

YouTube

VIDEO DOWNLOADER

Project Synopsis

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE
DEGREE OF BACHELOR OF TECHNOLOGY
IN
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

By

Name	Registration Number	Roll Number
Arnab Kumar Pati	202410100110005 of 2020-21	24100120039
Deeptangshu Dutta	202410100110016 of 2020-21	24100120028
Jit Kumar Naskar	202410100110020 of 2020-21	24100120024
Snehashis Bhowmik	202410100110008 of 2020-21	24100120036
Soumi Das	202410100110013 of 2020-21	24100120031

Under The Supervision of

Prof. Sudipta Hazra, Assistant Professor
Prof. Avishek Nath, Assistant Professor



**Swami Vivekananda Institute of
Science & Technology**

Department of Computer Science & Engineering
Dakshin Gobindapur. P.S.: Sonarpur, Kolkata – 700 145.

Affiliated by



**Maulana Abul Kalam Azad University
of Technology, West Bengal**
(Formerly West Bengal University of Technology)

Session 2021-22

Swami Vivekananda Institute of Science & Technology

Dakshin Gobindapur. P.S.: Sonarpur, Kolkata – 700 145.



Certificate of Approval By Supervisor

The foregoing project entitled **YouTube Video Downloader** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the degree for which it has been submitted. It is to be understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Name & Designation of Project Guide : Prof. Sudipta Hazra, Assistant Professor
Prof. Avishek Nath, Assistant Professor

Signature of Project Guide :

Date : ____ / ____ / ____

Place :

[Departmental Seal]

Acknowledgment

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Prof. Sudipta Hazra for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of Swami Vivekananda Institute of Science & Technology for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my team in developing the project and people who have willingly helped me out with their abilities.

Table of Contents

Abstract	5
1. Introduction.....	6
1.1. Purpose	6
1.2. Conventions Used.....	6
2. Requirement.....	7
2.1. Functional requirement	7
2.1.1. Minimum System requirements.....	7
2.2. Non-functional requirement	7
3. Project Estimation Techniques.....	8
3.1. Basic COCOMO Model.....	10
3.2. Intermediate COCOMO Model.....	11
3.3. Implications.....	13
3.3.1. Advantages	13
3.3.2. Drawbacks	13
4. Design of the system.....	14
4.1. Use case Diagram	14
4.2. Data Flow Diagram.....	14
4.2.1. Level – 0.....	14
4.2.2. Level – 1.....	15
4.3. UML Class Diagrams.....	15
5. Testing.....	16
6. User manual	24
6.1. Installation Manual	24
6.2. Maintenance Manual.....	26
7. Conclusion	29
7.1. Limitation	29
7.2. Future Scope.....	29
8. Bibliography.....	30

Abstract

YouTube has grown to become the largest and most highly visited online video-sharing service, and interest in the educational use of YouTube has become apparent. But downloading videos from YouTube is not an option due its policies. Few software programs that can be used to download videos from YouTube and convert them into a different format is referred to as a YouTube video downloader. There are many different types of YouTube video downloaders available, each with its own set of features. Some YouTube video downloaders are quick and simple to use, while others offer a more comprehensive set of features. There are a lot of software in the market today that offer to help you download videos from YouTube. While most of them are effective, they can be quite complicated to use. YouTube Video Downloader is different. It is quick and easy to use, and it comes in a format that is compatible with all devices. This software is also useful for people who want to download videos in a format that can be played on their TV or any other device. YouTube Video Downloader is the best software for those who want to download videos from YouTube in a quick and simple way.

Keywords: 1) *Video Downloader*; 2) *COCOMO*; 3) *Tkinter*; 4) *DFD*; 5) *GUI*; 6) *User Manual*.

1. Introduction

1.1. Purpose

There are many software programs that allow users to download videos from YouTube. Some of these programs have a graphical user interface (GUI), while others do not. Many of these programs are free to use while having In-App Advertisement, while some require a subscription or one-time payment. Some popular YouTube video downloaders include 4K Video Downloader, Any Video Converter, and Free YouTube Download. Any Video Converter is a paid program that supports downloading videos in up to 4K resolution. YouTube Video Downloader is a free program that supports downloading videos in up to 720p resolution.

The YouTube Downloader video GUI is a software that enables you to download videos from the popular video sharing website, YouTube. The software has a simple and straightforward interface, and it is very easy to use. With just a few clicks, you can download any video from YouTube in high quality. The software also supports downloading videos in multiple formats, so you can choose the one that best suits your needs.

1.2. Conventions Used

Python is a versatile language that you can use on the backend, frontend, or full stack of a web application. In this report, we'll focus on the main uses of Python in software development with the Python Tkinter Library.

Tkinter is Python's standard GUI (Graphical User Interface) package. It is open source and available under the Python License. Tkinter is bundled with Python for Windows and Mac OS X platforms. On Linux platforms, it is available via the apt package manager.

With Tkinter, you can create desktop applications with a graphical user interface (GUI). The Tkinter library comes with a number of pre-defined widgets that you can use in your applications: buttons, labels, text boxes, check boxes, radio buttons, etc. You can also create custom widgets using the Tkinter library.

2. Requirement

2.1. Functional requirement

Currently our setup installer is made only for Windows. This software is trial and tested thoroughly with the minimal system configurations.

2.1.1. Minimum System requirements

The following the few basic system requirements that are need to run the application:

- **Available on:** Desktop
- **Memory:** 1 GB
- **Operating System:** Windows 7 or higher
- **Architecture:** x64
- **Processor:** Intel or AMD processor, 1.5 GHz or faster processor
- **Storage Space:** 24.7 MB
- **Internet:** Must have Stable Internet Connection
- **Keyboard:** Integrated keyboard
- **Mouse:** Integrated mouse

2.2. Non-functional requirement

- **Network Speed:** Stable Internet Connection with Minimum speed of 10Mbps
- **Links:** The YouTube Video Link should be Public or Unlisted.
- **Video Type:** Should Not be a Live Streaming Video and should not have Age-Restrictions Enabled.

3. Project Estimation Techniques

A software project is not just about writing a few hundred lines of source code to achieve a particular objective. The scope of a software project is comparatively quite large, and such a project could take several years to complete. However, the phrase "quite large" could only give some (possibly vague) qualitative information. As in any other science and engineering discipline, one would be interested to measure how complex a project is. One of the major activities of the project planning phase, therefore, is to estimate various project parameters in order to take proper decisions. Some important project parameters that are estimated include:

- **Project size** : What would be the size of the code written say, in number of lines, files, modules?
- **Cost** : How much would it cost to develop a software? A software may be just pieces of code, but one has to pay to the managers, developers, and other project personnel.
- **Duration** : How long would it be before the software is delivered to the clients?
- **Effort** : How much effort from the team members would be required to create the software?

COCOMO (Constructive Cost Model) was proposed by Boehm.

According to him, there could be three categories of software projects:

- Organic
- Semidetached
- Embedded

The classification is done considering the characteristics of the software, the development team and environment. These product classes typically correspond to application, utility and system programs, respectively. Data processing programs could be considered as application programs. Compilers, linkers, are examples of utility programs. Operating systems, real-time system programs are examples of system programs. One could easily apprehend that it would take much more time and effort to develop an OS than an attendance management system.

The concept of organic, semidetached, and embedded systems is described below.

Organic			<ul style="list-style-type: none"> ✓ The project deals with developing a well understood application ✓ The development team is small. ✓ The team members have prior experience in working with similar types of projects.
a	b	c	
2.4	1.05	0.38	
Semidetached			<ul style="list-style-type: none"> • The team consists of some experienced as well as inexperienced staff. • Team members may have some experience on the type of system to be developed.
a	b	c	
3.0	1.12	0.3	
Embedded			<ul style="list-style-type: none"> ▪ Aims to develop a software strongly related to machine hardware. ▪ Team size is usually large.
a	b	c	
3.6	1.20	0.32	

Our team fits the **Organic criterion**. So, further, calculations will be done considering $a = 2.4$, $b = 1.05$, $c = 0.38$.

