International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Powered Helpdesk For College Administration

OUR TEAM GUIDE : **Mrs. V . V . VISHNUPRIYA** (Assistant professor)
Department of Computer science and engineering
Angel college of engineering and technology, Tirupur

Arivananthan . T
Department of computer science Engineering
Angel College of Engineering and technology , Tirupur
arivananthan144@gmail.com

Parvathi . K
Department of computer science Engineering
Angel College of Engineering and technology , Tirupur
parvathiad7@gmail.com

Udhayan . S
Department of computer science Engineering
Angel College of Engineering and technology , Tirupur
vasukisasikumar1000@gmail.com

Deva Priyan . G.H
Department of computer science Engineering
Angel College of Engineering and technology , Tirupur
deva637462@gmail.com

ABSTRACT

This paper presents the design and implementation of an AI-powered helpdesk system for college administration. The proposed system integrates a chatbot using Natural Language Processing (NLP) techniques to automate student queries related to attendance, fees, examinations, and placements. The system uses TF-IDF vectorization with a Naive Bayes classifier for intent classification, enabling accurate understanding of user queries.

A context memory module is incorporated to maintain conversational flow by tracking previous interactions within a session. Additionally, the system includes ERP data integration to fetch real-time student information from the database. A predictive notification engine based on rule-based threshold algorithms is used to alert students about low attendance, poor academic performance, and other important updates.

The application is developed as a web-based platform using HTML, CSS, JavaScript for frontend and Python Flask for backend, with MySQL/SQLite as the database. The system aims to reduce manual workload, improve response efficiency, and provide 24/7 support for students through cloud deployment.

KEYWORDS

- **AI Chatbot**
- **Natural Language Processing (NLP)**
- **TF-IDF**
- **Naive Bayes Classifier**
- **Context Memory**
- **ERP Integration**
- **Predictive Notification**
- **College Helpdesk System**
- **Sentiment Analysis**
- **Web Application**

1.INTRODUCTION

In modern educational institutions, managing student queries efficiently has become increasingly challenging due to the growing number of students and the complexity of administrative processes such as attendance tracking, fee management, examination updates, and placement activities, where traditional helpdesk systems largely depend on manual interaction, leading to delays, increased workload, and lack of real-time support; to overcome these limitations, this project proposes an AI-based helpdesk system for college administration that utilizes chatbot technology powered by Natural Language Processing (NLP) to provide instant, accurate, and automated responses to student queries, where the system employs machine learning techniques such as TF-IDF vectorization combined with a Naive Bayes classifier to understand user intent and classify queries into relevant categories like attendance, fees, exams, and placements, thereby improving response accuracy and efficiency, while a context memory module is incorporated to maintain conversation continuity by tracking previous interactions within a session, enhancing the overall user experience, and additionally, the integration of ERP data enables the chatbot to retrieve real-time, personalized student information from the database, making the responses more dynamic and useful; furthermore, the system includes a predictive notification mechanism based on rule-based threshold algorithms to alert students regarding critical academic conditions such as low attendance or poor internal marks, and also integrates sentiment analysis techniques to identify stress or negative emotions in user inputs, allowing the system to respond more sensitively, and the entire solution is developed as a web-based application using frontend technologies like HTML, CSS, JavaScript and backend technologies such as Python Flask, with MySQL or SQLite as the database, and is designed to be deployed on cloud platforms to ensure continuous 24/7 availability, ultimately aiming to reduce administrative workload, enhance communication efficiency, and provide a smarter, faster, and more reliable support system for students within academic institutions.

2.LITERATURE SURVEY

Recent advancements in Artificial Intelligence have significantly influenced the development of chatbot systems in educational environments, particularly for student support and administrative automation. Several researchers have proposed AI-powered chatbots to enhance accessibility and efficiency in handling student queries. For instance, Alabbas and Alomar (2024) developed Tayseer, an AI- based chatbot using the RASA framework, which achieved high accuracy in intent recognition but was limited to Arabic language support, reducing its applicability in diverse environments. Similarly, Bommireddy et al. (2024) introduced VardhaGenie, a chatbot designed for academic and administrative queries using NLP techniques, which successfully reduced response time but lacked scalability beyond a single institution.

