

Project Update Report: Steganography Tool

GUI-Based Steganography Tool for Hiding Text and Files Inside Images

Objective:

To design and implement a GUI application in Python that enables users to hide and extract:

- **Text messages**
- **Any file type** (images, videos, documents, etc.) within PNG/BMP images using **Least Significant Bit (LSB)** steganography.

Technologies Used:

- **Python**
- **Tkinter** (GUI)
- **Pillow** (Image handling)
- **Custom LSB logic** for binary encoding/decoding

Python code:

```
import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image

import stepic

# Function to encode message into image

def encode_image():

    img_path = filedialog.askopenfilename(filetypes=[("Image files", "*.png;*.bmp")])

    if not img_path:

        return

    try:

        img = Image.open(img_path)

        message = text_entry.get("1.0", tk.END).strip()
```

```

if not message:
    messagebox.showerror("Error", "Enter a message to hide.")
    return

encoded_img = stepic.encode(img, message.encode())

save_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG
files", "*.png")])

if save_path:
    encoded_img.save(save_path)
    messagebox.showinfo("Success", f"Image saved at:\n{save_path}")
except Exception as e:
    messagebox.showerror("Error", str(e))

# Function to decode message from image
def decode_image():
    img_path = filedialog.askopenfilename(filetypes=[("Image files", "*.png;*.bmp")])
    if not img_path:
        return

    try:
        img = Image.open(img_path)
        hidden_message = stepic.decode(img) # FIXED: Removed .decode() as it returns str
        text_entry.delete("1.0", tk.END)
        text_entry.insert(tk.END, hidden_message)
        messagebox.showinfo("Success", "Message extracted successfully!")
    except Exception as e:
        messagebox.showerror("Error", str(e))

# GUI Setup
window = tk.Tk()

