YouTube\_Video Downloader

#import necessary packages

from tkinter import \*

import pathlib

from tkinter.ttk import Progressbar

from pytube import YouTube

from PIL import Image,ImageTk

import os

import requests

import re

from tkinter import messagebox, filedialog

import threading

# Defining CreateWidgets() function

# to create necessary tkinter widgets

def Widgets():

url= Label(root, text="YouTube link :", bg="red",fg="white" )

url.place(x=50,y=60,height=30)

linkTxt = Entry(root,textvariable=video\_Link)

linkTxt.place(x=160,y=60,height=30,width=250)

destination\_lbl=Label(root, text="Destination :", bg="red",fg="white")

destination\_lbl.place(x=50,y=100,height=30)

destination\_Txt=Entry(root, textvariable=download\_Path)

destination\_Txt.place(x=160,y=100,height=30,width=200)

browse\_Btn=Button(root, text="Browse", command=Browse, bg="blue",fg="white")

browse\_Btn.place(x=370,y=105)

Download\_Btn=Button(root, text="Download", command=Download,bg="green",fg="white",activebackground="orange")

Download\_Btn.place(x=180,y=180,height=35,width=80)

cancel\_btn=Button(root,text="cancel",command=quit,bg="grey",fg="white",activebackground="purple")

cancel\_btn.place(x=300,y=180,height=35,width=100)

progress\_bar = Progressbar(root, orient=HORIZONTAL, length=590, mode='determinate')

progress\_bar.place(x=40, y=250, height=25, width=400)

label=Label(root,text="Status",bg="brown",fg="white").place(x=200,y=280)

# Defining Browse() to select a

# destination folder to save the video

def Browse():

#download\_Directory = filedialog.askdirectory(initialdir=pathlib.Path.cwd())

download\_Directory1=filedialog.asksaveasfilename(initialdir=pathlib.Path.cwd())

# Displaying the directory in the directory

# textbox

#download\_Path.set(download\_Directory)

download\_Path.set(download\_Directory1)

def getStandardSize(size):

itme=['bytes','KB','MB','GB','TB']

for x in itme:

if size < 1024.0:

return "%3.1f %s" % (size,x)

size/=1024.0

return size

# Defining Download() to download the video

def Download():

# getting user-input Youtube Link

url= video\_Link.get()

# downloadThread = threading.Thread(target=lambda: addDownloadItem(url))

#downloadThread.start()

# select the optimal location for

# saving file's

download\_Folder = download\_Path.get()

# Creating object of YouTube()

getVideo = YouTube(url)

# Getting all the available streams of the

# youtube video and selecting the first

# from the

videoStream = getVideo.streams.first()

# Downloading the video to destination

# directory

videoStream.download(download\_Folder)

# Displaying the message

messagebox.showinfo("SUCCESSFULLY", "DOWNLOADED AND SAVED IN\n"

+ download\_Folder)

def addDownloadItem(url):

if url!=None:

req=requests.get(url,stream=True)

if "Content-Length" in req.headers:

total\_size=req.headers['Content-Length']

else:

total\_size=None

if "Content-Disposition" in req.headers.keys():

fname=re.findall("filename=(.+)",req.headers["Content-Disposition"])[0]

else:

fname=url.split("/")[-1]

progress=Progressbar(root)

progress['value']=0

progress.place(x=10,column=100,sticky="nsew",mode="intermediate")

labelPercentage=Label(root,text="0 %",padx="5",anchor="w",bg="#E67E22",fg="white")

labelPercentage.grid(row=0,column=2)

labelsize=Label(root,text="0 KB",padx="5",anchor="w",bg="#E67E22",fg="white")

labelsize.grid(row=1,column=2)

with open(fname,"wb") as obj:

for chunk in req.iter\_content(chunk\_size=1024):

if chunk:

obj.write(chunk)

current\_size=os.path.getsize(fname)

labelsize.config(text=str(getStandardSize(current\_size)))

if total\_size!=None:

percentge=round((int(current\_size)/int(total\_size))\*100)

labelPercentage.config(text=str(percentge)+" %")

progress['value']=percentge

else:

percentge="Infinte"

progress.config(mode="indeterminate")

progress.start()

labelPercentage.config(text=str(percentge)+" %")

if total\_size!=None:

current\_size=os.path.getsize(fname)