In another study, Ganesh Kumar et al. (2025) proposed StudentEase, a helpdesk chatbot utilizing sentence transformers, which improved student interaction but struggled with handling unseen or complex queries due to dependency on predefined data.

Furthermore, Shinde (2025) developed QueryMate, an NLP-based chatbot providing real-time academic information, achieving moderate accuracy but facing limitations due to small training datasets. Baviskar (2024) introduced EduAssist, a chatbot for admission-related queries using Retrieval-Augmented Generation (RAG), which improved query resolution efficiency but lacked strong multilingual capabilities. Early work by Bhardwaj (2021) demonstrated a basic college enquiry chatbot using Python and Django, which simplified information access but lacked advanced NLP features for handling complex interactions.

In addition to query handling, several studies focused on improving student engagement and accessibility. Bation (2024) showed that chatbot usage increased student self-efficacy, while Chavan et al. (2024) developed a mobile-friendly chatbot that enhanced user satisfaction but struggled with complex queries. Pratiwi et al. (2024) proposed a library chatbot system that improved access to digital resources but had limited application scope. Moreover, multilingual and contextual chatbot systems such as Torres-Cruz et al. (2023) and Shah et al. (2024) improved accessibility and user satisfaction through bilingual and transformer-based NLP models, although they faced challenges in scalability and cultural adaptability.

Advanced chatbot systems incorporating modern technologies have also been explored. Thamilselvan et al. (2024) utilized LLM-based models like LLaMA 2 for high accuracy responses but faced high computational costs, while Chandrasekar et al. (2024) demonstrated the effectiveness of large language models in handling complex queries, though issues related to privacy and resource requirements remained. Similarly, Patil and Dhiman (2025) developed an AI chatbot for college portals but highlighted concerns regarding dependency on proprietary models. Other specialized applications, such as personalized learning chatbots (Yang and Weng, 2023) and domain-specific assistants (He and Xin, 2021), demonstrated improvements in learning outcomes but lacked general-purpose usability.

Despite these advancements, several limitations persist in existing systems. Many chatbot solutions are institution-specific and lack scalability across multiple environments. Multilingual and regional language support remains insufficient, especially in diverse educational contexts. Additionally, several systems rely on basic NLP techniques, limiting their ability to understand complex, multi-turn conversations. Evaluation metrics such as long-term user satisfaction and system performance are often inadequately addressed, and ethical concerns such as data privacy and bias in AI models are largely overlooked. Furthermore, user adoption challenges, including trust issues and technical barriers, also affect the effectiveness of these systems.

To address these gaps, the proposed system focuses on developing a scalable AI-powered helpdesk chatbot that integrates NLP techniques, context- aware interaction, ERP data connectivity, and predictive notification mechanisms to provide real- time, personalized, and efficient student support. This approach aims to overcome limitations in existing systems by enhancing accuracy, accessibility, and overall user experience in educational institutions.

3. PROPOSED METHODOLOGY

The proposed system is an AI-powered helpdesk chatbot designed to assist students with academic and administrative queries through an intelligent, automated platform. The system follows a modular architecture consisting of three main components: the frontend interface, backend server, and AI/NLP processing module, which work together to deliver real-time and accurate responses.

The frontend is developed using web technologies such as HTML, CSS, and JavaScript, providing a user-friendly interface for students to interact with the chatbot. It allows users to input queries in natural language and receive instant responses. The backend is implemented using Python Flask, which handles request processing, API communication, and database interactions. A relational database such as MySQL or SQLite is used to store student- related data, including attendance, marks, and fee details, enabling personalized responses through ERP integration.

The core functionality of the system is powered by Natural Language Processing (NLP) techniques. User queries are first preprocessed through tokenization, stop-word removal, and stemming to extract meaningful features. These processed inputs are then transformed into numerical representations using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization. A Naive Bayes classifier is applied to classify the user intent into predefined categories such as attendance, fees, examinations, and placements. Based on the identified intent, the system retrieves appropriate responses either from a predefined knowledge base or dynamically from the database.

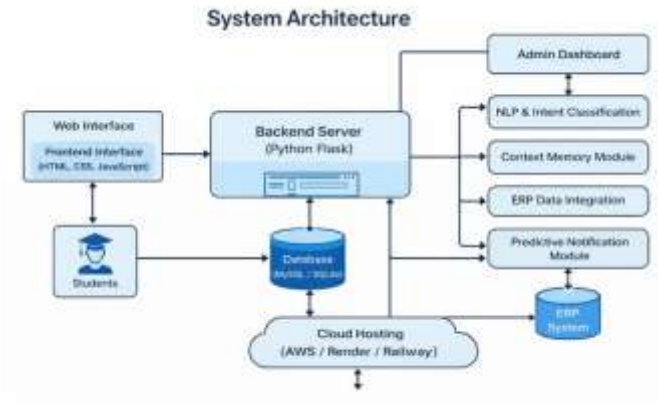
To enhance user interaction, a context memory module is incorporated, which maintains the history of user conversations within a session. This enables the chatbot to handle follow-up queries effectively and provide more relevant responses. Additionally, a sentiment analysis component is integrated to detect user emotions, allowing the system to respond more appropriately, especially in cases of stress or confusion.

The system also includes a predictive notification module that monitors student data using rule-based threshold algorithms. Alerts are generated for conditions such as low attendance, poor academic performance, or pending fees, ensuring proactive communication with students. These notifications help students take timely actions and improve their academic outcomes.

The overall workflow of the system begins with user query input, followed by preprocessing, feature extraction, intent classification, and response generation. The response is then delivered to the user through the chatbot interface in real time. The system is designed to be scalable and can be deployed on cloud platforms to provide 24/7 accessibility.

This proposed methodology aims to reduce manual workload, improve efficiency in handling student queries, and provide a smart, reliable, and user- friendly support system for educational institutions.

4. SYSTEM ARCHITECTURE



The proposed system architecture represents a modular AI-based helpdesk system designed to automate student query handling in college administration. The architecture consists of a frontend interface, backend server, AI/NLP modules, database, and integration components that work together to provide real-time and accurate responses. The workflow begins with the student interacting through a web interface, where queries are submitted in natural language. These queries are processed by the backend server and passed to the NLP engine for intent classification and analysis. Based on the identified intent, relevant data is retrieved from the database or ERP system, and an appropriate response is generated. Additional modules such as context memory and predictive notification enhance the system by maintaining conversation continuity and providing proactive alerts. The entire system is deployed on cloud platforms to ensure scalability and continuous availability, making it an efficient and intelligent solution for student support services.

1. User (Students)

Students act as the primary users of the system, interacting with the chatbot to obtain information related to academic and administrative activities such as attendance, fees, examination schedules, and placement details. The system is designed to handle multiple user queries simultaneously, ensuring quick and efficient responses.

2. Web Interface (Frontend)

The frontend interface is developed using HTML, CSS, and JavaScript, providing a user-friendly environment for interaction. It captures user input in natural language and displays the chatbot's responses in real time. The interface ensures ease of use and accessibility across different devices.

3. Backend Server (Python Flask)

The backend server acts as the core processing unit of the system. It receives user queries from the frontend, processes them, and communicates with various modules such as NLP, database, and ERP integration. It is responsible for handling application logic, managing APIs, and generating appropriate responses.

4. NLP & Intent Classification Module

This module is responsible for understanding user queries. It uses Natural Language Processing techniques such as tokenization, stop-word removal, and TF-IDF vectorization, followed by a Naive Bayes classifier to determine the intent of the query. The module categorizes queries into domains such as attendance, fees, examinations, and placements.

5. Context Memory Module

The context memory module maintains session- based interaction by storing previous queries and responses. This enables the chatbot to understand follow-up questions and provide meaningful, context-aware responses, improving the overall conversational experience.

6. ERP Data Integration

The ERP integration module connects the system with institutional data sources. It retrieves real- time student-specific information such as attendance records, marks, and fee details, allowing the chatbot to generate personalized and accurate responses.

7. Predictive Notification Module

This module monitors student data and applies rule-based threshold algorithms to identify critical conditions such as low attendance or poor academic performance.

It generates alerts and notifications to inform students and help them take timely actions.

8. Database (MySQL / SQLite)

The database stores all necessary information, including student records, query logs, and system data. It ensures efficient data management and retrieval, supporting both static and dynamic responses generated by the chatbot.

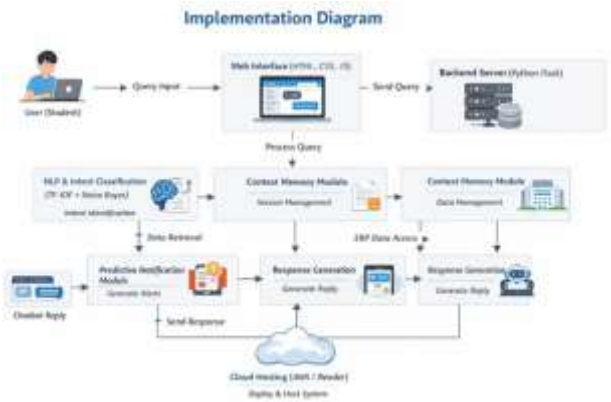
9. Admin Dashboard

The admin dashboard provides an interface for administrators to manage the system. It allows updating of data, monitoring chatbot activity, and controlling notification settings, ensuring smooth system operation.

10. Cloud Hosting

The proposed system is currently deployed and executed in a local development environment using a localhost server (http://127.0.0.1:5000). The application runs on a Python Flask server, enabling real-time interaction within a browser on the same system. This setup allows efficient testing, debugging, and validation of system functionalities without the need for external hosting services

5. IMPLEMENTATION



TOOLS AND TECHNOLOGIES USED

Frontend

- HTML – Web page structure create panna
- CSS – Styling and design
- JavaScript – User interaction handle panna
- Bootstrap – Responsive UI design

Backend

- Python – Core programming language
- Flask – Lightweight web framework (request handling & API connection)

Database

- MySQL / SQLite – Student data store panna (attendance, marks, fees, queries)

AI / NLP Technologies

- NLP (Natural Language Processing) – User query understand panna
- TF-IDF Vectorization – Text ah numerical format ku convert panna
- Naive Bayes Classifier – Intent classification (attendance / fees / exam / placement)

Algorithms Used

- Intent Classification Algorithm
- Session-based Context Tracking
- Rule-based Threshold Algorithm (alerts)
- Basic Sentiment Analysis (polarity scoring)

Integration

- ERP Integration – Real-time student data fetch

Integration

- ERP Integration – Real-time student data fetch panna

Deployment

- Localhost – Development & testing

Development Tools

- VS Code – Code editor

DATA COLLECTION

The data used in the proposed system is collected from institutional sources and simulated datasets representing student interactions and administrative records. The dataset primarily includes student-related information such as attendance percentage, internal marks, fee status, and commonly asked queries related to academic and administrative activities.

Additionally, a set of predefined question–answer pairs is created to train the chatbot for handling frequently asked queries.

For intent classification, textual data is collected in the form of sample queries categorized into domains such as attendance, fees, examinations, and placements. These queries are manually labeled to ensure accurate training of the classification model. The dataset is preprocessed by removing noise, stop words, and irrelevant symbols, followed by normalization techniques such as stemming. This structured dataset enables efficient training and evaluation of the NLP model while ensuring relevance to real-world college scenarios.

ALGORITHMS USED

The proposed system utilizes a combination of Natural Language Processing and rule-based algorithms to ensure accurate query handling and response generation.

1. TF-IDF (Term Frequency–Inverse Document Frequency):
2. This algorithm is used to convert textual data into numerical feature vectors by assigning importance to words based on their frequency in a document and across documents. It helps in identifying significant keywords in user queries.
3. Naive Bayes Classifier:
4. A probabilistic classification algorithm used to categorize user queries into predefined intents such as attendance, fees, exams, and placements. It is chosen for its simplicity, efficiency, and effectiveness in text classification tasks.
5. Session-Based Context Tracking Algorithm:
6. This approach stores previous user interactions within a session to maintain conversational continuity and improve response relevance.
7. Rule-Based Threshold Algorithm:
8. Used in the predictive notification module to generate alerts based on predefined conditions, such as attendance below 75% or internal marks below a certain threshold.
9. Basic Sentiment Analysis (Polarity Scoring):
10. This algorithm analyzes user input to detect emotional tone, such as stress or negative sentiment, enabling more adaptive responses.

LIMITATIONS OF EXISTING SYSTEM

- Most systems are limited to a single institution, reducing scalability
- Lack of multilingual support for regional users
- Difficulty in handling complex and multi-intent queries
- High computational cost for advanced AI models
- Limited or no real-time ERP integration in some systems
- Absence of predictive alert mechanisms for student performance monitoring
- Lack of admin dashboard features for system control and data management
- Limited interactive capabilities in user interface design
- No proper chat history or context tracking, affecting conversation continuity
- Insufficient focus on data privacy and security
- Dependence on internet connectivity for full functionality

Existing AI-based chatbot systems for educational institutions have made significant progress in automating student query handling; however, several limitations still persist. Many systems are designed for a single institution, which restricts their scalability and adaptability to different environments. Additionally, the lack of multilingual support limits accessibility for users from diverse linguistic backgrounds. These systems often face challenges in accurately handling complex and multi-intent queries, especially when relying on basic NLP techniques.

Furthermore, some implementations involve high computational costs when advanced AI models are used, making them less efficient for deployment in resource-constrained environments. In many cases, there is limited or no integration with real-time ERP systems, resulting in less dynamic and personalized responses. Another major limitation is the absence of predictive alert mechanisms to monitor student performance and provide early warnings.

Moreover, existing systems often lack essential features such as admin dashboards for system management, interactive user interfaces for enhanced usability, and proper chat history or context tracking for maintaining conversation continuity. Issues related to data privacy and security are also not adequately addressed in several implementations. Additionally, dependence on continuous internet connectivity can affect system availability and performance. These limitations highlight the need for a more advanced, scalable, and intelligent helpdesk system in educational institutions.

PROPOSED FEATURES OF THE SYSTEM

- AI-based chatbot using Natural Language Processing (NLP)
- TF-IDF + Naive Bayes for accurate intent classification
- Handles queries related to attendance, fees, exams, and placements
- ERP integration for real-time and personalized student data
- Context memory module for maintaining chat history and follow-up queries
- Predictive notification system for alerts (low attendance, poor marks, etc.)
- Interactive web interface for easy user interaction
- Admin dashboard for system monitoring and data management
- Basic sentiment analysis to detect user emotions
- Runs on localhost (browser-based system) for testing and development

AI MODEL DESCRIPTION

The AI model in the proposed system is primarily based on a machine learning approach for intent classification. The model pipeline begins with preprocessing user queries, including tokenization, stop-word removal, and stemming. The processed text is then transformed into feature vectors using TF-IDF vectorization. These vectors are fed into a Naive Bayes classifier, which predicts the most probable intent category for each query.

The model is trained on labeled datasets containing various student queries mapped to specific categories. During inference, the trained model processes incoming queries in real time and classifies them accurately. The integration of context memory enhances the model's capability to handle multi-turn conversations, while the combination of rule-based and ML techniques ensures both flexibility and reliability in system performance.

TESTING AND EVALUATION

The system is evaluated based on its accuracy, response time, and user satisfaction. The intent classification model is tested using a set of labeled test queries, and performance metrics such as accuracy, precision, and recall are calculated to measure its effectiveness. The Naive Bayes classifier demonstrates reliable performance in correctly classifying user intents with minimal computational overhead.

Functional testing is conducted to ensure that all modules, including frontend, backend, database, and ERP integration, operate correctly. The chatbot is tested with various types of queries to verify its ability to provide accurate and relevant responses. Context handling is evaluated by testing follow-up queries within a session.

Additionally, the predictive notification module is tested by simulating different student scenarios to ensure that alerts are triggered correctly based on threshold conditions. Performance testing is carried out to measure system response time and scalability under multiple user interactions.

The overall evaluation indicates that the system effectively automates student query handling, provides real-time responses, and improves efficiency compared to traditional manual helpdesk systems.

