

**PROJECT AND TEAM INFORMATION**

# Project Title

***SMART FILE RECOVERY SYSTEM***

# Student/Team Information

|  |  |
| --- | --- |
| Team Name: | TriTaskers |
| Team member 1 (Team Lead) | POOJA – 230421245  [rawatpooja1228@gmail.com](mailto:rawatpooja1228@gmail.com) |
| Team member 2 | PARUL – 23042566  nautiyalparul42@gmail.com |
| Team member 3 | MUSKAN RAWAT - 230421123  muskanrawat109@gmail.com |

**PROJECT PROGRESS DESCRIPTION**

# Project Abstract

The main goal of the File Recovery Tool project is to develop a user-friendly desktop application that enables users to safely delete, recover, and manage important files. Often, users accidentally delete critical files or want a safer way to manage file removal without permanently losing data. This tool provides a controlled environment where deleted files are moved to a custom folder instead of being erased, allowing them to be recovered later.

Additionally, the system maintains a detailed log of deleted and recovered files using JSON format, along with a history window to view restoration details. It features a graphical user interface built with Tkinter, including options to delete files, recover them to their original paths, view recovery logs, perform keyword-based search, and toggle between light/dark themes. This tool ensures a simple and effective solution for file management with improved traceability and user experience.

# Updated Project Approach and Architecture

The current approach follows a modular architecture using Python’s built-in standard libraries. The application interface is developed using the tkinter library, providing users with buttons to delete, recover, search, and view recovery history. Each user action invokes backend functions that interact with the file system.

For file handling, the os module is used to create and manage folders (DeletedFile, RecoveredFile), build file paths, and check for file existence.

The shutil module handles actual file movement (shutil.move) during deletion and copying (shutil.copy2) during recovery, preserving metadata. A JSON-based log (log.json) tracks original paths for future recovery. Similarly, restored files are copied back to their original paths and also stored in a RecoveredFile directory, with logs maintained in restore\_log.json Recovery events are timestamped using the datetime module for audit purposes. Light/Dark theme toggle using ttk.Style.

The tool maintains two directories- one for logically deleted files and one for recovered copies. Data flow starts from user interaction with the GUI and ends in file system manipulation and logging. No external communication protocols are used, as the system runs locally.

# Tasks Completed

|  |  |
| --- | --- |
| Task Completed | Team Member |
| * Designed and developed the main GUI layout using Tkinter with frames, buttons, labels, and listboxes. * Implemented file deletion logic using shutil.move() and maintained a persistent deletion log in JSON. * Developed file recovery functionality, restoring files to their original location and tracking recovery in a separate log. * Created a Treeview history window to display restored files with timestamps and paths. * Integrated search functionality to filter deleted files from the listbox. * Implemented a dark mode toggle functionality to switch themes dynamically using ttk.Style. * Added timestamp logging using the datetime module during file recovery. | POOJA  POOJA  POOJA  PARUL  PARUL  MUSKAN  MUSKAN |

# Challenges/Roadblocks

1. **File Integrity During Recovery:** Challenge: Ensuring recovered files are not corrupted or overwritten. Recovery: Used shutil.copy2() to restore files with full metadata and original content.
2. **Accurate Logging of File Operations**: Challenge: Maintaining clear logs of deletions and recoveries. Recovery: Implemented log.json and restore\_log.json to store timestamps and file paths.
3. **User-Friendly Interface Design**: Challenge: Designing an intuitive and accessible interface. Recovery: Built a clean GUI with Tkinter, including buttons, labels, dark mode, and file list display.
4. **Dark Mode Toggle Without UI Breaks:** Challenge: Implementing dark mode without affecting visibility or layout. Recovery: Used a set\_theme() function to dynamically update widget colors and backgrounds.
5. **Efficient File Search and Filtering**: Challenge: Managing and filtering large numbers of deleted files. Recovery: Added a search bar and search\_files() function to filter listbox entries dynamically.

# Tasks Pending

|  |  |
| --- | --- |
| Task Pending | Team Member |
| 1. Add confirmation dialogs before file deletion and recovery. 2. Add a "Refresh" button or auto-refresh after file operations. | PARUL  MUSKAN |

# Project Outcome/Deliverables

The key deliverables of the project include:

* A working desktop application with a GUI for file deletion and recovery.
* Persistent logging of deleted and recovered files using JSON.
* Search functionality and restore history viewer.
* Light/Dark mode toggle for UI accessibility.
* Well-organized source code with modular functions for maintainability.

The final outcome is a secure and user-friendly file recovery tool that ensures data safety, easy restoration, and traceable actions.

# Progress Overview

Approximately 90% of the project is completed. Core functionalities—deleting, recovering, searching, and logging are fully implemented and tested. The GUI is functional and user-friendly, and logging works reliably across sessions. We are currently working on improving usability (e.g., adding recovery and delete confirmation and refresh buttons).

# Codebase Information

# 

|  |
| --- |
| **Repository**:  **Branch**: main  **Important Commits**:   1. Initial GUI setup. 2. Added deletion and recovery logic. 3. Integrated log system and restore history. 4. Implemented dark mode and search functionality. |

# Testing and Validation Status

|  |  |  |
| --- | --- | --- |
| Test Type | Status (Pass/Fail) | Notes |
| 1. File Deletion 2. File Recovery 3. JSON Logging 4. GUI Interface 5. Search Filter 6. Recovery History | PASS  PASS  PASS  PASS  PASS  PASS | Moves files to Deleted folder.  Recovers and restores with metadata.  Log files are updated and saved correctly.  Buttons and lists respond as expected.  Accurately filters deleted file list.  Displays history properly. |

# Deliverables Progress

|  |  |
| --- | --- |
| **DELIVERABLE** | **STATUS** |
| GUI Development | Completed |
| Delete and Recover Logic | Completed |
| Restore History Window | Completed |
| Search Functionality | Completed |
| Light/Dark Mode Toggle | Completed |
| Confirmation Dialog (Delete/Recover)  Refresh Button | Pending  Pending |