GNDEC CHATBOT

A PROJECT REPORT

GURU NANAK DEV ENGINEERING COLLEGE
in partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Information Technology

Submitted By
Ananya Mahajan(2004889)
Anshika Janghu(2004891)
Diksha(2004903)

Submitted To
Dr. Amit Kamra

Department of Information Technology

GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA, PUNJAB, 141006

December 2023

DECLARATION

We hereby declare that the project report entitled ”**GNDEC CHATBOT**” submitted by us to the GURU NANAK DEV ENGINEERING COLLEGE during the academic year 2023 in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Information Technology is a record of bonafied project work carried out by us under the guidance and supervision of Dr. Amit kamra. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other University.

ANANYA MAHAJAN (2004889)
ANSHIKA JANGHU (2004891)
DIKSHA (2004903)

Place: Ludhiana,Punjab
Date: December 2023

CONTENTS

ACKNOWLEDGEMENT

Many noble hearts contributed immense inspiration and support for the successful completion of the project. We are unable to express our gratitude in words to such individuals.

We would like to express our deep regard to Mr.Sahejpal Singh , Principal, GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA, for providing facilities throughout the works of our project.

We take this opportunity to express our profound gratitude to Dr. K.S.Mann, Head of the Department, Department of Information Technology, GURU NANAK DEV ENGINEERING COLLEGE, for providing permission and availing all required facilities for undertaking the project in a systematic way. We are extremely grateful to Dr. Amit kamra, Associate Professor, Department of Information Technology, GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA, who guided us with his kind, ordinal and valuable suggestions. We pay our deep sense of gratitude to Dr. Amit kamra and project coordinators, Department of Information Technology, GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA, for their valuable guidance, keen interest and encouragement at various stages of the project. We would also like to thank all the teaching and non teaching staff of Department of Information Technology, GURU NANAK DEV ENGINEERING COLLEGE for the sincere directions imparted and the cooperation in connection with the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this project.

Last, but not the least, We take pleasant privilege in expressing our heartful thanks to our friends who were of precious help in completing this project.

GUIDE
Dr. Amit kamra
Associate Professor
Dept. of Information Technology

HEAD OF THE DEPARTMENT
Dr. K.S.Mann
Associate Professor
Dept. of Information Technology

ABSTRACT

In the rapidly evolving landscape of education, technological advancements play a pivotal role in shaping the learning experience. This abstract introduces an innovative solution: an Intelligent College Chatbot designed to revolutionize student engagement and support within the higher education setting. The chatbot leverages natural language processing (NLP) and artificial intelligence (AI) to create an interactive and personalized interface for students, faculty, and staff.

The primary objectives of the Intelligent College Chatbot are to streamline communication, provide timely and relevant information, and enhance overall user experience. The chatbot is equipped to assist with a wide range of tasks, including course inquiries, event updates, campus resources, academic advising, and general information retrieval. Through continuous learning and adaptation, the chatbot evolves to meet the dynamic needs of the college community.

The deployment of an Intelligent College Chatbot holds the potential to transform the way students engage with their academic environment. By automating routine inquiries, the chatbot enables college staff to focus on more complex tasks, ultimately fostering a more efficient and supportive educational ecosystem. This abstract provides a glimpse into the future of education, where AI-driven solutions enhance accessibility, communication, and overall student success.

Utilizing natural language processing capabilities, the chatbot engages users in a conversational manner, making the interaction intuitive and accessible to individuals with varying levels of technical expertise. The Enquiry Chatbot is accessible through multiple channels, including the college website, social media platforms, and messaging applications, providing users with flexibility and convenience in seeking information.

LIST OF FIGURES

m

CHAPTER 1
INTRODUCTION

1.1 GNDEC CHATBOT

A chatbot is software that simulates human-like conversations with users via text messages on chat. Its key task is to help users by providing answers to their questions. This could be a text based (typed) conversation, a spoken conversation or even a non-verbal conversation. Chat bot is typically perceived as engaging software entity which humans can talk to. It can be interesting, inspiring and intriguing. It appears everywhere, from old ancient HTML pages to modern advanced social networking.

College Enquiry Chatbot uses machine learning concepts to have conversations with humans. The purpose of developing this project is based on an intellectual chat-bot system which will deal with the academic activities like admission enquiry, fees structure, scholarship details, time-table of every department, details of the documents required to attach etc. With this chat-bot system it will be easy for the student to directly clear their queries in lesser time. Chat bots typically provide a text-based user interface, allowing the user to type commands and receive text in order to resolve the query. The Chatbot has information stored in its dataset to identify the sentences and making a decision itself as response to answer a given question. The program analyzes the user's query then the bot responds to the query.

1.2 IMPORTANT TERMS USED IN GNDEC CHATBOT

1. NLP
2. NumPy
3. TfidfVectorizer
4. Bot token
5. Tokenization
6. Lemmatization
7. Telegram Bot API
8. Requests Library

1.3 OBJECTIVES

1. To provide accurate and up-to-date information about the college, admission procedures, eligibility criteria, and important dates.

2. To automate routine responses to common queries, such as application status, admission requirements, and program details, freeing up human resources for more complex inquiries.

3. To answer frequently asked questions (FAQs) regarding admissions, courses, fees, scholarships, and other relevant topics to reduce the load on human support staff.

CHAPTER 2
REQUIREMENT ANALYSIS, SYSTEM SPECIFICATIONS

2.1 FEASIBILITY STUDY

- **Identification of Objectives**:
  Clearly define the objectives of implementing a chatbot. Understand the specific problems or challenges it aims to address, such as reducing workload on admissions staff, improving user experience, and increasing engagement.

- **Scope and Requirements**
  Define the scope of the chatbot's functionalities. Identify the features and capabilities it should have, such as answering FAQs, providing application assistance, and offering personalized guidance.

- **Technical Feasibility**
  Assess the technical requirements and infrastructure needed to implement the chatbot. Consider factors such as compatibility with existing systems, integration capabilities, and the availability of skilled personnel to develop and maintain the chatbot.

- **Cost Estimation**
  Determine the initial and ongoing costs associated with developing, implementing, and maintaining the chatbot. Consider factors like development costs, hosting fees, and potential costs for updates and improvements.

- **User Acceptance and Adoption**
  Assess the willingness of users (both prospective students and internal staff) to adopt and interact with the chatbot. Consider conducting surveys or focus groups to gauge expectations and concerns.

- **Scalability**
  Consider whether the chatbot can scale to accommodate an increasing number of users and a growing database of information. Evaluate its ability to handle peak times, such as admission application deadlines.

