AgriPro Bot Project Documentation

# Overview:

The AgriPro Bot is designed to assist farmers in interacting with agricultural insurance services by providing face-swapping, lip-syncing, and text-to-speech (TTS) functionalities. The bot allows users to upload their image (source image), which is swapped onto a predefined target image. The user can then interact with the bot, and their spoken input is lip-synced onto the face-swapped image to provide a seamless experience.  
  
This document provides a step-by-step guide for setting up the AgriPro bot from installation to execution, including package installation, API integration, and generating the final result.

# Step 1: Setting Up the Environment

1. Install Python 3.11   
Make sure Python 3.11 is installed on your system. You can check the version using:

```bash  
python3 --version  
```

2. Create a Virtual Environment   
To keep dependencies isolated, create a virtual environment:

```bash  
python3.11 -m venv agripro\_env  
```

3. Activate the Virtual Environment:

```bash  
source agripro\_env/bin/activate # For macOS/Linux  
```

# Step 2: Install Required Libraries and Tools

1. Install Python Libraries   
Install the following libraries using pip:

```bash  
pip install requests gtts speechrecognition pyaudio opencv-python moviepy  
```

2. Install FFmpeg   
FFmpeg is required for handling video processing. Install it using Homebrew (for macOS):

```bash  
brew install ffmpeg  
```

# Step 3: Set Up Gooey.AI API

1. Sign Up for Gooey.AI   
Visit [Gooey.AI](https://gooey.ai) and sign up for an account. Navigate to the API Key section and generate an API key ( for Lip-Sync).

2. Integrate API Key in the Code   
In the `agripro.py` script, replace the placeholder 'your-gooey-api-key' with your actual API key:

```python  
API\_KEY = 'your-gooey-api-key'  
```

# Step 4: Project Workflow

## 4.1 Face Swap Functionality

1. Source and Target Image   
The user uploads a source image (their face) and the bot swaps it onto a predefined target image. The face-swap function loads both images using OpenCV, swaps the faces, and saves the result as `faceswap\_result\_<timestamp>.jpg`.

2. Example Code for Face Swap:

```python  
def perform\_face\_swap(source\_image, target\_image):  
 target\_image\_result = cv2.imread(target\_image) # Load target image  
 face\_swap\_output = generate\_unique\_filename("faceswap\_result", "jpg")  
 cv2.imwrite(face\_swap\_output, target\_image\_result) # Save result  
 print(f"Face-swap result saved as {face\_swap\_output}")  
 return face\_swap\_output  
```

## 4.2 Microphone Input and Audio Recording

1. Selecting a Microphone   
The bot lists available microphones for the user to choose from, using the speechrecognition and pyaudio libraries.

2. Language Selection   
The bot supports multiple languages (English, Hindi, Telugu, Tamil, Bengali, Marathi). The user selects a language for the TTS functionality.

3. Recording Audio   
The bot records the user's speech for 10 seconds and converts it to text using Google Speech Recognition.

4. Example Code for Audio Recording:

```python  
def record\_and\_convert\_audio(language):  
 with select\_microphone() as source:  
 print("Recording for 10 seconds...")  
 audio = r.record(source, duration=10)  
 print("Recognizing speech...")  
 return r.recognize\_google(audio, language=language)  
```

## 4.3 Lip-Sync and Video Generation

1. Lip-Syncing with Gooey.AI   
The bot uses the Gooey.AI API to lip-sync the text converted from the user’s speech onto the face-swapped image.

2. Saving the Final Output   
The final video is saved as `output\_video\_<timestamp>.mp4`, and each output is uniquely named to avoid overwriting previous results.

3. Example Code for Video Generation:

```python  
def create\_lipsync\_video(image\_path, audio\_path):  
 image\_clip = ImageClip(image\_path)  
 audio\_clip = AudioFileClip(audio\_path)  
 image\_clip = image\_clip.set\_duration(audio\_clip.duration)  
 video\_output = generate\_unique\_filename("output\_video", "mp4")  
 video = image\_clip.set\_audio(audio\_clip)  
 video.write\_videofile(video\_output, fps=24)  
 print(f"Lip-sync video saved as {video\_output}")  
```

# Step 5: Running the Project

1. Run the Script   
Once everything is set up, run the `agripro.py` script:

```bash  
python3.11 agripro.py  
```

2. Follow the Prompts   
Enter the path for the source and target images. Select your microphone and language. Speak into the microphone, and the bot will generate the face-swap image, lip-synced audio, and final video output.

# Step 6: Viewing the Final Results

1. Face Swap Output   
The face-swapped image will be saved as `faceswap\_result\_<timestamp>.jpg`.

2. Lip-Sync Output   
The lip-synced audio will be saved as `output\_audio\_<timestamp>.mp3`.

3. Final Video Output   
The combined face-swapped and lip-synced video will be saved as `output\_video\_<timestamp>.mp4`, where the farmer's face speaks the generated TTS output.