MINI-PROJECT ASSIGNMENT

TITLE – IMAGE BASED SEGANOGRAPHY

```
from tkinter import *
from tkinter import ttk
import tkinter.filedialog
from PIL import ImageTk
from PIL import Image
from tkinter import messagebox
from io import BytesIO
import  os

class Stegno:
    output_image_size = 0

    def main(self,root):
        root.title('ImageSteganography')
        root.geometry('500x600')
        root.resizable(width =False, height=False)
        root.configure(bg = "black")
        f = Frame(root,bg = "black",highlightbackground="blue")

        #im1 = Image.open(r"C:\Users\lonew\Desktop\background-black-yellow-blue.jpg")
        #im2 = im1.resize((500,150))
        #im3 = ImageTk.PhotoImage(im2)
        #im1a = Label(f,image=im3,borderwidth=1,relief="solid")
        #im1a.photo = im3


        #im5 = Image.open(r"C:\Users\lonew\Desktop\Black-And-Red-Background-HD.jpg")
        #im6 = im5.resize((500,150))
        #im7 = ImageTk.PhotoImage(im6)
        #im5a = Label(f,image=im7,borderwidth=1,relief="solid")
        #im5a.photo = im7
        #im5a.grid(row=5)

        title = Label(f,text='Mini-Project\n\nImage Steganography',)
        title.config(font=('courier',25,'bold','italic'),relief='solid',fg='Antiquewhi
te',bg='black')
        title.grid(pady=10)

        b_encode = Button(f,text="Encode",bg="Black",fg =
"Tomato",activebackground="red",relief='solid',command= lambda :self.frame1_encode(f),
padx=5)
        b_encode.config(font=('Monotype Corsiva',24,'bold'))
        b_decode = Button(f, text="Decode",bg="Black",fg =
"Cyan",activebackground="blue",relief='solid',command=lambda :self.frame1_decode(f))
```

```
        b_decode.config(font=('Monotype Corsiva',24,'bold'))
        b_decode.grid(pady = 12)

        root.grid_rowconfigure(1, weight=1)
        root.grid_columnconfigure(0, weight=1)

        f.grid()
        title.grid(row=2)
        b_encode.grid(row=3)
        b_decode.grid(row=4)
        #im1a.grid(row=1)

    def home(self,frame):
            frame.destroy()
            self.main(root)

    def frame1_decode(self,f):
        f.destroy()
        d_f2 = Frame(root,bg='black')

        #im1 = Image.open(r"C:\Users\lonew\Desktop\Black-And-Red-Background-HD.jpg")
        #im2 = im1.resize((500,300))
        #im3 = ImageTk.PhotoImage(im2)
        #im1a = Label(d_f2,image=im3,borderwidth=1,relief="solid")
        #im1a.photo = im3
        #im1a.grid(row=1)

        l1 = Label(d_f2, text='Select Image with Hidden
text:\n',bg='black',fg='Antiquewhite')
        l1.config(font=('courier',18,'italic'))
        l1.grid(row=2)
        bws_button = Button(d_f2, text='Select',bg="Black",fg =
"Cyan",activebackground="blue",relief='solid', command=lambda
:self.frame2_decode(d_f2))
        bws_button.config(font=('courier',18))
        bws_button.grid(row=3)
        back_button = Button(d_f2, text='Cancel',bg="Black",fg =
"Tomato",activebackground="red",relief='solid', command=lambda :
Stegno.home(self,d_f2))
        back_button.config(font=('courier',18))
        back_button.grid(pady=15)
        back_button.grid(row=4)
        d_f2.grid()

    def frame2_decode(self,d_f2):
        d_f3 = Frame(root,bg='black')
```

```
        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png',
'*.png'),('jpeg', '*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))
        if not myfile:
            messagebox.showerror("Error","You have selected nothing !")
        else:
            myimg = Image.open(myfile, 'r')
            myimage = myimg.resize((300, 200))
            img = ImageTk.PhotoImage(myimage)
            l4= Label(d_f3,text='Selected Image :',bg='black',fg='Antiquewhite')
            l4.config(font=('courier',18,'bold'))
            l4.grid()
            panel = Label(d_f3, image=img,bg='black')
            panel.image = img
            panel.grid()
            hidden_data = self.decode(myimg)
            l2 = Label(d_f3, text='Hidden data is :',bg='black',fg='Antiquewhite')
            l2.config(font=('courier',18,'bold'))
            l2.grid(pady=10)
            text_area = Text(d_f3, width=50, height=10)
            text_area.insert(INSERT, hidden_data)
            text_area.configure(state='disabled')
            text_area.grid()
            back_button = Button(d_f3, text='Cancel',bg="Black",fg =
"Cyan",activebackground="blue",relief='solid', command= lambda :self.page3(d_f3))
            back_button.config(font=('courier',11))
            back_button.grid(pady=15)
            back_button.grid()
            show_info = Button(d_f3,text='More Info',bg="Black",fg =
"Tomato",activebackground="red",relief='solid',command=self.info)
            show_info.config(font=('courier',11))
            show_info.grid()
            d_f3.grid(row=1)
            d_f2.destroy()

    def decode(self, image):
        data = ''
        imgdata = iter(image.getdata())

        while (True):
            pixels = [value for value in imgdata.__next__()[:3] +
                      imgdata.__next__()[:3] +
                      imgdata.__next__()[:3]]
            binstr = ''
            for i in pixels[:8]:
                if i % 2 == 0:
                    binstr += '0'
                else:
```

```
                    binstr += '1'

            data += chr(int(binstr, 2))
            if pixels[-1] % 2 != 0:
                return data

    def frame1_encode(self,f):
        f.destroy()
        f2 = Frame(root,bg='black')

        #im5 = Image.open(r"C:\Users\lonew\Desktop\background-black-yellow-blue.jpg")
        #im6 = im5.resize((500,300))
        #im7 = ImageTk.PhotoImage(im6)
        #im5a = Label(f2,image=im7,borderwidth=1,relief="solid")
        #im5a.photo = im7
        #im5a.grid(row=1)

        l1= Label(f2,text='Select the Image in which \nyou want to hide text
:\n',bg='black' , fg ='Antiquewhite')
        l1.config(font=('courier',18,'italic'))
        l1.grid(row=2)

        bws_button = Button(f2,text='Select',bg="Black",fg =
"Cyan",activebackground="blue",relief='solid',command=lambda : self.frame2_encode(f2))
        bws_button.config(font=('courier',18))
        bws_button.grid(row=3)
        back_button = Button(f2, text='Cancel',bg="Black",fg =
"Tomato",activebackground="red",relief='solid', command=lambda : Stegno.home(self,f2))
        back_button.config(font=('courier',18))
        back_button.grid(pady=15)
        back_button.grid(row=4)
        f2.grid()


    def frame2_encode(self,f2):
        ep= Frame(root,bg ='black')
        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png',
'*.png'),('jpeg', '*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))
        if not myfile:
            messagebox.showerror("Error","You have selected nothing !")
