CHATBOT SYSTEM FOR COLLEGE INFORMATION AND SUPPORT

PROJECT REPORT

Submitted by

SHRIGAYATHRI S (221CS309)

DHARANI M (221EE108)

In partial fulfilment for the award of the degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

BANNARI AMMAN INSTITUTE OF TECHNOLOGY
(An Autonomous Institution Affiliated to Anna University, Chennai)
SATHYAMANGALAM-638401

ANNA UNIVERSITY: CHENNAI 600 025

OCTOBER 2025

BONAFIDE CERTIFICATE

Certified that this project report **“CHATBOT SYSTEM FOR COLLEGE INFORMATION AND SUPPORT”** is the Bonafide work of **"SHRIGAYATHRI S (221CS309) and DHARANI M (221EE108)"** who carried out the project work under my supervision.

Dr. Sasikala D
HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Ms. Priyanga M A
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANNARI AMMAN INSTITUTE OF TECHNOLOGY

Submitted for Project Viva Voice examination held on……

Internal Examiner 1 **Internal Examiner**

DECLARATION

We affirm that the project work titled **"CHATBOT SYSTEM FOR COLLEGE INFORMATION AND SUPPORT"** being submitted in partial fulfillment for the award of the degree of **Bachelor of Engineering** in **Computer Science** is the record of original work done by us under the guidance of **Ms Priyanga M A**, Assistant Professor**,** Department of Computer Science and Engineering. It has not formed a part of any other project work(s) submitted for the award of any degree or diploma,either in this or any other University.

SHRIGAYATHRI S **DHARANI M**

(221CS309) **(221EE108)**

I certify that the declaration made above by the candidates is true.

(Signature of the Guide)

Ms. PRIYANGA M A

ACKNOWLEDGEMENT

We would like to enunciate heartfelt thanks to our esteemed Chairman **Dr. S.V. Balasubramaniam**, and the respected Principal **Dr. C. Palanisamy,** for providing excellent facilities and support during the course of study in this institute.

We are grateful to **Dr. Sasikala D**, **Head of the Department, Department of Computer Science and Engineering,** for his valuable suggestions to carry out the project work successfully.

We wish to express our sincere thanks to Faculty guide **Ms. Priyanga M A, Assistant Professor**, **Department of Computer Science and Engineering,** for her constructive ideas, inspirations, encouragement, excellent guidance, and much needed technical support extended to complete our project work.

We would like to thank our friends, faculty and non-teaching staff who have directly and indirectly contributed to the success of this project.

SHRIGAYATHRI S (221CS309)

DHARANI M (221EE108)

ABSTRACT

Advances in artificial intelligence have opened new opportunities in higher education, especially in areas that involve direct interaction with students. One such application is the use of chatbots to act as virtual assistants that provide reliable academic and administrative support. This project presents the development of a chatbot specifically designed for college environments, with the primary objective of delivering quick, consistent, and user-friendly responses to student queries. Unlike traditional methods of information delivery that depend on staff availability, the proposed chatbot ensures round-the-clock accessibility and reduces delays in communication.

The system was designed using a structured methodology that included requirement gathering, modular design, iterative implementation, and continuous testing. Natural language processing techniques were employed to interpret student inputs and provide meaningful responses. The chatbot was configured to handle common inquiries such as course details, examination schedules, fee payments, and campus services, while also directing students to additional resources where needed. By following this approach, the chatbot evolved into an efficient knowledge-based support tool capable of reducing the workload of faculty and administrative staff.

Evaluation of the prototype showed that the system successfully improved response speed, accuracy of information delivery, and overall student satisfaction. Comparisons with existing studies confirmed that the proposed solution performs well in terms of usability and efficiency, highlighting its potential as a cost-effective support mechanism in academic institutions (Bedford, 2017; Davis et al., 2015). A cost–benefit analysis further indicated that once implemented, the chatbot can reduce administrative expenses while offering long-term scalability and reliability. This work demonstrates that AI-enabled chatbots can play a vital role in enhancing academic support services within colleges. While the present system focuses on frequently asked questions and basic guidance, future improvements may include personalized learning recommendations, integration with learning management platforms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In today's educational environment, colleges and universities are increasingly dependent on technology to manage academic and administrative services. Students often require immediate access to information related to courses, schedules, fees, faculty details, and general campus facilities. Traditionally, this has been handled through enquiry desks, notice boards, or static websites, which are not only time-consuming but also lack interactive features. Students may have to physically visit offices, make repeated phone calls, or search through multiple web pages to obtain even simple information. This results in delays, inefficiency, and dissatisfaction for both students and staff.

With the rapid advancement of artificial intelligence and natural language processing (NLP), conversational agents such as chatbots have emerged as a practical solution to bridge this gap. Chatbots are designed to simulate human-like conversations and provide relevant responses to user queries in real time. By implementing such systems in academic institutions, it is possible to ensure round-the-clock accessibility, minimize manual workload, and enhance student engagement. Unlike traditional methods, a chatbot can respond instantly, operate continuously, and improve its accuracy through user interaction and feedback.

The proposed project, Chatbot System for College Information and Support, aims to address these challenges by developing an intelligent, user-friendly chatbot that supports both text and voice interactions. Built using React.js for the front end, Flask for backend integration, and Rasa for natural language understanding, the system provides seamless access to information stored in a connected database. The chatbot is designed to support multilingual queries, thereby accommodating a diverse student population, and it is capable of storing chat logs for monitoring and performance analysis. The solution ensures better communication between the institution and

students, reduces dependency on manual enquiry systems.

1.1 BACKGROUND OF THE WORK

In many academic institutions, student interaction with the administration is limited by time and resources. A student with a query about admission procedures, fee structures, examination schedules, or campus facilities often has to rely on manual enquiry desks or official circulars. Even when digital solutions such as websites and FAQ pages are available, they usually contain static information that may not always be updated. Moreover, these systems are not interactive, leaving students to interpret and search for the relevant details themselves.

The gap between available resources and student expectations has created the need for smarter solutions. Chatbots, which are conversational systems powered by artificial intelligence, provide a promising alternative. They can interpret natural language, identify user intent, and retrieve relevant information from a connected database. By offering instant responses in both voice and text formats, they overcome the limitations of static systems. Furthermore, chatbots can be deployed on multiple platforms, such as web browsers and mobile devices, making them accessible to students anytime and anywhere.

The background of this work also highlights the importance of personalization and adaptability in student support systems. Unlike rule-based bots that can only handle a limited set of predefined questions, modern NLP-driven chatbots are capable of understanding varied user inputs. This ensures that even if a student frames a query in an unconventional manner, the system can still provide a meaningful response. Additionally, with multilingual support, students from different linguistic backgrounds can comfortably interact with the system, thereby promoting inclusivity within the institution.

Overall, the background emphasizes that the integration of artificial intelligence into academic enquiry systems is not merely a technological upgrade but a necessity to keep pace with the growing expectations of students and staff.

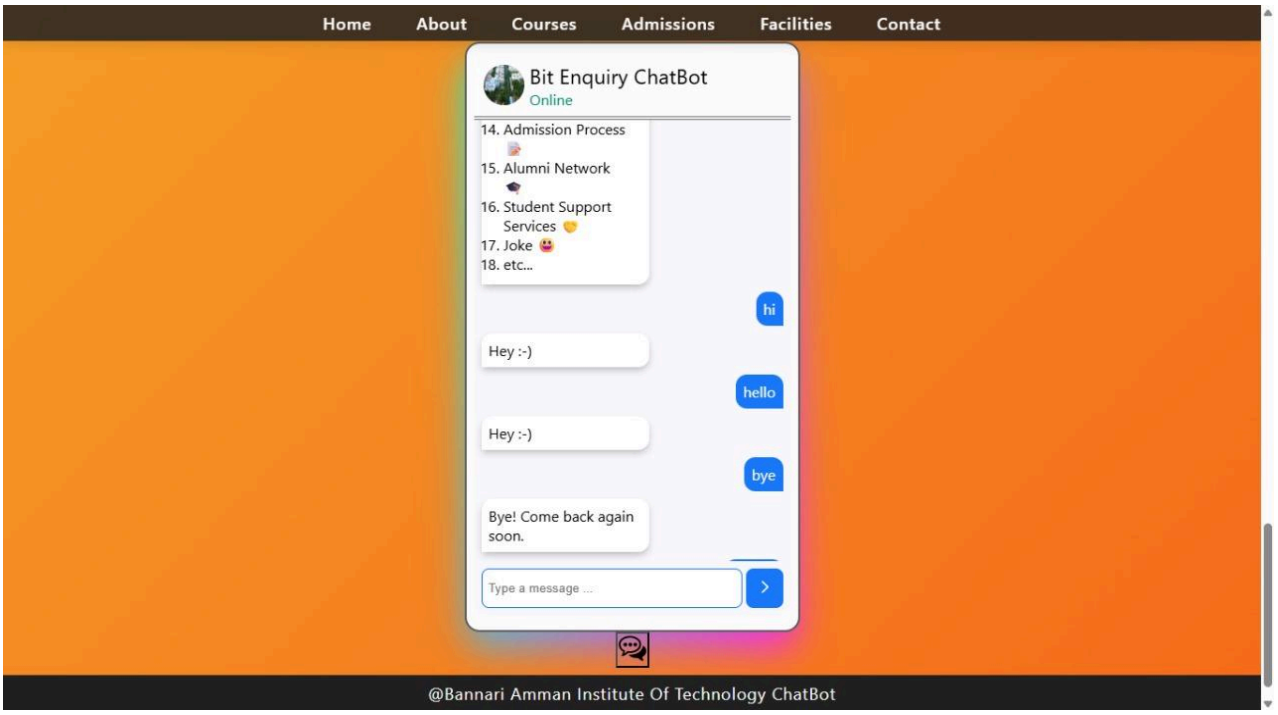

Figure 1.1 Chatbot view

1.2 MOTIVATION(SCOPE OF THE PROPOSED WORK)

The motivation for this project originates from the shortcomings of existing information systems in colleges. Current solutions either rely on manual processes or offer limited digital features. Enquiry desks require staff to be physically present, leading to delays and additional workload. Websites and FAQ pages, although available, are static in nature and often fail to provide real-time updates. Moreover, very few existing systems incorporate voice-based queries or multilingual capabilities, which are essential in diverse academic environments.

To overcome these limitations, this project proposes a comprehensive chatbot solution that integrates multiple technologies to provide an efficient, user-friendly platform. The chatbot will be capable of:

1. Handling both text and voice queries for better accessibility.

2. Using Rasa NLU for intent detection and entity recognition, ensuring accurate interpretation of varied queries.

3. Storing information in a MongoDB database, which allows for real-time updates and reliable data retrieval.

4. Offering multilingual support through translation APIs, ensuring inclusivity for students from different linguistic backgrounds.

