SMART ASSISTANT

- AI-Powered Smart Assistant Using Python
- Developed by [Rishav Kumar Patel]
- [21-07-2024]

INTRODUCTION

- Brief overview of the smart assistant project.
- Purpose and objectives of creating the smart assistant.

TECHNOLOGIES AND LIBRARIES USED

- Overview of the technologies and libraries used in the project.
- List of libraries: subprocess, wolframalpha, pyttsx3, tkinter, json, random, operator, speech_recognition, datetime, wikipedia, webbrowser, os, smtplib, ctypes, time, requests, shutil, clint, ecapture, BeautifulSoup, win32com.client, urllib.request

TEXT-TO-SPEECH INITIALIZATION

- import pyttsx3
- engine = pyttsx3.init('sapi5')
- voices = engine.getProperty('voices')
- engine.setProperty('voice', voices[1].id)

- def speak(audio):
-     engine.say(audio)
-     engine.runAndWait()
- Initialized pyttsx3 for text-to-speech functionality.
- Set up the voice properties.
- Defined the speak function to convert text to speech.

GREETING FUNCTIONALITY

```
def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour < 12:
        speak('Good Morning Sir!')
    elif hour >= 12 and hour < 18:
        speak('Good Afternoon Sir!')
    else:
        speak('Good Evening Sir!')

    assname = 'Missy 2 point o'
    speak('I am your Assistant')
    speak(assname)
```

- Function to greet the user based on the time of day.
- Personalized assistant name announcement.

USER IDENTIFICATION

- def username():
-     speak('What should I call you, sir?')
-     uname = takeCommand()
-     speak('Welcome Sir')
-     speak(uname)
-     columns = shutil.get_terminal_size().columns

-     print('#'.center(columns))
-     print('Welcome Sir.', uname.center(columns))
-     print('#'.center(columns))

-     speak('How can I help you, Sir')
- Function to ask for and recognize the user's name.
- Greets the user by name and prompts for commands.

VOICE COMMAND RECOGNITION

```
def takeCommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print('Listening...')
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print('Recognizing...')
        query = r.recognize_google(audio, language='en-in')
        print(f'User said: {query}\n')
    except Exception as e:
        print(e)
        print('Unable to recognize your voice.')
        return 'None'
    return query
```

- Function to capture voice commands using the microphone.
- Utilizes Google's speech recognition API for converting speech to text.

MAIN FUNCTION

```
if __name__ == '__main__':
    clear = lambda: os.system('cls')
    clear()
    wishMe()
    username()

    while True:
        query = takeCommand().lower()
        # Further command handling...
```

- Main function that initializes the assistant, greets the user, and enters a loop to process commands.

WIKIPEDIA SEARCH

- if 'wikipedia' in query:
-     speak('Searching Wikipedia...')
-     query = query.replace('wikipedia', '')
-     results = wikipedia.summary(query, sentences=3)
-     speak('According to Wikipedia')
-     print(results)
-     speak(results)
- Processes queries related to Wikipedia searches.
- Provides a brief summary of the requested topic.

WEB BROWSER COMMANDS

- elif 'open google' in query:
-     speak('Here you go to Google\n')
-     webbrowser.open('google.com')
- Opens Google in a web browser upon user command.

TIME ANNOUNCEMENT

- elif 'the time' in query:
- strTime = datetime.datetime.now().strftime('%H:%M:%S')
- speak(f'Sir, the time is {strTime}')
- Announces the current time upon user request.

BASIC INTERACTION COMMANDS

- elif 'how are you' in query:
-     speak('I am fine, thank you')
-     speak('How are you, Sir')

- elif 'fine' in query or 'good' in query:
-     speak('It's good to know that you're fine')

- elif 'exit' in query:
-     speak('Thanks for giving me your time')
-     exit()
- Handles basic interaction commands such as asking about well-being and exiting the program.

CREATOR INFORMATION

- elif 'who made you' in query or 'who created you' in query:
- speak('I have been created by Rishav.')
- Responds to queries about the creator of the assistant.

CALCULATOR FEATURE

- elif 'calculate' in query:
-     app_id = 'Wolframalpha API ID'
-     client = wolframalpha.Client(app_id)
-     indx = query.lower().split().index('calculate')
-     query = query.split()[indx + 1:]
-     res = client.query(' '.join(query))
-     answer = next(res.results).text
-     print('The answer is ' + answer)
-     speak('The answer is ' + answer)
- Utilizes WolframAlpha API for performing calculations based on user commands.

WEB SEARCH AND PLAYBACK

- elif 'search' in query or 'play' in query:
-     query = query.replace('search', '')
-     query = query.replace('play', '')
-     webbrowser.open(query)
- Allows the user to search or play content directly from the web.

DEVICE CONTROL COMMANDS

- elif 'shutdown system' in query:
-     speak('Hold On a Sec! Your system is on its way to shut down')
-     subprocess.call('shutdown /p /f')

- elif 'lock window' in query:
-     speak('locking the device')
-     ctypes.windll.user32.LockWorkStation()
- Provides commands to control the system, such as shutting down or locking the computer.

TAKING NOTES

```
•  elif 'write a note' in query:
•      speak('What should I write, sir')
•      note = takeCommand()
•      file = open('Missy.txt', 'w')
•      speak('Sir, should I include date and time?')
•      snfm = takeCommand()
•      if 'yes' in snfm or 'sure' in snfm:
•          strTime = datetime.datetime.now().strftime('%H:%M:%S')
•          file.write(strTime)
•          file.write(' :- ')
•          file.write(note)
•      else:
•          file.write(note)
```

- Allows the user to take and save notes.

ADDITIONAL FEATURES

- Explanation of other features such as changing the background, taking a photo, and more.
- Highlight the flexibility and extensibility of the assistant.

CONCLUSION

- Recap of the smart assistant’s capabilities.
- Future improvements and potential enhancements.
- Invitation for questions and feedback.

Q&A

- Open floor for questions and discussion.

THANK YOU