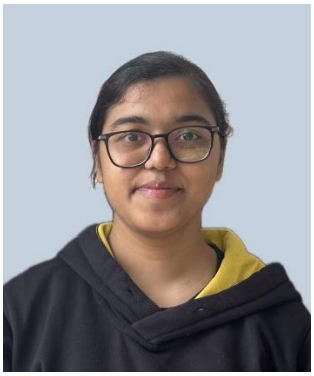
PROJECT AND TEAM INFORMATION

Project Title

File System Simulator: A Visual Learning Tool

Student / Team Information

Team Name: Team #	**SE(OS)-VI-T137** **SE(OS)-VI-T137**
Team member 1 (Team Lead) Name: Khushi Kumari Jha Student Id: *-220221015* E-mail : Khushijhakj2311@gmail.com	
Team member 2 Name: Divyanshi Rasotia Student Id: *220221037* E-mail : ddivyanshi693@gmail.com	

Team member 3 Name: Shambhavee Shukla Student Id: - 220222042 E-mail : Shambhaveeshukla11@gmail.com	
Team member 4 Name: Bhawna Bisht Student Id: - 220121655 E-mail : bebhawna3@gmail.com	

PROPOSAL DESCRIPTION (10 pts)

Motivation (1 pt)

- Understanding file and memory management in Operating Systems is often theoretical and abstract.
- Most learning methods lack visual or interactive elements, making concepts harder to grasp.
- A GUI-based simulator can make complex OS processes intuitive and engaging.
- Visual interaction helps students build a stronger conceptual foundation through hands-on experience.
- This project bridges the gap between theory and practice in OS education.

State of the Art / Current solution (1 pt)

- Modern OS file systems use advanced storage and paging mechanisms but lack customized educational simulations.
- Simulations help in understanding file allocation and memory management without altering real OS kernels.
- Researchers like Hoare (1973) laid the foundation for paging systems used today.
- Studies by Thekkath et al. (1994) and Gupta & Verma (2015) highlight the value of simulating storage and allocation strategies.
- Fiat et al. (1991) show how paging algorithms affect access performance and disk use.
- GUI tools like PyQt are effective for building interactive file system simulators, as supported by books from Willman and Summerfield.

Project Goals and Milestones (2 pts)

