**Speech Recognition & Task Execution**

**Overview**

This program is designed to capture voice commands, convert them to text, and perform specific actions based on the recognized commands. The program uses Google's Text-to-Speech (gTTS) and Speech Recognition APIs, and it can interact with the user's system to perform tasks such as opening web pages, taking screenshots, and providing verbal responses.

**Features**

* **Voice Command Recognition**: Capture voice commands through the microphone.
* **Text-to-Speech Conversion**: Convert recognized text into speech and play it back to the user.
* **GUI Automation**: Perform actions such as taking screenshots and opening web pages based on voice commands.
* **Customizable Commands**: The program can be extended with additional commands as needed.

**Prerequisites**

Before running the program, ensure that the dependencies are installed as specified in the requirements.txt file.

**How to Use**

1. **Running the Program**:
   * Execute the program by running the Speech Recognition & Task Execution.py script.
2. **Voice Commands**:
   * **Greeting**: Say "Hello" to receive a greeting from the program.
   * **Ask for Name**: Ask "What is your name?" to learn the program's name ("Jarvis").
   * **Ask About Well-being**: Ask "How are you?" or similar phrases to receive a response about the program's state.
   * **Take a Screenshot**: Say "Take a screenshot" or similar phrases to capture a screenshot and save it as screenshot.png.
   * **Open YouTube**: Say "Open YouTube" to open the YouTube website in the default web browser.
   * **Read the News**: Say "Read the news" or similar phrases to open a news website in the default web browser.
   * **Goodbye**: Say "Goodbye" or similar phrases to end the program.
3. **Error Handling**:
   * If the program doesn't recognize your voice input, it will prompt you to try again. After three failed attempts, the program will exit automatically.
4. **Customization**:
   * You can add more voice commands and corresponding actions by modifying the process\_voice\_command() function. Simply add more elif conditions to handle additional commands.

**Troubleshooting**

* **Failed Speech Recognition**: If the program fails to recognize your voice, try speaking clearly and slowly and adjusting the microphone sensitivity.