**ACADEMIC TASK - 2 CSE316**

(OPERATING SYSTEMS)



**TOPIC: “Real-Time Memory Allocation Tracker”**

# PROJECT REPORT

By

**Reg. No. Name**

12302841 Deepak Chaudhary

12307484 Prakhar Singh

**TO**

Amandeep Kaur(UID: 31019)

## 1. Project Overview

The "Real-Time Memory Allocation Tracker" is a tool that shows how memory is used and freed up as it happens. It focuses on simple memory tricks like paging and segmentation. The tool pretends to have a fixed amount of memory, letting you take or give back pieces of it using easy methods like best-fit and worst-fit. It also draws a picture of the memory and shows some basic computer stats. The goal is to teach you about memory stuff, show different ways to handle it, and let you try it out yourself. You’ll get a desktop app that updates live, gives clear numbers, and makes memory easy to see. It’s a basic Python program that runs on one thread, but you can add more features like paging later if you want.

## 2. Module-Wise Breakdown

The project is divided into three modules:

* Memory Management Core:
  + Purpose: Manages the logic for memory allocation, deallocation, and algorithm evaluation. o Role: Serves as the backend, maintaining the memory state and providing data for visualization.
* Graphical User Interface (GUI):
  + Purpose: Offers an interactive interface for users to control and observe memory operations. o Role: Displays memory blocks, system stats, and user controls, connecting the core logic to the user.
* System Monitoring and Visualization:
  + Purpose: Tracks system performance (CPU, RAM) and visualizes memory allocation graphically. o Role: Provides context with real-world system data and a dynamic visual representation of memory.

## 3. Functionalities

* Memory Management Core:
  + Allocate memory blocks based on size and algorithm (e.g., allocate 50 units with best-fit).
  + Deallocate memory by ID, merging free blocks (e.g., free ID 3).
  + Evaluate best-fit vs. worst-fit algorithms (e.g., 20 random operations, reporting fragmentation).
  + Track memory stats (e.g., "Total: 1024, Used: 300, Free: 724").
* Graphical User Interface (GUI):
  + Manual allocation/deallocation controls (e.g., enter "50" and click "Allocate").
  + Algorithm selection via radio buttons (e.g., switch to "Worst-Fit"). o Automated testing feature (e.g., simulate 10 random operations).
  + Error feedback (e.g., "Not enough memory" popup).
* System Monitoring and Visualization:
  + Display CPU/RAM usage (e.g., "CPU: 45%, RAM: 4 GB / 8 GB"). o List top 10 processes by memory usage (e.g., "chrome.exe: 512 MB").
  + Visualize memory blocks (e.g., red for allocated, green for free).
  + Real-time updates (e.g., canvas refreshes after allocation).

## 4. Technology Used

* Programming Languages:
  + Python: Chosen for its simplicity, extensive library support, and alignment with the provided code.
* Libraries and Tools:
  + Tkinter: Used for building the GUI, providing a lightweight and native desktop interface. o psutil: Monitors system resources (CPU, RAM, processes) with cross-platform compatibility. o random: Generates random sizes and operations for automated testing.
  + time: Manages timing for real-time updates and delays.
* Other Tools:
  + [GitHub for version control]: Recommended for tracking revisions (placeholder until repository is created). o [PyCharm/VS Code]: Suggested IDEs for development and debugging.

## 5. Flow Diagram



## 6. . Revision Tracking on GitHub

* Repository Name**:** [Real-Time-Memory-Allocation]
* GitHub Link**:** [**https://github.com/DeepakCdry/Real-Time-Memory-Allocation-OS-Project**](https://github.com/DeepakCdry/Real-Time-Memory-Allocation-OS-Project)

## 7. Conclusion and Future Scope

The project successfully delivers a functional tool for visualizing memory allocation in real-time, with support for best-fit and worst-fit algorithms, system monitoring, and an interactive GUI. It provides a solid foundation for understanding memory management concepts. However, it currently lacks full paging and segmentation implementations, which could be added for a more comprehensive demonstration. Future enhancements could include:

* Implementing paging with fixed-size pages and a page table visualization.
* Adding segmentation with variable-size segments and metadata display.
* Supporting multiple memory allocation strategies (e.g., first-fit, next-fit).
* Shifting it to a web-based version for easier access by more users.

* Integrating real-time memory leak detection or performance benchmarking.

8. References

* Python Documentation: <https://docs.python.org/3/>
* Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>
* psutil Documentation: <https://psutil.readthedocs.io/en/latest/>
* Operating System Concepts (Silberschatz, Galvin, Gagne) – For memory management theory.

