



Student Name: Arghyanil Chowdhury

Branch: MCA (CCD)

Semester: 1st

Subject Name: Python Programming

UID: 24MCC20021

Section/Group: 24MCD1-A

Date of Performance: 21/10/24

Subject Code: 24CAH-606

Q. Title of Project :- Steganography Using Python: Hiding Secret Messages in Images.

1. Aim/Overview of the practical:

The aim of this practical is to develop a Python-based application that implements steganography to conceal secret text messages within digital images. This project uses the Least Significant Bit (LSB) technique, which allows users to hide a message within the pixels of an image without significantly altering its appearance. Through a user-friendly graphical interface built with Tkinter, the application enables users to load an image, input a secret message, hide the message in the image, and save the modified image. Additionally, the application allows users to retrieve and display hidden messages from images. The project demonstrates the practical application of steganography for secure communication and data protection.

2. Task to be done:

- Develop a graphical user interface (GUI) using Tkinter to facilitate image selection and message input/output.
- Implement functionality to load and display an image file for steganographic operations.
- Utilize the Least Significant Bit (LSB) technique from the stegano.lsb library to hide a secret text message within an image.
- Enable the retrieval and display of hidden messages from an image using the LSB method.
- Implement image resizing to ensure proper display within the application window without distortion.
- Provide functionality to save the modified image with the hidden message as a PNG file.
- Ensure that the application supports error-free operations, such as file handling and message processing.
- Perform tests to verify that the hidden message can be correctly retrieved from the modified image.

3. Code for experiment/practical:

```
from tkinter import *
from tkinter import filedialog
import tkinter as tk
from PIL import Image, ImageTk
import os
from stegano import lsb

root=Tk()
root.title("Steganography - Hide a Secret Text Message in an Image")
root.geometry("700x500+150+180")
root.resizable(False,False)
root.configure(bg="#2f4155")

def showimage():
    global filename
    filename = filedialog.askopenfilename(initialdir=os.getcwd(),
                                          title='Select Image File',
                                          filetype= (("PNG file", "*.png"),
                                                    ("JPG File", "*.jpg"), ("All file", "*.txt")))

    if filename:
        img = Image.open(filename)

        img_width, img_height = img.size

        max_width, max_height = 250, 250

        if img_width > max_width or img_height > max_height:
            ratio = min(max_width / img_width, max_height / img_height)
            new_width = int(img_width * ratio)
            new_height = int(img_height * ratio)
            img = img.resize((new_width, new_height), Image.Resampling.LANCZOS)

        img = ImageTk.PhotoImage(img)

        lbl.configure(image=img, width=250, height=250)
        lbl.image = img

def Hide():
    global secret
    message=text1.get(1.0,END)
    secret=lsb.hide(str(filename),message)

def Show():
    clear_message=lsb.reveal(filename)
    text1.delete(1.0,END)
    text1.insert(END,clear_message)

def save():
    file_path = filedialog.asksaveasfilename(defaultextension=".png",
                                              filetypes= (("PNG files", "*.png")],
                                              title="Save Image As")

    if file_path:
        secret.save(file_path)

image_icon=PhotoImage(file="C:/Users/Arghyanil/Desktop/Python Projects/logo.jpg.jpg")
root.iconphoto(False,image_icon)
```

```
logo=PhotoImage(file="C:/Users/Arghyanil/Desktop/Python Projects/logo.png")
Label(root,image=logo,bg="#2f4155").place(x=10,y=0)

Label(root, text="STEGANOGRAPHY USING PYTHON",bg="#2d4155",fg="white",font="arial 25 bold").place(

f=Frame(root,bd=3,bg="black",width=340,height=280,relief=GROOVE)
f.place(x=10,y=80)

lbl=Label(f,bg="black")
lbl.place(x=40,y=10)

frame2=Frame(root,bd=3,width=340,height=280,bg="white",relief=GROOVE)
