

File Management System

PROJECT SYNOPSIS

Submitted by

S. No	NAME	UID
1.	Chaitanya Gaba	22BCS15984
2.	Swastik Adhikary	22BCS17205
3.	Karan Joshi	22BCS16768
4.	Mitanshu Garg	22BCS16644

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



Chandigarh University

January – May 2025

INTRODUCTION

The File Management System is a structured and efficient application designed to help users organize, store, and manage their files orderly. It enables users to create, update, search, and delete files while ensuring data security and accessibility. This system eliminates the inefficiencies of traditional manual file handling by providing a structured way to manage documents, making retrieval faster and file organization more systematic. Using file handling and indexing techniques in C++, users can perform file-related operations seamlessly, ensuring an efficient and user-friendly experience.

The File Management System is not only designed for convenience but also for security. Unlike conventional storage methods, where files are at risk of misplacement or corruption, this system ensures data integrity and allows for backup and recovery. Additionally, it is designed to be lightweight, ensuring low system overhead, while remaining scalable for future enhancements such as encryption, cloud integration, and GUI-based interfaces.

A digital file management system significantly improves the efficiency of data handling by allowing users to systematically store and retrieve information. Unlike traditional methods, which require manual browsing through directories and folders, this system provides search and sorting functionalities to quickly locate files using filenames, timestamps, or keywords.

Moreover, this project contributes to digital transformation by offering an organized approach to data management, reducing dependency on physical storage, and enhancing workflow efficiency. The File Management System is particularly useful for students managing academic files, professionals handling work documents, and organizations storing critical data. The integration of search, security, and automation ensures that it serves as a reliable and effective file-handling solution for various use cases.

PROJECT FLOW

The **File Management System** follows a structured flow to ensure smooth user interaction and efficient data handling. Below is a detailed breakdown of the project workflow:

1. File Creation and Modification

Users can create a new file by specifying the filename and entering content. The system assigns a timestamp (date and time) to each file for easy tracking. Files can be edited anytime, allowing users to update or append new content while ensuring data integrity.

2. View Files

Users can view a list of all files they have created. The system retrieves stored files and displays their contents in structured form including filename, timestamp, and data. This feature helps users quickly access their documents without manually searching through directories.

3. Search & Sort

Users can search for files using different criteria:

- **Filename-based search:** Users enter a filename, and the system retrieves the matching file instantly.
- **Date-based search:** Users can enter a date, and the system filters files created or modified on that day.
- **Keyword-based search:** The system scans file contents and returns all documents containing the specified keyword.

4. Backup & Recovery

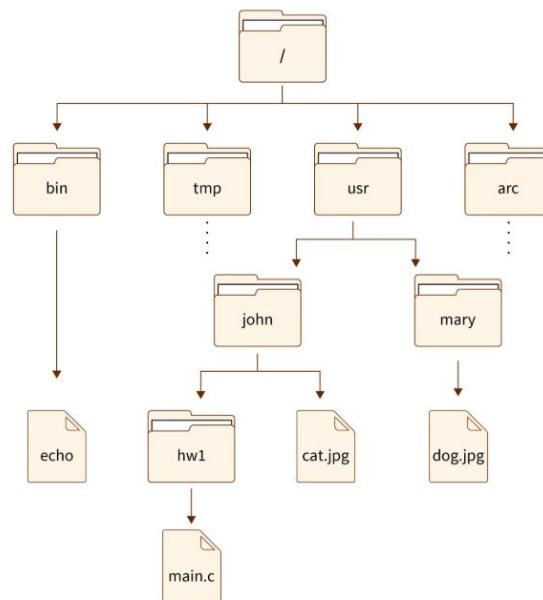
- Users can manually create backups of important files to prevent data loss.
- In case of accidental deletion, files can be restored from the backup system before permanent removal.

5. Delete Entry

- If a user wants to **delete a file**, the system asks for confirmation before permanently removing it.
- Deleted files are **moved to a recovery folder first**, allowing users to restore them if needed.
- If not recovered, files are permanently deleted after a specified duration to free up storage.

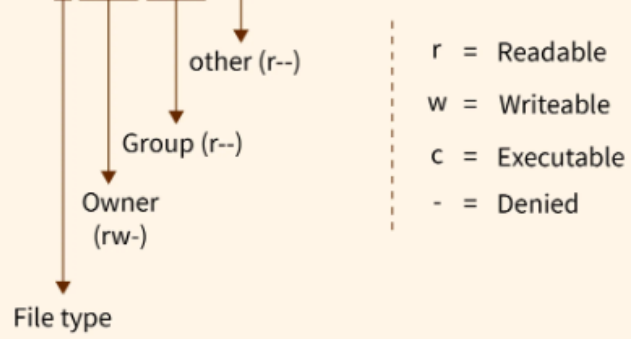
6. Exit System

- When a user exits, the system ensures that **all changes (new files, modifications, deletions) are properly saved**.
- A confirmation message is displayed, and the system safely terminates.



