

```

import difflib
from googletrans import Translator
import json

class Chatbot:
    def __init__(self, knowledge_base_file="knowledge_base1.json"):
        self.knowledge_base_file = knowledge_base_file
        self.load_knowledge_base()
        initial_knowledge = {
            "What is your name?": "I am a chatbot.",
            "How are you?": "I'm just a program, so I don't have feelings, but thanks for asking!",
        }

        self.knowledge_base.update(initial_knowledge)
        self.save_knowledge_base()

    def load_knowledge_base(self):
        try:
            with open(self.knowledge_base_file, 'r') as file:
                self.knowledge_base = json.load(file)
        except FileNotFoundError:
            self.knowledge_base = {}

    def save_knowledge_base(self):
        with open(self.knowledge_base_file, 'w') as file:
            json.dump(self.knowledge_base, file, indent=4)

    def get_response(self, user_input):
        if "translation" in user_input.lower() or "translate" in user_input.lower():
            return self.perform_translation()

        if user_input in self.knowledge_base:
            return self.knowledge_base[user_input]
        else:
            matched_question = self.find_similar_question(user_input)
            if matched_question:
                response = self.knowledge_base[matched_question]
                self.knowledge_base[user_input] = response
                self.save_knowledge_base()
                return response

            response = input("I'm not sure about that. Please provide an answer: ")
            self.knowledge_base[user_input] = response
            self.save_knowledge_base()
            return response

    def find_similar_question(self, user_input):
        potential_matches = difflib.get_close_matches(user_input, self.knowledge_base.keys(), n=1, cutoff=0.6)
        if potential_matches:
            return potential_matches[0]
        else:
            return None

    def perform_translation(self):
        print("Language Translation Program")

        while True:
            user_input = input("Enter text to translate: ")

            if user_input.lower() == 'exit':
                print("Exiting the translation program.")
                return "Exiting the translation program."

            target_language = input("Enter target language (e.g., 'es' for Spanish): ")

            try:
                translator = Translator()
                translation = translator.translate(user_input, dest=target_language)
                return f"Translated text: {translation.text}"

            except Exception as e:
                return f"Error during translation: {e}"

chatbot = Chatbot()

while True:
    user_input = input("You: ")
    if user_input.lower() == 'exit' or user_input.lower() == 'quit':
        chatbot.save_knowledge_base()
        break
    response = chatbot.get_response(user_input)
    print("Chatbot:", response)

```