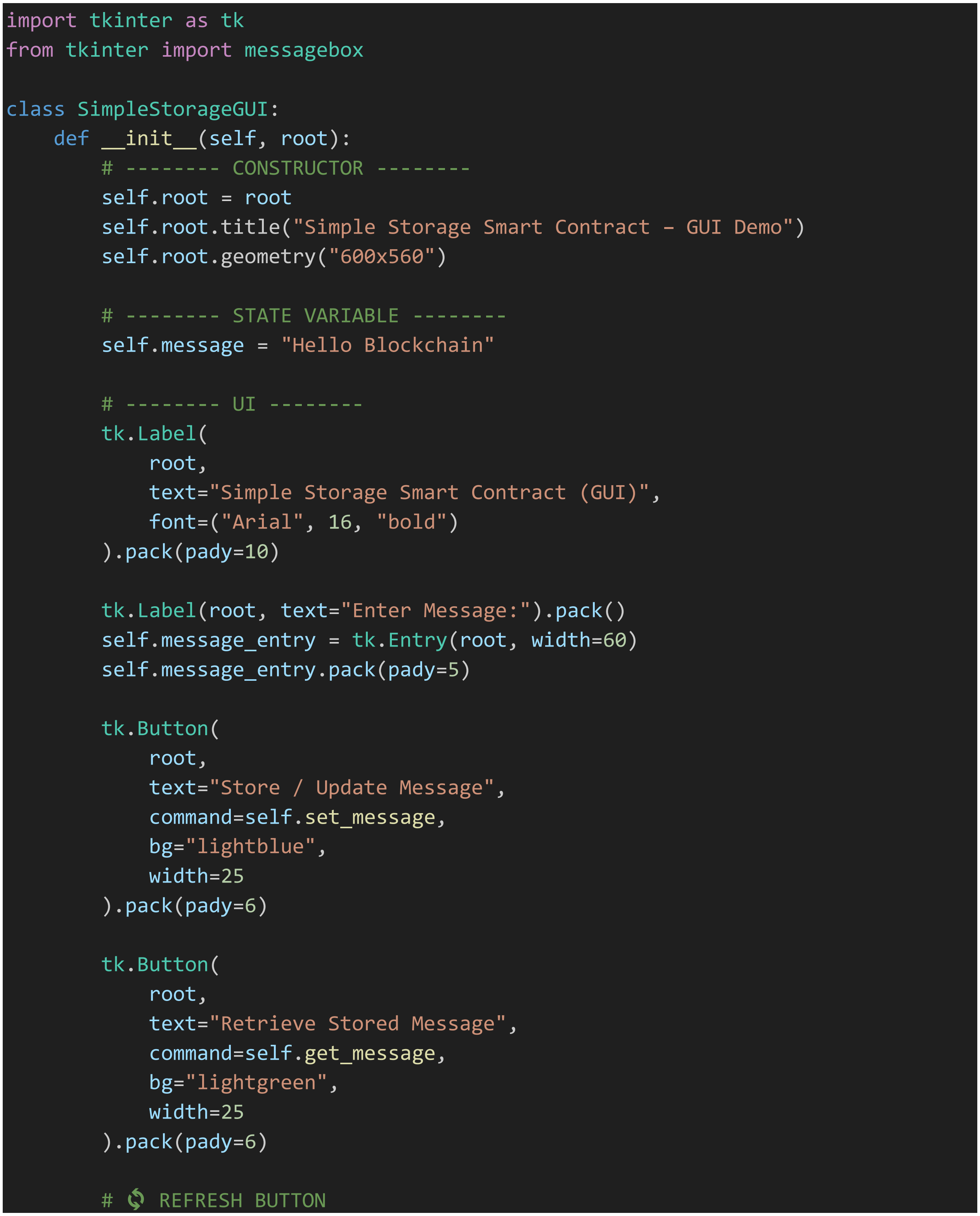
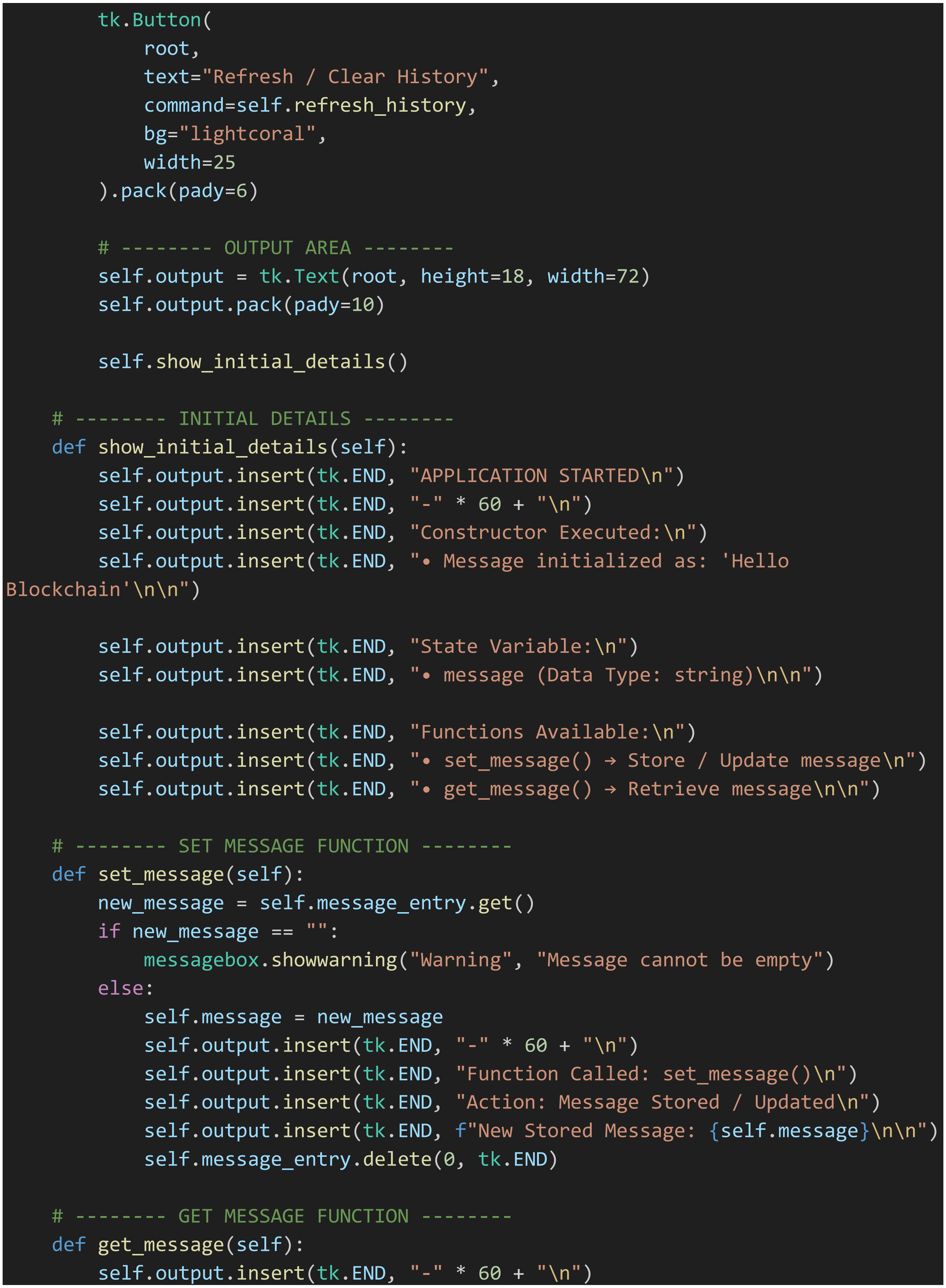
ASSINGMENT 3

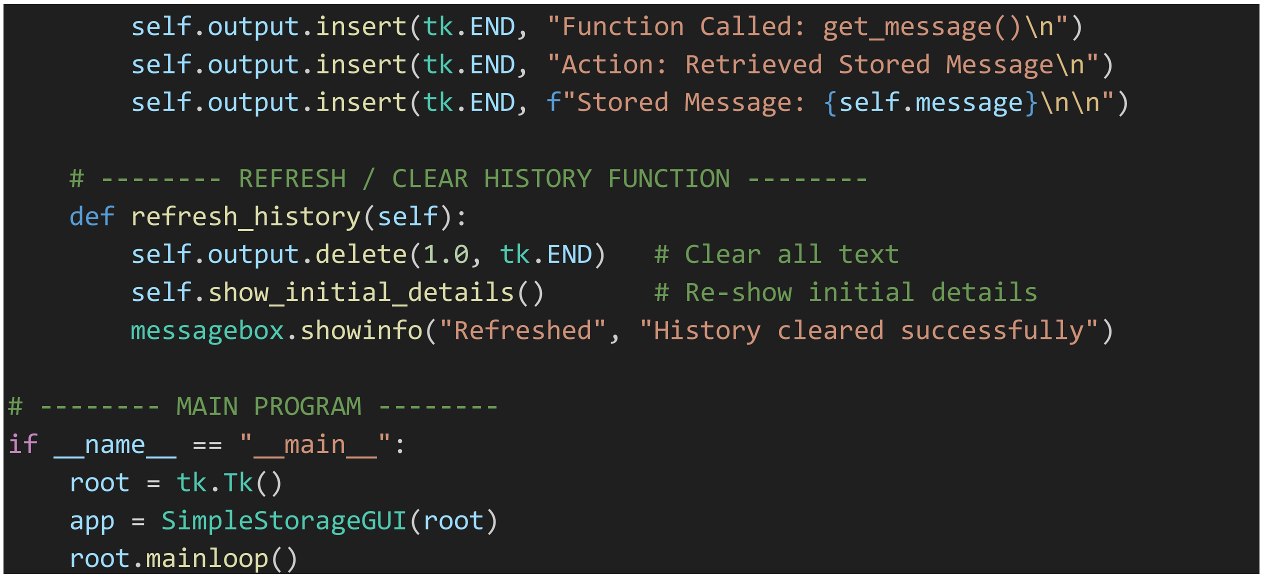
Code:



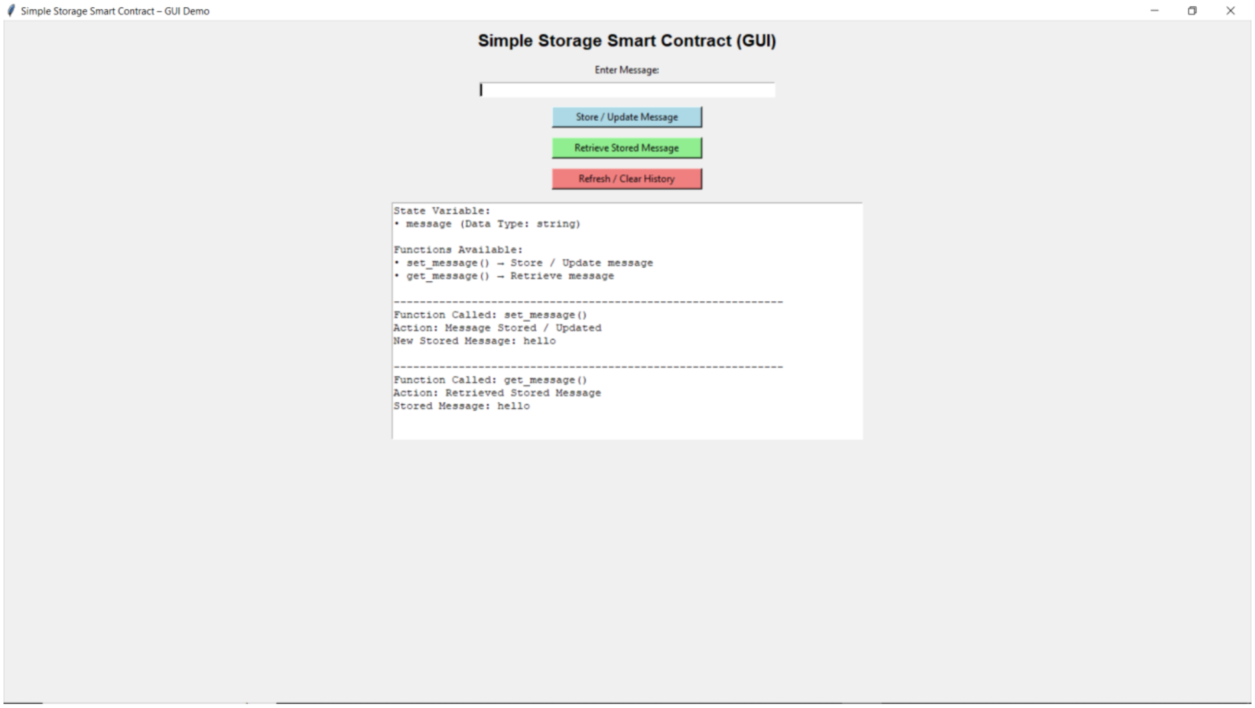
```
import tkinter as tk
from tkinter import messagebox

class SimpleStorageGUI:
    def __init__(self, root):
        # -------- CONSTRUCTOR --------
        self.root = root
        self.root.title("Simple Storage Smart Contract – GUI Demo")
        self.root.geometry("600x560")

        # -------- STATE VARIABLE --------
        self.message = "Hello Blockchain"

        # -------- UI --------
        tk.Label(
            root,
            text="Simple Storage Smart Contract (GUI)",
            font=("Arial", 16, "bold")
        ).pack(pady=10)

        tk.Label(root, text="Enter Message:").pack()
        self.message_entry = tk.Entry(root, width=60)
        self.message_entry.pack(pady=5)

        tk.Button(
            root,
            text="Store / Update Message",
            command=self.set_message,
            bg="lightblue",
            width=25