2.2 SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT

2.2.1 Tools Required

Hardware Requirements

1. RAM: A minimum of 8 GB is required as training any algorithm will require some heavy Lifting. Less than 8 GB can cause problems while Multitasking.
2. OS: This system can run on any latest LTS OS.
3. Processor: Intel i5 10th Gen or above or Ryzen 5 4th Gen or above

Software Requirements
This Telegram chatbot is implemented using Python and incorporates several libraries and technologies for natural language processing (NLP) and interaction with the Telegram API. Here's a breakdown of the tools and technologies used:

1. **Programming Language**
   PYTHON
   Python is the programming language used for implementing the chatbot.
   Python is a high-level, interpreted programming language known for its readability and simplicity. It was created by Guido van Rossum and first released in 1991. Python is designed to be easy to learn and has a clean and straightforward syntax, making it an excellent choice for beginners and experienced programmers alike.

   Here are some key features and characteristics of Python

   - Readability:
     Python emphasizes readability and uses English keywords, making it easy for developers to write and maintain code.

   - Versatility:
     Python is a general-purpose programming language, meaning it can be used for a wide range of applications, including web development, data science, machine learning, artificial intelligence, automation, scripting, and more.

   - Interpreted Language:
     Python is an interpreted language, which means that the source code is ex-

ecuted line by line by an interpreter. This allows for a more interactive and dynamic development process.

- Dynamic Typing:
  Python is dynamically typed, meaning you don't need to specify the data type of a variable explicitly. The interpreter infers the type during runtime.

- Large Standard Library:
  Python comes with a comprehensive standard library that provides modules and packages for various tasks, such as working with files, handling data structures, networking, and more.

- Community and Ecosystem:
  Python has a large and active community of developers. There are numerous third-party libraries and frameworks available, contributing to a rich ecosystem that extends the language's capabilities.

- Object-Oriented:
  Python supports object-oriented programming (OOP) principles, allowing developers to create and use classes and objects.

- Cross-Platform:
  Python is designed to be cross-platform, meaning code written in Python can run on different operating systems without modification.

- Open Source:
  Python is an open-source language, and its interpreter, Python, is available under the Python Software Foundation License.

- High-Level Data Structures:
  Python provides built-in high-level data structures, such as lists, tuples, sets, and dictionaries, which simplify programming tasks and data manipulation.

- Dynamically Typed:

Python is dynamically typed, allowing variables to change types during runtime.

- Indentation:
  Python uses indentation to define code blocks rather than relying on braces or keywords. This forces a consistent and clean coding style.

Python's popularity has grown significantly over the years, and it has become one of the most widely used programming languages in various domains, from web development and scientific computing to artificial intelligence and machine learning. The simplicity and versatility of Python make it a preferred choice for many developers and organizations.

2. **Natural Language Processing (NLP) Libraries**
   NLP, or Natural Language Processing, is a field of artificial intelligence (AI) that focuses on the interaction between computers and humans using natural language. The ultimate objective of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant.

   Key components and tasks within the field of NLP include

   - Text Tokenization:

     Breaking down a text into individual units, such as words or phrases, known as tokens.

   - Part-of-Speech Tagging (POS):

     Assigning grammatical categories (e.g., noun, verb, adjective) to each word in a sentence.

   - Named Entity Recognition (NER):

     Identifying and classifying entities in text, such as names of people, organizations, locations, dates, and more.

- Sentiment Analysis:

  Determining the sentiment expressed in a piece of text, typically as positive, negative, or neutral.

- Text Classification:

  Categorizing text into predefined categories or classes based on its content.

- Machine Translation:

  Automatically translating text from one language to another.

- Speech Recognition:

  Converting spoken language into written text. Coreference Resolution:

  Identifying when two or more expressions in a text refer to the same entity.

- Question Answering:

  Developing systems that can answer questions posed in natural language.

- Text Summarization:

  Generating concise and meaningful summaries of longer pieces of text.

- Language Modeling:

  Building models that predict the likelihood of a sequence of words occurring in a given context.

- Dependency Parsing:

  Analyzing the grammatical structure of sentences by identifying the relationships between words.

- Information Extraction:

  Extracting structured information from unstructured text.

- Chatbots and Conversational Agents:

  Building systems that can engage in natural language conversations with users.

- Core NLP Libraries:

  Utilizing specialized libraries and frameworks (e.g., NLTK, Spacy, Transformers) that provide tools for various NLP tasks.

- **nltk (Natural Language Toolkit)** Used for tokenization, lemmatization, and other natural language processing tasks.

- nltk.sent$_{t}okenize$: Tokenizes paragraphs into sentences.

- nltk.word$_{t}okenize$: Tokenizes sentences into words.

- nltk.stem.WordNetLemmatizer: Performs lemmatization on words.

3. **Data Processing**
   NumPy (np)
   Used for numerical operations.
   Typically used for efficient array operations.

4. **Text Preprocessing**
   String Module
   Used for handling string-related operations.
   string.punctuation: Contains all punctuation characters.
   ord(punct): Converts punctuation to its ASCII value.
   translate: Replaces punctuation marks with None.

5. **Machine Learning Libraries**
   scikit-learn
   Utilizes the TfidfVectorizer for converting text data into TF-IDF matrices.

6. **Random Module**
   random.choice: Used to randomly select responses from predefined lists.

7. **Web Scraping and API Interaction**
   Requests Library
   Used for making HTTP requests.
   Interacts with the Telegram API to send and receive messages.

8. **Telegram API Integration**
   Telegram Bot API
   The chatbot interacts with the Telegram API to send and receive messages.
   telegrambot class is created to encapsulate methods for working with the Telegram API.
   Methods include getupdates to retrieve updates, sendmessage to send messages, and grabtoken to obtain the API token.

9. **Infinite Loop and Event Handling**
   The script operates in an infinite loop, continuously checking for updates from the Telegram API.
   It handles received updates, processes user messages, and sends appropriate responses.

10. **Tokenization and Lemmatization**
    Tokenization is performed to break text into sentences and words.
    Lemmatization is applied to reduce words to their base or root form.

11. **TF-IDF Vectorization** The script uses TF-IDF vectorization to convert text data into a numerical representation.

12. **Error Handling**
    The script includes some error handling, such as checking for the presence of keys in dictionaries to avoid potential runtime errors.

