# CHAPTER 1

## INTRODUCTION

In today's digital age, accessing reliable information quickly is essential, especially for students seeking guidance about their academic journey. A chatbot designed to provide accurate and instant information about colleges can significantly improve the user experience, saving time and effort while offering a modern solution to traditional inquiry methods. The College Information Chatbot is an AI-powered application tailored to provide comprehensive details about a college, including admission requirements, courses offered, tuition fees, placement statistics, and campus facilities.

The chatbot effectively interacts with users by combining natural language processing (NLP) and user-friendly graphical interfaces. It caters to both text and voice input, making it accessible to a diverse audience. With features like real-time question answering, voice-based interaction, and theme customization, the chatbot delivers an engaging and dynamic experience.

Developed using Python and the Tkinter library for the GUI, the chatbot integrates modules like speech_recognition, gTTS, and pygame for voice capabilities, alongside a structured database of predefined responses. These responses cover common student queries, such as admission criteria, available programs, and hostel information, ensuring a quick resolution to frequently asked questions.

The primary objective of this chatbot is to bridge the communication gap between prospective students and colleges by offering 24/7 assistance. It serves as an efficient and scalable alternative to manual inquiry systems, ensuring that users receive accurate, concise, and relevant information without human intervention. Furthermore, the inclusion of interactive features, such as FAQs and the ability to save chat history, enhances the chatbot’s usability and effectiveness.

This project aligns with the growing demand for AI-based solutions in the education sector, emphasizing user-centric design and automation. By simulating a conversational assistant, the chatbot not only provides information but also creates a personalized experience, empowering students to make informed decisions about their future.

## 1.1 Objectives

- Develop an AI chatbot capable of providing accurate and efficient college-related information.
- Ensure the chatbot can respond to user queries about admissions, courses, faculty, campus facilities, and events.
- Implement natural language processing (NLP) for understanding and interacting in a human-like manner.
- Simplify access to college information for students, faculty, and prospective applicants.

## 1.2 Problem Statement

- Accessing relevant college information often involves navigating complex websites, resulting in a time-consuming and frustrating experience for users.
- Many users lack the technical knowledge to find specific details efficiently.
- Existing chatbots or systems may not cater to the dynamic nature of queries or provide personalized responses.
- The project aims to address these gaps by creating a user-friendly AI chatbot capable of handling diverse queries and improving user experience.

# CHAPTER 2

## LITERATURE SURVEY

The development of AI-powered chatbots for educational purposes has gained significant attention over the past decade. The ability of chatbots to provide quick and accurate responses makes them an ideal tool for addressing the growing demand for efficient information retrieval systems in academic institutions. This literature survey reviews various studies and technological advancements that form the foundation of the College Information Chatbot project.

Chatbot Development and Usability: Research by Shawar and Atwell (2007) highlights the importance of natural language processing (NLP) in creating intelligent systems capable of understanding and responding to user queries. Chatbots like ELIZA and ALICE laid the groundwork for modern conversational agents by using pattern-matching algorithms to simulate human interaction.

AI in Education: Studies such as those by Wang et al. (2020) demonstrate the potential of AI in enhancing student engagement and streamlining administrative processes. Chatbots integrated with educational platforms improve efficiency by automating tasks such as providing course details, scheduling, and answering FAQs.

User-Centric Design: Research by Jain et al. (2018) emphasizes the importance of designing user-friendly interfaces for educational chatbots. Features such as voice input, customizable themes, and multilingual support make these systems accessible to a diverse user base.

Speech Recognition and Synthesis: The use of speech recognition technologies like Google Speech API and text-to-speech systems (e.g., gTTS) is explored in studies by Singh and Arora (2019). These technologies enable seamless voice-based interactions, broadening the scope of chatbot applications.AI-Powered Admission Systems: A case study by Kumar et al. (2021) discusses the implementation of AI chatbots in college admissions, showcasing their effectiveness in providing real-time information on admission requirements, fee structures, and course offerings.

Integration with Python Libraries: The adoption of Python for chatbot development is widely documented due to its extensive library support and ease of integration. Libraries such as Tkinter for GUI, Pygame for multimedia features, and SpeechRecognition for voice capabilities have been proven effective in building robust chatbot systems.

Data Security and Privacy: Studies by Gupta and Sharma (2020) highlight the challenges of maintaining data security and privacy in AI applications. They recommend implementing secure data handling practices to ensure user trust and compliance with regulations.