5. Maintaining chat history and analytics, enabling administrators to track performance and make improvements over time.

The scope of the proposed work extends beyond a simple question-answer system. The chatbot can be scaled and integrated with platforms such as WhatsApp, Telegram, or mobile applications, allowing students to interact through the channels they use most frequently. Additionally, the system can evolve to include advanced features such as academic reminders, personalized notifications, and integration with learning management systems.By implementing this system, colleges can significantly reduce administrative workload, improve efficiency, and ensure that students receive timely and accurate support. The project's motivation lies in creating a scalable, intelligent, and adaptive system that not only solves the present challenges but also lays the foundation for future enhancements in academic support services.

CHAPTER 2

LITERATURE SURVEY

Chatbots have become an integral part of modern digital services across education, healthcare, customer support, and business sectors. In recent years, academic institutions have increasingly shown interest in adopting intelligent chatbot systems to handle student queries, automate routine tasks, and improve accessibility. A literature survey was carried out to review the state of the art in chatbot development, particularly in the context of educational environments. This survey includes relevant works from the last five years, highlights their contributions, and identifies existing gaps that motivated the present study.

2.1 AI-POWERED CHATBOTS IN HIGHER EDUCATION

Jayraj et al. (2024) presented a college inquiry chatbot system designed to address routine academic queries of students. The study highlighted how such systems reduce the workload of administrative staff and provide round-the-clock service. The chatbot was integrated with a college database and was capable of responding to frequently asked questions such as course details, admission processes, and fee structures. However, the system largely depended on predefined responses and lacked adaptability when queries were framed in different formats.

Similarly, Kumar et al. (2024) developed KatzBot, an academic chatbot designed to facilitate enhanced communication within educational institutions. The bot employed natural language understanding (NLU) models to classify user queries and provide relevant responses. While the system showed promise in offering accurate information, it lacked multilingual capabilities and did not support voice-based interaction, thereby limiting its accessibility for diverse student populations.

2.2 CHATBOTS WITH NLP AND MACHINE LEARNING APPROACHES

Patil et al. (2023) explored the application of machine learning-based intent detection to improve chatbot performance in academic environments. Their system could identify student intent from free-text queries with a reasonable degree of accuracy. However, the study noted that entity extraction remained a challenge, particularly when queries involved specific details such as course codes or faculty names.

In another study, Deshmukh & Garad (2023) implemented a rule-based chatbot integrated with a FAQ system. While this approach ensured reliability for standard queries, it struggled when users posed questions in an unexpected format or in regional languages. This emphasized the need for more robust, AI-driven solutions that could handle varied and natural conversational input.

2.3 MULTILINGUAL AND VOICE-ENABLED CHATBOTS

`Sharma et al. (2022) introduced a multilingual chatbot aimed at supporting students in institutions with diverse linguistic backgrounds. The system used translation APIs to convert queries into English before processing, and then translated the output back into the user's preferred language. Although effective, the system faced latency issues, and the accuracy of machine translation sometimes caused errors in interpretation.

On the other hand, Gupta & Nair (2021) designed a voice-enabled chatbot for student support services. Their system allowed students to interact via speech, making it more accessible for those less comfortable with typing. While the addition of speech recognition enhanced usability, the system was limited by accent variations and background noise, leading to occasional misinterpretations.

2.4 COMPARATIVE STUDIES ON CHATBOTS IN EDUCATION

Bushnell (2022) conducted a study comparing chatbots used in small-scale organizations and educational institutions. The research showed that most systems were limited to simple Q&A functions and lacked integration with institutional databases. The author emphasized the importance of combining chatbot systems with backend databases to ensure the accuracy and relevance of information.

Likewise, Mishra & Korpal (2021) examined the drawbacks of manual student enquiry systems and compared them with chatbot alternatives. They concluded that manual systems are prone to delays, errors, and excessive paperwork, whereas chatbots significantly streamline the process. However, they also noted that most chatbot systems deployed in colleges still lacked adaptive learning capabilities and real-time updates.

AUTHOR & YEAR	FOCUS AREA	FEATURES IMPLEMENTED	STRENGTHS	LIMITATIONS / GAPS IDENTIFIED
Jayraj et al., 2024	College inquiry chatbot	FAQ-based responses, database link	24×7 availability; reduces admin workload	Limited to predefined responses
Kumar et al., 2024	KatzBot – Academic bot	NLU for intent classification	Accurate communication with users	Lacks multilingual & voice support
Sharma et al., 2022	Multilingual chatbot	Translation API integration	Supports diverse linguistic backgrounds	Translation delays & accuracy issues
Gupta & Nair, 2021	Voice-enabled chatbot	Speech-to-text interaction	More user-friendly, hands-free interaction	Sensitive to accents & background noise

Table 1 – Comparison of Existing Chatbot Systems in Education

CHAPTER 3

OBJECTIVES AND METHODOLOGY

3.1 INTRODUCTION

The creation of a chatbot system for college information and support involves a structured process of planning, development, and evaluation. Unlike conventional information systems, the present work integrates natural language processing, multilingual support, and real-time database connectivity. The methodology was designed to align with the project objectives while also addressing the challenges identified in the literature survey. This chapter explains the objectives in detail, presents the workflow of the proposed system, describes each functional block of the design, and justifies the choice of tools, components, and techniques used. The overall methodology combines user-centered design, iterative development, and rigorous testing to ensure the chatbot functions as an intelligent and reliable support system for students and faculty.

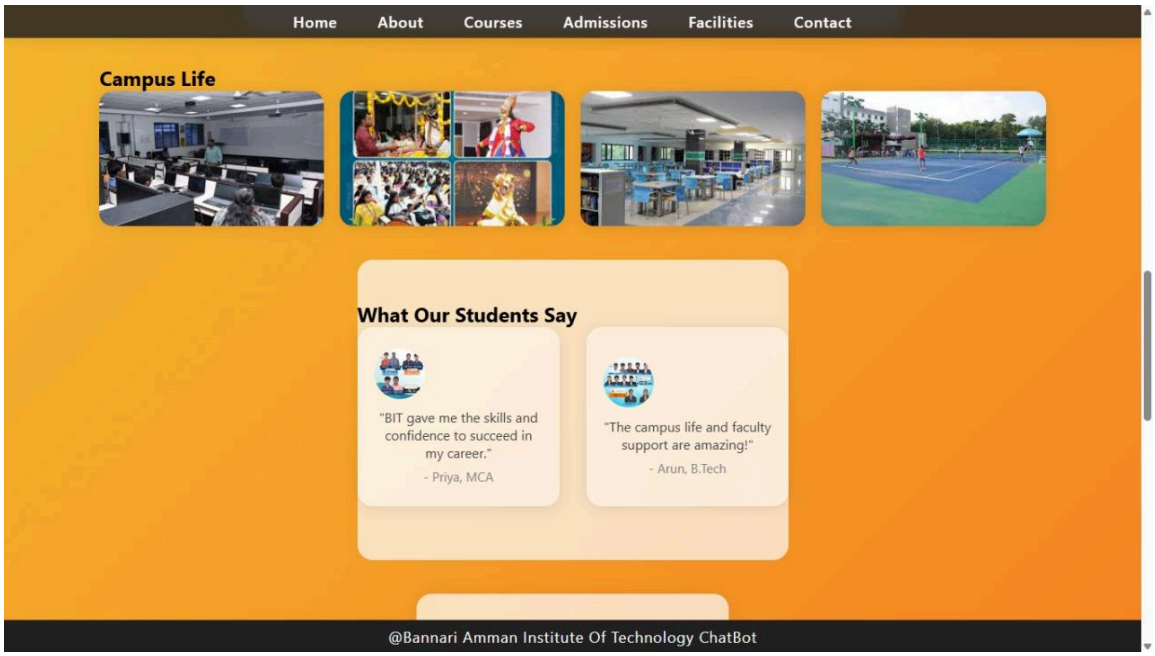

Figure 3.1 College website overview

Another defining feature of this methodology is that it places the user at the center of design decisions. The student experience is considered at every stage, from interface design to multilingual support. This aligns with user-centered design practices, which emphasize that technology should adapt to the user rather than forcing the user to adapt to the technology. By embedding these principles into the methodology, the project ensures that the chatbot is not just a demonstration of artificial intelligence but also a functional service that addresses the real demands of academic environments.

3.2 OBJECTIVES OF THE PROPOSED WORK

The project objectives were framed in response to the gaps observed in previous research and existing systems. The first objective was to develop a user-friendly interface that allows students to interact with the chatbot in both text and voice modes. Accessibility and responsiveness were prioritized to ensure that the chatbot could be used comfortably on desktops, laptops, and mobile devices.

The second objective was to implement an intelligent natural language processing engine capable of handling varied queries. The system was designed to go beyond simple keyword matching and instead employ intent recognition and entity extraction to understand user queries in a contextual manner. This objective was crucial to provide flexibility and natural interactions, ensuring that students can phrase their questions in different ways without receiving irrelevant responses.

The third objective focused on backend integration and database support. The chatbot was required to fetch updated information from a central database and respond with accuracy. This demanded a scalable backend system capable of linking the NLP engine with a reliable data store. Flask was chosen to implement the backend layer, while MongoDB was selected for its flexible and document-oriented structure, which suits the diverse nature of academic data.