3.1. Basic COCOMO Model

The basic COCOMO model helps to obtain a rough estimate of the project parameters. It estimates effort and time required for development in the following way:

$$\text{Effort} = a \times KLOC^b \text{ PM}$$

$$T_{dev} = (2.5 \times \text{Effort}^c) \text{ Months}$$

where

- $KLOC$ is the estimated size of the software expressed in Kilo Delivered Source Instructions
- a, b, c are constants determined by the category of software project
- Effort denotes the total effort required for the software development, expressed in person months (PMs)
- T_{dev} denotes the estimated time required to develop the software (expressed in months)

So, the projected effort required for this project becomes

$$\begin{aligned}\text{Effort (E)} : \quad \text{Effort}(E) &= a \times KLOC^b \\ &= 2.4 \times 9^{1.05} \\ &= 24\end{aligned}$$

$$\therefore \text{Effort} = 24 \text{ PM (approx.)}$$

So, around 24 person-months are required to complete this project. With this calculated value for effort we can also approximate the development time required:

$$\begin{aligned}T_{dev} : \quad 2.5 * (\text{Effort})^c \text{ months} \\ &= 25 \times 24^{0.38} \\ &= 8.364 = 8.5\end{aligned}$$

$$\therefore T_{dev} = 8.5 \text{ months}$$

So, the project is supposed to be complete by eight and a half months. However, **estimations using Basic COCOMO are largely idealistic.** Let us **refine them using Intermediate COCOMO.**

3.2. Intermediate COCOMO Model

The basic COCOMO model considers that effort and development time depends only on the size of the software. However, in real life there are many other project parameters that influence the development process. The intermediate COCOMO take those other factors into consideration by defining a set of 15 cost drivers (multipliers) as shown in the table below. Thus, any project that makes use of modern programming practices would have lower estimates in terms of effort and cost. Each of the 15 such attributes can be rated on a six-point scale ranging from "very low" to "extra high" in their relative order of importance. Each attribute has an effort multiplier fixed as per the rating. The product of effort multipliers of all the 15 attributes gives the **Effort Adjustment Factor (EAF)**.

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required Software Reliability	0.75	0.88	1.00	1.15	1.40	
Size of Application Database		0.94	1.00	1.08	1.16	
Complexity of the Product	0.70	0.85	1.00	1.15	1.30	1.65

Hardware Attributes						
Runtime Performance Constraints			1.00	1.11	1.30	1.66
Memory Constraints			1.00	1.06	1.21	1.56
Volatility Of The Virtual Machine Environment	0.87		1.00	1.15	1.30	
Required Turnabout Time	0.87		1.00	1.07	1.15	
Personnel Attributes						
Analyst Capability	1.46	1.19	1.00	0.86	0.71	
Applications Experience	1.29	1.13	1.00	0.91	0.82	
Software Engineer Capability	1.42	1.17	1.00	0.86	0.70	
Virtual Machine Experience	1.21	1.10	1.00	0.90		
Programming Language Experience	1.14	1.07	1.00	0.95		
Project Attributes						
Application Of Software Engineering Methods	1.24	1.10	1.00	0.91	0.82	

Use of Software Tools	1.24	1.10	1.00	0.91	0.83	
Required Development Schedule	1.23	1.08	1.00	1.04	1.10	

The cells with yellow backgrounds highlight our choice of weight for each of the cost drivers. EAF is determined by multiplying all the chosen weights. So, we get

$$EAF = 0.703885$$

Using this EAF value we refine our estimates from basic COCOMO as shown below

$$\begin{aligned} \text{Effort}_{\text{corrected}} &= \text{Effort} \times EAF \\ &= 24 \times 0.703885 \\ &= \mathbf{16.89324 \text{ PM}} \end{aligned}$$

$$\begin{aligned} T_{\text{dev}_{\text{corrected}}} &= 2.5 * (\text{Effort}_{\text{corrected}})^c \\ &= 2.5 * (16.89324)^{0.38} \\ &= \mathbf{7.319323296 \text{ months}} \end{aligned}$$

3.3. Implications

3.3.1. Advantages

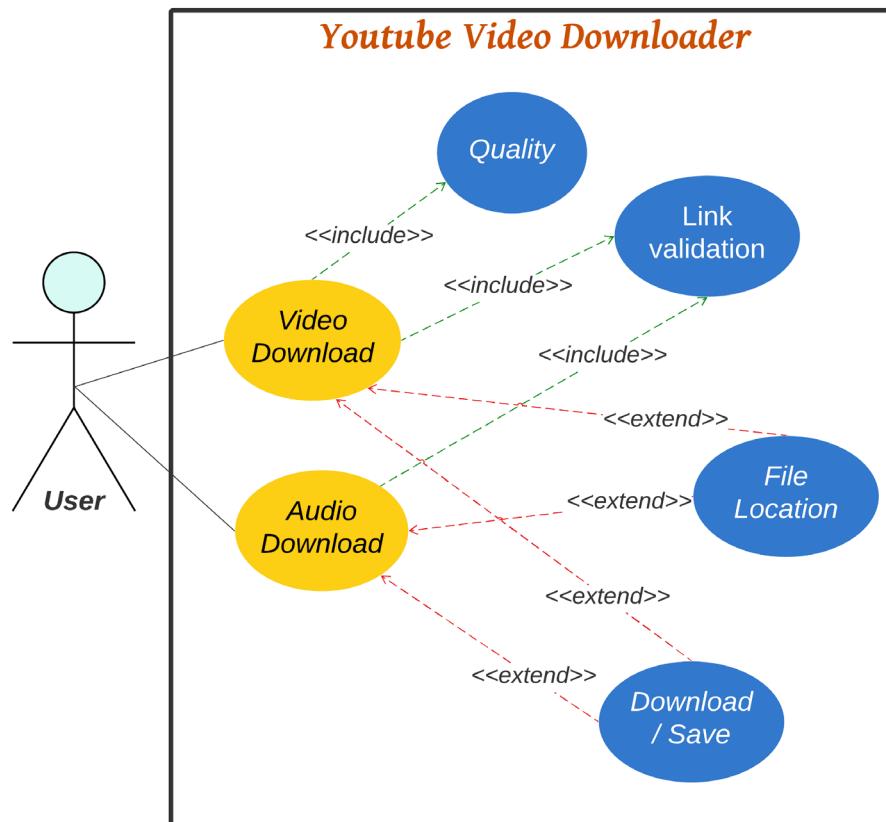
COCOMO is a simple model, and should help one to understand the concept of project metrics estimation.

3.3.2. Drawbacks

COCOMO uses KLOC, which is not a proper measure of a program's size. Indeed, estimating the size of a software is a difficult task, and any slight miscalculation could cause a large deviation in subsequent project estimates. Moreover, COCOMO was proposed in 1981 keeping the waterfall model of project life cycle in mind. It fails to address other popular approaches like prototype, incremental, spiral, agile models. Moreover, in present day a software project may not necessarily consist of coding of every bit of functionality. Rather, existing software components are often used and glued together towards the development of a new software. COCOMO is not suitable in such cases.