Chatbots in College Environments: Research conducted by Fernandes et al. (2019) demonstrates the role of chatbots in improving operational efficiency on campuses. These systems address student queries about facilities, placements, and financial aid, reducing dependency on human staff.

These studies collectively underline the transformative role of AI-driven chatbots in educational settings. By leveraging advancements in NLP, voice processing, and user interface design, the College Information Chatbot builds upon these research findings to deliver a practical and innovative solution tailored to the needs of students and academic institutions.

# CHAPTER 3

# SYSTEM DESIGN

## 3.1 ARCHITECTURE

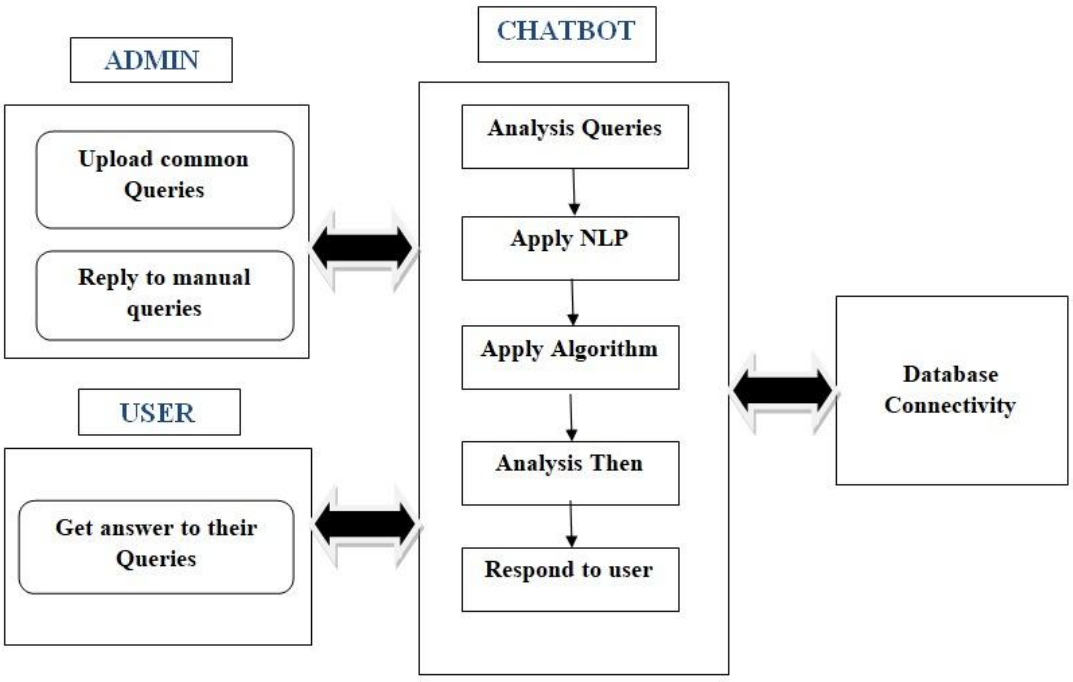


Fig 3.1: Architecture of the Chatbot model

## 3.2 DATAFLOW DIAGRAM

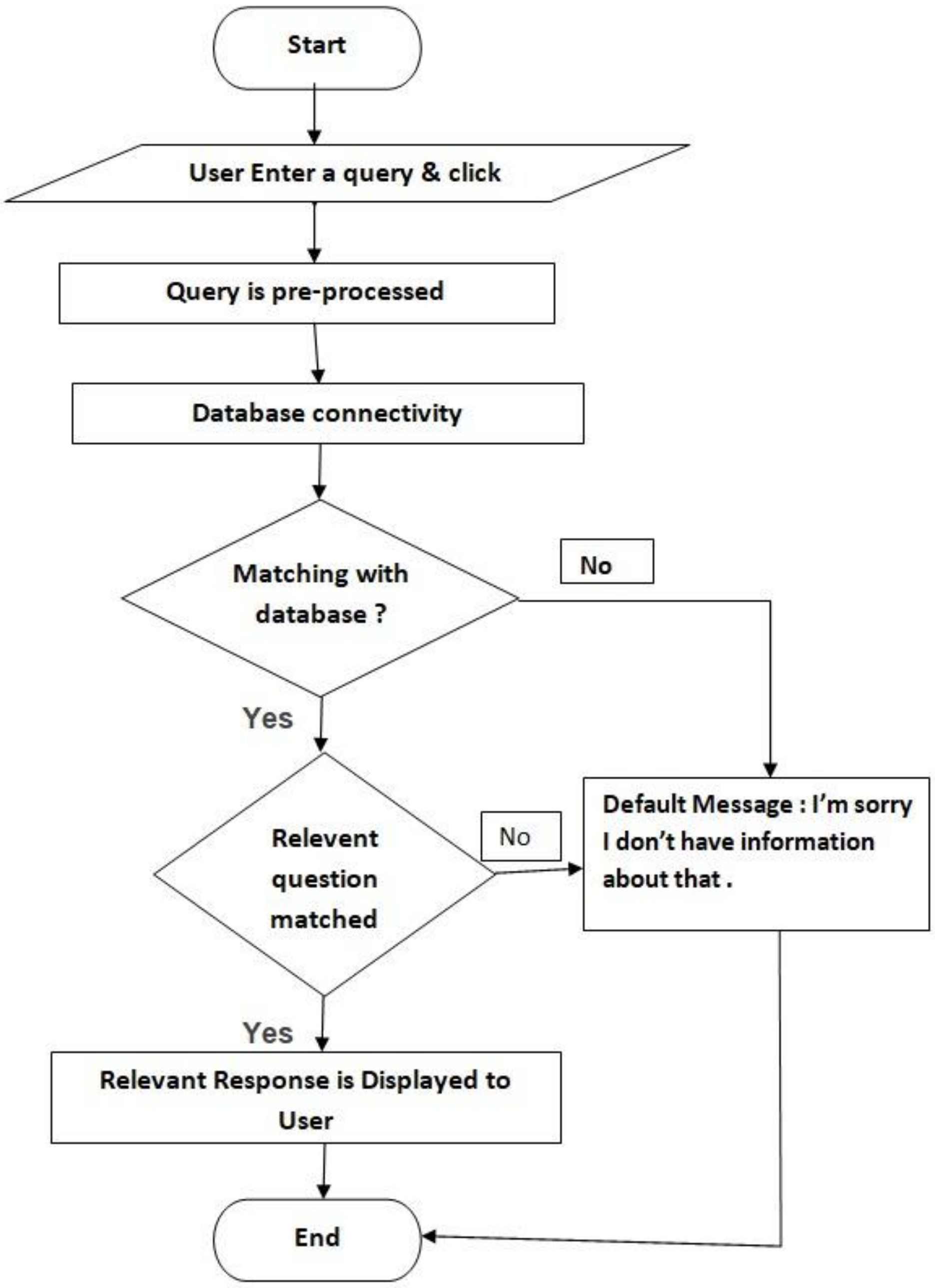

Start
User Enter a query & click
Query is pre-processed
Database connectivity
Matching with database ?
No
Yes
Relevent question matched
No
Default Message : I'm sorry I don't have information about that .
Yes
Relevant Response is Displayed to User
End

