“In Pursuit of Technical Excellence”

A
Project Report on

“YouTube Audio Video Downloader”

For The Degree of

MASTER OF COMPUTER APPLICATION

Submitted by

RAKESH ARUN CHAURE

(MC20F14F012)

SANKET HARSHWARDHAN GHODKE

(MC20F14F021)

GOPAL SUBHASH GHULE

(MC20F14F022)

SWAPNIL SANJAY JADHAV

(MC20F14F025)

Under the Guidance of

Dr. Praveen Shetiye

Department of Master of Computer Application

Government College of Engineering, Aurangabad

(An Autonomous Institute of Government of Maharashtra)

(2021-2022)

CERTIFICATE

This is to certify that the major project report on **"YouTube Audio Video Downloader",** which is being submitted by **'Rakesh A. Chaure'** Enrollment No. **MC20F14F012, 'Sanket H.Ghodke'** Enrollment No. **MC20F14F021, 'Gopal S. Ghule'** Enrollment No. **MC20F14F022, 'Swapnil S. Jadhav'** Enrollment No. **MC20F14F025** is a bonafied work completed under my supervision and guidance. In partial fulfilment for award of the degree of **'Master of Computer Application'** of Government Engineering College, Aurangabad (An Autonomous Institute of Government of Maharashtra) affiliated to Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Date: 18-12-2021 **Place:** Aurangabad

Dr. Praveen Shetiye
Guide
MCA Department
Aurangabad

Dr. Praveen Shetiye
Head
MCA Department
Aurangabad

Dr. U. J. Kahalekar
Principal
Government College of
EngineeringAurangabad

ACKNOWLEDGEMENT

Firstly, we would like to express our gratefulness to Dr. Praveen Shetiye Sir is our project guide, for their work supervision and valuable suggestions and thoughts providing guidelines, co-operation and extended help during this work of dissertation has made it possible to complete this project work. Their broad-based knowledge it this field realized us appreciation and encouragement served as a constant.

I am thankful to our Head of Department (Master of Computer Application) Dr. Praveen Shetiye for his advice & suggestion throughout the project. Last, but not the least, we would like to thank for the timely assistance and suggestion of numerous other faculty members, friends and parents.

ABSTRACT

YouTube Audio Video Downloader is a windows based project. It acts a windows application for the public to download YouTube videos/audios in any quality as per the user needs. And the videos once downloaded can be shared with others. There is no need of any login to download any video and the downloader is ads and pop-ups free. It also has GUI based user friendly interface through which user has to just paste any YouTube URL in the URL bar and user will be introduce to a List Box containing all the video resolutions then the user can select any quality based on his choice and the video will be started downloading with time left and progress bar showing up.

Contents

List of Figures/Screenshots

List of Tables

1. INTRODUCTION

With so much information, entertainment, and educational material online, watching videos is a great way to learn more about specific topics of interest. However, it can be hard to watch these videos if you happen to have a slow or intermittent internet connection, as it can cause videos to buffer or lag.

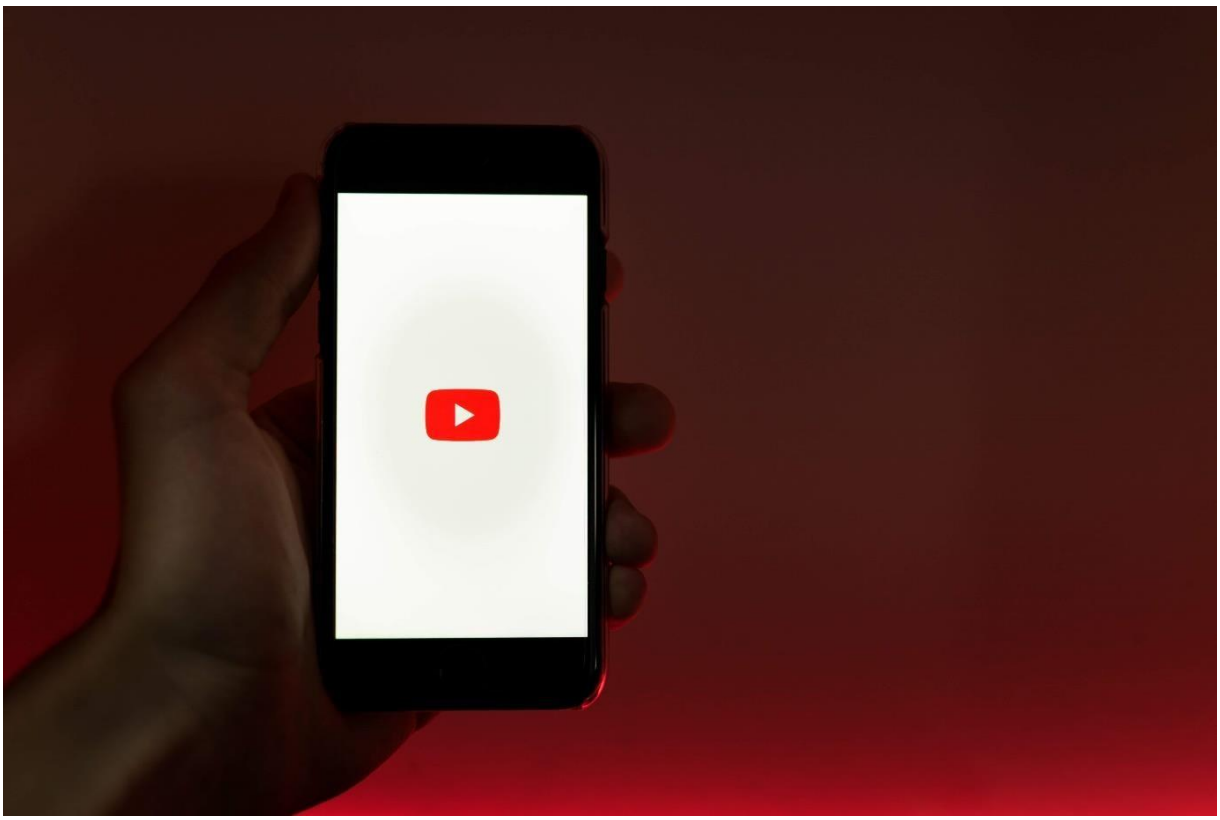


Figure 1. YouTube Sample Image

That's why we are going to be using Python to create a YouTube video downloader that can be used to download our favorite YouTube videos directly to your computer. This will allow you to watch them directly from your device, making them accessible at all times. Whether you are new to Python or have some prior experience, this project is a great way to learn more about making functional programs and helps you apply your knowledge in a practical and fun way, and is a great Python project.

The YouTube Video downloader aims at downloading any type of video from YouTube in a fast, simple and easy way. The user has to copy the YouTube video URL link which needs to be pasted in the application. Later, the user needs to click on the download button in order to download the video. Python YouTube Video Downloader is an application to download videos from YouTube. This provides users to download videos they need in their devices and watch them offline.

We all watch videos on YouTube, right? YouTube is full of videos of our interest. But YouTube doesn't give you the liberty to download their video and save it in your local storage. Everyone faces difficulty in downloading YouTube videos. But we are programmers we are not common at all.

Now with the help of the Python programming language, you can download any YouTube video in the highest quality possible. All you need is a good internet connection. So, let's see how it works. Take a cup of coffee and let's begin the hack. The YouTube downloader project is a python project. The object of this project is to download any type of video in a fast and easy way from YouTube in your device.

In this python project, user has to copy the YouTube video URL that they want to download and simply paste that URL in the 'paste link here' section and click on the download button, it will start downloading the video. When video downloading finishes, it shows a message 'downloaded' popup on the window below the download button

YouTube is very popular video sharing website. Downloading a video from YouTube is a tough job. Downloading the Downloader and get the video using that or go to any other website which fetches the video and saves on your computer. Using Python, this task is very easy. Few lines of code will download the video from YouTube for you. For this, there a python library named as 'tkinter'. tkinter is a lightweight, dependency-free Python library which is used for downloading videos from the web.

YouTube is a very popular video-sharing website. Downloading a video's/playlist from YouTube is a tedious task. Downloading that video through Downloader or trying to download it from a random website increases the risk of licking your personal data. Using the Python Tkinter package, this task is very simple-efficient-safe. Few bunch codes will download the video for you. The basic idea is that the app consists of a text bar where the user enters the URL of the video and then clicks on the "Download" button which then downloads the video. The app shows a message when the download is complete.

Python is a high-level general-purpose programming language which is used for various applications. Using python, you can make a web application, desktop application, Games, etc. In this python project, we will discuss how to download YouTube videos using python and also, we will create a nice UI using the Tkinter library.

YouTube is a widely used video-sharing platform downloading a video from YouTube is a hectic task. To download a video, you need to copy and paste the link to another site or you need to install an extension which will display unwanted ads. To avoid all the headaches and also to develop your programming skill I will show you how to download a YouTube video.

1. 100 -120 hours of video content is added to YouTube every 10 minutes.
2. To watch all the available YouTube videos, it will take more than 1,00,000 years.
3. More than 1.8 billion users sign into YouTube every month.
4. Most of the programmers around the world use YouTube as the main source of learning, because of the huge knowledge base (Lots of educational videos).

The main reason why we are talking about YouTube is that with huge amount of content available on YouTube, most of us might have been using third-party applications to download videos from YouTube to access them offline. And even though we use third-party applications or plug-ins we still end up downloading low quality videos and have to upgrade to so called "Premium Account" to download the videos in high quality.

So why not build your own application to download YouTube videos? How?

Using tkinter module of python.

Unlike some other famous libraries, tkinter doesn’t have any third-party dependencies. tkinter is a light-weight python library with a rich set of features for downloading YouTube videos developed by Nick Ficano. This open-source project is very easy to use.

1.1 Background:

1.1.1. About YouTube Video Downloader:

Aren't there a lot of YouTube videos you wished you could've watched without the internet on your laptops or desktops?

Well, here's your solution. In this project, we will create a YouTube video downloader where you can download the video straight from YouTube to your laptop. The objective of this is to create a GUI based YouTube Video Downloader. To build this, you will need a little understanding of Tkinter. This project is very good and it solves a problem pertinent to most.

Figure 2. YouTube Downloader Application

To build this Python YouTube Video Downloader project, we will need the following libraries:

- **Tkinter -** To Create the GUI.
- **Pafy** – To Download the YouTube videos and audios.

1.1.2. Python GUI – tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit or in simple words Tkinter is used as a Python Graphical User interface. Tkinter is the native library, you don't need to install it externally, just import, while you use it.

Python has a lot of GUI frameworks, but Tkinter is the only framework that's built into the Python standard library. Tkinter has several strengths. It's cross-platform, so the same code works on Windows, macOS, and Linux. Visual elements are rendered using native operating system elements, so applications built with Tkinter look like they belong on the platform where they're run.

- **To create a tkinter app:**
    1. Importing the module – tkinter
    2. Create the main window (container)
    3. Add any number of widgets to the main window
    4. Apply the event Trigger on the widgets.

