YOUTUBE SUBTITLES DOWNLOADER

Project report in partial fulfilment of the requirement for the award of the degree of

Bachelor of Computer Applications

Submitted By

Raja Saha	University Enrolment No. 12019004009099
Arindip Dutta	University Enrolment No. 12019004009098
Tanay Mondal	University Enrolment No. 12019004009043
Soumesh Santra	University Enrolment No. 12019004009107
Rishav Khamrui	University Enrolment No. 12019004009057

Under the guidance of

Prof. Ankur Biswas

Department of Computer Applications

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700156

UNIVERSITY
OF ENGINEERING & MANAGEMENT
'University Area', Plot No. III-B/5, Main Arterial Road, New Town, Action Area-III, Kolkata - 700 160, W.B., India
City Office : 'ASHRAM', GN-34/2, Salt Lake Electronics Complex, Kolkata - 700 091, W.B., India
(Established by Act XXV of 2014 of Govt. of West Bengal & recognised by UGC, Ministry of HRD, Govt. of India)

Ph. (Office) : 91 33 2357 7649
: 91 33 2357 2969
: 91 33 6888 8608
Admissions : 91 33 2357 2059
Fax : 91 33 2357 8302
E-mail : vc@uem.edu.in
Website : www.uem.edu.in

CERTIFICATE

This is to certify that the project titled YOUTUBE SUBTITLES DOWNLOADER submitted by Raja Saha (University Enrolment No. 12019004009099), Arindip Dutta (University Enrolment No. 12019004009098), Tanay Mondal (University Enrolment No. 12019004009043), Soumesh Santra (University Enrolment No. 12019004009107) and Rishav Khamrui (University Enrolment No. 12019004009057), students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of the requirement for the Degree of Bachelor of Computer Applications, is a bonafide work carried out by them under the supervision and guidance of Prof. Ankur Biswas during 6th Semester of the academic session of 2021 - 2022.

The content of this report has not been submitted to any other university or institute. I am glad to inform you that the work is entirely original and its performance is found to be quite satisfactory.

_____

Prof. Ankur Biswas
Assistant Professor
Department of Computer Applications
UEM, Kolkata

_____

Prof. Kaustuv Bhattacharjee
Head of Department,
Department of Computer Applications
UEM, Kolkata

ACKNOWLEDGMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remain invaluable to us.

We are sincerely grateful to our guide Prof. Ankur Biswas of the Department of Computer Applications, UEM, Kolkata, for his wisdom, guidance, and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Kaustuv Bhattacharjee, HOD, Department of Computer Applications, UEM, Kolkata, and all other departmental faculties for their ever-present assistance and encouragement.

Last but not least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Raja Saha

Arindip Dutta

Tanay Mondal

Soumesh Santra

Rishav Khamrui

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

As we stand in the middle of the year 2022, YouTube is the top and major video sharing platform among the non-subscription-based video distribution platform. Since its launch (The year 2005), no other platform made its way to the top over YouTube. In the past 16 years, YouTube has gained a major audience base of 2 billion per month. Every day there are almost 1 billion hours of videos uploaded on YouTube. But whenever we open a video, we sometimes realize that not every single video is in our native language. So that's why YouTube implemented Video Captions or Subtitles which show a written transcript over the video as an overlay. But sometimes we need those captions for our work, education, and research purposes. So, we made a simple tool using Python to extract and download the subtitle in a text file from the particular YouTube video Link. This tool is very simple and only needs a YouTube video link that has at least one caption to operate.

All the installation instructions, examples and necessary details are mentioned below for a brief working behavior for this tool.

2. LITERATURE SURVEY

2.1 Problem Statement: -

Nowadays in the digital world, we can acquire knowledge from any source and one of the best examples of this kind of platform is YouTube. In recent days more and more teachers and educational groups, they are teaching on YouTube, with live classes or pre-recorded classes. While studying from this kind of platform, for better understanding sometimes the transcript is necessary. In other words, we need a complete conversational subtitle in a readable format which is very useful. For this reason, our team came up with this idea, a script that will help the user to download the subtitle of any video from YouTube.

