ABSTRACT

This Python Project provides a graphical user interface (GUI) application for downloading YouTube videos using the pytube library and the tkinter library for creating the GUI. The GUI includes a label for displaying the title of the YouTube video, an input field for entering the YouTube link, a label for displaying the download status, a label for displaying the download progress percentage, a progress bar for visualizing download progress, and a download button for initiating the download process.

The code uses customtkinter, which appears to be a custom module for creating themed tkinter widgets. The appearance mode is set to "System" and the default color theme is set to "blue". The code also includes an "on_progress" function that is called during the download process to update the progress percentage and progress bar.

When the user enters a YouTube link and clicks the download button, the "startDownload" function is called. This function uses the pytube library to download the YouTube video with the highest resolution available. If the download is successful, the title of the video is displayed, and if there is an error, an error message is displayed. The progress of the download is displayed in both percentages and through a progress bar. The GUI runs in an infinite loop using the "app.mainloop()" function, allowing the user to interact with the GUI until it is closed.

Table of Contents

Chapter 1: Introduction

1.1 About Youtube Video Downloader

This Python Project is a GUI application that allows users to download YouTube videos. It utilizes the pytube library for handling video downloads and tkinter library for creating the graphical interface. Users can input a YouTube link, initiate downloads, and monitor progress through a progress bar and percentage display. The code also includes customtkinter, a custom module for creating themed tkinter widgets with "System" appearance mode and "blue" default color theme. The application provides feedback on download status, such as displaying video titles upon successful downloads and error messages for download failures. The GUI runs indefinitely, enabling user interaction until closure. This user-friendly YouTube downloader GUI offers a convenient way to download YouTube videos.

The objectives of this YouTube downloader GUI project are:

1. Provide a user-friendly and visually appealing interface for users to easily download YouTube videos.

2. Utilize the pytube library to handle the download process, allowing users to download videos with the highest available resolution.

3. Implement a progress bar and percentage display to provide real-time feedback on the download progress.

4. Display the video title upon successful download, giving users confirmation of a completed download.

5. Show error messages in case of download failures, helping users identify and resolve issues.

6. Utilize customtkinter, a custom module for creating themed tkinter widgets, to enhance the appearance of the GUI with a "System" mode and a "blue" color theme.

7. Allow for user interaction through an infinite loop using the tkinter mainloop() function, enabling users to interact with the GUI until it is closed.

8. Provide a convenient and efficient way for users to download YouTube videos within a single application, eliminating the need for separate tools or commands.

Different Files/Modules used:

1. **`tkinter`:** This is the standard Python library for creating GUI applications. It is used to create the graphical interface of the YouTube downloader application, including windows, labels, buttons, entry fields, and progress bars.

2. **`customtkinter`:** This is a custom module used in the project for creating themed tkinter widgets. It provides appearance modes and color themes to customize the look and feel of the GUI.

3. **`pytube`:** This is a popular Python library used for handling YouTube video downloads. It is utilized in the project to fetch YouTube video information, such as video title and available streams, and to download the selected video stream.

4. **`main.py`:** This is the main Python script that contains the main logic of the YouTube downloader application. It includes functions for handling the download process, updating the GUI elements, and managing user interactions.

.

Libraries used:

The YouTube downloader GUI project utilizes the following libraries:

1. `tkinter`: This is the standard Python library for creating graphical user interfaces (GUIs). It provides a wide range of widgets, such as windows, labels, buttons, entry fields, and progress bars, that can be used to create the visual components of the GUI.

2. `customtkinter`: This is a custom module used in the project that provides themed tkinter widgets. It allows for customization of the appearance of the GUI by providing different appearance modes (e.g., "System" mode) and color themes (e.g., "blue" theme) for the widgets.

3. `pytube`: This is a popular Python library for downloading YouTube videos. It provides functionalities for fetching YouTube video information, such as video title, available streams, and download progress. It also allows for downloading videos with different resolutions and formats.

These libraries are used in the project to create the graphical interface of the YouTube downloader application, handle the download process, update the GUI elements (e.g., labels, progress bar), and provide a visually appealing appearance to the GUI using customtkinter. They enable the project to provide a user-friendly and efficient way for users to download YouTube videos within a single application.

