Create a Tkinter project to the below functionalities:

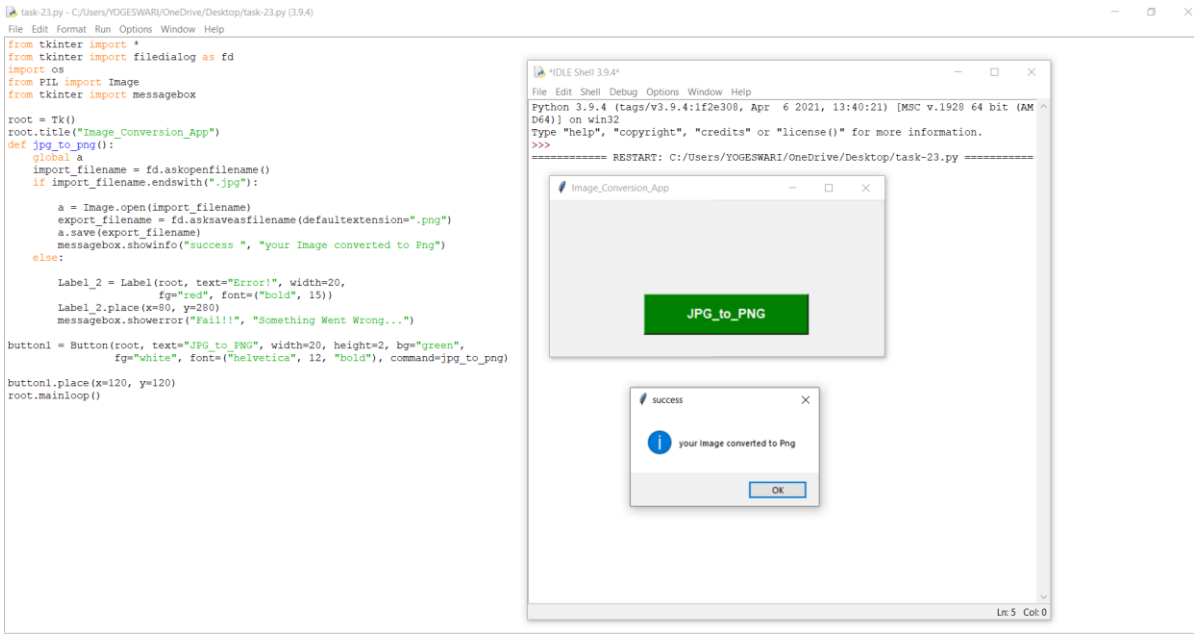
1.Create a browse option with a specific folder which has all the JPEG Files & create a Convert button to convert the image from JPEG to PNG – Basic Image converter App

```
from tkinter import *
from tkinter import filedialog as fd
import os
from PIL import Image
from tkinter import messagebox
root = Tk()
root.title("Image_Conversion_App")
def jpg_to_png():
    global a
    import_filename = fd.askopenfilename()
    if import_filename.endswith(".jpg"):
        a = Image.open(import_filename)
        export_filename = fd.asksaveasfilename(defaultextension=".png")
        a.save(export_filename)
        messagebox.showinfo("success ", "your Image converted to Png")
    else:
        Label_2 = Label(root, text="Error!", width=20,
                        fg="red", font=("bold", 15))
        Label_2.place(x=80, y=280)
        messagebox.showerror("Fail!!", "Something Went Wrong...")

button1 = Button(root, text="JPG_to_PNG", width=20, height=2, bg="green",
                 fg="white", font=("helvetica", 12, "bold"), command=jpg_to_png)

button1.place(x=120, y=120)
```

root.mainloop()



2.Create another button as ‘fetch button’ and have a functionality of fetching the weather on a given location in text box

from tkinter import *

import tkinter as tk

import requests

HEIGHT = 500

WIDTH = 500

def test_function(entry):

print("This is the entry:", entry)

def format_response(weather):

try:

name = weather['name']