1.1.3. Pafy Module:

This Python and Tkinter based GUI allows users to directly download the Best Resolution Videos and Audios from YouTube. Pafy is a Python library to download YouTube content and retrieve metadata.

Use the package manager pip to install pafy. Pafy along with YouTube-dl should be installed.

- pip install pafy
- pip install youtube_dl

Pafy is optionally depends on youtube_dl so therefore for more stable usage it is recommended to install youtube_dl before installing pafy. Below is the command to install youtube_dl.

1.1.4 youtube_dl:

Each and every day, you must be watching some video or another on YouTube which may be related to Music, Movies, Studies, Research, Leisure etc. You might want to store some video for future use where it will be required due to lack of internet or saving data or any other reason.

What if we tell you that you can do this exact very thing using Python.
Install the module with this command –

- **pip install youtube_dl**

1.1.5 Pyinstaller:

Pyinstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or any modules. Pyinstaller reads a Python script written by you. It analyses your code to discover every other module and library your script needs in order to execute. Then it collects copies of all those files – including the active Python interpreter! – and puts them with your script in a single folder, or optionally in a single executable file.

Pyinstaller is tested against Windows, Mac OS X, and GNU/Linux. However, it is not a cross-compiler: to make a Windows app you run Pyinstaller in Windows; to make a GNU/Linux app you run it in GNU/Linux, etc. Pyinstaller has been used successfully with AIX, Solaris, FreeBSD and OpenBSD, but is not tested against them as part of the continuous integration tests.
Pyinstaller is available on PyPI. You can install it through pip:

- **Pip install pyinstaller**

1.1.6 Python Random Module:

Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

Example: Printing a random value from a list

1.1.7 Tkinter File dialog:

Python Tkinter (and TK) offer a set of dialogs that you can use when working with files. By using these you don't have to design standard dialogs yourself. Example dialogs include an open file dialog, a save file dialog and many others. Besides file dialogs there are other standard dialogs. File dialogs help you open, save files or directories. This is the type of dialog you get when you click file, open. This dialog comes out of the module, there's no need to write all the code manually. Tkinter does not have a native looking file dialog, instead it has the customer tk style. You can see these below. The file dialog will work on all desktop platforms.

1.1.8. Threading Library in Python:

Python threading allows you to have different parts of your program run concurrently and can simplify your design. If you've got some experience in Python and want to speed up your program using threads. These different parts are usually individual and have a separate unit of execution belonging to the same process. The process is nothing but a running program that has individual units that can be run concurrently. For example, A web-browser could be a process, an application running multiple cameras simultaneously could be a process; a video game is another example of a process.

A thread has its flow of execution, which means that the process will have multiple things happening at one time. It is important to note that each process has at least one thread, and that is called the main thread. If your program doesn't have any defined thread, then it will at least have the main thread, i.e., the process itself.

1.2 Existing System and Need for System:

Comparison Between YouTube Downloader vs YT1s YouTube Downloader:

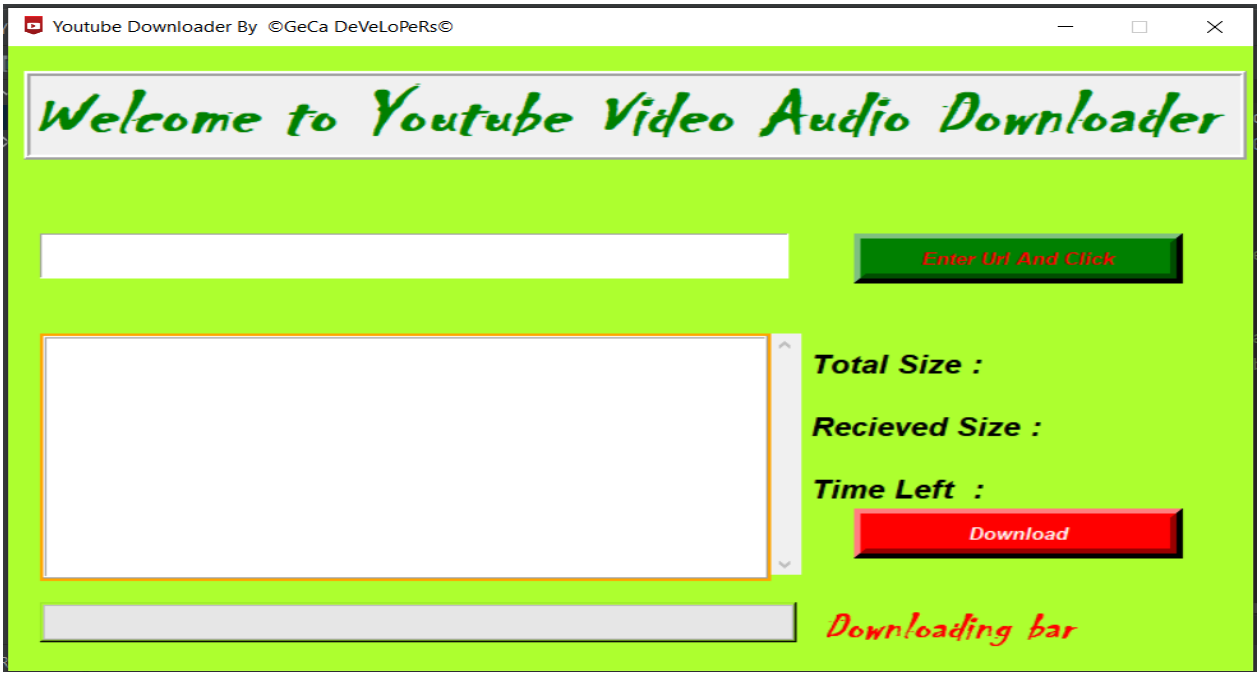


Figure 3. Welcome to YouTube Video Audio Downloader

1. YouTube Downloader is a free YouTube video downloader for PC to save YouTube videos. You can use this tool to convert YouTube videos to MP3. With YouTube Downloader you can save videos YouTube channel and thumbnail in 480P,720P,1080P, or 4K formats.

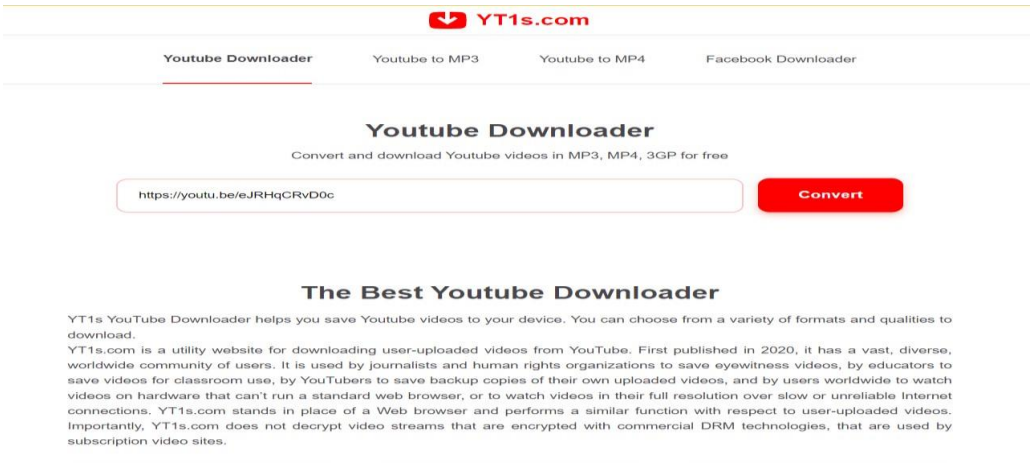


Fig 4. YT1s YouTube Downloader

1. YT1s YouTube Downloader is a free YouTube video downloader for PC to save YouTube videos. You can use this tool to convert YouTube videos to MP3. With YouTube Downloader you can save videos. But in the YT1s not possible to downloadthe thumbnails.

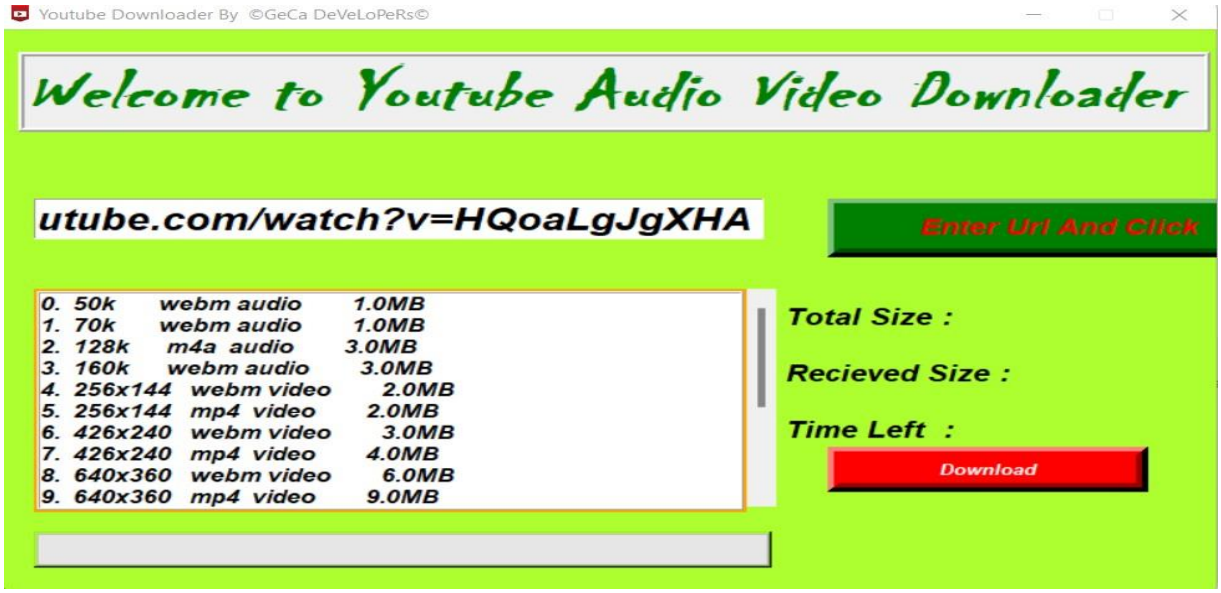


Figure 5. Enter URL and Click the Button

2. User has to copy the YouTube video URL that they want to download and simply pastethat URL in the 'paste link here' section and click on the download button, it will start downloading the video. When you entered the URL that shows the media type without any advertise and downloading that video through Downloader or trying to download it from a random website increases the risk of licking your personal data. It's very Secure Downloader.

