F.E.T AGRA COLLEGE, AGRA

AFFILIATED TO

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY,

U.P. (LUCKNOW)

Mini Project Report

On

"Smart Image Utility System"

SUBMITTED BY

Dheeraj Pal (2100020100035)

Lakshya Sharma (2100020100053)

Yogesh Kumar (2100020100117)

2nd year Computer Science and Engineering

SUBMITTED TO

Dr. Anuj Parashar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CONTENTS

ACKNOWLEDGEMENT

The completion of this project, the Smart Image Utility System, would not have been possible without the support and guidance of several individuals. We would like to express our sincere gratitude to the following people for their invaluable contributions:

- Our project guide Dr. Anuj Parashar, who provided us with invaluable technical support and guidance throughout the project.
- The Department Of Computer Science And Engineering at Faculty of Engineering and Technology, for providing us with the necessary resources and facilities to complete this project.
- Our classmates and peers, who have been a constant source of motivation and support.

We are deeply thankful to all of them for their support and for helping us to make this project a success.

INTRODUCTION

The Smart Image Utility System is a college mini project that aims to solve real-life problems related to image management. This system comprises of four utilities: Duplicate Image Deleter, Text Extractor, QR Code Scanner, and Image Compressor. Each utility addresses a specific issue in image management and provides a user-friendly solution. The Duplicate Image Deleter helps remove redundant images, the Text Extractor extracts text from images, the QR Code Scanner scans and decodes QR codes, and the Image Compressor reduces the size of images without sacrificing quality. This system is designed to simplify and streamline the process of image management for users.

****

INTRODUCTION POINTS

****

DEVELOPMENT TOOLS:-

.....

We have made our Project with:

1) Language: Python.

2) We have used Sublime Text as our editor.

3) Modules we have used are : tkinter(for GUI),PIL/Pillow for image handling),pytesseract(for ocr reading),open-cv(for images),numpy(for handling multi-dimensional array),webbrowser(to open links),os(for operating system feautres)

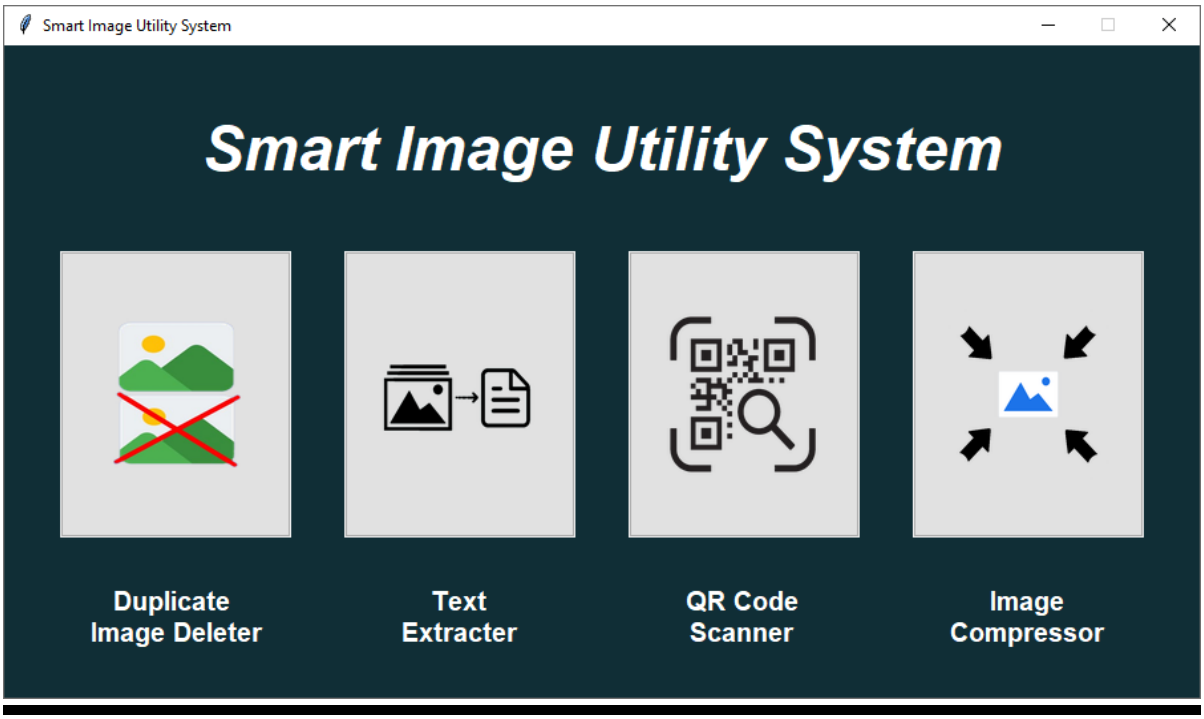
4) Web-Browser: Google Chrome

FEATURES:-

- User friendly
- Portable
- Real life problem solving
- Easy to understand.
- JPG, PNG, jpeg type of file sharing support.

