Abstract

This project develops a voice-controlled assistant using Python that listens to user commands via the microphone and executes predefined tasks. It leverages the speech_recognition library to convert spoken commands into text, and pyttsx3 for providing audible responses. The assistant can perform actions such as opening applications, launching websites, reporting the current time, and delivering information about the project itself. The program continuously listens for commands, responding verbally to recognized inputs, and terminates only when the user explicitly requests to exit. This simple yet functional voice assistant demonstrates the integration of speech interfaces to facilitate hands-free interaction with a computer.

Introduction

Voice assistants have transformed user interaction, enabling hands-free and intuitive control of digital devices. The goal of this project is to build a personal voice-controlled assistant using Python that can execute voice commands to perform tasks such as opening applications, retrieving information, and browsing the web. This assistant enhances accessibility, productivity, and user convenience by allowing natural language interaction without needing keyboard input.

Tools Used

- Python: Programming language for development
- speech_recognition: Converts speech to text by capturing microphone input
- pyttsx3: Converts text to speech for audio feedback
- os: To execute system commands (e.g., open Notepad)
- webbrowser: To open webpages like Google or LinkedIn

Steps Involved in Building the Project :

1. Initialize the speech recognizer and text-to-speech engine.
2. Capture voice input through the microphone and convert it to text using Google's Speech Recognition API.
3. Process recognized text to identify commands like opening applications, telling time, or browsing websites.

4. Generate spoken responses using pyttsx3 for interactive feedback.
5. Implement a continuous listening loop that runs until the user says an exit command.
6. Handle exceptions and ambient noise to ensure robustness and accuracy in speech recognition.

Conclusion

The Voice-Controlled Assistant provides a practical example of integrating speech recognition and synthesis for human-computer interaction. It supports essential commands that automate common tasks and responds audibly, making user engagement seamless. This project lays the groundwork for more complex assistants that can be tailored to specific user needs, such as personalized reminders or external API integration. Overall, it highlights the feasibility and utility of voice interfaces driven by Python libraries in modern computing environments.