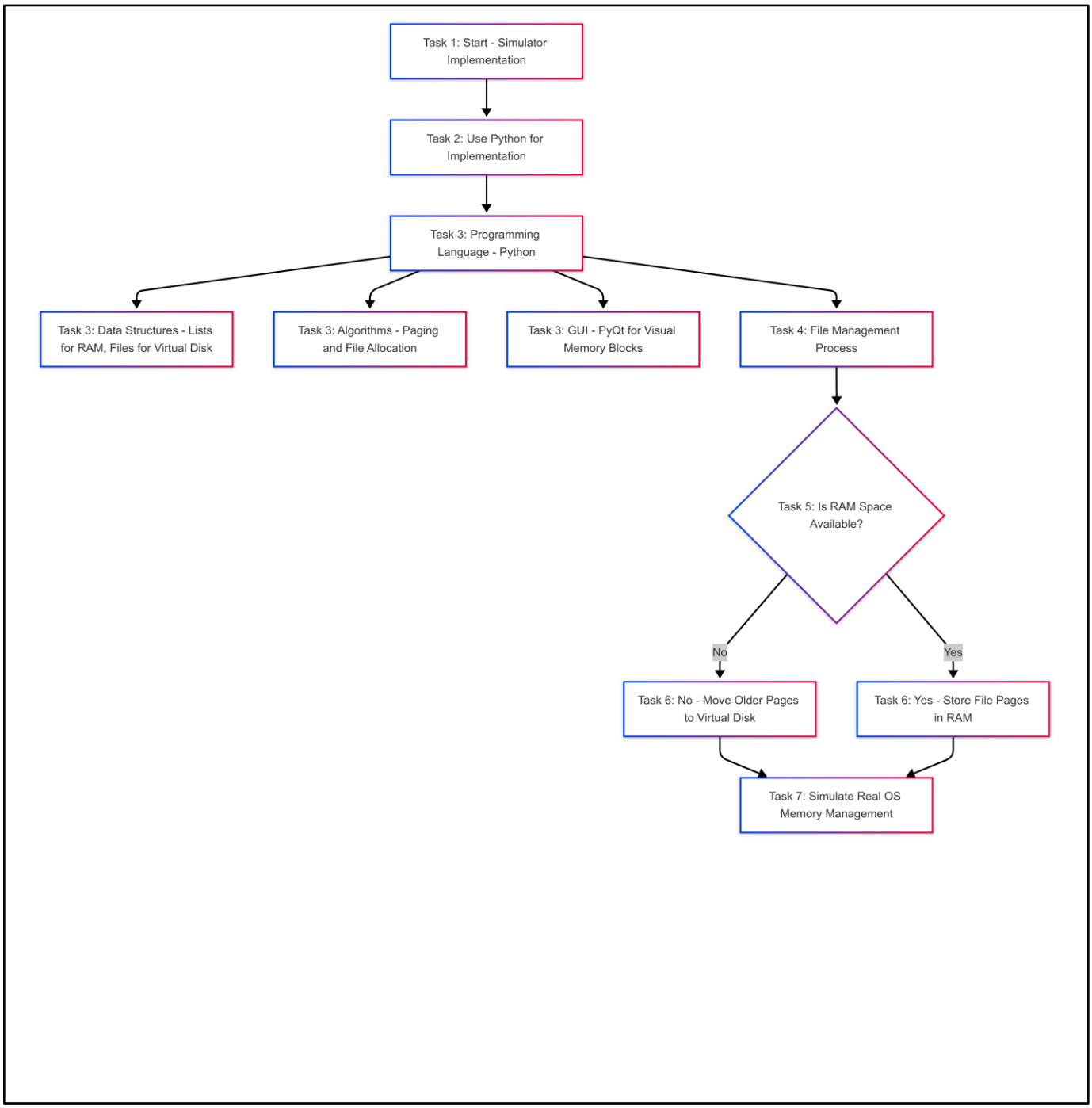
Goals:

- Develop a File System Simulator that visually represents memory allocation, paging, and swapping.
- Implement file storage techniques.
- Simulate paging mechanisms.
- Provide a GUI interface for user interaction.
- Offer real-time visualization of memory usage.

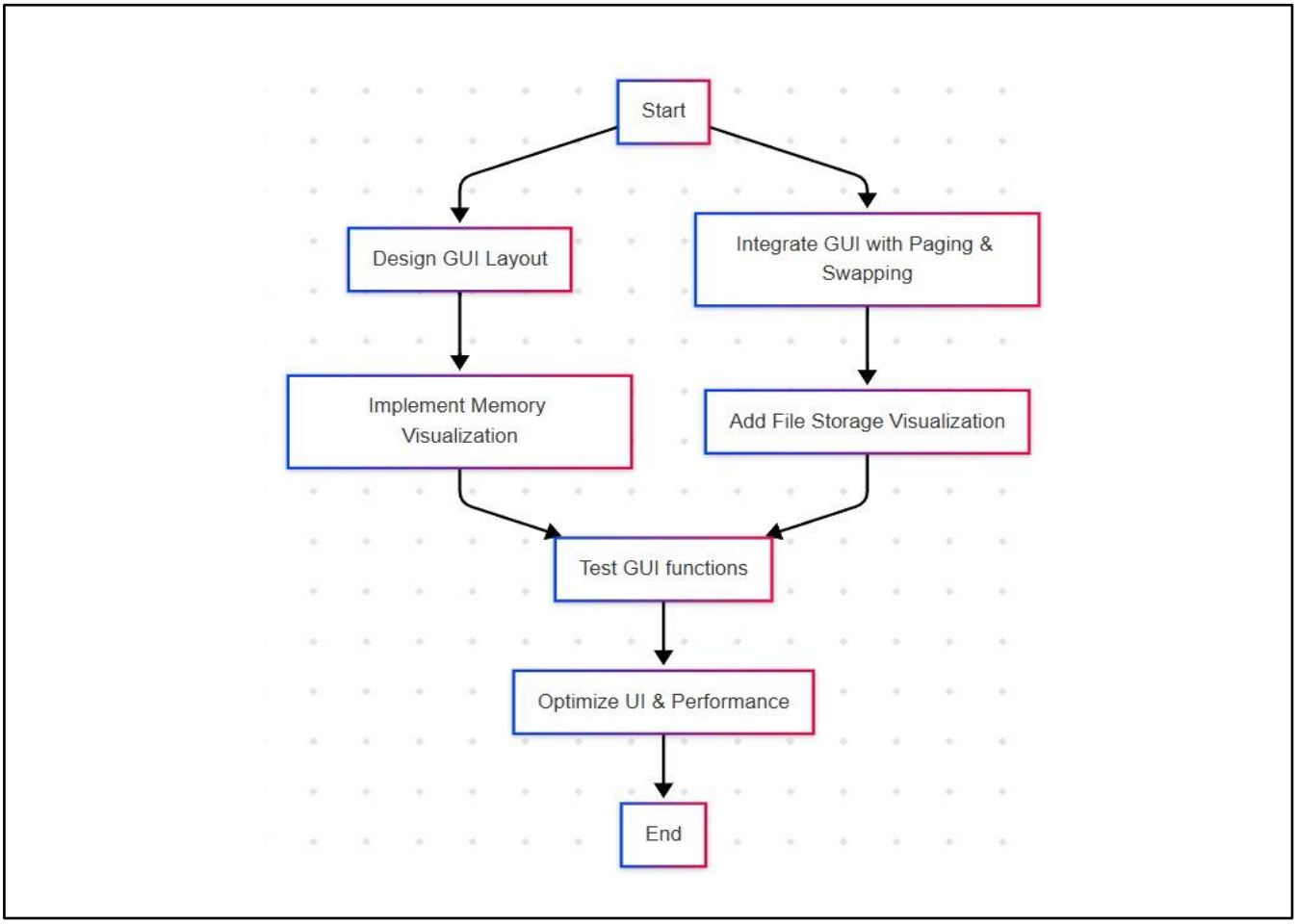
Milestones:

- Design system architecture and define data structures.
- Implement memory management (RAM simulation, paging).
- Develop file allocation methods and virtual disk storage.
- Implement swapping techniques.
- Implement page replacement algorithms.
- Build a user interface (GUI), testing, and optimization.

Project Approach (3 pts)



System Architecture (High Level Diagram)(2 pts)



Project Outcome / Deliverables (1 pts)

- A fully functional File System Simulator capable of demonstrating memory paging and swapping.
- A graphical user interface (GUI) to visually display memory allocation.
- A report explaining how file storage and memory management work.
- Source code and documentation for future improvements.

Assumptions

The simulator will not interact with the real OS file system; all operations are virtual.

The RAM size is fixed (e.g., 4-page frames), and paging will be triggered when RAM is full.

Users will operate the system via a GUI interface.

The project will be built using Python and standard file-handling techniques.

References

[1].Thekkath, C. A., Wilkes, J., & Lazowska, E. D. (1994). Techniques for file system simulation. Software: Practice and Experience, 24(11), 981-999.

[2].Gupta, P., & Verma, R. (2015). File System Simulation.

[3].Fiat, A., Karp, R. M., Luby, M., McGeoch, L. A., Sleator, D. D., & Young, N. E. (1991). Competitive paging algorithms. Journal of Algorithms, 12(4), 685-699.

[4].Hoare, C. A. R. (1973). A structured paging system. The Computer Journal, 16(3), 209-215.

[5].Willman, J. M. (2020). Beginning PyQt. Berkeley, CA: Apress.

[6].Summerfield, M. (2007). Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming. Pearson Education.