SNAPSHOTS

➢ HOME PAGE

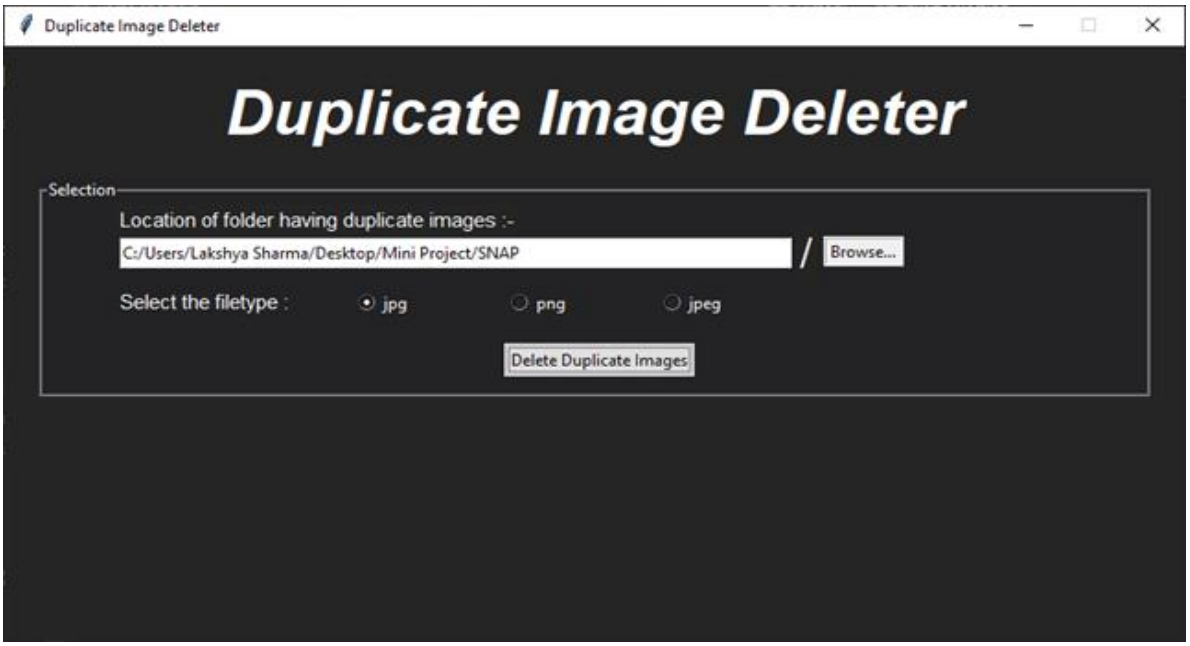


- SMART IMAGE UTILITY SYSTEM provides following features :

1. Duplicate Image Deleter
2. Text Extracter
3. QR Code Scanner
4. Image Compressor

- User can choose any of these features as per their requirement

➢ DUPLICATE IMAGE DELETER

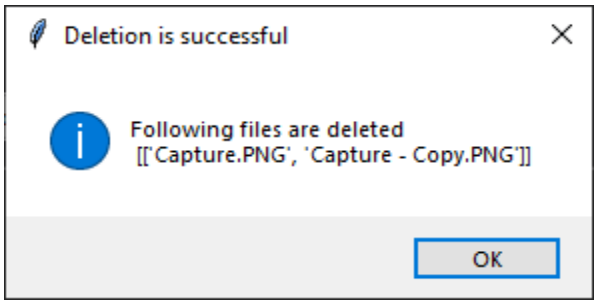


- The above window shows the Duplicate Image Deleter Utility.

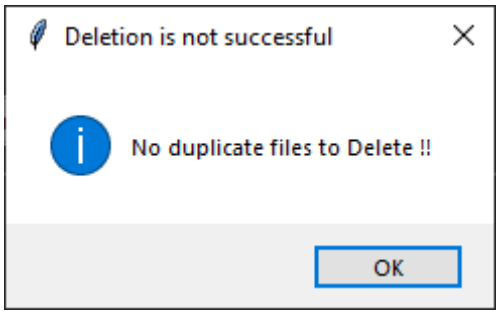
- User can choose the folder having duplicate images either by entering its path into the entry box or by using browse button and can select the respective file type by the radio buttons.

- On clicking "Delete Duplicate Images" button the two possible outputs will be shown as follows:-

- The first window specifies that the duplicate have deleted.



- The second window specifies that the given folder does not contains duplicate images.



