**File Manager Tool using Python**

**Project Member:**

**Aliyan Ahmed Cheema FA22-BCE-028**

**Abstract:**

The "File Manager Tool" is a Python-based application developed to simplify file and directory management tasks. Leveraging Python's powerful os and shutil libraries, this tool provides a range of functionalities, including listing directory contents, creating, renaming, and deleting files, moving files across directories, organizing files by type, searching for specific files, and checking disk usage. Designed to streamline daily file management tasks, the tool offers a user-friendly menu-driven interface, ensuring ease of use for both novice and experienced users.

**Introduction:**

File management is a fundamental aspect of any operating system, allowing users to store, organize, retrieve, and manipulate data effectively. With the increasing volume of digital files, there is a growing need for efficient tools that can automate repetitive file operations. The "File Manager Tool" addresses this need by providing a comprehensive solution that integrates various file handling features into a single application. Written in Python, the project demonstrates the practical use of os and shutil libraries to interact with the file system programmatically.

**Project Overview:**

The File Manager Tool offers the following functionalities:

1. **Listing Directory Contents:** Displays all files and subdirectories within a given path.
2. **File Creation:** Allows users to create empty files at specified locations.
3. **File Deletion:** Deletes specified files safely.
4. **Renaming Files/Directories:** Renames files or directories based on user input.
5. **Moving Files/Directories:** Moves files or directories to specified destinations.
6. **File Search:** Searches for files containing specific keywords in their names within a directory and its subdirectories.
7. **Disk Space Checker:** Displays total, used, and free disk space for a given path.
8. **File Organization:** Organizes files into categorized folders (e.g., Images, Documents, Videos) based on their extensions.

**Topics Covered:**

1. **File System Navigation:** Utilizing os.listdir and os.path for directory traversal.
2. **File Handling:** Demonstrating read/write operations using Python.
3. **Error Handling:** Implementing robust error-catching mechanisms.
4. **Disk Usage Analysis:** Using shutil.disk\_usage to extract disk space information.
5. **File Organization Techniques:** Automating sorting and organization of files based on extensions.
6. **User Interaction:** Creating a command-line interface with input handling and validation.
7. **Modular Programming:** Structuring the project with modular functions to improve readability and maintainability.

**Technical Implementation:**

The project is built entirely in Python, relying on standard libraries for its implementation. The os library is used for navigating directories, handling paths, and performing basic file operations, while the shutil library facilitates advanced operations like moving files and calculating disk usage. The application employs a menu-driven interface, enabling users to interact with the tool through simple numeric choices. Input validation and error handling ensure the tool operates smoothly, even with unexpected inputs or inaccessible paths.

**Usage Instructions:**

1. Run the script in a Python-supported environment.
2. Select an option from the menu displayed.
3. Follow the prompts to provide necessary file paths or other details.
4. The tool executes the selected operation and provides feedback on the result.
5. To exit, choose the corresponding menu option.

**Conclusion:**

The File Manager Tool is a versatile and practical application that demonstrates the power of Python in automating file management tasks. By integrating multiple file handling features into a single interface, the project provides a valuable resource for users seeking to optimize their file management workflow. Furthermore, it serves as a learning exercise in applying Python's standard libraries to real-world problems.

**References:**

1. Python Official Documentation: https://docs.python.org/3/library/os.html
2. Python shutil Library: https://docs.python.org/3/library/shutil.html