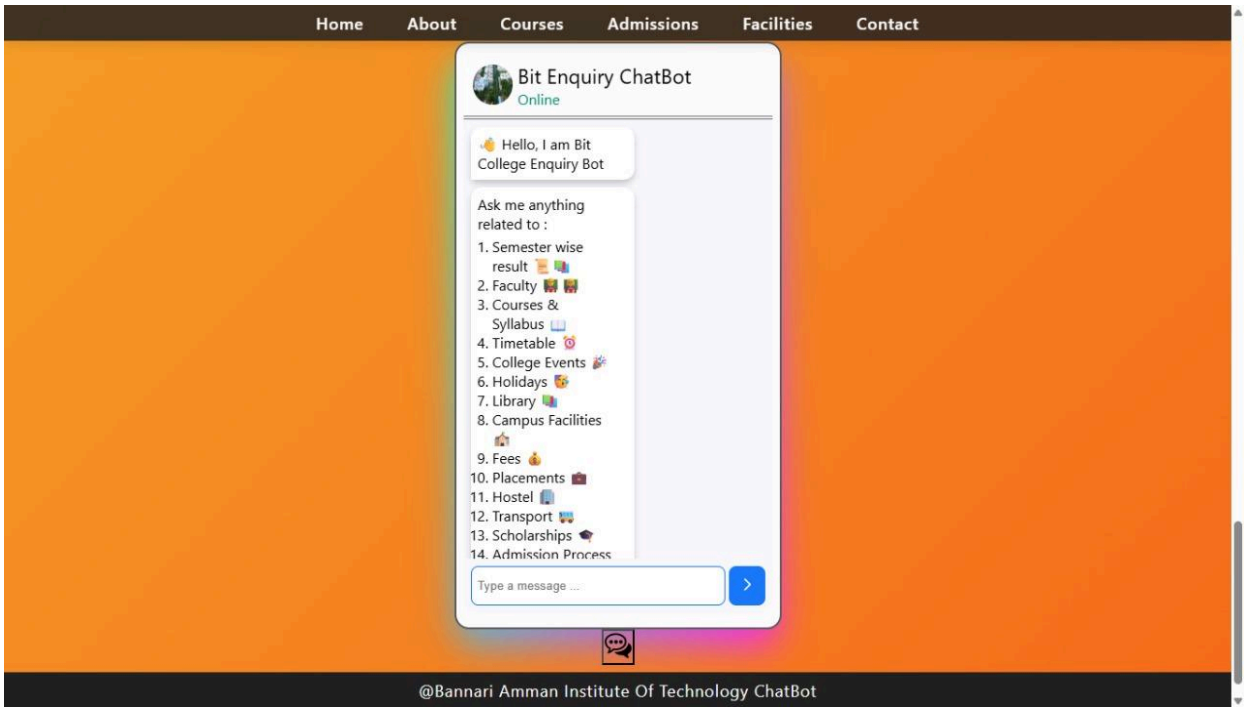

Figure 3.2 Dynamic Answer support

The fourth objective was to incorporate multilingual and voice-based features. Since colleges serve students from multiple regions and linguistic backgrounds, a system limited to English text input would not achieve inclusivity. By integrating translation APIs and speech recognition technologies, the chatbot was designed to support queries in multiple languages and provide voice output for added convenience.

The fifth objective involved building an analytics and feedback module. A chatbot cannot remain static; it must evolve with user needs. Therefore, performance monitoring, feedback collection, and analysis were incorporated into the methodology. This module enables administrators to refine the system, retrain models, and improve accuracy over time.

3.3 METHODOLOGICAL FRAMEWORK

The methodological framework of this project serves as the backbone of development, outlining the systematic approach adopted to design and implement the chatbot system for college information and support. Unlike traditional software projects, where the flow of development may be strictly linear, this project follows an iterative and modular framework that allows flexibility, continuous testing, and integration of user feedback. The framework ensures that the chatbot is not only technically functional but also aligned with the actual needs of students and academic staff.

The framework begins with problem identification and requirement analysis. During this stage, the existing systems in educational institutions were carefully studied, and their shortcomings were documented. Most colleges still rely on enquiry counters or static web portals, which are unable to provide real-time support or interactive communication. From this analysis, the essential requirements of the proposed chatbot were defined: the system had to operate 24×7, handle both voice and text queries, support multiple languages, and provide accurate and timely responses. This early stage shaped the later design and implementation process by clearly setting the scope and direction of the work.

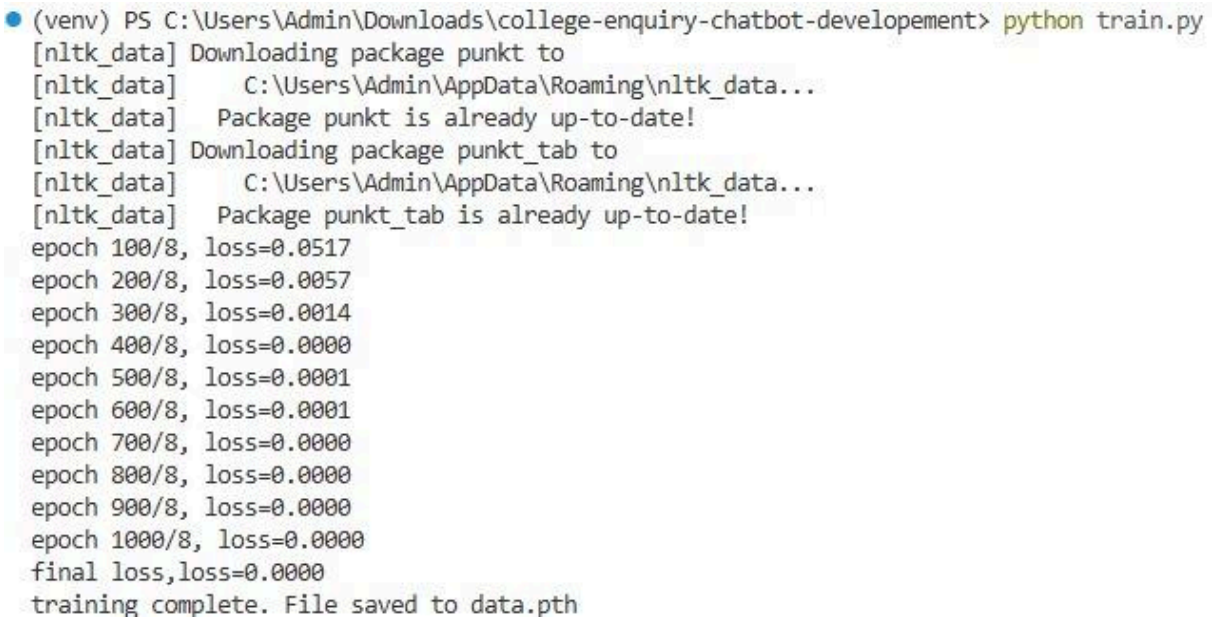

Figure 3.3 Model training

Once the requirements were formalized, the framework progressed to the design phase, which focused on modular architecture. Each core function of the chatbot was treated as an independent block: the user interface, the natural language processing engine, the backend server, the database, and the multilingual and analytics modules. This modular approach allowed the system to be designed in a way that is scalable and adaptable. For instance, if in the future the NLP engine needs to be replaced with a more advanced model, the replacement can be done without redesigning the entire system. Similarly, the backend and database modules can be upgraded independently. Such modularity is essential for long-term sustainability in educational environments, where systems often need to evolve to accommodate new requirements.

The next step in the methodological framework was the development cycle, which followed an iterative approach inspired by Agile practices. Instead of building the chatbot in a single linear process, the system was developed in smaller cycles. Each module was designed, implemented, tested, and refined before being integrated into the larger system. This cycle ensured that errors could be detected early and that each module was functional before integration. It also allowed user feedback to be incorporated at multiple stages. For example, early versions of the user interface were tested with students to gather feedback on usability, which was then used to refine the design before final deployment.

Another important element of the framework is the flow of data between modules. The framework defines a pipeline beginning with user input and ending with the system response. When a query is entered through the interface, it is first pre-processed and then passed to the NLP engine. The engine interprets the intent and entities contained in the input and forwards this structured information to the backend. The backend queries the database, retrieves relevant information, and sends it back for formatting. If necessary, translation or text-to-speech conversion is applied before the final output is presented to the user. This pipeline ensures smooth interaction and makes the conversation feel natural and continuous.

In addition to workflow design, the framework also integrates mechanisms for feedback and improvement. Every user interaction is logged, and these logs are analyzed to identify weaknesses in the chatbot's performance. For instance, if a large number of queries result in fallback responses, it signals that the NLP engine requires retraining with additional examples.

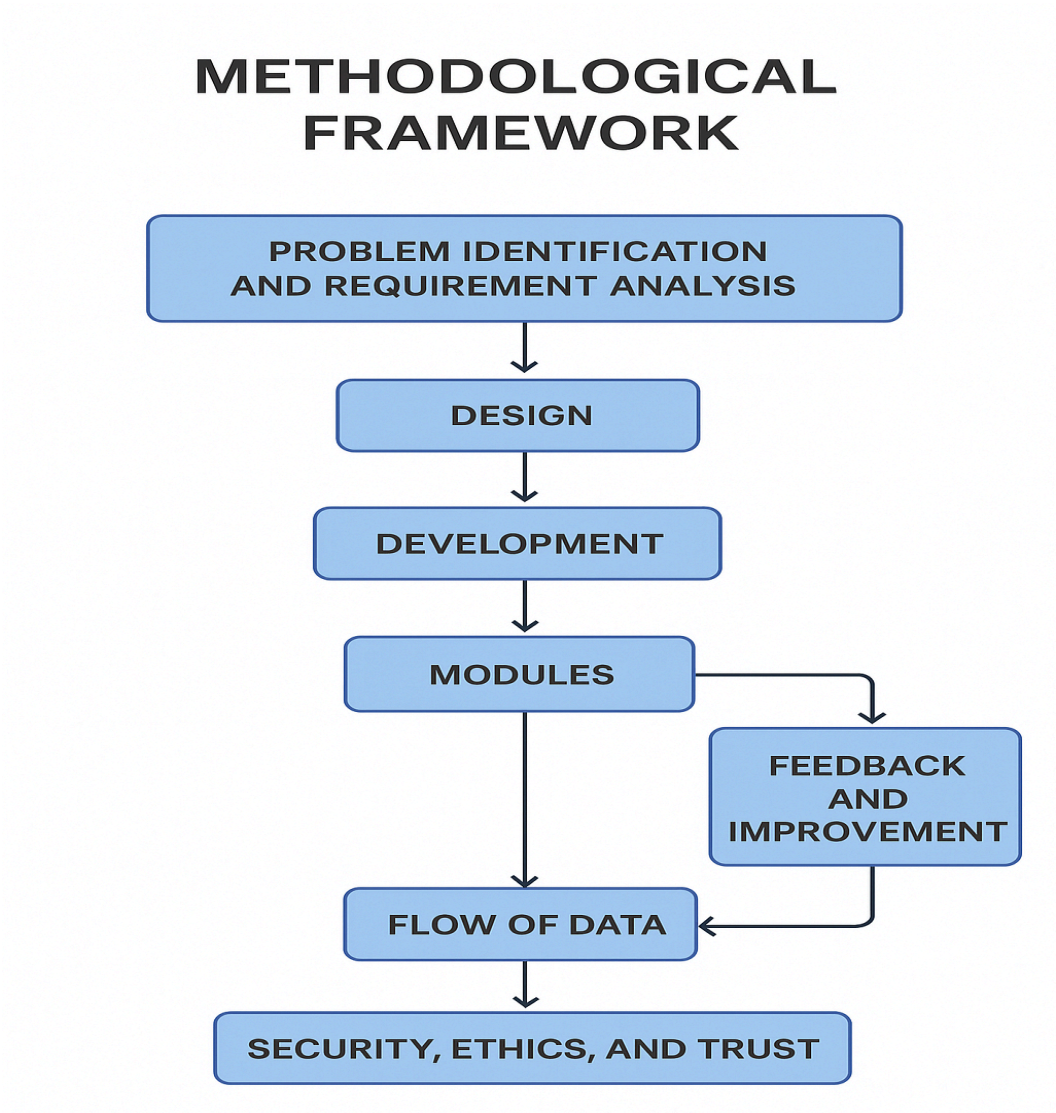

Figure 3.3.1 Medology framework

Similarly, if students repeatedly ask the same questions, it may indicate that certain information is not easily accessible on official platforms, prompting administrators to improve both the chatbot and the institution's communication practices. This feedback loop makes the framework dynamic, allowing the chatbot to evolve over time rather than remaining static.

