MINI PROJECT – CHAT BOT

Aim:

To create a chatbot.

Procedure:

1. I have used Python for this self learning chatbot.
2. First we need to create a JSON file where all our question and answer will be store.
3. Now create a Python file for our chatbot.
4. Install and import necessary libraries like json and get_close_matches from difflib.
5. Now we have to create functions for loading Knowledge base, save knowledge base, find best match , to get answer for question.
6. Now for the main Chat bot function , we have to call the the functions in order
7. Run the program in terminal.
8. Now you type something first.
9. The bot says “I don't know, can you teach me?”, now you can teach what to reply when the question is raised like that or you can type “Leave” and leave that question.
10. Now when you question it , the bot can recall from its previous answer and present it to you.

Features:

1. Self learn: As each and every question can be learned , we can teach the bot a lot of things to remember.
2. Memory: Even after the session has ended , the memory is not wiped as it is stored inside the JSON file.

Code:

```
import json

from difflib import get_close_matches


def load_knowledge_base(file_path: str) -> dict:
    """Loads knowledge base data from a JSON file."""
    with open('knowledge_base.json', 'r', encoding="utf-8") as file:
        data: dict = json.load(file)
    return data
```

```
def save_knowledge_base(file_path: str, data: dict):
    """Saves knowledge base data to a JSON file."""
    with open('knowledge_base.json', 'w', encoding='utf-8') as file:
        json.dump(data, file, indent=2)



def find_best_match(user_question: str, questions: list[str]) -> str | None:
    """Finds the closest match to a user question from a list of questions."""
    matches: list = get_close_matches(user_question, questions, n=1, cutoff=0.6)
    return matches[0] if matches else None



def get_answer_for_question(question: str, knowledge_base: dict) -> str | None:
    """Retrieves the answer for a specific question from the knowledge base."""
    for q in knowledge_base["questions"]:
        if q["question"] == question:
            return q["answer"]
    return None



def chat_bot():
    """Main function for the chat bot."""
    # Load knowledge base
    knowledge_base = load_knowledge_base('knowledge_base.json')
```

```
    while True:
        user_input: str = input('You: ')

        if user_input.lower() == 'quit':
            break
        # import pdb; pdb.set_trace()
        # Find best match
        best_match = find_best_match(user_input, [q["question"] for q in
knowledge_base["questions"]])

        if best_match:
            answer = get_answer_for_question(best_match, knowledge_base)
            if answer:
                print(answer)
        else:
            print("Bot: I don't know, can you teach me? ")
            new_answer: str = input("Type the answer or 'leave' to leave: ")

            if new_answer.lower() != 'leave':
                knowledge_base["questions"].append({'question': user_input, 'answer': new_answer})
                save_knowledge_base('knowledge_base.json', knowledge_base)
                print("Bot: Thanks man! I have learned something new!")

if __name___== '__main__':
    chat_bot()
```

Ouput:

```
You: hi
hello
You: Whats the capital of India?
Bot: I don't know, can you teach me?
Type the answer or 'leave' to leave: Delhi
Bot: Thanks man! I have learned something new!
You: Whats the capital
Delhi
You:
```

Result:

The creation of chatbot has been done successfully.