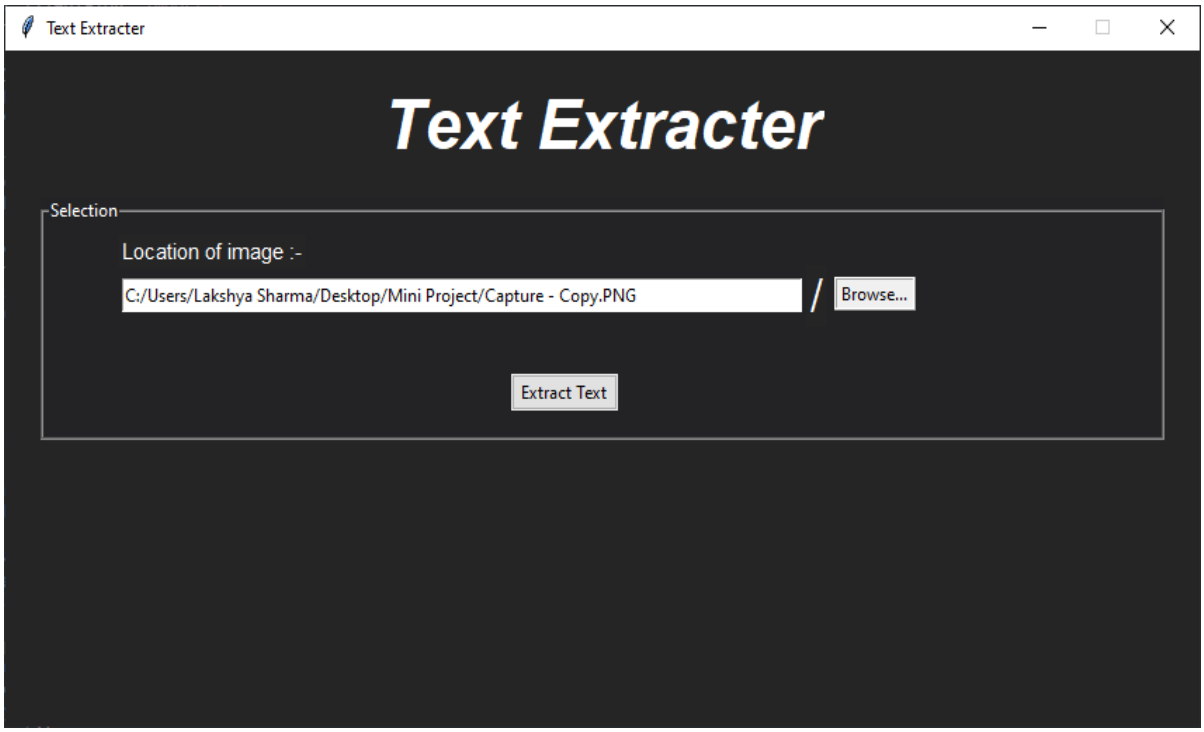
Description:-

1.This program converts every image to its grey scale image and then compare them with each other.

2.Images having similar gray scale are considered as duplicate images and then it keeps the original copy of image and deletes the other copies.

3.This program can delete duplicate files having "JPG", "PNG" and "JPEG" extensions.

➢ TEXT EXTRACTER



- The above window shows the Text Extracter Utility.

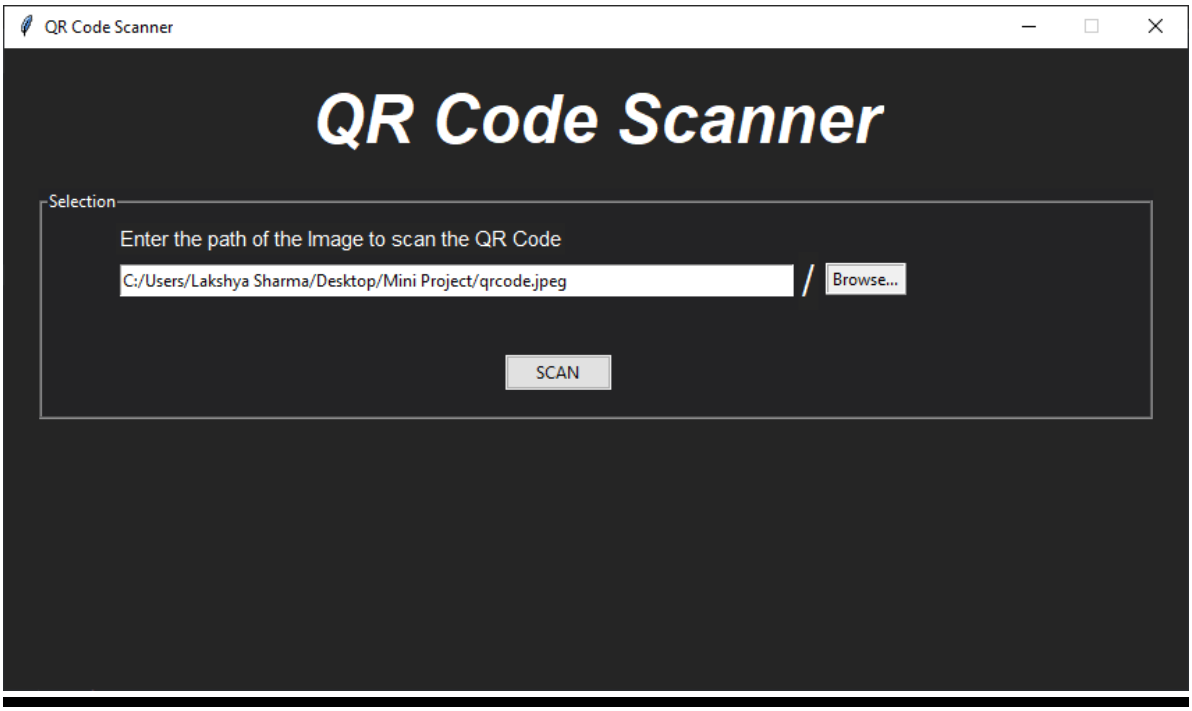
- User can choose the image having text either by entering its path into the entry box or by using browse button.