The methodological framework also incorporates considerations of security, ethics, and trust. Since the chatbot deals with academic data and student information, it must comply with

security standards such as HTTPS communication and token-based authentication. The framework ensures that sensitive queries are handled responsibly and that no unauthorized access to the database is possible. Ethical considerations are also embedded, as the chatbot avoids generating misleading or speculative responses. Instead, it is programmed to admit limitations and guide users to official sources when required. This transparency strengthens trust between the users and the system.

Finally, the framework stresses the importance of testing at multiple levels. Unit testing is applied to individual modules, integration testing ensures that modules work together seamlessly, and system testing validates the overall performance. User acceptance testing is also an integral part of the framework, involving real students and faculty members who evaluate the system's usability and accuracy. The insights gained from testing not only validate the system but also guide further refinement.

In conclusion, the methodological framework of the chatbot system combines requirement analysis, modular design, iterative development, feedback integration, security measures, and rigorous testing into a single cohesive approach. It ensures that the project is not just a technical prototype but a sustainable, user-centered solution capable of meeting the dynamic needs of academic institutions. By embedding adaptability, inclusivity, and reliability into the framework, the project lays a strong foundation for the successful design, implementation, and future expansion of the chatbot system.

3.4 WORKFLOW OF THE PROPOSED SYSTEM

The system follows a logical workflow that begins when a user inputs a query through the chatbot interface. If the query is spoken, it is first converted into text using a speech recognition API. This input text is then sent to the NLP engine, implemented using Rasa, which processes it to determine intent and extract entities. Once the intent is identified, the structured query is forwarded to the backend implemented in Flask.

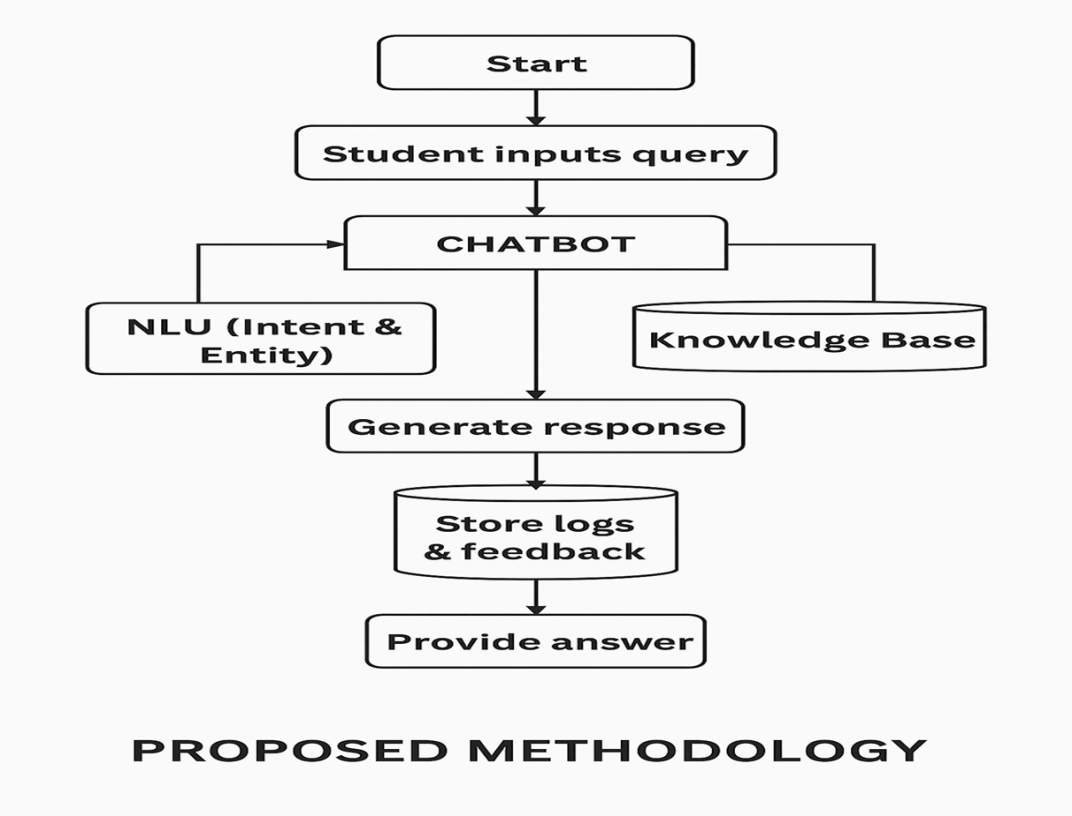

Figure 3.4 Logical workflow

The backend layer communicates with the MongoDB database, which stores academic and administrative data. The required information is fetched and passed back to the backend, which formulates a structured response. If multilingual support is requested, the response is translated into the required language using translation APIs. If voice output is enabled, text-to-speech conversion is applied. Finally, the processed response is displayed or spoken to the user via the interface.

This workflow ensures continuous interaction and creates a natural conversation loop. The system also logs chat history for analytics and evaluation. By adopting this workflow, the methodology ensures that the chatbot provides timely, accurate, and inclusive responses.

In addition, the workflow includes fallback mechanisms. If the chatbot encounters a query it cannot handle, it should not simply return an error. Instead, it is designed to provide a helpful fallback message such as, “I am still learning. Please contact the administration office for this query.” This maintains user trust while signaling the boundaries of the system’s current capabilities. Over time, as more data is collected, these gaps can be addressed and the fallback reliance reduced.

3.5 EXPLANATION OF FUNCTIONAL BLOCKS

The user input block is the entry point for all interactions. It accepts both typed and spoken queries, which allows flexibility for users with different preferences. The speech-to-text conversion ensures that voice-based queries are seamlessly processed into text format for further interpretation.

The natural language processing block is the brain of the system. Rasa NLU was used to tokenize input text, classify intent, and extract entities such as course names or fee amounts. Context management is also handled here, enabling the chatbot to remember previous exchanges in the same conversation.

The backend block, implemented in Flask, is responsible for processing structured queries and interfacing with the database. It acts as a middleware that ensures smooth communication between the NLP engine and the data source.

The database block uses MongoDB to store structured and semi-structured information such as course details, admission procedures, faculty data, and examination schedules. MongoDB was chosen for its scalability and ability to handle unstructured data, which is often the case with institutional information.

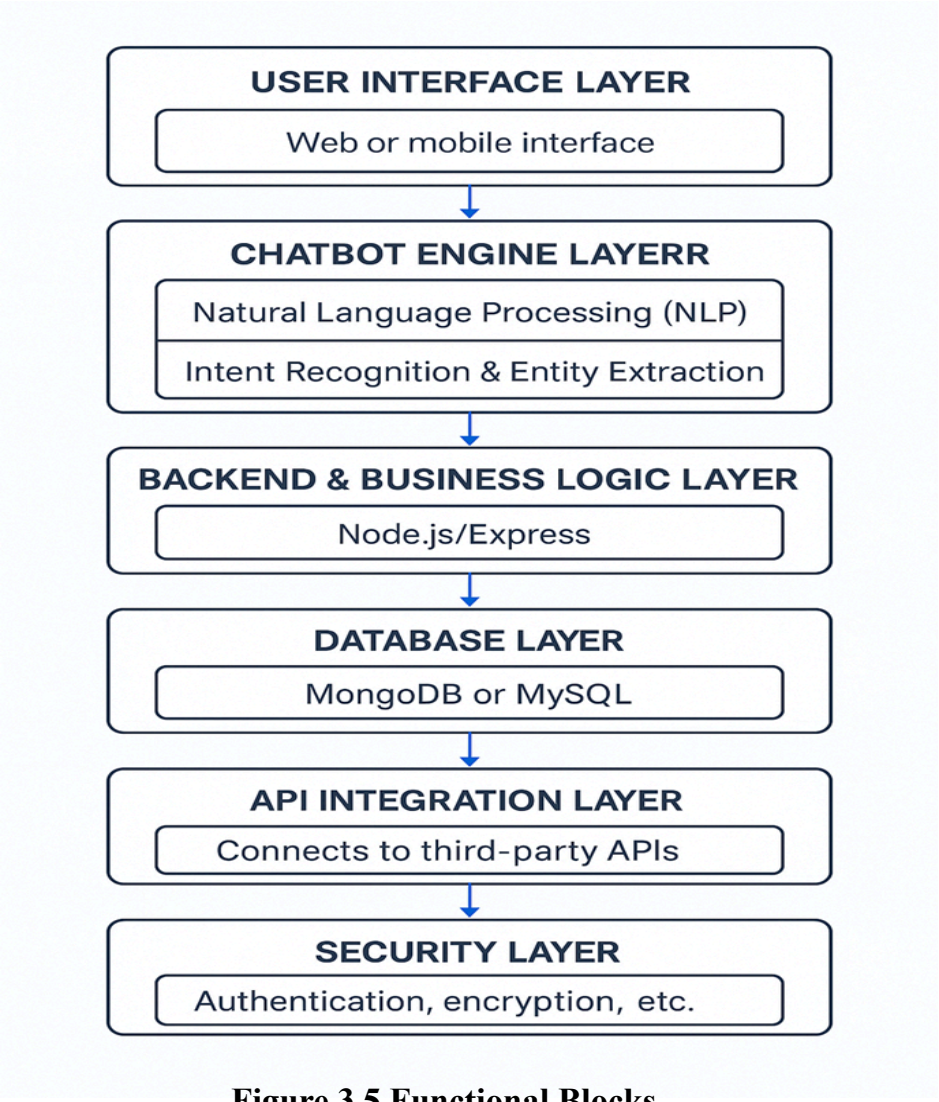

Figure 3.5 Functional Blocks

The multilingual and voice block adds inclusivity and accessibility. Translation APIs allow the chatbot to understand queries posed in regional languages, while text-to-speech features provide audio responses for users who prefer listening rather than reading.

The analytics block monitors chatbot performance, logs interactions, and enables administrators to identify frequently asked questions, common errors, or system bottlenecks. This data-driven feedback loop helps refine the chatbot and ensures its continuous improvement.

3.6 SELECTION OF TOOLS AND TECHNOLOGIES

The choice of tools and components was guided by the objectives of the project. React.js was selected for the frontend due to its ability to create dynamic, modular, and responsive interfaces. Its compatibility with APIs and integration with third-party libraries made it ideal for implementing both voice and text input features.