desc = weather['weather'][0]['description']

```
                temp = weather['main']['temp']


                if Cbutton:
                        final_str = 'Place: %s \nWeather: %s \nTemperature: %s' % (name, desc,
temp)
        except:
                final_str = 'There was a problem retrieving that information'
        return final_str


def get_weather(city):
        weather_key = 'a4aa5e3d83ffefaba8c00284de6ef7c3'
        url = 'https://api.openweathermap.org/data/2.5/weather'
        params = {'APPID': weather_key, 'q': city, 'units': 'Imperial'}
        response = requests.get(url, params=params)
        weather = response.json()
        label['text'] = format_response(weather)


def get_weatherM(city):
        weather_key = 'a4aa5e3d83ffefaba8c00284de6ef7c3'
        url = 'https://api.openweathermap.org/data/2.5/weather'
        params = {'APPID': weather_key, 'q': city, 'units': 'Metric'}
        response = requests.get(url, params=params)
        weather = response.json()
        label['text'] = format_response(weather)
root = tk.Tk()
canvas = tk.Canvas(root, height=HEIGHT, width=WIDTH,bg='black')
canvas.pack()
frame = tk.Frame(root, bg='white', bd=25)
frame.place(relx=0.5, rely=0.1, relwidth=0.75, relheight=0.15, anchor='n',)
entry = tk.Entry(frame, font=30)
entry.place(relwidth=0.5, relheight=1)
```

Cbutton = tk.Button(frame, text="Fetch Tempearture", font='times 13', command=lambda: get_weatherM(entry.get()))

Cbutton.place(relx=0.6,rely=0,relheight=1, relwidth=0.45)

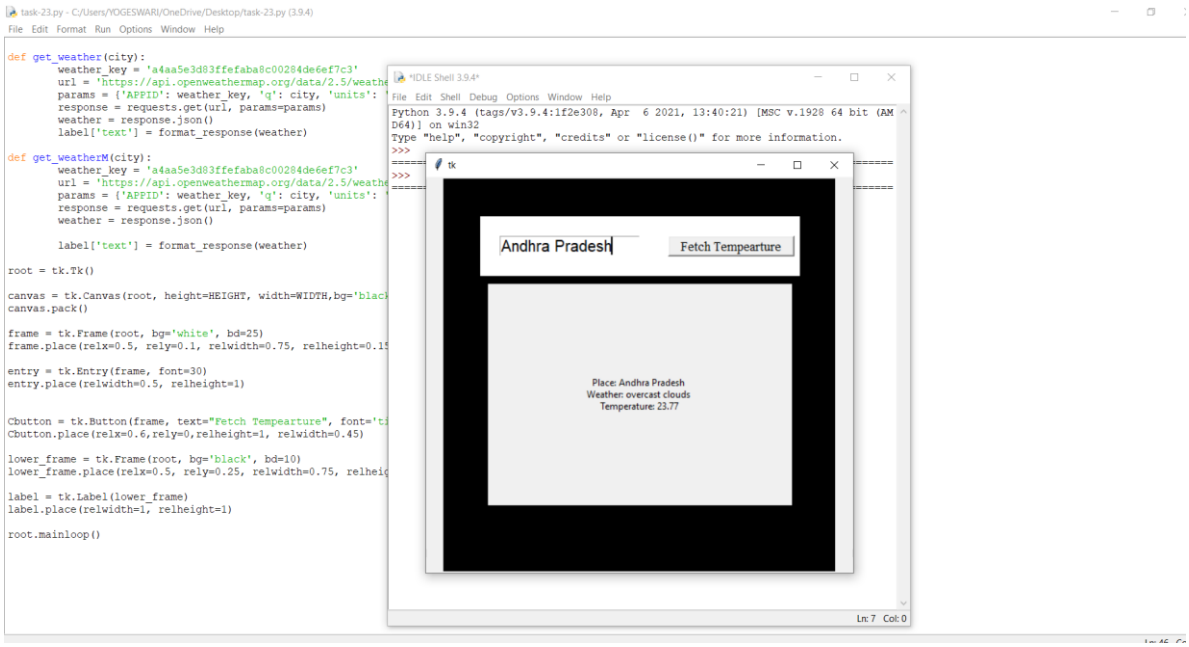
lower_frame = tk.Frame(root, bg='black', bd=10)

lower_frame.place(relx=0.5, rely=0.25, relwidth=0.75, relheight=0.6, anchor='n')

label = tk.Label(lower_frame)

label.place(relwidth=1, relheight=1)

root.mainloop()



