

Task 1 : Project Report

Title:

Voice Assistant Using Python

Objective

The objective of this project is to develop a voice assistant capable of interacting with the user via spoken commands to perform multiple tasks including:

- Greeting the user.
- Announcing the current time and date.
- Conducting web searches.
- Providing real-time weather updates.
- Reading the latest news headlines.
- Setting reminders.
- Playing music on YouTube.

The project demonstrates how to integrate speech recognition, text-to-speech, and multiple APIs into a single Python application.

Tools and Technologies Used

- **Python 3**
- **speech_recognition** – Convert speech to text.
- **pyttsx3** – Convert text to speech.

- **requests** – Make HTTP API calls.
 - **pywhatkit** – Automate YouTube playback.
 - **WeatherAPI** – Retrieve weather data.
 - **NewsAPI** – Fetch the latest news headlines.
-

Implementation Steps

1. Environment Setup

- Installed Python and all required libraries.
- Created API keys for weather and news services.

2. Speech Recognition

- Used `speech_recognition` to listen to the microphone.
- Converted spoken input into text commands.

3. Text-to-Speech

- Utilized `pyttsx3` to provide audible responses to the user.

4. Time and Date Retrieval

- Used the `datetime` module to announce current date and time.

5. Web Search

- Opened browser tabs to search Google for queries.

6. Weather Information

- Queried WeatherAPI to fetch real-time weather data.
- Parsed JSON responses and spoke the results.

7. News Updates

- Queried NewsAPI to retrieve the latest news headlines.
- Read out top headlines from the response.

8. Reminders

- Implemented a countdown timer to trigger reminders after a set interval.
- Added threading to avoid freezing the application while waiting.

9. Music Playback

- Played requested songs via YouTube.

10. Main Loop

- Continuously listened for commands and performed actions accordingly.

Features

Greetings

Responds to greetings such as “Hello” or “Hi.”

Time and Date

Announces the current time and date.

Web Search

Performs Google searches for user queries.

Weather Updates

Provides current weather conditions for any specified city.

News Headlines

Fetches and reads the latest top headlines from BBC News using NewsAPI.

Reminders

Sets reminders with countdown timers.

Music Playback

Plays requested songs on YouTube.

Outcome

The assistant was successfully implemented and is capable of:

- Understanding spoken commands in real-time.
- Providing accurate and dynamic information (weather and news).
- Automating common tasks like reminders and web searches.
- Offering a foundation for more complex assistants in the future.

Challenges Faced

- Ensuring accurate recognition of speech in noisy environments.
- Handling API authentication errors (e.g., invalid keys).
- Avoiding application freeze while waiting for reminders.
- Parsing and formatting JSON responses from APIs.
- Standardizing numeric input for reminder durations.

Future Enhancements

- Support for natural language processing for more flexible commands.
- Integration with email or messaging platforms.
- Support for smart home devices.
- Persistent storage of reminders and preferences.

- Multi-language support.
-

Conclusion

This project demonstrates how Python libraries and APIs can be combined to create a functional, interactive voice assistant. It provided valuable experience in speech recognition, text-to-speech synthesis, and integrating external data sources like WeatherAPI and NewsAPI.

Contact Information

Prepared by:

UDITTA SHARMA

Contact Email: udittasharma7@gmail.com

Linkedin :

https://www.linkedin.com/in/uditta-sharma-38b617279?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=android_app