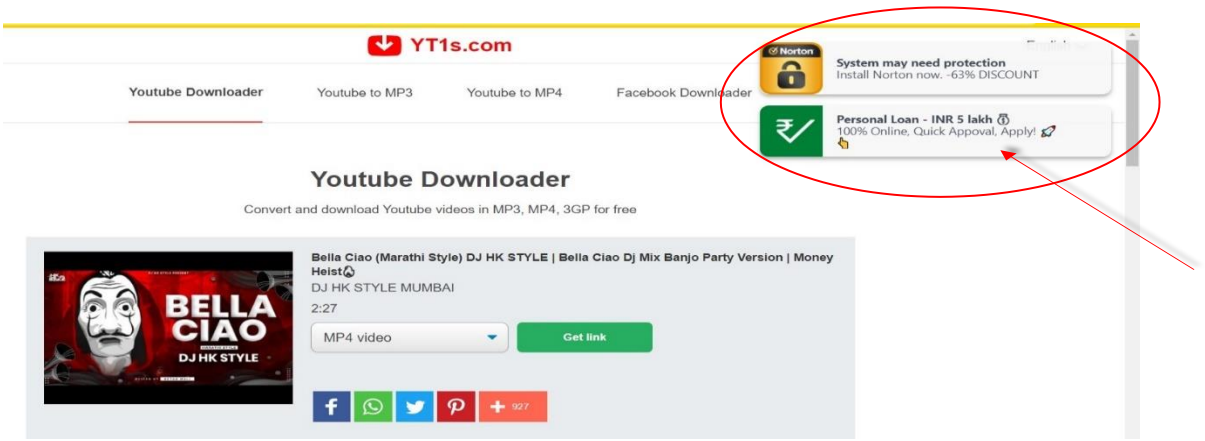


Figure 6. Enter URL and Click the Button and wait for the pop-ups

1. In the YT1s to download a video, you need to copy and paste the link you need to install an extension which will display unwanted ads. To download and convert at the same time requires pro version {especially those with both the free and pro versions}.

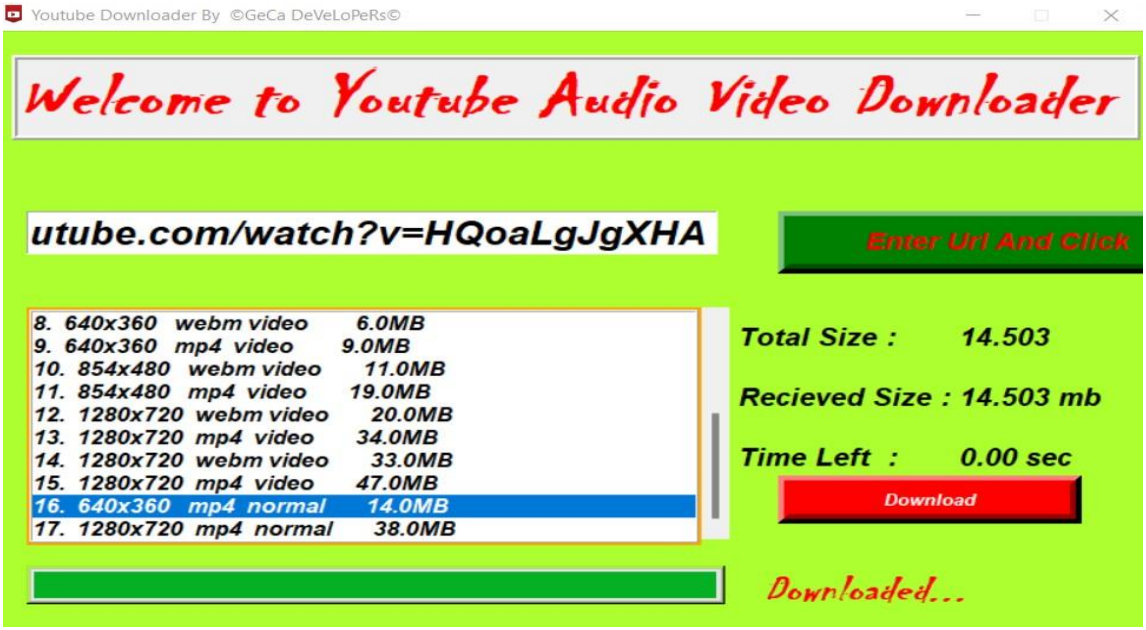


Figure 7. YTD Video Downloaded

1. click on the download button, it will start downloading the video. When video downloading finishes, it shows a message 'downloaded' popup on the window below the download button

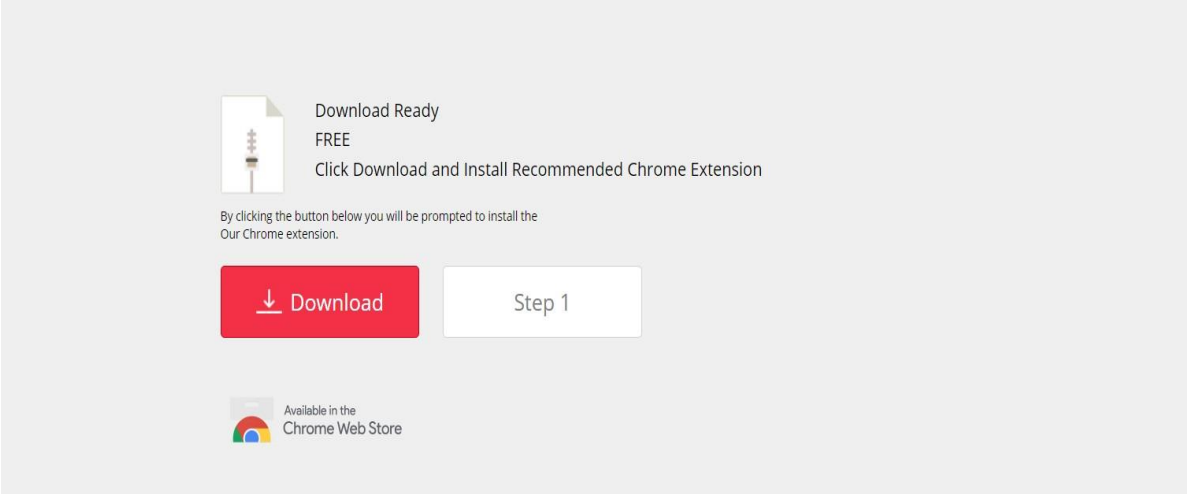


Figure 8. YT1s Showing the Download third party application pop-up

2. In YT1s that showing third party app. Ask you to download other software's and toolbars (I included this because there are some people that are so hasty, they would just click next and next without knowing that they are installing something). Without downloading the other software, you can't download the video.

1.3 Purpose of YouTube Video Downloader:

YouTube is the most-watched video streaming website. Launched in 2005, the online video platform features a wide variety of content, including TV clips, educational videos, music & gaming videos, entertainment stuff, and much more. A slow internet connection can create problems while streaming videos online. With a YouTube video downloader app, you can download videos on your PC or smartphone. This way, you can watch videos at a suitable time without that nagging buffering icon.

Users can enjoy the music or videos offline through video downloaders. The internet has thousands which provide services differently. The downloader tool will assist you to listen offline, thus saving on data and battery. Once you get a proper video downloader, you can have a batch of movies and music ago. YouTube video downloader is a video downloader which provides the facility to download the YouTube videos absolutely free. It has an engaging user interface that categorized popular websites and videos. It offers you to search YouTube videos from its search bar directly.

To download the YouTube videos using YouTube video downloader, follow the below steps:

- Launch the YouTube video downloader application and select the YouTube mobile site from its homepage.
- Search for the video which you want to download in its search bar.
- When the list of videos appears and you find the right video, then select it and click on the red download button.
- Select the quality of video, and your video gets starts downloading.

1.4 Scope:

1. In Future We are adding some new features you can download the 4K,8K, MKV, ETC. Videos using the YouTube video downloader.
2. Changing the User Interface that anyone can easily understood the Software (ease to use UI).

3. We are updating the software then you can download the playlist of any channel.
4. Upgrading the downloading speed using new libraries.

2. LITERATURE SURVEY

2.1 Functional Requirement:

A. def Video Url (): Resets or clears the List Box when new YouTube URL is pasted into the URL textbox and It also resets all the label values.

B. def get video (): Get video details (streams) into the list box using pafy library. A video description is **a piece of metadata that helps YouTube understand the content of a video**. Descriptions that are well optimized can lead to higher rankings in YouTube search. Along with your video title and tags, YouTube uses your description to understand the content (and context) of your video content.

C. def Select Cursor (evt): Connecting list box with Mouse Click. Basically the Select Cursor enables the connection between user's mouse click and the quality selected within the List Box.

D. def Download Video (): Start downloading video streams from the selected quality of the user's choice. It also enables the Downloading thread.

E. def DownloadVideoData (): Downloading Streams and converting file size into MBs and enabling a file dialog to download video to browser to a specific location user want to.

F. def ChangeIntroLabelColor (): Changes the color of Intro Label for every 20 second from the random colors to make the Intro Label fancier to look.

2.2 Feasibility study:

After doing the project You Tube Audio Video downloader study and analyzing all the required functionalities of the system, the next task is to do that the feasibility study for the project. All project is feasible – given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be that future changes can be easily done based on the future upcoming requirements.

2.3 Operational Feasibility:

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system.

After doing the project YouTube Audio Video Downloader, study and analyzing all the required functionalities of the system, the next task is to do is the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be that future changes can be easily done based on the future upcoming requirements.

2.4 Operating Environment:

Hardware Requirements:

- 2.00 GHZ Processor And Above
- 512MB RAM And Above
- Operating System :- WINDOWS 7 , 8 And 10.
- 1 TB Hard Disk
- 64 Bit Operating System

Development Tool:

- Python Idle Version 3.9.6
- PyCharm Community 2021

2.5 Detailed Description About Python and PyCharm IDE:

1.Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form withoutcharge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

2. PyCharm:

In PyCharm, a library is a file or a set of files whose functions and methods are added to PyCharm's internal knowledge in addition to the functions and methods that PyCharm retrieves from the project code that you edit. In the scope of a project, its libraries by default are write-protected.

PyCharm reserves two predefined auto-generated library folders:

- node_modules for keeping Node.js packages listed in the dependencies object of your project package.json. See Configuring node_modules library for details.

- External Libraries for storing downloaded TypeScript definition files or libraries referenced via CDN links as well as Node.js Core nodules or any custom third-party libraries.

Figure 9. PyCharm Auto-generated Library Folders

PyCharm uses libraries only to enhance coding assistance (that is, code completion, syntax highlighting, navigation, and documentation lookup). Please note that a library is not a way to manageyour project dependencies.

Using TypeScript community stubs (TypeScript definition files)

TypeScript community stubs are also known as TypeScript definition files, or TypeScript declaration files, or Definitely Typed stubs, or just **d.ts** files.

In PyCharm, DefinitelyTyped_stubs can be configured and used as libraries, which is in particular helpful in the following cases:

- To improve code completion, resolve symbols for a library or a framework that is too sophisticated for PyCharm static analysis, and add type information for such symbols.

- To resolve globally defined symbols from test frameworks.

The example below shows a piece of code from an Express application where the post () function is not resolved:

Figure 10. Code Snippet where post () is not resolved

PyCharm successfully resolves post () after you install the suggested TypeScript definition file:

Figure 11. Code Snippet where post () is resolved

PyCharm lets you download TypeScript definition files right from the editor, using an intention action,or you can do it on the **Settings: JavaScript Libraries** page.