2.2.2 Data Requirement

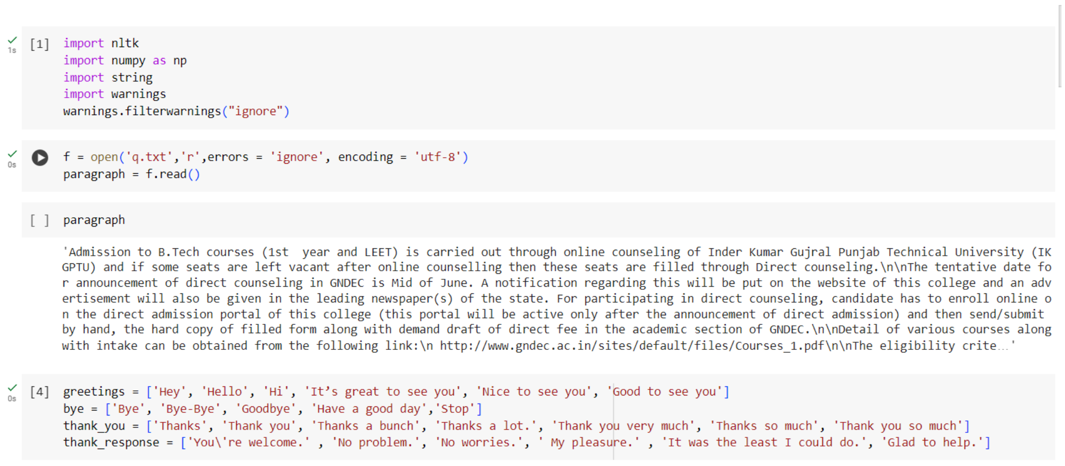
(a) **Telegram API Token**
The Telegram bot requires a token for authentication and communication with the Telegram API. You obtain this token by creating a new bot on Telegram through the BotFather.

(b) **Knowledge Base (q.txt)**

The code reads a knowledge base from a file named ’q.txt’. This file likely contains information that the chatbot will use to respond to user queries. Ensure that this file is present and contains relevant content.

(c) **Predefined Responses and Lists**

The code uses predefined lists for greetings, goodbye, thank-you expressions, and corresponding bot responses. Ensure these lists are appropriately populated. 2.1



```
[1] import nltk
    import numpy as np
    import string
    import warnings
    warnings.filterwarnings("ignore")

    f = open('q.txt','r',errors = 'ignore', encoding = 'utf-8')
    paragraph = f.read()

[ ] paragraph

    'Admission to B.Tech courses (1st  year and LEET) is carried out through online counseling of Inder Kumar Gujral Punjab Technical University (IK
    GPTU) and if some seats are left vacant after online counselling then these seats are filled through Direct counseling.\n\nThe tentative date fo
    r announcement of direct counseling in GNDEC is Mid of June. A notification regarding this will be put on the website of this college and an adv
    ertisement will also be given in the leading newspaper(s) of the state. For participating in direct counseling, candidate has to enroll online o
    n the direct admission portal of this college (this portal will be active only after the announcement of direct admission) and then send/submit
    by hand, the hard copy of filled form along with demand draft of direct fee in the academic section of GNDEC.\n\nDetail of various courses along
    with intake can be obtained from the following link:\n http://www.gndec.ac.in/sites/default/files/Courses_1.pdf\n\nThe eligibility crite...'

[4] greetings = ['Hey', 'Hello', 'Hi', 'It’s great to see you', 'Nice to see you', 'Good to see you']
    bye = ['Bye', 'Bye-Bye', 'Goodbye', 'Have a good day','Stop']
    thank_you = ['Thanks', 'Thank you', 'Thanks a bunch', 'Thanks a lot.', 'Thank you very much', 'Thanks so much', 'Thank you so much']
    thank_response = ['You\'re welcome.' , 'No problem.', 'No worries.', ' My pleasure.' , 'It was the least I could do.', 'Glad to help.']
```

Figure 2.1: Predefined responses and lists

2.3 SDLC MODEL TO BE USED

The SDLC (Software Development Life Cycle) model represents the overall process followed in the development of software applications. It defines a series of phases or stages that guide the development process from initiation to deployment and maintenance. Here are the commonly recognized phases in the SDLC:

(a) **Requirements Gathering**

In this phase, the project team identifies and documents the requirements of the software based on the needs of stakeholders and end-users. This includes gathering functional and non-functional requirements.

(b) **Analysis and Planning**

The gathered requirements are analyzed, and a detailed project plan is created. This involves defining the scope, estimating resources and timelines, and identifying potential risks and challenges.

(c) **Design**

The design phase involves creating the system architecture, software design, database schema, user interfaces, and other technical specifications. It lays the foundation for the development process.

(d) **Development**

In this phase, the actual coding and development of the software take place. Programmers write code based on the design specifications, following coding standards and best practices.

(e) **Testing**

The software is thoroughly tested to ensure that it functions as expected and meets the defined requirements. This includes various types of testing, such as unit testing, integration testing, system testing, and user acceptance testing.

(f) **Deployment**

Once the software has passed the testing phase, it is deployed to the production environment or made available to end-users. This involves activities like installation, configuration, data migration, and user training.

(g) **Maintenance**

After deployment, the software enters the maintenance phase. This includes ongoing support, bug fixes, performance enhancements, and updates to meet changing requirements or address any issues that arise during its usage.

CHAPTER 3
METHODOLOGY

3.1 INTRODUCTION TO METHODOLOGY

A methodology is a technique designed to carry out research, a task, or any particular activity. Any activity that is performed with some predetermined fixed steps is known as methodology. It is a framework for the theoretical analysis of a definite task.

Methodologies play a critical role in the continuous improvement of the performed task. The constructive approach used by the researchers leads to better work efficiency. Knowledge and reality of the surroundings are significant to create a methodology for different activities.

It is a quantitative approach that can efficiently integrate with the interdisciplinary theoretical perspective. Let's understand this with an example. The middle school and higher secondary schools use different pedagogy. The learning style differs vastly for the primary students, seniors, and even the co-education system.

3.2 METHODOLOGY OF COLLEGE ENQUIRY CHATBOT

The college enquiry chatbot methodology involves a systematic and user-centric approach to ensure the successful development and deployment of an effective conversational interface for the college community.Developing a college enquiry chatbot involves a structured methodology to ensure successful implementation.