Fig 3.2: Dataflow diagram of the proposed system

## 3.3 USE CASE DIAGRAM

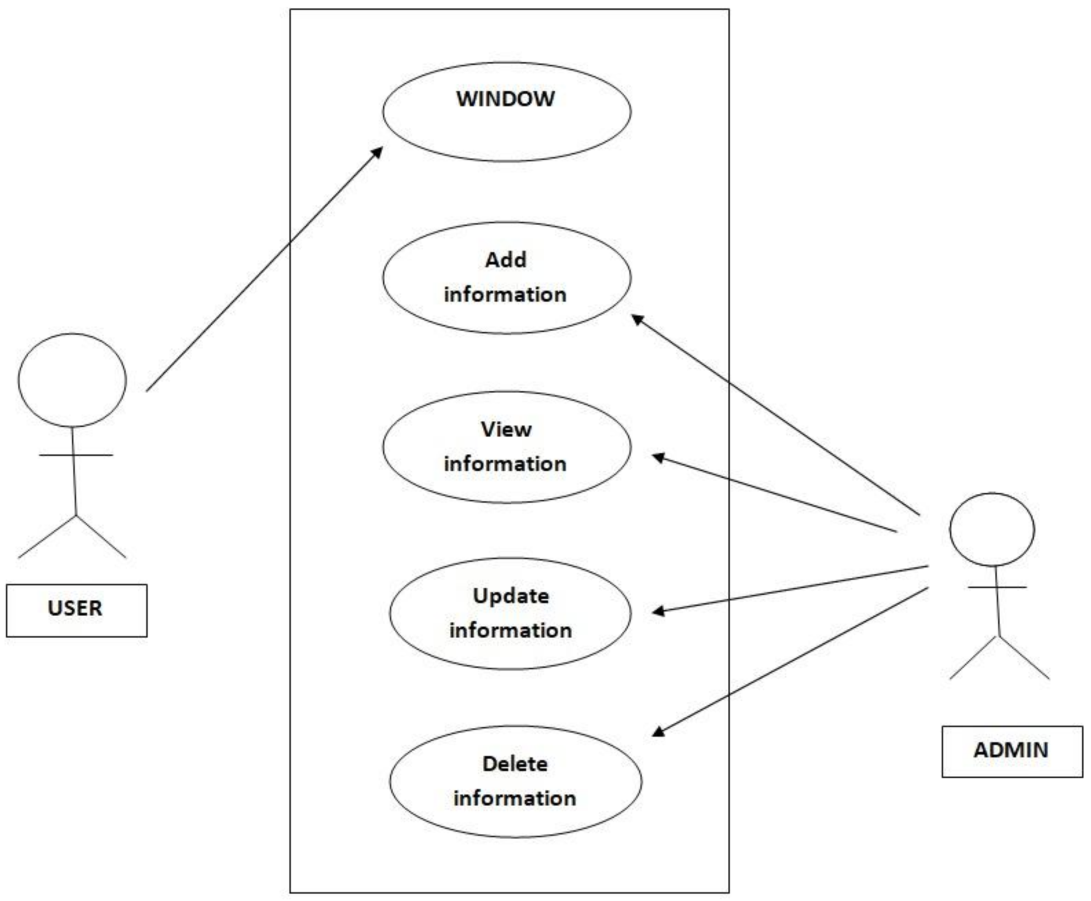


Fig 3.3: Use case diagram of the proposed system

# CHAPTER 4

# METHODOLOGY

The development process of the CIT College Chatbot involved several phases, including:

## 4.1 Planning

To create an interactive chatbot for a college that provides relevant information to users through a graphical user interface (GUI).

**Requirements Gathering:** Identify the chatbot's purpose, key features, and the type of information it should provide. Examples include admission details, contact information, programs offered, and hostel facilities.

**Target Audience:** Prospective students, current students, and parents.

**Tools and Libraries:**

Python programming language.

tkinter for GUI development.

pygame for audio playback.

gTTS for text-to-speech.

speech_recognition for voice input.

## 4.2 Design

**Architecture:** The chatbot consists of three main components:

**Frontend:** A user-friendly GUI using tkinter.

**Backend:** Logic to process user input and fetch relevant responses.

**Audio Processing:** Speech-to-text and text-to-speech functionalities.

### Interface Design:

**Chat History Area:** To display the conversation.

**Input Area:** For text or voice input.

**Buttons:** For sending messages, voice input, saving chat history, and theme switching.

**Menu Bar:** Includes "About" and "Help" sections.

## 4.3 Implementation

The chatbot was implemented in Python using the following steps:

**A. GUI Development with tkinter:**

A Text widget displays chat history.

Entry widget allows user text input.

Buttons and menus provide additional functionality (e.g., voice input, save chat, toggle themes).

**B. Backend Logic:**

**Response Matching:** The chatbot uses keywords to match user queries with predefined responses stored in a dictionary (college_info).

**Dynamic Responses**: Based on user input, relevant details like admissions, fees, and placements are fetched.

**Default Response:** If no keywords match, the chatbot replies with a fallback response.

**C. Audio Functionality:**

**Text-to-Speech (gTTS):** Converts chatbot responses to audio and plays them using pygame.

**Speech-to-Text (speech_recognition):** Converts spoken queries to text for processing.

**D. Additional Features:**

**Save Chat History:** Uses file dialog to save the conversation to a text file.

**Theme Toggle:** Allows switching between light and dark themes.

**FAQs Window:** Provides a list of frequently asked questions for quick access.

## 4.4 Testing

**Functional Testing:**

Verified response accuracy for different user queries.

Ensured smooth operation of text and voice inputs.

Checked the functionality of additional features like saving chat and toggling themes.

**Usability Testing:**

Tested the GUI for readability and ease of navigation.

Evaluated the chatbot's responsiveness to various types of input.