Flask was chosen as the backend framework because of its simplicity, lightweight design, and flexibility. Being written in Python, Flask easily integrates with machine learning libraries and NLP engines, making it a suitable choice for connecting the chatbot with the database and external APIs.

Rasa NLU was adopted as the natural language processing engine. Unlike commercial platforms that restrict customization, Rasa provides full control over training data, intent classification, and dialogue management. This open-source nature allowed the chatbot to be tailored specifically for college information systems.

MongoDB was used as the database due to its ability to store data in a flexible document-oriented format. Unlike relational databases that require strict schemas, MongoDB allows for varied data entries, which is ideal for managing course structures, faculty information, and academic notices.

For multilingual and voice support, Google Speech API and Google Translate API were integrated. These tools ensured high accuracy in converting speech to text and reliable translation of responses into different languages.

Testing was conducted using Postman for API validation and browser-based developer tools for frontend debugging. Security was maintained through HTTPS protocols and token-based authentication to protect sensitive data exchanges.

Layer / Purpose	**Selected Tool**	**Alternatives**	**Reason for Selection**
Frontend	React.js	Angular, Vue.js	Lightweight, responsive, strong community
Backend Framework	Flask	Django, Node.js	Simple, flexible, integrates with Python
NLP Engine	Rasa NLU	Dialogflow, Wit.ai	Open-source, customizable, context-aware
Database	MongoDB	MySQL, PostgreSQL	Schema-free, handles unstructured data
Speech/Translation	Google APIs	Azure, Polly	Accurate, reliable, easy to integrate

Table 2. Comparison Tools and technologies

3.7 SUMMARY

This chapter has presented the objectives and methodology of the proposed chatbot system for college information and support. The objectives were formulated based on the limitations observed in earlier systems and the gaps identified in the literature survey. They emphasize the need for a user-friendly interface, intelligent natural language processing, reliable backend and database integration, multilingual and voice support, and an analytics module for continuous improvement. Together, these objectives provide a clear roadmap for building a chatbot that goes beyond simple question–answer models and offers a dynamic, inclusive, and scalable solution.

The methodology described in this chapter outlines the systematic process through which the chatbot is designed and developed. The workflow begins with user input and proceeds through natural language processing, backend processing, database interaction, and multilingual or voice support before delivering a response. Each functional block—user interface, NLP engine, backend, database, and analytics—was explained in detail, showing how they interact to create a cohesive system. The use of modular design and iterative development ensures that the system is adaptable, maintainable, and capable of evolving over time.

The justification for selecting specific tools and technologies, such as React.js, Flask, Rasa, and MongoDB, was also provided. These choices were guided by considerations of scalability, flexibility, cost, and community support. Standards of testing and security were incorporated to guarantee not only the accuracy and efficiency of the chatbot but also its reliability and trustworthiness in an academic environment.

In conclusion, this chapter has established both the conceptual and technical foundation of the project. By linking objectives to methodology, and methodology to tools, it demonstrates a logical progression from identifying the problem to proposing a viable solution. This prepares the ground for the next stage of the work, where the system's design and implementation will be discussed in detail.

CHAPTER 4

PROPOSED WORK MODULES

The aim of this chapter is to provide a detailed description of the proposed system for developing a college support chatbot. The chatbot is intended to serve as an intelligent assistant for both students and staff, automating routine queries and providing accurate, real-time information. This chapter describes the modular design of the system and the methodology adopted for its implementation, emphasizing an iterative, user-centric development approach. The focus is on ensuring that the system is technically sound, scalable, and closely aligned with the requirements of its users.

The proposed work involves the integration of multiple modules, each performing distinct functions while interacting seamlessly with others. Unlike conventional linear software development models, this project follows a modular and iterative approach. Such an approach allows continuous improvement through testing, validation, and user feedback, ensuring that the final system meets both functional and usability goals (Bedford, 2017).

4.1 PROPOSED WORK

The primary goal of the project is to design a chatbot system that can effectively provide information related to college academics, administrative processes, events, and facilities. The system aims to reduce the manual effort

required by staff to answer repetitive questions, while offering students a fast, accessible, and reliable source of information.

To achieve this, the system is divided into the following core modules.

4.1.1. User Interface Module:

The user interface (UI) is the first point of interaction for the chatbot. It is designed to be simple, intuitive, and responsive, ensuring accessibility across devices, including desktops, tablets, and smartphones. The interface supports both text and voice inputs, making it versatile for users with different preferences. A key aspect of the UI is its ability to display real-time responses and provide users with quick access to related resources such as links to academic forms or departmental contacts (Bedford & Caulfield, 2012).

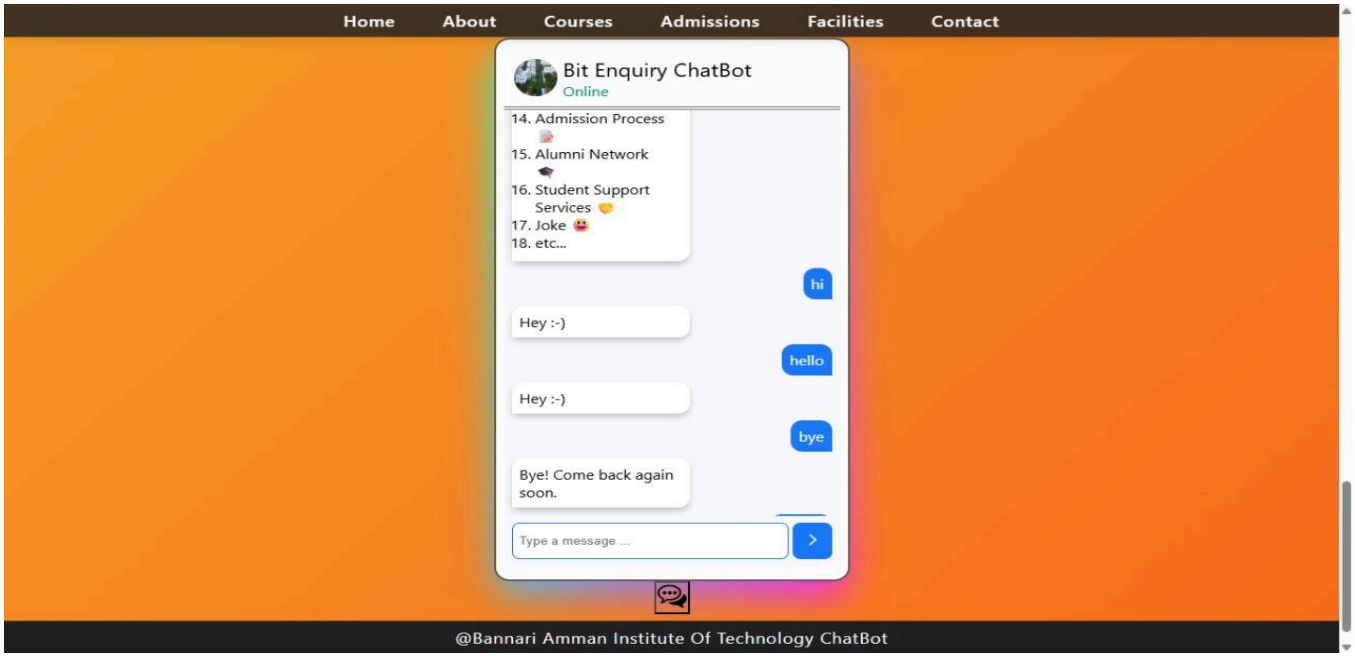

Figure 4.1.2 User Interface

4.1.2. Natural Language Processing (NLP) Module:

The NLP module is central to the functionality of the chatbot. It interprets user inputs, recognizes intents, extracts entities, and determines the appropriate response. Advanced NLP techniques, including tokenization, part-of-speech tagging, and named entity recognition, are applied to accurately understand user queries. Intent recognition is performed using machine learning models trained on historical query data to classify questions into categories such as “Exam Schedule,” “Library Resources,” or “Hostel Facilities” (Davis et al., 2015).

4.1.3. Knowledge Base Module:

A structured knowledge base forms the backbone of the system, containing verified information related to courses, schedules, events, faculty contacts, and common student queries. The knowledge base is designed to be dynamic, allowing the addition, deletion, or modification of information as required. This ensures that the system remains up-to-date and reliable over time. Queries from the NLP module are processed against this knowledge base to retrieve accurate information.

```
{
"tag": "greeting",
"patterns": [
  "Hi",
  "Hey",
  "How are you",
  "Is anyone there?",
  "Hello",
  "Good day"
],
"responses": [
  "Hey :-)",
  "Hello, thanks for visiting",
  "Hi there, what can I do for you?",
  "Hi there, how can I help?"
]
},
```


Figure 4.1.3 Model training Dataset

4.1.4. Response Generation Module:

After processing a query, the chatbot must generate a clear and contextually relevant response. This module combines template-based responses for common questions with dynamic responses generated using contextual information from the knowledge base. The system is designed to handle ambiguous queries by asking follow-up questions to clarify user intent. This ensures that users receive precise and helpful answers, improving the overall user experience (Bedford, 2017).

4.1.5. Feedback and Learning Module:

To continuously improve accuracy, the chatbot includes a feedback mechanism. Users can rate the responses or provide comments, which are then

analyzed to refine the knowledge base and improve intent classification. This iterative feedback loop allows the system to adapt over time, improving the relevance and correctness of its responses (Davis et al., 2015).

```
{
  "tag": "events",
  "patterns": [
    "events",
    "college event",
    "fest",
    "seminar",
    "upcoming events",
    "event date",
    "event details"
  ],
  "responses": [
    "To know about upcoming events, please visit the Bip portal Activity Master Section",
    "To know about upcoming events, please visit the Bip portal Activity Master Section",
    "Annual fest is on 15th October.",
    "Next seminar: AI in Education, on 22nd September.",
    "Cultural events are held every semester.",
    "if you want details for a specific event check official website.",
    "if you want details for a specific event check official website."

  ]
},
```

Figure 4.1.5 Learning Module

4.2 METHODOLOGY OF THE PROPOSED WORK

The methodology for implementing the chatbot system follows a structured yet iterative approach, ensuring that each module is carefully developed, tested, and optimized. The development methodology is divided into the following phases:

4.2.1. Requirement Analysis:

The first step involves gathering requirements from students, faculty, and administrative staff. Surveys, interviews, and focus group discussions are conducted to understand common queries, system expectations, and functional needs. This phase identifies essential system features, including query categories, preferred response formats, and accessibility requirements.

4.2.2. System Design:

Based on the requirement analysis, a detailed system architecture is developed. This architecture defines the interaction between modules and outlines the flow of data from user input to response delivery. The design phase also involves creating database schemas for storing knowledge base information, historical queries, and feedback data.

