

```

import tkinter as tk
from tkinter import messagebox, scrolledtext
import subprocess
import os

class TruffleManager:
    def __init__(self, root):
        self.root = root
        self.root.title("Portfolio Smart Contract Manager")
        self.root.geometry("800x600")
        self.root.configure(bg="#1e1e1e")

        # Title
        tk.Label(root, text="Truffle Smart Contract Dashboard", fg="#ecf0f1",
                 bg="#1e1e1e", font=("Arial", 18, "bold")).pack(pady=15)

        # Control Buttons
        btn_frame = tk.Frame(root, bg="#1e1e1e")
        btn_frame.pack(pady=10)

        self.btn_compile = tk.Button(btn_frame, text="1. Compile", width=15, height=2,
                                     bg="#3498db", fg="white", font=("Arial", 10, "bold"),
                                     command=lambda: self.execute(["truffle", "compile", "--all"]))
        self.btn_compile.grid(row=0, column=0, padx=10)

        self.btn_test = tk.Button(btn_frame, text="2. Run Tests", width=15, height=2,
                                 bg="#2ecc71", fg="white", font=("Arial", 10, "bold"),
                                 command=lambda: self.execute(["truffle", "test", "--compile-all"]))
        self.btn_test.grid(row=0, column=1, padx=10)

        # Terminal Output
        tk.Label(root, text="Process Output:", fg="#bdcc7c", bg="#1e1e1e").pack(padx=25, anchor="w")
        self.terminal = scrolledtext.ScrolledText(root, width=90, height=20, bg="black", fg="#00ff00", font=("Courier New", 10))
        self.terminal.pack(pady=10, padx=20)

    def execute(self, command):
        self.terminal.insert(tk.END, f"\n> Starting: {' '.join(command)}\n")
        self.root.update_idletasks()

        try:
            # shell=True is vital for Windows to find truffle.cmd
            proc = subprocess.run(command, capture_output=True, text=True, shell=True)

            if proc.stdout:
                self.terminal.insert(tk.END, proc.stdout)
            if proc.stderr:
                # We show warnings (like uWS) but don't treat them as fatal
                self.terminal.insert(tk.END, f"\nSystem Warnings:\n{proc.stderr}")

            if "1 passing" in proc.stdout:
                messagebox.showinfo("Success", "Unit Test Passed Successfully!")

        except Exception as e:
            self.terminal.insert(tk.END, f"ERROR: {str(e)}")
            messagebox.showerror("Error", "Command failed to execute.")

            if "1 passing" in proc.stdout:
                messagebox.showinfo("Success", "Unit Test Passed Successfully!")

        except Exception as e:
            self.terminal.insert(tk.END, f"ERROR: {str(e)}")
            messagebox.showerror("Error", "Command failed to execute.")

        self.terminal.see(tk.END)

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

```

Output:

The screenshot shows the Truffle Smart Contract Dashboard interface. On the left, there's a sidebar with various icons and a list of recent files: '004 BE Practical-3.docx' and '004 BLOCK CHAIN E...'. The main area is titled 'Truffle Smart Contract Dashboard' with two buttons: '1. Compile' (blue) and '2. Run Tests' (green). Below these buttons is a 'Process Output' section containing terminal logs. A modal window titled 'Success' is displayed in the center, stating 'Unit Test Passed Successfully!' with an 'OK' button. To the right of the modal, there's a table showing account details: Address, Balance, Tx Count, and Index. The first account has address 533BCd85179f76, balance 100.00 ETH, tx count 0, and index 0. Other accounts listed have similar details.

This screenshot shows the same Truffle Smart Contract Dashboard interface. The 'Process Output' section displays the terminal logs for the 'truffle compile --all' and 'truffle test --compile-all' commands. The logs show the compilation of the 'Portfolio.sol' contract and the execution of tests, both of which were successful. The rest of the interface is identical to the first screenshot, including the sidebar and the table of account balances.