Download TypeScript definitions using an intention action:

- Position the caret at the require statement with this library or framework, press Alt+Enter, and choose **Install TypeScript definitions for better type information**:

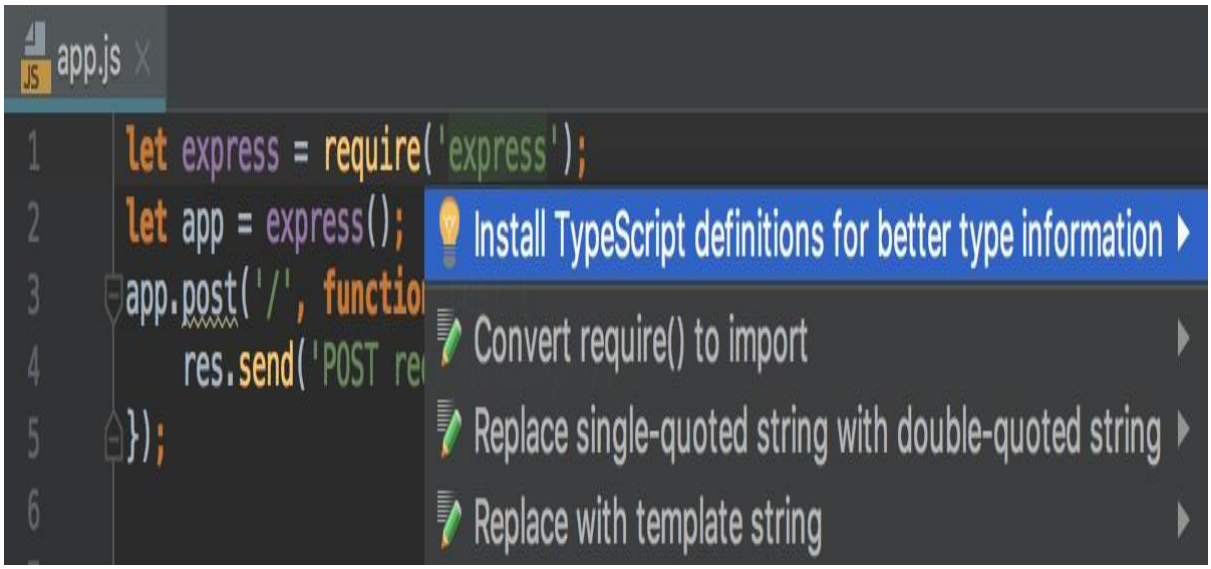


Figure 12. Downloading TypeScript definitions for the library

PyCharm downloads the type definitions for the library and adds them to the list of libraries on the JavaScript. Libraries page:

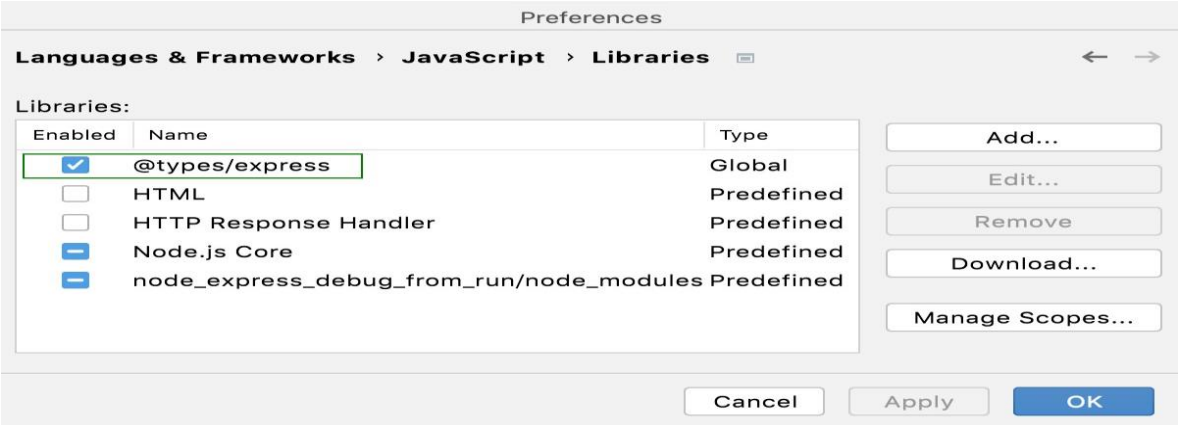


Figure 13. Adding TypeScript definitions to the library list

3. MODELLING/DEVELOPMENT OF SYSTEM

3.1 Explanation of Flow Chart:

The YouTube downloader project is a python project. The object of this project is to download any type of video in a fast and easy way from YouTube in your device.

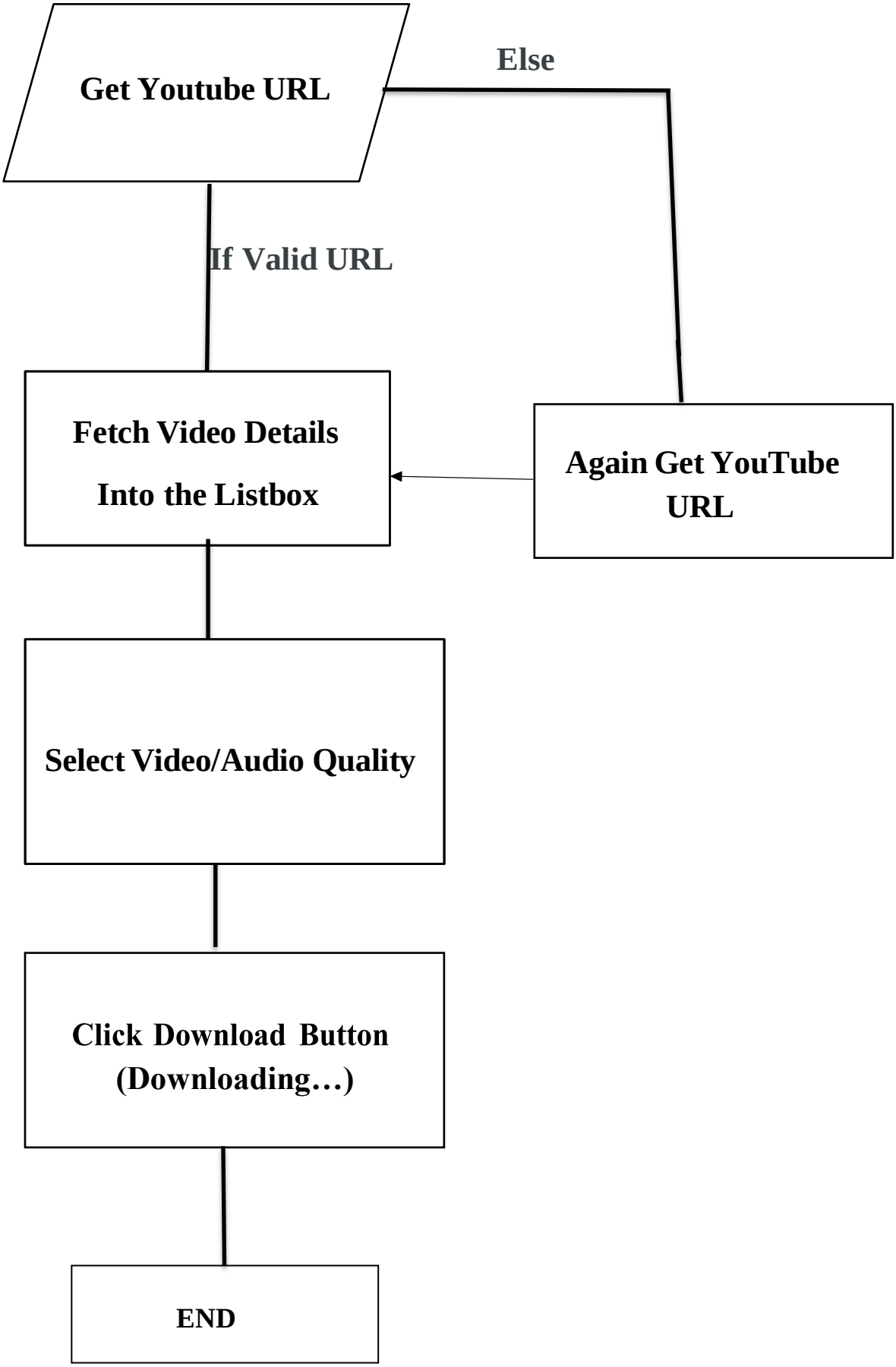
In this python project, user has to copy the YouTube video URL that they want to download and simply paste that URL in the **'Enter URL and click here'** section and click on the download button, it will start downloading the video. When video downloading finishes, it shows a message 'downloaded' popup on the window below the download button.

These are the following steps to build YouTube audio video downloader project in python:

- Import libraries
- Create display window
- Create field to enter link
- Create function to start downloading

These are the following steps to work with **YouTube audio video downloader** project:

1] Start The YouTube Downloader.
2] Get YouTube URL from YouTube.
3] Select the Audio/Video Quality.
4] Click Download Button (Downloading….).
5] Video Downloaded.


Get Youtube URL
Else
If Valid URL
Fetch Video Details
Into the Listbox
Again Get YouTube
URL
Select Video/Audio Quality
Click Download Button
(Downloading…)
END

3.2 System performance analysis:

3.2.1 YouTube Downloader GUI Design:

Before starting of actual working of YouTube downloader we need to know how it is design. We use Tkinter library to design frontend of YouTube downloader. YouTube is a very popular video-sharing website. Downloading a video's/playlist from YouTube is a tedious task. Downloading that video through Downloader or trying to download it from a random website increases the risk of licking your personal data. Using the Python Tkinter package, this task is very simple-efficient-safe. Few bunch codes will download the video for you. For this, there are two Python libraries – Tkinter and pafy.

Create a display Window:

To create a display window, we need take Label, Button, text, font etc.

- **Label ()** widget use to display text that users can't able to modify.
- **root** is the name of the window
- **text** which we display the title of the label
- **font** in which our text is written
- **pack** organized widget in block
- **link** is a string type variable that stores the YouTube video link that the user enters.
- **Entry ()** widget is used when we want to create an input text field.
- **width** sets the width of entry widget

Following is the look of YouTube Downloader that is design by above all Label and Features.

