AceInterview

Ace Interview is a comprehensive platform that utilizes artificial intelligence and speech recognition technologies to enhance the interview experience. It features real-time speech-to-text conversion, interactive chat with AI, text-to-speech capabilities, and file parsing functionalities, all wrapped in a user-friendly graphical interface. This documentation provides a complete guide on the application's features, installation steps, dependencies, and usage.

Table of Contents

---

1. Overview

Ace Interview is designed to simulate an interview environment by combining real-time speech recognition and AI-powered interactions. It provides users with an intuitive interface to practice interviews and receive feedback in real time, allowing for an engaging and educational experience. The platform is especially useful for interview preparation, HR training, or as an educational tool for enhancing communication skills.

2. Features

1. **Real-Time Speech Recognition**
   Utilizes the SpeechRecognition library to convert spoken language into text, allowing the user to speak naturally and see their words transcribed on the screen.

2. **Text-to-Speech**
   Provides audible responses to the user's input using the pyttsx3 library, enabling a more immersive and interactive experience.

3. **Interactive Chat**
   Incorporates Google's Generative AI model (google.generativeai) to simulate conversation, allowing users to ask interview-related questions or engage in meaningful dialogue with AI.

4. **File Parsing**
   Extracts and parses text from various document formats using the PyPdf library, enabling the platform to pull in information from resumes, cover letters, or interview guidelines.

5. **Graphical User Interface (GUI)**
   A clean and intuitive interface built with tkinter, designed to provide a smooth user experience with easy navigation between different functionalities.

6. **Image Handling**
   Leverages PIL (Pillow) for managing images, allowing users to display, resize, and manipulate images within the GUI.

3. Installation

Ace Interview can be installed locally from the source code or by downloading the executable file for a simplified installation process.

Option 1: Install from Source

Step 1: Clone the Repository

You can obtain the source code from GitHub by running the following command:

git clone https://github.com/SHAIK-ABDUL-KHADEER/AceInterview.git

cd ace-interview

Step 2: Create a Virtual Environment

It's recommended to create a virtual environment to isolate your project dependencies:

python -m venv venv

source venv/bin/activate # On Windows, use `venv\Scripts\activate`

Step 3: Install Dependencies

Use pip to install the required dependencies:

pip install -r requirements.txt

Ensure the following libraries are listed in your requirements.txt:

- smtplib
- email
- psycopg2
- opencv-python
- pyttsx3
- SpeechRecognition
- google-generativeai
- PyPdf
- tkinter
- pandas
- pillow

Step 4: Run the Application

Once the dependencies are installed, you can run the application using the following command:

python main.py

Option 2: Install Using Executable File

For users who prefer a simple installation process, you can download the executable file and run it without setting up a development environment.

1. Download the executable file from this DriveLink.
2. Once downloaded, run the .exe file.
3. Follow the on-screen instructions to install and launch the application.

---

4. Usage

Once the application is installed, you can begin using Ace Interview to simulate interviews and engage with AI.

Starting the Application

1. If installed from the source, run python main.py in the project directory.
2. If using the executable, simply launch the .exe file.

Once the application starts, you'll be presented with a user-friendly GUI. From here, you can begin exploring the various features.

Core Functionalities

1. **Start Interview**
    - Begins the interview simulation, where users can respond to questions using voice commands.

- Responses will be transcribed in real time and displayed in the GUI.

2. **Chat with AI**
    - Users can type questions or speak using voice commands to interact with the AI model.
    - The AI responds based on input, generating conversational and relevant interview-based replies.
3. **Voice Commands**
    - Utilizes the SpeechRecognition module to process voice input and translate it into text, making it easier for users to engage in natural speech.
4. **Text-to-Speech Responses**
    - The pyttsx3 library is used to convert text responses from the AI into audible speech, simulating a real conversation.
5. **File Parsing**
    - Load documents (e.g., resumes) to parse text and extract information from them using PyPdf.

---

5. Dependencies

Ace Interview relies on the following libraries:

1. **random**
    - Used to generate random values, often for simulating interview scenarios.
2. **psycopg2**
    - A PostgreSQL adapter for Python, allowing for the management of interview data using a PostgreSQL database.
3. **tkinter**
    - Python’s standard GUI toolkit, used for creating the graphical user interface.

4. **opencv-python (cv2)**
    - Used for image processing and handling image inputs during interviews (e.g., displaying candidate photos or visual aids).
5. **pyttsx3**
    - A text-to-speech conversion library, allowing the application to provide audible feedback and responses.
6. **SpeechRecognition**
    - For converting spoken language into text, used to capture the user's voice input.
7. **google-generativeai**
    - Interacts with Google's Generative AI model, which provides AI-based conversational capabilities.
8. **PyPdf**
    - A library for parsing PDF documents, enabling the extraction of text from resumes or other documents.
9. **pandas**
    - A powerful data manipulation tool, used for processing and managing interview data.
10. **Pillow (PIL)**
    - For handling and manipulating images in the GUI.

---

6. Contact

For any questions or technical issues, please reach out to the project maintainer:

Email: sak78620@gmail.com

You can also report issues or request additional features on the GitHub repository.