RV College of Engineering®

Go, change the world

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DSA LAB EL REPORT

IS233AI

Submitted by

Yashvanth M U	**1RV23IS141**

Title:

File management simualtion

Under the guidance of

Rekha B S
Assistant Prof.
Dept. of ISE
RV College of Engineering®

2024-2025

1. Introduction:

The project, "File Management Simulation," is an interactive and educational application that simulates the core functionalities of a file system. This simulation is designed to provide users with a better understanding of how modern file systems manage data through hierarchical structures. By enabling users to create, delete, traverse, and visualize directories and files, the application serves as a hands-on learning tool for understanding file management concepts.

The application incorporates version control to ensure changes in the file system can be tracked and reverted if needed. It also features a real-time graphical representation of the directory tree, enhancing the user's comprehension of directory hierarchies and relationships. Built using Python's Tkinter library, the application combines simplicity, functionality, and interactivity to deliver a robust educational experience.

This report outlines the design, implementation, and applications of the "File Management Simulation" project, emphasizing its role as a foundational tool for learning file system operations.

2. Problem Statement:

Efficient file management is critical in both educational and professional environments to ensure data is well-organized, accessible, and secure. Many individuals struggle to grasp file system principles due to the abstract nature of these concepts. The project addresses this gap by providing a simulated environment to:

- Explore file system operations interactively.
- Learn how hierarchical structures are created and managed.
- Understand the importance of version control in tracking and restoring changes.
- Visualize directory and file relationships through a dynamic graphical interface.

This tool is particularly useful for students and developers seeking to strengthen their understanding of file system operations in a simplified yet practical manner.

3. Objectives:

1. 1. **Design Hierarchical Structures:**
    - Implement a simulation of directories and files using a tree-based structure.
2. **Support Core Operations:**
    - Enable users to create, delete, and navigate directories and files.
3. **Version Control and Restoration:**
    - Provide a mechanism to save and restore previous versions of the file system.
4. **Interactive Visualization:**
    - Offer a graphical representation of the directory tree for intuitive understanding.
5. **Educational Tool:**
    - Facilitate hands-on learning for students and professionals exploring file systems.
6. **Usability and Efficiency:**
    - Build a responsive and user-friendly graphical interface for seamless interaction.

.

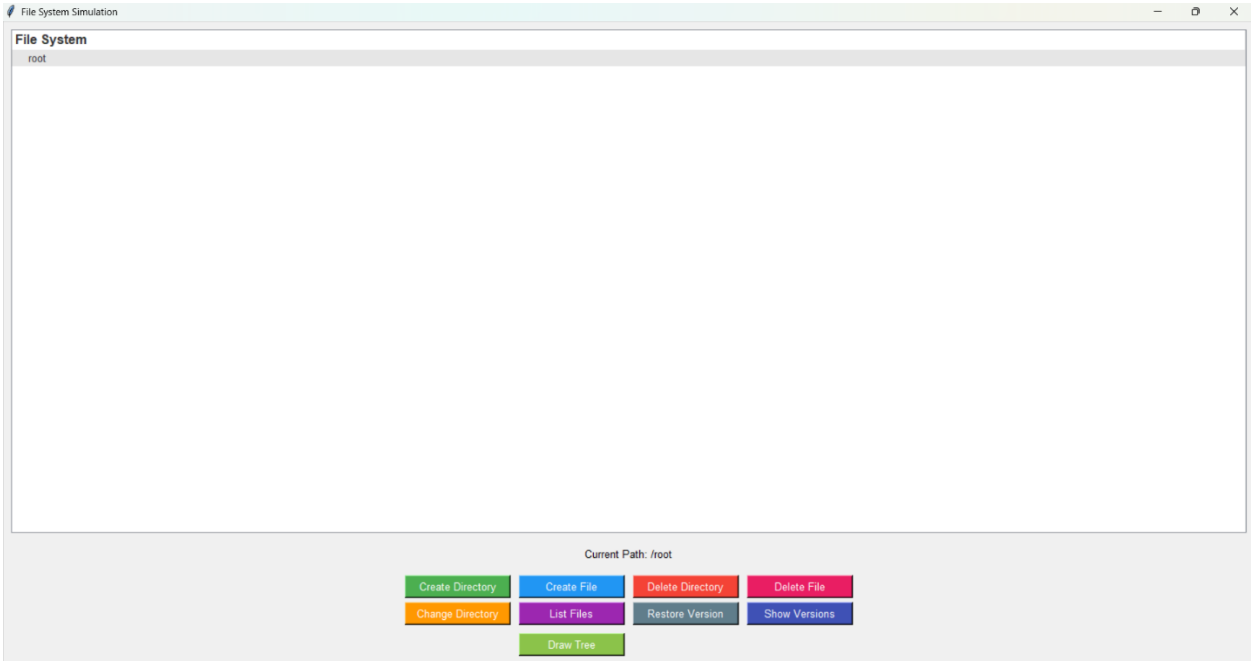
4. DSA Concepts Used:

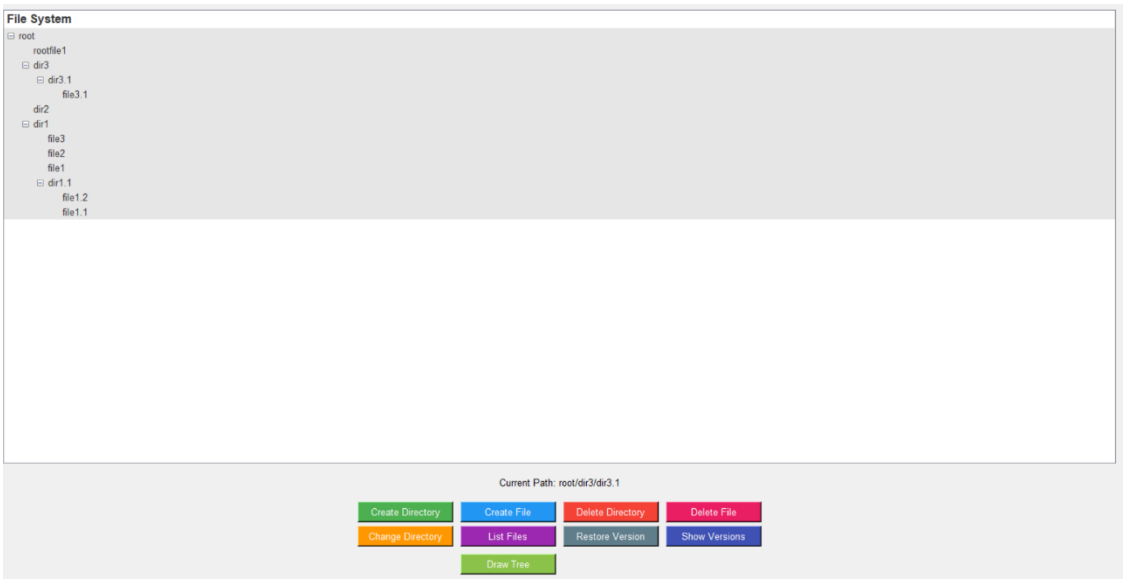
1. **Linked Lists:**
    - Used for managing files within a directory, allowing dynamic insertion and deletion.
2. **Tree Structures:**
    - Represent the hierarchical structure of directories, supporting efficient traversal and operations.
3. **Stacks:**
    - Utilized in version control for storing snapshots of the file system and supporting rollback operations.
4. **Recursive Algorithms:**
    - Enable directory traversal, tree cloning, and path generation for absolute paths.
5. **Dynamic Memory Management:**
    - Efficiently handle the creation and deletion of directories and files without static constraints.

