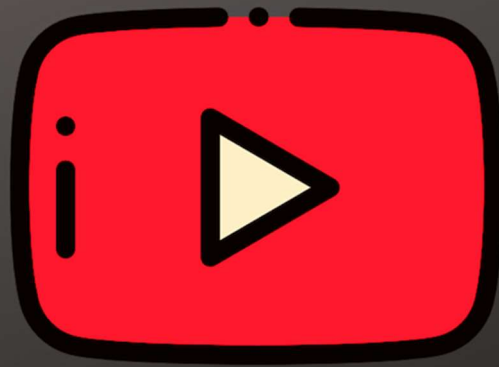


18CSC207J
ADVANCED PROGRAMMING PRACTICE



YouTube Downloader

Project title:

YouTube downloader

BATCH.NO:05

SUBMITTED BY:

ARULNIDHI A(RA2011032020031)

MOSES EVAN(RA2011032020023)

MADHUSUDHANAVAMSI(RA2011032020005)

KARTHIK M(RA2011032020010)

BALACHANDAR(RA2011032020030)

Aim:

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.

Design:

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.

Procedure:

To implement this project, we use basic concept of python, tkinter, pytube library.

- **Tkinter** is a standard GUI library and it is one of the easiest ways to build a GUI application.
- **pytube** used for downloading videos from YouTube.
- **Urllib** package is the **URL handling module for python**. It is used to fetch URLs.
- **PIL** is the Python Imaging Library which provides the python interpreter with image editing capabilities.

To install the required modules run pip installer command on the command line:

- 1.pip install tkinter
- 2.Pip install pytube
- 3.pip install urllib
- 4.pip install pil
- 5.pip install os

Videos are downloaded using ytl-dlp and ffmpeg command line programs.

Source code:

```
from tkinter import*
from tkinter import ttk as ttk
from tkinter.filedialog import askdirectory, asksaveasfile
from tkinter.messagebox import*
from pytube import YouTube
from PIL import Image, ImageTk
import urllib.request
import os

win = Tk()
win.title("Youtube Downloader")
win.iconbitmap(r"data\logo_48.ico")
win.geometry("1024x546")
win.resizable(width=FALSE, height=FALSE)
#Background
bg_img = PhotoImage(file = r"data\bg_1.png")
bg_label = Label(win, image=bg_img)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
#Choose Location
def chooseloc():
    global videoLoc
    path = askdirectory(title="Choose A Directory")
    videoLoc = path
    choose_dir.config(text='Chosen',bg = 'red')
def choose_res():
    global res_got
    res_but.config(text='Selected',bg='red')
    rg = res.get()
    if rg == 'Low audio quality':
        res_got = "139"
    elif rg == 'Best audio quality':
        res_got = "140"
    elif rg == "360p":
```

```

    res_got = "18"
elif rg == '480p':
    res_got = "135+140 --merge-output-format mp4"
elif rg == '720p':
    res_got = "22"
elif rg == '1080p':
    res_got = "137+140 --merge-output-format mp4"
else:
    res_got = "22"

```

#Download

```

def download():
    url = link_entry.get()
    #print(res_got)
    os.system(f"yt-dlp -f{res_got} -o {videoLoc}/{%(title)s-%(id)s.%(ext)s {url}}")
    #res_got = video_object.streams.get_by_resolution(rg)
    #res_got.download(videoLoc)
    down.config(text='Downloaded',bg='black',fg='#00FF00')
    showinfo("Finished Downloading","Your File is Downloaded")

```

#Quit mssg

```

def quit():
    os.remove('temp.jpg')
    showinfo("Bye!", "Thank You For Using Youtube Downloader ◦^_ ^◦ ")

```

#Get Link

```

def details():
    global video_object, res, res_but, choose_dir, down, img
    video_object = YouTube(link_entry.get())
    #title
    title = Label(win, text="Title: ",bg="black", fg="White").place(x=300, y=200)
    title_lab = Label(win, text=video_object.title,bg="black", fg="White").place(x=350, y=200)
    #Author
    author = Label(win, text="Author: ",bg="black", fg="White").place(x=300, y=220)
    author_lab = Label(win, text=video_object.author,bg="black", fg="White").place(x=350, y=220)
    #Length