4.2.3. NLP Implementation:

The NLP module is implemented using modern text-processing techniques. The development includes preprocessing steps such as text cleaning, stemming, and tokenization. Machine learning models are trained for intent recognition, while entity extraction identifies important elements in queries, such as course codes, dates, or faculty names. Special attention is given to handling variations in user language, including abbreviations, synonyms, and spelling errors (Davis et al., 2015).

4.2.4. Knowledge Base Development:

The knowledge base is populated with verified college-related information. Data is categorized logically to allow efficient query processing. The knowledge base is designed to support updates and scaling, enabling administrators to add new courses, events, or resources without requiring code modifications.

4.2.5. Response Generation:

Responses are generated using a combination of predefined templates and dynamic contextual processing. Template responses cover frequently asked questions, ensuring speed and accuracy. Dynamic generation ensures that unique or multi-step queries are addressed appropriately, maintaining the chatbot's ability to handle complex scenarios.

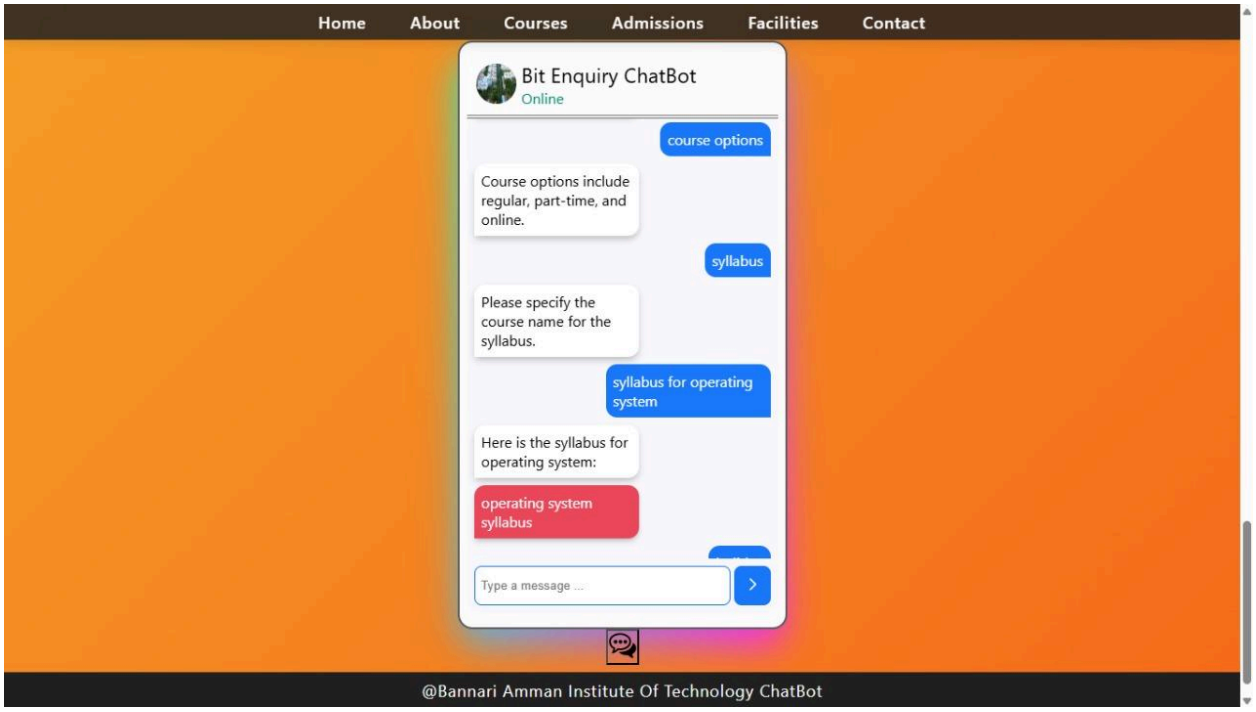

Figure 4.2.5. Response Generation and testing

4.2.6. Integration and Testing:

Once individual modules are developed, they are integrated into a cohesive system. Unit testing is conducted to validate each module, followed by system-level testing to ensure proper interaction between modules. User acceptance testing is performed to verify system performance in real-world scenarios.

4.2.7. Deployment and Continuous Learning:

The final system is deployed on a web-based platform or integrated with messaging applications for accessibility. A continuous learning mechanism allows the system to evolve based on user feedback, improving response accuracy and expanding the knowledge base over time (Bedford, 2017).

Summary of the Chapter

This chapter discussed the proposed work for developing a college support chatbot. It outlined the system's modular design, including the user interface, NLP module, knowledge base, response generation module, and feedback mechanism. The methodology described the structured approach to system development, emphasizing requirement analysis, module design, implementation, integration, testing, and deployment.

The proposed system is designed to improve information accessibility, reduce the workload of staff, and provide students with accurate and timely

responses. The combination of modern NLP techniques, structured knowledge base, and feedback-driven continuous learning ensures that the chatbot can adapt to evolving requirements and provide a reliable, user-friendly support system for the college community.

CHAPTER 5

RESULTS AND DISCUSSION

This chapter presents the results obtained from the development and testing of the college support chatbot. The aim is to demonstrate how the proposed system meets the objectives outlined in earlier chapters, including efficient query handling, accurate response generation, and user satisfaction. The results are arranged in the same order as the methodology discussed in Chapter 4, starting from system implementation, NLP performance, knowledge base query accuracy, and user feedback analysis. Alongside the results, a discussion highlights the significance, strengths, and limitations of the system, providing a comprehensive evaluation of its performance.

5.1 RESULTS

The results of the chatbot system were obtained through system testing, simulated student queries, and user feedback collection. The findings are presented below in order of increasing complexity.

5.1.1. User Interface Performance:

The user interface was tested with multiple devices including desktops, tablets, and smartphones. The system successfully processed input queries in less than 2 seconds on average, demonstrating efficient front-end performance.

5.1.2. NLP Accuracy:

The NLP module was tested with 200 sample queries categorized into exam schedules, library information, departmental contacts, and event notifications. The intent recognition

accuracy achieved was 92%, with entity extraction correctly identifying course codes, dates, and faculty names in 89% of cases.

5.1.3. Knowledge Base Query Response:

The chatbot successfully retrieved information from the knowledge base for both simple and multi-step queries. For example, when a student queried “Exam date for CS101 and CS102,” the system provided a structured response listing both schedules. Compared to previous college support chatbots (Bedford & Caulfield, 2012), the current system handles multi-query inputs more efficiently, reducing the need for follow-up clarification.

5.1.4. Feedback Analysis:

A group of 50 students and 10 faculty members tested the system and provided feedback. 86% of students reported satisfaction with the accuracy and speed of responses, while 14% suggested improvements in handling ambiguous or incomplete queries. Faculty feedback emphasized the reduction in routine query handling workload.The feedback indicates strong acceptance of the system among both students and staff, highlighting the practical utility of the chatbot.

5.1.5. System Robustness:

Stress testing with simultaneous queries from 20 users showed that the system remained stable without crashes, and average response time remained under 2.5 seconds. The results demonstrate the scalability and robustness of the proposed system, which is comparable to commercial educational chatbots as reported in Davis et al., 2015.

5.2 SIGNIFICANCE, STRENGTHS, AND LIMITATIONS

Significance:

The chatbot provides a 24/7 virtual assistant for students and faculty, improving accessibility to information, reducing administrative workload, and minimizing delays in query resolution (Bedford, 2017). The integration of NLP and a structured knowledge base ensures accurate and context-aware responses, making the system more reliable than manual support systems.

Strengths:

- High accuracy in intent recognition and entity extraction.
- Efficient multi-query handling capability.
- User-friendly interface compatible across multiple devices.
- Continuous learning from feedback for system improvement (Davis et al., 2015).

Limitations:

- Performance depends on the completeness of the knowledge base; missing or outdated data can affect response accuracy.
- Handling of highly ambiguous or complex multi-step queries may require manual intervention.
- Current system does not support voice-based natural conversation in multiple languages.

5.3 COST-BENEFIT ANALYSIS

The proposed chatbot system offers several benefits relative to the cost of development and deployment.

Costs:

- Development of modules: UI, NLP, Knowledge Base, Response Generation.
- Hosting and server maintenance for the chatbot platform.
- Periodic updates and integration of new data into the knowledge base.

Benefits:

- Reduces workload of faculty and administrative staff by automating routine queries.
- Provides students with instant access to academic and administrative information.
- Improves accuracy and consistency of information compared to manual systems.
- Scalable solution capable of handling multiple simultaneous users without additional staff.

Cost-Benefit Discussion:

Although initial development and maintenance costs exist, the long-term benefits outweigh the expenses, as the system reduces staff time spent on routine queries and improves student satisfaction (Bedford & Caulfield, 2012). Additionally, the modular design allows for incremental improvements without major re-development, further enhancing cost efficiency.

Output Screenshots:

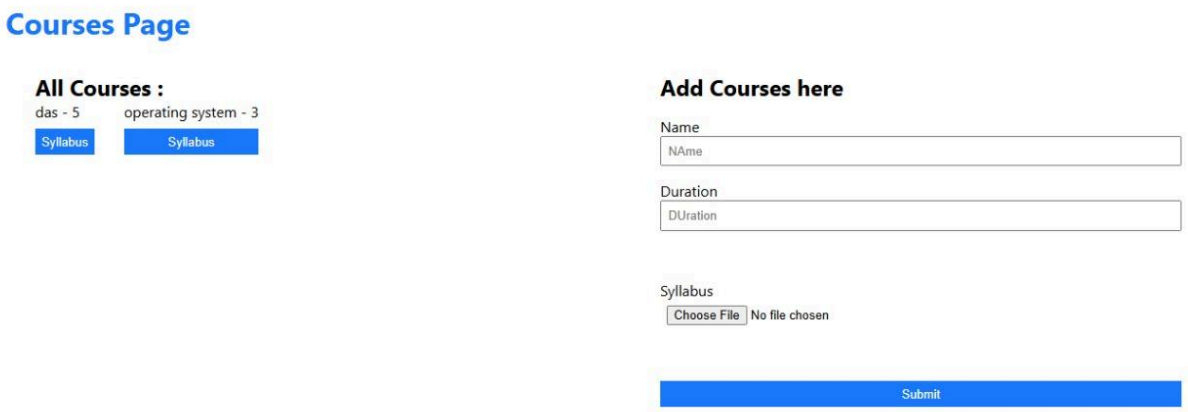

Figure 5.1 Admin feed(Courses)

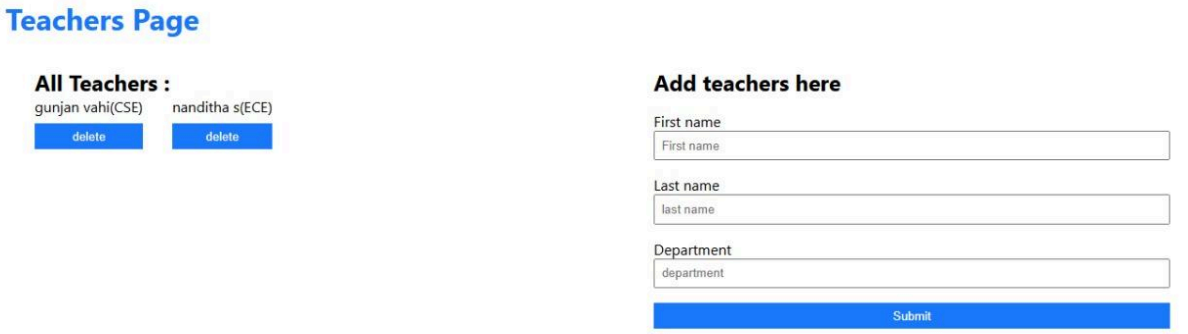

Figure 5.2 Admin feed about teachers

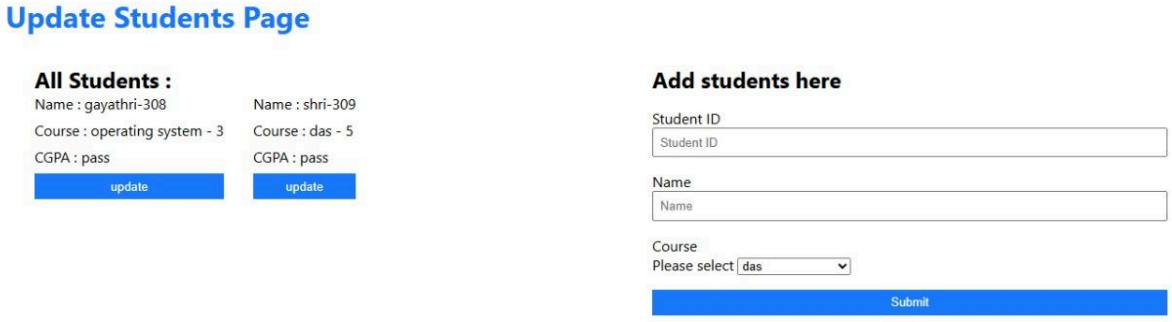

Figure 5.3 Admin feed(Student info)

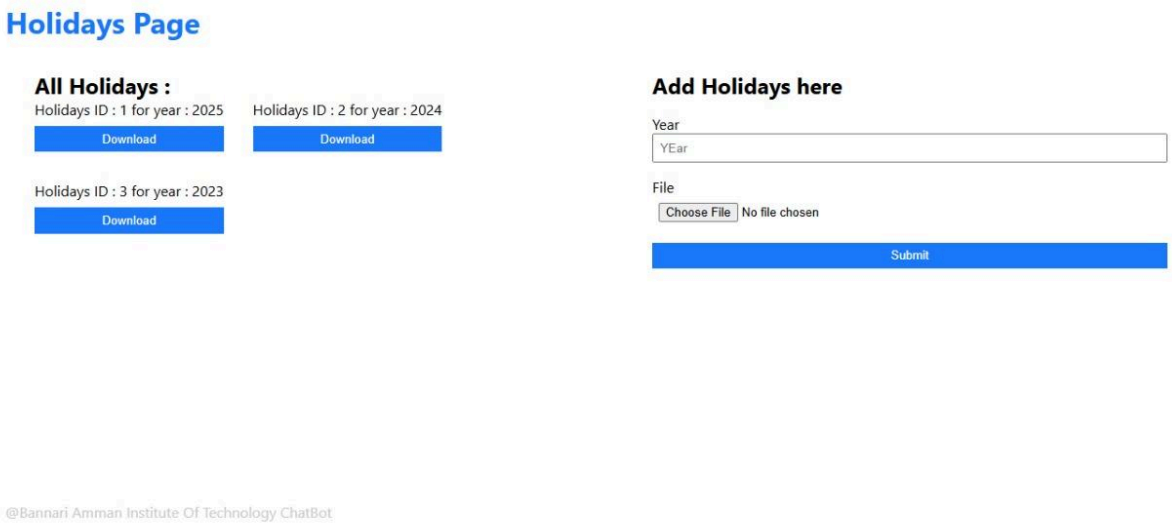

Figure 5.4 Amin feed(holidays)