        else:
            myimg = Image.open(myfile)
            myimage = myimg.resize((300,200))
            img = ImageTk.PhotoImage(myimage)
            l3= Label(ep,text='Selected Image',fg='Antiquewhite',bg='black')
            l3.config(font=('courier',18))
            l3.grid()
```

```
            panel = Label(ep, image=img)
            panel.image = img
            self.output_image_size = os.stat(myfile)
            self.o_image_w, self.o_image_h = myimg.size
            panel.grid()
            l2 = Label(ep, text='Enter the message',fg='AntiqueWhite',bg='black')
            l2.config(font=('courier',18))
            l2.grid(pady=15)
            text_area = Text(ep, width=50, height=10)
            text_area.grid()
            encode_button = Button(ep, text='Cancel',bg="Black",fg =
"Tomato",activebackground="red",relief='solid', command=lambda : Stegno.home(self,ep))
            encode_button.config(font=('courier',11))
            data = text_area.get("1.0", "end-1c")
            back_button = Button(ep, text='Encode',bg="Black",fg =
"Cyan",activebackground="Blue",relief='solid', command=lambda :
[self.enc_fun(text_area,myimg),Stegno.home(self,ep)])
            back_button.config(font=('courier',11))
            back_button.grid(pady=15)
            encode_button.grid()
            ep.grid(row=1)
            f2.destroy()


    def info(self):
        try:
            str = 'original image:-\nsize of original image:{}mb\nwidth: {}\nheight:
{}\n\n' \
                  'decoded image:-\nsize of decoded image: {}mb\nwidth: {}' \
                '\nheight: {}'.format(self.output_image_size.st_size/1000000,
                                      self.o_image_w,self.o_image_h,
                                      self.d_image_size/10,
                                      self.d_image_w,self.d_image_h)
            messagebox.showinfo('info',str)
        except:
            messagebox.showinfo('Info','Unable to get the information')
    def genData(self,data):
        newd = []

        for i in data:
            newd.append(format(ord(i), '08b'))
        return newd

    def modPix(self,pix, data):
        datalist = self.genData(data)
        lendata = len(datalist)
        imdata = iter(pix)
```

```
        for i in range(lendata):
            # Extracting 3 pixels at a time
            pix = [value for value in imdata.__next__()[:3] +
                   imdata.__next__()[:3] +
                   imdata.__next__()[:3]]
            # Pixel value should be made
            # odd for 1 and even for 0
            for j in range(0, 8):
                if (datalist[i][j] == '0') and (pix[j] % 2 != 0):

                    if (pix[j] % 2 != 0):
                        pix[j] -= 1

                elif (datalist[i][j] == '1') and (pix[j] % 2 == 0):
                    pix[j] -= 1
            # Eigh^th pixel of every set tells
            # whether to stop or read further.
            # 0 means keep reading; 1 means the
            # message is over.
            if (i == lendata - 1):
                if (pix[-1] % 2 == 0):
                    pix[-1] -= 1
            else:
                if (pix[-1] % 2 != 0):
                    pix[-1] -= 1

            pix = tuple(pix)
            yield pix[0:3]
            yield pix[3:6]
            yield pix[6:9]

    def encode_enc(self,newimg, data):
        w = newimg.size[0]
        (x, y) = (0, 0)

        for pixel in self.modPix(newimg.getdata(), data):

            # Putting modified pixels in the new image
            newimg.putpixel((x, y), pixel)
            if (x == w - 1):
                x = 0
                y += 1
            else:
                x += 1

    def enc_fun(self,text_area,myimg):
        data = text_area.get("1.0", "end-1c")
```

```
        if (len(data) == 0):
            messagebox.showinfo("Alert","Kindly enter text in TextBox")
        else:
            newimg = myimg.copy()
            self.encode_enc(newimg, data)
            my_file = BytesIO()
            temp=os.path.splitext(os.path.basename(myimg.filename))[0]
            newimg.save(tkinter.filedialog.asksaveasfilename(initialfile=temp,filetype
s = ([('png', '*.png')]),defaultextension=".png"))
            self.d_image_size = my_file.tell()
            self.d_image_w,self.d_image_h = newimg.size
            messagebox.showinfo("Success","Encoding Successful\nFile is saved as
Image_with_hiddentext.png in the same directory")

    def page3(self,frame):
        frame.destroy()
        self.main(root)

root = Tk()

o = Stegno()
o.main(root)

root.mainloop()
```