# Image Steganography

(Steganography hide secret text message inside image using python and tkinter as GUI)

Internship Report Submitted in partial fulfilment of the requirement

for undergraduate degree of

**Bachelor of Technology**

In

**COMPUTER SCIENCE AND ENGINEERING**

By

Shriya R Setru

**HU21CSEN0101351**

Under the Guidance

of

**Mrs.G.Lalitha,**

Assistant Professor


Department Of Computer Science and Engineering

GITAM School of Technology

GITAM (Deemed to be University) Hyderabad-502329

December 2023

# DECLARATION

I submit this industrial training work entitled **"Steganography hide secret text message inside image using python and tkinter as GUI**" to GITAM (Deemed to Be University), Hyderabad in partial fulfilment of the requirements for the award of the degree of "**Bachelor of Technology**" in "**Computer Science and Engineering**". I declare that it was carried out independently by me under the guidance of **Mrs.G.Lalitha ,** Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD Name: Shriya R Setru

Date: 22-12-2023 Student ID: HU21CSEN0101351

# ACKNOWLEDGEMENT

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **D. Sambasiva Rao**, Pro Vice Chancellor, GITAM Hyderabad and **Dr. N. Seetharamaiah**, Principal, GITAM Hyderabad.

I thank respected **Mr. Mohaboob Basha Shaik ,** Head of the Computer Science and Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present an internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties **Mrs.G.Lalitha** who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Shriya R Setru

HU21CSEN0101351

# ABSTRACT

Nowadays, computer-based communications are at the threshold of making life easier for everyone in the world; from sharing information, to communicating with each other, to exchanging electronic documents, and to checking bank balances and paying bills. Nonetheless, information security is an essential factor, which must be taken into consideration to ensure secure communications. There are significant interests in security approaches that aim to protect information and digital data, since the growing increase in uses of the internet and multimedia, have raised the interests in image steganography in order to secure and protect them. Steganography serves as an excellent method to secure information as the unauthorized party barely recognizes the secret message or a text that is hidden behind the original or the cover image.This report explores the concept of steganography, a technique for hiding confidential information within digital media. It delves into the implementation of a Python-based steganography application utilizing the Tkinter library for its graphical user interface (GUI). The application effectively embeds text messages within images while maintaining visual fidelity. This report discusses the application's functionality, key techniques, potential applications, and limitations.

Keywords: Steganography, Python, Tkinter, GUI, Image Processing, Cryptography, Information Security

# Table of Contents

## 1. INTRODUCTION:

Every organization or an individual has to transfer the data in a network or as end to end. The main aim of the communication is that the information that anyone is transferring or sending must be accurate, efficient and the most important factor, it must be “secured”. So to secure our data from any unauthorized accesses or grants, information security has various techniques of data hiding. Data hiding involves hiding or securing the secret data (text, image, and video, audio) inside the digital data (text, image, and video, audio).


Suppose there is one company that has to give its Bank Account details to the respective Bank. There are very high chances that hackers or other unauthorized users may track the details and misuse the information provided by the company. So in such cases to securing the communication, data hiding is needed. Steganography is one example of the Data hiding technique used in Information Security.

The word steganography is made of Greek words “Steganos” which means cover and “Graph” that means to construct. The word Steganography was first used in the Golden age in Greece. It is believed that during the golden age, people of Greece used this technique to input secret messages in woods.

This project is on Image steganography in which a cover image or the original image is altered with the secret textual message that is to be sent to the receiver side. And then after successfully sending the cover

image with text, the receiver decodes the secret text message from the cover image.

Image Steganography can be implemented by various algorithms such as LSB, Bit plane Algorithm, Spiral Embedding etc. In our Project Image steganography is done by the most efficient and simple algorithm known as the Least Significant Bit substitution (LSB) algorithm.

In LSB algorithm the last bit of the image pixel is substituted by the binary bits of the Secret messages. Original image altered with the secret message makes the Stego image, which is transferred to the receiver and the receiver will just extract out the least significant bits of the stego image and combine it to get the entire secret text. Secret message transferring is also done by Cryptography techniques in which the original message is converted into the cipher text or unreadable text.

This technique of cryptography still has chances of being cracked or hacked as the unauthorized party have an idea that something is being transferred and they can apply all sort of permutations to encode the cipher text but in Steganography the hackers barely have any idea about the hidden data as the changes are not visible in the images in which secret message is being hidden. So the ideal way of increasing the security of the communication is to use both cryptography and steganography together to make a strong communication process.

So in this project we have designed our own cryptography algorithm named “Star-Dollar Algorithm”. We will discuss this algorithm later on. Steganography is in demand these days as it is a very accurate, efficient method and serves as the most vital method for the secure sharing of the information.

## 1.2 Problem Statement:

The main problem of sharing data between the users is interference of the unauthorized users or parties. Main aim of the communication over a network is that the data that is shared between the users must be secured and safely transferred. So through our project we design a Steganography Algorithm that allows safe and secured data sharing.

## 1.3 Objective:

The main purpose of the entire project on Data Hiding using Image Steganography is building a secured and completely safe communication and data sharing. Main aim on which we focused during the entire project is to hide the secret textual data behind the images in an efficient manner so that the changes in the original images should not be visible to the naked eyes. This is a vital and an interesting field of information security that has various useful and important applications like modern printers for encoding serial numbers, Intelligence services (FBI, RFIS) for secret communication, copyrights protection, watermarking etc.

## 1.4 Methodology:

In our project on "Data hiding using Image steganography", we used a very popular substitution algorithm named "Least Significant Bit Substitution" (LSB).

In this algorithm the least significant bit which in our project was the last binary digit of the pixel values is replaced with the Binary 0s and 1s of the secret message. The combined pixel binary bits and binary zeros and ones of the secret texts are encoded in the original image which is termed as the "stego" image. This digital image is transmitted to the desired receiver and the receiver decodes the stego image and extracts the last bits from it which are the least significant bits and contains the secret information or data that is sent by the sender.