**Audio Testing:**

Ensured audio playback and speech recognition worked correctly across devices.

## 4.5 Deployment

**Execution:**

Packaged the application as a Python script.

Used the requirements.txt file for dependency installation.

**User Accessibility:**

Provided instructions for running the chatbot on local systems.

Recommended Python and library installation for end-users.

## 4.6 Maintenance

Regularly updated information stored in the dictionary (e.g., admissions or placement data).

Addressed bugs reported by users.

Planned enhancements like integration with a database.

## 4.7 Algorithm

Step 1: Initialize libraries and global variables:

Initialize libraries for GUI (Tkinter), audio (pygame), text-to-speech (gTTS), and speech recognition (speech_recognition).

Set up initial variables for audio playback control (is_playing, current_audio_file).

Step 2: Create a college information dictionary (college_info):

Store key pieces of information in this dictionary for quick reference when responding to user queries.

Step 3: Define a play_audio function:

Use gTTS to generate an audio file from the text response.

Play the audio in a separate thread to avoid blocking the main GUI thread.

Delete the audio file after playback.

Step 4: Define a stop_audio function:

Stop the currently playing audio using pygame's music player.

Step 5: Define respond_to_input:

Determine the response to the user's input based on predefined keywords.

Update the chat history UI and trigger audio playback of the response.

Step 6: Define voice_input:

Use speech recognition to capture the user’s voice and convert it to text.

Update the text entry field with the recognized text and call respond_to_input to process the query.

Step 7: Define save_chat:

Prompt the user to choose a file path and save the entire chat history to that file.

Step 8: Define toggle_theme:

Switch between light and dark themes by adjusting the background and text colors.

Step 9: Define show_faq:

Display a list of common questions in a new window.

When a user selects a question, it is entered into the input field and responded to.

Step 10: Define show_about and show_help:

Show information about the chatbot and instructions for usage in dialog boxes.

Step 11: Set up the main Tkinter window (root):

Create the main window and add widgets (buttons, labels, text fields) for user interaction.

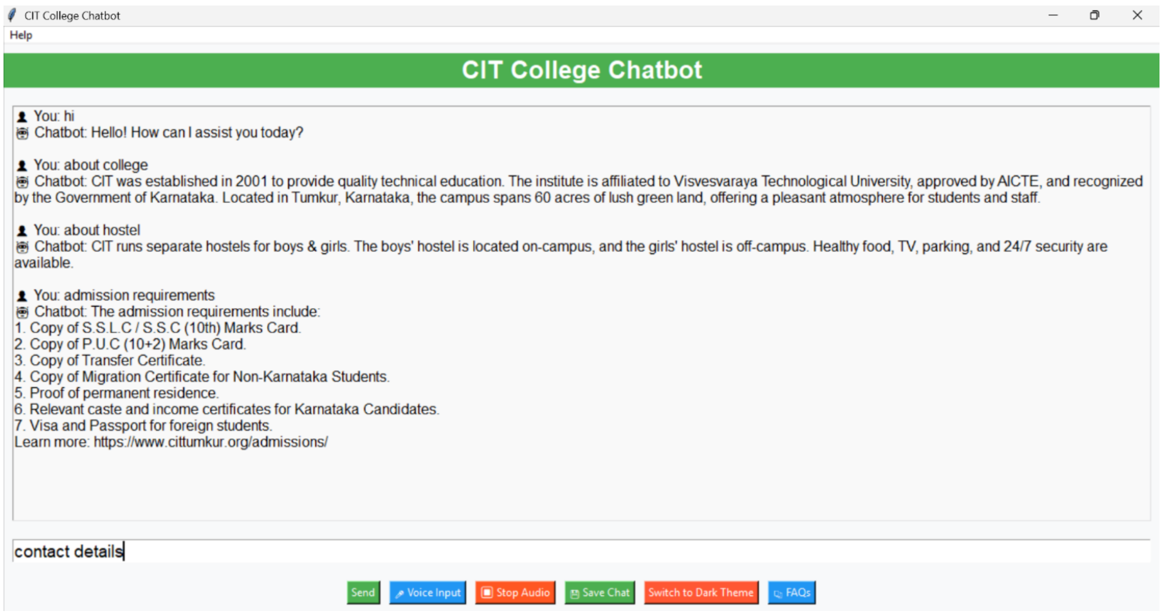
Step 12: Start the Tkinter event loop:

Use root.mainloop() to run the GUI and handle user input.