labelsize.config(text=str(getStandardSize(current\_size)))

labelPercentage.config(text=str(percentge) + " %")

percentge=round((int(current\_size)/int(total\_size))\*100)

progress['value']=percentge

else:

current\_size=os.path.getsize(fname)

labelsize.config(text=str(getStandardSize(current\_size)))

labelPercentage.config(text="100 %")

progress['value'] = 100

# Creating object of tk class

root = Tk()

icon=PhotoImage(file="icon1.png")

root.iconphoto(False,icon)

root.geometry("500x400")

root.resizable(False,False)

root.title("YouTube Video Downloader")

root.config(background="black")

label1=Label(root,text="Video Downloader",font="Dosis,10,Bold",bg="red",fg="white",width=40,borderwidth=10).place(x=50,y=10,height=40)

# Creating the tkinter Variables

video\_Link = StringVar()

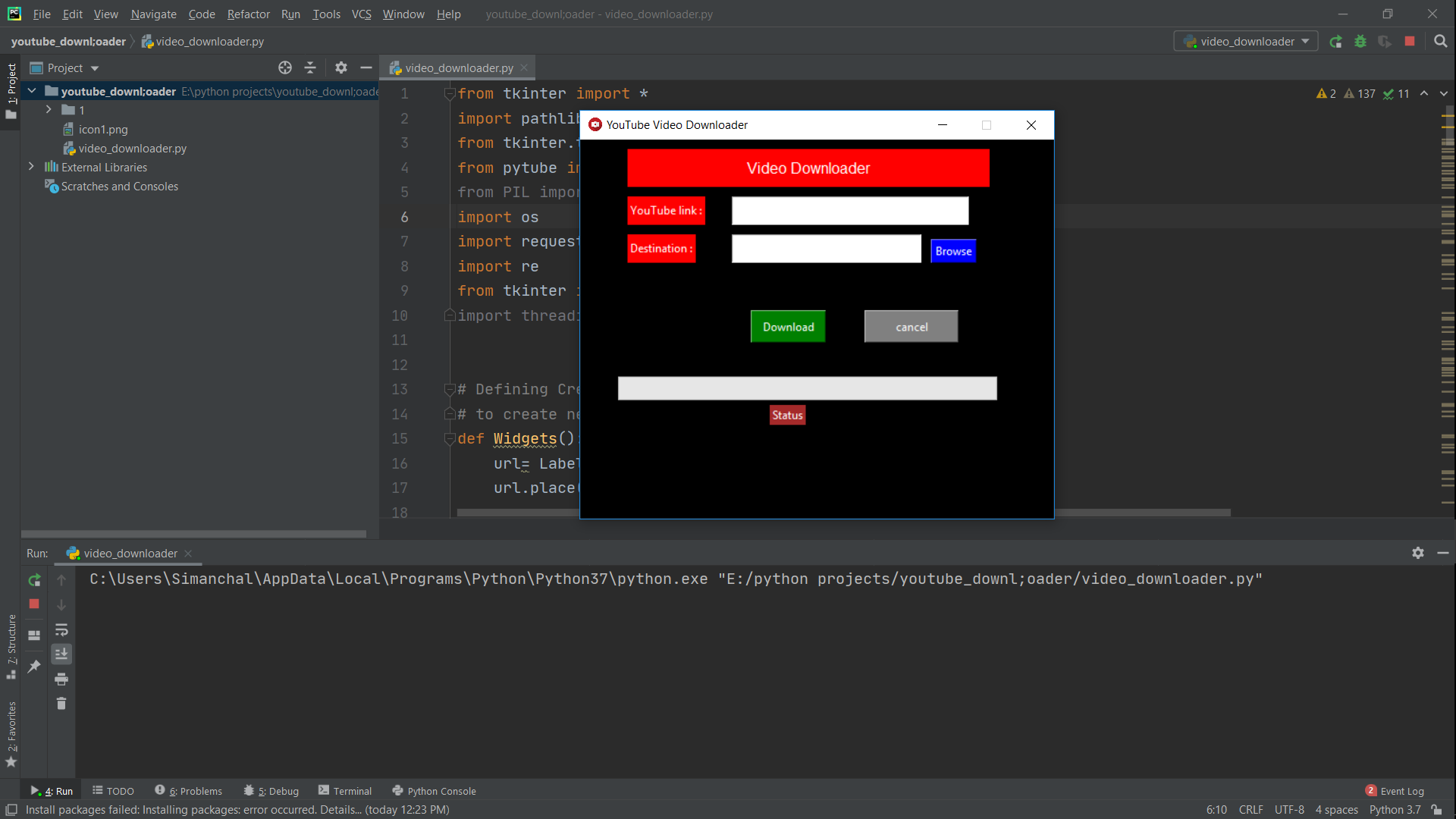
download\_Path = StringVar()