3.Create two browse button and place the .pdf file for the buttons and create a merge pdf option - Watermark Merger App

import tkinter as tk

from tkinter.filedialog import askopenfilename

from PyPDF2 import PdfFileMerger, PdfFileReader

from pathlib import Path

filelist = []

merger = PdfFileMerger()

def open_file(files):

```
    filepath = askopenfilename(
        filetypes=[("PDF Files", "*.pdf"), ("All Files", "*.*")] )
    if not (filepath and Path(filepath).exists()):
        return
    files.append(filepath)
    lbl_items["text"] = '\n'.join(str(f) for f in files)
    if len(files) >= 2 and btn_merge['state'] == "disabled":
        btn_merge["state"] = "normal"
def merge_pdfs(files):
    for f in files:
        merger.append(PdfFileReader(open(f, "rb")))
    output_filename = ent_output_name.get()
    if not output_filename:
        output_filename = "Untitled.pdf"
    elif ".pdf" not in output_filename:
        output_filename += ".pdf"
    merger.write(output_filename)
window = tk.Tk()
window.title("PDFMerger Tk")
window.geometry("500x500")
window.resizable(0, 0)
fr_bg1 = tk.Frame(window, bd=3)
lbl_open = tk.Label(fr_bg1, text="Select files to merge ")
lbl_open.grid(row=0, column=0, sticky="ew", padx=5, pady=5)
btn_open = tk.Button(fr_bg1, text="Open file", bg='red', fg='white', font=('helvetica', 12, 'bold'),
command=lambda: open_file(filelist))
btn_open.grid(row=1, column=0, sticky="ew", padx=5)
lbl_items = tk.Label(fr_bg1, text="")
lbl_items.grid(row=2, column=0, pady=5)
fr_bg1.pack()
fr_bg2 = tk.Frame(window, bd=3)
```

```
lbl_to_merge = tk.Label(fr_bg2, text="Now save as a pdf by giving name\n the file get stored in root folder")

lbl_to_merge.grid(row=0, column=0, sticky="ew", padx="5", pady="5")


ent_output_name = tk.Entry(master=fr_bg2, width=7)

ent_output_name.grid(row=1, column=0, sticky="ew")


btn_merge = tk.Button(fr_bg2, bg='red', font=('helvetica', 12, 'bold'),

                      text="Merge PDF",

                      state="disabled",

                      command=lambda: merge_pdfs(filelist))

btn_merge.grid(row=2, column=0, sticky="ew", padx=5, pady=5)

fr_bg2.pack()


btn_exit = tk.Button(window, text="Exit", command=window.destroy, bd=2, bg='royalblue', fg='black',

                     font=('helvetica', 12, 'bold'), )

btn_exit.pack(side=tk.BOTTOM, fill=tk.BOTH, expand=tk.FALSE)


if __name__ == "__main__":

   window.mainloop()
```

task-23.py - C:/Users/YOGESWARI/OneDrive/Desktop/task-23.py (3.9.4)

File Edit Format Run Options Window Help

```
    output_filename = ent_output_name.get()

    if not output_filename:
        output_filename = "Untitled.pdf"
    elif ".pdf" not in output_filename:
        output_filename += ".pdf"
    merger.write(output_filename)



window = tk.Tk()
window.title("PDFMerger Tk")
window.geometry("500x500")
window.resizable(0, 0)

fr_bg1 = tk.Frame(window, bd=3)
lbl_open = tk.Label(fr_bg1, text="Select files to merge ")
lbl_open.grid(row=0, column=0, sticky="ew", padx=5, pady=5)

btn_open = tk.Button(fr_bg1, text="Open file", bg='red', fg='white', font=('
btn_open.grid(row=1, column=0, sticky="ew", padx=5)
lbl_items = tk.Label(fr_bg1, text="")
lbl_items.grid(row=2, column=0, pady=5)
fr_bg1.pack()


fr_bg2 = tk.Frame(window, bd=3)
lbl_to_merge = tk.Label(fr_bg2, text="Now save as a pdf by giving name\n the
lbl_to_merge.grid(row=0, column=0, sticky="ew", padx="5", pady="5")

ent_output_name = tk.Entry(master=fr_bg2, width=7)
ent_output_name.grid(row=1, column=0, sticky="ew")

btn_merge = tk.Button(fr_bg2, bg='red', font=('helvetica', 12, 'bold'),
                      text="Merge PDF",
                      state="disabled",
                      command=lambda: merge_pdfs(filelist))
btn_merge.grid(row=2, column=0, sticky="ew", padx=5, pady=5)
fr_bg2.pack()

btn_exit = tk.Button(window, text="Exit", command=window.destroy, bd=2, bg='
                     font=('helvetica', 12, 'bold'), )
btn_exit.pack(side=tk.BOTTOM, fill=tk.BOTH, expand=tk.FALSE)

if __name__ == "__main__":
    window.mainloop()
```

IDLE Shell 3.9.4

File Edit Shell Debug Options Window Help

```
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr  6 2021, 13:40:21) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/YOGESWARI/OneDrive/Desktop/task-23.py ==========
```