5. Implementation Details:

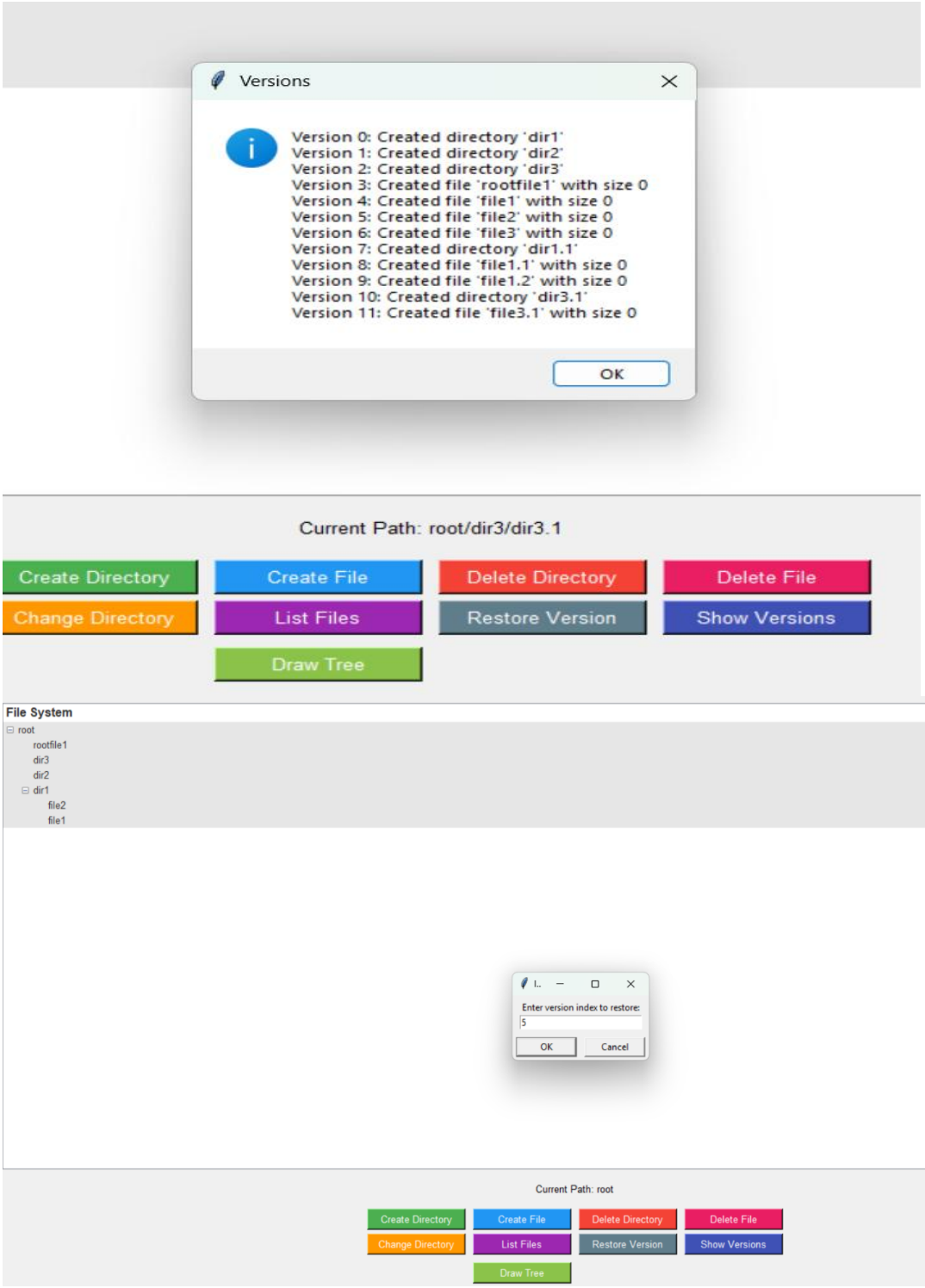
- **Directory and File Management:**
    - Directories and files are managed using custom data structures, ensuring a realistic simulation of hierarchical file systems.
    - Operations like creation, deletion, and navigation are implemented with efficiency and precision.
- **Version Control:**
    - Tracks every change in the file system, enabling users to restore previous states using a stack-based approach.
    - Each snapshot includes details of directory structure and file content.
- **Graphical Visualization:**
    - A real-time tree view is generated to depict the directory structure dynamically. Users can explore the hierarchy interactively.
- **Path Navigation:**
    - The application displays the current path, helping users keep track of their location within the file system.
- **Intuitive GUI:**
    - Built with Python's Tkinter, the interface includes buttons and prompts for key operations, ensuring ease of use for beginners and experts alike.
- **File List and Directory Traversal:**
    - Users can list files within a directory and navigate between directories using straightforward commands.

Prototype :