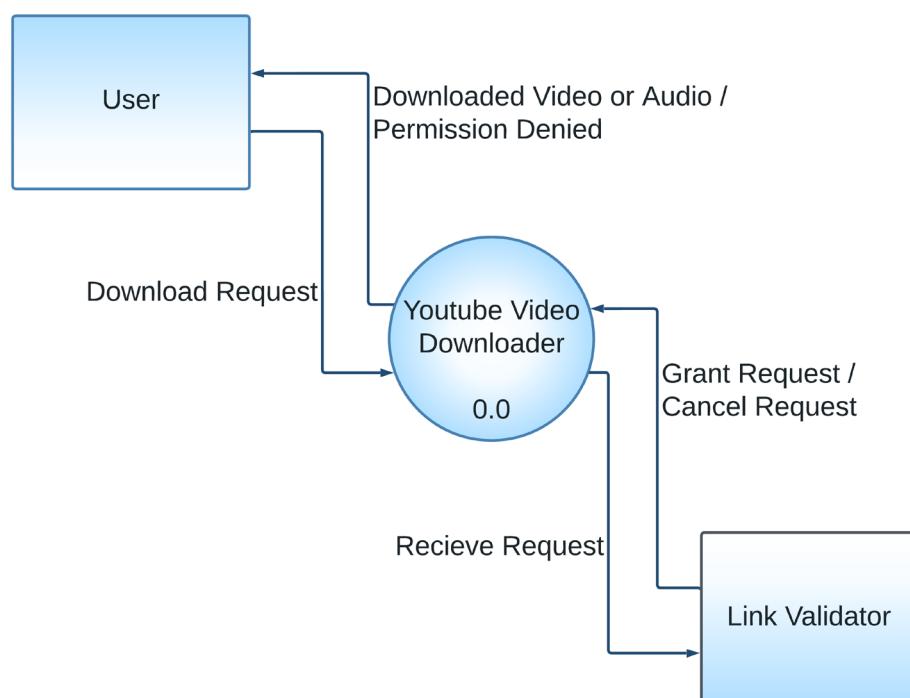
4. Design of the system

4.1. Use case Diagram

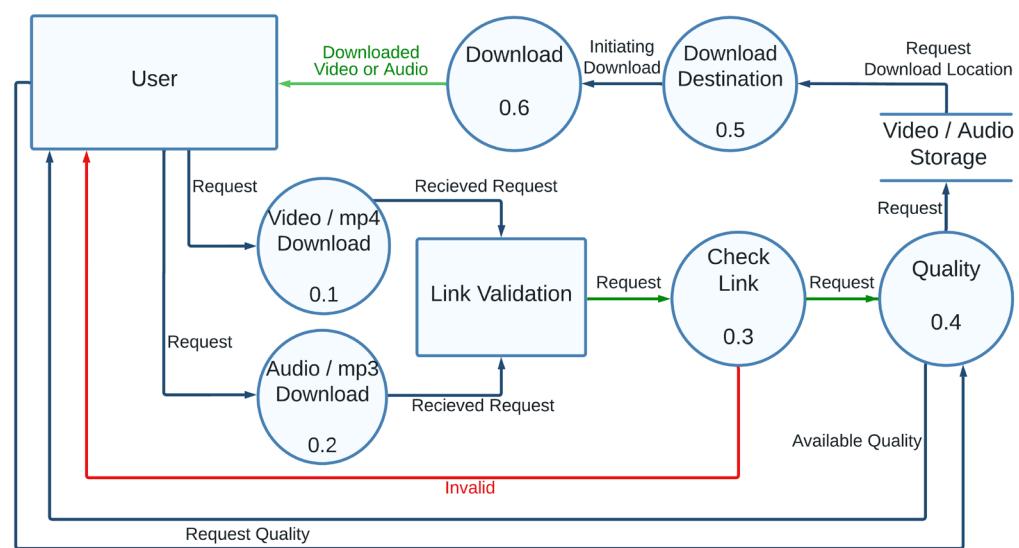


4.2. Data Flow Diagram

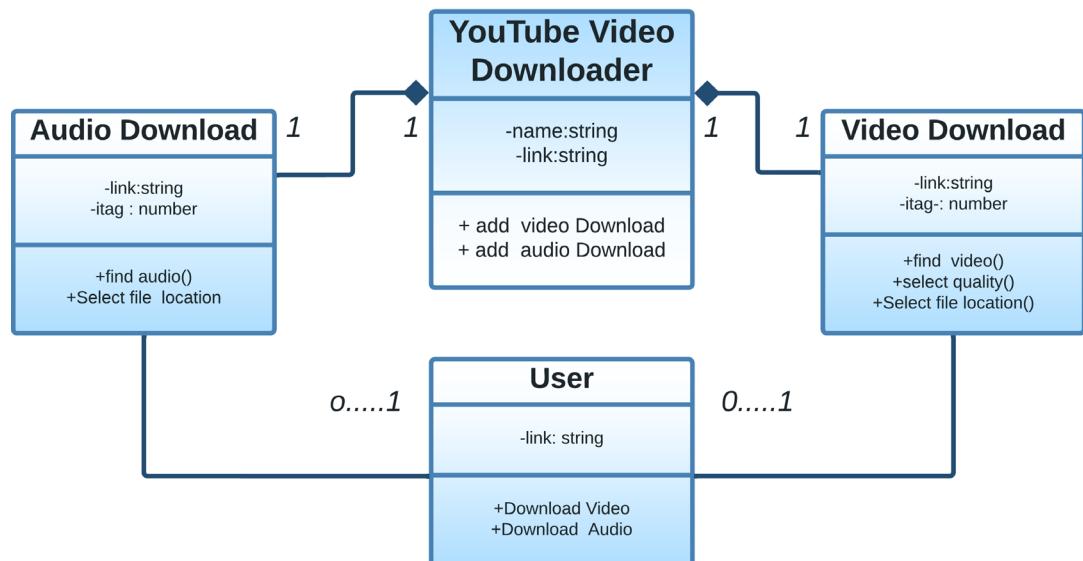
4.2.1. Level – 0



4.2.2. Level – 1



4.3. UML Class Diagrams



5. Testing

5.1. Video Download

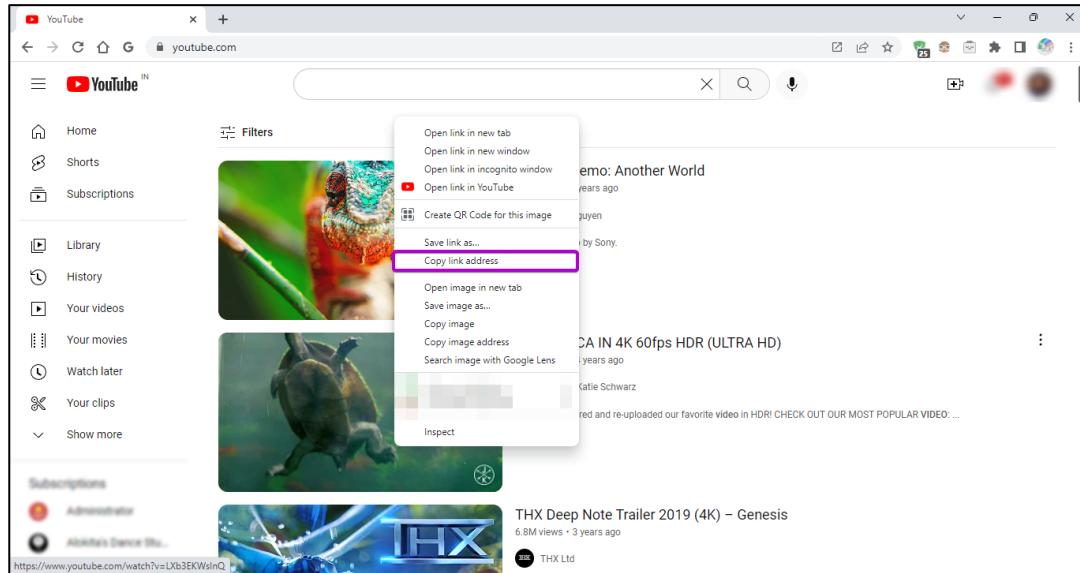


Fig:5.1: Getting Link of Desired Video from youtube.com

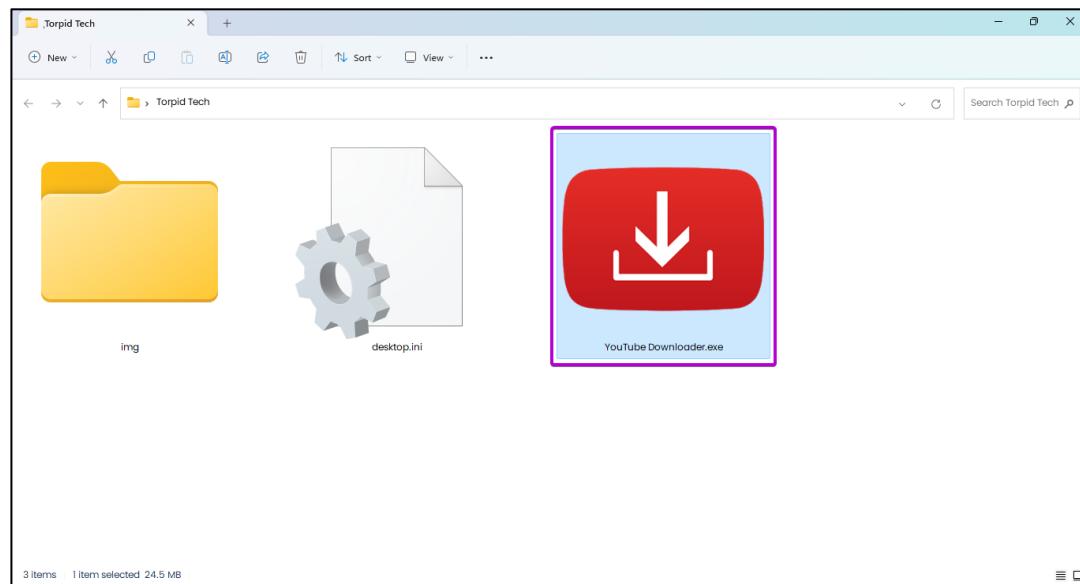


Fig 5.2: Launching the software

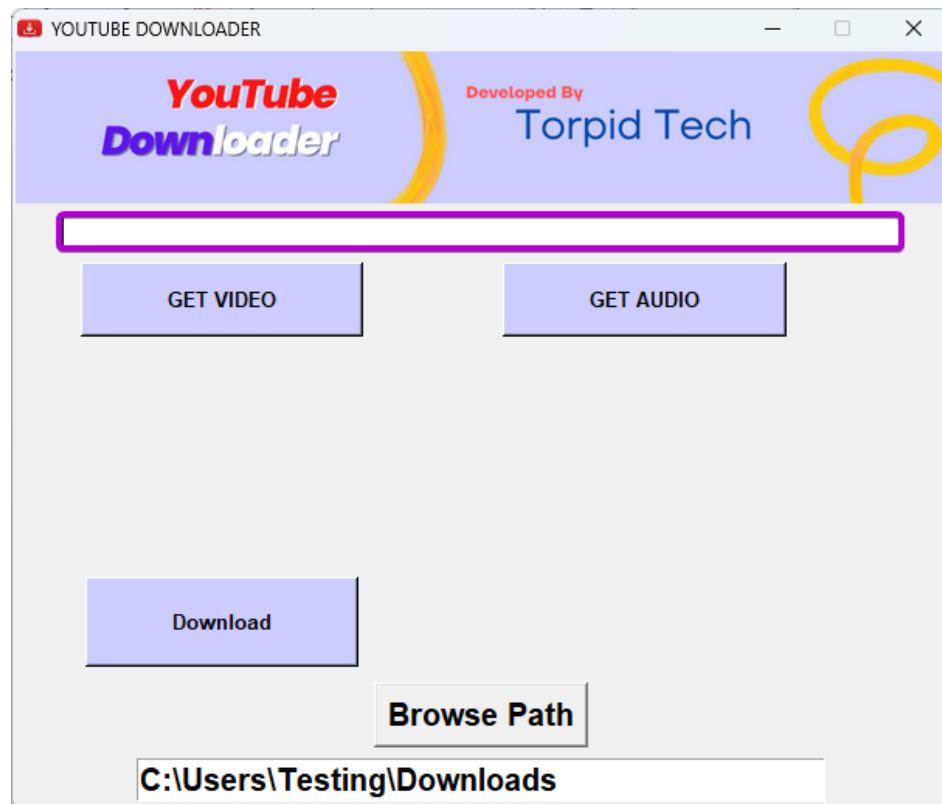


Fig 5.3: Pasting the YouTube Video Link Copied Earlier

