Aistie 1.0.0

Module Progress Report

Capabilities:

1. **Speech Recognition:** The code can recognize speech and convert it into text using the speech_recognition library.
2. **Text-to-Speech:** The code can convert text into speech using the pyttsx3 library.
3. **Web Browsing:** The code can open websites and search for things on the internet using the webbrowser library.
4. **Reminders:** The code can set reminders and store them in a text file.
   For now not working cause of some API Issues in the weather api connection
5. **Gesture Control:** The code can use gesture control to draw or control the mouse using the gesture and hand_control_draw libraries.
6. **Wikipedia Search:** The code can search for information on Wikipedia using the wikipedia library.
7. **YouTube Search:** The code can search for videos on YouTube using the pywhatkit library.
8. **System Control:** The code can open and close applications, and perform other system-level tasks using the os library.
9. **Conversational AI:** The code can engage in basic conversations using the groq library and the LLaMA model.

APIs Used:

1. **OpenWeatherMap API:** Used to retrieve weather information.
   Currently some issues in weather API.

2. **Wikipedia API:** Used to search for information on Wikipedia.
   Wikipedia API referring to Wikipedia Python Library.

3. **YouTube API:** Used to search for videos on YouTube.
   Using pywhatkit for playing songs on youtube directly.

4. **Groq API:** Used to engage in conversational AI.
   The only paid API in the Aistie 1.0.0 Model…if we use any alternative method for weather system.

Working Mechanisms:

- **Speech Recognition:** The code uses the speech_recognition library to recognize speech and convert it into text.
- **Text Processing:** The code processes the recognized text to determine the user's intent.
- **Intent Identification:** The code identifies the user's intent based on the processed text.
- **Task Execution:** The code executes the corresponding task based on the identified intent.
- **Feedback Loop:** The code provides feedback to the user through speech or text output.

Main Components:

- **Modules and Variables:**

1. **Speech Recognition Module:**
- **sr (SpeechRecognition) library**: Handles speech recognition and converts speech into text.
- **pyttsx3 library:** Handles text-to-speech conversion and speaks the output.

2. **Intent Identification Module:**
- **execute() function:** Uses the groq library to identify the user's intent based on the processed text.

3. **Task Execution Module**:
- **webbrowser library:** Opens websites and searches for things on the internet.
- **pywhatkit library:** Plays songs directly on YouTube.
- **os library:** Opens and closes applications, and performs other system-level tasks.
  (os = operating system to perform tasks in the user device)
- **wikipedia library:** Searches for information on Wikipedia.

4. **Feedback Module:**
- **pyttsx3 library:** Provides feedback to the user through speech output.
- **tkinter library:** Provides feedback to the user through text output in the GUI.

5. **GUI Module:**
- **tkinter library:** Handles the graphical user interface.

Workflow Management:

1. **Main Loop:**
- The code runs in an infinite loop, waiting for user input.

2. **Speech Recognition:**
- The code recognizes speech and converts it into text using the sr library.

3. **Intent Identification:**
- The code identifies the user's intent based on the processed text using the execute() function and the groq library.

4. **Task Execution:**
- The code executes the corresponding task based on the identified intent using the relevant libraries (e.g. webbrowser, pywhatkit, os, etc.).

5. **Feedback:**
- The code provides feedback to the user through speech or text output using the pyttsx3 and tkinter libraries.

6. **Repeat:**
- The code repeats the process until the user exits the application.

Lists and Variables:

1. **reminds:** A dictionary that stores reminders.

2. **path:** A string that stores the path to the reminders text file.

3. **capasity:** A string that stores the capabilities of the AI assistant.

- **Need modifications in these two**

{

4. **ar:** A boolean variable that indicates whether the AI assistant is awake or not.

5. **is_awake:** A boolean variable that indicates whether the AI assistant is awake or not.

}