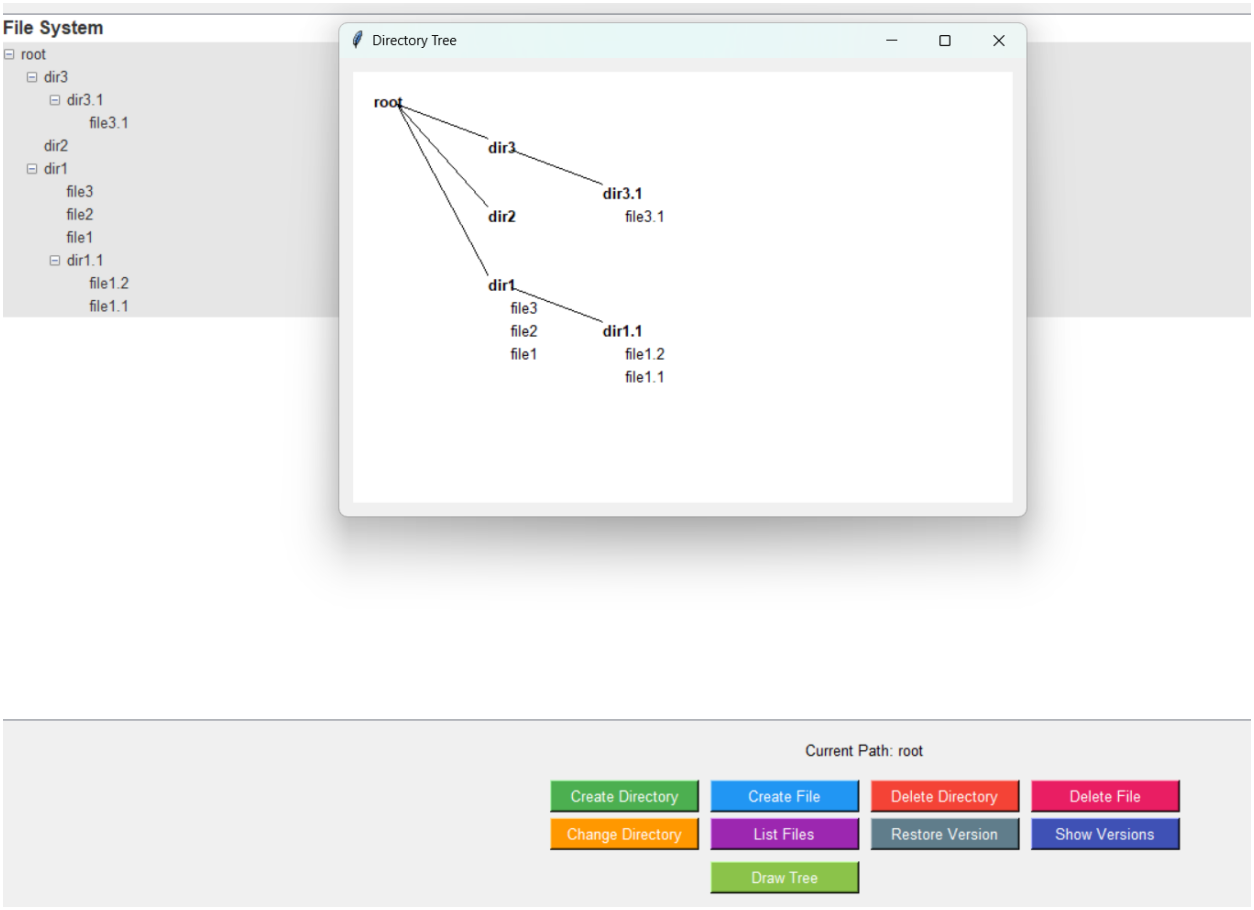

File System Simulation
File System
root
Current Path: /root
Create Directory
Create File
Delete Directory
Delete File
Change Directory
List Files
Restore Version
Show Versions
Draw Tree


File System
root
rootfile1
dir3
dir3.1
file3.1
dir2
dir1
file3
file2
file1
dir1.1
file1.2
file1.1
Current Path: root/dir3/dir3.1
Create Directory
Create File
Delete Directory
Delete File
Change Directory
List Files
Restore Version
Show Versions
Draw Tree

Version control:



Graphical Visualization:



Applications

1. **Educational Tool:**
   - Ideal for students learning data structures and file system management concepts.
2. **Demonstration of File Operations:**
   - Serves as a practical demonstration for teaching hierarchical structures and version control.
3. **Interactive Simulations:**
   - Allows users to simulate real-world scenarios of file and directory management.
4. **Code Testing and Debugging:**
   - Offers a sandbox environment for experimenting with file system operations.

Conclusion.

The "File Management Simulation" project bridges the gap between theoretical understanding and practical implementation of file systems. By simulating essential file management operations, it provides an engaging and hands-on way to explore concepts like directory hierarchy, linked lists, and version control.

The intuitive GUI, dynamic visualization, and efficient operations make this project a versatile tool for both educational and personal use. Future enhancements could include features like file search, user authentication, and cloud integration, making the system even more comprehensive and applicable to real-world scenarios.

This project not only strengthens foundational knowledge but also serves as a stepping stone for building more complex file system utilities, reflecting the importance of structured data management in modern computing.