# Calling the Widgets() function

Widgets()

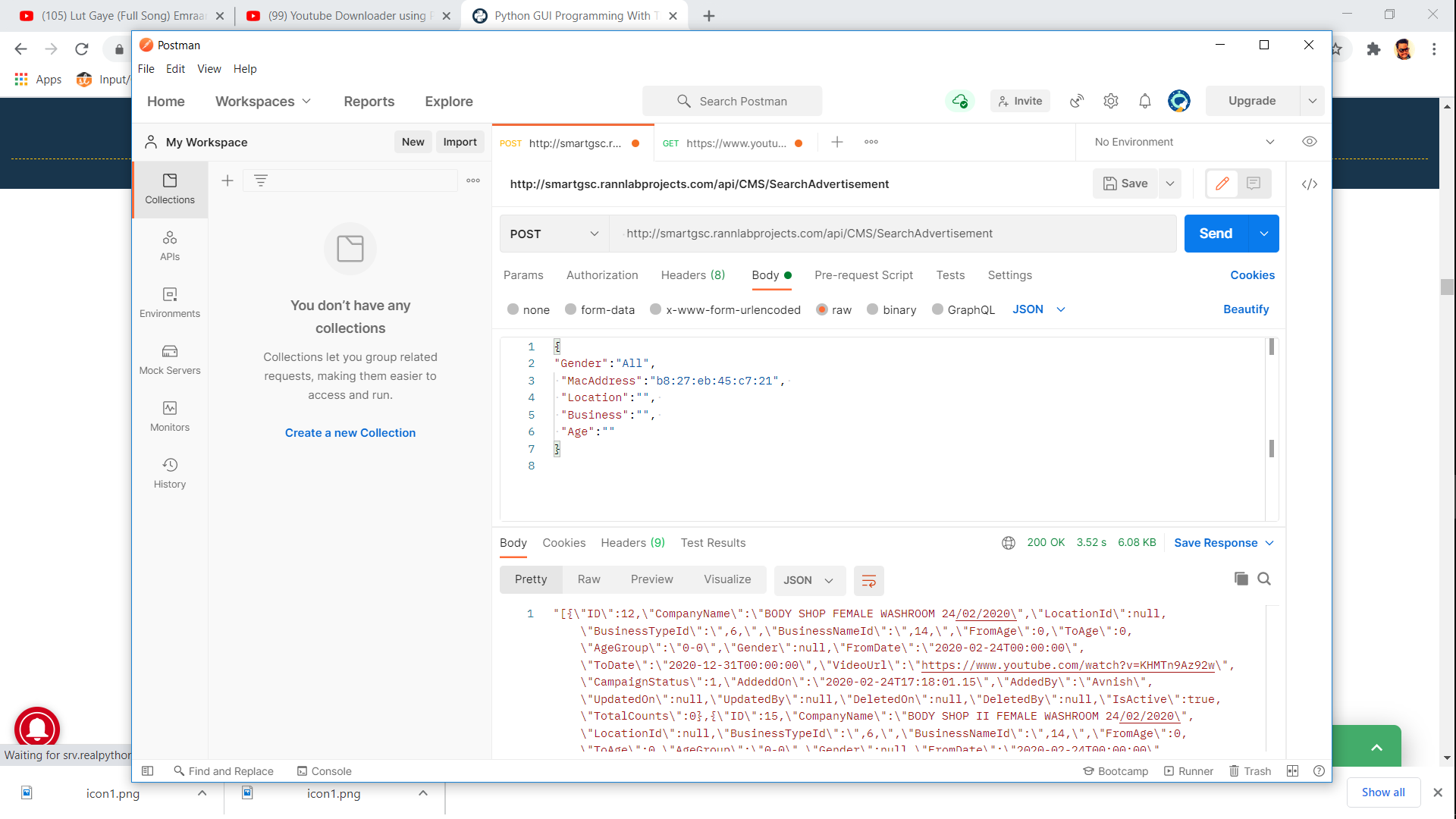
root.mainloop()