ls -l file_name

- r w - r - - r - -



TECHNOLOGIES (FRONTEND/BACKEND)

1. Frontend:

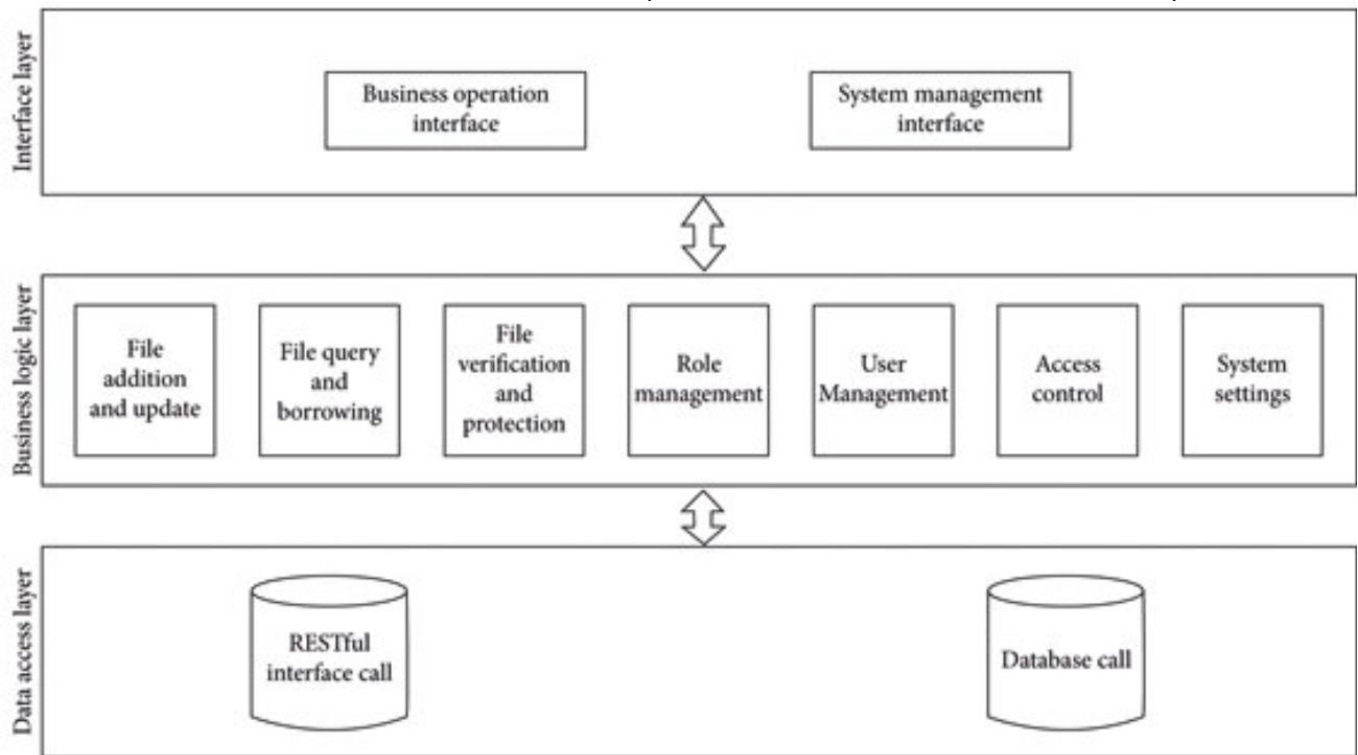
- **User Interface:** A Command Line Interface (CLI) is implemented for user interaction, allowing efficient file management operations.
- **Search Functionality:** Allows users to **search for files by name, content, or date**, making retrieval quick and convenient.
- **Sorting Mechanism:** The system supports **sorting files based on name, creation date, or size** for better organization.

2. Backend:

- **Programming Language:** The system is entirely developed in C++, leveraging its efficiency and speed for handling file operations.
- **File Handling:** Uses **fstream library** in C++ to enable **reading, writing, updating, and deleting files**, ensuring data persistence.
- **Data Storage:** All files are stored in a **structured format**, allowing quick access and organization without requiring an external database.
- **Security:** Implements **basic encryption techniques and access controls** to restrict unauthorized modifications to files.

This setup ensures that the File Management System remains lightweight, efficient, and scalable, making it suitable for both personal and professional use.