frame2.place(x=350,y=80)

text1=Text(frame2,font="Robote 20",bg="white",fg="black",relief=GROOVE,wrap=WORD)
text1.place(x=0,y=0,width=320,height=295)

scrollbar1=Scrollbar(frame2)
scrollbar1.place(x=320,y=0,height=300)

scrollbar1.configure(command=text1.yview)
text1.configure(yscrollcommand=scrollbar1.set)

frame3=Frame(root,bd=3,bg="#2f4155",width=330,height=100,relief=GROOVE)
frame3.place(x=10,y=370)

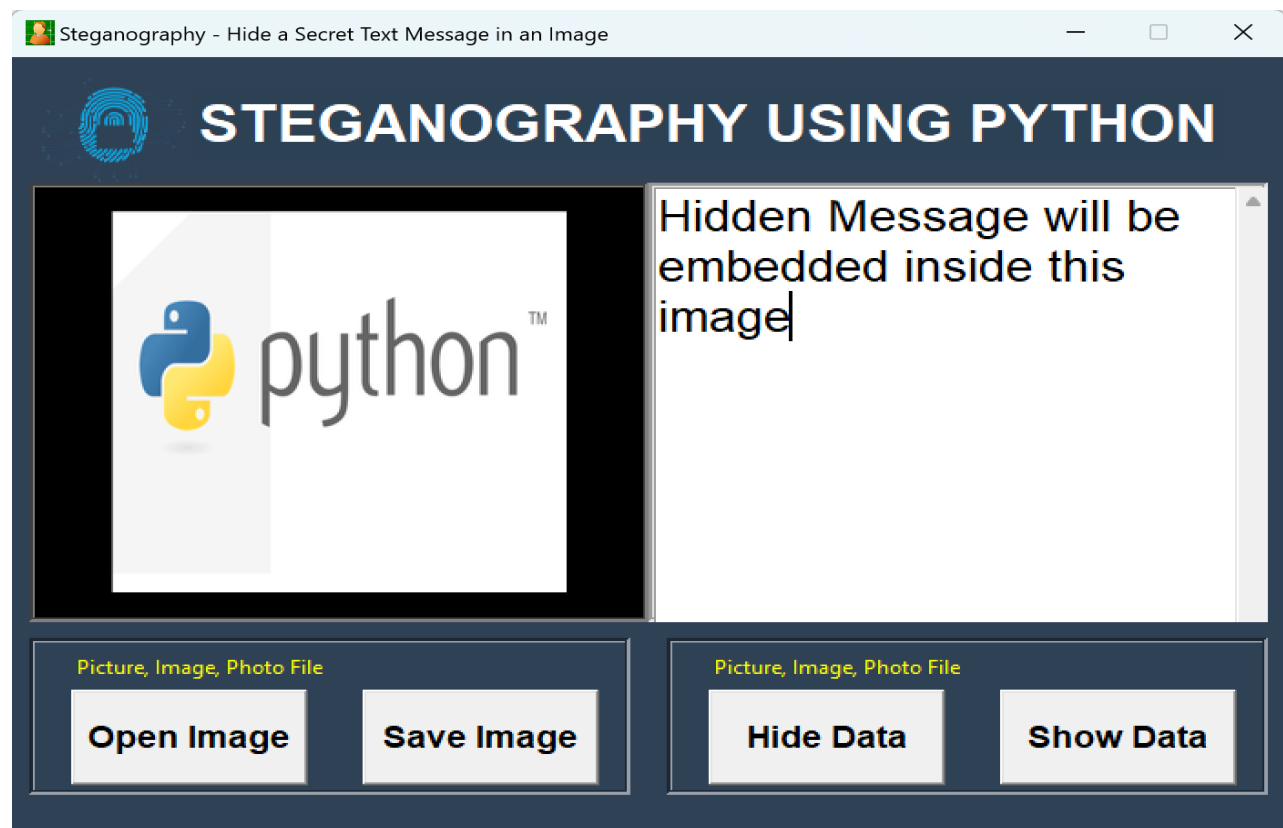
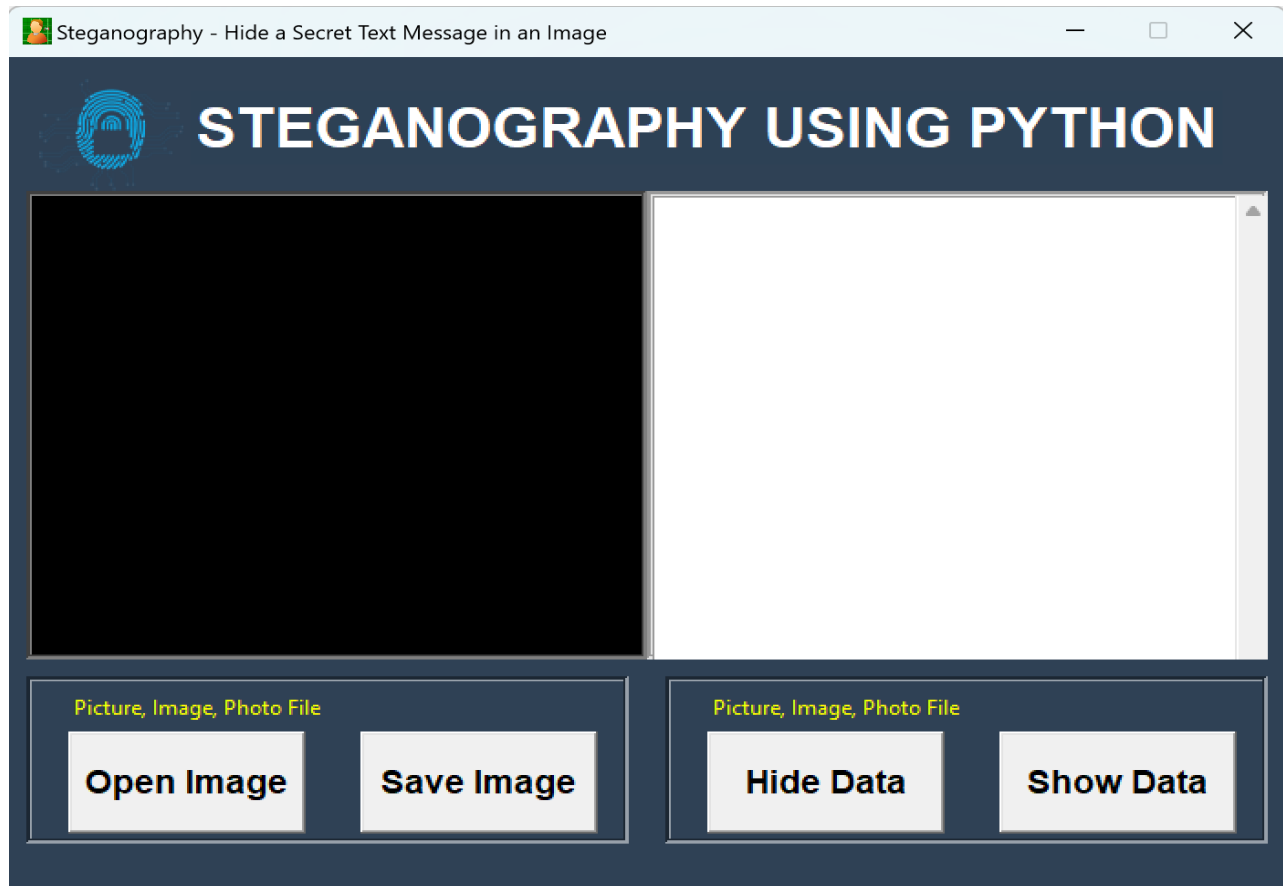
Button(frame3,text="Open Image",width=10,height=2,font="arial 14 bold",command=showimage).place(x=
Button(frame3,text="Save Image",width=10,height=2,font="arial 14 bold",command=save).place(x=180,y
Label(frame3,text="Picture, Image, Photo File",bg="#2f4155",fg="yellow").place(x=20,y=5)

frame4=Frame(root,bd=3,bg="#2f4155",width=330,height=100,relief=GROOVE)
frame4.place(x=360,y=370)

Button(frame4,text="Hide Data",width=10,height=2,font="arial 14 bold",command=Hide).place(x=20,y=3
Button(frame4,text="Show Data",width=10,height=2,font="arial 14 bold",command=Show).place(x=180,y=
Label(frame4,text="Picture, Image, Photo File",bg="#2f4155",fg="yellow").place(x=20,y=5)

root.mainloop()
```

4. Result/Output/Writing Summary:



Learning outcomes (What I have learnt):

- Gained hands-on experience with the Tkinter library to build a user-friendly graphical user interface for desktop applications.
- Learned how to implement steganography using the Least Significant Bit (LSB) technique to securely hide text messages within image files.
- Improved understanding of file handling in Python, particularly working with image formats such as PNG and JPG.
- Developed skills in using the stegano.lsb library for embedding and extracting secret messages from digital images.
- Gained practical knowledge of image manipulation, including resizing and maintaining aspect ratio using the PIL (Python Imaging Library).
- Enhanced problem-solving abilities by addressing challenges related to hiding, revealing, and saving modified images.
- Strengthened understanding of secure communication techniques and their practical applications through steganography.
- Improved ability to integrate multiple Python libraries to create a cohesive application that meets specified requirements.