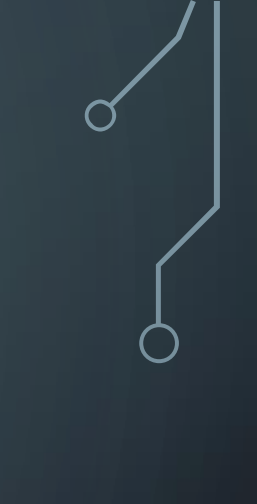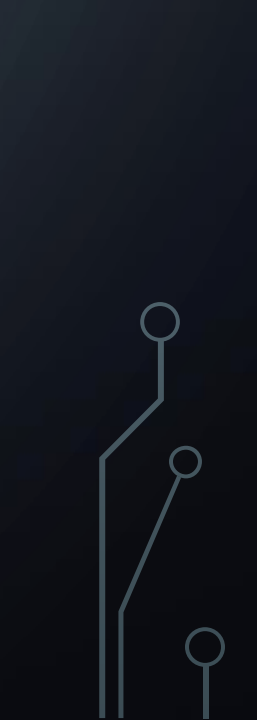# MINI PROJECT

YOUTUBE DOWNLOADER

THEERTHA PS, MILAN, AMJADH, FAHAD NAJEEB

INTRODUCTION

- PyTube-powered YouTube downloader with a user-friendly and a simple interface, changing the way you save and organize your favorite videos. Embrace convenience and elevate your multimedia experience. We used tkinter as the User Interface window used to paste the link of the desired video and and progess bar for current download information

OBJECTIVE

- Our objective for this college project is to develop a PyTube-powered YouTube downloader with a streamlined user interface, enabling users to effortlessly save and manage their preferred videos. We aim to enhance convenience and optimize the multimedia experience without creating a hassle on depending any third party services compromising privacy and safety
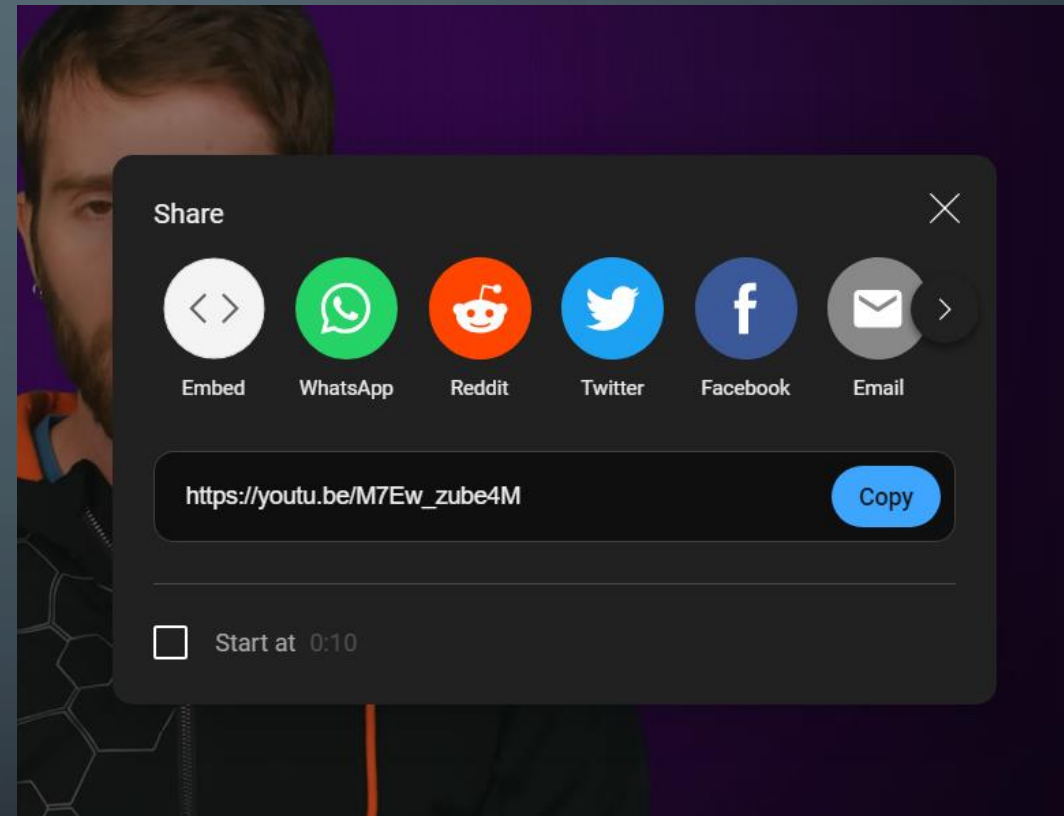
FEATURES AND UI

- We provide a simple functional and lightweight user interface which creates no confusion and requires only python and the PyTube library installed on your client.

- Interactive progress bar. Entire code fits under 2KB.
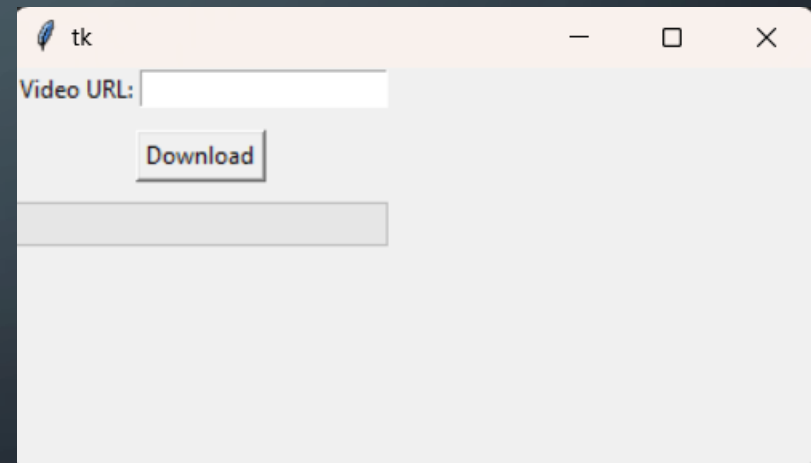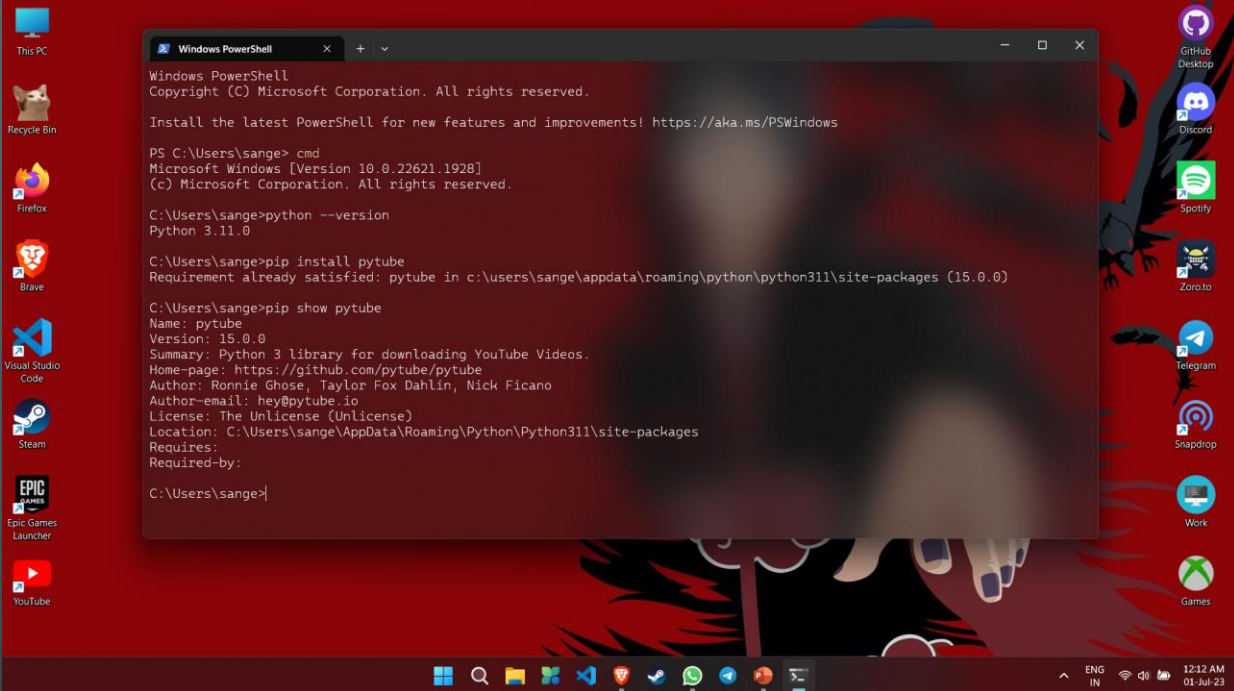
COPYING THE LINK

Go the desired YouTube video and copy the video link .