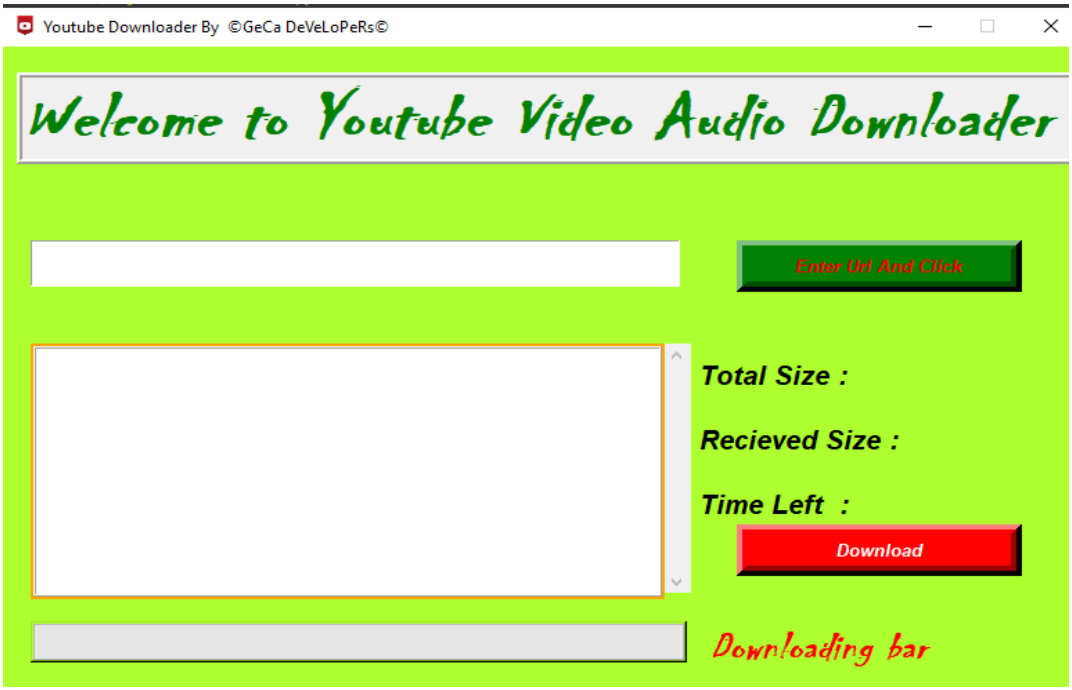


Figure 14. Get YouTube URL and Click the Enter Url and Click Button

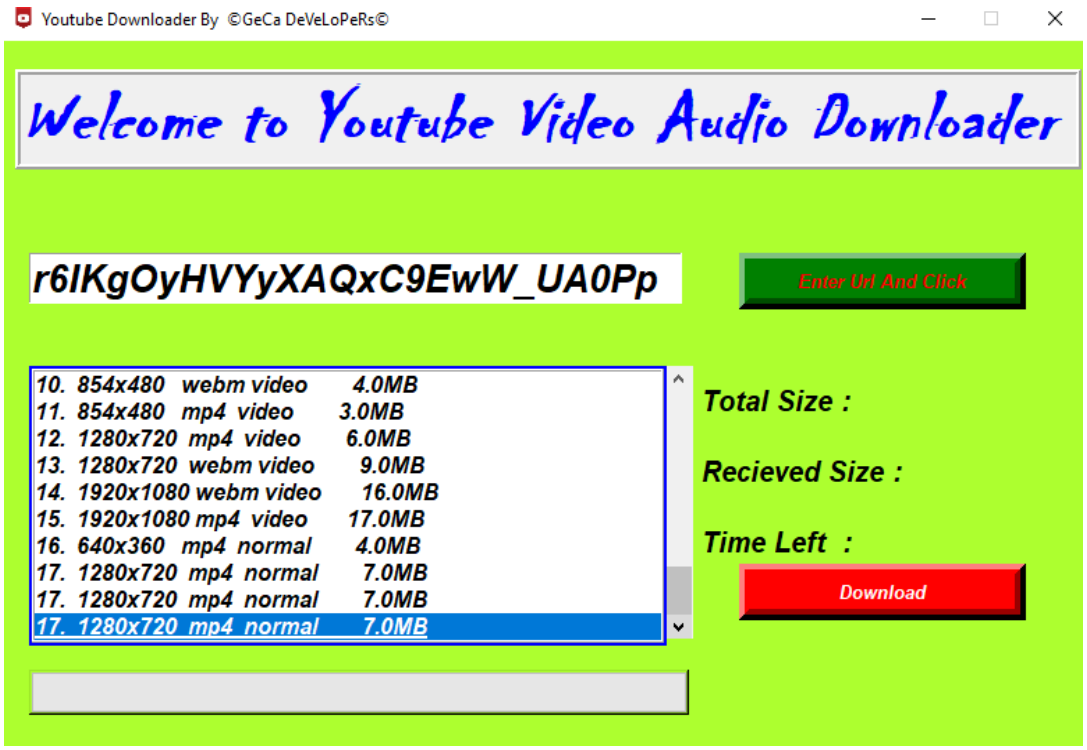


Figure 15. List of Audio/Video Streams are displayed inside the List Box

Get URL from YouTube and click the Enter URL here Button then we see the number of size and Quality of video. We can choose only Audio, only Video or normal as per our requirement that we want. In above picture we select normal they are useful to download audio and video. After selecting size click to download button. With this project in python, we have successfully developed the YouTube video downloader project using python. We used the popular Tkinter library that used for

rendering graphics. We use the pafy library to download video details from YouTube URL.

Video Downloading….:

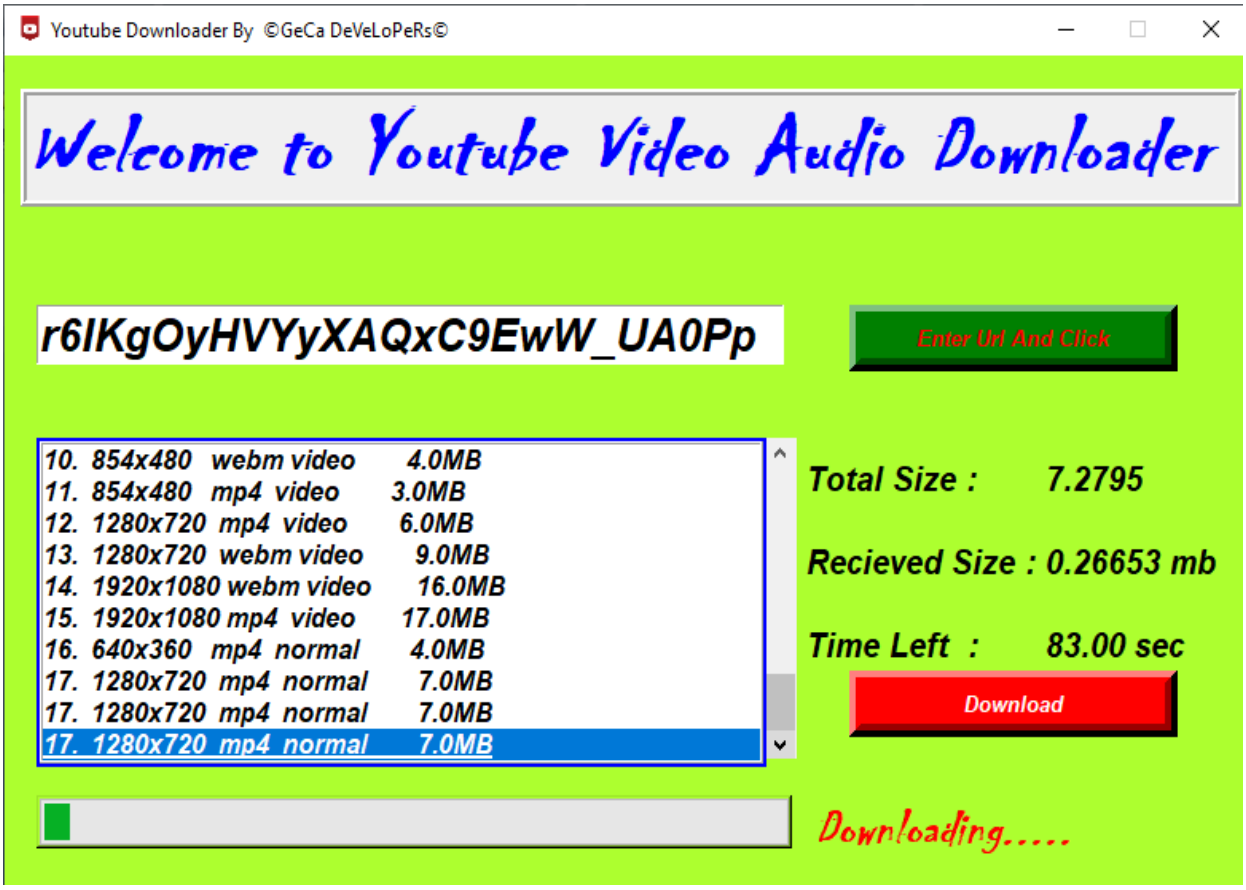


Figure 16. The Video Has Begun to Download in 720p quality

Video Downloaded….:

Figure 17. The Video has been finished downloading

Each and every day, you must be watching video or another YouTube which is related to Music, Movies, Studies, lecture etc. you might want to store some video for future use where it will be necessary , due to lack of internet saving data or any other reason you are unable to do it so that why YouTube audio video downloader is helpful and it is ad free ones.

3.3 Test Procedure and Implementation:

Test Procedure

The document that describes the steps to be taken in running a set of tests and specifies the executable order of the tests is called a test procedure, and is also known as a test script. When test procedure specification is prepared then it is implemented and is called Test implementation. Test script is also used to describe the instructions to a test execution tool.

We are tested to be manually rather than using a test execution tool. In every Label we tested them as Try and error basis to how it make perfect. In design of YouTube downloader, we apply colors that are suitable for them. In the process of design, we run them after adding every tool and fixed it properly. Suppose we paste

more than one URL then it will not work.

3.4 System Testing:

System testing is concerned with testing the entire system, where I will use the Release testing which is concerned that system meets its requirements, and to ensure that system is dependable. Release testing is usually “black box” testing where the system is too tested on working performance properly or not.

1. White Box Testing

White box testing is to test the system internal structure which is also known as glass box testing or structural testing. Without acknowledgement of program arrangement, white box testing is applied to test the code of the program to examine its output whether is appropriate. Testing will be performed only if knowing what the programmer is supposed to do. Tester will see if the program diverges from its intended goal. White box testing does not account for errors caused by omission, and all visible code must also be readable Since the system will be designed, implemented and tested all by author, the code testing has initially been done in the constructing the system process so it is not necessary to carry out the white box testing to examine the program code again.

2. Black Box Testing

Black box testing technique is used to select test cases while as white box testing is mainly applied to determine whether two objects resulting from the program execution of a test case are observationally equivalent. My universal mail client, the input is from the keyboard and output is Computer’s display. To evaluate the entire system, every class will be checked by using black box strategy. The function in these classes will be tested as a unit examines to see whether it meets the desired functionality. Black box testing behavior is determined by studying the input and related outputs. The following section is adopted this method to test the system. Input and Output Accordingly test area will comprise all the functions in the system, the input values are specified as requirements and output will be system displays.

Some guidelines of designing the input and output are:

1. Design inputs that force the system to generate all error messages.
2. Design inputs that cause input buffer to flow.
3. Repeat the same input or a series of input numerous times.
4. Force invalid outputs to be generated.

5. Force computation results to be too large or too small.

Consequentially I design the input values based on these selective guidelines and the detailed inputs and system outputs will be put in Appendix Two.