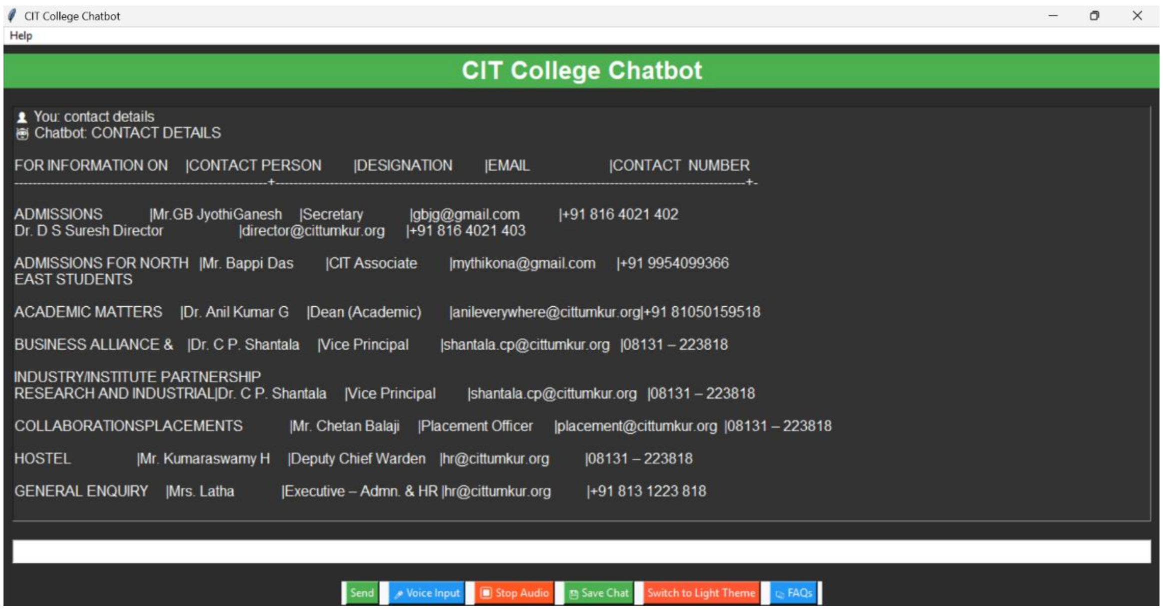
# CHAPTER 5

## RESULTS

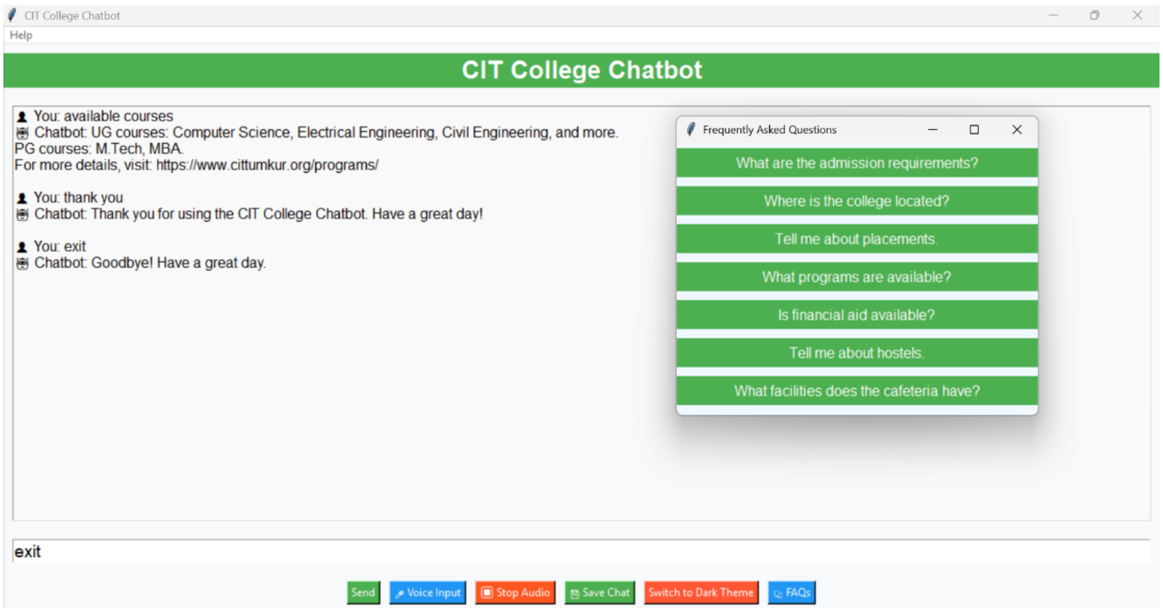
### SCREENSHOTS:



```
You: hi
Chatbot: Hello! How can I assist you today?

You: about college
Chatbot: CIT was established in 2001 to provide quality technical education. The institute is affiliated to Visvesvaraya Technological University, approved by AICTE, and recognized by the Government of Karnataka. Located in Tumkur, Karnataka, the campus spans 60 acres of lush green land, offering a pleasant atmosphere for students and staff.

You: about hostel
Chatbot: CIT runs separate hostels for boys & girls. The boys' hostel is located on-campus, and the girls' hostel is off-campus. Healthy food, TV, parking, and 24/7 security are available.

You: admission requirements
Chatbot: The admission requirements include:
1. Copy of S.S.L.C / S.S.C (10th) Marks Card.
2. Copy of P.U.C (10+2) Marks Card.
3. Copy of Transfer Certificate.
4. Copy of Migration Certificate for Non-Karnataka Students.
5. Proof of permanent residence.
6. Relevant caste and income certificates for Karnataka Candidates.
7. Visa and Passport for foreign students.
Learn more: https://www.cittumkur.org/admissions/

contact details
```