```

```
window.title("Steganography Tool")

window.geometry("500x400")

window.resizable(False, False)


frame = tk.Frame(window)

frame.pack(pady=20)


tk.Label(frame, text="Enter Message to Hide:", font=("Arial", 12)).pack()

text_entry = tk.Text(frame, height=5, width=50)

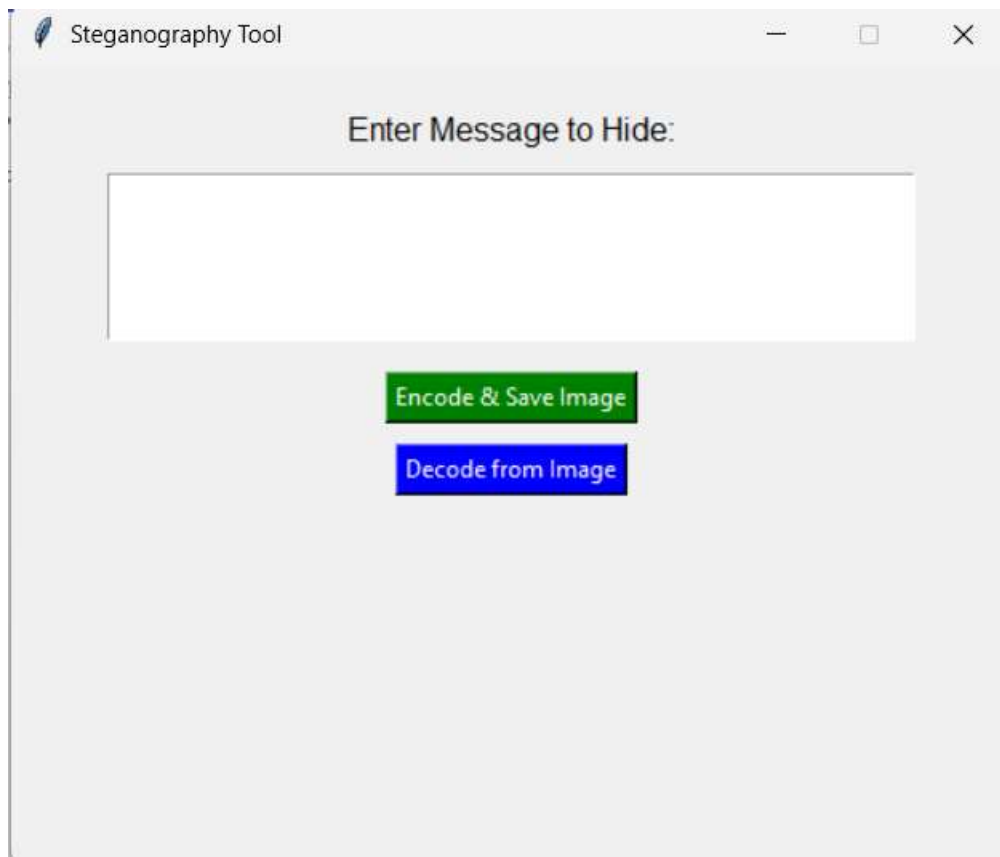
text_entry.pack(pady=10)


tk.Button(frame, text="Encode & Save Image", command=encode_image, bg="green",
fg="white").pack(pady=5)

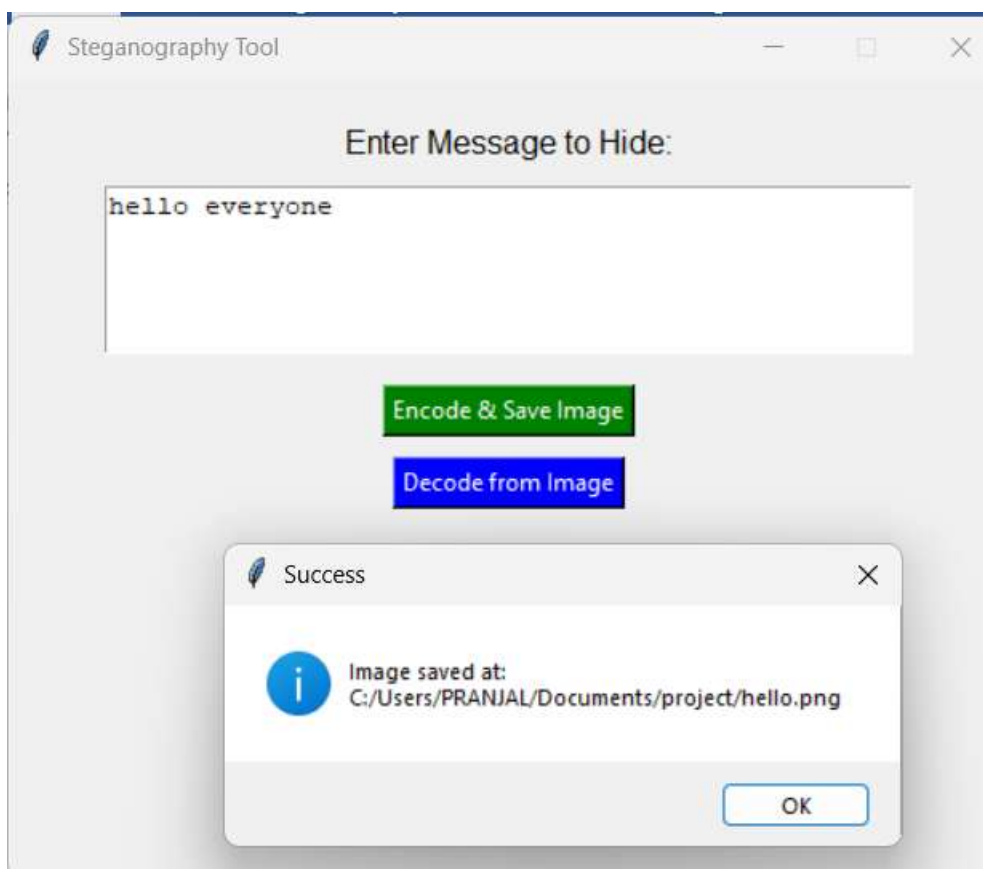
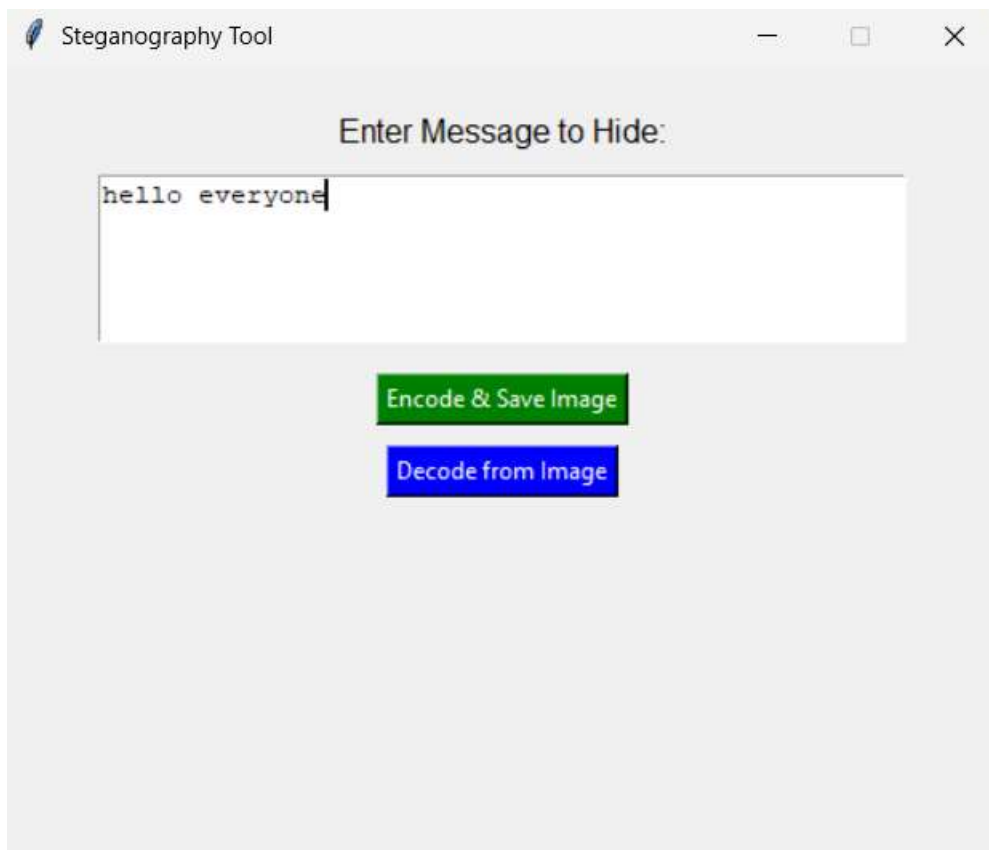
tk.Button(frame, text="Decode from Image", command=decode_image, bg="blue",
fg="white").pack(pady=5)


window.mainloop()
```

output:



FOR ENCODING DATA:



FOR DECODING:

