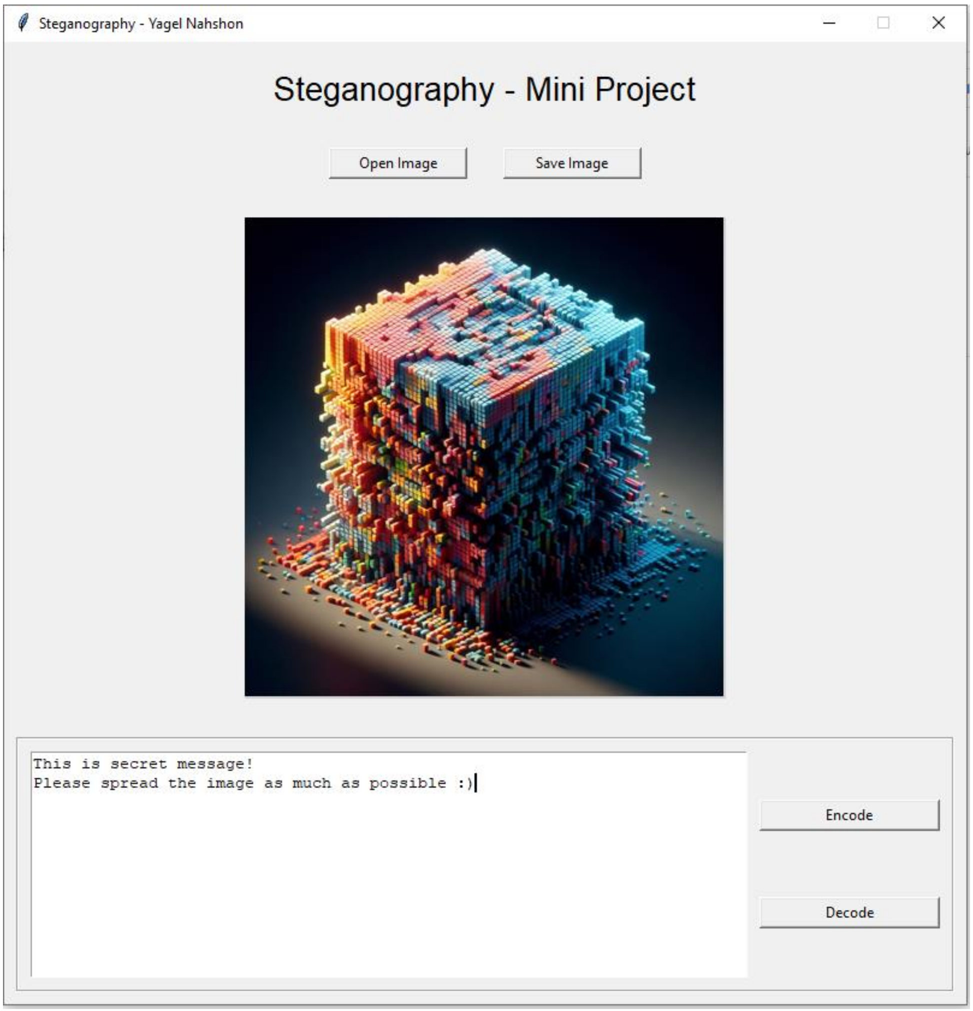
Mini Project - Steganography Tool

Name: Yagel Nahshon

ID: 207055781



```
This is secret message!
Please spread the image as much as possible :)
```

1. Purpose and Functionality

The Steganography Tool is designed to encode and decode text messages within image files using the least significant bit (LSB) method. The tool serves the purpose of hiding sensitive information within seemingly innocuous image files, providing a means of covert communication.

The tool allows users to:

- Open an image file (in PNG or JPG format).
- Encode a text message into the selected image.
- Save the encoded image with the hidden message.
- Decode a hidden message from an encoded image.

2. Problems Encountered

During the development of the Steganography Tool, several challenges were encountered:

- **Library Compatibility:** Ensuring compatibility and proper integration of external libraries, such as tkinter for GUI development and stegano for steganography functionalities.
- **Error Handling:** Implementing robust error handling mechanisms to handle edge cases and user input errors effectively.
- **User Experience:** Designing an intuitive user interface that provides clear instructions and feedback to the user.

3. Strengths and Weaknesses

Strengths

- **Innocuous Appearance:** The tool allows users to create an image that looks innocent to casual observers, even though it may contain hidden messages. This feature enhances the tool's effectiveness for covert communication purposes.
- **User-Friendly Interface:** The tool features a simple and intuitive graphical user interface (GUI) that allows users to easily interact with the application.
- **Functionality**: The tool effectively achieves its primary objective of encoding and decoding text messages within image files.
- **Flexibility**: Users can select any PNG or JPG image for encoding and decoding messages, providing flexibility in usage.

Weaknesses:

- **Limited Steganographic Techniques:** The tool only employs the LSB method for steganography, which may not be the most robust technique for hiding information.
- **Security Concerns:** Since the tool utilizes basic steganographic techniques, it may not provide sufficient security for highly sensitive information.
- **Dependency on External Libraries:** The tool relies on external libraries such as tkinter and stegano, which may introduce compatibility issues or dependency concerns in certain environments.

4. Tool Description and Code Documentation

Functions:

- **open_image():** Opens an image file selected by the user and displays it on the canvas.
- **save_image():** Saves the encoded image with the hidden message.
- **encode_message():** Encodes the text message entered by the user into the selected image.
- **decode_message():** Decodes the hidden message from the selected encoded image.

Workflow:

1. Launch the application.
2. Open an image file by clicking the "Open Image" button.
3. Enter a text message in the text box.
4. Click the "Encode" button to hide the message within the image.
5. Save the encoded image using the "Save Image" button.
6. The encoded image will be saved in the project directory as "EncodedMessage.png".
7. To decode a message from an encoded image, open the encoded image and click the "Decode" button.

5. References and Tools Used

- **Tkinter:** Python library for creating graphical user interfaces.
- **PIL (Python Imaging Library):** Library for image processing.
- **stegano:** Python library for steganography.