### Name**: Bhavya Bhaskar** Registration Number: **22BCE7958** Project Title: **Chatbot for Simple Question**

# **Project Report and Documentation**

## **1. Project Title**

**Chatbot for Simple Question using Rule-Based Logic**

## **2. Introduction**

This project aims to create a simple chatbot that can answer predefined questions using rule-based logic in Python. The chatbot is designed to interact with users via a command-line interface, responding to basic questions from a fixed set and allowing users to add new question-answer pairs dynamically.

## **3. Problem Statement**

Many users seek quick answers to common questions but may not have access to immediate human assistance. A simple chatbot that provides automated responses to frequently asked questions can be helpful in such scenarios, especially for educational or customer support contexts.

## **4. Objectives**

* To develop a Python-based chatbot that understands and responds to 100 predefined simple questions.
* To allow users to expand the chatbot's knowledge dynamically by adding new questions and answers during interaction.
* To create an easy-to-use command-line interface for user interaction.

## **5. Tools and Technologies Used**

* Python 3.x
* Standard Python libraries (no external dependencies)
* Command-line interface (CLI)

## **6. System Design and Flow**

### **Flow Description:**

1. The user starts the chatbot program.
2. The chatbot greets the user and waits for input.
3. The user types a question or command.
4. The chatbot converts input to lowercase and looks for a matching question in its dictionary.
5. If found, it responds with the stored answer.
6. If the user types add, the chatbot prompts to enter a new question and answer pair, which is added to the dictionary.
7. If the user types exit, the program terminates.
8. For unknown questions, it replies that it does not understand.

### **Flowchart:**

*(Include a simple flowchart image showing this sequence if possible)*

## **7. Algorithm**

plaintext

CopyEdit

Start  
Initialize dictionary with predefined question-answer pairs  
Display welcome message and instructions  
  
Loop until user types 'exit':  
 Take input from user and convert to lowercase  
 If input is 'exit':  
 Terminate the program  
 Else if input is 'add':  
 Prompt user to enter new question and answer  
 Add them to the dictionary  
 Confirm addition to user  
 Else:  
 Search for input in dictionary  
 If found:  
 Display corresponding answer  
 Else:  
 Display default message (not understood)  
End Loop  
End

## **9. Sample Output**

plaintext

CopyEdit

🤖 Hello! I'm a simple chatbot. Ask me a question (type 'exit' to quit, 'add' to add new Q&A).  
You: what is your name?  
Bot: I am a Simple Chatbot.  
You: who created you?  
Bot: I was created by a student using Python.  
You: add  
  
📝 Add a new question and answer  
Enter your new question: what is python?  
Enter the answer for this question: Python is a popular programming language.  
✅ New question-answer pair added successfully!  
  
You: what is python?  
Bot: Python is a popular programming language.  
You: exit  
🤖 Goodbye! Have a great day!

## **10. Prototype Results**

* The chatbot correctly answers all predefined questions.
* The chatbot successfully adds new questions and answers during runtime.
* Unknown questions are handled gracefully with an appropriate message.
* The interaction is smooth and intuitive via the command line.

## **11. Conclusion**

The project successfully demonstrates the development of a simple rule-based chatbot in Python that can answer a set of predefined questions and learn new ones dynamically. While this chatbot lacks advanced NLP capabilities, it serves as a solid foundation for understanding basic chatbot mechanics and can be expanded with more complex features like machine learning models or external APIs in the future.

## **12. Future Enhancements**

* Save new questions and answers to a file or database for persistence.
* Integrate natural language processing libraries (like NLTK or spaCy) for better understanding.
* Develop a graphical user interface (GUI) or web interface for improved usability.
* Add voice input and output capabilities.