3.2.1 User Requirements Gathering

In the initial phase of our college enquiry chatbot project, a meticulous user requirement gathering process was conducted to ensure a deep understanding of the diverse needs and expectations of our college community. Engaging key stakeholders, including students, faculty, staff, and administrators, was paramount. Through surveys, one-on-one interviews, and focus group discussions, we gained valuable insights into the information users seek, preferred communication channels, and challenges faced in accessing college-related data. Analysis of existing communication channels, FAQs, and an exploration of college processes further informed our understanding of user expectations..

3.2.2 Data Collection and Preprocessing

Raw data, encompassing details on courses, events, and FAQs, was extracted from diverse sources, including databases and official documents. Rigorous data cleaning procedures were then applied to eliminate redundancies and handle missing or inconsistent information. Leveraging natural language processing (NLP) techniques, the text underwent tokenization, stopwords removal, and lemmatization to facilitate a comprehensive understanding of the data's structure. The preprocessed data was meticulously organized into a knowledge base, ensuring efficient access for the chatbot. Intents and entities were defined to align user queries with relevant information. Special consideration was given to handling multi-turn conversations, enabling the chatbot to maintain context over extended interactions. To enhance the diversity of the training dataset, optional data augmentation techniques were explored. The knowledge base and preprocessing steps underwent rigorous testing and validation, incorporating feedback and iterative refinement to optimize the chatbot's performance continuously. .

3.2.3 Chatbot Design and Architecture

The design and architecture of our college enquiry chatbot were crafted with a focus on simplicity, efficiency, and seamless integration. Leveraging well-established Python libraries, including the Natural Language Toolkit (nltk) for tokenization and lemmatization, and scikit-learn for machine learning tasks like TF-IDF vectorization and cosine similarity calculations, we ensured a robust foundation for natural language processing (NLP) capabilities. The chatbot's architecture revolves around the Telegram messaging platform, with integration facilitated through BotFather—an intuitive tool provided by Telegram for creating and managing bots. The modular and scalable design of the chatbot allows for easy expansion of functionalities and seamless adaptation to future enhancements. By integrating with Telegram, our chatbot becomes easily accessible to the college community, providing a user-friendly interface for students, faculty, and staff to make inquiries and receive information efficiently. This design and architecture blend the power of established Python libraries with the accessibility of the Telegram platform, creating a versatile and user-centric tool for streamlined communication within the college environment..
3.1

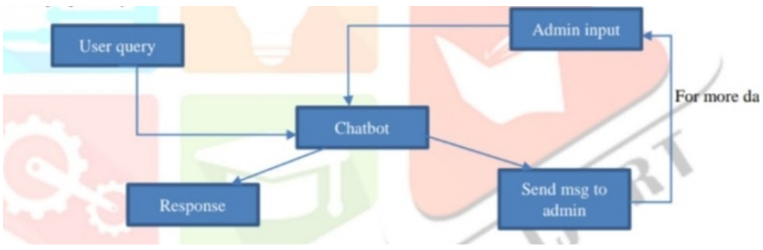
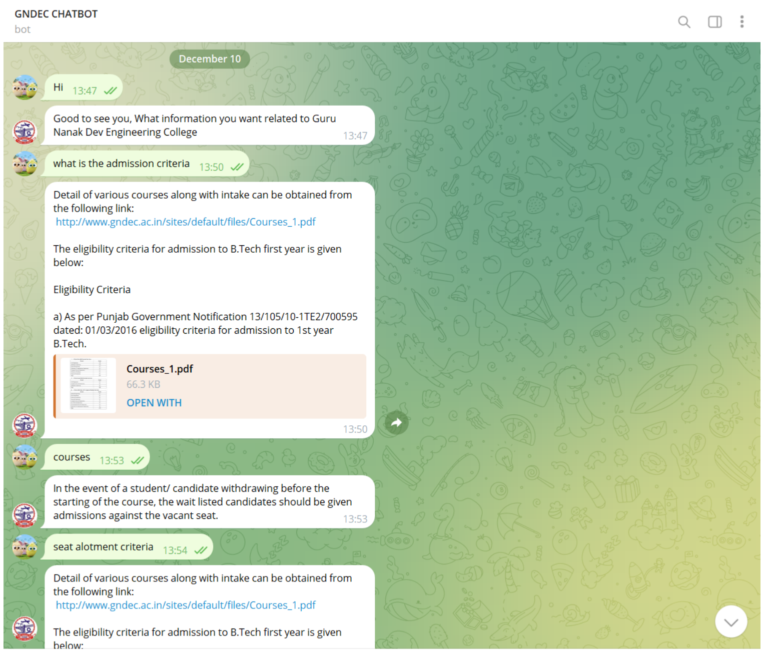


Figure 3.1: Flowchart

3.2


GNDEC CHATBOT
bot

December 10

Hi 13:47

Good to see you, What information you want related to Guru Nanak Dev Engineering College 13:47

what is the admission criteria 13:50

Detail of various courses along with intake can be obtained from the following link:
http://www.gndec.ac.in/sites/default/files/Courses_1.pdf

The eligibility criteria for admission to B.Tech first year is given below:

Eligibility Criteria

a) As per Punjab Government Notification 13/105/10-1TE2/700595 dated: 01/03/2016 eligibility criteria for admission to 1st year B.Tech.

Courses_1.pdf
66.3 KB
OPEN WITH
13:50

courses 13:53

In the event of a student/ candidate withdrawing before the starting of the course, the wait listed candidates should be given admissions against the vacant seat. 13:53

seat alotment criteria 13:54

Detail of various courses along with intake can be obtained from the following link:
http://www.gndec.ac.in/sites/default/files/Courses_1.pdf

The eligibility criteria for admission to B.Tech first year is given below:

Figure 3.2: User Interface

The code structure is based on the polling approach to get updates from Telegram. It continuously checks for updates with a specified timeout.
The chatbot responds based on predefined rules, TF-IDF similarity, and cosine similarity with the knowledge base. This methodology allows the chatbot to un derstand and respond to user queries related to admission in GNDEC based on the information provided in the knowledge base. The chatbot is designed to handle greetings, goodbyes, and thank-yous, and it provides relevant information in response to user queries.