TECHNOLOGIES (FRONTEND/BACKEND)



MODULES OF THE PROJECT

1. **User Authentication Module** – Manages **secure user access** with an **optional login system**. It ensures **only authorized users** can access files, enhancing security.
2. **File Operation Module** – Handles **file creation, editing, deletion, and retrieval**. This module ensures **proper organization**, making document management efficient.
3. **Search Module** – Implements **search algorithms** to quickly locate files **by name, keywords, or date**. Sorting options help organize files systematically.
4. **File Handling Module** – Manages **reading, writing, updating, and deleting files** while maintaining **data integrity and accessibility** for smooth operations.
5. **User Interface Module** – Provides a **simple command-line interface**, allowing users to **navigate file operations** such as searching, sorting, and deleting files.
6. **Backup & Recovery Module** – Enables users to **create and restore backups**, preventing **data loss due to accidental deletion or system failures**.
7. **Encryption & Security Module** – Implements **encryption techniques** to **secure files** from unauthorized access, ensuring **data privacy and protection**.
8. **Future Enhancement Module** – Includes **cloud storage, GUI development, and AI-based file organization** for smarter and **more efficient data management**.
9. **Multi-User Collaboration Module** - Allows multiple users to **share and access files securely**, providing **permissions and access control** for better file management.
10. **Encryption** - Implements **encryption techniques** to **secure files** from unauthorized access, ensuring **data privacy and protection**.
11. **Error Handling** - Ensures **smooth system performance** by **detecting and managing errors** like invalid inputs, file corruption, or system crashes.
12. **Storage Optimization Module** - Reduces **file size through compression techniques**, ensuring **efficient storage management and faster file access**.

ROLES OF TEAM MEMBERS

Team Members	Role	Responsibilities
Chaitanya Gaba	Utils.h, Response.h & Search.h	Utils.h provides essential helper functions that streamline various operations within the file management system, while Response.h handles system responses and error messaging to ensure smooth user interactions. Search.h enables fast and precise searching of files and content, improving data retrieval efficiency.
Swastik Adhikary	Read.h, Encryption.h & Delete.h	Read.h handles the retrieval and reading of file content, ensuring efficient access to stored data, while Encryption.h secures file data by applying encryption techniques to protect sensitive information. Delete.h manages the removal of files, ensuring secure deletion and proper freeing of resources.
Karan Joshi	Size.h & Edit.h	Size.h manages file size allocation and ensures efficient storage by tracking and adjusting file sizes, while Edit.h facilitates content modification by handling operations for reading, writing, and editing files.
Mitanshu Garg	Testing	Enhances performance by optimizing file handling and memory usage. Implements encryption and security measures for data protection.

PROJECT APPLICATIONS

1. **Personal Document Management:** Store and manage important personal documents, notes, and records in an **organized and secure** manner, ensuring easy access when needed.
2. **Workplace File Handling:** Maintain **structured records of reports, meeting notes, project documentation, and official communications** for professional use.
3. **Educational Resource Organization:** Systematically store **lecture notes, assignments, research papers, and study materials**, making academic document management efficient.
4. **Health Records Management:** Keep **medical reports, prescriptions, fitness logs, and wellness records** stored safely for easy tracking and future reference.
5. **Legal Document Storage:** Securely store and organize **contracts, case files, agreements, and client records**, ensuring data protection and confidentiality.
6. **Business Data Handling :** Manage **business plans, financial records, client details, and operational documents** for structured and efficient access.
7. **Financial Tracking:** Log expenses, manage budgets, and monitor financial records with secure storage and retrieval options.
8. **Self-Improvement & Goal Setting:** Store **goal-setting plans, personal projects, learning materials, and progress tracking files** to enhance productivity.
9. **Data Protection & Backup:** Securely **store and back up important files**, preventing accidental loss and ensuring long-term data safety.

PROJECT FUTURE SCOPE

1. **Graphical User Interface (GUI):** Upgrading from a **command-line interface (CLI)** to a **graphical user interface (GUI)** will enhance user experience by allowing **intuitive file browsing, drag-and-drop functionality, and visual organization** of stored files.
2. **Database Integration:** Replacing text-based storage with a **database system** will improve **data management, faster search operations, and better security**, ensuring efficient handling of large volumes of files.
3. **Cloud-Based File Synchronization:** Implementing **cloud storage support** will allow users to **access and manage their files across multiple devices**, ensuring data availability and real-time synchronization.
4. **Advanced Encryption Techniques:** Incorporating **advanced encryption algorithms** will **strengthen data protection**, preventing unauthorized access and ensuring the security of sensitive files.
5. **AI-Powered Insights:** Using **machine learning algorithms**, the system can **automatically categorize, sort, and suggest file organization techniques**, improving accessibility and efficiency.
6. **Mobile App Development:** Expanding the system to mobile platforms will provide on- the-go access, increasing convenience and flexibility for users.

This project aims to create a modern, secure, and scalable file management system, laying the groundwork for future advancements in digital file handling and organization.