

Code:

```
import tkinter as tk
from tkinter import messagebox, scrolledtext

# 1. Simulated Blockchain Logic (No Web3/External connection needed)
class SimulatedBlockchain:
    def __init__(self):
        self.stored_message = "No message stored yet."

    def set_message(self, message):
        self.stored_message = message

    def get_message(self):
        return self.stored_message

# 2. GUI Application
class SmartContractApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Blockchain Message Storage Simulation")
        self.root.geometry("600x500")
        self.blockchain = SimulatedBlockchain()
        self.create_widgets()

    def create_widgets(self):
        tk.Label(self.root, text="Solidity Message Storage (Simulation)", font=("Arial", 14, "bold")).pack(pady=10)

        tk.Label(self.root, text="Enter Message:").pack()
        self.message_entry = tk.Entry(self.root, width=50)
        self.message_entry.pack(pady=5)

        tk.Button(self.root, text="Store Message", command=self.store_message, bg="green", fg="white").pack(pady=5)
        tk.Button(self.root, text="Retrieve Message", command=self.retrieve_message, bg="blue", fg="white").pack(pady=5)

        tk.Label(self.root, text="Output Log:").pack(pady=(10, 0))
        self.output_box = scrolledtext.ScrolledText(self.root, width=60, height=8, state="disabled")
        self.output_box.pack(pady=10)

        # This is where your previous code had the Syntax Error
        tk.Label(self.root, text="Contract Code Reference:").pack()
        self.code_display = scrolledtext.ScrolledText(self.root, width=60, height=8)
        self.code_display.insert(tk.END, 'pragma solidity ^0.8.0;\ncontract MessageStorage {\n    string private message;\n    function setMessage(string memory _msg) public {\n        message = _msg;\n    }\n}\n\n// SPDX-License-Identifier: MIT')
        self.code_display.config(state="disabled")
        self.code_display.pack()

    def store_message(self):
        msg = self.message_entry.get().strip()
        if msg:
            self.blockchain.set_message(msg)
            self.display_output(f'SUCCESS: Stored "{msg}" in simulated state.')
        else:
            messagebox.showwarning("Input Error", "Please enter a message.")

    def retrieve_message(self):
        msg = self.blockchain.get_message()
        self.display_output(f'RETRIEVED: {msg}')

    def display_output(self, text):
        self.output_box.config(state="normal")
        self.output_box.insert(tk.END, text + "\n")
        self.output_box.config(state="disabled")
        self.output_box.see(tk.END)

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

Output :