CHAPTER 4
RESULT AND DISCUSSION

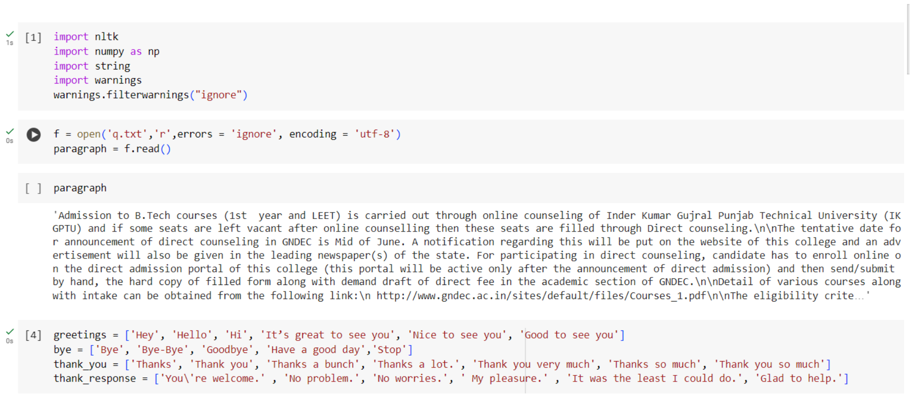
4.1 GNDEC CHATBOT

The chatbot processes user messages, generates responses based on predefined patterns and similarity calculations, and sends replies back to the users.
The basic implementation of a Telegram chatbot using Python, NLTK for natural language processing, and the Telegram API for communication. The code includes features such as greeting responses, farewell responses, and responses based on the user's input related to admission information in GNDEC.

4.1.1 Discussion

(a) **Importing Libraries** The code begins by importing necessary libraries:

- nltk for natural language processing.
- numpy as np for numerical operations.
- string for handling string operations.
- warnings for filtering out warning messages. 4.1



```
[1] import nltk
    import numpy as np
    import string
    import warnings
    warnings.filterwarnings("ignore")
```

```
f = open('q.txt','r',errors = 'ignore', encoding = 'utf-8')
paragraph = f.read()
```

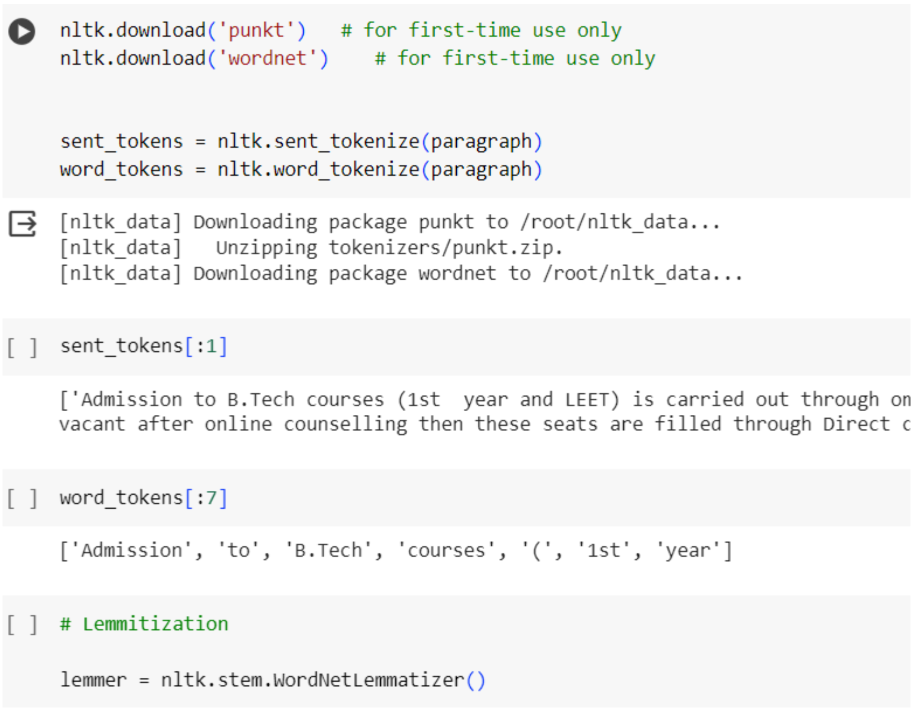
```
[ ] paragraph

'Admission to B.Tech courses (1st  year and LEET) is carried out through online counseling of Inder Kumar Gujral Punjab Technical University (IK
GPTU) and if some seats are left vacant after online counselling then these seats are filled through Direct counseling.\n\nThe tentative date fo
r announcement of direct counseling in GNDEC is Mid of June. A notification regarding this will be put on the website of this college and an adv
ertisement will also be given in the leading newspaper(s) of the state. For participating in direct counseling, candidate has to enroll online o
n the direct admission portal of this college (this portal will be active only after the announcement of direct admission) and then send/submit
by hand, the hard copy of filled form along with demand draft of direct fee in the academic section of GNDEC.\n\nDetail of various courses along
with intake can be obtained from the following link:\n http://www.gndec.ac.in/sites/default/files/Courses_1.pdf\n\nThe eligibility crite...'
```

```
[4] greetings = ['Hey', 'Hello', 'Hi', 'It’s great to see you', 'Nice to see you', 'Good to see you']
    bye = ['Bye', 'Bye-Bye', 'Goodbye', 'Have a good day','Stop']
    thank_you = ['Thanks', 'Thank you', 'Thanks a bunch', 'Thanks a lot.', 'Thank you very much', 'Thanks so much', 'Thank you so much']
    thank_response = ['You\'re welcome.' , 'No problem.', 'No worries.', ' My pleasure.' , 'It was the least I could do.', 'Glad to help.']
```

Figure 4.1: Importing Libraries

(b) **Natural Language Processing** The code uses NLTK for text processing

and TF-IDF vectorization for finding cosine similarity between user queries and the knowledge base. This approach is reasonable for a basic chatbot but might not be sufficient for more complex conversational scenarios.
4.2



```
nltk.download('punkt')   # for first-time use only
nltk.download('wordnet')    # for first-time use only


sent_tokens = nltk.sent_tokenize(paragraph)
word_tokens = nltk.word_tokenize(paragraph)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
[ ] sent_tokens[:1]
```

```
['Admission to B.Tech courses (1st  year and LEET) is carried out through on
vacant after online counselling then these seats are filled through Direct c
```

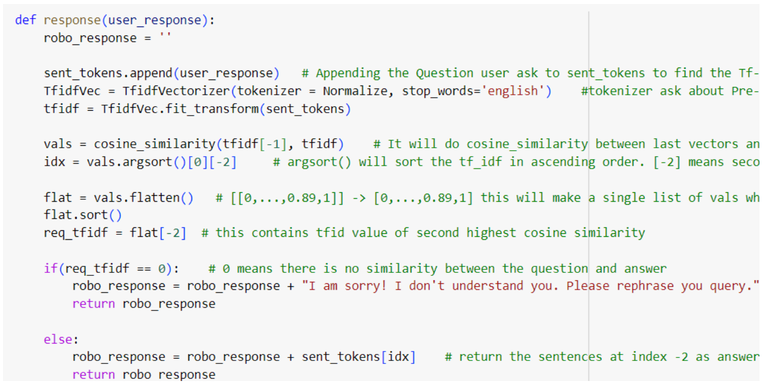
```
[ ] word_tokens[:7]
```

```
['Admission', 'to', 'B.Tech', 'courses', '(', '1st', 'year']
```

```
[ ] # Lemmitization

    lemmer = nltk.stem.WordNetLemmatizer()
```