- After the desired path is entered in the entry box then click on "Extract Text" button.

- Extracted text will be shown into an editable text box.

- User can copy the extracted text just by clicking on the "Copy To Clipboard" button.

➢ QR CODE SCANNER



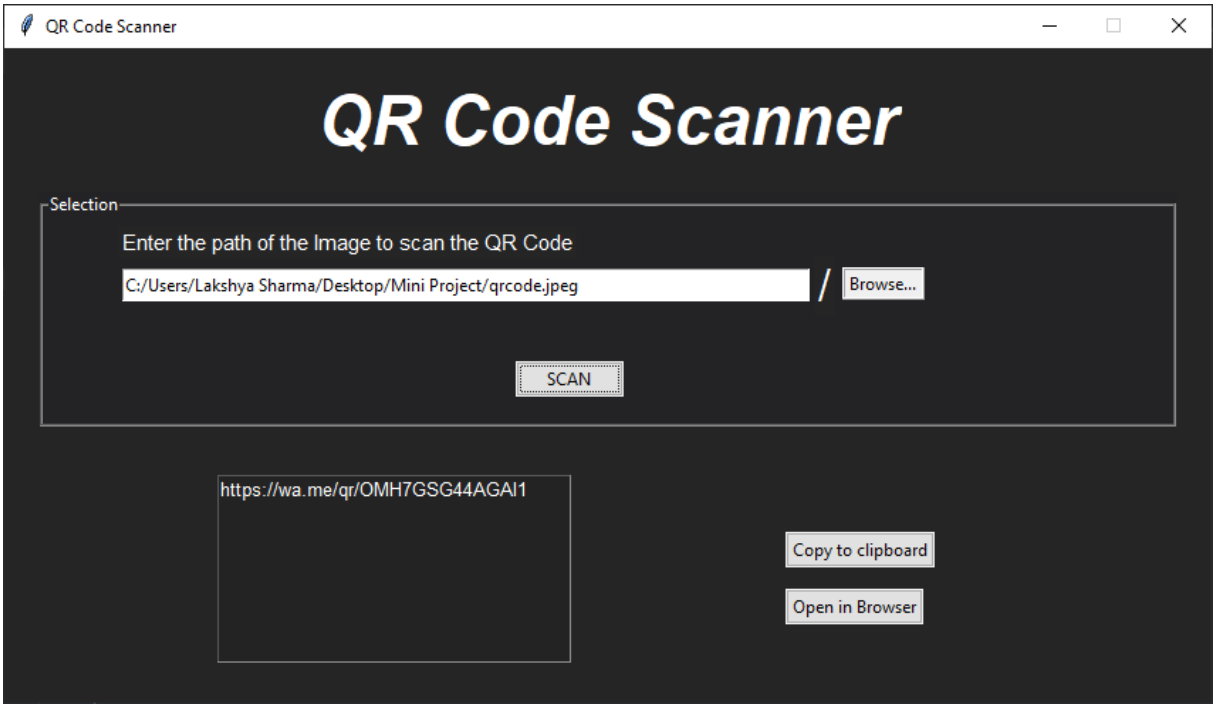
- The above window shows the Duplicate QR Code Sanner Utility.

- User can choose the QR code image either by entering its path into the entry box or by using browse button.

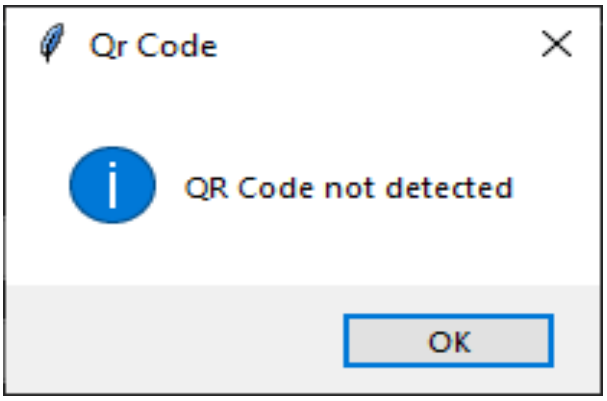
- After the desired path is entered in the entry box then click on "SCAN" button.

- Decoded Data will be shown into an editable text box.