SETUP PYTUBE LIBRARY AND RUN THE APP

Run `pip install pytube` in a command prompt windows in windows.

Use terminal in Linux based operating system and use the command `pip install pytube`

RUN THE PROGRAM AND DOWNLOAD THE VIDEO

Run the youtube_downloader.py
Paste the link in the tk window pop up box.

Click on download and wait for the progress to finish.

The downloaded file will be stored in the same directory the where the youtube_downloader.py file is placed.
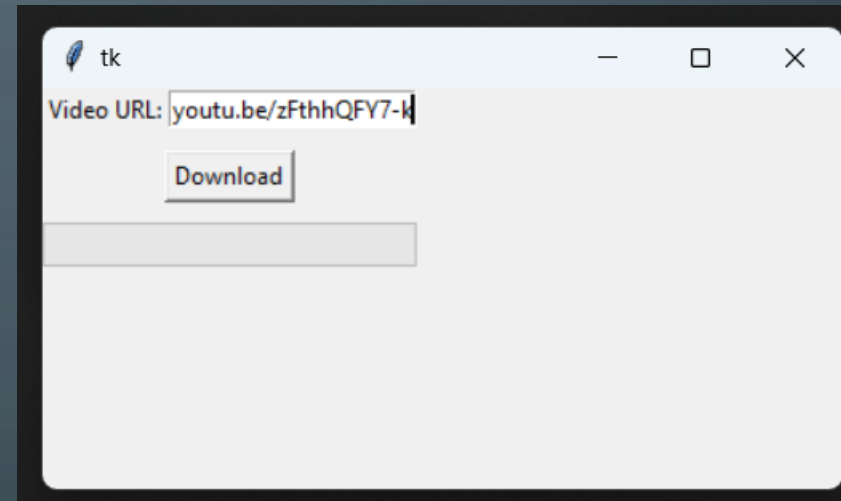
TECHNOLOGIES

PYTHON 3.11

PYTUBE

TKINTER

TKINTER.TK

CODE

```python
import os
import tkinter as tk
from tkinter import ttk
from tkinter.ttk import Progressbar
import pytube
#Creating class for the UI
class App(tk.Tk):
    def __init__(self):
        super().__init__()
        self.video_url = tk.StringVar()
```

```python
# Create the UI elements
        self.video_url_label = tk.Label(self,
text="Video URL:")
        self.video_url_entry = tk.Entry(self,
textvariable=self.video_url)
        self.download_button = tk.Button(
            self, text="Download",
command=self.download_video
        )
        self.progress_bar = Progressbar(self,
mode="determinate")
```

```python
# Layout the UI elements
        self.video_url_label.grid(row=0, column=0)
        self.video_url_entry.grid(row=0, column=1)
        self.download_button.grid(row=1, column=0,
columnspan=2, pady=10)
        self.progress_bar.grid(row=2, column=0,
columnspan=2, sticky="we")

        # Set the window size
        self.geometry("400x200")
```

```python
#Video Download Function When The Buttom is Clicked
    def download_video(self):
        # Get the video URL from the user
        video_url = self.video_url.get()

        # Default download path to current directory
        download_path = os.getcwd()

        # Download the video
        try:
            video = pytube.YouTube(video_url)
            stream = video.streams.get_highest_resolution()
            if stream:
                stream.download(output_path=download_path)
                print("Download completed!")
            else:
                print("No video available")
    except Exception as e:
        print(e)
```

```python
if __name__ == "__main__":
    app = App()
    app.mainloop()
```

ADVANTAGES

- Lightweight. Entire code fits in 55 lines.

- Platform independent. Runs anywhere. Literally

- Easy to understand and uses open source lib like tkinder and pytube.

- No ads, easy to run fast and efficient.

- Code is written in a way for people to customize, improve, adapt to specifc needs and upgradation in the future.

DISADVANTAGES

- tkinder UI looks outdated and a bit finicky.

- Lack of customization in configuring the downloads.

- User still needs to setup python and pytube library.

- Sudden breaking API changes to YouTube videos could make the current version of pytube and user will need to wait till pytube releases and update

THANK YOU