# Documentation for Automated YouTube Transcript Downloader

## Overview

This project provides a simple Python GUI application for inputting YouTube links and downloading their transcripts as a text file. The application uses the tkinter library for the GUI and the youtube-transcript-api library to fetch transcripts.

## Prerequisites

1. **Python Installation**: Ensure you have Python installed on your system. You can download it from [python.org](https://www.python.org/downloads/).
2. **Library Installation**: Install the required libraries using pip:

pip install youtube-transcript-api

## File Structure

bash

Copy code

transcript\_downloader/

│

└── transcript\_downloader.py # Main script file

## Main Script (transcript\_downloader.py)

### Imports

import tkinter as tk

from tkinter import messagebox

from youtube\_transcript\_api import YouTubeTranscriptApi

from youtube\_transcript\_api.formatters import TextFormatter

* **tkinter**: Used for creating the GUI.
* **messagebox**: Provides message boxes for user feedback.
* **YouTubeTranscriptApi**: Fetches YouTube transcripts.
* **TextFormatter**: Formats the transcripts into plain text.

### Function: fetch\_transcript

This function handles the fetching and saving of YouTube transcripts.

def fetch\_transcript():

links = text\_input.get("1.0", "end-1c").strip().split('\n')

transcript\_texts = []

for link in links:

video\_id = link.split("v=")[-1]

try:

transcript = YouTubeTranscriptApi.get\_transcript(video\_id)

formatter = TextFormatter()

transcript\_text = formatter.format\_transcript(transcript)

transcript\_texts.append(f"Transcript for {link}:\n\n{transcript\_text}\n\n")

except Exception as e:

messagebox.showerror("Error", f"Could not fetch transcript for {link}: {e}")

return

with open("transcripts.txt", "w", encoding="utf-8") as file:

for transcript\_text in transcript\_texts:

file.write(transcript\_text)

messagebox.showinfo("Success", "Transcripts saved to transcripts.txt")

* **Reading Input**: Reads the input links from the text box.
* **Fetching Transcripts**: Extracts the video ID and fetches the transcript for each video.
* **Formatting Transcripts**: Formats the transcripts into plain text.
* **Saving to File**: Saves the combined transcripts to a text file.
* **User Feedback**: Shows success or error messages.

### GUI Setup

The following code sets up the GUI.

root = tk.Tk()

root.title("YouTube Transcript Downloader")

instructions = tk.Label(root, text="Enter YouTube links (one per line):")

instructions.pack(pady=10)

text\_input = tk.Text(root, height=10, width=50)

text\_input.pack(pady=10)

fetch\_button = tk.Button(root, text="Fetch Transcript", command=fetch\_transcript)

fetch\_button.pack(pady=10)

root.mainloop()

* **Root Window**: Initializes the main window with a title.
* **Instructions Label**: Provides instructions for the user.
* **Text Input**: A text widget for users to input multiple YouTube links.
* **Fetch Button**: A button that triggers the fetch\_transcript function.
* **Main Loop**: Starts the GUI event loop.

## Running the Application

1. **Save the Script**: Save the provided code in a file named transcript\_downloader.py.
2. **Run the Script**: Open a terminal or command prompt and navigate to the directory containing the script. Run the script using Python:

python transcript\_downloader.py

1. **Use the GUI**:
   * Enter YouTube video links in the text box (one per line).
   * Click the "Fetch Transcript" button.
   * The transcripts will be fetched and saved to transcripts.txt in the same directory as the script.

## Additional Features

### Error Handling

The application includes basic error handling to notify users if a transcript could not be fetched.

### Customization

Users can extend the functionality by:

* Adding options for selecting the output file location.
* Enhancing error handling for various types of errors.
* Improving the user interface for better user experience.

### Further Development

For advanced features, consider integrating more robust NLP techniques for transcript processing and adding functionalities like combining multiple transcripts into a single cohesive document.

## Conclusion

This documentation covers the setup, usage, and code explanation for the Automated YouTube Transcript Downloader. By following these steps, users can easily input YouTube links and download their transcripts as a text file. The project can be further extended to include more advanced features as needed.