## 1.5 Technical Tools and Technology:

**Software Requirements:**

- Python 3.9
- Python Libraries:
  1. Pillow (PIL)
  2. Cryptography for cryptographic operations.
  3. Fernet

4. Tkinter(GUI)
5. Stegano for steganography operations.

➢ Operating system: Windows 11

**Hardware Requirements:**

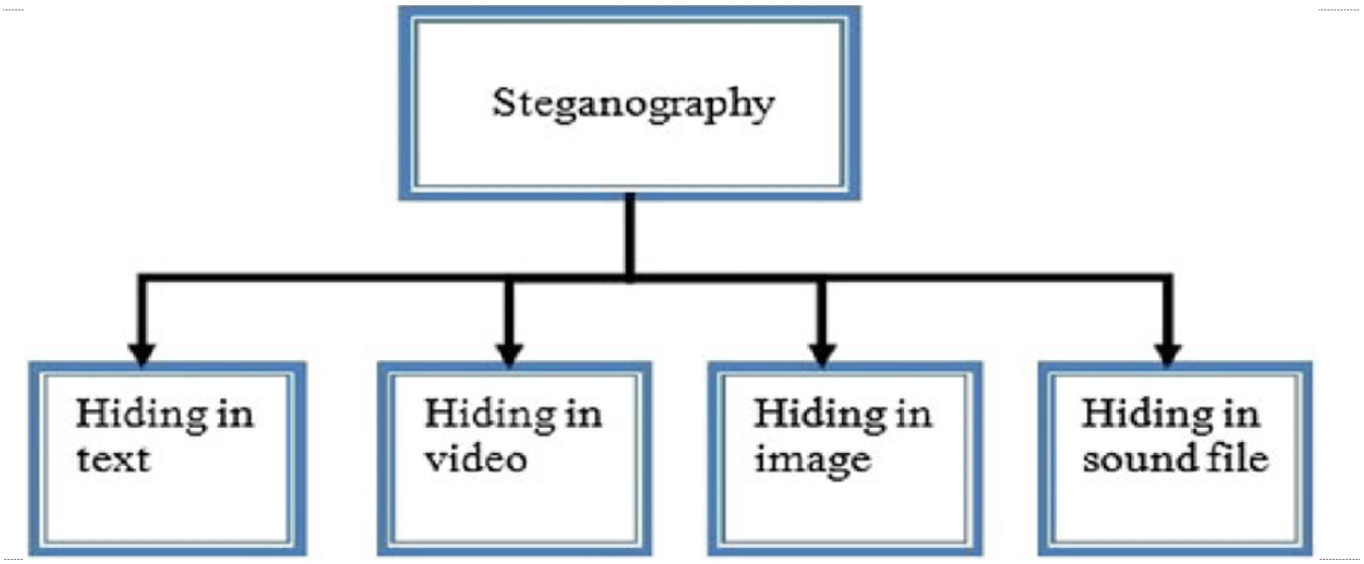
➢ Core i3 or higher,(cache- 3MB or 4MB recommended)

➢ Memory(RAM): Minimum 2GB; Recommended 4GB or above

## 2.Principles of Steganography:

### 2.1. Types of Steganography-

Data hiding involves hiding or securing the secret data (text, image, and video, audio) inside the digital data (text, image, and video, audio).




1. **Hiding Image Inside Another Image:**

The application allows users to hide one image inside another image. The user selects the two images to be used, and the application generates a new image that contains the hidden image. The user can download the new image or save it to the database.

2. **Text Inside Audio:**

The application allows users to hide text inside an audio file. The user selects the audio file and the text to be hidden, and the application generates a new audio file that contains the hidden text. The user can download the new audio file or save it to the database.

3. **Text Inside Sound file:**

In the context of hiding text inside a sound file, this involves embedding textual information within the audio data without significantly altering the perceived audio quality. There are various techniques for text steganography within sound files. One common method is frequency domain steganography, where the textual information is encoded into the frequency components of the audio signal.

4. **Text Inside Image:**

The application allows users to hide text inside an image. The user selects the image and the text to be hidden, and the application generates a new image that contains the hidden text. The user can download the new image or save it to the database.

**2.2 Basic Techniques-**

In the project, we will be implementing two common steganography techniques using the tkinter library in Python for graphical user interface (GUI) development. These techniques involve hiding secret text messages within digital images while maintaining the visual integrity of the images. Here are the details:

**1. LSB (Least Significant Bit) Substitution Technique:**

In this technique, the least significant bits of the pixel values in the image are modified to embed the secret text message. The pixel values in an image are typically represented in binary form. By replacing the least significant bits with the secret message bits, the changes are often imperceptible to the human eye. The tkinter GUI will provide options to select an image file and input the secret text message. The LSB substitution algorithm will be applied to embed the message into the image. The modified image with the hidden message can be saved as a new file.

**2. Image Masking Technique:**

This technique involves using a masking image to hide the secret text message within the cover image. The masking image acts as a key or pattern to determine the pixel positions where the secret message will be embedded. The pixels in the cover image corresponding to the specific positions defined by the masking image will be modified to carry the secret message. The tkinter GUI will allow the user to select both the cover image, the masking image, and input the secret message. The masking algorithm will be applied to embed the secret message into the cover image. The resulting image with the hidden message can be saved as a new file.

**3. Transform Domain Techniques (e.g., Discrete Cosine Transform - DCT):**

Transform domain techniques involve transforming the cover image into a different domain, such as the frequency domain, before embedding the secret message. The Discrete Cosine Transform (DCT) is commonly used to convert the image into a frequency representation. The secret message is embedded by modifying the coefficients of the transformed image. The alterations are made in the frequency domain, allowing for more robust hiding and increased resistance to attacks. To extract the hidden message, the reverse transformation is applied to retrieve the modified coefficients and recover the secret message.