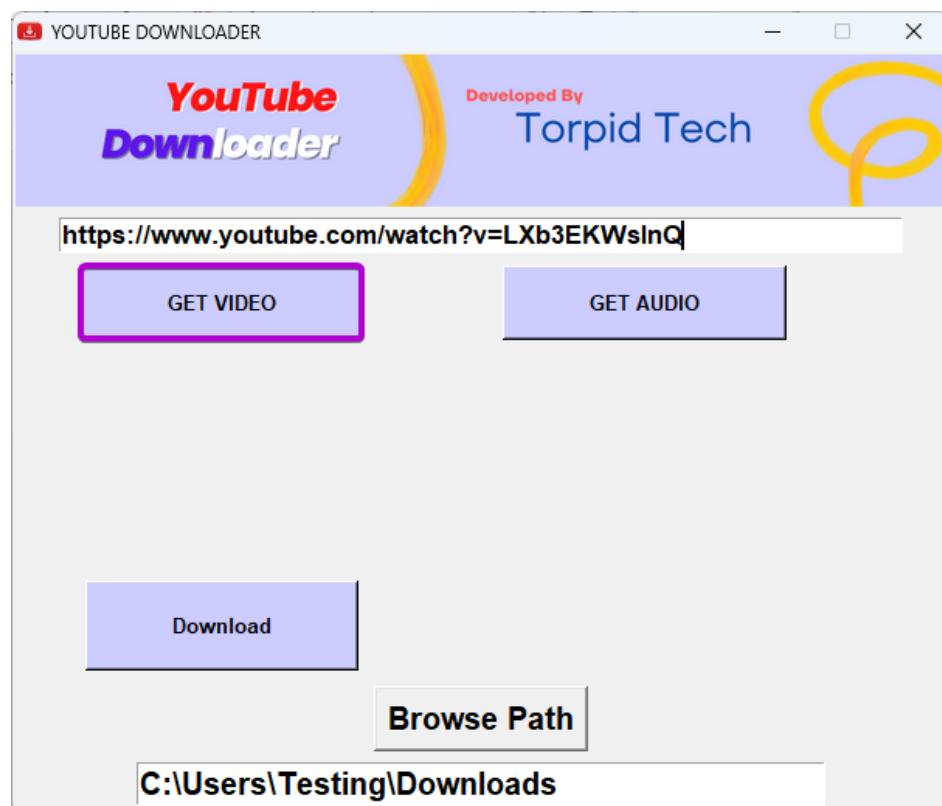


Fig 5.4: Click Get Video to process the link



Figure 5.5 Selecting the resolution



Figure 5.6: Select the preferred resolution, for now selecting the 720p quality



Figure 5.7: Click Download to Start the download process

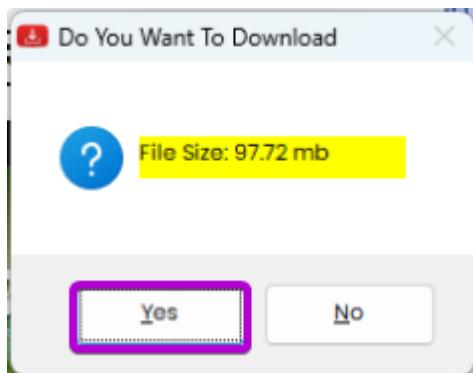


Figure 5.8: After Clicking on Download the above pop window appears, Check the file size and to continue click on Yes



Figure 5.9: Download process has been commenced

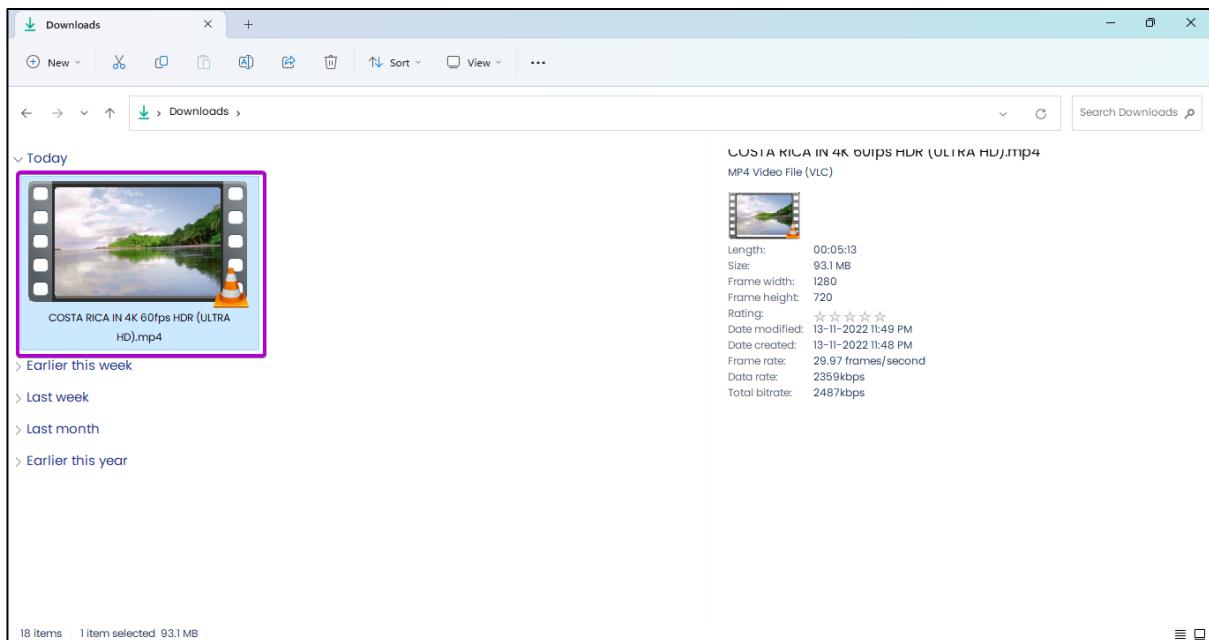


Figure 5.10: File Has Successfully been downloaded and ready to be played or shared.