**Appendix**

## A. AI-Generated Project Elaboration/Breakdown Report

**Project Overview**: Goals include visualizing memory allocation in real-time, demonstrating paging/segmentation, and providing an interactive tool. Expected outcomes are a graphical desktop app with real-time updates and stats. Scope is a single-threaded Python app with a fixed memory size.

**Module-Wise Breakdown**:

* Memory Management Core: Backend for allocation/deallocation logic.
* GUI: Interface for user interaction and control.
* System Monitoring and Visualization: Tracks system stats and visualizes memory.

**Functionalities**:

* Core: Allocate/free memory, evaluate algorithms, track stats.
* GUI: Manual controls, algorithm selection, auto-testing, feedback.
* Visualization: System stats, process list, memory block display, real-time updates.

**Technology Recommendations**: Python with tkinter, psutil, random, time.

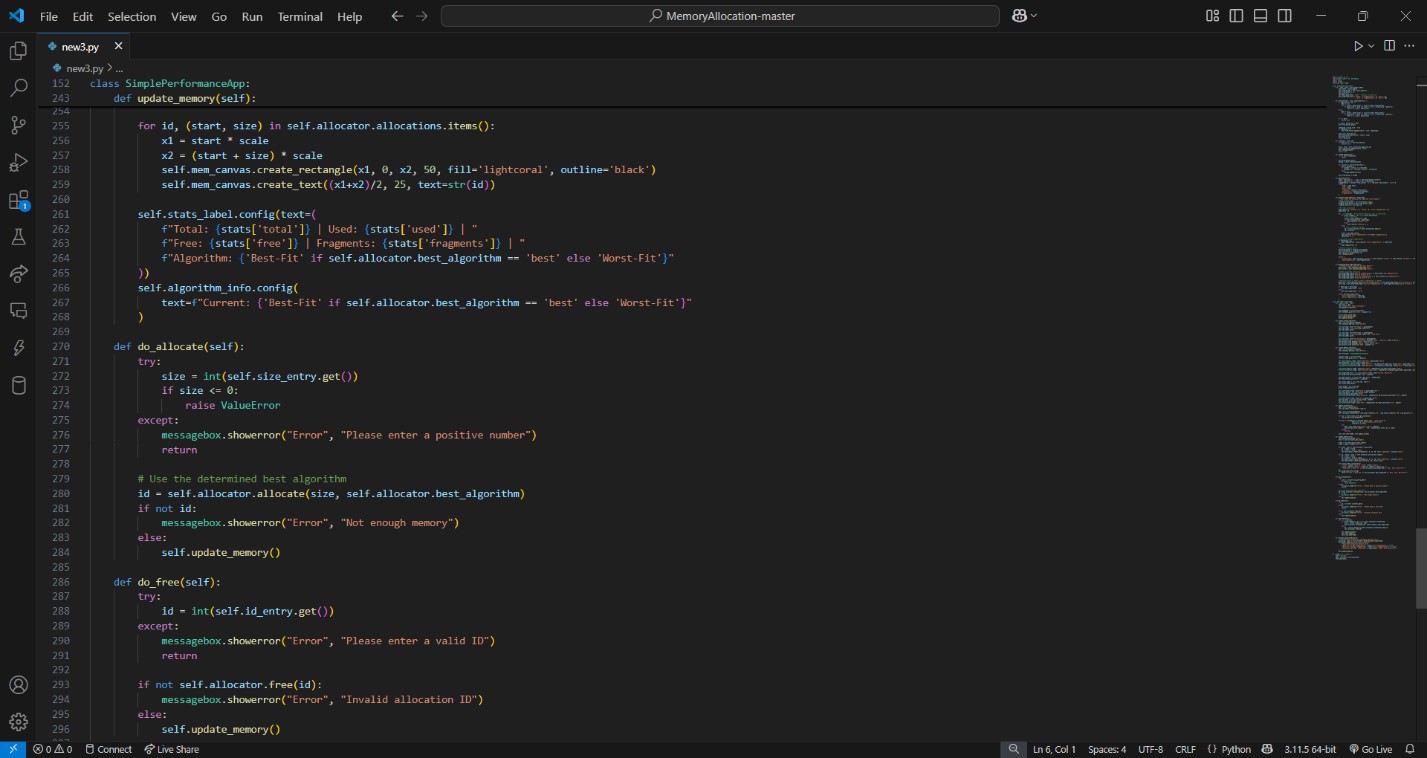
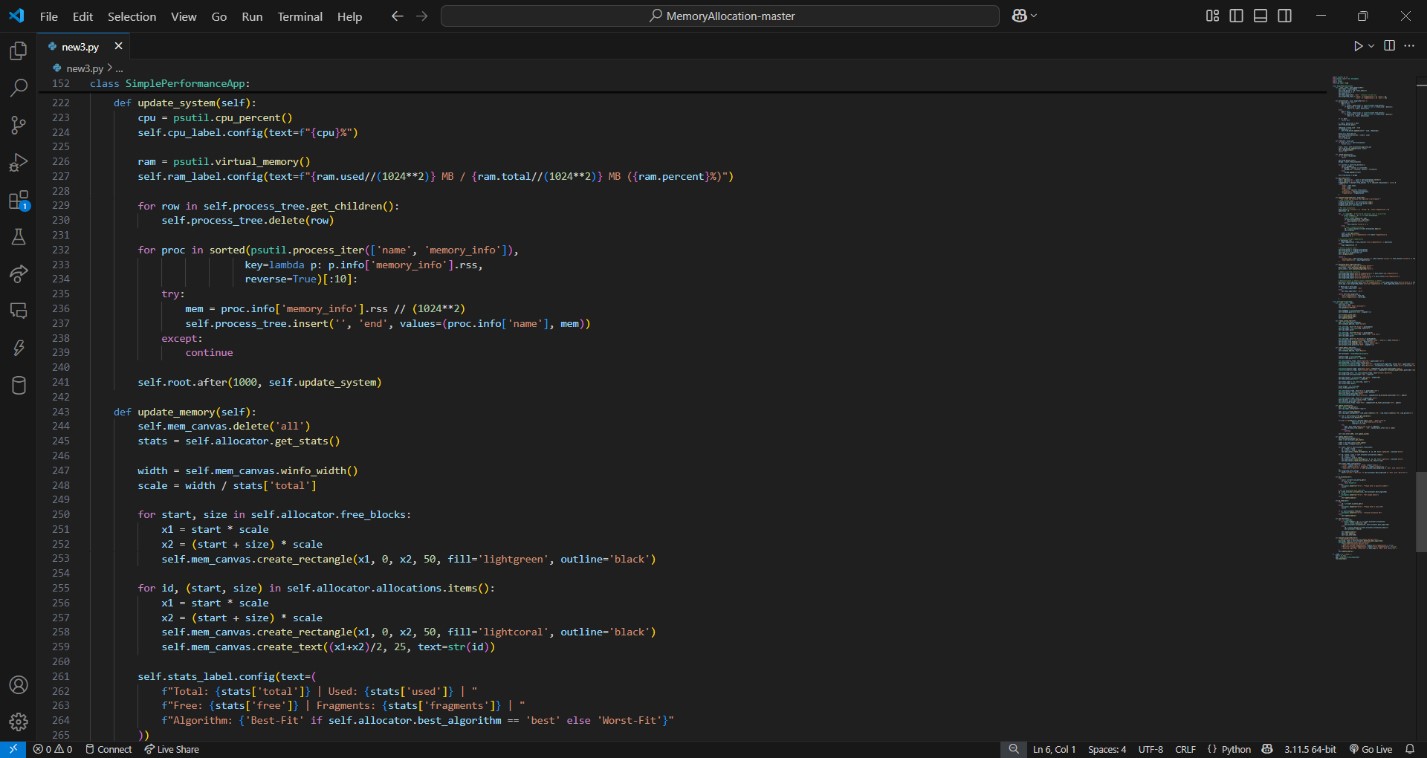
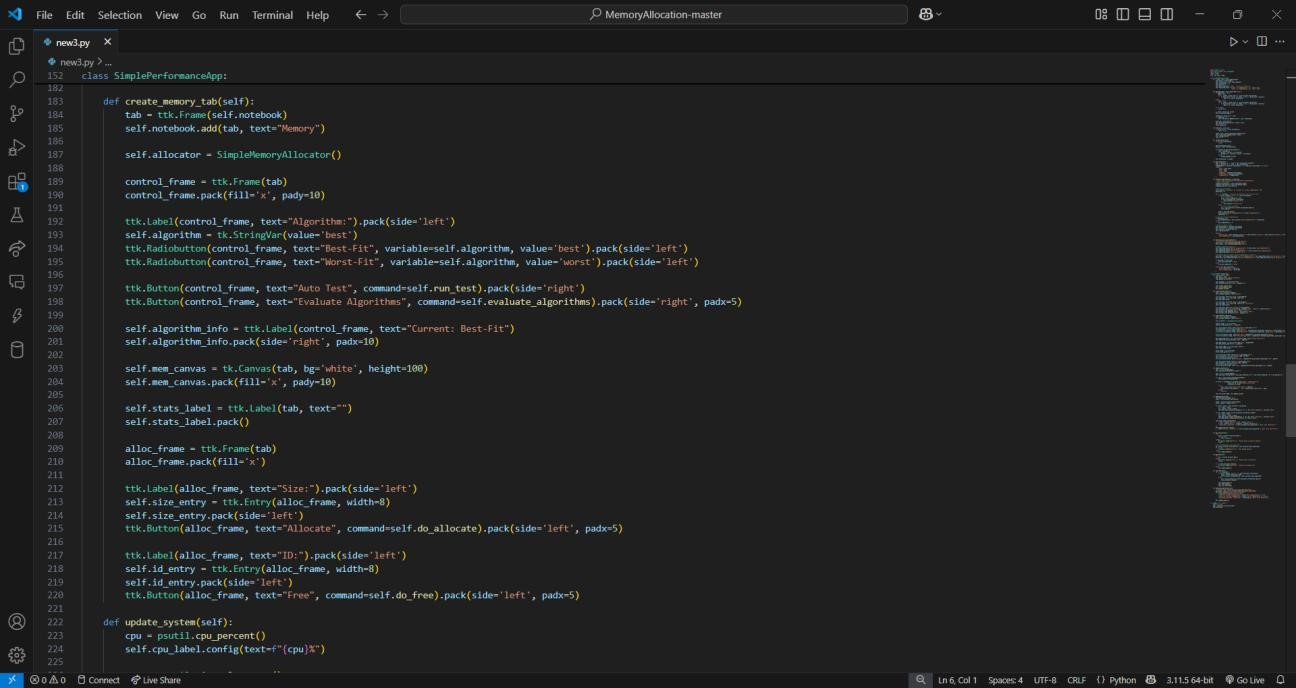
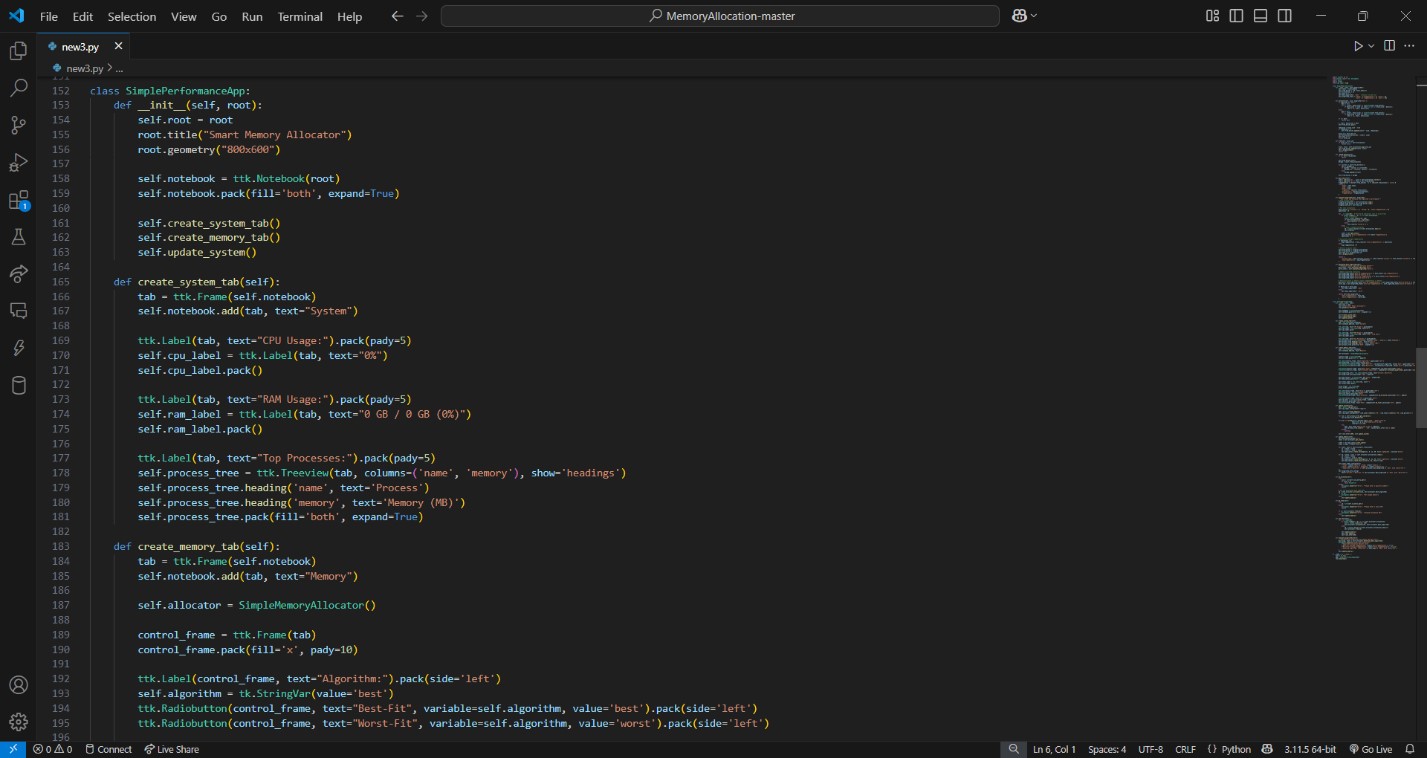
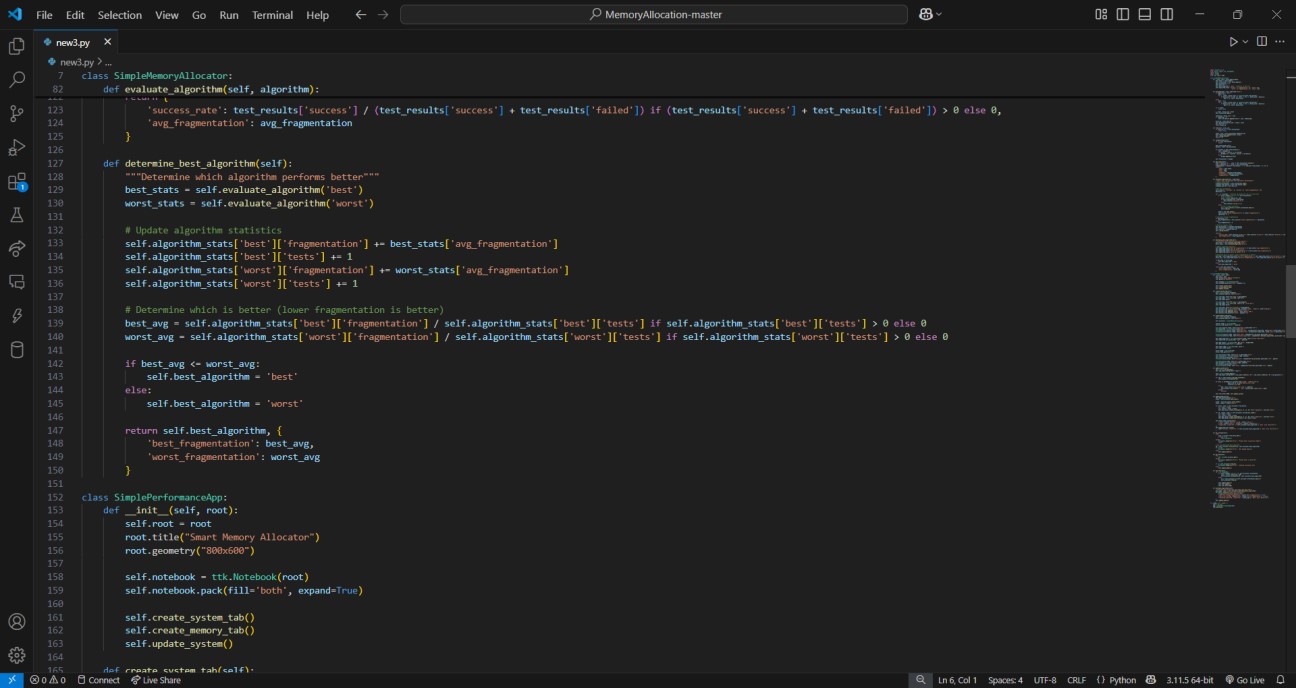