OUTPUT:



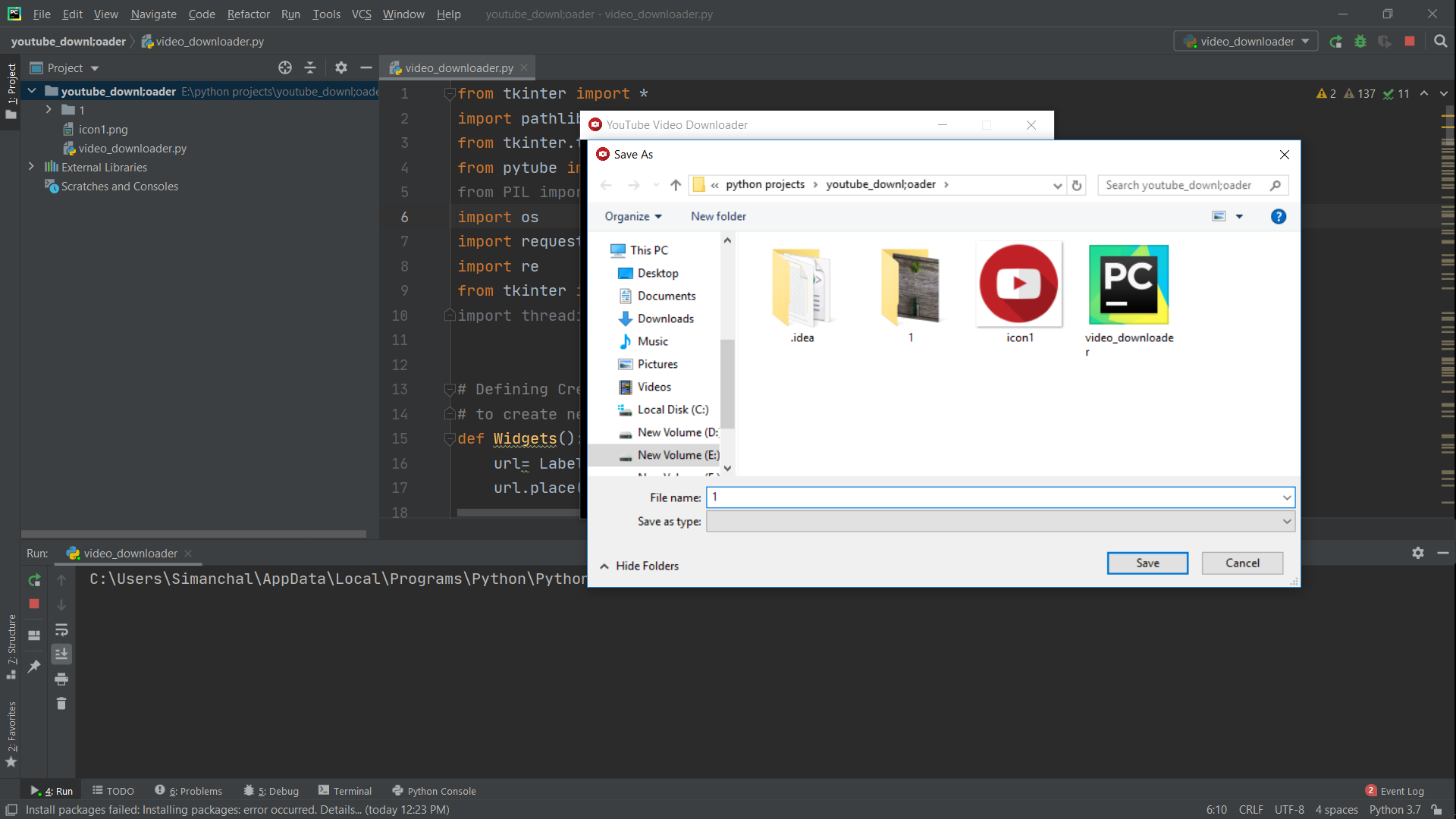
**Step-2:**

Open postman app and set the request in POST method and paste the urls API then click the Send button.



Step-3:

Copy the urls from postman and paste in video\_downloader entry label and set the file name then click download button.



Click download button :