        ).pack(pady=6)

        tk.Button(
            root,
            text="Retrieve Stored Message",
            command=self.get_message,
            bg="lightgreen",
            width=25
        ).pack(pady=6)

        # 🔄 REFRESH BUTTON
```



```
        tk.Button(
            root,
            text="Refresh / Clear History",
            command=self.refresh_history,
            bg="lightcoral",
            width=25
        ).pack(pady=6)

        # -------- OUTPUT AREA --------
        self.output = tk.Text(root, height=18, width=72)
        self.output.pack(pady=10)

        self.show_initial_details()

    # -------- INITIAL DETAILS --------
    def show_initial_details(self):
        self.output.insert(tk.END, "APPLICATION STARTED\n")
        self.output.insert(tk.END, "-" * 60 + "\n")
        self.output.insert(tk.END, "Constructor Executed:\n")
        self.output.insert(tk.END, "• Message initialized as: 'Hello
Blockchain'\n\n")

        self.output.insert(tk.END, "State Variable:\n")
        self.output.insert(tk.END, "• message (Data Type: string)\n\n")

        self.output.insert(tk.END, "Functions Available:\n")
        self.output.insert(tk.END, "• set_message() → Store / Update message\n")
        self.output.insert(tk.END, "• get_message() → Retrieve message\n\n")

    # -------- SET MESSAGE FUNCTION --------
    def set_message(self):
        new_message = self.message_entry.get()
        if new_message == "":
            messagebox.showwarning("Warning", "Message cannot be empty")
        else:
            self.message = new_message
            self.output.insert(tk.END, "-" * 60 + "\n")
            self.output.insert(tk.END, "Function Called: set_message()\n")
            self.output.insert(tk.END, "Action: Message Stored / Updated\n")
            self.output.insert(tk.END, f"New Stored Message: {self.message}\n\n")
            self.message_entry.delete(0, tk.END)

    # -------- GET MESSAGE FUNCTION --------
    def get_message(self):
        self.output.insert(tk.END, "-" * 60 + "\n")
```



```python
        self.output.insert(tk.END, "Function Called: get_message()\n")
        self.output.insert(tk.END, "Action: Retrieved Stored Message\n")
        self.output.insert(tk.END, f"Stored Message: {self.message}\n\n")

    # -------- REFRESH / CLEAR HISTORY FUNCTION --------
    def refresh_history(self):
        self.output.delete(1.0, tk.END)   # Clear all text
        self.show_initial_details()       # Re-show initial details
        messagebox.showinfo("Refreshed", "History cleared successfully")

# -------- MAIN PROGRAM --------
if __name__ == "__main__":
    root = tk.Tk()
    app = SimpleStorageGUI(root)
    root.mainloop()
```



```
State Variable:
• message (Data Type: string)

Functions Available:
• set_message() → Store / Update message
• get_message() → Retrieve message

------------------------------------------------------------
Function Called: set_message()
Action: Message Stored / Updated
New Stored Message: hello

------------------------------------------------------------
Function Called: get_message()
Action: Retrieved Stored Message
Stored Message: hello
```