This app is built with **Python and Tkinter**, and it allows users to **hide** a message inside an image and **extract** it later with a password.

## **🔁 Importing Libraries**

import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image, ImageTk

import os

import secrets

import string

import smtplib, ssl

from email.message import EmailMessage

import mimetypes

* tkinter: Used to create the GUI.
* filedialog: Lets user browse and pick files.
* messagebox: Shows pop-up alerts or errors.
* PIL.Image: Used to handle images (open, edit, save).
* PIL.ImageTk: Used to show images in Tkinter.
* os: Works with file paths and folders.
* secrets and string: Used to make secure random passwords.
* smtplib, ssl: Used to send email securely.
* EmailMessage: Helps format and send emails.
* mimetypes: Detects file type (like .jpg or .png).

## **📁 Setup**

PASSWORD\_FILE = "passwords.txt"

UPLOAD\_DIR = "output\_images"

os.makedirs(UPLOAD\_DIR, exist\_ok=True)

* PASSWORD\_FILE: File to save image-name and its password.
* UPLOAD\_DIR: Folder where encoded (output) images will be saved.
* os.makedirs(): Creates that folder if it doesn't exist.

## **🔐 Generate a Password**

def generate\_password(length=8):

return ''.join(secrets.choice(string.ascii\_letters + string.digits) for \_ in range(length))

* This creates a **random 8-character password** (letters + numbers).

## **📤 Send Email with Password & Image**

def send\_password\_via\_email(sender\_email, smtp\_password, receiver\_email, password, attachment\_path):

* Sends the password and image to someone’s email.

### **Email Setup:**

message = EmailMessage()

message["Subject"] = "Your Steganography Password and Image"

...

message.set\_content(f"...{password}")

* Sets email subject and body text with the password.

### **Attach Image:**

mime\_type, \_ = mimetypes.guess\_type(attachment\_path)

...

with open(attachment\_path, 'rb') as ap:

message.add\_attachment(...)

* Detects file type and attaches the image.

### **Send:**

context = ssl.create\_default\_context()

with smtplib.SMTP\_SSL(...) as server:

server.login(...)

server.send\_message(message)

* Logs in securely to Gmail and sends the email.

## **🧬 Hiding the Message in the Image**

def hide\_text\_in\_image(image\_path, message):

* Takes the image and hides your message inside it.

### **Step-by-step:**

1. Open image.
2. Make a copy to edit.
3. Convert message to binary (0s and 1s).
4. Replace last bit of RGB color values with message bits.
5. Save new image.
6. Generate a password and store it.

python

CopyEdit

binary\_msg = ''.join(format(ord(c), '08b') for c in message)

* Converts each character to 8-bit binary.

python

CopyEdit

r = (r & ~1) | int(binary\_msg[idx])

* Replaces last bit of red with a message bit (LSB method).

python

CopyEdit

out\_path = ...

encoded.save(out\_path)

* Saves the modified image to disk.

python

CopyEdit

with open(PASSWORD\_FILE, 'a') as f:

f.write(f"{filename}:{password}")

* Stores filename and its password.

## **🧪 Extracting Message from Image**

def extract\_text\_from\_image(image\_path, password\_input):

* Reads image and retrieves hidden message if password is correct.

### **Steps:**

1. Check if password matches.
2. If correct, extract binary from each pixel's last bit.
3. Convert binary to characters.
4. Stop when it reaches chr(0) (end marker).

## **🎨 Main Window with Tkinter**

def create\_main\_window():

* Displays the welcome page of the app.

### **It does:**

* Sets window title and size.
* Shows heading and logo (if logo.png exists).
* Two buttons: **Hide Text** and **Extract Text**.

## **🖊 Hide Text Window**

def open\_hide\_window():

* Lets user choose image, type message, enter emails and click "Hide".

### **UI Elements:**

* Entry for file path
* Textbox for message
* Email fields
* Password field
* "Hide Text" button calls handle\_hide() which:  
  + Hides the message
  + Emails the password and image

## **🔍 Extract Text Window**

def open\_extract\_window():

* Lets user:  
  + Pick stego image
  + Enter password
  + Click "Extract Text" to view the message

## **▶️ Run the App**

if \_\_name\_\_ == "\_\_main\_\_":

create\_main\_window()

* Runs the main Tkinter window when the file is executed.