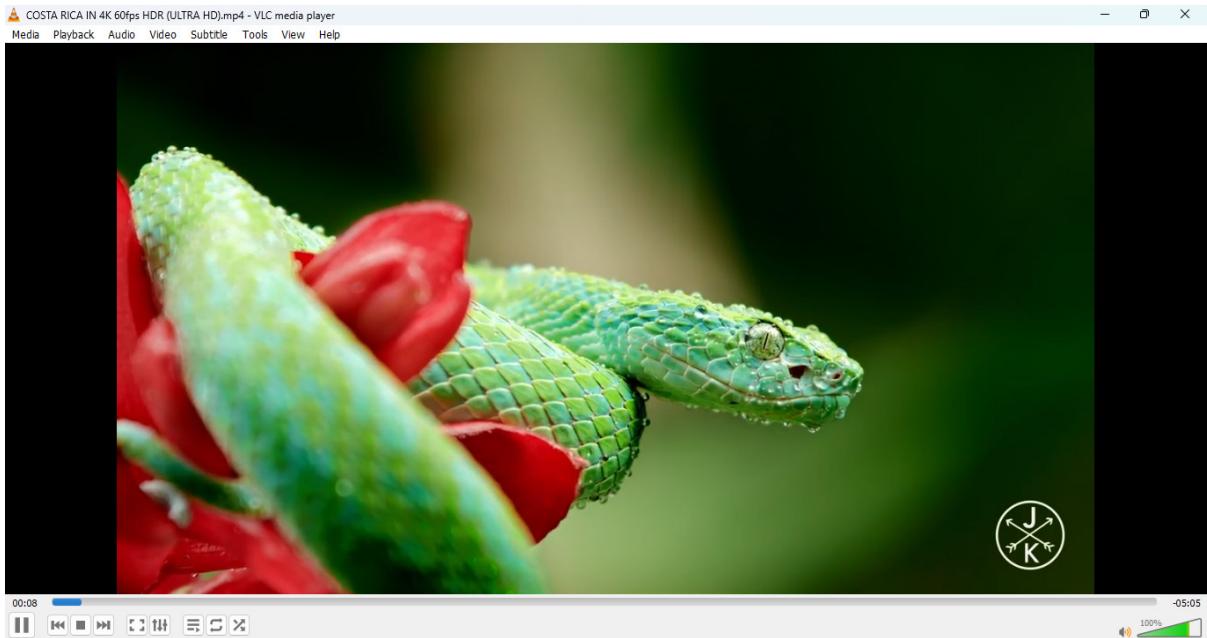


Figure 5.11: Video can be viewed successfully

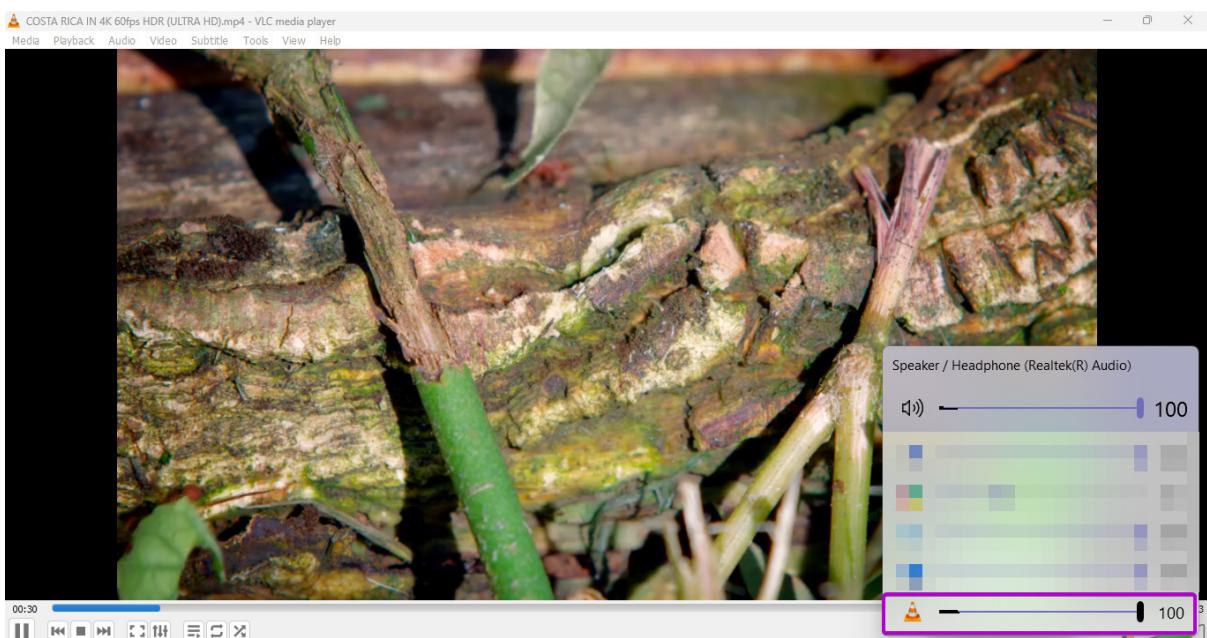


Figure 5.12: Video is being played with audio

5.2. Audio Download



Figure 5.2.1: Select Get Audio to process the link



Figure 5.2.2: Now Click Download to start downloading

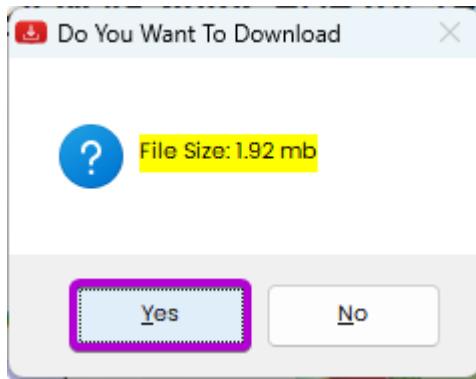


Figure 5.2.3: Now Checking the File Size and clicking Yes to start download

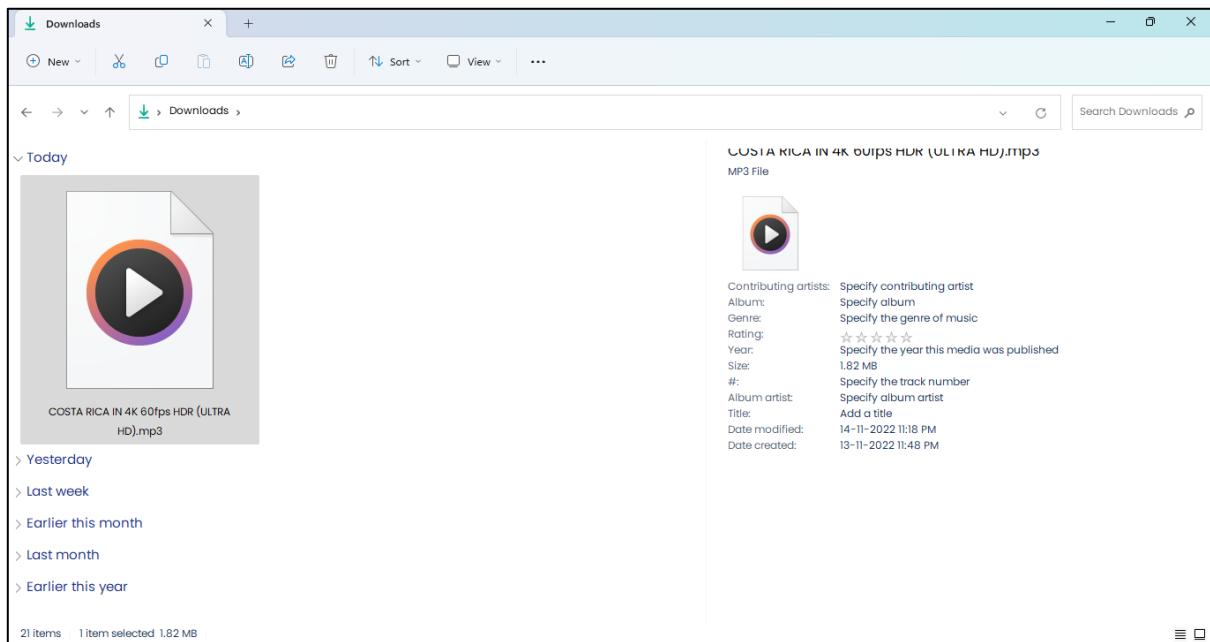


Figure 5.2.4: Download has been completed successfully and is ready for streaming or sharing

