Mini project:

Create a Tkinter project with 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

In [6]:

```
# import all prerequisite
from tkinter import *
from tkinter import filedialog as fd
import os
from PIL import Image
from tkinter import messagebox

root = Tk()

# naming the GUI interface to image_conversion_APP
root.title("Image_Conversion_App")
l1=Label(root,text='To convert your Image .jpg to .png click the button')
# creating the Function which converts the jpg_to_png
def jpg_to_png():
    global im1

    # import the image from the folder
    import_filename = fd.askopenfilename()
    if import_filename.endswith(".jpg"):

        im1 = Image.open(import_filename)

        # after converting the image save to desired
        # location with the Extension .png
        export_filename = fd.asksaveasfilename(defaultextension=".png")
        im1.save(export_filename)


        # displaying the Messaging box with the Success
        messagebox.showinfo("success ", "your Image converted to Png")
    else:
         # if Image select is not with the Format of .jpg
        # then display the Error
        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="blue",
                 fg="white", font=("helvetica", 12, "bold"), command=jpg_to_png)
button1.place(x=120, y=120)
l1.place(x=100,y=100)
root.geometry("500x500+400+200")
root.mainloop()
```

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

In [15]:

```
from configparser import ConfigParser
import requests
from tkinter import *
from tkinter import messagebox

# extract key from the
# configuration file
config_file = "config.ini"
config = ConfigParser()
config.read(config_file)
api_key = config['vinita']['api']
url = 'http://api.openweathermap.org/data/2.5/weather?q={}&appid={}'
# explicit function to get
# weather details
def getweather(city):
    result = requests.get(url.format(city, api_key))

    if result:
        json = result.json()
        city = json['name']
        country = json['sys']
        temp_kelvin = json['main']['temp']
        temp_celsius = temp_kelvin-273.15
        weather1 = json['weather'][0]['main']
        final = [city, country, temp_kelvin,
                 temp_celsius, weather1]
        return final
    else:
        print("NO Content Found")


# explicit function to
# search city
def search():
    city = city_text.get()
    weather = getweather(city)
    if weather:
        location_lbl['text'] = '{} ,{}'.format(weather[0], weather[1])
        temperature_label['text'] = str(weather[3])+"   Degree Celsius"
        weather_l['text'] = weather[4]
    else:
        messagebox.showerror('Error', "Cannot find {}".format(city))


# Driver Code
# create object
app = Tk()
# add title
app.title("Weather App")
# adjust window size
app.geometry("300x300")
# add labels, buttons and text
city_text = StringVar()
city_entry = Entry(app, textvariable=city_text)
city_entry.pack()
Search_btn = Button(app, text="Search Weather",
                    width=12, command=search)
Search_btn.pack()
location_lbl = Label(app, text="Location", font={'bold', 20})
```

```
location_lbl.pack()
temperature_label = Label(app, text="")
temperature_label.pack()
weather_l = Label(app, text="")
weather_l.pack()
app.mainloop()
```

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

In [17]:

```
import tkinter as tk
from tkinter.filedialog import askopenfilename
from PyPDF2 import PdfFileMerger, PdfFileReader
from pathlib import Path

filelist = []

# initiate merger Object
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)
    # list out all filenames
    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)

# create desktop GUI
window = tk.Tk()
window.title("PDFMerger Tk")
window.geometry("500x500")
# not allowed resizing x y direction
window.resizable(0,0)

# --- Ask open files ---
fr_bg1 = tk.Frame(window, bd=3)
lbl_open = tk.Label(fr_bg1, text="Please choose PDFs to join: (2 and above)")
lbl_open.grid(row=0, column=0, sticky="ew", padx=5, pady=5)

btn_open = tk.Button(fr_bg1, text="Open file",bg='royalblue', fg='white',font=('helvetica',
                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()

# --- Button to merge PDFs ---
fr_bg2 = tk.Frame(window, bd=3)
lbl_to_merge = tk.Label(fr_bg2, text="Merge selected files (in PDF)")
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='royalblue',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=
btn_exit.pack(side=tk.BOTTOM, fill=tk.BOTH, expand=tk.FALSE)

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

In [ ]: