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

### **Abstract: Chatbot for Simple Question**

In recent years, the rise of conversational AI has transformed how users interact with digital systems, paving the way for accessible, user-friendly interfaces. This project, **“Chatbot for Simple Question”**, focuses on designing a rule-based chatbot system using Python that can respond accurately to a predefined set of simple, frequently asked questions. Unlike AI models that require vast datasets and complex training procedures, this chatbot uses a deterministic logic-based approach to recognize user queries and deliver predefined responses, making it lightweight, fast, and easy to deploy for beginner applications.

#### **Problem Statement & Overview**

Many users seek instant answers to basic queries — greetings, factual questions, or casual interaction — without the need for internet-based searches or AI-intensive solutions. The problem addressed in this project is the lack of a simple, offline, and reliable system that can provide responses to such predefined questions efficiently. The chatbot provides a solution by recognizing the input text and matching it against stored patterns using string matching and conditional branching techniques.

#### **Tools and Applications Used**

* **Programming Language:** Python
* **Development Environment:** Jupyter Notebook / Visual Studio Code
* **Libraries Used:** random for dynamic responses, re for optional pattern matching
* **Optional Tools:** JSON for storing intents and responses, Flowcharting tools (draw.io, Lucid chart)

#### **Submodules of the Project**

1. **User Interface Module:**  
    Enables users to input their queries via the console. This module is responsible for capturing user input and displaying the chatbot's response.
2. **Text Preprocessing Module:**  
    Performs basic preprocessing such as converting input to lowercase and removing unnecessary characters to improve match accuracy.
3. **Rule-Based Logic Engine:**  
    Core logic that matches the processed input with a predefined dictionary or list of phrases and returns an appropriate response. Uses conditional statements and optional keyword searches.
4. **Response Generation Module:**  
    Once a match is found, the corresponding response is either selected randomly (to add variety) or displayed directly from a predefined set.
5. **Loop and Exit Control:**  
    Allows the chatbot to run continuously until the user types a termination keyword such as "exit" or "bye".

#### **Design and Flow of the Project**

The overall design follows a **modular and linear flow**, ensuring simplicity and clarity in execution. The block diagram consists of the following stages:

* **Input Layer:** Accepts user input via text.
* **Preprocessing Stage:** Cleans and prepares text for matching.
* **Matching Engine:** Compares input against known questions.
* **Response Generator:** Selects the appropriate reply.
* **Output Layer:** Displays the reply to the user.

This flow is visually represented using a flowchart and block diagram, which helps explain the decision-making process and the flow of control through different modules.

#### **Conclusion / Expected Output**

The expected output is a functioning rule-based chatbot capable of handling basic questions like greetings ("Hi", "Hello"), identity ("Who are you?"), time-related queries, and farewells ("Bye", "See you"). It provides deterministic responses and serves as an introductory application to chatbot development. This project not only demonstrates the use of Python for building interactive applications but also introduces students to fundamental AI concepts such as intent recognition, pattern matching, and modular design. In conclusion, this chatbot is an ideal educational prototype that can be expanded into more intelligent systems using NLP or machine learning in future versions.