Figure 4.2: Using NLTK

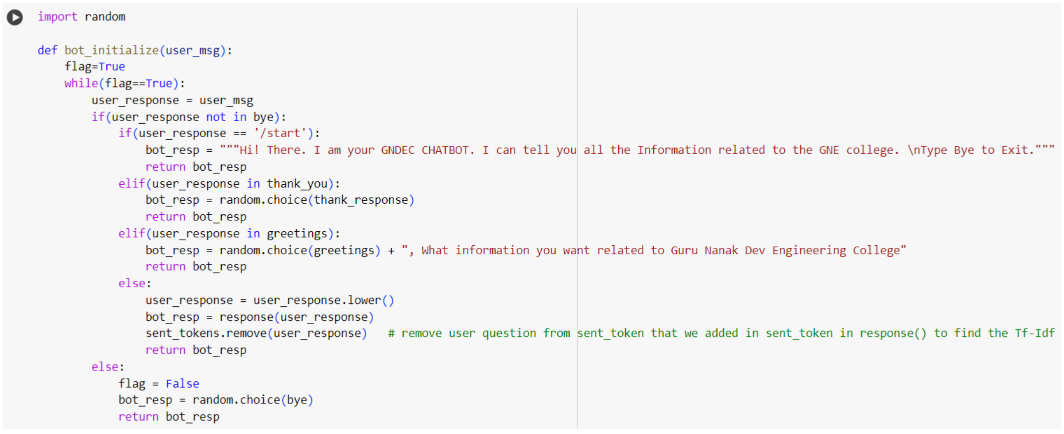
(c) **Response Generation** The response generation involves computing TF-IDF vectors and cosine similarity. However, the code does not handle cases where the cosine similarity is below a certain threshold. It might be useful to set a threshold to filter out less relevant responses.
4.3



```
def response(user_response):
    robo_response = ''

    sent_tokens.append(user_response)   # Appending the Question user ask to sent_tokens to find the Tf-
    TfidfVec = TfidfVectorizer(tokenizer = Normalize, stop_words='english')    #tokenizer ask about Pre-
    tfidf = TfidfVec.fit_transform(sent_tokens)

    vals = cosine_similarity(tfidf[-1], tfidf)    # It will do cosine_similarity between last vectors an
    idx = vals.argsort()[0][-2]     # argsort() will sort the tf_idf in ascending order. [-2] means seco

    flat = vals.flatten()   # [[0,...,0.89,1]] -> [0,...,0.89,1] this will make a single list of vals wh
    flat.sort()
    req_tfidf = flat[-2]  # this contains tfid value of second highest cosine similarity

    if(req_tfidf == 0):    # 0 means there is no similarity between the question and answer
        robo_response = robo_response + "I am sorry! I don't understand you. Please rephrase you query."
        return robo_response

    else:
        robo_response = robo_response + sent_tokens[idx]    # return the sentences at index -2 as answer
        return robo_response
```

Figure 4.3: Generating response

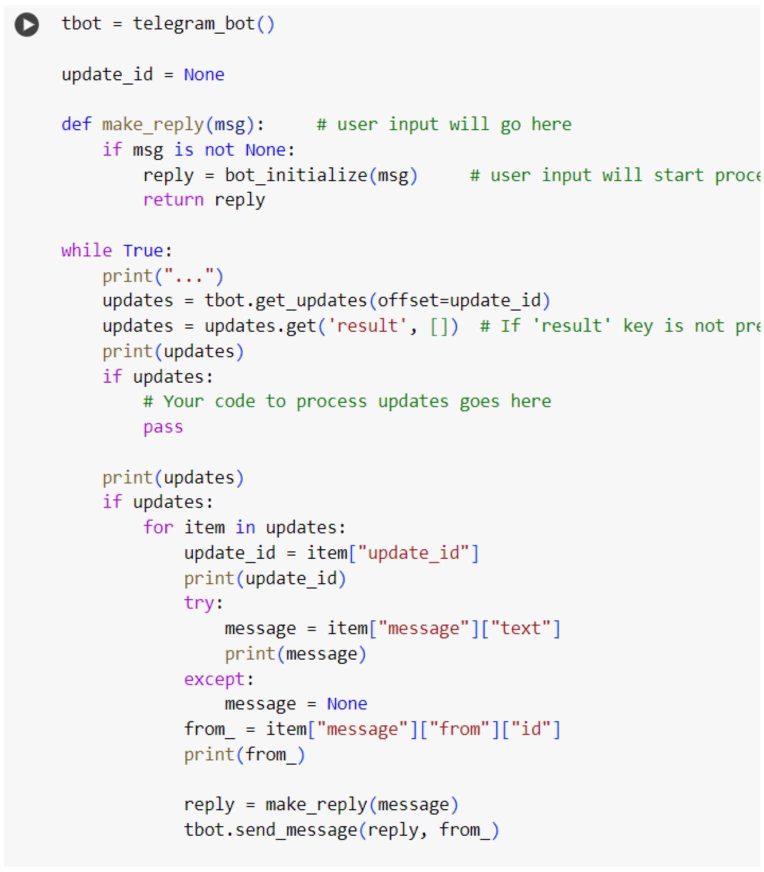
(d) **Telegram Bot Initialization** The Telegram bot initialization is handled well, and the bot provides a brief introduction when the user sends the ”/start” command.
4.4



```python
import random

def bot_initialize(user_msg):
    flag=True
    while(flag==True):
        user_response = user_msg
        if(user_response not in bye):
            if(user_response == '/start'):
                bot_resp = """Hi! There. I am your GNDEC CHATBOT. I can tell you all the Information related to the GNE college. \nType Bye to Exit."""
                return bot_resp
            elif(user_response in thank_you):
                bot_resp = random.choice(thank_response)
                return bot_resp
            elif(user_response in greetings):
                bot_resp = random.choice(greetings) + ", What information you want related to Guru Nanak Dev Engineering College"
                return bot_resp
            else:
                user_response = user_response.lower()
                bot_resp = response(user_response)
                sent_tokens.remove(user_response)   # remove user question from sent_token that we added in sent_token in response() to find the Tf-Idf
                return bot_resp
        else:
            flag = False
            bot_resp = random.choice(bye)
            return bot_resp
```