These techniques will provide users with the capability to hide and extract secret text messages within images using the graphical interface developed with tkinter. The step-by-step explanations and code implementation in the report will provide further details on how these techniques are applied using the tkinter library in Python.

## 3. Advantages and disadvantages of steganography:

Data hiding using image steganography has its own set of advantages and disadvantages. Let's explore them in detail:

**Advantages:**

1. Inconspicuousness: One of the key advantages of image steganography is its ability to hide data in plain sight. The hidden information is embedded within the pixels of an image, making it difficult to detect without specialized tools or knowledge.

2. Security through Obscurity: Steganography provides a form of security through obscurity. Since the existence of hidden data is not apparent, it adds an extra layer of protection. Attackers may not even realize that there is something hidden within the image, reducing the chances of unauthorized access.

3. Wide Applicability: Image steganography can be applied to various types of digital images, including photographs, graphics, and even video frames. This versatility allows for the concealment of different types of data, such as text, audio, or other digital files.

4. Ease of Transmission: Hidden data can be easily transmitted by sharing or transferring the image file through common communication channels like email, instant messaging, or social media platforms. This makes steganography a convenient method for covert communication.

5. Resistance to Interception: Steganography can provide resistance against interception and surveillance. Since the hidden data is concealed within an image, it is less likely to attract attention or trigger suspicion during transmission or storage.

**Disadvantages:**

1. Limited Capacity: The capacity for data hiding within an image is limited by factors such as image size, color depth, and the desired level of imperceptibility. Large amounts of data cannot be embedded without significantly degrading the image quality or making the hidden data more detectable.

2. Vulnerability to Attacks: Steganography is not foolproof and can be vulnerable to various attacks. Sophisticated analysis techniques, including statistical analysis, visual inspection, and steganalysis algorithms, can potentially detect the presence of hidden data.

3. Lossy Compression Impact: If the steganographic image undergoes lossy compression, such as JPEG compression, it can potentially alter or remove the hidden data. Lossy compression techniques

discard certain image information, which may affect the integrity of the hidden message.

4. Complexity of Extraction: Extracting the hidden data from an image requires the knowledge of the steganography technique used and, in some cases, a secret key or password. If the extraction process is not properly executed, it may result in the loss or corruption of the hidden data.

5. Legal and Ethical Concerns: The use of steganography raises legal and ethical concerns when it is employed for malicious purposes, such as concealing illegal activities or transmitting sensitive information without proper authorization. Misuse of steganography can have serious consequences and may be subject to legal penalties.

It is important to consider these advantages and disadvantages when employing data hiding using image steganography, and to ensure that it is used responsibly and within the legal framework.

# 4. Introduction to Tkinter:

Tkinter is a popular Python library used for creating graphical user interfaces (GUIs). It provides a set of tools and widgets that enable developers to build interactive and user-friendly applications. In the context of image steganography, Tkinter can be utilized to create a GUI-based application that allows users to hide and extract secret text messages within digital images.

The integration of Tkinter with image steganography techniques offers several benefits. It enhances the usability and accessibility of the steganography system by providing a visually appealing and intuitive interface for users to interact with. Tkinter's capabilities allow for the selection of image files, inputting secret text messages, displaying visual feedback, and performing the embedding and extraction processes seamlessly.

### 4.1 Overview of Tkinter Library in Python:

Tkinter is a standard library in Python, which means it comes pre-installed with most Python distributions. It provides a set of modules that allow developers to create GUI applications. Some key features and concepts of Tkinter include:

1. Widgets: Tkinter offers a wide range of pre-built widgets, such as buttons, labels, text boxes, and image display areas. These widgets serve as building blocks for constructing the GUI interface.

2. Event-Driven Programming: Tkinter follows an event-driven programming model, where actions or events, such as button clicks or key presses, trigger specific functions or methods. This allows for responsive and interactive user interfaces.

3. Layout Management: Tkinter provides different layout managers, such as pack, grid, and place, to arrange and position widgets within the GUI window. These layout managers facilitate the organization and alignment of various interface elements.

4. Event Binding: Tkinter allows developers to bind functions or methods to specific events. For instance, a function can be associated with a button click event to execute a particular action, such as initiating the embedding or extraction process.

**4.2 Explanation of GUI Development using Tkinter:**

When developing a GUI application using Tkinter for image steganography, the following steps are typically involved:

1. Creating a GUI Window: Initialize a main window or frame using Tkinter's `Tk()` or `Toplevel()` class. This window serves as the container for other GUI elements.

2. Adding Widgets: Utilize Tkinter's widget classes, such as `Button`, `Label`, `Entry`, and `Canvas`, to create the necessary components for the steganography application. These widgets are added to the main window or frames using layout managers.

3. Defining Event Handlers: Define functions or methods that handle specific events, such as button clicks or menu selections. These event handlers are responsible for executing the corresponding steganography operations, such as embedding or extracting data.

4. Binding Events: Associate the defined event handlers with the appropriate widgets using Tkinter's `bind()` method. This ensures that the specified functions are called when the associated events occur.

5. Displaying the GUI: Use the `mainloop()` method of the main window to start the GUI event loop. This loop continuously listens for user interactions and keeps the GUI responsive.

By leveraging the capabilities of Tkinter, developers can create an intuitive and interactive interface for users to perform image steganography operations seamlessly. The GUI-based application built using Tkinter simplifies the process of selecting images, inputting secret messages, and performing the embedding and extraction processes, thereby enhancing the overall user experience.

In the project, a detailed overview of Tkinter library in Python and its integration with the steganography techniques will be provided, along with step-by-step explanations and code implementation, to showcase the development of the GUI-based steganography application.

