DESCRIPTION ABOUT PROJECT
DESKTOP ASSISTANT USING CHATGPT

Overview:

- This project features a sophisticated voice-controlled assistant named "Jarvis."
- It combines speech recognition, AI, and voice synthesis to enhance user interaction and automate tasks.

Key Features

Voice Recognition:

- Utilizes the `speech_recognition` library to capture and process spoken commands.
- Converts spoken words into text using Google's speech recognition API.

AI-Powered Responses:

- Integrates with OpenAI's GPT-3.5 model to generate responses based on user prompts.
- Handles a variety of queries, providing contextually relevant answers.

Voice Synthesis:

- Uses `win32com.client`'s SAPI.SpVoice component to convert AI-generated text into spoken responses.

Task Automation:

- Executes tasks like opening websites, playing music, checking the time, and opening the camera based on voice commands.
- Automates actions for improved convenience and user experience.

File Management:

- Saves responses from the AI to text files in an "Openai" directory.
- Organizes files with dynamic naming based on user queries.

How It Works:

Initialization:

- Greets the user and sets up the voice synthesis for interactive responses.

Listening for Commands:

- Continuously listens for commands through the microphone.
- Processes and converts speech into text for further action.

Executing Commands:

- Opens specified websites or applications based on user requests.
- Provides spoken time updates or plays music when requested.
- Saves and manages files containing AI-generated responses.

AI Interaction:

- Sends user prompts to OpenAI's GPT-3.5 model for intelligent responses.
- Reads the model's responses aloud and saves them to text files.

Chat Functionality:

- Maintains conversational context with the user using a global chat history.
- Provides relevant and context-aware responses during interactions.

Technologies Used:

- **Speech Recognition:** Leveraging `speech_recognition` library for speech-to-text conversion.
- **Web Interaction:** Using `webbrowser` module to open websites.
- **Voice Synthesis:** Employing `win32com.client` for text-to-speech functionality.
- **AI Integration:** Connecting with OpenAI's GPT-3.5 for generating conversational responses.
- **File Management:** Utilizing Python's file handling for saving text responses.

Example Usage:

Command: "Open Google"

- **Action:** Opens Google in the default web browser.

Command: "What is the time?"

- **Action:** Provides the current time in spoken words.

Command: "Using Artificial Intelligence"

- **Action:** Sends the query to GPT-3.5 and reads the response aloud.

This project highlights the practical use of AI and speech technologies to create an interactive, intelligent assistant that enhances user convenience and functionality.