# Project Report: YouTube Video Downloader

Prepared by: Abhyudit Singh  
Reg. Number: 23FE10CSE00790  
Date: 25 Nov 2024

## Abstract

This project aims to provide a simple and efficient solution for downloading YouTube videos. Using Python's Tkinter library for the GUI and yt-dlp for downloading functionality, the application allows users to paste a video link and download the highest-resolution version effortlessly. The project is designed to simplify video downloads and provide users with a seamless experience.

## Introduction

Downloading videos from YouTube is a common requirement for offline viewing, learning, or entertainment. However, many available tools are either paid or complicated to use. This project leverages Python's simplicity to create a user-friendly application for downloading YouTube videos.

## Requirements

Software:

* Python 3.x
* yt-dlp
* Tkinter

Hardware: Any computer with internet access.

## Implementation

The project uses Tkinter for creating a graphical user interface (GUI). yt-dlp is integrated for downloading YouTube videos by fetching the video link from the user. The application provides real-time feedback on the download status.

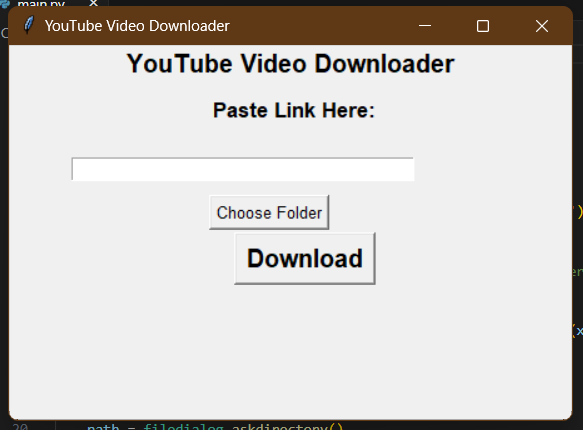
Key Features:

* Simple GUI for entering video links.
* Downloads the highest-resolution video available.
* Status updates for better user interaction.

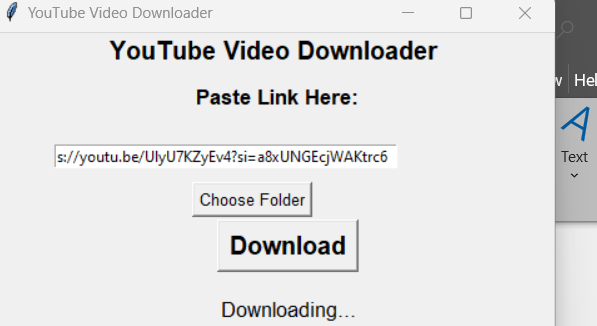
## Results

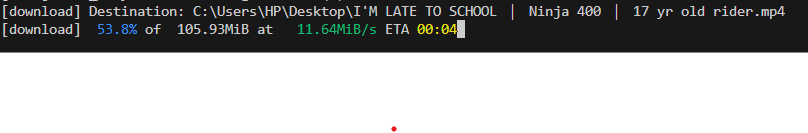
The application successfully downloads YouTube videos and stores them locally. Below are screenshots illustrating the program in action:

1.Input screen

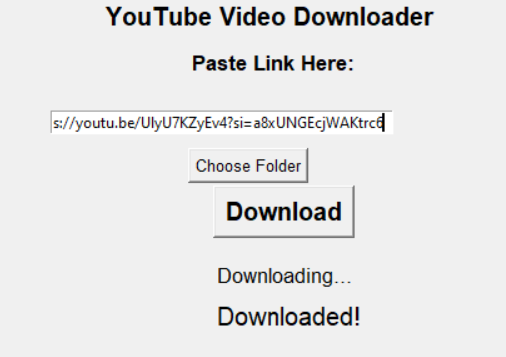


2. Download Progress





3. Download complete



## Future Enhancements

• Add functionality to download playlists.

• Provide options for selecting formats (e.g., MP4, MP3).

• Enhance error handling for unsupported URLs.

## Conclusion

This project demonstrates the utility of Python for developing practical tools. The YouTube Video Downloader is a user-centric application designed to make video downloads accessible to all. Future enhancements will further improve its usability and functionality.

## References

1. yt-dlp Documentation (https://github.com/yt-dlp/yt-dlp)

2. Python.org for Python resources and documentation.

3. Stack Overflow for resolving implementation challenges.