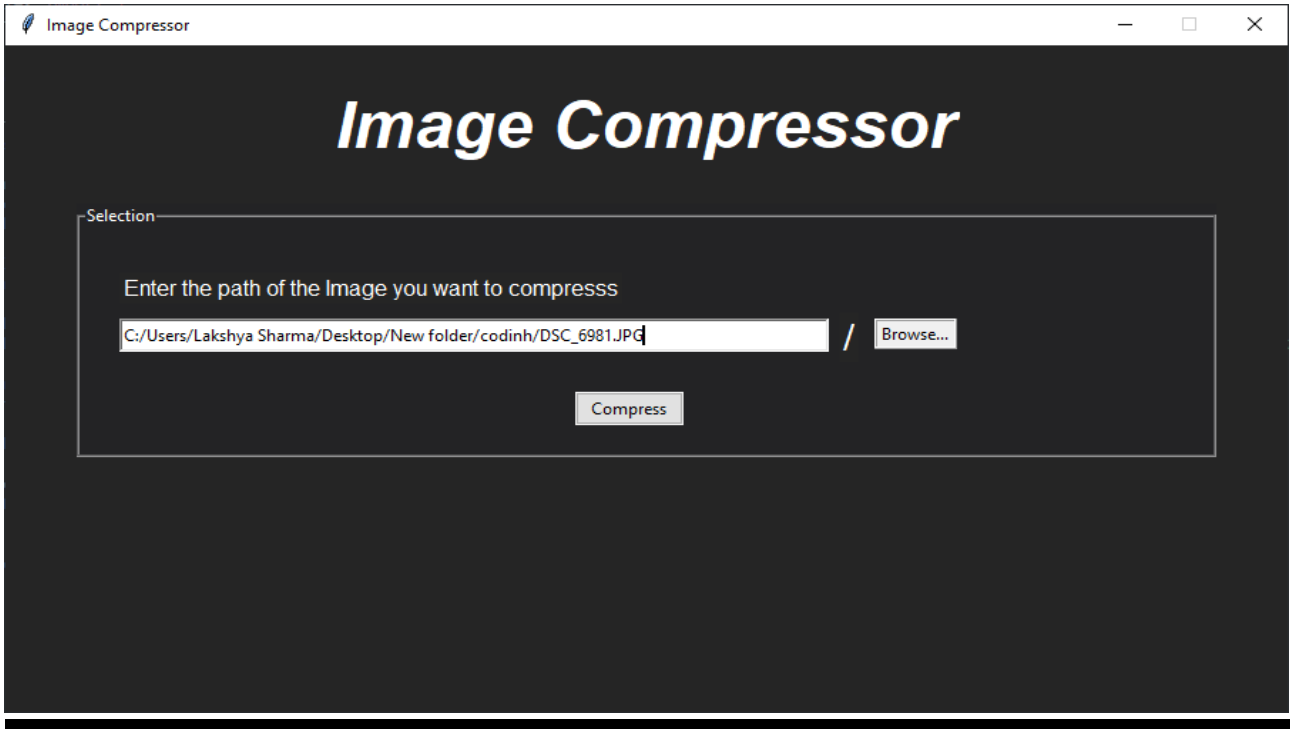
- After the data is decoded from the QR code image user can either copy the data by clicking on the "Copy to clipboard " button or can open it into the default internet browser.



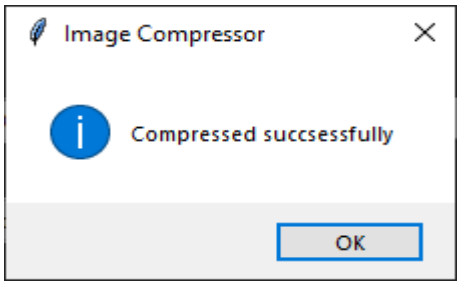
- Otherwise QR Code not detected message will be shown in a pop-up window.



➢ Image Compressor



- The above window shows the Image Compressor Utility.
- User can choose the image that user want to compress either by entering its path into the entry box or by using browse button.
- After the desired path is entered in the entry box then click on "Compress" button.
- After the compression of image a pop-up message will be shown.