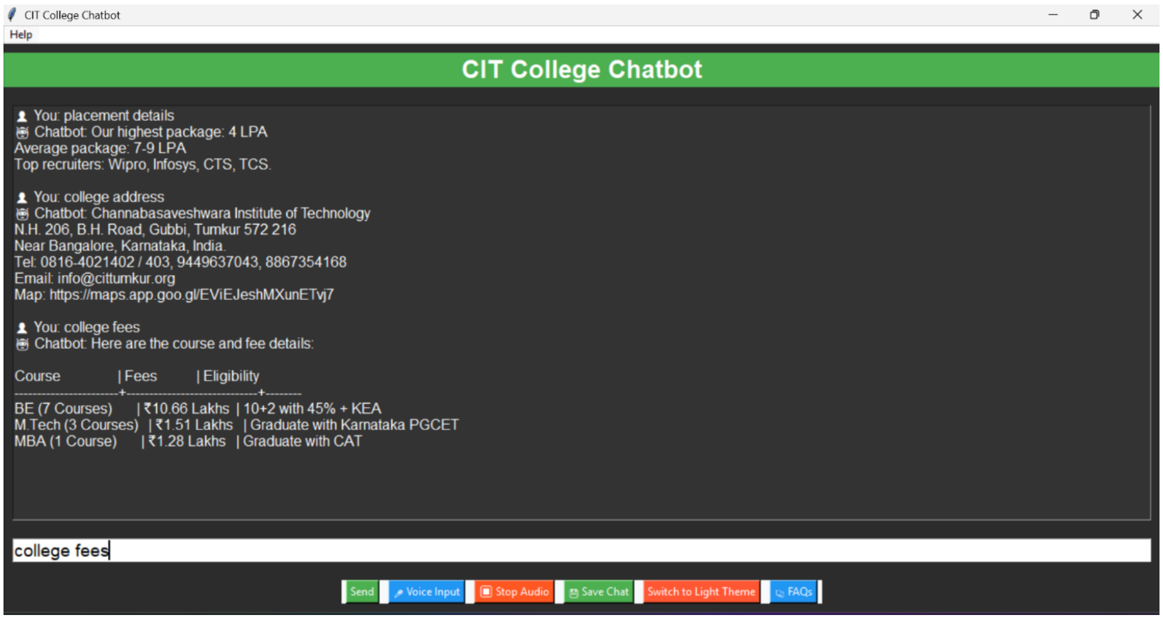


```
You: contact details
Chatbot: CONTACT DETAILS

FOR INFORMATION ON   |CONTACT PERSON    |DESIGNATION    |EMAIL              |CONTACT  NUMBER
--------------------------------------------+-------------------------------------------------------------------------------------+-

ADMISSIONS         |Mr.GB JyothiGanesh   |Secretary        |gbjg@gmail.com        |+91 816 4021 402
Dr. D S Suresh Director               |director@cittumkur.org    |+91 816 4021 403

ADMISSIONS FOR NORTH  |Mr. Bappi Das     |CIT Associate      |mythikona@gmail.com     |+91 9954099366
EAST STUDENTS

ACADEMIC MATTERS    |Dr. Anil Kumar G    |Dean (Academic)     |anileverywhere@cittumkur.org|+91 81050159518

BUSINESS ALLIANCE &   |Dr. C P. Shantala    |Vice Principal      |shantala.cp@cittumkur.org  |08131 – 223818

INDUSTRY/INSTITUTE PARTNERSHIP
RESEARCH AND INDUSTRIAL|Dr. C P. Shantala    |Vice Principal      |shantala.cp@cittumkur.org  |08131 – 223818

COLLABORATIONSPLACEMENTS        |Mr. Chetan Balaji    |Placement Officer    |placement@cittumkur.org  |08131 – 223818

HOSTEL           |Mr. Kumaraswamy H   |Deputy Chief Warden   |hr@cittumkur.org       |08131 – 223818

GENERAL ENQUIRY     |Mrs. Latha        |Executive – Admn. & HR |hr@cittumkur.org       |+91 813 1223 818
```