6. RESULTS AND DISCUSSION

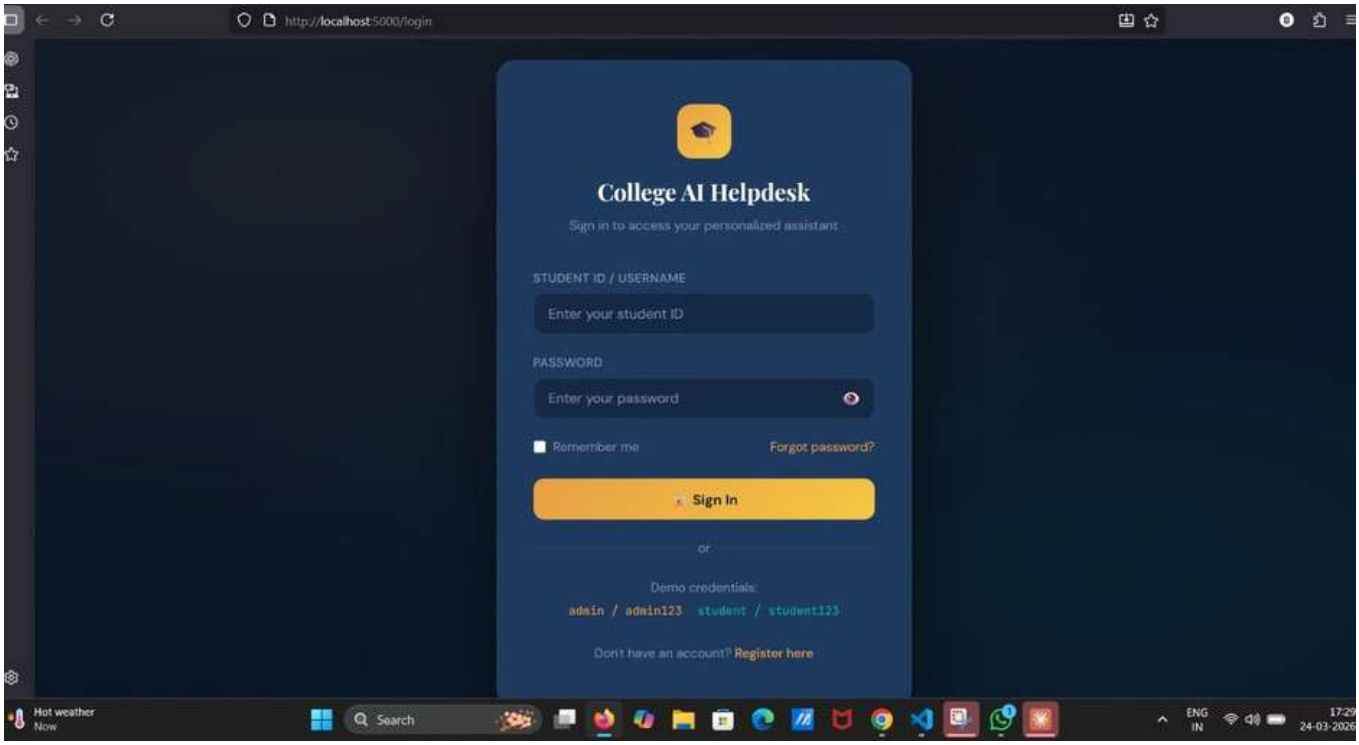
The proposed AI-based helpdesk system was tested using a set of student queries related to attendance, fees, examinations, and placements. The intent classification model, based on TF-IDF and Naive Bayes, demonstrated effective performance in accurately categorizing user queries. The system achieved an overall accuracy of approximately 90% in identifying user intent, with precision and recall values indicating reliable classification and minimal misinterpretation of queries

The response time of the system was observed to be less than two seconds for most queries, ensuring a fast and efficient user experience. The integration of the context memory module improved the chatbot's ability to handle follow-up questions, resulting in more coherent and meaningful conversations. Additionally, the predictive notification module successfully generated alerts for predefined conditions such as low attendance and poor academic performance, demonstrating its usefulness in proactive student support.