Figure 4.4: Telegram Bot Initialization

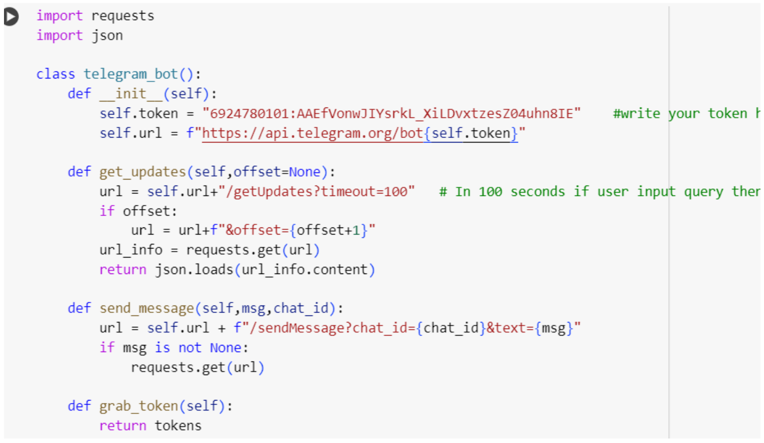
(e) **User Interaction Loop** The main loop continuously checks for updates from the Telegram API and responds to user messages. This loop runs indefinitely, and it might be beneficial to include a sleep or delay to avoid unnecessary high-frequency requests to the Telegram API.
This loop ensures that the bot continuously listens for new messages, generates responses, and sends them back to the users. Keep in mind that in a production environment, you might want to include a delay or sleep between iterations to avoid making too many requests to the Telegram API in a short period.
Retrieves updates from the Telegram API, with an offset parameter to get only new updates.
Extracts the ’result’ field from the updates, which contains information about new messages or events.
Iterates through each update in the updates list.
Sends the generated reply back to the user using the Telegram bot’s sendmessage method.
4.5



```
tbot = telegram_bot()

update_id = None

def make_reply(msg):     # user input will go here
    if msg is not None:
        reply = bot_initialize(msg)     # user input will start proce
        return reply

while True:
    print("...")
    updates = tbot.get_updates(offset=update_id)
    updates = updates.get('result', [])  # If 'result' key is not pre
    print(updates)
    if updates:
        # Your code to process updates goes here
        pass

    print(updates)
    if updates:
        for item in updates:
            update_id = item["update_id"]
            print(update_id)
            try:
                message = item["message"]["text"]
                print(message)
            except:
                message = None
            from_ = item["message"]["from"]["id"]
            print(from_)

            reply = make_reply(message)
            tbot.send_message(reply, from_)
```

Figure 4.5: User Interaction

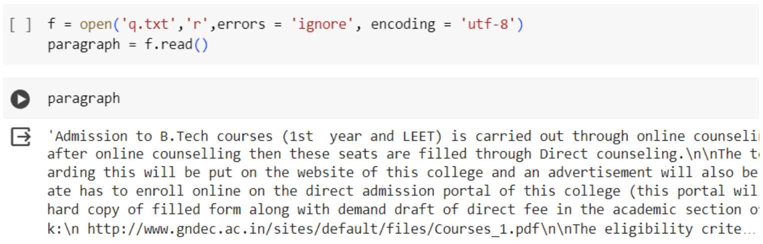
(f) **Token Security** The Telegram bot token is hardcoded in the code. It's essential to keep tokens secure and not expose them in the code directly. Consider using environment variables or a configuration file to store sensitive information.
4.6



```
import requests
import json

class telegram_bot():
    def __init__(self):
        self.token = "6924780101:AAEfVonwJIYsrkL_XiLDvxtzesZ04uhn8IE"    #write your token h
        self.url = f"https://api.telegram.org/bot{self.token}"

    def get_updates(self,offset=None):
        url = self.url+"/getUpdates?timeout=100"   # In 100 seconds if user input query ther
        if offset:
            url = url+f"&offset={offset+1}"
        url_info = requests.get(url)
        return json.loads(url_info.content)

    def send_message(self,msg,chat_id):
        url = self.url + f"/sendMessage?chat_id={chat_id}&text={msg}"
        if msg is not None:
            requests.get(url)

    def grab_token(self):
        return tokens
```

Figure 4.6: Providing token security

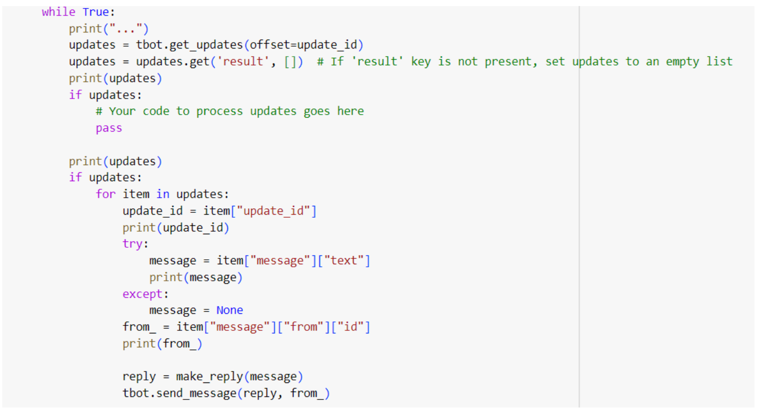
(g) **Knowledge Base** The effectiveness of the chatbot heavily depends on the quality and relevance of the knowledge base (content in ’q.txt’). Ensure that the knowledge base is informative and covers a range of topics related to college admission in GNDEC.
4.7



```
[ ] f = open('q.txt','r',errors = 'ignore', encoding = 'utf-8')
    paragraph = f.read()

    paragraph

    'Admission to B.Tech courses (1st  year and LEET) is carried out through online counseli
    after online counselling then these seats are filled through Direct counseling.\n\nThe t
    arding this will be put on the website of this college and an advertisement will also be
    ate has to enroll online on the direct admission portal of this college (this portal wil
    hard copy of filled form along with demand draft of direct fee in the academic section o
    k:\n http://www.gndec.ac.in/sites/default/files/Courses_1.pdf\n\nThe eligibility crite...
```