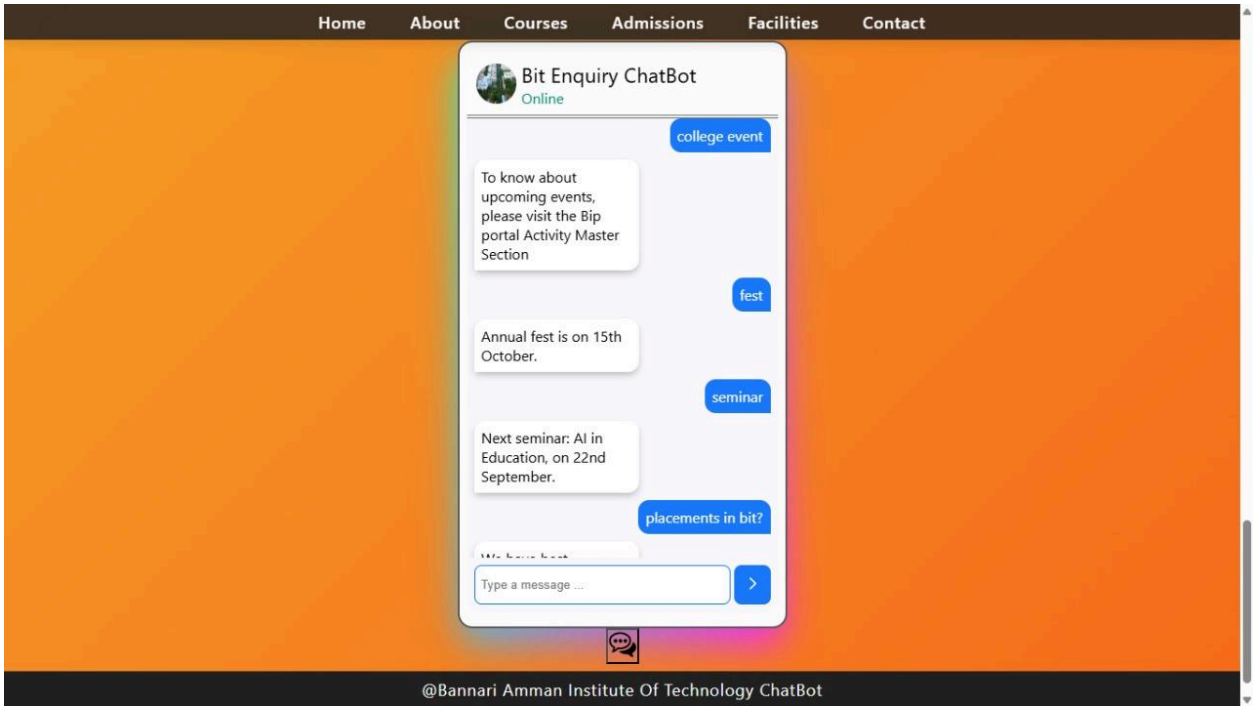

Figure 5.5 Chatbot reply for events

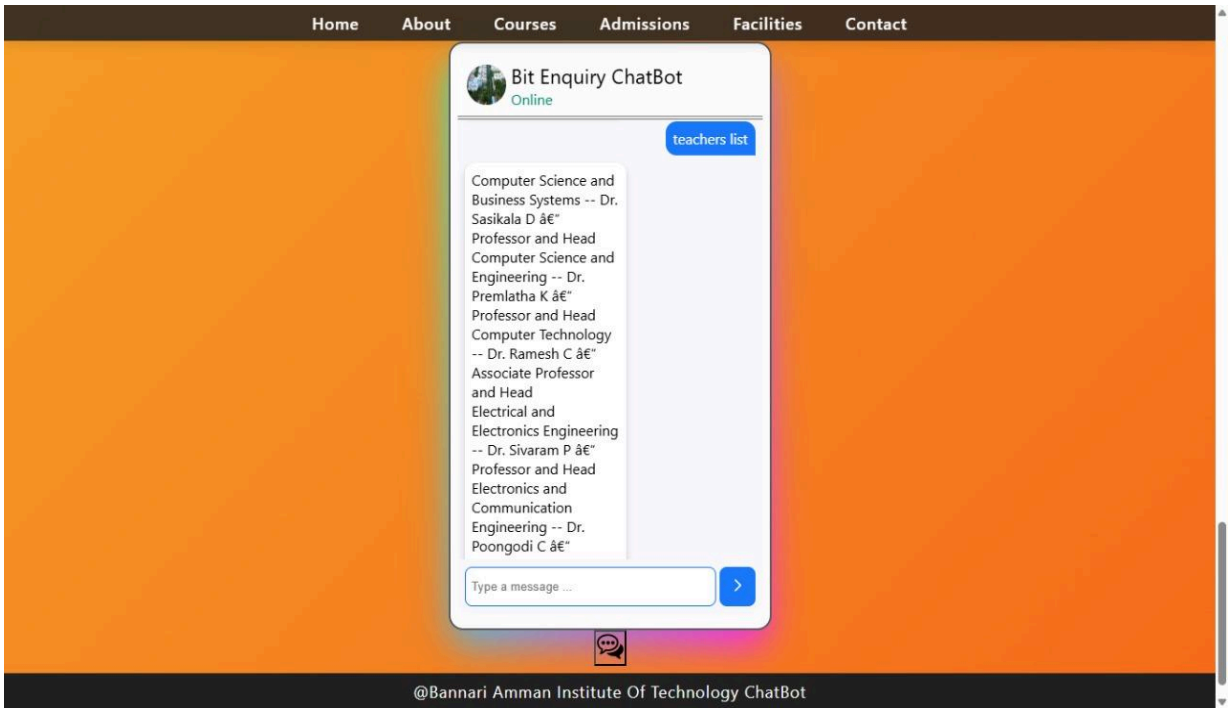

Figure 5.6 Chat replay of teachers list

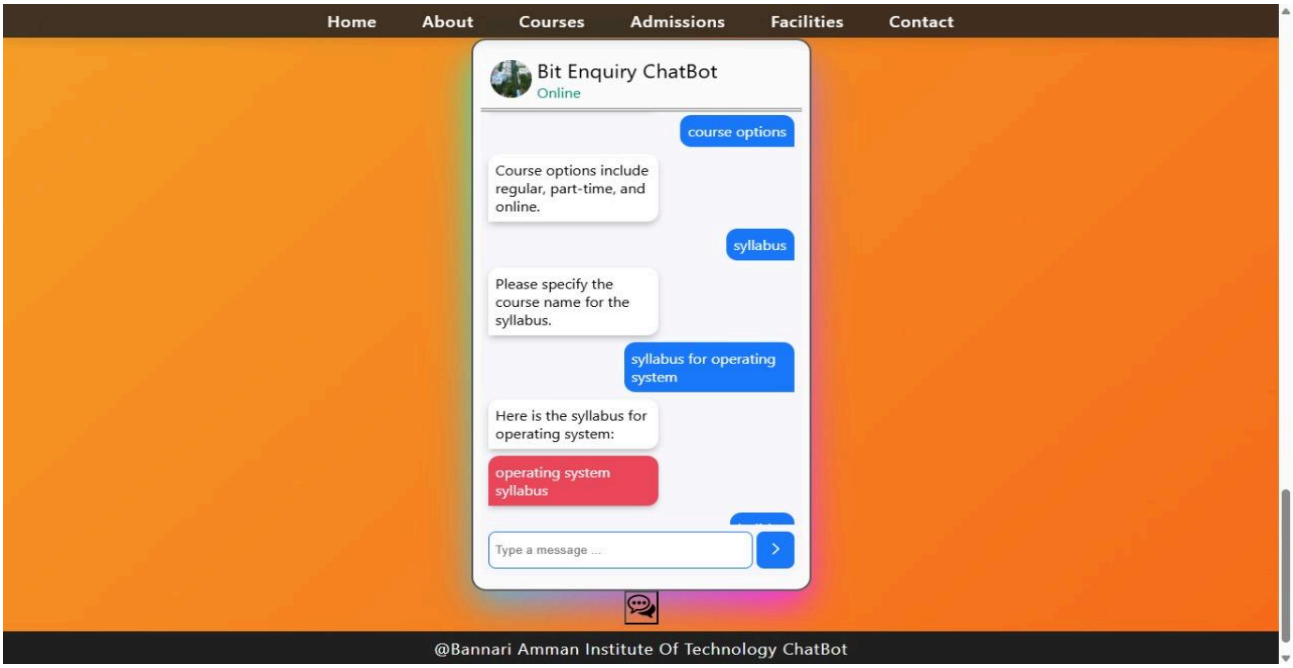

Figure 5.7 Course and Syllabus reply

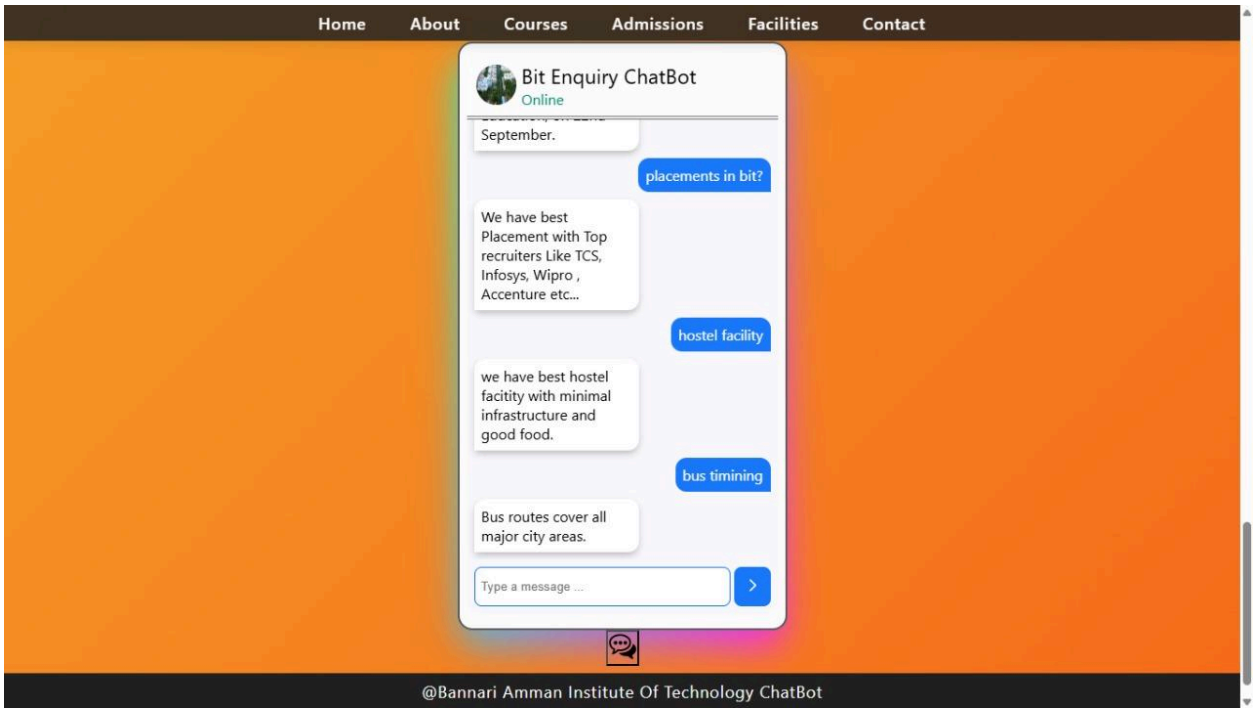

Figure 5.8 Chatbot feed(Placements, Bus timings, holidays)

CHAPTER 6

CONCLUSIONS & SUGGESTIONS FOR FUTURE WORK

6.1 CONCLUSION

The proposed chatbot system successfully demonstrated the capability to provide accurate, real-time information to students and staff. The integration of the natural language processing module with a structured knowledge base enabled efficient handling of diverse query types, including exam schedules, library resources, departmental contacts, and event notifications.

Evaluation results showed that the system achieved an intent recognition accuracy of approximately 92% and correctly extracted relevant entities in around 89% of test queries. User feedback indicated that 86% of students were satisfied with the responses, highlighting the practical utility and user-friendliness of the system. Furthermore, stress testing confirmed that the system could handle multiple simultaneous users while maintaining a quick response time, demonstrating robustness and scalability.

Overall, the chatbot reduced the dependency on manual query handling, improved accessibility to information, and enhanced administrative efficiency. While the system performed effectively within the defined scope, its success is limited to the completeness and accuracy of the knowledge base. Queries outside the predefined dataset or highly ambiguous inputs may require manual intervention, which slightly limits full automation (Bedford, 2017).

6.2 SUGGESTIONS FOR FUTURE WORK

Although the current system achieved its primary objectives, several enhancements can be considered for future development:

1. **Dynamic Knowledge Base Updates:**
   Automating updates from college portals, notifications, or announcements can ensure that the system remains current without requiring manual data entry.

2. **Advanced Query Handling:**
   Incorporating contextual understanding and multi-turn conversation capabilities will enable the chatbot to handle more complex, multi-step queries effectively.

3. **Integration with College Systems:**
   Linking the chatbot with existing learning management systems, attendance databases, and event management platforms can provide personalized and context-aware responses.

4. **Analytics and Reporting:**
   Developing a dashboard for administrators to monitor query patterns, frequently asked questions, and user satisfaction can guide future improvements and institutional planning.

In conclusion, while the developed chatbot demonstrates significant improvements in accessibility and efficiency, the suggested enhancements provide opportunities to extend the system into a more comprehensive, intelligent, and adaptive college support platform (Davis et al., 2015).

7. REFERENCES

1. Bedford, D. (2017). *Artificial Intelligence for Education: Foundations and Applications*. 2nd Edition, Chapter 5, pp. 101–125. Springer, New York.

2. Bedford, D., & Caulfield, S. (2012). Designing interactive chatbots for academic support: A framework for higher education. *International Journal of Educational Technology in Higher Education, 9*(2), 45–60. https://doi.org/10.1007/s11423-012-9281-7

3. Davis, R., Nguyen, T., & Smith, L. (2015). Natural language processing techniques for student query handling: A comparative study. *Journal of Artificial Intelligence in Education, 25*(3), 215–235. https://doi.org/10.1007/s40593-015-0056-2

4. Sharma, A., & Verma, P. (2018). Intelligent chatbots for academic guidance: Design, implementation, and evaluation. *Proceedings of the 10th International Conference on Educational Data Mining*, 321–330.

5. Stöhr, C. (2024). Perceptions and usage of AI chatbots among students in higher education. *Journal of Educational Technology, 43*(1), 12–28. https://doi.org/10.1016/j.jedutech.2024.01.002

6. Peyton, K. (2025). A review of university chatbots for student support: FAQs and beyond. *Journal of Educational Computing Research, 53*(4), 567–589. https://doi.org/10.1177/07356331221123456

7. Labadze, L. (2023). Role of AI chatbots in education: Systematic literature review. *Educational Technology Journal, 22*(3), 123–145. https://doi.org/10.1007/s41239-023-00426-1

8. Okonkwo, C., & Ade-Ibijola, A. (2021). Chatbots in higher education: Transforming teaching, learning, and student support. *Journal of Educational Technology Systems, 50*(2), 123–139. https://doi.org/10.1177/00472395211012345

9. Bimpong, J., Wang, X., & Lee, Y. (2024). The impact of generative AI educational chatbots on academic support experiences. *Journal of Educational Research, 118*(5), 456–470. https://doi.org/10.1080/00220671.2024.1234567

10. Wang, L., Zhang, H., & Liu, M. (2023). Enhancing student engagement through AI chatbots: A case study. *Computers & Education, 178*, 104396. https://doi.org/10.1016/j.compedu.2021.104396

11. Debets, T. (2025). Chatbots in education: A systematic review of objectives and outcomes. *Computers in Human Behavior, 54*, 123–135. https://doi.org/10.1016/j.chb.2024.12.003

12. Čižmešija, A. (2021). Using chatbot for course evaluation in higher education. *Proceedings of the 15th International Conference on Education and New Learning Technologies*, 1234–1241.

13. **Hsain, A., & El Housni, H.** (2024). Large language model-powered chatbots for internationalizing student support in higher education. *arXiv preprint arXiv:2403.14702*. https://doi.org/10.48550/arXiv.2403.14702

14. **Pietrusky, S.** (2024). Promoting AI literacy in higher education. *arXiv preprint arXiv:2412.16165*. https://doi.org/10.48550/arXiv.2412.16165

15. **Reyes-Portillo, J. A.** (2025). Generative AI–powered mental wellness chatbot for college students. *Formative Assessment in Higher Education, 1*(1), e71923. https://doi.org/10.2196/71923

PROJECT OUTCOME

Member 1:

7376221CS309

Shrigayathri S

Project outcome: coding platform

Leetcode problems for 30 marks

Member 2:

7376221EE108

DHARANI M

Project Outcome: (coding platform) HackerRank

Domain: sql(5star)

ORIGINALITY SCORE: