He, "Retrieval-Polished Response Generation for Chatbot," *IEEE Access*, vol. 8, pp. 123882–123890, 2020, doi: 10.1109/ACCESS.2020.3000123.
 - [7] Various Authors, *Advanced Natural Language Processing for Technical Applications*. Springer, 2024–2025.
 - [8] Various Authors, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review." Elsevier, 2024.
 - [9] Gunnam et al., "Assessing Performance of Cloud-Based Heterogeneous Chatbot Systems," *ACM Digital Library*, 2024.
 - [10] G. A. Santos et al., "A Conversation-Driven Approach for Chatbot Management," *IEEE Access*, vol. 10, pp. 8474–8489, 2022.