PROJECT AND TEAM INFORMATION

Project Title

TimeVaultFS – A File System with Rollback and Snapshots

Student / Team Information

Team Name: ***TeamSync***	
Team member 1 (Team Lead) ***Name - Anmol Bisht*** ***University Roll no. - 2318395*** ***Student ID - 23011370*** ***Mail - anmolbisht432@gmail.com***	
Team member 2 ***Name - Akshat Kumar*** ***University Roll no. - 2318274*** ***Student ID - 23011372*** ***Mail – akshat8066@gmail.com***	
Team member 3 ***Name - Pratishtha Goyal*** ***University Roll no. - 2319267*** ***Student ID - 23012745*** ***Mail – pratishthagoyal001@gmail.com***	

PROPOSAL DESCRIPTION (10 pts)

Motivation (1 pt)

Modern file systems are prone to accidental deletions, overwrites, or corruption caused by unexpected crashes. Users often lose valuable data due to the absence of simple rollback mechanisms. While enterprise-grade solutions like version control systems or professional backup tools exist, they are either too complex or resource-heavy for smaller systems.
Our motivation is to design a lightweight file system that provides ***rollback to previous states****,* ***crash recovery****, and* ***manual snapshots****, while still supporting essential file operations. This project will not only demonstrate practical OS concepts like file handling, journaling, and recovery but also present a solution that is simple, user-friendly, and educational.*

State of the Art / Current solution (1 pt)

Current operating systems implement file system reliability using ***journaling*** *or* ***backup tools (e.g. OneDrive)****. However, these solutions are designed for large-scale production environments and often lack transparency for learning purposes.*
Version control systems like Git offer rollback but are specialized for text/code files, not general file management. Our solution aims to bridge this gap by offering ***a simplified, educational, rollback-enabled file system*** *that can be demonstrated in academic and lightweight environments.*

Project Goals and Milestones (2 pts)

- *Develop a functional file system supporting core operations (create, write, read, delete).*
- *Implement rollback using journaling logs.*
- *Provide snapshot functionality for manual state saving and restoring.*
- *Ensure robustness against accidental deletions or overwrites.*

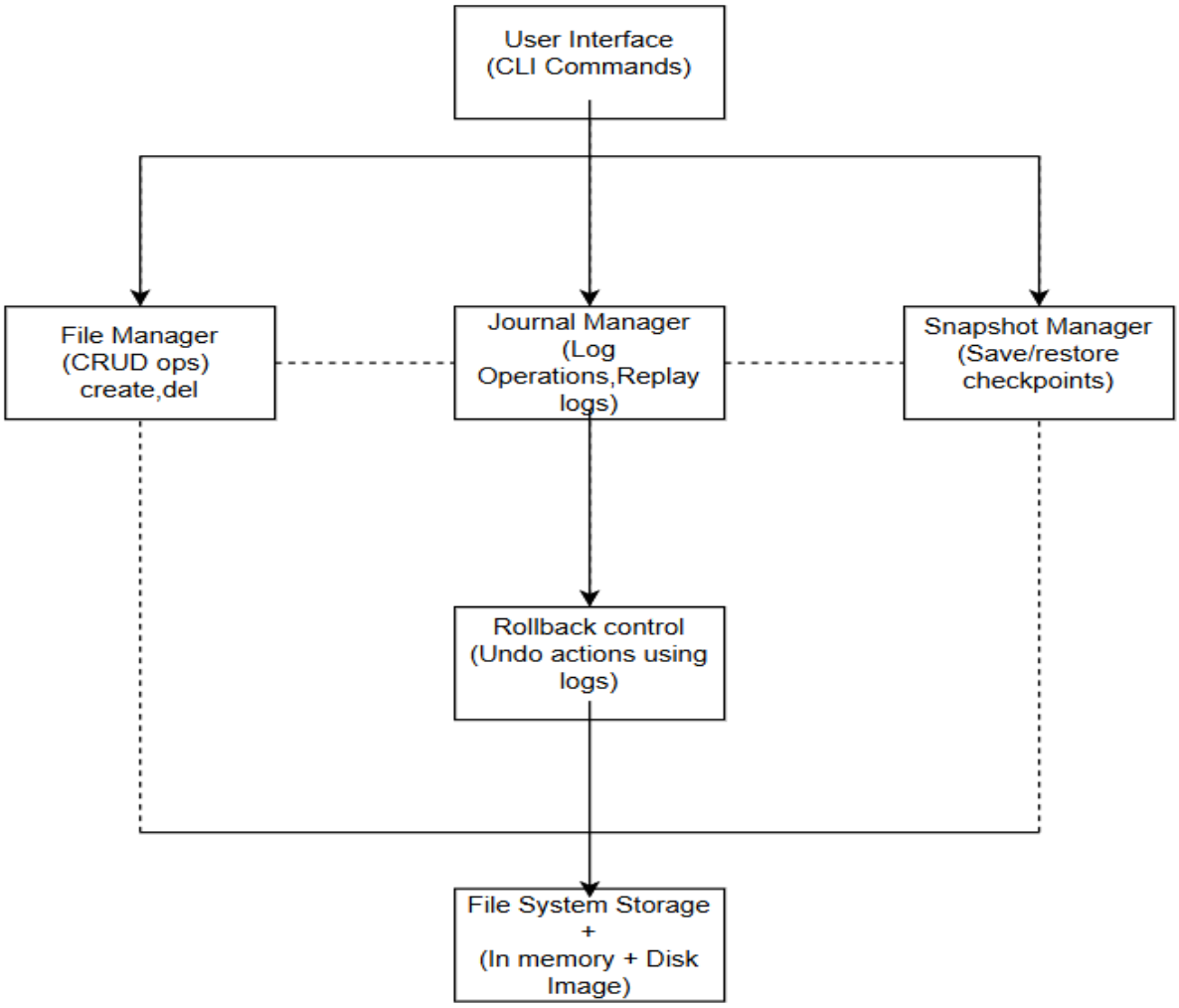
Project Approach (3 pts)

1. ***File Manager:*** *Handles creation, deletion, and modification of files and directories.*
2. ***Journal Manager:*** *Logs every operation for recovery and rollback.*
3. ***Rollback Controller:*** *Enables undoing actions and restoring to consistent states.*
4. ***Snapshot Manager:*** *Allows users to save and restore system states manually.*

Technologies:

- ***Language:*** *C (for low-level OS) or Python (for faster prototyping).*
- ***Interface:*** *Command Line Shell (supporting commands like create, write, rollback, snapshot).*
- ***Data Storage:*** *In-memory structures with optional persistent storage via files.*

System Architecture (High Level Diagram)



Project Outcome / Deliverables (1 pts)

*The project will deliver a **functional custom file system prototype** with:*

- *Support for core file operations.*
- *Journaling and rollback to previous states.*
- *Manual snapshot creation and restoration.*
- *A CLI interface for demonstration.*

Additionally, the project report will document the architecture, design decisions, and implementation challenges, providing both theoretical and practical value

Assumptions

- *The system will be designed for educational and prototype purposes, not large-scale deployment.*
- *Users interact via command line interface only.*
- *Memory and storage constraints are simulated, not hardware-bound.*

References

- ***GeeksforGeeks – File System Implementation***
  https://www.geeksforgeeks.org/file-system-implementation/

- ***What is a File System?***
  https://www.youtube.com/watch?v=KN8YgJnShPM