SOURCE CODE:-

➢ HOME PAGE GUI

```
from tkinter import *
from tkinter import ttk
from subprocess import call
from PIL import Image,ImageTk
import PIL

def click_qr():
    call(["python", "gui_qrscan.py"])
def click_imgcomp():
    call(["python", "gui_imgcomp.py"])
def click_text():
    call(["python", "gui_textextracter.py"])
def click_dup():
    call(["python","gui_dup.py"])



root=Tk()
root.title("Smart Image Utility System")
root.geometry("900x501")
root.config(bg="#102e36")
root.resizable(width=False, height=False)
label=ttk.Label(text="Smart Image Utility System",font="Helvetica 35
bold italic",background="#102e36",foreground="white").pack(pady=50)
f1=Frame(background="#102e36")



img1 = Image.open("ico1.png")
img1 = img1.resize((120,130), PIL.Image.ANTIALIAS)
photo1 = ImageTk.PhotoImage(img1)
```

```
img2 = Image.open("ico2.png")
img2 = img2.resize((120,130), PIL.Image.ANTIALIAS)
photo2 = ImageTk.PhotoImage(img2)

img3 = Image.open("ico3.png")
img3 = img3.resize((120,130), PIL.Image.ANTIALIAS)
photo3 = ImageTk.PhotoImage(img3)

img4 = Image.open("ico4.png")
img4 = img4.resize((120,130), PIL.Image.ANTIALIAS)
photo4 = ImageTk.PhotoImage(img4)


b1=ttk.Button(f1,command=click_dup,image=photo1).pack(side=LEFT,pad
y=0,padx=20,ipady=40,ipadx=22)
b2=ttk.Button(f1,command=click_text,image=photo2).pack(side=LEFT,pa
dy=0,padx=20,ipady=40,ipadx=22)
b3=ttk.Button(f1,command=click_qr,image=photo3).pack(side=LEFT,pady
=0,padx=20,ipady=40,ipadx=22)
b4=ttk.Button(f1,command=click_imgcomp,image=photo4).pack(side=LE
FT,pady=0,padx=20,ipady=40,ipadx=22)
#b1=Button(f,text="Duplicate \nImage
Deleter",image=img,relief=GROOVE).pack(anchor=CENTER,side=LEFT,pa
dy=20,padx=20,ipady=65,ipadx=25)
f1.pack()

f2=Frame(background="#102e36")
bt1=Button(f2,command=click_dup,text="Duplicate \nImage
Deleter",font="Lucida 15
bold",bg="#102e36",fg="white",relief=FLAT).pack(side=LEFT,pady=0,padx
=20,ipady=40,ipadx=14)
bt2=Button(f2,command=click_text,text="Text\nExtracter",font="Lucida
15
bold",bg="#102e36",fg="white",relief=FLAT).pack(side=LEFT,pady=0,padx
=18,ipady=40,ipadx=35)
```

```
bt4=Button(f2,command=click_qr,text="Image\nCompressor",font="Luci
da 15
bold",bg="#102e36",fg="white",relief=FLAT).pack(side=RIGHT,pady=0,pad
x=20,ipady=40,ipadx=20)
bt3=Button(f2,command=click_imgcomp,text="QR
Code\nScanner",font="Lucida 15
bold",bg="#102e36",fg="white",relief=FLAT).pack(side=RIGHT,pady=0,pad
x=21,ipady=40,ipadx=35)
#b1=Button(f,text="Duplicate \nImage
Deleter",image=img,relief=GROOVE).pack(anchor=CENTER,side=LEFT,pa
dy=20,padx=20,ipady=65,ipadx=25)
f2.pack()
root.mainloop()
```

➢ Duplicate Image Deleter GUI

```
from utilities import *
from tkinter import *
from tkinter import ttk
from tkinter.filedialog  import *
import tkinter.messagebox as tmsg


root=Tk()
path=""
menul=[]
root.config(bg="#252525")
root.title("Duplicate Image Deleter")
root.geometry("850x450")
root.resizable(width=False, height=False)
def folderc():
    location=askdirectory(title="Select the Folder")
    folderv.set(location)
frame1=LabelFrame(root,text="Selection",bg="#232325",fg="#ffffff",heig
ht=165,width=800)
label=Label(root,text="Duplicate Image
Deleter",bg="#252525",fg="#ffffff",font="Helvetica 35 bold italic")
label.pack(pady=20)
folderv=StringVar()
l1=Label(root,text="Location of folder having duplicate images :-
",font="Helvetica 11",bg="#252525",fg="white").place(x=80,y=120)
foldere=Entry(root,textvariable=folderv,width=80).place(x=83,y=145,heig
ht=23)
def take_input():

    path=folderv.get()
    ftype=filetype.get()
```

```
        dl=[]
        if ftype=="jpg":
            dl=delete_duplicate(path,"jpg")
            dl.append(delete_duplicate(path,"JPG"))
            if dl!=[[]]:
                tmsg.showinfo("Deletion is successful",f"Following files
are deleted \n {dl}")
            else:
                tmsg.showinfo("Deletion is not successful",f"No
duplicate files to Delete !!")
        elif ftype=="png":
            dl=delete_duplicate(path,"png")
            dl.append(delete_duplicate(path,"PNG"))
            if dl!=[[]]:
                tmsg.showinfo("Deletion is successful",f"Following files
are deleted \n {dl}")
            else:
                tmsg.showinfo("Deletion is not successful",f"No
duplicate files to Delete !!")
        else:
            dl=delete_duplicate(path,ftype)
            if dl!=[[]]:
                tmsg.showinfo("Deletion is successful",f"Following files
are deleted \n {dl}")
            else:
                tmsg.showinfo("Deletion is not successful",f"No
duplicate files to Delete !!")
l2=Label(root,text="/",fg="white",font="Helvetica
24",bg="#252525").place(x=570,y=136)
b1=Button(root,text="Browse...",command=folderc,relief="ridge").place(
x=590,y=144,height=23)
filetype=StringVar()
filetype.set("type")
Label(root,text="Select the filetype :",font="Helvetica
11",bg="#232325",fg="#ffffff").place(x=80,y=182)
```

```
radio = Radiobutton(root, text="jpeg",activebackground="#252525",
activeforeground="white",
selectcolor="#252525",bg="#232325",fg="#ffffff",variable=filetype,
value="jpeg").place(x=470,y=182)
radio = Radiobutton(root, text="jpg",activebackground="#252525",
activeforeground="white",
selectcolor="#252525",bg="#232325",fg="#ffffff",variable=filetype,
value="jpg").place(x=250,y=182)
radio = Radiobutton(root, text="png",activebackground="#252525",
activeforeground="white",
selectcolor="#252525",bg="#232325",fg="#ffffff", variable=filetype,
value="png").place(x=360,y=182)
submit=ttk.Button(root,text="Delete Duplicate
Images",command=take_input).place(x=360,y=225)
frame1.pack()

root.mainloop()
```

