**AI-Powered Assistive Tool**

**Introduction**

The AI-Powered Assistive Tool is a web-based application designed to enhance accessibility for users by leveraging advanced image processing and text-to-speech technologies. This project is especially beneficial for individuals with visual impairments, enabling them to gain insights from images and access textual content in an auditory format.

The tool uses **Google Generative AI's Gemini model**, **Optical Character Recognition (OCR)** powered by Tesseract, and **Google Text-to-Speech (gTTS)** to provide high-quality scene understanding and seamless content accessibility.

**Features**

**1. Real-Time Scene Understanding**

* The tool generates descriptive textual interpretations of the uploaded images.
* Users can listen to the scene descriptions via text-to-speech functionality.
* Powered by **Google Generative AI’s Gemini model**, the feature provides natural and detailed descriptions.

**2. Text-to-Speech Conversion for Visual Content**

* Extracts text from images using **OCR (Tesseract)**.
* Converts the extracted text into audible speech using **gTTS**, ensuring accessibility to written content.
* Supports text formatting preservation for better auditory output.

**3. Project Description in Sidebar**

* A concise project description is displayed in the app's sidebar to provide context and usage instructions.

**4. Progress Indicators**

* Progress bars are displayed during heavy operations (like generating descriptions or processing OCR) to keep users informed about task status.

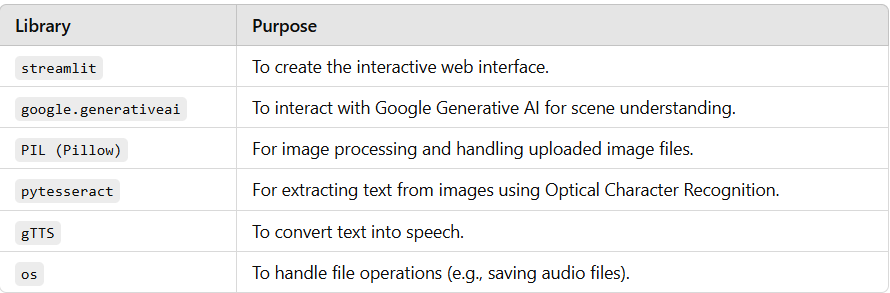
**How It Works**

**Steps to Use the Application**

1. **Upload an Image**:
   * Supported formats: PNG, JPG, JPEG.
2. **Select a Functionality**:
   * **Real-Time Scene Understanding**: Generates a description of the image and provides an option to listen to it.
   * **Text-to-Speech Conversion for Visual Content**: Extracts and reads out text from the image.
3. **Audio Output**:
   * Listen to the generated scene description or extracted text through the built-in audio player.

**Dependencies**

**Libraries Used**



**Setup and Installation**

**Prerequisites**

1. Python (>= 3.8)
2. A valid **Google Generative AI API key** with access to the Gemini model.
3. Tesseract-OCR installed on the system. You can install it as follows:
   * **Windows**: Download from [Tesseract GitHub](https://github.com/UB-Mannheim/tesseract/wiki).
   * **Linux/macOS**: Install via package manager (e.g., sudo apt-get install tesseract-ocr for Ubuntu).

**Installation**

1. Clone the repository:

**git clone https://github.com/your-repo/ai-powered-assistive-tool.git**

**cd ai-powered-assistive-tool**

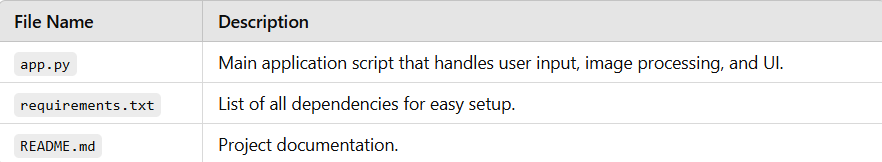
1. Install dependencies:

**pip install streamlit google-generativeai pytesseract gtts pillow**

1. Configure the Google Generative AI API key: Replace the placeholder in the code with your API key:

**genai.configure(api\_key="YOUR\_API\_KEY\_HERE")**

**Code Structure**



**Core Functionalities**

**1. Real-Time Scene Understanding**

* **Input**: User uploads an image.
* **Processing**:
  + The image is passed to the **Google Generative AI's Gemini model** with a prompt: *"Describe the scene in this image."*
  + The model generates a descriptive text output.
* **Output**:
  + Displays the textual description.
  + Converts the description into speech using gTTS.
* **Example**:
  + Input: An image of a dog sitting on a sofa.
  + Output: *"A brown dog is sitting on a gray sofa in a well-lit living room."*

**2. Text-to-Speech Conversion for Visual Content**

* **Input**: User uploads an image containing text (e.g., a document, signboard).
* **Processing**:
  + Text is extracted from the image using pytesseract.
  + The extracted text is cleaned to remove unnecessary characters or formatting.
* **Output**:
  + Displays the extracted text.
  + Converts the text into speech using gTTS.
* **Example**:
  + Input: An image of a signboard reading "Welcome to AI City."
  + Output:
    - Text: *"Welcome to AI City."*
    - Speech: Plays the audio reading the extracted text.

**Usage Instructions**

**Running the App**

1. Run the application:

**streamlit run app.py**

1. Open the local server link (e.g., http://localhost:8501) in your browser.
2. Interact with the app:
   * Upload images.
   * Choose between **Real-Time Scene Understanding** and **Text-to-Speech Conversion**.

**Future Improvements**

**Features to Add**

1. **Support for Multiple Languages**:
   * Expand OCR and text-to-speech capabilities to support languages other than English.
2. **Interactive Image Regions**:
   * Allow users to select specific parts of the image for analysis.
3. **Enhanced Scene Understanding**:
   * Use object detection or segmentation models to provide more detailed descriptions.

**UI Enhancements**

* Add a **dark mode**.
* Allow drag-and-drop functionality for image uploads.

**Performance Optimizations**

* Cache results for previously processed images to improve efficiency.
* Optimize image preprocessing for better OCR accuracy.

**Acknowledgments**

This project uses:

* **Google Generative AI** for scene understanding.
* **Tesseract-OCR** for text extraction.
* **Google Text-to-Speech (gTTS)** for audio output.

Special thanks to the open-source community for providing the tools and resources that made this project possible.

