AccFileOpenner.py

```
import os
import re
import tkinter as tk
from tkinter import messagebox, filedialog
import shutil
import sys
import time
from gui import BaseApp
from tkinterdnd2 import TkinterDnD
import subprocess

sys.path.append(os.path.dirname(os.path.dirname(__file__)))
import _Exe_Util

FILE_CATALOG = {
    "indesign": {
        "lock_file_extension": ".idlk",
        "lock_file_begin_template": "~{}",
        "file_extension": ".indd",
    },
    "rhino": {
        "lock_file_extension": ".rhl",
        "lock_file_begin_template": "{}",
        "file_extension": ".3dm",
    },
    "cad": {
        "lock_file_extension": ".dwl",
        "lock_file_begin_template": "{}",
        "file_extension": ".dwg",
    },
    "pdf": {
        "lock_file_extension": None,
        "lock_file_begin_template": "",
        "file_extension": ".pdf",
    },
    "excel": {
        "lock_file_extension": ".xlsx",
        "lock_file_begin_template": "~${}",
        "file_extension": ".xlsx",
        "prefer_cloud": True
    },
    "word": {
        "lock_file_extension": ".docx",
        "lock_file_begin_template": "~${}",
        "file_extension": ".docx",
        "prefer_cloud": True
    },
    "photoshop": {
        "lock_file_extension": None,
        "lock_file_begin_template": "",
```

gui.py

```
import os
import tkinter as tk
from tkinter import scrolledtext
from PIL import Image, ImageTk
import datetime
from tkinterdnd2 import DND_FILES
import sys
sys.path.append(os.path.dirname(os.path.dirname(__file__)))
```

```
import _GUI_Base_Util
EXPIRATION_DATE = datetime.date(2025, 1, 1)

class BaseApp(_GUI_Base_Util.BaseGUI):
    def __init__(self, root):
        self.root = root
        self.root.configure(bg=self.BACKGROUND_COLOR_HEX)
        self.root.geometry("1000x800")
        self.setup_icon()
        self.create_widgets()
        self.setup_bindings()
        self.update_title_with_days_left()
        self.create_dashboard()

    def setup_icon(self):
        icon_path = os.path.join(os.path.dirname(__file__), "icon_ennead-e.ico")
        self.root.iconbitmap(icon_path)
        self.logo_path = os.path.join(os.path.dirname(__file__), "logo.png")
        self.logo_image = Image.open(self.logo_path)
        self.logo_image = self.logo_image.resize((self.logo_image.width // 2,
self.logo_image.height // 2), Image.LANCZOS)
        self.logo_photo = ImageTk.PhotoImage(self.logo_image)

    def create_widgets(self):
        self.logo_label = tk.Label(self.root, image=self.logo_photo,
bg=self.BACKGROUND_COLOR_HEX)
        self.logo_label.grid(row=0, column=0, columnspan=3, sticky="nsew")

        self.editing_files_frame = tk.Frame(self.root,
bg=self.BACKGROUND_COLOR_HEX)
        self.editing_files_frame.grid(row=1, column=0, columnspan=3,
sticky="nsew")

        self.editing_files_text =
scrolledtext.ScrolledText(self.editing_files_frame, width=60, height=15,
bg=self.BACKGROUND_COLOR_HEX, fg='white', font=('Helvetica', 12, 'bold'),
wrap=tk.WORD)
        self.editing_files_text.pack(fill=tk.BOTH, expand=True)
        self.editing_files_text.config(height=10)
        self.editing_files_text.config(height=10)  # Setting fixed height

        note = "The file will open automatically after picked/dropped."
        note += "\nAccepting File types of Indesign, Rhino, Word, Excel, PDF,
Photoshop, Illustrator"
        self.instructions_label = tk.Label(self.root, text=note,
bg=self.BACKGROUND_COLOR_HEX, fg='white', font=('Helvetica', 12),
wraplength=800, justify=tk.LEFT)
        self.instructions_label.grid(row=2, column=0, columnspan=3, sticky="nw",
padx=20, pady=10)

        for i in range(3):
            self.root.grid_columnconfigure(i, weight=1)
        self.root.grid_rowconfigure(1, weight=1)
```