# 5. Implementing steganography using tkinter:

I'll provide a detailed explanation of the code and its implementation of steganography using Tkinter:

## 5.1 Modules used in the code:

The code example provided uses the following modules:

**1. tkinter (`Tk`, `Label`, `Button`, `Text`, `filedialog`):**

- `Tkinter` is the standard GUI toolkit library for Python.
- `Tk` is used to create the main window.
- `Label` is used to display text labels in the GUI.
- `Button` is used to create buttons with associated functions.
- `Text` is used to create a multiline text entry field.
- `filedialog` is used to open a file dialog for selecting an image file.

**2. PIL (`Image`, `ImageTk`):**

- `PIL` (Python Imaging Library) is a powerful library for image processing in Python.
- `Image` is used to open and manipulate images.
- `ImageTk` is used to create a Tkinter-compatible photo image object from an image.

**3. cryptography (`Fernet`, `InvalidToken`):**

- `cryptography` is a library for secure communications and cryptography in Python.
- `Fernet` is a high-level symmetric encryption module in `cryptography`.
- `Fernet` is used to generate a key, encrypt messages, and decrypt messages.
- `InvalidToken` is an exception class in `cryptography` that is raised when decryption fails due to an invalid or corrupted token.

To use these modules, make sure they are installed and import them into your Python script using the appropriate import statements. For example:

python

```
from tkinter import Tk, Label, Button, Text, filedialog
from PIL import Image, ImageTk
from cryptography.fernet import Fernet, InvalidToken
```

Ensure that you have the required modules installed by running the following commands in your command-line interface:

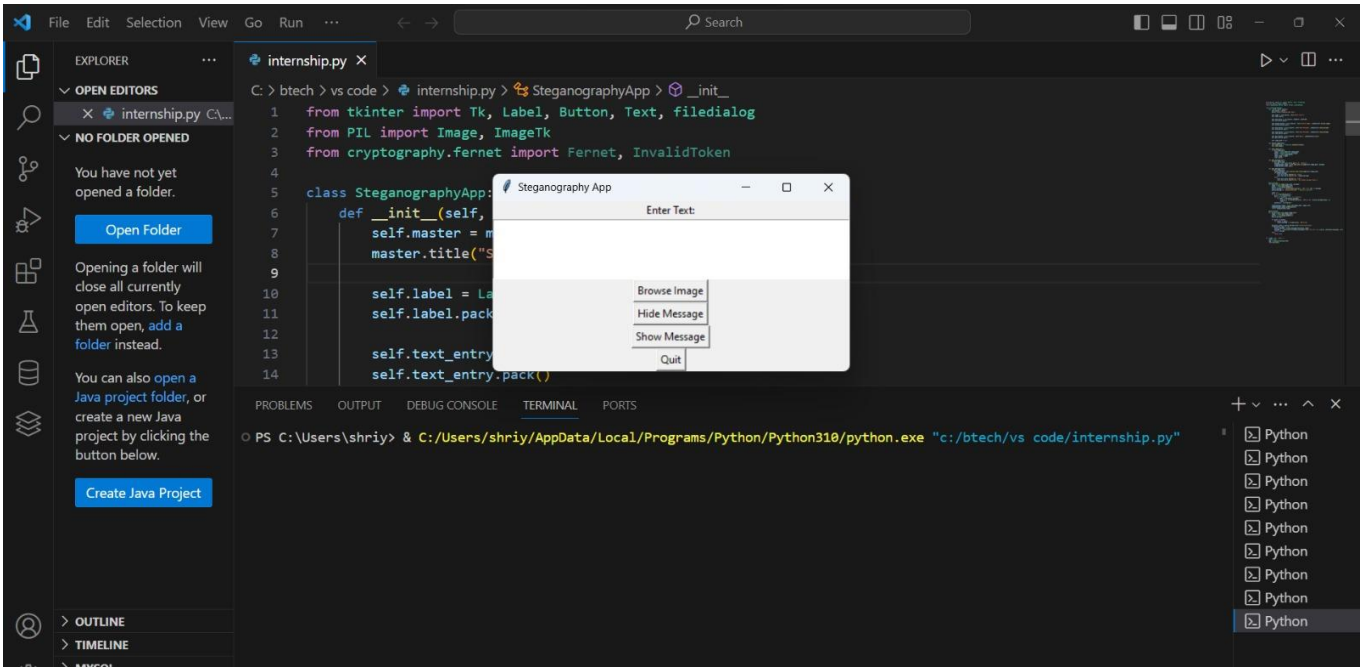
- shell
  - ➢ pip install tkinter
  - ➢ pip install pillow
  - ➢ pip install cryptography

Note: The module names may vary slightly depending on your Python version and installation.