```

```

length = Label(win, text="Length: ",bg="black", fg="White").place(x=300, y=240)

length_lab = Label(win, text=f'{round(video_object.length/60,2)} mins',bg="black",
fg="White").place(x=350, y=240)

#Views

views = Label(win, text="Views: ",bg="black", fg="White" ).place(x=300, y=260)

v=video_object.views

if v>=1000 and v<100000:

    k = v/1000

    views_lab = Label(win, text=f'{k} Thousand Views',bg="black", fg="White").place(x=350, y=260)
elif v>=100000 and v<10000000:

    k = v/100000

    views_lab = Label(win, text=f'{k} Lakh Views',bg="black", fg="White").place(x=350, y=260)
elif v>=10000000:

    k = v/10000000

    views_lab = Label(win, text=f'{k} Crore Views',bg="black", fg="White").place(x=350, y=260)
else:

    k=v

    views_lab = Label(win, text=f'{k}',bg="black", fg="White").place(x=350, y=260)

#thumbnail

thumb_link = video_object.thumbnail_url

urllib.request.urlretrieve(thumb_link,'temp.jpg')

img_open = Image.open('temp.jpg')

resized_image = img_open.resize((256,192), Image.ANTIALIAS)

img = ImageTk.PhotoImage(resized_image)

thumb = Label(win, image=img, bg = "Black")

thumb.place(x=650,y=230)

#Resolution

res_lab = Label(win, text="Select The Resolution: ",bg="black", fg="White").place(x=300, y=300)

#style for Combobox

style= ttk.Style()

style.theme_use('clam')

style.configure("TCombobox", fieldbackground= "Black", background= "white" , foreground =
"red")

#selected_res = StringVar()