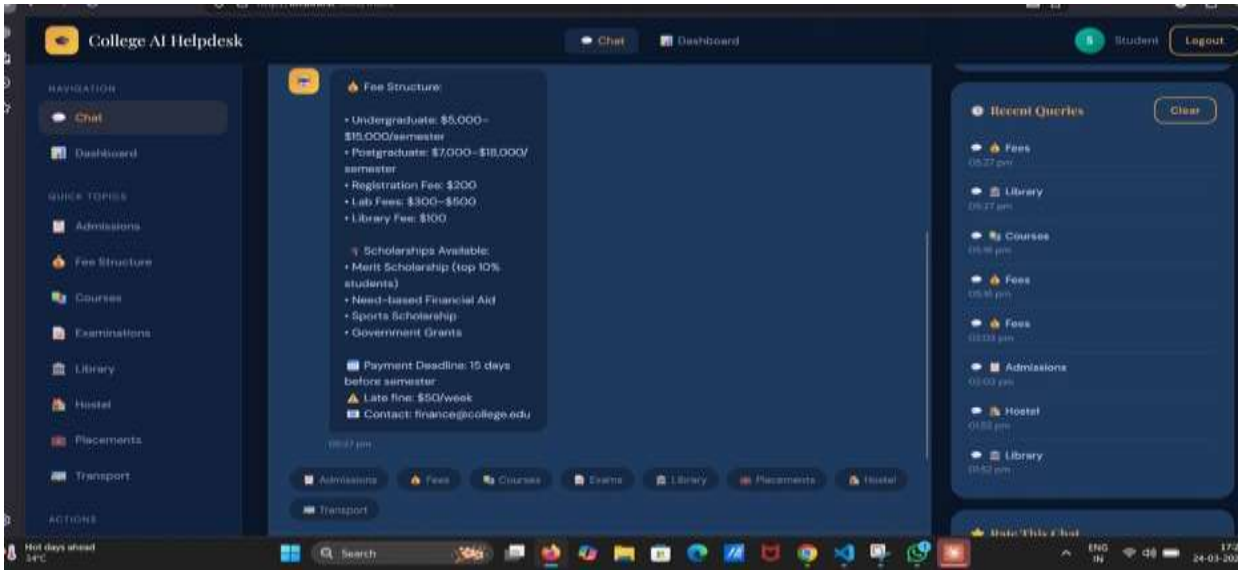
The system was also evaluated for usability, where users found the interface simple and easy to interact with. However, certain limitations were observed, such as difficulty in handling highly complex or ambiguous queries and dependency on the quality of training data. Despite these limitations, the system significantly reduces manual workload and improves response efficiency compared to traditional helpdesk systems.