## 5.2 Step by step explanation of the code:

**1. GUI Setup:**

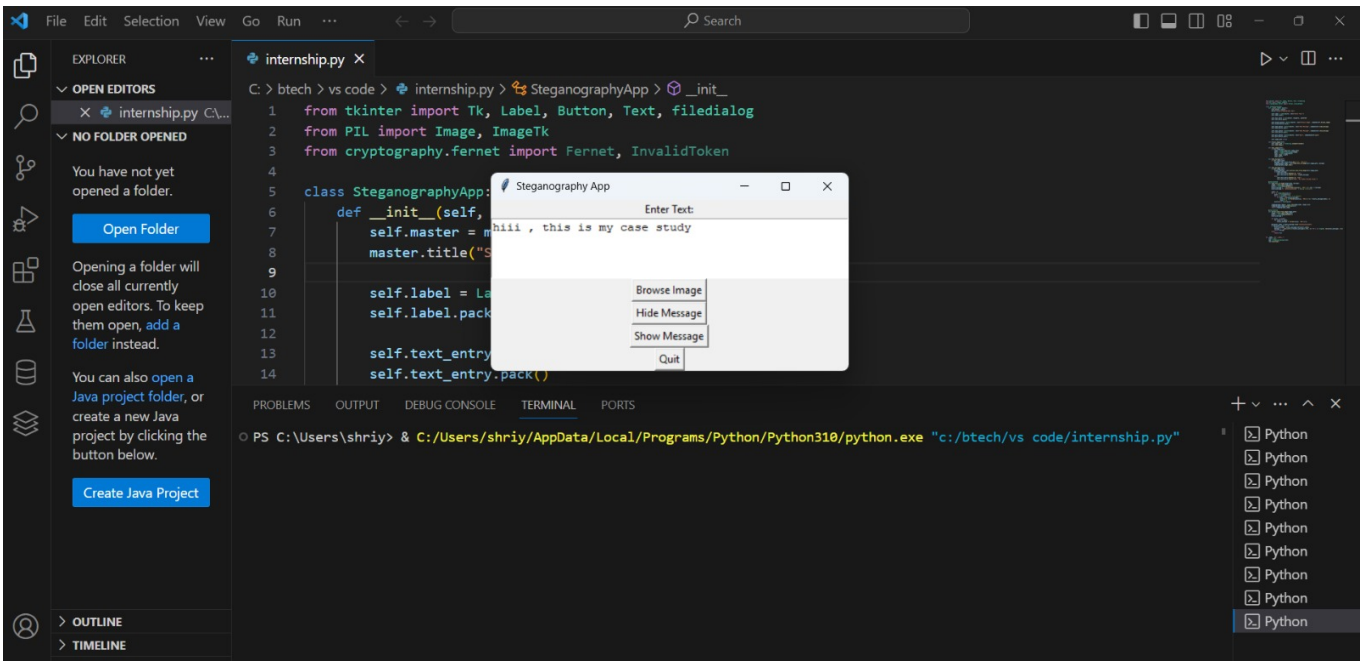
- The code imports necessary libraries: `tkinter` for GUI, `PIL` for image processing, and `cryptography` for encryption.
- It creates a Tkinter window and a `SteganographyApp` class to manage the application's logic.
- The `__init__` method initializes the GUI elements:
  - Label for text input
  - Text box for entering the message
  - Buttons for browsing images, hiding/encrypting, showing/decrypting, and quitting




**2. Image Browsing and Display:**

- The `browse_image` method allows users to select an image using a file dialog.

- The `show_image` method opens the selected image, converts it to a Tkinter-compatible format using `ImageTk`, and displays it in a label.




**3. Hiding the Message:**

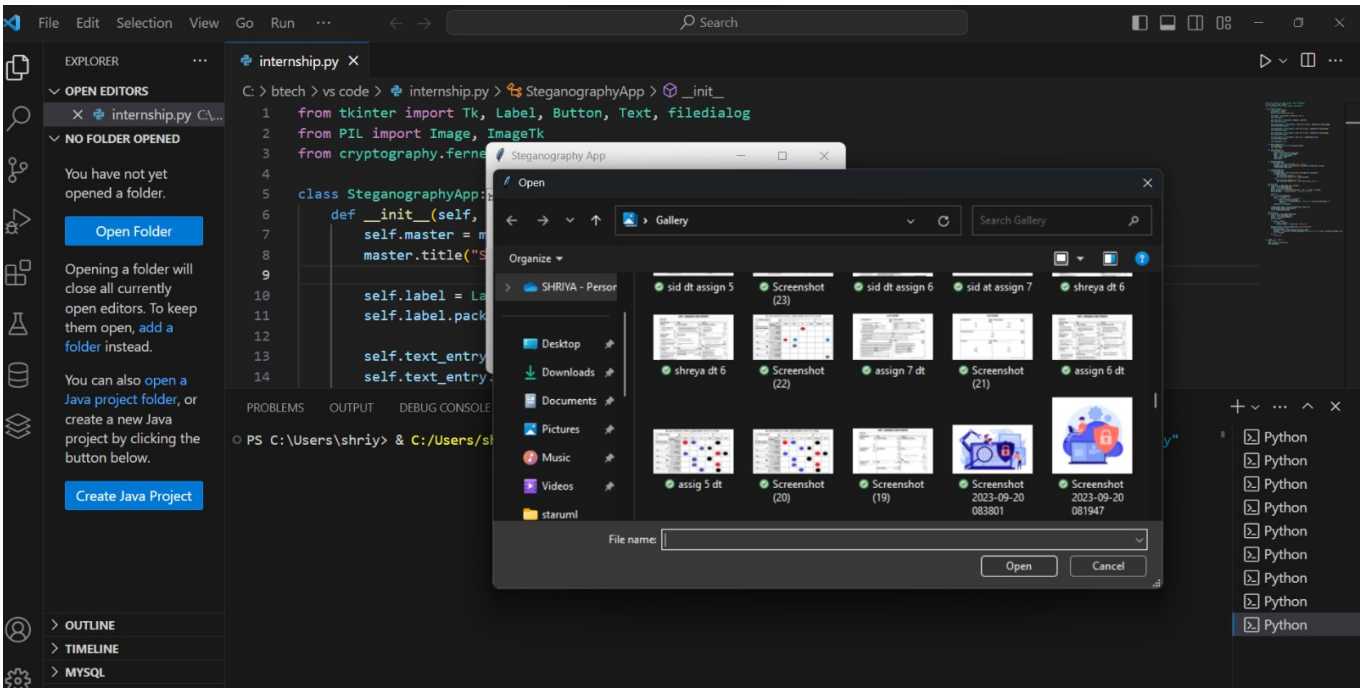
hide_message:

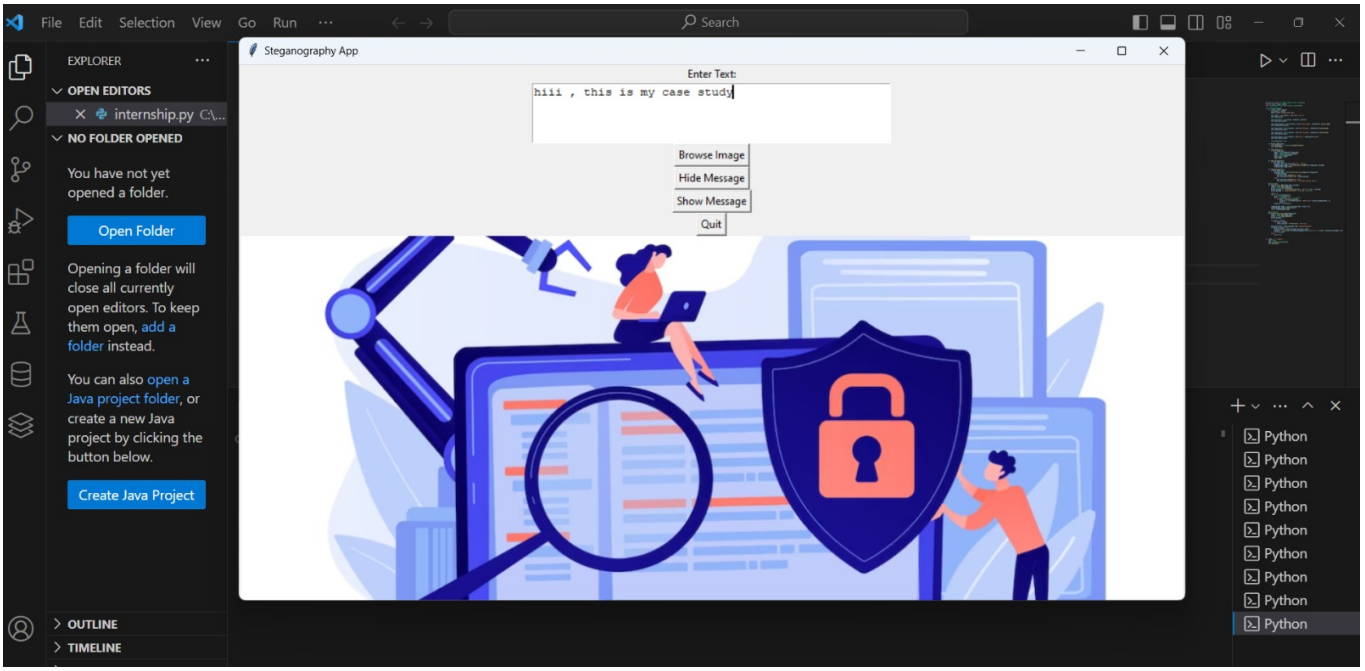
-Retrieves the message from the text box