```

```

avail_res = ["Low audio quality","Best audio quality","360p","480p","720p","1080p"]

res = ttk.Combobox(win, values=avail_res, width = 30)

res.place(x=303, y=320)

res_but = Button(win, text="Select",bg = 'black', fg = 'white', width=10, activebackground="red",
activeforeground='white',command=choose_res)

res_but.place(x=512, y=319)

#Choose Dir

choose_dir = Button(win, text = "Choose Directory",bg = 'black', fg = 'white', width=15,
activebackground="red", activeforeground='white',command=chooseloc)

choose_dir.place(x=303, y=360)

#Download Button

down = Button(win, text = "Download",bg = 'black', fg = 'white', width=15,
activebackground="red", activeforeground='white',command=download)

down.place(x=303, y=395)

#copy from clipboard

def clip():

    clipboard = win.clipboard_get()

    print(clipboard)

    link_entry.insert(0,clipboard)

#Label

paste_but = Button(win,text = 'Paste',command=clip, bg = 'black', fg = 'white', width=8,
activebackground="red", activeforeground='white')

paste_but.place(x = 390, y=148)

link_lab = Label(win, text = "Enter The Link:",fg = "White", bg = "Black")

link_lab.place(x=300,y=150)

link_entry = Entry(win, width=50, bg="black", fg="White", insertbackground="red")

link_entry.place(x=470,y=150)

go_button = Button(win, text = "Go", command=details, bg = 'black', fg = 'white', width=10,
activebackground="red", activeforeground='white').place(x=790, y=148,)

#quit

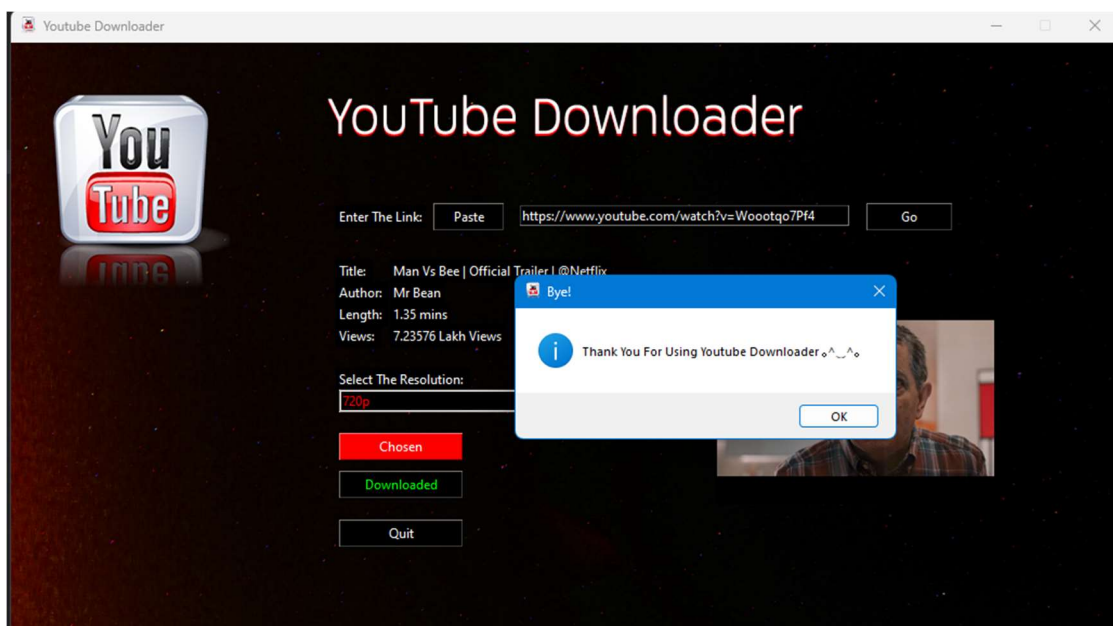
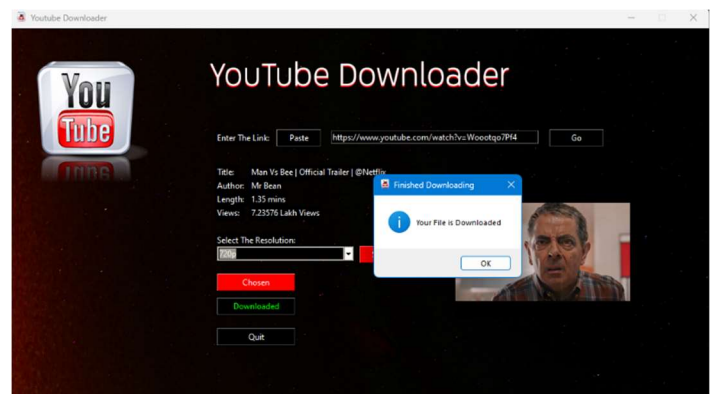
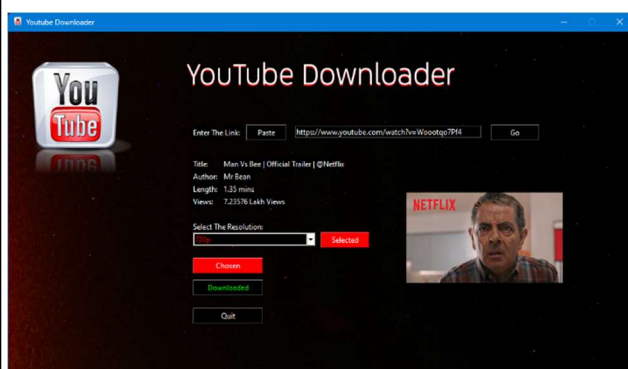
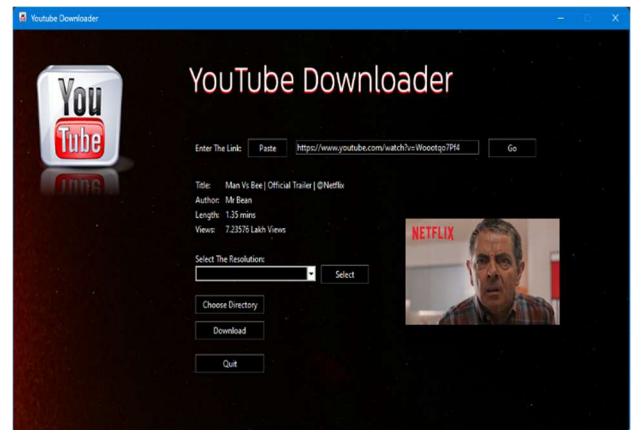
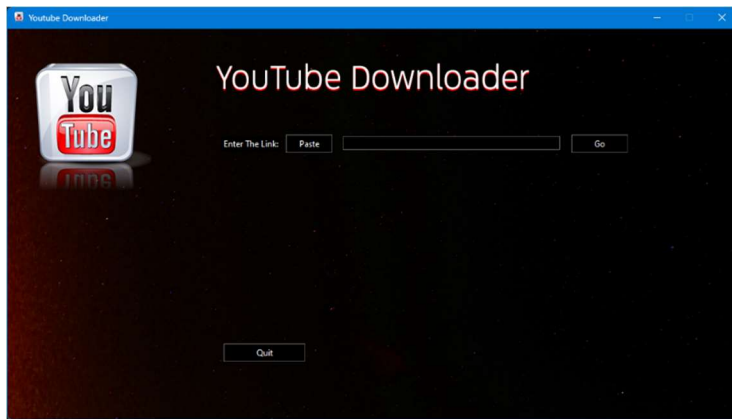
quit_button = Button(win, text = "Quit",bg = 'black', fg = 'white', width=15, activebackground="red",
activeforeground='white',command=lambda: [quit(), win.destroy()]).place(x=303, y=440)

win.protocol('WM_DELETE_WINDOW', lambda: [quit(), win.destroy()])

win.mainloop()

```

SCREENSHOTS:



RESULT:

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 pytube library to download videos from YouTube.