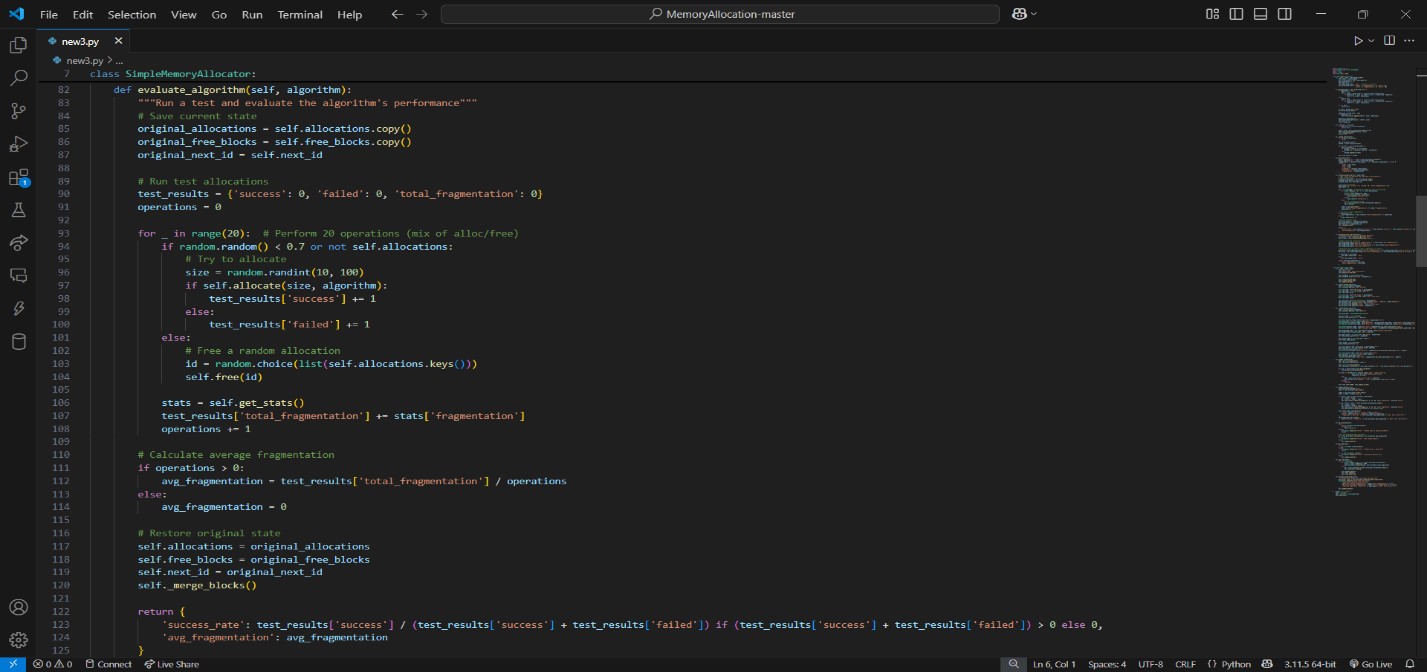
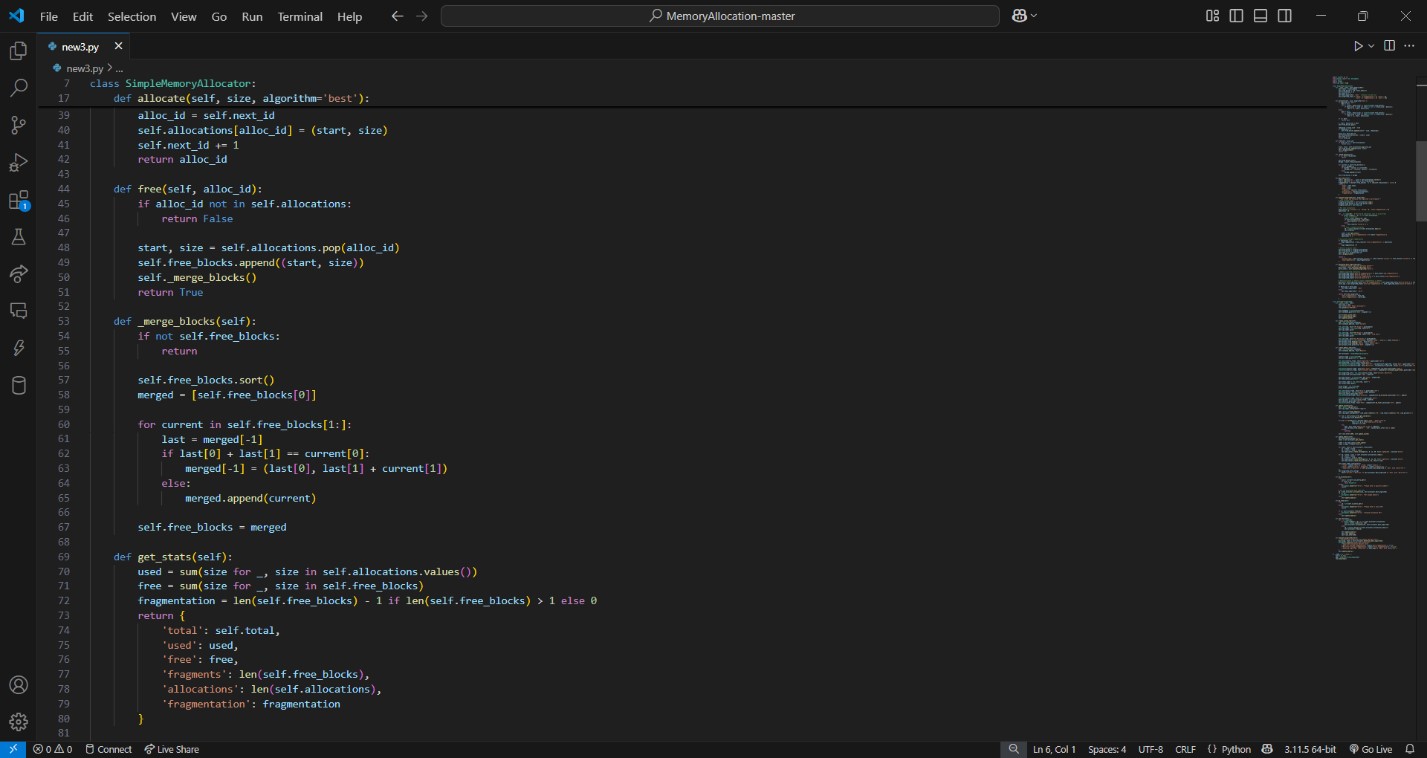
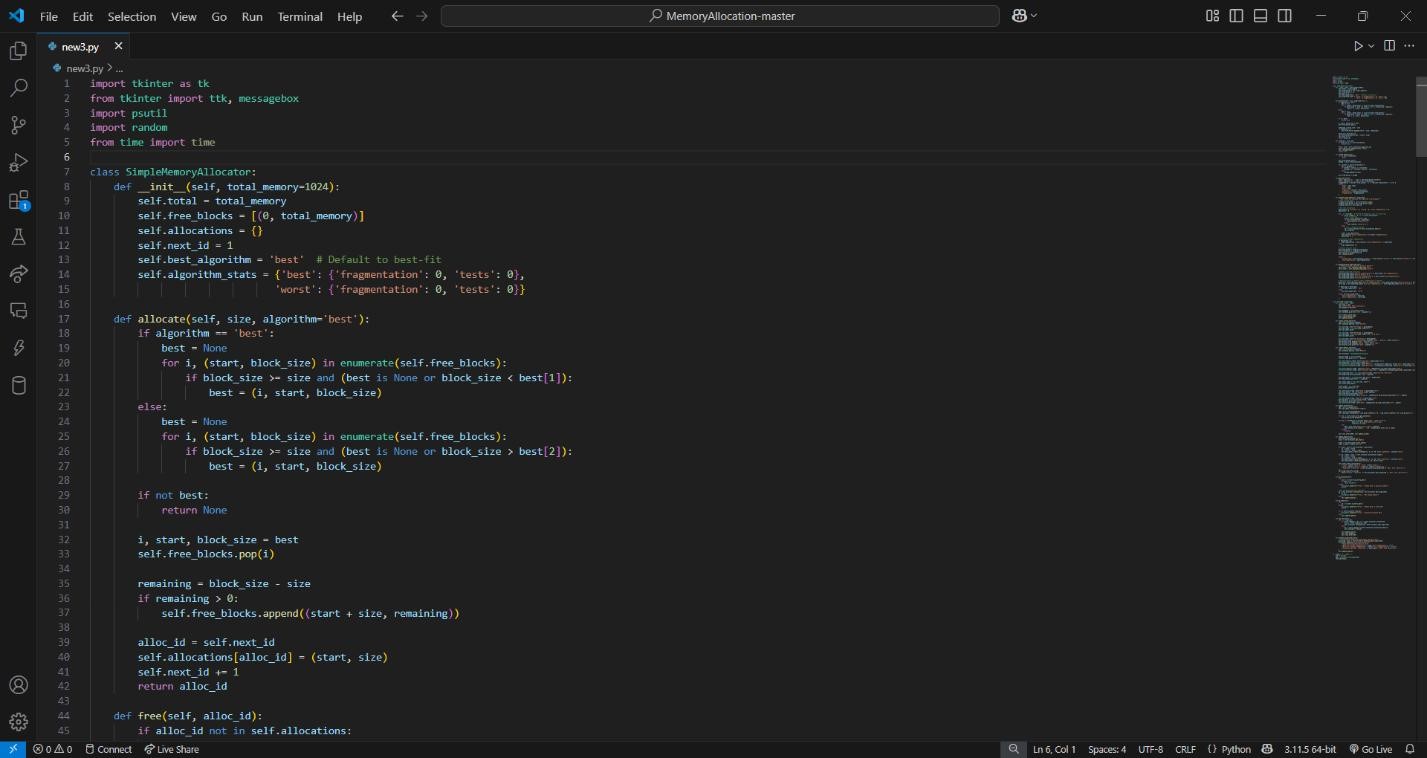
**Execution Plan:** Setup environment, enhance core, refine GUI, integrate visualization, test/debug, finalize.