2.2 Initiative: -

In the beginning, our team didn't know how to start or where to start. After doing some research we decided that we will make this project/script with the python programming language. The reason why we used python, over other languages is because of the libraries. It's very easy to understand as well as to use in the script. And in our group, everyone can work with python, this is also a reason why we used python in our project

We started learning python from the Books, the internet, PDFs, websites, and YouTube channels.

While learning the language we came to know that for our project, we need one very important python library – [*youtube-transcript-api]* and we learned about this library from a very popular website "Geeks for Geeks" [i].

And about the library, we gathered the required knowledge from the PyPI website [ii].

The next problem was how to specify and shorten the YouTube link to our script so that it can understand and use it without any issue. So we solved this problem with the Inbuilt function [RegEx] and we also learned about this function from W3School [iii].

In the next step, we customized our script with the help of ANSI ESCAPE CODE [iv].

The next problem was how to save the output file in a text format and we solved this problem with help of the documentation from the Geeks for Geeks website [v].

For documentation purposes, we needed a Use Case Diagram, and we learned that from the tutorials point website [vi].

And to make our diagram we used this website [vii].

2.3 Other Similar Tools: -

There are a vast number of websites and software available to download subtitles. But whenever we use this third-party tool, we are being used for ad targeting or it may be collecting our valuable data and selling it to the companies which later used it for ad targeting.

But the python tool which we have created is completely ad-free, easy to use, and open source for anyone to use or modify for their preferences.

3. REQUIREMENT ANALYSIS

3.1 Hardware Requirements

To run or use our script the hardware requirements are very minimal a system compatible with running python version 3.9 it could be either windows / Linux/ mac it must run python and also must have pip installed. Then used have to install the required library to start using our script.

3.2 Software Requirement

As we made this script in a Linux based OS but not everyone is using Linux OS so there might be some users who are using windows or mac OS they can use IDE(VScode, pycharm, IntelliJ IDEA) and they can download the library in their IDE and then start using our script. Not only IDE they can use this in PowerShell and for mac users, but they can also run it in the terminal.

4. WORKING BACKGROUND

We used one Python library and it is [youtube_transcript_api] this is the main library that we used to do the main work which is fetching the text from the page source and give back to the user. About the library, This is a python API that allows you to get the transcript/subtitles for a given YouTube video. It also works for automatically generated subtitles and supports translating subtitles. To install it the user must type this command [pip install youtube_transcript_api]. And the way it's doing it using the unique video id of the youtube video, for example, the video URL is - "youtube.com/watch?v=PP9bexwAdwA", in this URL the unique id is - SW14toda_KI the library is using the id to fetch the sentence/phrase and return it to the user.

5. PROPOSED WORK

After gathering all the knowledge, all the necessary documentation for the re`3 we started our program build, here are the steps that we followed –

- At first we call our main library "youtube-transcript-api" which is the heart of our program.
- Then call "sys" to save our output in the system. And at the last, we call "re" for RegEx.
- We want to make our code as simple as possible so we divide the code into small parts using function.
- In the program we call the function "welcome()", which is welcome the user.
- After that our program asks for the YouTube link from the user, and our program's function "id_extractor()" checks whether the YouTube link is valid or not. If the YouTube link is valid then the program extracts that specific id from the link.
- Then the id goes as an input to our next function "working()". Our "working()" function is giving choice to the user of normal English subtitles or Translated subtitles.
- If the user selects the normal subtitles our program goes to the "real_caption()" function and executes directly.
- If the user selects the Translated subtitles our program goes to the "For_translation()" function and asks for Hindi, Bengali, and Spanish translation, and based on user choice our program prints that output.
- Next our program's function "for_output()" ask for a name for the user's output file and it saves on the current directory.
- And we have the "wrong()" function which helps the user to make sure any unwanted actions are made by the user.
- At the last our "exit()" function greets the user and print a thank you message.

In this way, we finished our script. After this step, we customized our script.

6. RESULTS AND DISCUSSION

As the result of our script in other words the output is straight forward after the user gave the preferred output language our script will fetch the text from the page source and make it a text file and return it to the folder where the user has the script. If there is any error while fetching the text or any kind of problem while downloading it as a text file it will give the error in the terminal. For example, if that particular video has no subtitle or there is no auto-generated transcript, our script can not fetch it so the user might get an error. In future improvements, we will try to give a proper response to the user that ‘ This video is incompatible with transcripts/auto-generated subtitles ’. In the end, we have uploaded our project to our GitHub page to make it available for everyone. Because our project is open-source.

7. HOW TO USE

For this script [*youtube_transcript_api*] python library is required and to install it the user should type [*pip install youtube_transcript_api*] or user can user requirements.txt file to install all the necessary modules and to do that user can simply type [*pip install -r requirements*]

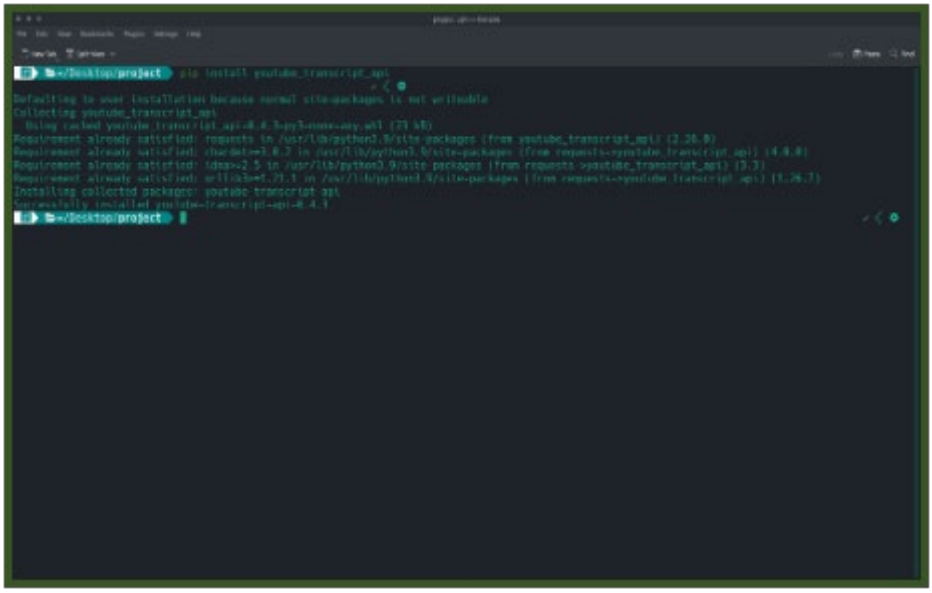

Figure 1 Starting YouTube Downloader

After completing the basic requirements user can proceed further.

STEP 1:-

Open the terminal change the current working directory to the folder where the script is [cd "the folder name"] after that to use the script user must type the following -

[python Youtube-Subtitle-Downloader.py]

After this step, the script will ask for the video URL, enter the video URL

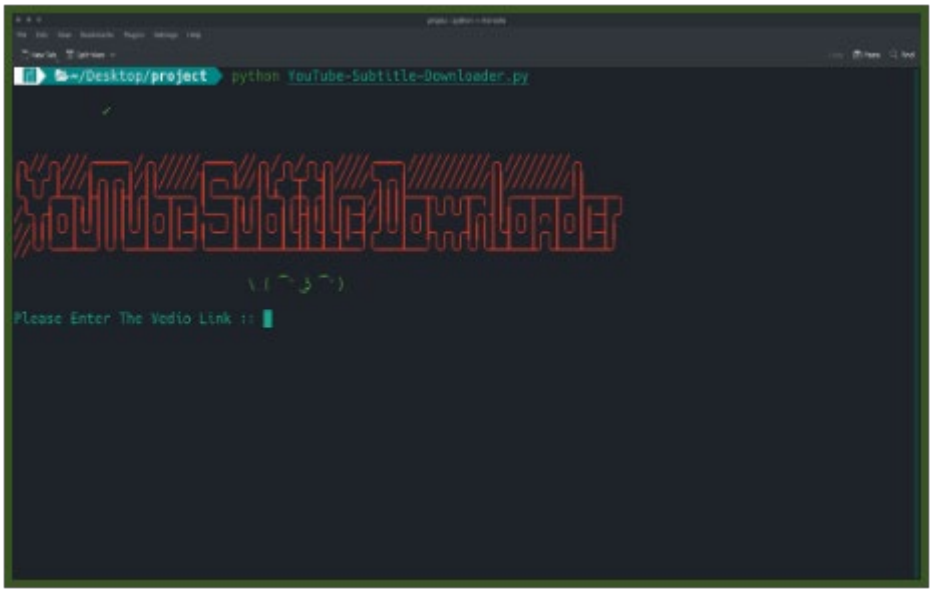

Figure 2 Starting Page - Insert Link

STEP 2:-

Now, we need a YouTube video that at least has one subtitle file attached to it, we need to copy the URL of that particular video.

for example, use this YouTube link – "youtube.com/watch?v=PP9bexwAdwA"

STEP 3:-

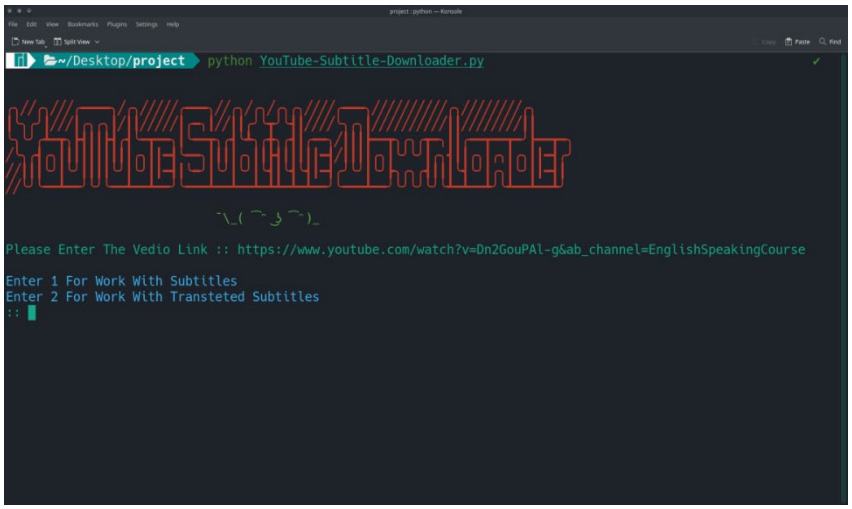

Figure 3 Menu Selection

When you enter the video URL, there you can find two options, select 1 for the original English subtitle and select 2 for translated subtitles.

If you choose option 1 for the original English subtitle then you are directly redirected to choose the preferred file name for the subtitle file.

And if the user wants to work with translated subtitles they have to select option 2

After selecting option 2, there you can find 3 languages Hindi, Bengali, and Spanish. The next step is to select the language using code like ‘hi’ for HINDI, ‘bn’ for BENGALI, and ‘es’ for SPANISH.

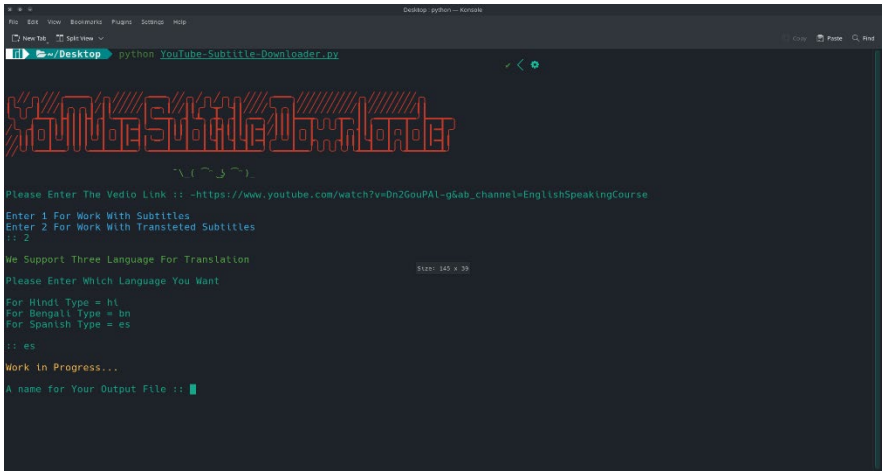

Figure 4 Output filename declaration

Again in the same way input your preferred file name,

STEP 5:-

In the end, you can find your subtitle file in the dedicated folder

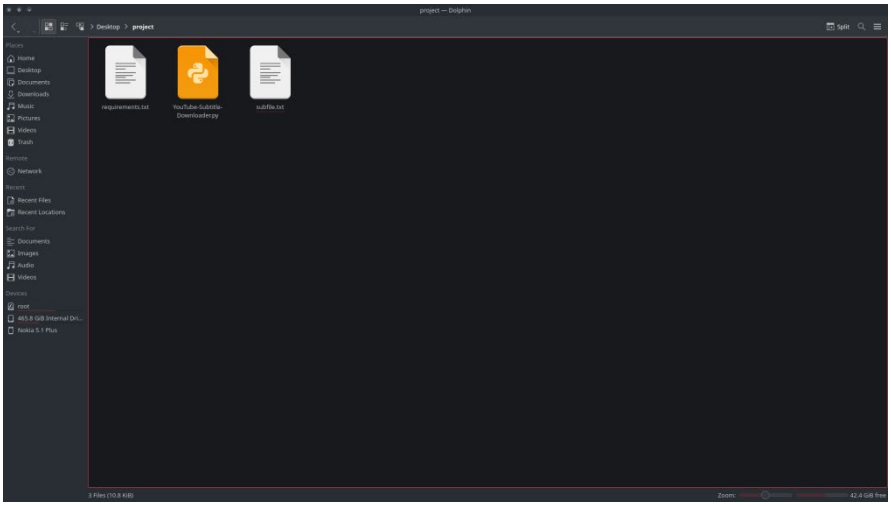

Figure 5 File creation

8. USE A CASE DIAGRAM

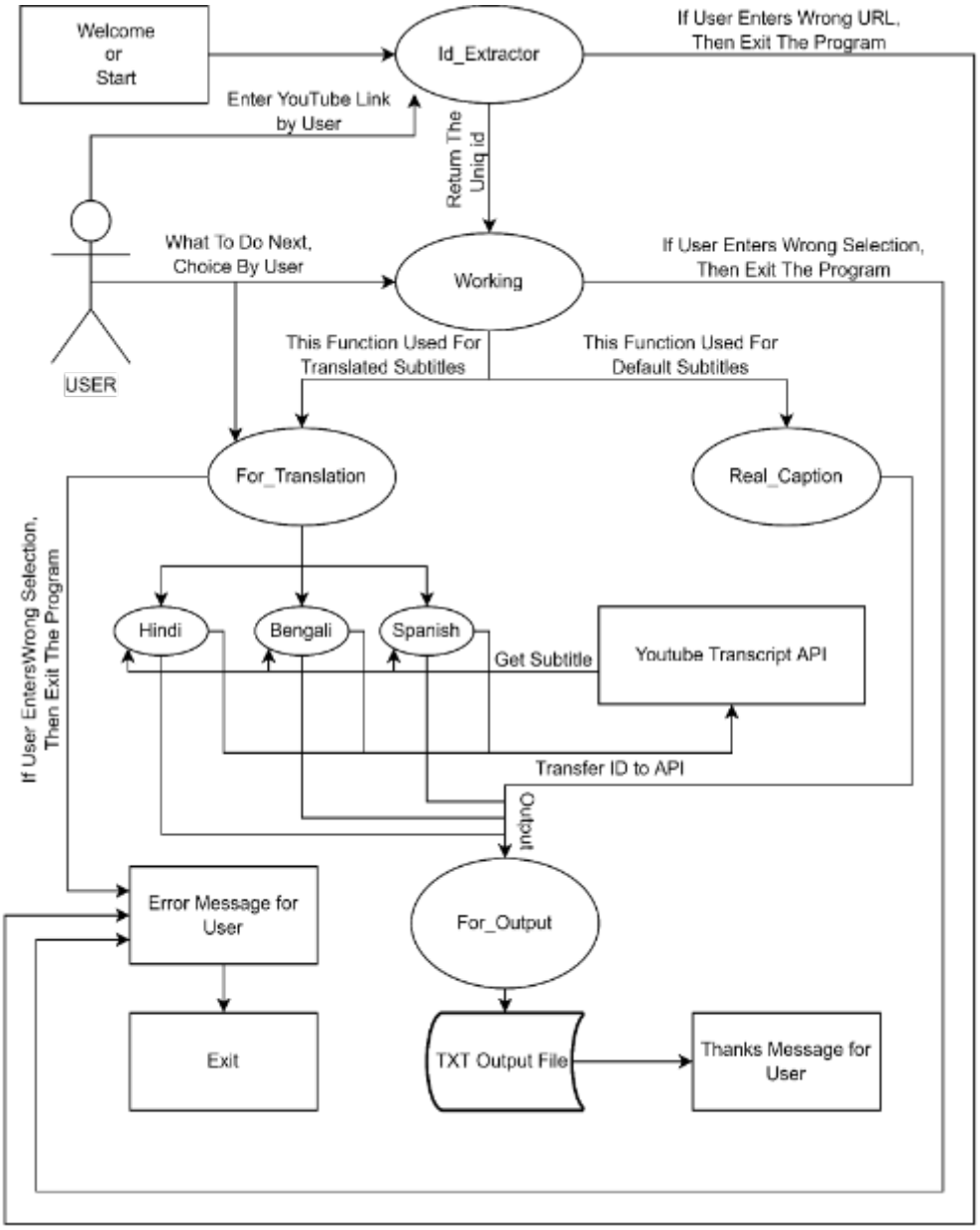

Figure 6 Use Case Diagram

9. CONCLUSION

At the conclusion of the project, we achieved a better understanding of how to use libraries and APIs in our scripts, and also we learned new concepts in python language.

The best significance of our script is open source and everyone can see the inner working of the program. There are some websites, applications to do the same work but we don’t know and we can not see what is going on behind their process, there might be some trackers, or they might collect data, we don’t know that so in this purpose our script is very secure and trustworthy.

10. FUTURE WORKS

Currently, our script supports only 4 languages, but our aim is to make it useable for more languages.

Currently, we are preparing to make this script as a chrome extension so that users can download the subtitle in one click.

For example, if the user is watching any video on YouTube, and wants to download the video with our chrome extension user can easily click on it and it will download the subtitle file in text format. And in this way, everyone will be able to use our script.

11. REFERENCE

- Headfirstst python. Author – Paul Barry
- Elements of Programming Interviews in Python: The insider's Guide. Author - Adnan, Amit, and Tsung-Hsien
- [*youtube-transcript-api* website "Geeks for Geeks"
    - [i] Page URL- (https://www.geeksforgeeks.org/python-downloading-captions-from-youtube/)
- PyPIpi website
    - [ii] Page URL- (https://pypi.org/project/youtube-transcript-api/)
- [RegEx] from W3School.
    - [iii] Page URL- (https://www.w3schools.com/python/python_regex.asp)
- Customized our script with the help of ANSI ESCAPE CODE.
    - [iv] Page URL- (https://www.geeksforgeeks.org/how-to-add-colour-to-text-python/)
- How to save the output file in a text format we solved this problem with help of the documentation from the Geeks for Geeks website.
    - [v] Page URL- (https://www.geeksforgeeks.org/saving-text-json-and-csv-to-a-file-in-python/)
- Use Case Diagram, and we learned that from the tutorials point website.
    - [vi] Page URL- (https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm)
- And to make our diagram we used this website.
    - [vii]Page URL- (https://app.diagrams.net/)