.

Chapter 2: Working

The YouTube downloader GUI project is designed to provide a graphical user interface (GUI) for users to download YouTube videos. The working of the project can be summarized in the following steps:

1. GUI Creation: The tkinter library is used to create the visual components of the GUI, such as windows, labels, buttons, entry fields, and progress bars. The customtkinter module is also used to customize the appearance of the GUI with different appearance modes and color themes.

2. User Input: The user is prompted to input a YouTube link for the video they want to download. The input is obtained through an entry field created using tkinter, and the input is stored in a variable.

3. Video Information Retrieval: The pytube library is used to fetch the video information, such as the video title and available video streams, from the YouTube link provided by the user.

4. Display Video Information: The video information, such as the video title, is displayed in a label using tkinter. This provides feedback to the user about the selected video.

5. Download Process: The user initiates the download process by clicking the "Download" button, which triggers the startDownload() function. This function uses the pytube library to download the video stream with the highest resolution.

6. Download Progress: The on_progress() function, which is passed as a callback to the pytube library, is responsible for updating the progress of the download. It calculates the percentage of completion based on the bytes downloaded and the total file size, and updates the progress bar and percentage label in the GUI accordingly.

7. Download Completion: Once the download is complete, the GUI displays a "Downloaded" message in a label using tkinter. If any error occurs during the download process, an "Download Error" message is displayed in red.

8. User Interaction: The GUI allows for user interactions, such as inputting the YouTube link, initiating the download, and viewing the download progress. Users can also close the application window or interact with other GUI elements as needed.

9. Resource Management: The project may utilize additional resources, such as images or icons, to enhance the visual appearance of the GUI. These resources can be stored in a directory and loaded into the GUI using tkinter as needed.

Overall, the YouTube downloader GUI project provides a user-friendly and efficient way for users to download YouTube videos by utilizing tkinter for GUI creation, customtkinter for appearance customization, and pytube for video information retrieval and download management.