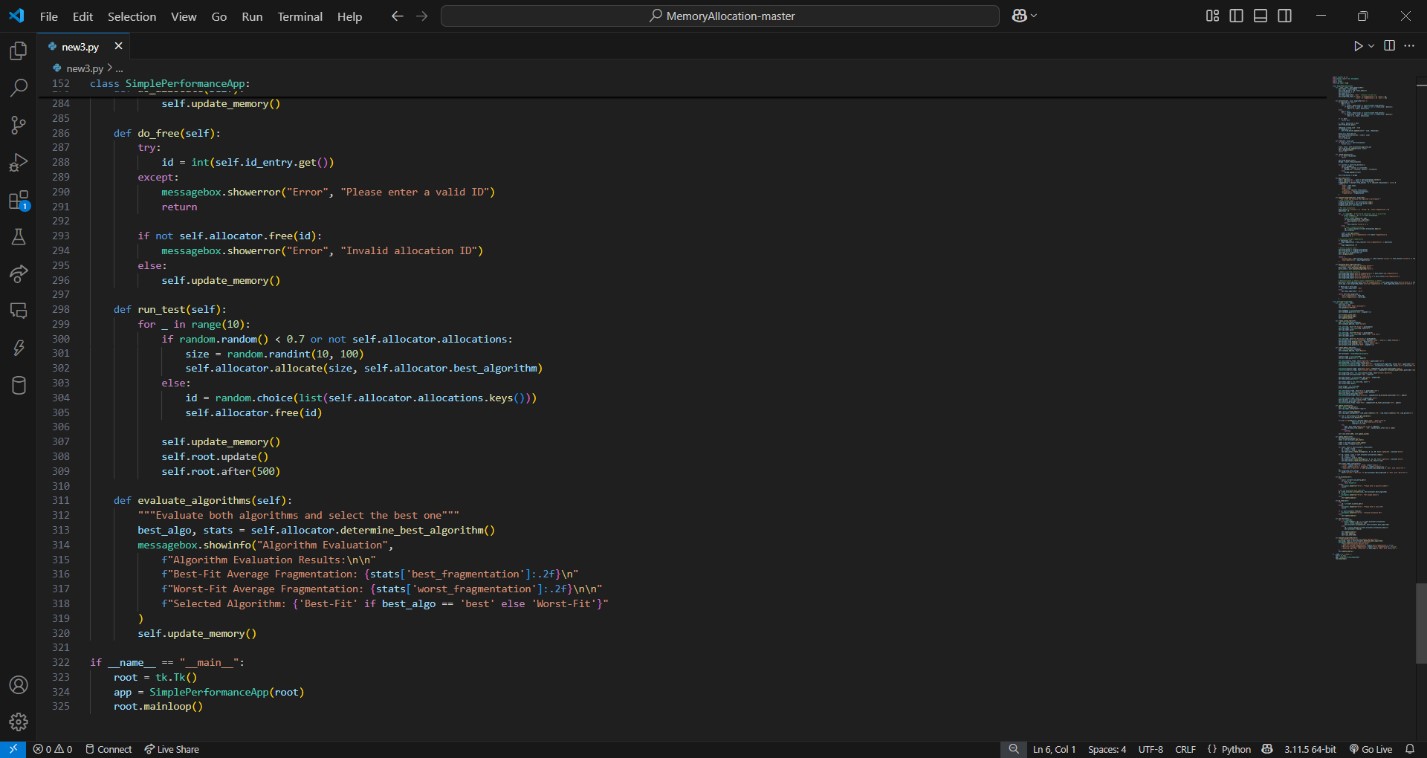
## B. Problem Statement

"Real-Time Memory Allocation Tracker

Description: Build a tool that visualizes memory allocation in real-time, showcasing paging and segmentation."

## C. Solution/Code





## D. Real-time Results UI