CIT College Chatbot
Help
CIT College Chatbot
Frequently Asked Questions
What are the admission requirements?
Where is the college located?
Tell me about placements.
What programs are available?
Is financial aid available?
Tell me about hostels.
What facilities does the cafeteria have?
Send
Voice Input
Stop Audio
Save Chat
Switch to Dark Theme
FAQs

```
You: available courses
Chatbot: UG courses: Computer Science, Electrical Engineering, Civil Engineering, and more.
PG courses: M.Tech, MBA.
For more details, visit: https://www.cittumkur.org/programs/

You: thank you
Chatbot: Thank you for using the CIT College Chatbot. Have a great day!

You: exit
Chatbot: Goodbye! Have a great day.
```

exit



```
You: placement details
Chatbot: Our highest package: 4 LPA
Average package: 7-9 LPA
Top recruiters: Wipro, Infosys, CTS, TCS.

You: college address
Chatbot: Channabasaveshwara Institute of Technology
N.H. 206, B.H. Road, Gubbi, Tumkur 572 216
Near Bangalore, Karnataka, India.
Tel: 0816-4021402 / 403, 9449637043, 8867354168
Email: info@cittumkur.org
Map: https://maps.app.goo.gl/EViEJeshMXunETvj7

You: college fees
Chatbot: Here are the course and fee details:

Course              | Fees          | Eligibility
--------------------+---------------+--------
BE (7 Courses)      | ₹10.66 Lakhs  | 10+2 with 45% + KEA
M.Tech (3 Courses)  | ₹1.51 Lakhs   | Graduate with Karnataka PGCET
MBA (1 Course)      | ₹1.28 Lakhs   | Graduate with CAT
```

college fees

# CHAPTER 6

## CONCLUSION

An AI-powered chatbot for college information retrieval is a transformative tool that enhances the accessibility and efficiency of information dissemination. By providing instant, accurate, and user-specific responses, it serves as a 24/7 virtual assistant for students, faculty, and visitors. This chatbot simplifies processes like admission inquiries, campus navigation, course details, and event updates, significantly reducing administrative workload.

With continuous learning capabilities and integration with college databases, the chatbot ensures scalability and adaptability to changing requirements. Its user-friendly interface promotes seamless interaction, fostering better engagement and satisfaction. Overall, implementing an AI chatbot in a college setting is a forward-thinking step toward modernizing campus communication and improving user experience.

# REFERENCE

[1] https://chatbotconf.com.ua/en/article/top-3-glavnih-nedostatkov-chat-botov-66332

[2] The Anatomy of A.L.I.CE : Dr.Richard S.Wallace http://www.alicebot.org/anatomy.html

[3] Artificial Intelligence Markup Language (AIML), A.L.I.C.E. AI Foundation, http://alicebot.org/TR/2001/WD-aiml/

[4] http://developers.facebook.com

[5] AIML Interpreter Overview 2004, http://www.aimlbots.com/en/aiml-interpreters.html

[6] Computing machinery and intelligence, Alan Turing [1950], http://www.abelard.org/turpap/turpap.htm

[7] Using Dialogue Corpora to Train a Chatbot (Bayan Abu Shawar, Eric Atwell) http://www.comp.leeds.ac.uk/andyr/research/papers/tec hreport2003_0