➢ Text Extracter GUI

```
from utilities import *
from tkinter import *
from tkinter import ttk
from tkinter.filedialog  import *
import tkinter.messagebox as tmsg
import numpy
from PIL import Image
from pytesseract import pytesseract

def text_extracter(image_path):
        import cv2
        path_to_tesseract = r'C:\Users\Lakshya
Sharma\AppData\Local\Tesseract-OCR\tesseract.exe'
        img = Image.open(image_path)
        pytesseract.tesseract_cmd = path_to_tesseract
        text = pytesseract.image_to_string(img)
        #Displaying the extracted text
        return text[:-1]
import os

root=Tk()
path=""

root.title("Text Extracter")
root.config(bg="#252525")
root.geometry("850x460")
root.resizable(width=False, height=False)
def imagec():
        location=askopenfilename(title="Select the Image")
        imagev.set(location)
frame1=LabelFrame(root,text="Selection",bg="#232325",fg="#ffffff",heig
ht=165,width=800)
```

```
label=Label(root,text="Text
Extracter",bg="#252525",fg="#ffffff",font="Helvetica 35 bold italic")
label.pack(pady=20)
imagev=StringVar()
l1=Label(root,text="Location of image :-",font="Helvetica
11",bg="#252525",fg="white").place(x=80,y=125)
imagee=Entry(root,textvariable=imagev,width=80).place(x=83,y=155,heig
ht=23)
def take_input():
    txt=text_extracter(imagev.get())
    t= Text(root, state='disabled', width=35, height=8,font="Helvetica
10",bg="#232323",fg="white")
    t.place(x=150,y=300)
    t.configure(state='normal')
    t.insert("end",txt)
    def copy():
        root.clipboard_append(txt)

    b1=ttk.Button(root,text="Copy to
clipboard",command=copy).place(x=550,y=350)

l2=Label(root,text="/",fg="white",font="Helvetica
24",bg="#252525").place(x=570,y=146)
b1=Button(root,text="Browse...",command=imagec,relief="ridge").place(
x=590,y=154,height=23)
submit=ttk.Button(root,text="Extract
Text",command=take_input).place(x=360,y=220)
frame1.pack()
root.mainloop()
```

➢ QR Code Scanner GUI

```
from tkinter import *
from tkinter import ttk
from tkinter.filedialog import *
from utilities import scanqrcode
import webbrowser
import tkinter.messagebox as tmsg


root = Tk()
root.title("QR Code Scanner")
def browse():
    file_path = askopenfilename()
    entry1.set(file_path)
    #print(f"The Path of the Image is:{file_path}")

def execute():
    file_path = entry1.get()
    qr = scanqrcode(file_path)
    if qr!="QR Code not detected":
        t= Text(root, state='disabled', width=35, height=8,font="Helvetica
10",bg="#232323",fg="white")
        t.place(x=150,y=300)
        t.configure(state='normal')
        t.insert("end",qr)
        def copy():
            root.clipboard_append(qr)
        def openinbr():
            webbrowser.open(qr, new=2)
        b1=ttk.Button(root,text="Copy to
clipboard",command=copy).place(x=550,y=340)
        b2=ttk.Button(root,text="Open in
Browser",command=openinbr).place(x=550,y=380)
    else:
```

```
        tmsg.showinfo("Qr Code","QR Code not detected")
root.geometry("850x460")
root.config(bg='#252525')
root.resizable(width=False, height=False)
label1 = Label(root,text="QR Code Scanner",font="Helvitica 36 bold
italic",background="#252525",foreground="#ffffff").pack(pady="20")

frame1=LabelFrame(root,text="Selection",bg="#232325",fg="#ffffff",heig
ht=165,width=800)
label2 = Label(root, text="Enter the path of the Image to scan the QR
Code",font="Helvetica 11",bg="#252525",fg="white").place(x=80,y=125)
entry1 = StringVar()
entry2 =
Entry(root,textvariable=entry1,width=80).place(x=83,y=155,height=23)

l2=Label(root,text="/",fg="white",font="Helvetica
24",bg="#252525").place(x=570,y=146)
b1=Button(root,text="Browse...",command=browse,relief="ridge").place(
x=590,y=154,height=23)
submit=ttk.Button(root,text="SCAN",command=execute).place(x=360,y=
220)
frame1.pack()

root.mainloop()
```