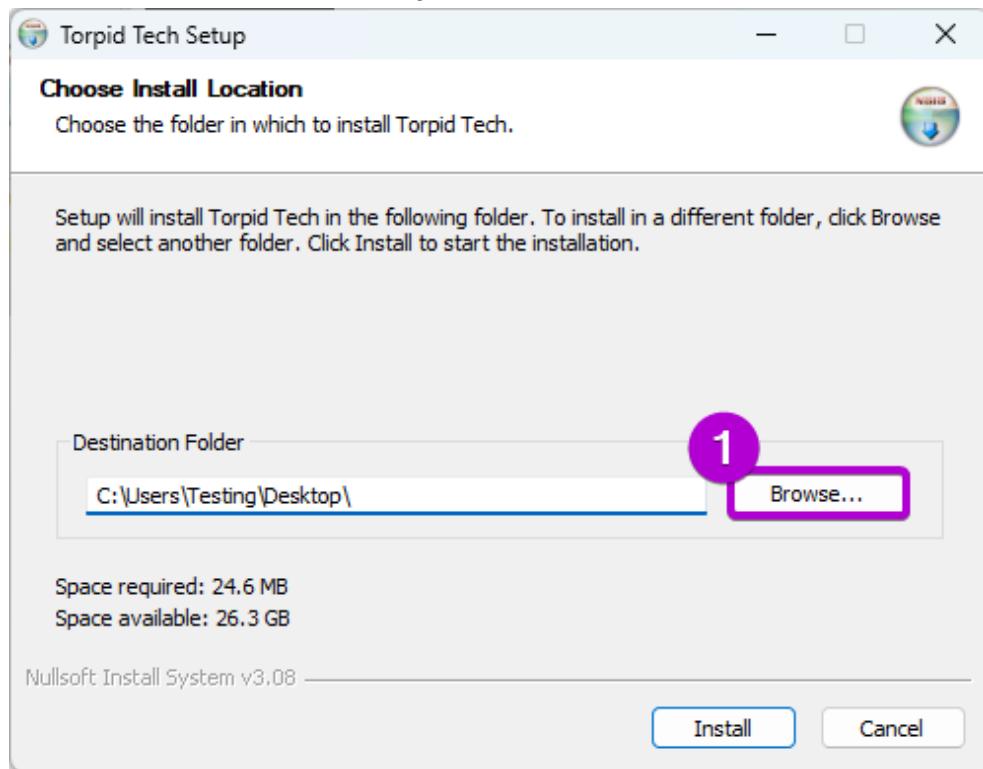


Figure 5.2.4: Downloaded Audio file can be streamed

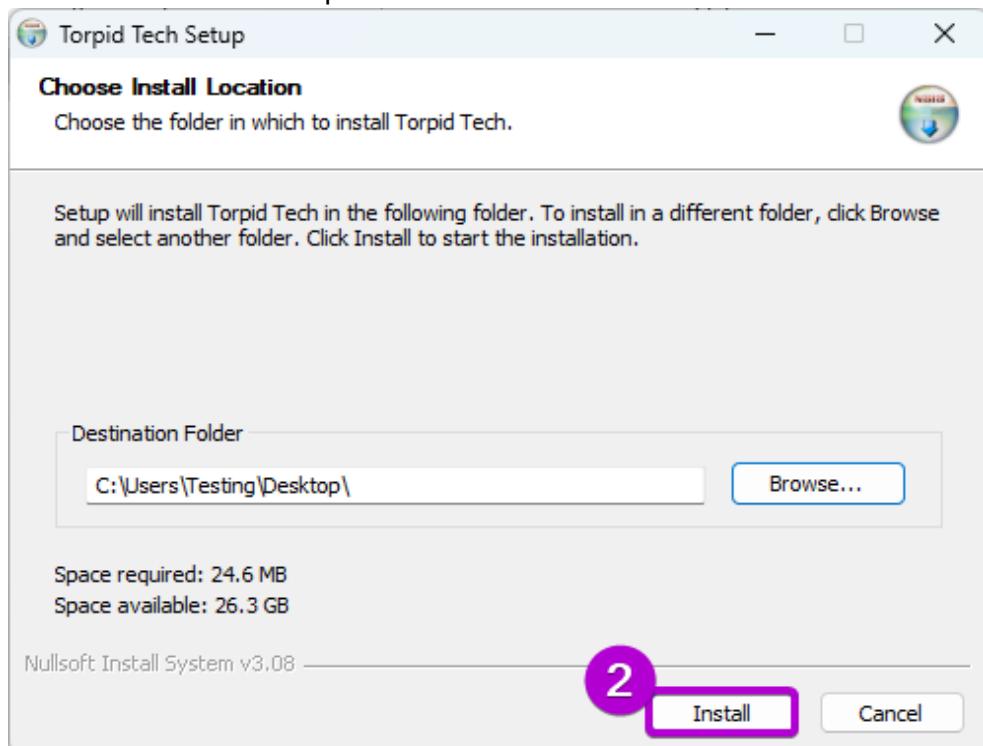
6. User manual

6.1. Installation Manual

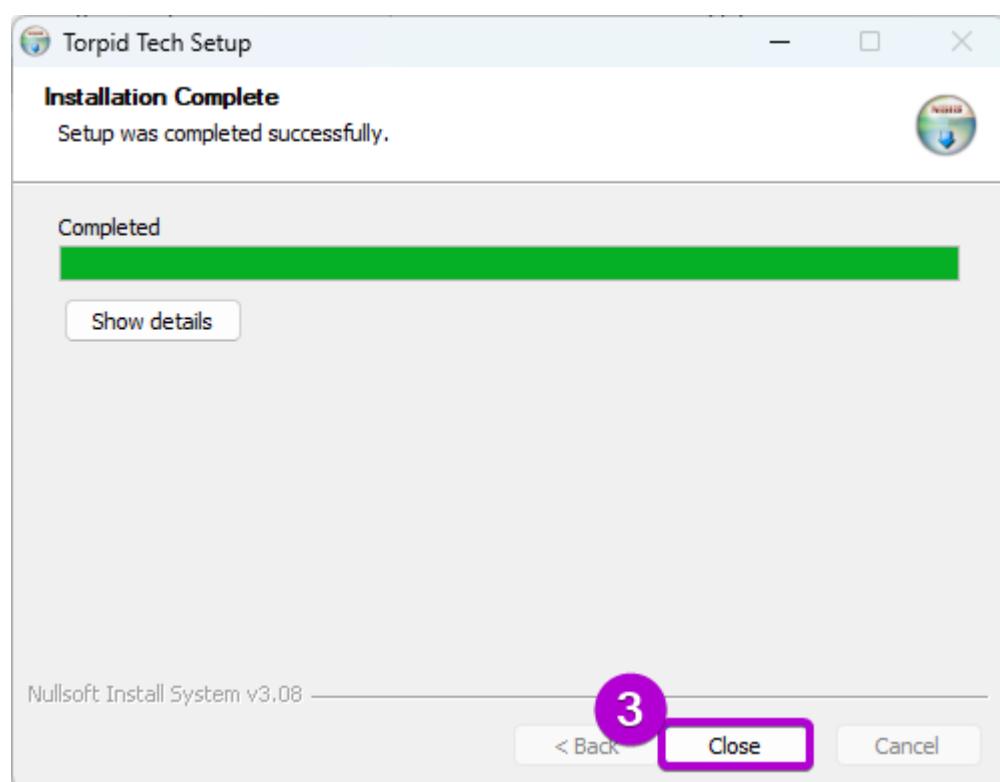
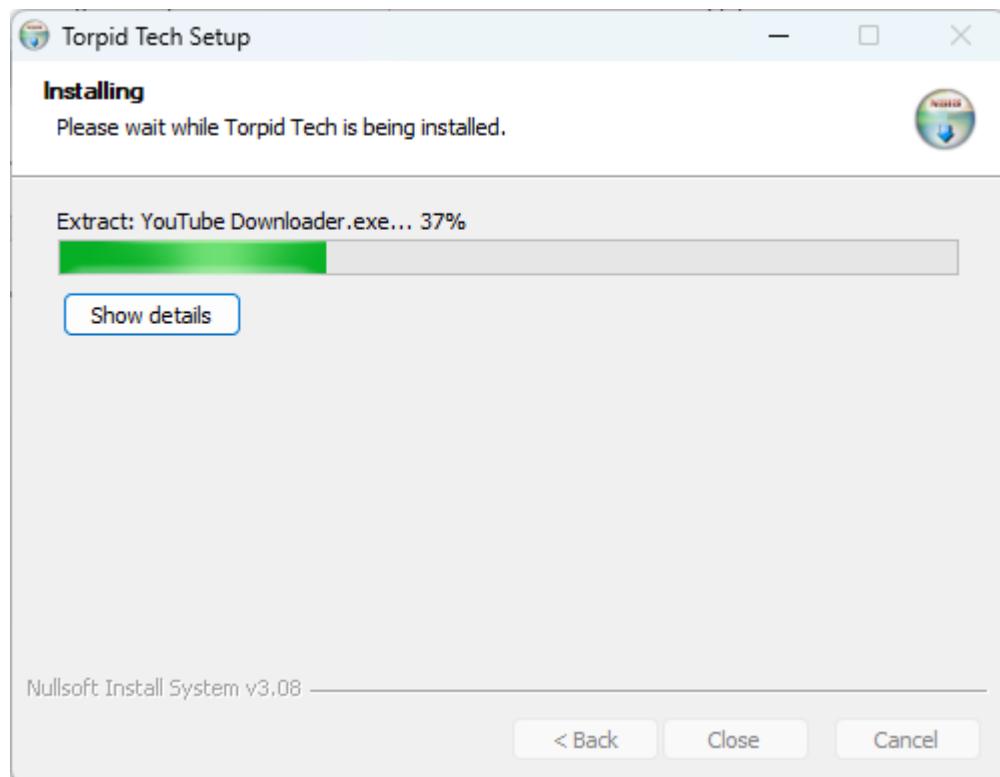
Step 1: Select Installation Folder. By default the setup installer location is the installation directory.



Step 2: Now Click *Install* to proceed with the installation

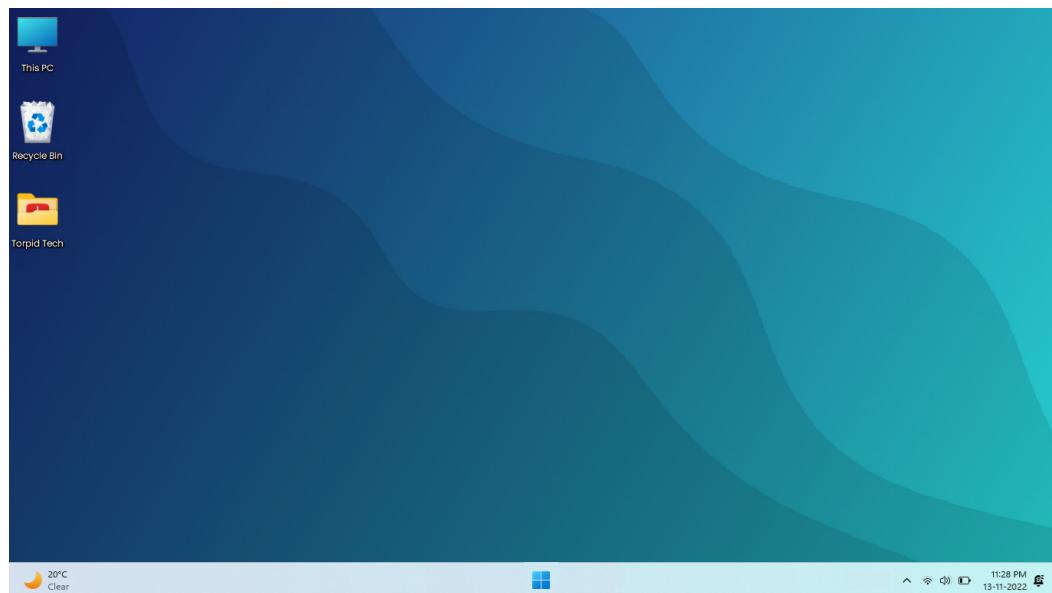


Step 3: Let the installation complete and press *Close* to exit the installer.

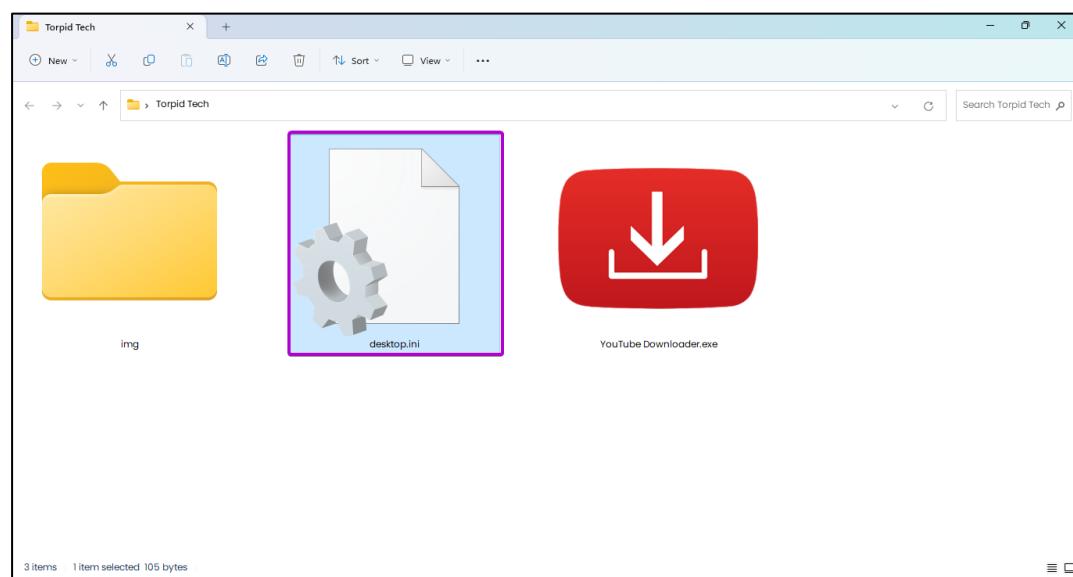


6.2. Maintenance Manual

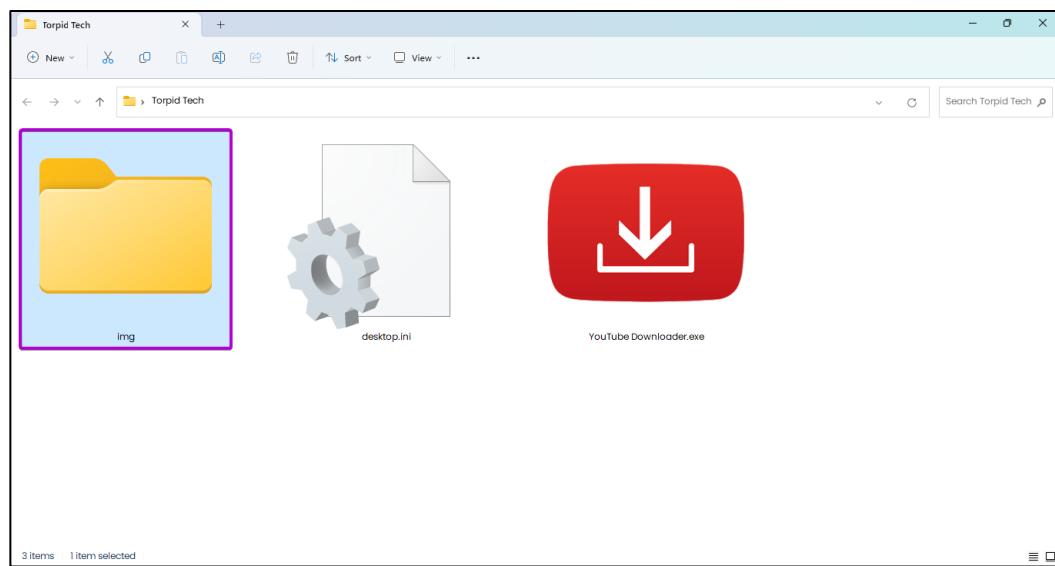
After Installation has been Completed a Folder named *Torpid Tech* is created. In the supporting image you may observe the location of the folder is in desktop.



