PROJECT REPORT ON Self Learning Bot

REVIEW - I

Student Details

Roll No	Name
20201CAI0104	Maheswar Reddy
20201CAI0131	Karthik Sharma
20201CAI0152	Arun Teja
20201CAI0096	Salapakshi Ganesh

Under the Supervision of :

Prof . Dr.Asif Mohammed H.B

November, 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

BENGALURU

Abstract:

This report outlines the current voice-activated AI assistant program. It suggests an evolution towards a more comprehensive training model that integrates technical and non-technical modules for a more well-rounded training experience.

Introduction:

The present AI assistant, "Friday," is designed as a voice-activated interface to cater to various commands and tasks. However, our training regimen should not solely emphasize technical aspects. To foster a more adaptable and multifaceted workforce, the proposal is to extend the training program to include both technical and non-technical modules.

Overview of Current AI Assistant Program:

The existing AI assistant, "Friday," is a functional voice-activated interface that responds to voice commands. It encompasses functions like time reporting, web browsing, YouTube music playback, and WhatsApp messaging.

CODE:

```
import speech_recognition as sr
import pyttsx3
import datetime
import webbrowser
import pywhatkit as kit

# Function for converting text to speech
def speak_text(text):
    engine = pyttsx3.init()
    voices = engine.getProperty("voices")
    engine.setProperty("rate", 150)
    engine.setProperty("voice", voices[1].id)
    engine.say(text)
    engine.runAndWait()

def start_friday():
    current_time = datetime.datetime.now().strftime("%H:%M")
    greeting = f"Hello sir. This is Friday. The current time is {current_time}. How can I help you?"
    speak_text(greeting)
```

```
# YouTube
def play_music(command):
    try:
        kit.playonyt(command)
    except Exception as e:
        speak_text(f"An error occurred while playing the music: {str(e)}")

# WhatsApp message
def whatsapp_message(number, message, time):
    try:
        kit.sendwhatmsg(f"+{number}", message, time.hour, time.minute)
        speak_text("Sending the message")
    except Exception as e:
        speak_text(f"Error sending the message: {str(e)}")

# WhatsApp group message
def group_message(group_name, message, time_obj):
    try:
        kit.sendwhatmsg_to_group(f"+{group_name}", message, time_obj.hour, time_obj.minute)
        speak_text("Sending the message")
    except Exception as e:
```

```
        speak_text(f"Error sending the message: {str(e)}")


# Initialize the recognizer
recognizer = sr.Recognizer()
# starting
start_friday()
iteration_count = 0
while True:
    print("Listening...")
    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source, duration=0)
        audio = recognizer.listen(source)
        try:
            # Recognize the speech using Google's speech recognition
            command = recognizer.recognize_google(audio).lower()
            print('You said:', command)

            if "Friday" in command:
                speak_text("Yes sir! I am listening")

            elif "open" in command:
```

```
            website = command.replace("open", "").strip()
            speak_text(f"Opening {website}...")
            webbrowser.open(f"https://www.{website}.com")

        elif "play" in command:
            # Extract the song name from the command
            song_name = command.replace("play", "").strip()
            speak_text(f"Playing {song_name} on YouTube...")
            play_music(song_name)

        elif "send message" in command:
            speak_text("Enter the number with country code:")
            number = input("Enter the number with country
code: ")
            speak_text("Enter the time to send (24-hour format
hh:mm):")
            time_input = input("Enter the time in 24-hour
format (hh:mm): ")
            time_obj = datetime.datetime.strptime(time_input,
'%H:%M')
            speak_text("Enter the message:")
            message = input("Enter the message:")
            whatsapp_message(number, message, time_obj)
```

```
        elif "send a group message" in command:
            speak_text("Enter the group name:")
            group_name = input("Enter the group name: ")
            speak_text("Enter the time to send (24-hour format hh:mm):")
            time_input = input("Enter the time in 24-hour format (hh:mm): ")
            time_obj = datetime.datetime.strptime(time_input, '%H:%M')
            speak_text("Enter the message:")
            message = input("Enter the message:")
            group_message(group_name, message, time_obj)

        elif "quit" in command or "bye" in command:
            speak_text("Goodbye!")
            break

    except sr.UnknownValueError:
        print("I am listening, sir...")
    except sr.RequestError as e:
        print(f"Error with the speech recognition service: {e}")

  iteration_count += 1
```

```
if iteration_count >= 50:
    iteration_count = 0
    start_friday()
```

Proposal for Future Training Enhancement:

1. Integration of Non-Technical Modules:

- Introduction of non-technical modules focusing on soft skills, communication, and problem-solving.
- Content includes effective communication, conflict resolution, time management, and leadership skills.
- Emphasis on fostering interpersonal skills to bolster team dynamics.

2. Enrichment of Technical Modules:

- Expansion of technical modules to encompass a wider array of topics within our industry.
- Inclusion of advanced subjects and emerging technologies pertinent to our field.
- Incorporation of practical applications and case studies to bridge theory with real-world scenarios.

3. Benefits of the Enriched Training Approach:

- Holistic skill development in technical and non-technical areas.
- Empowerment of employees with a broader skill set, facilitating adaptability.
- Enhanced team collaboration and problem-solving capabilities due to a well-rounded skill spectrum.

Future Scope:

- The integration of both technical and non-technical modules within the training program represents an ongoing journey. Continuous assessment, refinement, and adaptation of training modules will be crucial to keep pace with industry advancements and workforce demands.

- This report serves as a blueprint for the strategic integration of technical and non-technical training modules into the existing program, emphasizing the importance of a well-rounded skill set for the workforce.

- This report delineates the vision of integrating technical and non-technical modules into the existing training program. The objective is to equip the workforce with a versatile skill set to navigate diverse challenges in the ever-evolving work landscape.

References:

- https://betterprogramming.pub/an-introduction-to-pyttsx3-a-text-to-speech-converter-for-python-4a7e1ce825c3
- https://www.geeksforgeeks.org/python-convert-speech-to-text-and-text-to-speech/
- https://pypi.org/project/pywhatkit/
- https://github.com/Ankit404butfound/PyWhatKit/wiki

- https://docs.python.org/3/library/webbrowser.html

Conclusion:

The proposed expansion of the training program to encompass both technical and non-technical modules underscores our commitment to fostering a diverse and adaptable workforce. This forward-thinking approach aims to equip our team members with a comprehensive skill set, ensuring their preparedness to face evolving challenges in the workplace.