➢ Image Compressor GUI

```
from tkinter import *
from tkinter.filedialog import *
from tkinter import ttk
from utilities import *
import tkinter.messagebox as tmsg

root = Tk()
root.title("Image Compressor")
def browse():
    file_path = askopenfilename()
    entry1.set(file_path)
def execute():
    comp=compress(entry1.get())
    tmsg.showinfo("Image Compressor","Compressed succsessfully")
label1 = Label(root,text="Image Compressor",font="Helvitica 36 bold
italic",background="#252525",foreground="#ffffff",pady="30").pack()

frame1=LabelFrame(root,text="Selection",bg="#232325",fg="#ffffff",heig
ht=190,width=800)
root.geometry("900x500")
root.config(bg='#252525')
root.resizable(width=False, height=False)
label2 = Label(root, text="Enter the path of the Image you want to
compresss",background="#252525",foreground="#ffffff",font="Helvitica
12 ").place(x=80,y=170)

entry1 = StringVar()
entry2 =
Entry(root,textvariable=entry1,width="82",borderwidth="2",).place(x=80,
y=205,height="25")
```

```
label3 = Label(root,text="/",bg="#252525",fg="#ffffff",font="helvitica 21
").place(x=585,y=200)

b1=Button(root,text="Browse...",command=browse,relief="ridge").place(
x=610,y=205,height=23)
submit=ttk.Button(root,text="Compress",command=execute).place(x=40
0,y=260)
frame1.pack()

root.mainloop()
```

➢ Utilities

```
def delete_duplicate(image_folder,filetype):
        # import required libraries
        import os
        import cv2
        import numpy as np
        duplicate_files= []
        dall=[]
        not_duplicate_files= []
        image_files = [_ for _ in os.listdir(image_folder) if
_.endswith(filetype)]
        for file_org in image_files:
            if file_org not in duplicate_files:
                fo=image_folder+"//"+file_org
                img1=cv2.imread(fo)
                img1=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
                for file_check in image_files:
                    if file_check != file_org:
                        fc=image_folder+"//"+file_check
                        img2=cv2.imread(fc)
                        img2=cv2.cvtColor(img2,
cv2.COLOR_BGR2GRAY)
                        #print(img1,"\n",img2)
                        if np.array_equal(img1, img2):
                            duplicate_files.append(file_check)
                            dall.append(file_check)
                            dall.append(file_org)
        #print(duplicate_files)
        if (duplicate_files!=[]):
            for i in duplicate_files:
                f=image_folder+"\\"+i
                os.remove(f)
        else:
            pass
        return list(set(dall))
```

```
def text_extracter(image_path):
    from PIL import Image
    from pytesseract import pytesseract
    # import cv2
    path_to_tesseract = r'C:\Users\Lakshya
Sharma\AppData\Local\Tesseract-OCR\tesseract.exe'
    img = Image.open(image_path)
    pytesseract.tesseract_cmd = path_to_tesseract
    text = pytesseract.image_to_string(img)

    #Displaying the extracted text
    print(text[:-1])


def scanqrcode(img):
    import cv2
    import numpy as np
    import sys
    import time
    inputImage = cv2.imread(img)
    # Display barcode and QR code location
    def display(im, bbox):
        n = len(bbox)
        for j in range(n):
            cv2.line(im, tuple(bbox[j][0]), tuple(bbox[ (j+1) % n][0]), (255,0,0),
3)

            # Display results
            cv2.imshow("Results", im)

    qrDecoder = cv2.QRCodeDetector()

    # Detect and decode the qrcode
    data,bbox,rectifiedImage = qrDecoder.detectAndDecode(inputImage)
```

```
    if len(data)>0:
        return format(data)
        #display(inputImage, bbox)
        #rectifiedImage = np.uint8(rectifiedImage);
        #cv2.imshow("Rectified QRCode", rectifiedImage);
    else:
        return "QR Code not detected"
        #cv2.imshow("Results", inputImage


def compress(file_path):
        import PIL
        from PIL import Image
        from tkinter.filedialog import askopenfilename,asksaveasfilename
        img = PIL.Image.open(file_path)
        myHeight, mywidth = img.size
        #img = img.resize((160,300), PIL.Image.ANTIALIAS)
        img = img.resize((myHeight,mywidth), PIL.Image.ANTIALIAS)
        save_path = asksaveasfilename()
        try:
                img.save(save_path+"_compressed.JPG")
        except:
                img.save(save_path+"_compressed.PNG")
```

FUTURE SCOPE

- Big data Handling
- Artificial Image orientation correcter
- Extracting text from various languages.
- Classification of images using face recognition(A.I)
- On-Screen Text extraction.

Limitation

- Only jpg,png and jpeg files are accepted.
- Text extractor can only extract text from clear images.
- Text extractor cannot extract from other languages.
- Image can’t be compressed as per given size.

CONCLUSION

In conclusion, the Smart Image Utility System is a project that addresses real-life problems in image management. The four utilities - Duplicate Image Deleter, Text Extractor, QR Code Scanner, and Image Compressor - provide a user-friendly solution for various image management issues. The system's graphical user interface (GUI) makes image management simple and efficient for users. The project has demonstrated that it is possible to develop a comprehensive solution for image management using GUI-based tools. The results of this project have implications for further research in the area of image management, and provide a valuable contribution to the field.