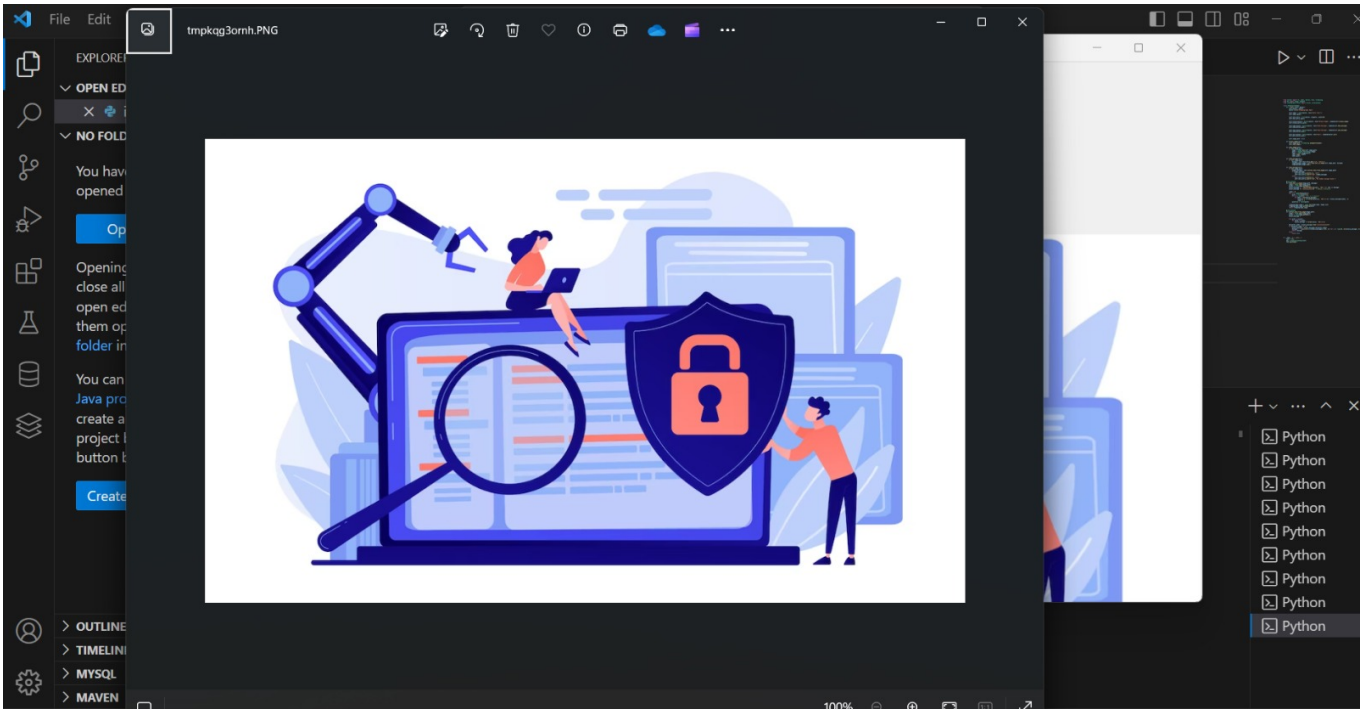
-Encrypts it using Fernet

-Embeds the encrypted message in the image using hide_text_in_image

-Displays the steganographic image






**4. Showing the Message:**

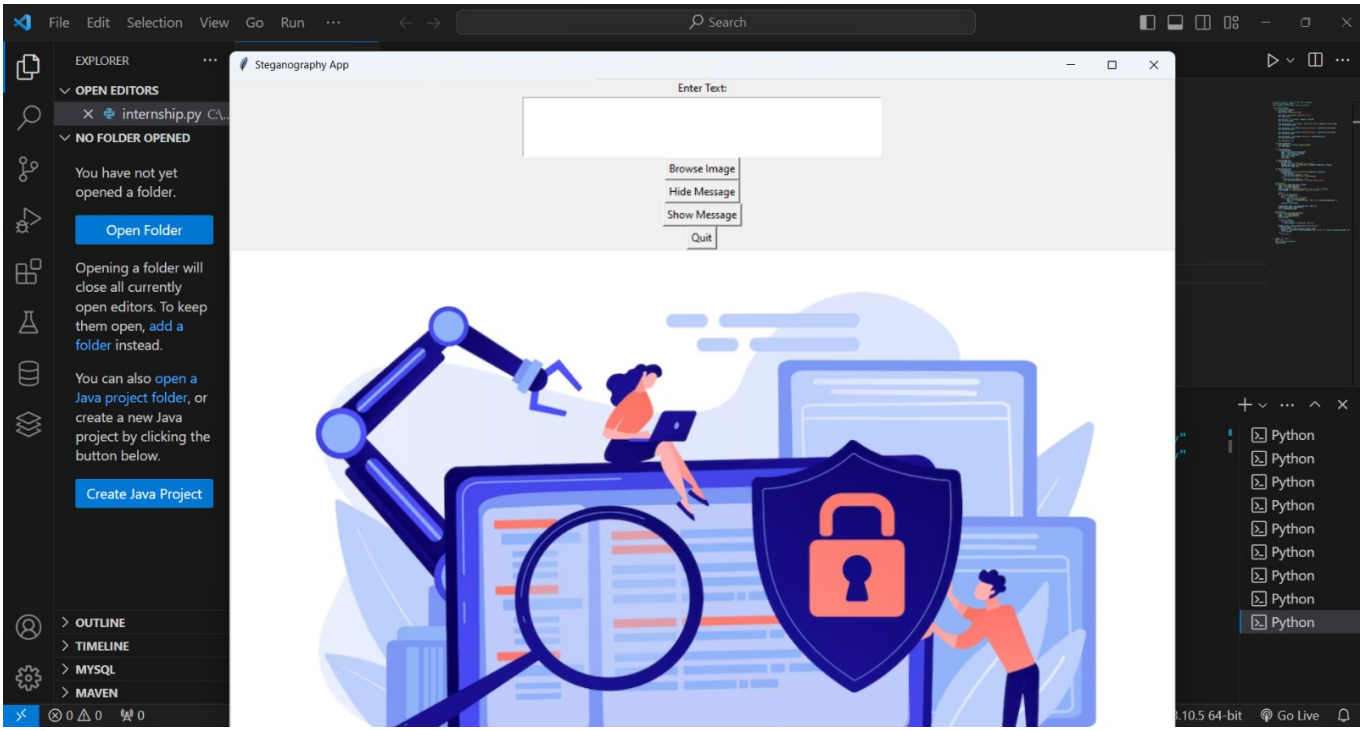
show_message:

-Extracts the hidden message from the image using extract_text_from_image

-Decrypts it using Fernet

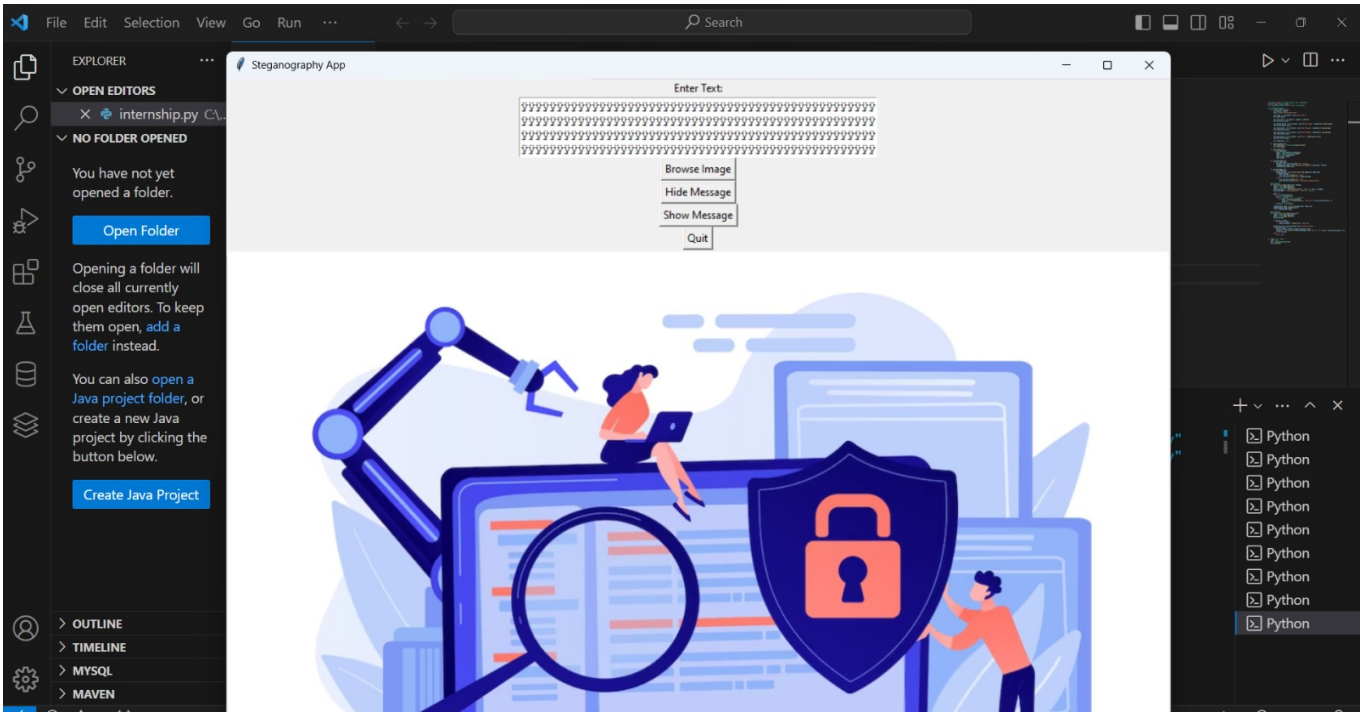
-Displays the decrypted message in the text box

