📄 Steganography Python GUI Code Explanation

This file explains the Python code used for building the **Image Steganography GUI Tool** step-by-step in simple language.

🔰 Imported Libraries:

```
from tkinterdnd2 import DND_FILES, TkinterDnD
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image
import stepic
import os
```

Explanation:

- `tkinter` – used to build the GUI.
- `tkinterDnD2` – adds drag-and-drop support.
- `filedialog` – to browse and choose a file.
- `messagebox` – to show popup messages.
- `PIL.Image` – for opening and converting images.
- `stepic` – used to hide/extract text inside images.
- `os` – to work with file paths.

📁 Functions:

1. browse_file()

```
def browse_file():
    file = filedialog.askopenfilename(...)
    if file:
        file_path.set(file)
        drop_label.config(text=os.path.basename(file))
        check_inputs()
```

👉 Lets the user browse for an image file and saves its path.

2. drop_file(event)

```
def drop_file(event):
    file = event.data.strip("{}")
    if os.path.isfile(file):
        file_path.set(file)
        drop_label.config(text=os.path.basename(file))
        check_inputs()
```

👉 When a user drags and drops a file into the box, this function captures that file path.

3. check_inputs()

```
def check_inputs():
    if mode.get() == "hide":
        if file_path.get() and message_entry.get():
            action_btn.config(state="normal")
        else:
            action_btn.config(state="disabled")
    elif mode.get() == "extract":
        if file_path.get():
            action_btn.config(state="normal")
        else:
            action_btn.config(state="disabled")
```

👉 This function enables or disables the **"Next" button** depending on whether the required inputs are provided.

4. switch_mode()

```
def check_inputs():
    if mode.get() == "hide":
        if file_path.get() and message_entry.get():
            action_btn.config(state="normal")
        else:
            action_btn.config(state="disabled")
    elif mode.get() == "extract":
        if file_path.get():
            action_btn.config(state="normal")
        else:
            action_btn.config(state="disabled")
```

👉 When user selects "Hide" or "Extract", this function updates the UI accordingly.

5. clear_all()

```
def clear_all():
    file_path.set("")
    drop_label.config(text="Drag & Drop Image Here or Click Browse")
    message_entry.delete(0, tk.END)
    action_btn.config(state="disabled")
```

👉 Clears the form so the user can start fresh.

6. perform_action()

```
def perform_action():
    filepath = file_path.get()
    ext = filepath.split('.')[-1].lower()
    ...
```

👉 This is the **main function**. It does the hiding or extracting based on the selected mode.

🔐 If "Hide Message" is selected:

```
msg = message_entry.get()
image = Image.open(filepath).convert('RGB')
encoded_img = stepic.encode(image, msg.encode())
encoded_img.save(output_path)
```

✅ Message is hidden in the image using **LSB technique** and saved in Downloads folder.

🔓 If "Extract Message" is selected:

```
image = Image.open(filepath)
decoded = stepic.decode(image)
messagebox.showinfo("Hidden Message Found", decoded)
```

✅ Reads the hidden message from image and shows it in a popup.

🖥️ GUI Setup:

```
root = TkinterDnD.Tk()
root.title("Steganography Tool")
root.geometry("520x350")
```

✅ Starts the window, sets title and size.

📋 GUI Components:

- **Radio buttons** for selecting mode
- **Drag & Drop label** for dropping files
- **Browse Button** to choose files manually
- **Entry box** to write the message (only in hide mode)
- **Action button** to perform the task

📜 Summary:

- Code is written using **Tkinter**.
- You can **hide** secret text in JPG, PNG, BMP images.
- You can also **extract** text from encoded image.
- The result image goes to the **Downloads** folder.