Chapter 3: Source Code

Figure 1 Code Part 1

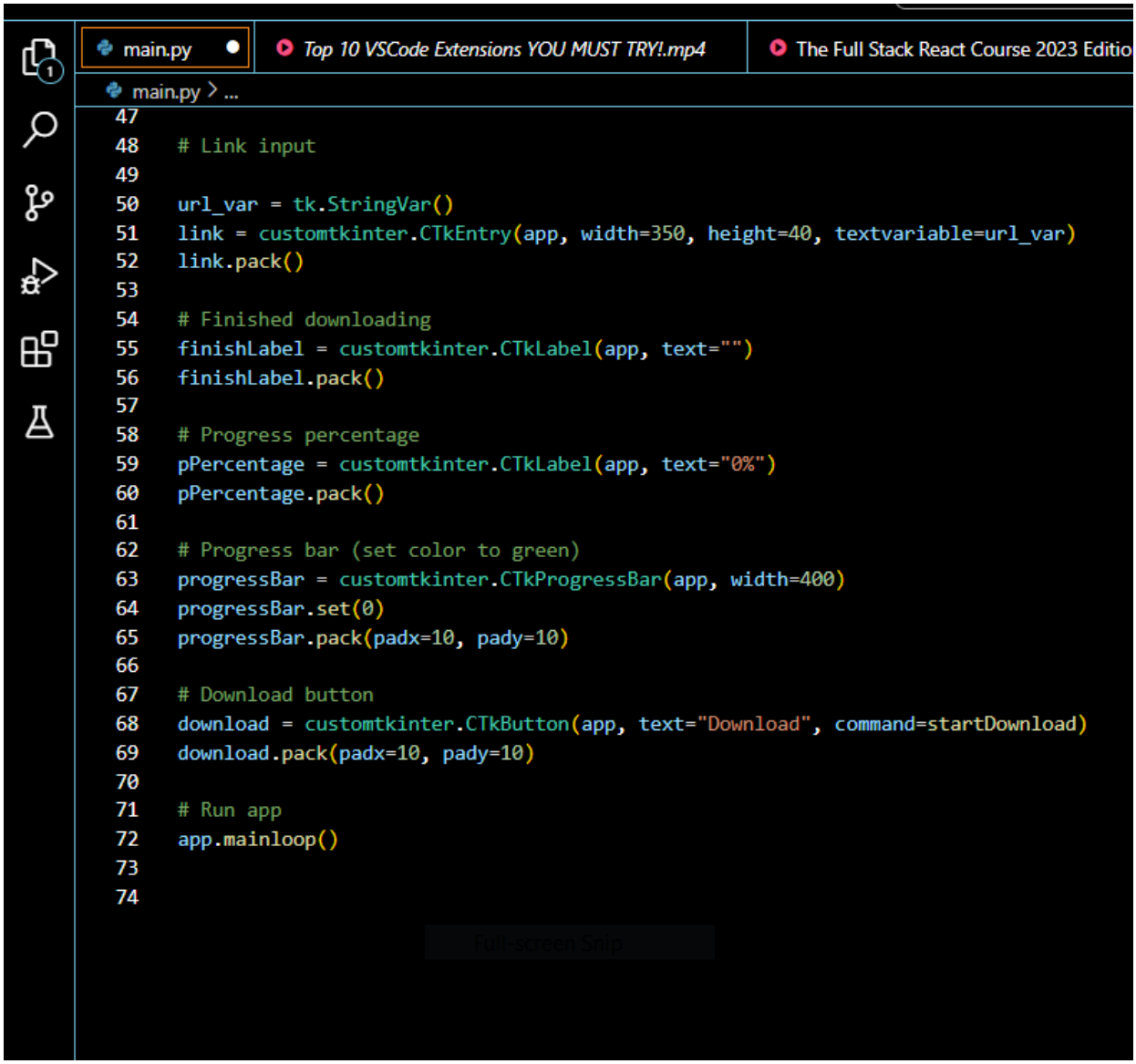

Figure 2 Code part 2

Chapter 4: Result

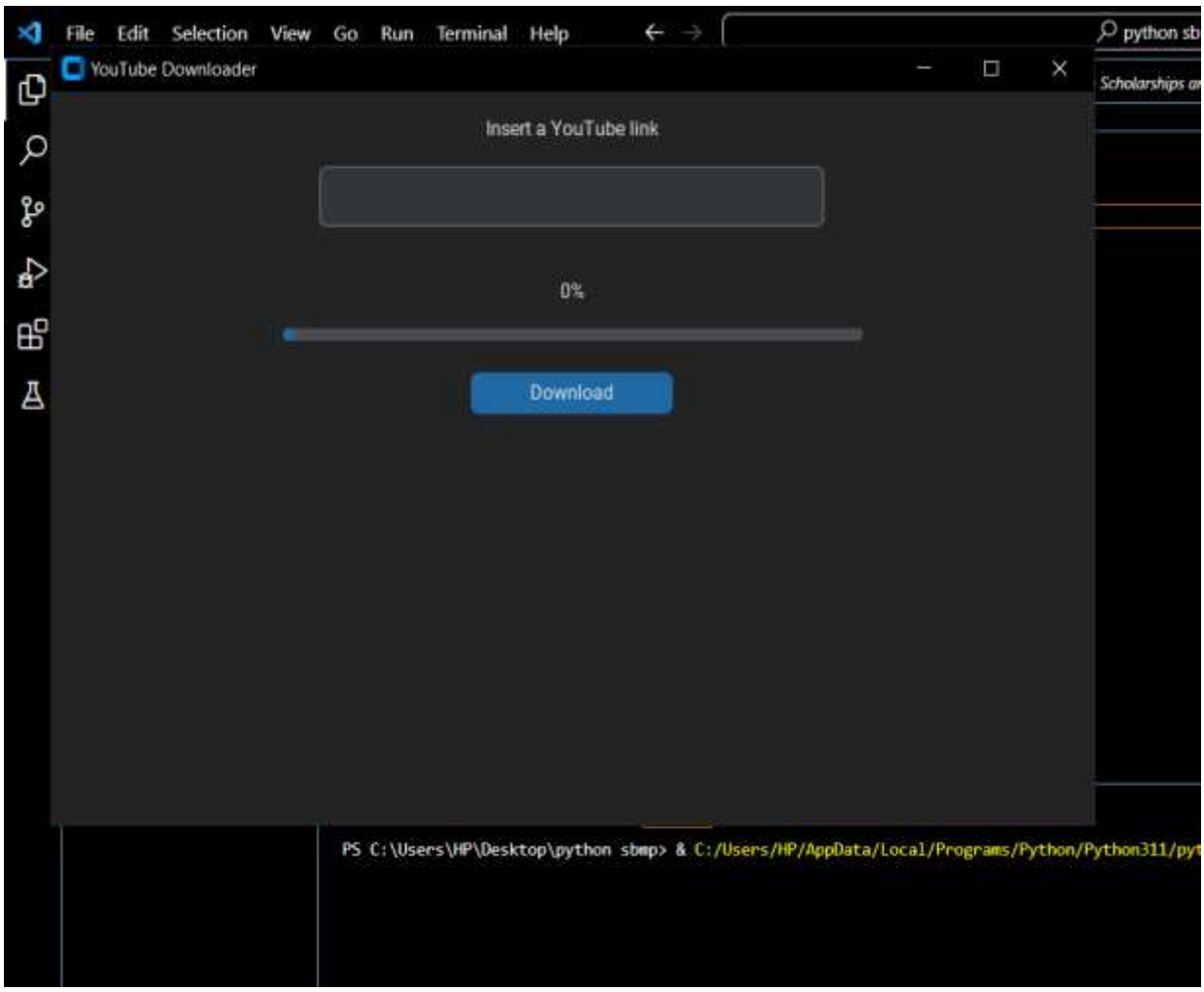

Figure 3 Youtube Video Downloader Window

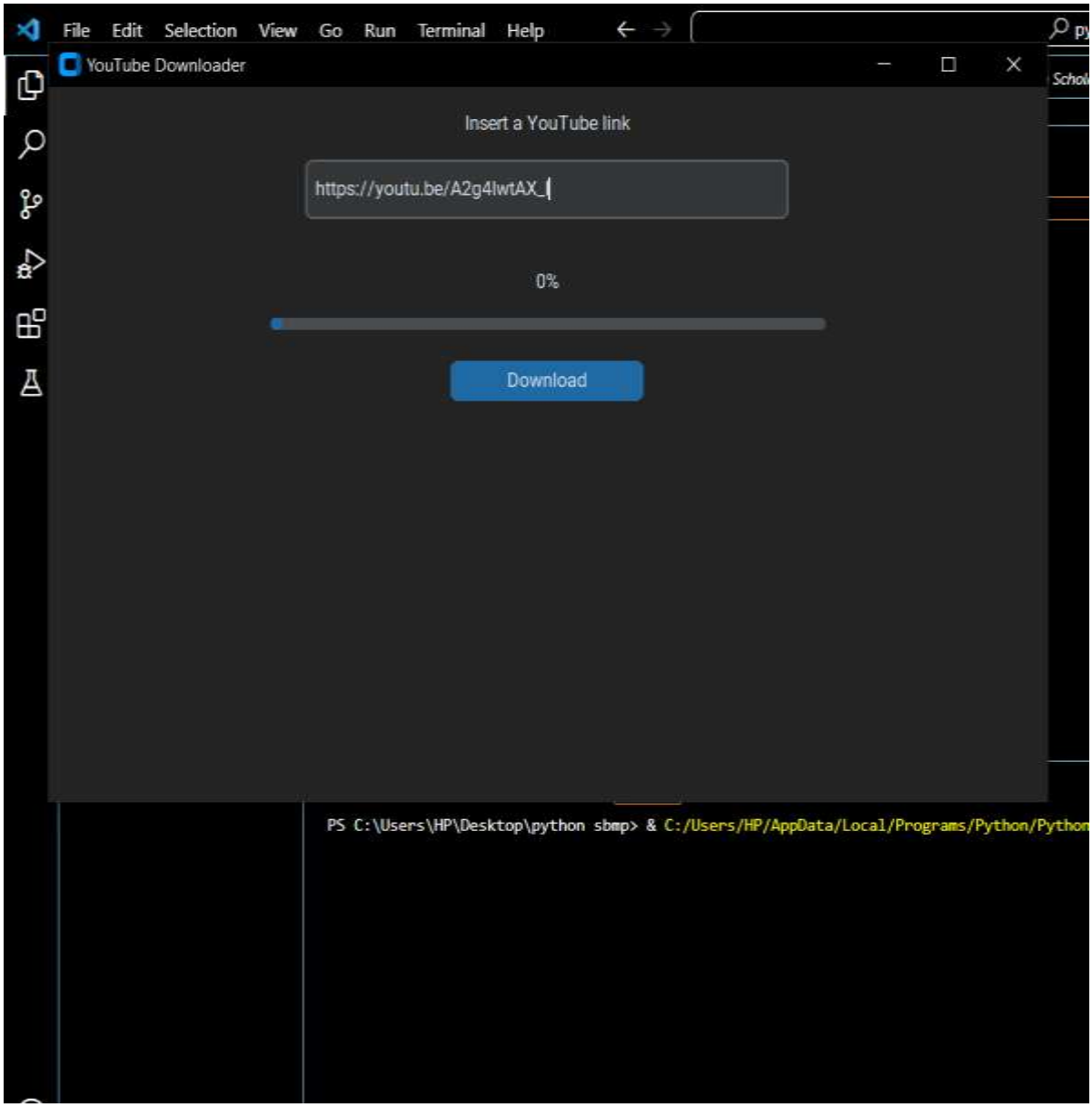

Figure 4 After pasting the link

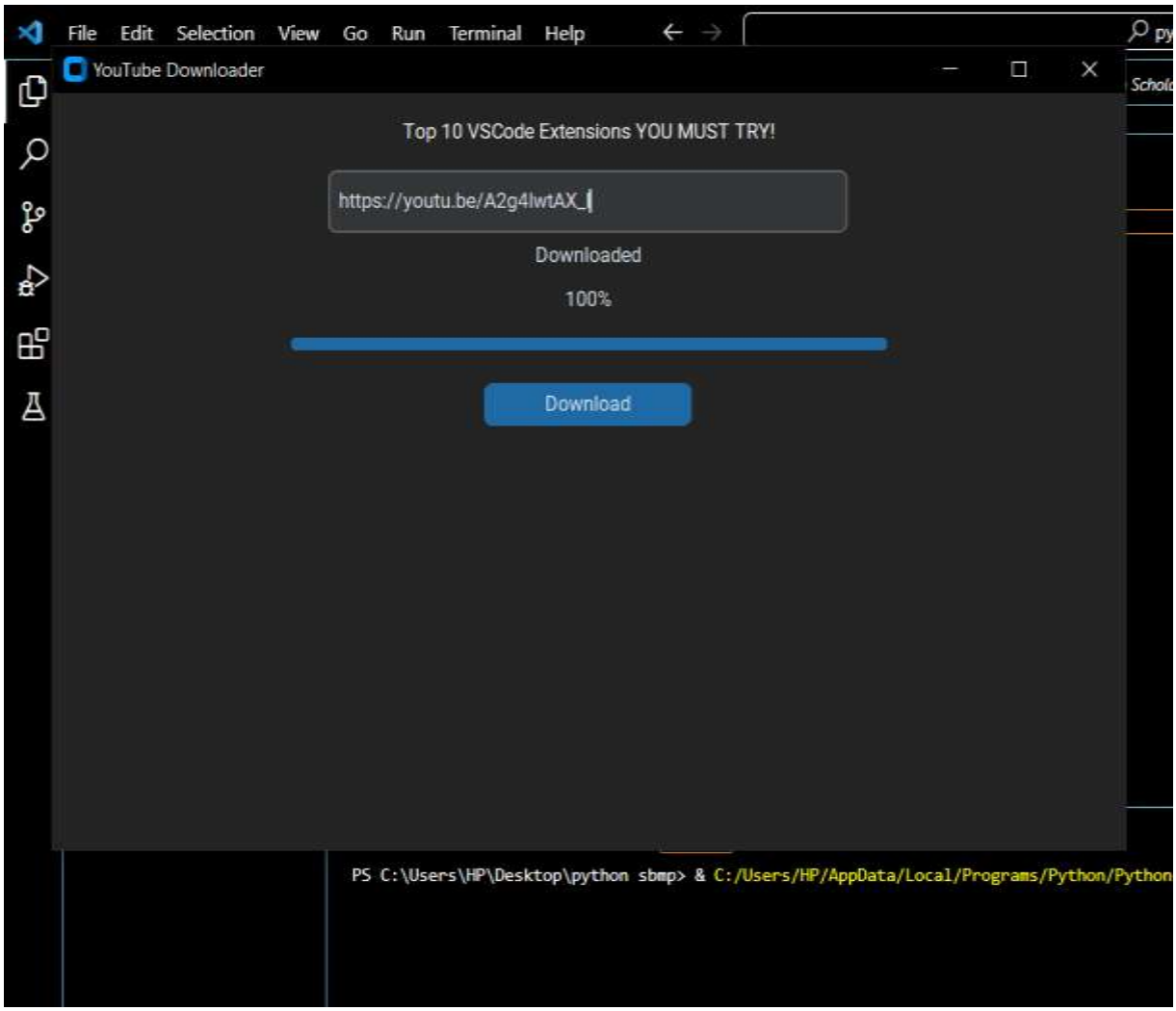

Figure 5 Youtube Video is downloaded

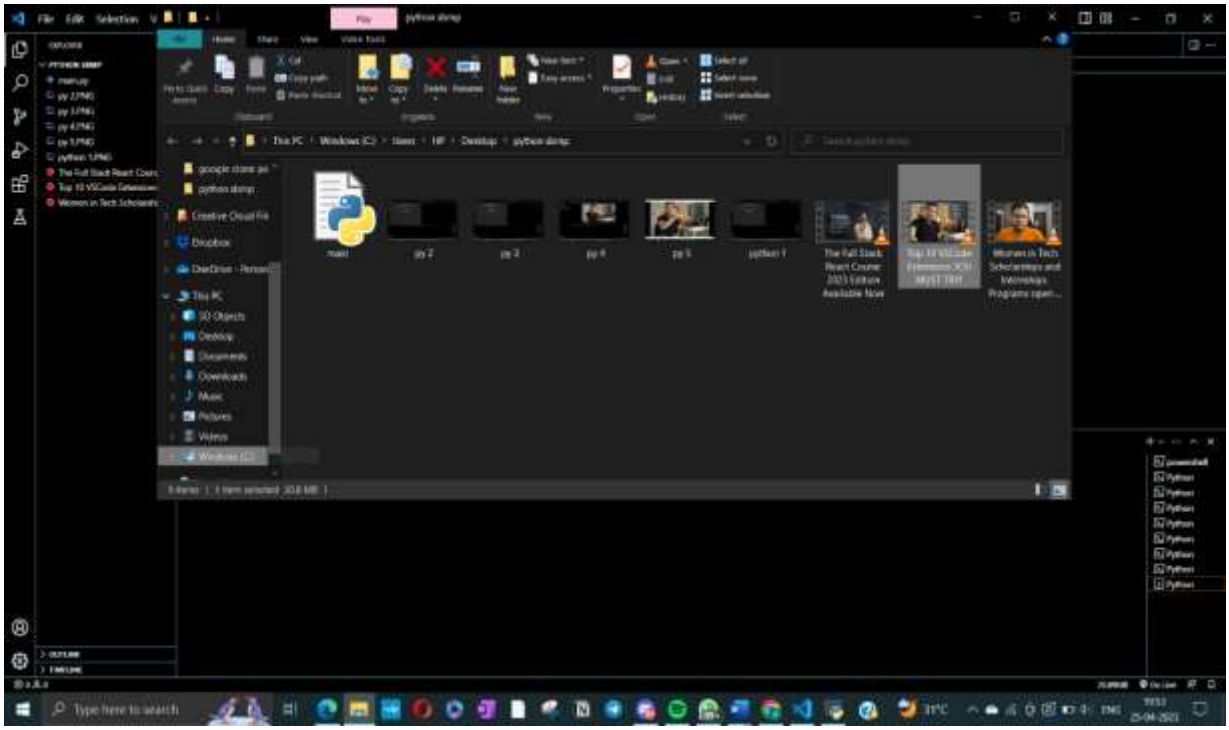

Figure 6 Downloaded video appears in folder

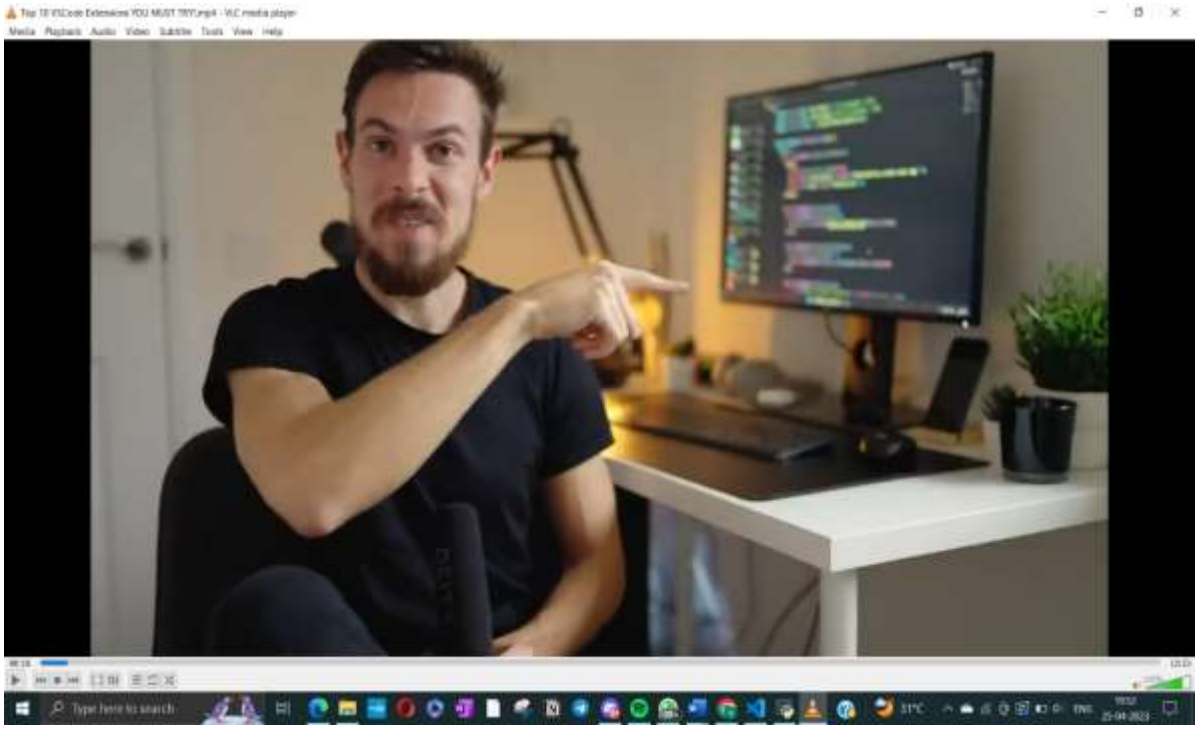

Figure 7 Downloaded video playing in VLC Media Player

Chapter 5: Conclusion

In conclusion, the YouTube downloader GUI project demonstrates the use of tkinter, customtkinter, and pytube libraries to create a user-friendly interface for downloading YouTube videos. The project allows users to input a YouTube link, fetch video information, initiate and monitor the download process, and receive feedback on the download progress. The customization options provided by customtkinter allow for a visually appealing GUI, while the pytube library handles the video information retrieval and download management tasks. This project serves as a practical example of how to create a GUI-based YouTube downloader application using Python, providing users with a convenient way to download videos from YouTube.

References:

https://youtu.be/NI9LXzo0UY0

www.google.com