-Handles cases where no message is found




## 5. Encryption and Decryption:

encrypt_message and decrypt_message: Handle encryption and decryption using Fernet




## 6. Steganography Functions:

hide_text_in_image:

-Opens the image

-Converts the message to binary

-Appends a delimiter

-Embeds the binary message in image pixels

-Creates a new steganographic image

extract_text_from_image:

-Opens the image

-Extracts the binary message from pixels

-Identifies the delimiter

-Returns the extracted message

```python
@staticmethod
def hide_text_in_image(image_path, message):
    image = Image.open(image_path)
    pixels = list(image.getdata())
    binary_message = ''.join(format(ord(char), '08b') for char in message)
    binary_message += '1111111111111110'  # Adding a delimiter

    index = 0
    for i in range(len(pixels)):
        pixel = list(pixels[i])
        for j in range(3):  # R, G, B channels
            if index < len(binary_message):
                pixel[j] = int(format(pixel[j], '08b')[:-1] + binary_message[index], 2)
                index += 1
        pixels[i] = tuple(pixel)

    steganography_image = Image.new(image.mode, image.size)
    steganography_image.putdata(pixels)
    return steganography_image
```

```
@staticmethod
def extract_text_from_image(image_path):
    image = Image.open(image_path)
    pixels = list(image.getdata())
    binary_message = ''

    for pixel in pixels:
        for value in pixel:
            binary_message += format(value, '08b')[-1]

    delimiter_index = binary_message.find('1111111111111110')
    if delimiter_index != -1:
        binary_message = binary_message[:delimiter_index]
        message = ''.join([chr(int(binary_message[i:i+8], 2)) for i in range(0, len(binary_message), 8)])
        return message
    else:
        return None
```

**7. Main Execution:**

- The `if __name__ == "__main__":` block:
  - Creates the Tkinter main window.
  - Instantiates the `SteganographyApp` class.
  - Starts the Tkinter event loop to handle user interactions.

```
if __name__ == "__main__":
    root = Tk()
    app = SteganographyApp(root)
    root.mainloop()
```

The main execution block checks if the module is being run directly. It creates an instance of the Tk class, initializes the SteganographyApp with the Tk instance, and starts the main event loop using root.mainloop(). This ensures that the GUI application runs properly.

**Key Points:**

-The code implements a simple steganography application with a user-friendly GUI.

-It utilizes Tkinter for the GUI, PIL for image processing, and Fernet for encryption.

-It allows users to hide and extract text messages within images, enhancing confidentiality.

Overall, this code implements a simple steganography application using Tkinter, allowing users to hide and extract text messages within images using encryption techniques.

# 5. CONCLUSIONS:

## 5.1 Conclusion:

Image steganography is a perfect solution for the information leaks and data insecurity. Image steganography allows secretly passing on the information between the users and the alterations and the other changes are not visible. It allows lossless compression of the cover image. The size, the quality of the original cover image and the secret text embedded image (stego image) remains unchanged. There are the alterations in the pixels but not visible to the human eye. That is the biggest reason that this technique is one of the best techniques used in the information security for the data hiding and secured communication.

Nowadays when cryptographic methods are used for communication, the developers and software engineers are looking up for the steganography techniques to make the communication highly confidential, secured and safe. LSB steganography is in demand as it has been the most popular encoding algorithm due to its simplicity and high accuracy. According to the various digital forensics organizations, many of the today's algorithms are unable to transfer the information in complete secured way but the Steganography combined with Cryptography has made the unauthorized parties almost impossible to track the secret codes hidden in the cover files. Therefore steganography in the coming years for sure will be a big thing for the Information Security purposes.

## 5.2 Applications of the Project:

1. Securing data while Breaches: Image steganography is an important technique used in digital forensics area during the occurrences of the data breaches or the data leak incidents. Basically whenever a data breach occurs, crucial data can be hidden using the steganography technique.

2. End-to-End Private communication: If a client and a server want to share some peer to peer confidential information over a network, then they can use the data hiding and steganography technique to avoid any outer unauthorized interference.

3. Protection of Intellectual property Rights and Ownership: Data hiding through image steganography can be used to hide the owner's designations or signatures secretly in any original cover image, audio, video or text so that the ownership rights remains with the owner and others cannot steal the owner's work.

4. Multimedia Fingerprinting: This is similar to the ownership rights. The owner of the media can embed the secret copyrights on the covers so that the originality of the owner's work remains unaltered.

5. Smart Identity Cards: Various Intelligence organizations make use of the image steganography to hide the details of their employees in their ID card images.

## 5.2 Future Scope of the Project:

Though our project is able to reach highly accepted results and is able to hide data very efficiently, it can definitely be improved in some areas. Our project works on hiding the textual data behind the cover images but in future, algorithms can be implemented to hide other format of data in it as well like audios, videos etc. To make the communication more secured and impossible to hack, various existing ciphers can be implemented with the steganography algorithms. Our algorithms works on the lossless format of images or converts the lossy formats to lossless, but various techniques could be used in future for decreasing the amount of compression and the data loss in lossy formats, so that Image steganography by LSB can be implemented for the lossless form of cover data.