3.5 Test Cases:

Table 1. Test Cases of YouTube audio video downloader

Test Caseid	Objective	Steps	Expected Result	Actual Result	Status
T001	Testing for IntroLabel	Open the YouTube Downloader	Flashing of the IntroLabel Color.	Flashing of the IntroLabel Color.	Pass
T002	Testing for TexturlLabel	Paste the url	Show the url in italic bold on the label.	Show the url in italic bold on thelabel.	Pass
T003	Testing for TexturlLabel	Paste the invalid url	No Output Expected in the List Box	No Output Expected in the List Box	Pass
T004	Testing for Enter URL here Button	Enter URl and Click.	Show the size and Quality of Video/Audio	Show the size and Quality of Video/Audio	Pass
T005	Testing for Choose the Size from listbox.	Select size of video.	Select the actual size There is a cursor.	Select the actual sizeThere is a cursor.	Pass
T006	Testing for Download Button	Click on A Download Button	Start downloading video	Start downloading video	Pass
T007	Testing for Downloading SizeLabelResult	Click the Download Button.	After clicking Download Button it shows the total sizeof video.	After clicking Download Button itshows the total sizeof video.	Pass
T008	Testing for Downloading LabelTimeleftResult	Click the Download Button.	After clicking Download Button it shows the total timeleft.	After clicking Download Button it shows the total Timeleft.	Pass

T009	Testing for Downloading ProgressbarLabel	Enter the Download Button.	After clicking Download Button it shows the progress bar working.	After clicking Download Button itshows the progress bar working.	Pass
T010	Testing for Download Button	Enter the Download Button	It ask to where to store video in our device and get proper path to savevideo.	It ask to where to store video in our device and get proper path to savevideo.	Pass
T011	Testing for Window resizable.	Open the YouTube downloader.	It cannot minimize or maximize thesize of the window	It cannot minimize or maximize the size of the window	Pass
T012	Testing for connection of Scrollbar with the listbox.	Scroll the Scrollbar.	After scrolling the scrollbar it automatically scroll down the content inside the listbox.	After scrolling the scrollbar it automatically scroll down the content inside the listbox.	Pass

4. IMPLEMENTATION

System implementation projects are long difficult journeys by which organizations move from an old set of technology/methods/procedures to a new one. A software implementation method is a systematic structured approach to effectively integrate software based service or component into the workflow of an organizational structure or an individual end-user. The complexity of implementing product software differs on several issues. Examples are: the number of end users that will use the product software, the effects that the implementation has on changes of tasks and responsibilities for the end user, the culture and the integrity of the organization where the software is going to be used and the budget available. It is vital to select the right strategy for implementing the application to assure successful results.

4.1 Implementation Strategy:

Since the software application consists of three modules as per in the high level architectural design, the implementation was done using iterative, incremental approach. Phase wise implementation process enables to execute by incrementally aligning the product with the end-user. Phase wise implementation process is shown in Figure 18.

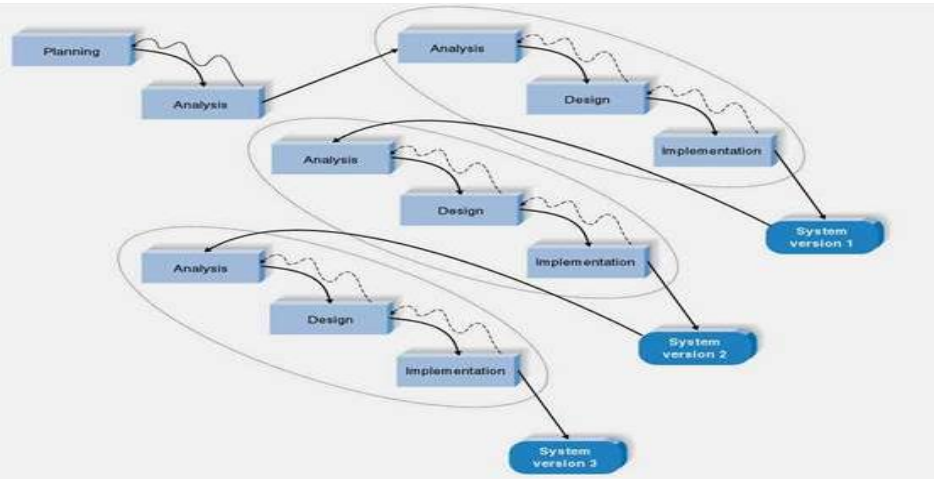


Figure 18. Phase Implementation Process

Comparison of the ways that can be used in implementing or introducing a new systemis shown in below Table 2.

Table 2. Comparison of Software Implementation Methods

Method	Main Advantages	Main Disadvantages
1. Immediate cutover Straight from old system to new system on a single date	Rapid, low cost	High risk. Major disruption if serious errors with the system
2. Parallel running Old system and new system run side-by-side for a period	Low risk than immediate cutover	Slower and higher cost than immediate cutover
3. Phased Implementation Different modules of the system are introduced sequentially	Good compromise between methods 1 and 2	Difficult to achieve due to interdependencies between modules
4. Pilot System Trial implementation occurs before widespread deployment		Has to be used in combination with the other methods.

4.2 Coding:

1.Designing the GUI

```
from tkinter import *

from tkinter.ttk import Progressbar

from tkinter import PhotoImage

from tkinter import filedialog

import random

import threading # to keep running downloadvideo()

import pafy #downloading youtube video details

root = Tk()

root.title('Youtube Downloader By ' " ©GeCa DeVeLoPeRs©")
```

```
root.configure(bg='#ADFF2F')

root.geometry('780x500')

root.resizable(False, False)

root.attributes()

photo=PhotoImage(file=("C:\\Users\\sghod\PycharmProjects\\YoutubeDownloader\\Youtube1.png"))

root.iconphoto(False,photo)
```

Figure 19. Designing the GUI Interface

2. Adding Labels & Scrollbar and configuring it

```
# scrollbar

scrollbar = Scrollbar(root)

scrollbar.place(x=477, y=230, height=193, width=20)

# Youtube Url text
```

```
urltext = StringVar()

UrlEntry = Entry(root, textvariable=urltext, font=('arial', 20, 'italic bold'), width=31)

UrlEntry.place(x=20, y=150)

# Labels

IntroLabel = Label(root, text='Welcome to Youtube Audio Video Downloader ', width=36, relief='ridge', bd=4,font=('chiller', 40, 'italic bold'), fg='red')

IntroLabel.place(x=10, y=20)

ChangeIntroLabelColor()

ListBox = Listbox(root, yscrollcommand=scrollbar.set, width=50, height=10, font=('arial', 12, 'italic bold'),relief='ridge', bd=2, highlightcolor="blue", highlightbackground="orange", highlightthickness=2)

ListBox.place(x=20, y=230)

ListBox.bind("<<ListboxSelect>>", SelectCursor)

scrollbar.configure(command=ListBox.yview)
```

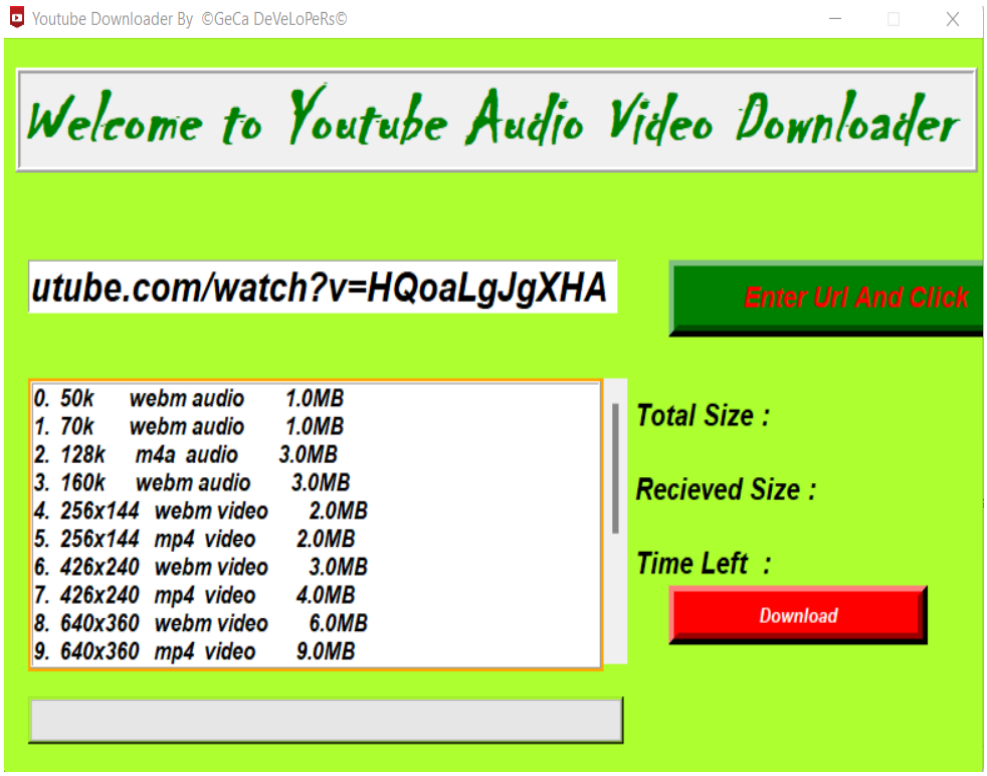


Figure 20. Designing of Scrollbar and other Labels

```
# Downloading Size label

DownloadnigSizeLabel = Label(root, text='Total Size : ', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigSizeLabel.place(x=500, y=240)

DownloadnigLabel = Label(root, text='Recieved Size : ', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigLabel.place(x=500, y=290)

DownloadnigTime = Label(root, text='Time Left  : ', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigTime.place(x=500, y=340)

DownloadnigSizeLabelResult = Label(root, text='', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigSizeLabelResult.place(x=650, y=240)

DownloadnigLabelResult = Label(root, text='', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigLabelResult.place(x=650, y=290)

DownloadnigLabelTimeLeft = Label(root, text='', font=('arial', 15, 'italic bold'), bg='#ADFF2F')

DownloadnigLabelTimeLeft.place(x=650, y=340)

DownloadingBarTextLable = Label(root, text='Downloading bar', width=36, font=('chiller', 23, 'italic bold'), fg='red',bg='#ADFF2F')

DownloadingBarTextLable.place(x=370, y=445)

DownloadingProgressBarLabel = Label(root, text='', width=36, font=('chiller', 40, 'italic bold'), fg='red', bg='#ADFF2F', relief='raised')

DownloadingProgressBarLabel.place(x=20, y=445)
```

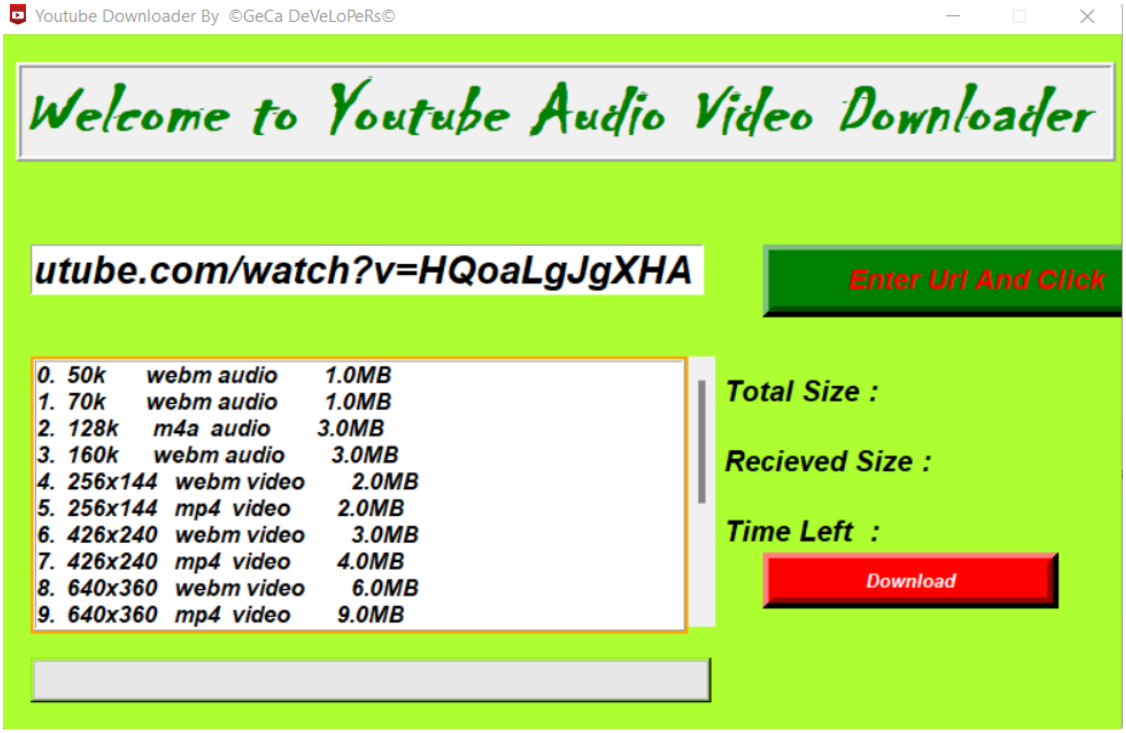


Figure 21. Designing of Downloading Size Labels

3.Adding Buttons and Progress Bar

Progress Bar

DownloadnigProgressBar = Progressbar(DownloadingProgressBarLabel, orient=HORIZONTAL, value=0, length=100, maximum= total12)

DownloadnigProgressBar.grid(row=0, column=0, ipadx=185, ipady=3)

Buttons

ClickButton = Button(root, text='Enter Url And Click', font=('Arial', 14, 'italic bold'), bg='green', fg='red',activebackground='blue', width=23, bd=8, command=VideoUrl)

ClickButton.place(x=530, y=150)

DownloadButton = Button(root, text='Download', font=('Arial', 10, 'italic bold'), bg='red', fg='white',activebackground='blue', width=23, bd=8,command=DownloadVideo)

DownloadButton.place(x=530, y=370)

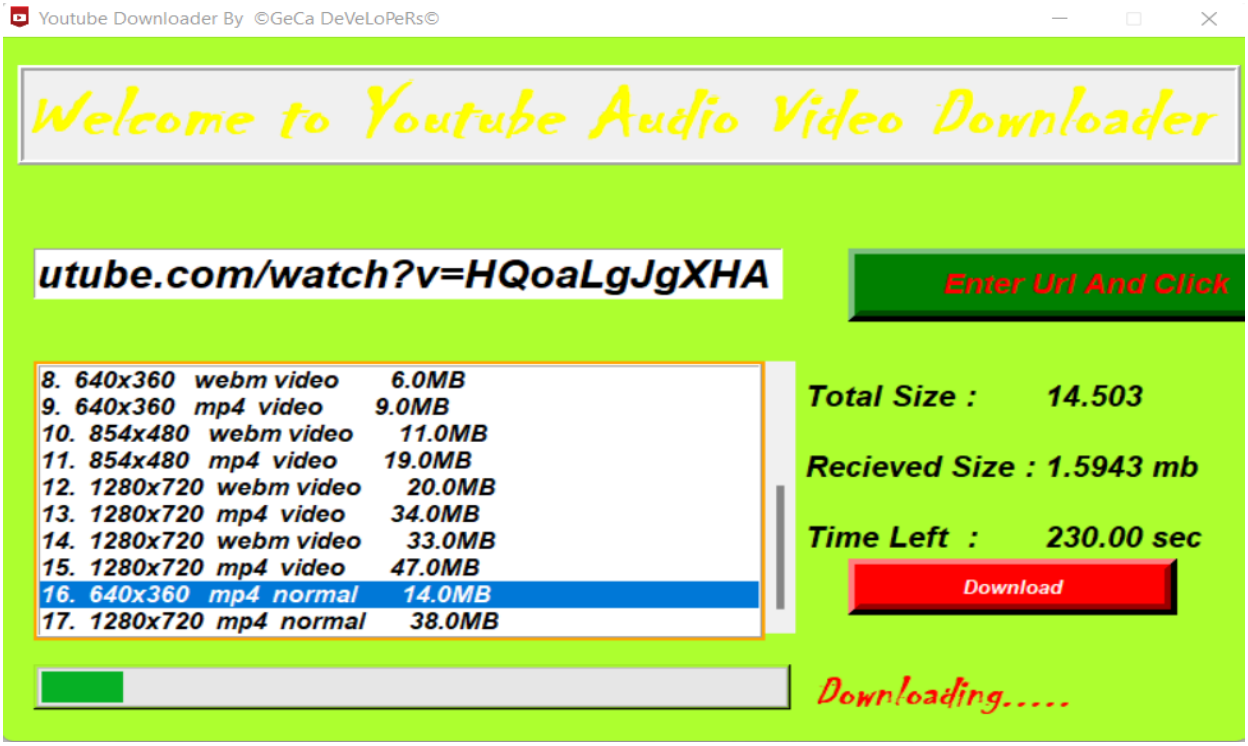


Figure 22. Adding Enter Url and Click and Download Buttons with Progress Bar

4.Adding ChangeIntroLabel() Function

```
colors = ['red', 'green', 'blue', 'yellow', 'gold', 'pink']

def ChangeIntroLabelColor():

    ss = random.choice(colors)

    IntroLabel.configure(fg=ss)

    IntroLabel.after(20, ChangeIntroLabelColor)
```

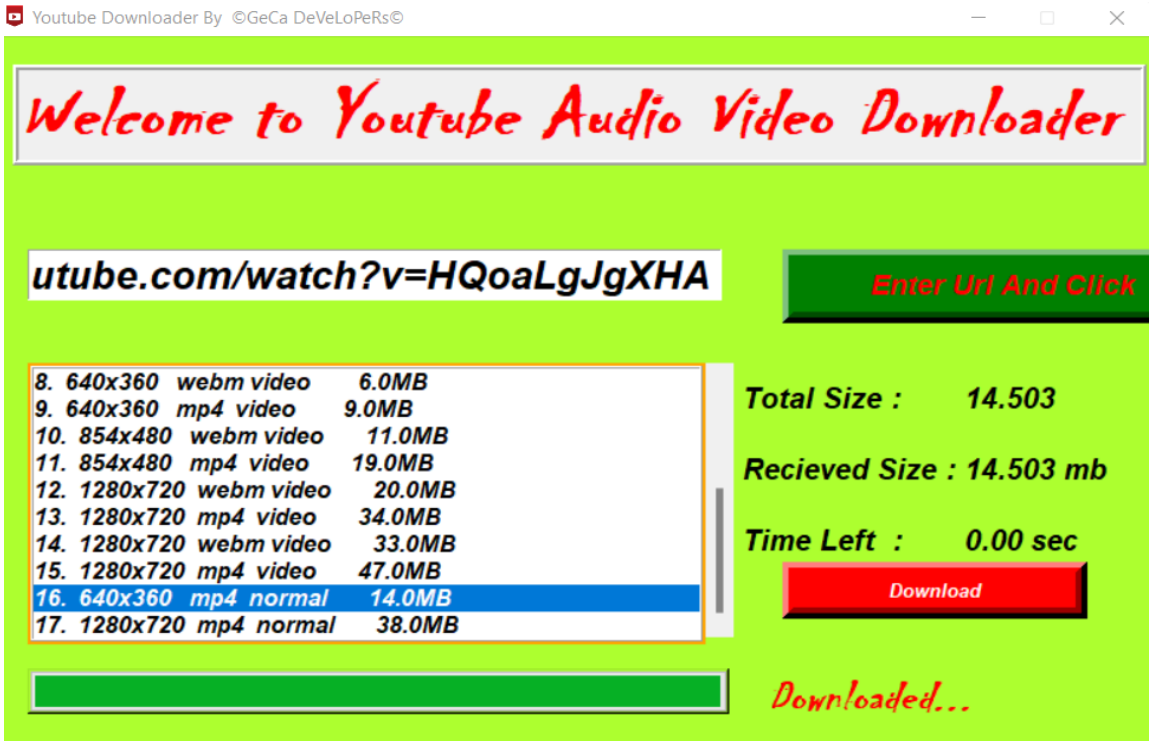


Figure 23. Adding ChangeIntroLabel() to make introlabel more fancier to look

5. Getting Video Details Using Pafy Library from the fetched URL

```
def VideoUrl():
    DownloadingBarTextLable.configure(text="")
    DownloadnigLabelResult.configure(text="")
    DownloadnigSizeLabelResult.configure(text="")
    DownloadnigLabelTimeLeft.configure(text="")
    getdetail = threading.Thread(target=getvideo)
    getdetail.start()
def getvideo():
    global streams
    ListBox.delete(0, END)
    url = urltext.get()
    data = pafy.new(url)
```

```
    streams = data.allstreams #getting all types of streams to download
    index = 0
    for i in streams:
        du = '{:0.1f}'.format(i.get_filesize()//(1024*1024))
        datas = str(index) + '.'.ljust(3, ' ') + str(i.quality).ljust(10, ' ') + str(i.extension).ljust(5, ' ') +
str(i.mediatype) + ' ' + du.rjust(10, ' ') + "MB"
        ListBox.insert(END, datas)
        index += 1
```

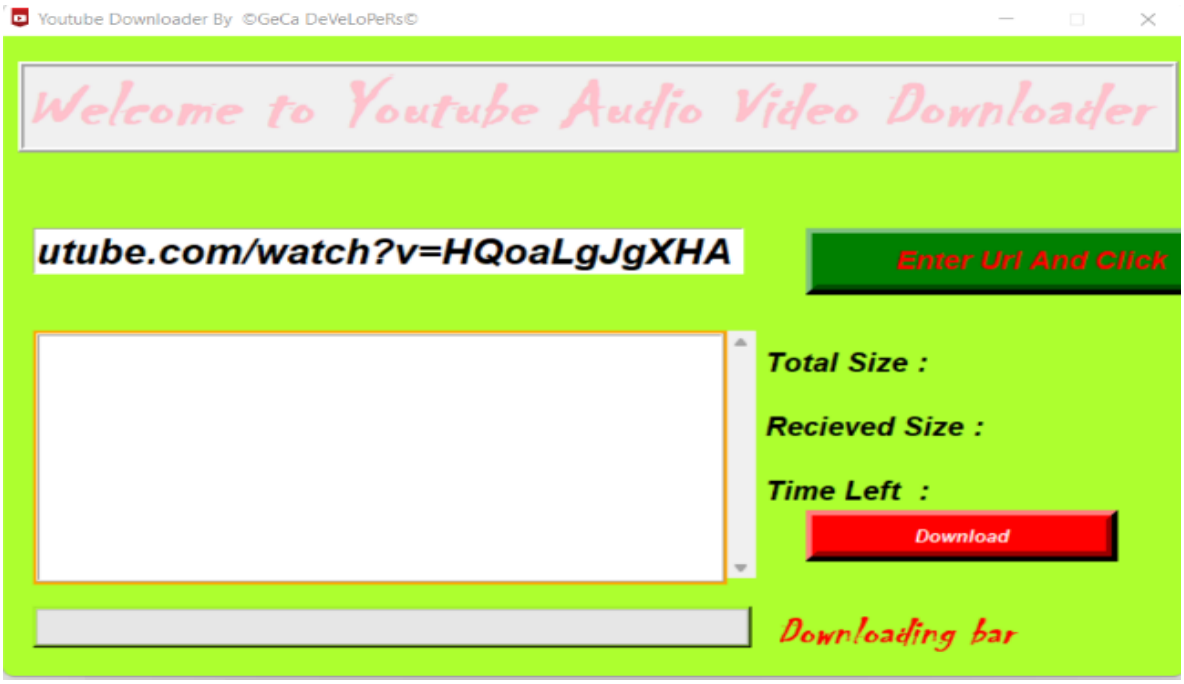


Figure 24. Resetting all values when new link is inserted

6. Binding the Mouse Click Event with ListBox and adding Threads to keep the program running

```
def SelectCursor(evt):
    global downloadindex
    listboxdata = ListBox.get(ListBox.curselection())
    print(listboxdata)
```

```
    downloadstream = listboxdata[:3]
    downloadindex = int(''.join(x for x in downloadstream if x.isdigit()))
def DownloadVideo():
    getdata = threading.Thread(target=DownloadVideoData)
    getdata.start()
```

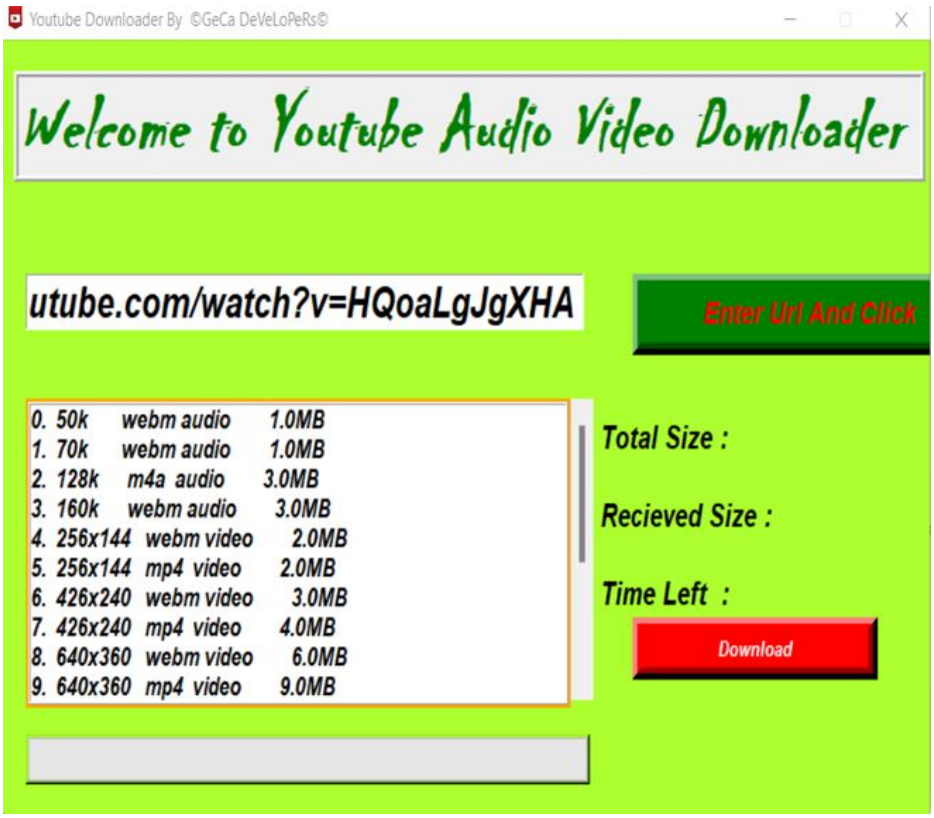


Figure 25. Binding Mouse Click with the List Box

7. Adding FileDialog and Changing Label Values

```
def DownloadVideoData():
    global downloadindex
    fgr = filedialog.askdirectory()
    DownloadingBarTextLable.configure(text="Downloading.....")
    def mycallback(total, recvd, ratio, rate, eta):
        global total12
```

```
    total12 = float('{:.5}'.format(total/(1024*1024)))
    DownloadnigProgressBar.configure(maximum=total12)
    recieved1 = '{:.5} mb'.format(recvd / (1024 * 1024))
    eta1 = '{:.2f} sec'.format(eta)
    DownloadnigSizeLabelResult.configure(text=total12)
    DownloadnigLabelResult.configure(text=recieved1)
    DownloadnigLabelTimeLeft.configure(text=eta1)
    DownloadnigProgressBar['value'] = recvd/(1024*1024)
streams[downloadindex].download(filepath=fgr, quiet=True, callback=mycallback)
DownloadingBarTextLable.configure(text="Downloaded...")
```

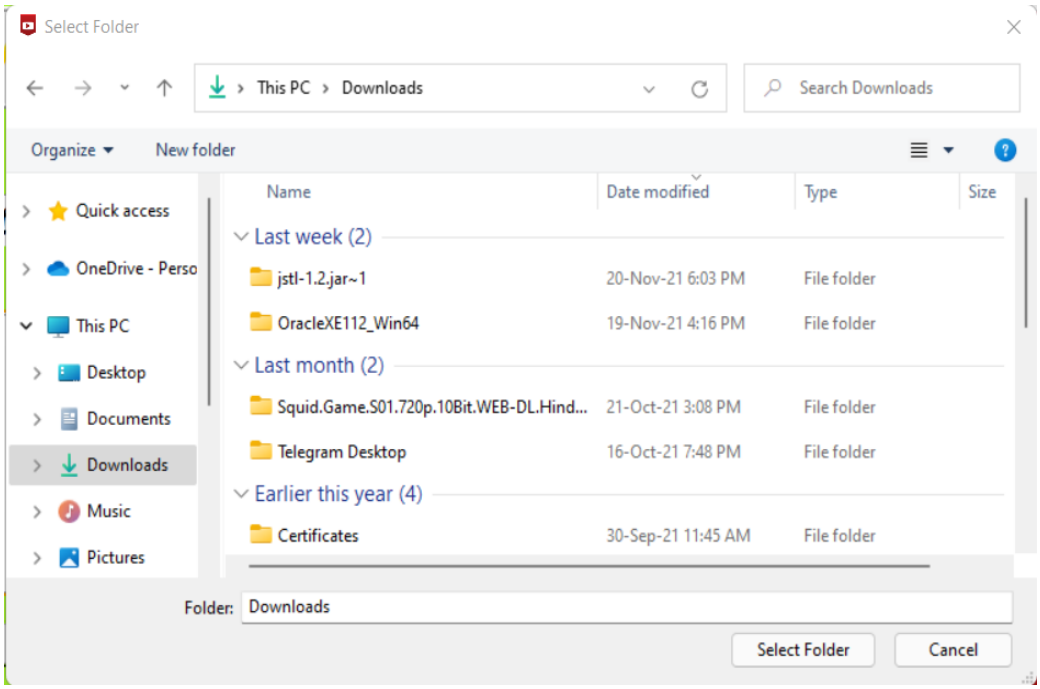


Figure 26. Adding File Dialog Control to choose Download Location

4.3 Drawbacks and Limitations:

Although we have put our best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by us. Though the software presents a broad range of options to its users some intricate options could not be covered into it; partly because of logistic & partly due to lack of sophistication. Lack of time also compelled us to ignore some part such as downloading playlists and parallel download support etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance.

List of Limitations:

- No Playlist Download Supported
- No Parallel Download Support
- Downloading Speed meter yet to design
- Help Desk yet to be implemented

4.4 Future Scope:

It is an easy way to download YouTube Videos/Audios by just pasting YouTube URLs and downloading as per user's choice the quality he/she wants. Well I and my team member have worked hard in order to present an improved application better than the existing one's regarding the YouTube Downloader. Still, we found out that the project can be done in a better way. Primarily, in this system user copies and paste the links from YouTube website or app and can download videos and audios.

The next enhancement is we will develop Download Speed meter and Parallel Download Support as well as Playlist Download Support for the enhanced version. So that, users can download multiple videos and whole playlists at once and will be able to check their downloading speeds.

5.CONCLUSION

YouTube Audio Video Downloader is not unlike many other project implementation, control and termination. The Project Implementation Phase captures YouTube URLs and fetched video details using Pafy Library and inserts the streams into the List Box and then user downloads the video /audio according to his/her choice of resolution quality.

The major objective that have been decide in initial phase of requirement analysis are achieved successfully. After implementation the result provides reliable results. The system istotally user friendly, which make it easy for the user with limited knowledge of windows environment to operate the developed system. We have tried our level best to make the site asdynamic as possible.

BIBLIOGRAPHY/REFERENCES

The following are the details of books and the sites which have details regarding the Python, Tkinter, Pafy, youtube-dl, pyinstaller etc.

Reference Books:

- "Python: The Complete Reference", Author: Martin Brown.
- "Python GUI Programming - A Complete Reference Guide: Develop Responsive and Powerful GUI Applications with PyQt and Tkinter", Author: Alan D. Moore and B. M. Harwani.

Web References:

- *https://docs.python.org/3/*
- *https://www.w3schools.com/python/*
- *https://www.youtube.com/watch?v=Q4lm8eYulw&list=PLu0W_9lII9ajLcqRcj4PoEihkukF_OTzA*
- *https://www.youtube.com/watch?v=GhzpvvIXlM&list=PLS1QulWo1RIY6fmY_iTjEhCMsdtAjgbZM*
- *https://docs.python.org/3/library/tkinter.html*
- *https://pythonhosted.org/pafy/*
- *https://pypi.org/project/youtube_dl/*
- *https://pypi.org/project/pyinstaller/*