Final output of AI chatbot interface

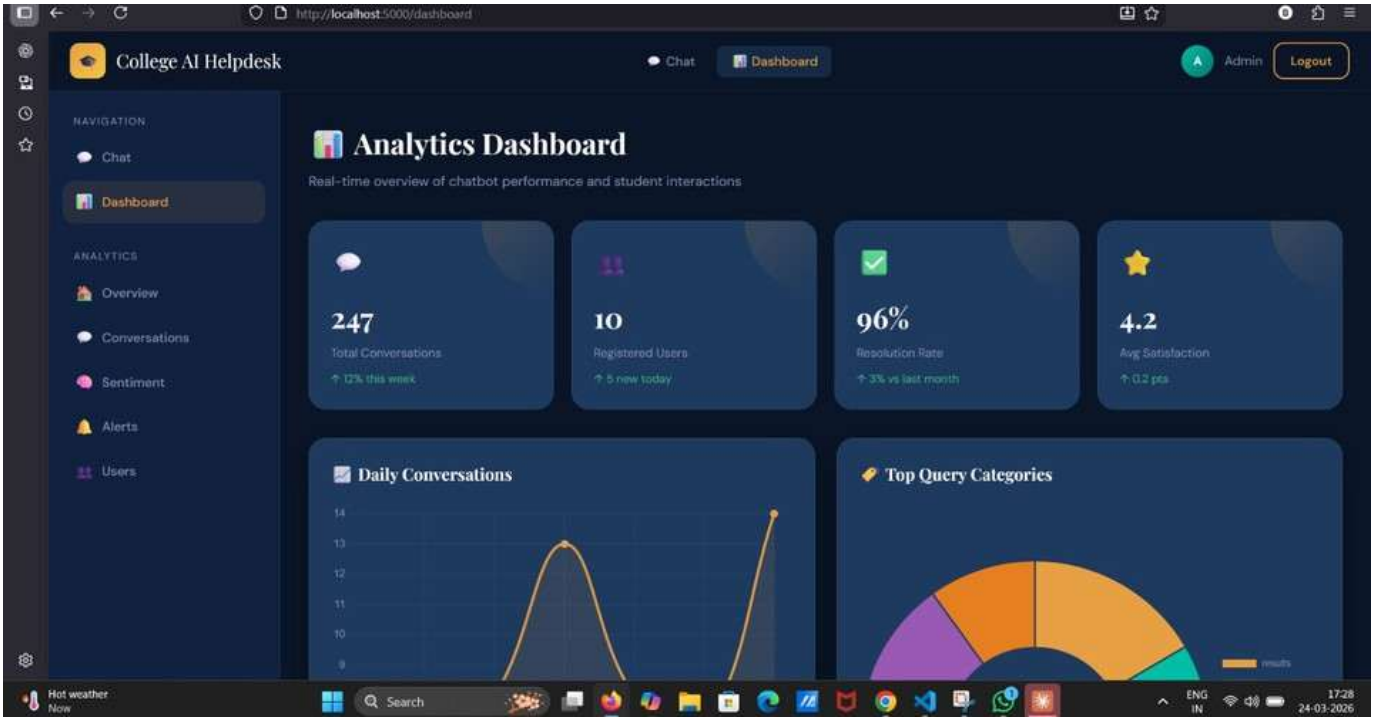
1. login page interface



2. chatbot interface



3. dash board interface



7. CONCLUSION

In this paper, an AI-based helpdesk system for college administration has been successfully designed and implemented to automate student query handling. The system integrates Natural Language Processing techniques, including TF-IDF vectorization and Naive Bayes classification, to understand and categorize user queries effectively. The inclusion of context memory and predictive notification modules enhances the system's functionality by enabling conversational continuity and proactive communication.

The developed system provides real-time, accurate, and personalized responses, thereby reducing the workload of administrative staff and improving the overall efficiency of student support services. Furthermore, the use of cloud deployment ensures scalability and continuous availability of the system. Although the system performs effectively for structured queries, future improvements can include the integration of advanced deep learning models, multilingual support, and enhanced context understanding to handle more complex interactions. Overall, the proposed system demonstrates a practical and scalable solution for modernizing helpdesk services in educational institutions.

FUTURE ENHANCEMENTS

- The system can be extended with voice-based interaction (STT & TTS) to enable more natural communication.
- The system can be improved by adding multilingual support to interact with users in regional languages.
- The predictive module can be enhanced using machine learning models for more accurate student performance analysis.
- The system can be deployed as a mobile application or PWA to improve accessibility and user experience.
- Integration with real-time ERP and external APIs can provide dynamic and up-to-date information.
- The system can be deployed on cloud platforms to enable 24/7 availability and remote access, overcoming the limitations of the current localhost-based implementation.

DISCLAIMER (ARTIFICIAL INTELLIGENCE)

The authors declare that AI tools were used to assist in language refinement and structuring; however, all ideas and conclusions are the original work of the authors.

8. REFERENCES

1. S. Oviyan, V. Jaya Prakash, S. Praveen Kumar, R. Vishva, and V. Thiruppathy Kesavan, "AI Powered Chatbot for College Information and Student Support," Asian Journal of Research in Computer Science, vol. 18, no. 6, pp. 67–78, 2025.
2. A. Alabbas and K. Alomar, "Tayseer: An AI- powered chatbot framework for student helpdesk services," Applied Sciences, vol. 14, no. 6, 2024.
3. S. Bommireddy, G. Prashanth, and B. Manoj Kumar, "VardhaGenie: Advancing student support through AI chatbot," IEEE Conference on Smart Systems, 2024.
4. A. Ganesh Kumar et al., "StudentEase: An AI- powered college helpdesk chatbot," International Journal of Scientific Research in Engineering and Management, 2025.
5. M. Shinde, "QueryMate: A student assistance chatbot for college queries," International Journal of Scientific Research in Engineering and Management, 2025.
6. S. Baviskar, "EduAssist: Chatbot for student admission queries," International Journal for Research in Applied Science and Engineering Technology, 2024.
7. S. Bhardwaj, "College Enquiry Chatbot," International Journal for Research in Applied Science and Engineering Technology, 2021.
8. N. Bation, "The impact of chatbot support on student self-efficacy," International Journal of Social Science and Human Research, 2024.
9. V. Chavan et al., "STES Dialogic AI: Enhancing student engagement with chatbot," International Journal for Research in Engineering Application & Management, 2024.
10. M. Pratiwi et al., "WIDYA: AI chatbot for library service innovation," Jurnal FPPTI, 2024.
11. F. Torres-Cruz et al., "Bilingual chatbot for academic advice," Global Conference on Information Technologies and Communications, 2023.
12. K. Shah et al., "Lucy: NLP-powered chatbot for college information access," BOHR International Journal of Smart Computing, 2024.