MacroMouse - Dynamic Variable Insertion + Reference File Support

# Project Objective

Enhance MacroMouse to support two features:  
1. Dynamic placeholders inside macro text, which prompt the user to fill in values at copy time.  
2. Ability to upload and view a user-maintained reference file (plain .txt), so the user can consult their tag explanations, notes, or best practices on-demand from the app menu.

# Key Requirements

- Use existing XML macro structure for macros.

- Introduce optional Reference File (`reference.txt`) that is user-selectable via menu.

- Modify the existing `copy\_macro()` function to:

• Scan macro text for placeholders in `{{tag}}` format.

• For each found tag, prompt the user to input a value using a simple input dialog.

• Replace all instances of the tag with the entered value.

• Optionally warn the user if unresolved tags remain.

# Reference File Handling

- Add a new menu item: "Reference File".

- Allow user to upload/select a `.txt` file containing notes, tag definitions, etc.

- Store path in `config.json` for future sessions.

- Add a simple "View Reference File" option that opens the file in the system default text viewer.

# Implementation Steps

## 1. Tag Placeholder Format

Use double curly braces `{{}}` as placeholders inside macro text. Example:

I'll get back to you by {{promised\_date}}.

## 2. Modify `copy\_macro()`

Insert this logic before the `pyperclip.copy()`:

import re  
import tkinter.simpledialog as sd  
  
pattern = r'\{\{(.\*?)\}\}'  
placeholders = set(re.findall(pattern, content))  
  
for ph in placeholders:  
 user\_input = sd.askstring("Fill Placeholder", f"Enter value for '{ph}':")  
 if user\_input is not None:  
 content = content.replace(f"{{{{{ph}}}}}", user\_input)

## 3. Add Reference File Support

1. Add 'Reference File' menu to the app.  
2. Allow user to select/upload a `.txt` file.  
3. Store file path in `config.json` under `reference\_file`.  
4. When user clicks 'View Reference File', open the file in the default OS viewer.  
5. Optionally add a shortcut button near macro preview to open reference quickly.

## 4. Keep It Stupid Simple

Do not over-engineer this.  
- No need for tag validators or fancy editors.  
- Assume the user knows what they're typing.  
- Focus on simple file management and runtime prompts only.

# Summary

This enhancement keeps the system flexible and user-controlled, while giving the user an easy, always-available place to store their tag references.  
No tag libraries, no enforcement—just clean prompting and user-maintained reference.