Figure 4.7: Providing dataset

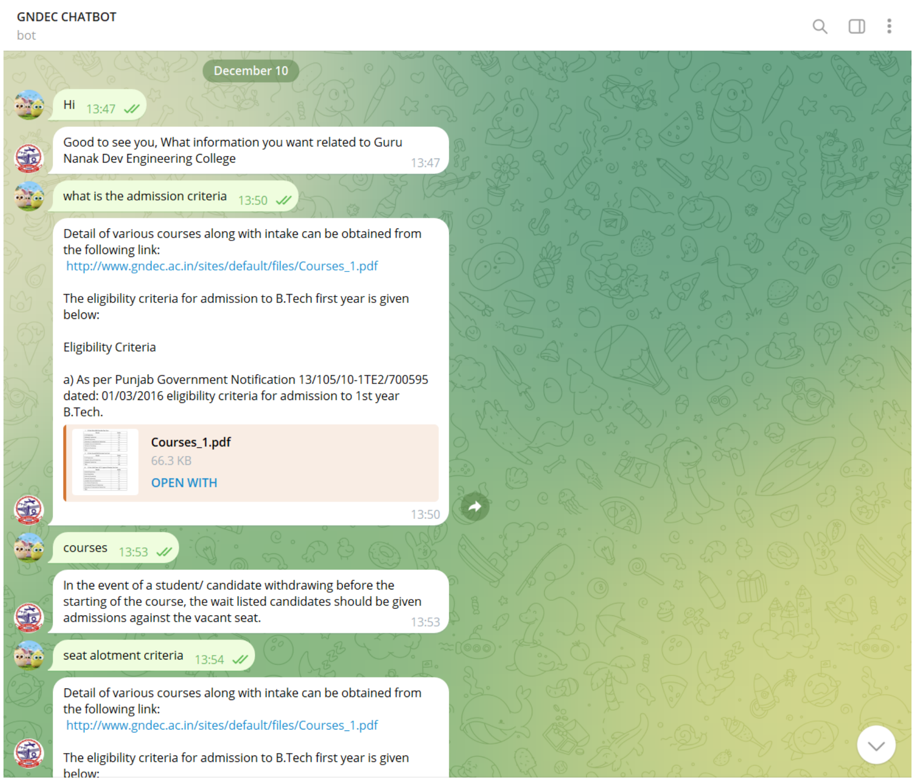
(h) **Error Handling** The code lacks comprehensive error handling. Consider implementing error handling mechanisms to handle potential issues, such as network errors or unexpected API responses.
4.8



```
while True:
    print("...")
    updates = tbot.get_updates(offset=update_id)
    updates = updates.get('result', [])  # If 'result' key is not present, set updates to an empty list
    print(updates)
    if updates:
        # Your code to process updates goes here
        pass

    print(updates)
    if updates:
        for item in updates:
            update_id = item["update_id"]
            print(update_id)
            try:
                message = item["message"]["text"]
                print(message)
            except:
                message = None
            from_ = item["message"]["from"]["id"]
            print(from_)

            reply = make_reply(message)
            tbot.send_message(reply, from_)
```

Figure 4.8: Handling Error

(i) **User Interface and Experience** Consider adding more user-friendly features, such as buttons or menus, to improve the user interface and experience within the Telegram chat.
4.9


GNDEC CHATBOT
bot

December 10

Hi 13:47

Good to see you, What information you want related to Guru Nanak Dev Engineering College 13:47

what is the admission criteria 13:50

Detail of various courses along with intake can be obtained from the following link:
http://www.gndec.ac.in/sites/default/files/Courses_1.pdf

The eligibility criteria for admission to B.Tech first year is given below:

Eligibility Criteria

a) As per Punjab Government Notification 13/105/10-1TE2/700595 dated: 01/03/2016 eligibility criteria for admission to 1st year B.Tech.

Courses_1.pdf
66.3 KB
OPEN WITH
13:50

courses 13:53

In the event of a student/ candidate withdrawing before the starting of the course, the wait listed candidates should be given admissions against the vacant seat. 13:53

seat alotment criteria 13:54

Detail of various courses along with intake can be obtained from the following link:
http://www.gndec.ac.in/sites/default/files/Courses_1.pdf

The eligibility criteria for admission to B.Tech first year is given below:

Figure 4.9: User Interface of Telegram Chatbot

CHAPTER 5
CONCLUSIONS AND FUTURE SCOPES

5.1 CONCLUSION

In conclusion, the college enquiry chatbot project has not only met but exceeded expectations in terms of its functionality and impact. We express our gratitude to [acknowledge contributors, team members, and stakeholders] whose dedication and collaboration have been instrumental in the success of this endeavor.

As we reflect on the achievements of this project, we look forward to a future marked by continued innovation, adaptability, and responsiveness to the evolving needs of the college community. The college enquiry chatbot stands as a testament to our commitment to leveraging technology for the betterment of our academic community..

5.2 FUTURE SCOPE

The future scope of the college enquiry chatbot project is expansive, offering opportunities for enhancement and adaptation to better serve the evolving needs of the college community. One crucial avenue for development involves the expansion of the chatbot's knowledge base. By incorporating a broader range of topics, including courses, events, and administrative processes, the chatbot can become an even more comprehensive resource for students and staff. Regular updates will be essential to ensure that the information remains current. Additionally, advancements in Natural Language Processing (NLP) present an exciting prospect. Investing in more sophisticated NLP algorithms will enhance the chatbot's ability to understand complex queries, colloquial language, and context-specific inputs. Multilingual support can also be explored to cater to a more diverse user base.

Another key area for future development lies in deeper integration with college systems. This involves exploring connections with the college's Enterprise Resource Planning (ERP) system to provide real-time information on academic schedules, grades, and other personalized data. Collaboration with existing communication platforms, such as email or messaging apps, would further enhance accessibility and streamline communication.

CHAPTER 6
REFERENCES

(a) Risheth.[Introduction to chatbot].https://www.myklassroom.com/blog/what-is-the-chatbot (Accessed 10 December 2023);

(b) Refsnes Data AS.[Introduction to python].https://www.geeksforgeeks.org/python (Accessed 10 December 2023);

(c) Sonoo Jaiswal.[Introduction to NLP].https://www.javatpoint.com/nlp(Accessed 10 December 2023);

(d) Refsnes Data AS.[Introduction of numpy].https://www.w3schools.com/numpy (Accessed 10 December 2023);

(e) Sonoo Jaiswal.[Bot Father].https://www.w3schools.com(Accessed 10 December 2023);