After opening the folder you may find there is a file *desktop.ini* (highlighted below) which you may delete without any complications.



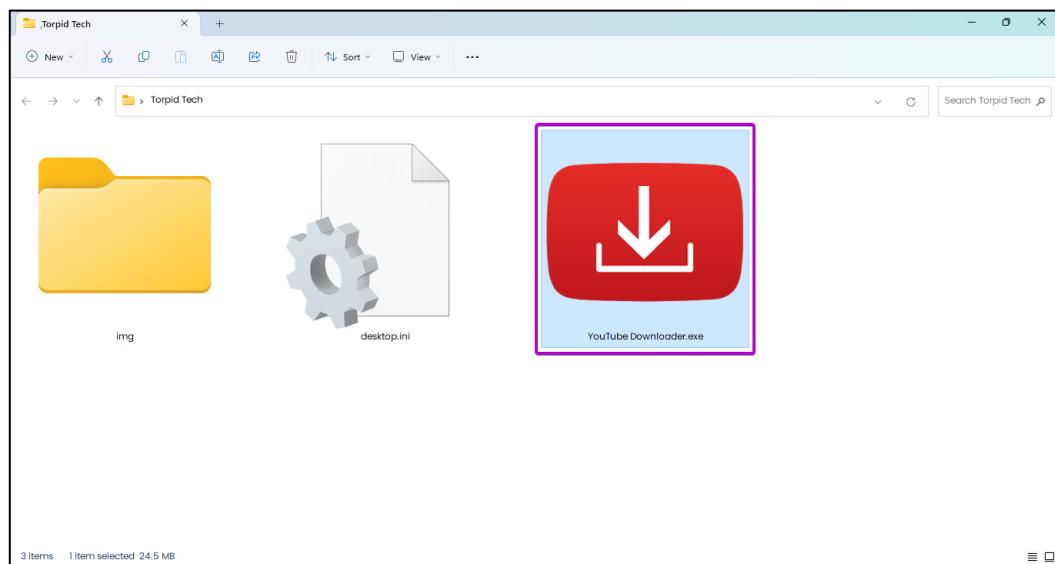
But do not delete the *img* folder (*highlighted below*). This folder contains few important file for the application to work seamless.



[P.S: Do not move the application file from this folder otherwise the application may break.]

You may create a desktop shortcut for ease of access by following the steps given below:

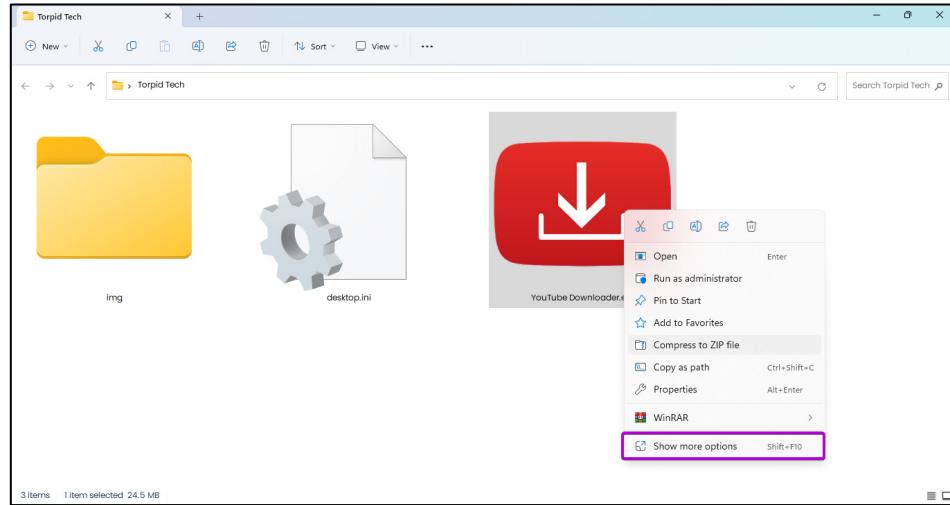
Step 1: Navigate to the installation after closing the setup installer.



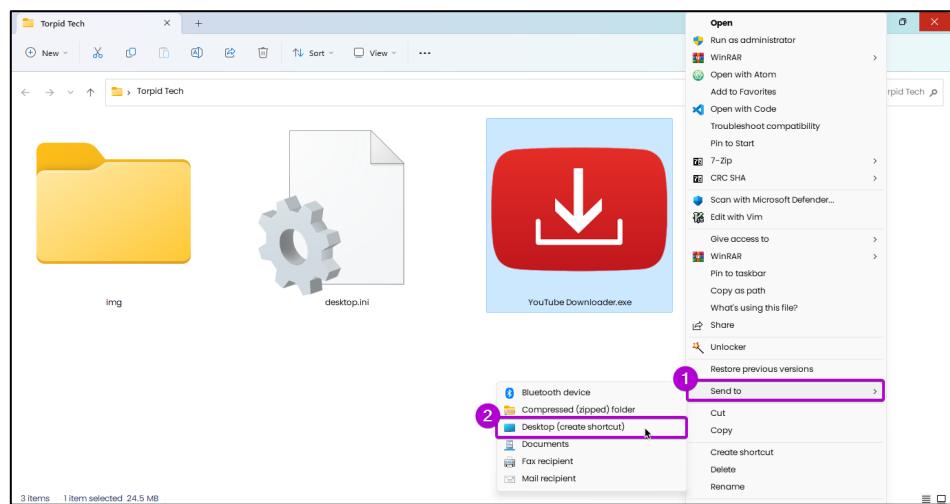
Step 2: Find the application and right-click over the icon.

If you are using Windows 11 there are some extra work to do.

Click on *Show more options* (Windows 11 users only).



Now on Expand *Send to* option and Select *Desktop (create shortcut)* option.



And Now you have it in your desktop ready to launch anytime you require to download a YouTube Video.

7. Conclusion

There are a lot of great reasons to use a YouTube video downloader. Whether you want to keep a video offline for later viewing, or you want to be able to watch it without an internet connection, a downloader can help. Plus, if you have a slow internet connection, or if you're worried about streaming data usage, downloading videos is a great way to save time and money. We hope this article has helped you understand the benefits of using a YouTube video downloader. If you're looking for a quick and easy way to download YouTube videos, then you'll want to check out our YouTube Video Downloader. With just a few clicks, you can have your favourite videos saved on your computer so that you can watch them anytime, anywhere. Plus, our downloader is completely free to use! So what are you waiting for? Go ahead and give it a try!

7.1. Limitation

Downloading YouTube Videos is against its policies. Currently all video formats are not supported but our tools download the videos in the widely used video format i.e., mp4, as well as audio in mp3 format. Our tool is limited to download videos with resolution capped at 720p which is generally the one of the most common resolution. To summarize everything, here is a list of all the implications:

Pros:

- Downloading YouTube videos allows for offline viewing or sharing purposes.
- Convenient and easy to use.
- YouTube Video Downloader is a free tool.

Cons:

- May be against YouTube's terms of service to download videos.
- Requires a stable internet connection to work properly.
- Can be a bit resource taking while downloading large videos.
- Supports mainstream formats that are MP4 and MP3.

7.2. Future Scope

In the coming days our goal would be making the list of cons as few as possible while primarily we work on increasing the list of downloadable video formats and resolutions.

8. Bibliography

We got immense help from our project guide Prof. Sudipta Hazra Sir without whom the project would not be successful. We also got few information from various websites namely:

- [1] <https://pypi.org/project/pytube>
- [2] <https://pytube.io/en/latest>
- [3] <https://github.com/pytube>
- [4] <https://snyk.io/advisor/python/pytube>
- [5] <https://readthedocs.org/projects/python-ytube/downloads/pdf/stable/>
- [6] <https://www.kaggle.com/code/patnaiksatyajit/downloading-youtube-videos-in-python-using-pytube/notebook>
- [7] <https://python.plainenglish.io/youtube-video-downloader-using-python-and-tkinter-b97462542300>
- [8] <https://nsis.sourceforge.io/Docs/Chapter1.html>
- [9] <https://github.com/kichik/nsis>
- [10] <https://nsis.sourceforge.io/>