

Smart File Organizer Application

A Java-based application for automatically organizing files in a directory based on their file extensions. The application provides both a command-line interface (CLI) and a graphical user interface (GUI) for ease of use.

The File Organizer Application helps users keep their directories tidy by sorting files into categorized subfolders. It supports default organization rules for common file types (images, documents, videos, audio, archives, and code) and allows users to add custom rules. The application includes features like dry-run mode, undo operations, logging, and file information retrieval.

Features:

- **Automatic File Organization:** Sorts files into folders based on file extensions.
- **Default Rules:** Pre-configured rules for common file types:
 - **Images:** .jpg, .jpeg, .png, .gif, .bmp
 - **Documents:** .pdf, .rtf, .doc, .docx, .txt
 - **Videos:** .mp4, .avi, .mov, .wmv
 - **Audio:** .mp3, .wav, .aac, .m4a
 - **Archives:** .zip, .rar, .7z
 - **Code:** .java, .py, .js, .html, .css
- **Custom Rules:** Add or remove custom organization rules.
- **Dry Run Mode:** Preview organization changes without executing them.
- **Undo Operations:** Revert the last organization operation.
- **Undo all operations:** Revert all previous operations
- **Logging:** Detailed logging of all operations to a log file.
- **File Information:** Retrieve properties about specific files.
- **Dual Interfaces:** Available as both CLI and GUI applications.

Prerequisites

- Java 8 or higher
- Maven 3.6 or higher (for building the project)

Usage:

Command-Line Interface (CLI)

Run the CLI version using the MainApp class:

- ```
-java -cp target/classes com.mycompany.file_handling.MainApp [base_directory]
[log_file]
```
- ``base_directory``: (Optional) The directory to organize files in. Defaults to ``./files``.
  - ``log_file``: (Optional) The path to the log file. Defaults to ``file_organizer.log``.

The application will present a menu with the following options:

1. Organize Files (Dry Run)
2. Organize Files (Actual Execution)
3. Undo Last Operation
4. Undo all Operations
4. Show Organization Rules
5. Add Custom Rule
6. Remove Rule
7. Show Operation History
8. Get File Information
9. Clear Operation History
10. Exit

### Graphical User Interface (GUI)

### Run the GUI version using the SwingFileOrganizerApp class:

```
java -cp target/classes com.mycompany.file_handling.SwingFileOrganizerApp
```

The GUI will prompt you to select a base directory and then provide buttons for all major operations.

### How It Works:

1. Initialization: The application sets up default organization rules and initializes logging.
2. File Scanning: Scans the specified base directory for files.
3. Rule Matching: Matches each file's extension against the organization rules.
4. Organization: Moves files to appropriate subfolders based on their categories.
5. Logging: Records all actions in a log file for review.
6. Undo Support: Tracks operations to allow reversal if needed.

### Project Structure

file\_handling/

```
├── src/
│ ├── main/
│ │ ├── java/
│ │ │ ├── com/
│ │ │ │ ├── mycompany/
│ │ │ │ │ ├── file_handling/
│ │ │ │ │ │ ├── FileLogger.java # Logging functionality
│ │ │ │ │ │ ├── FileOperation.java # File operation tracking
│ │ │ │ │ │ └── FileOrganizer.java # Core organizer logic
```

- |                   └─ MainApp.java            # CLI application
- |                   └─ OrganizationRule.java   # Rule definition
- |                   └─ SwingFileOrganizerApp.java # GUI application
- |                   └─ pom.xml                # Maven configuration
- |                   └─ README.md             # This file

Below is the visual illustration of the GUI running version (which is made to be the default run)

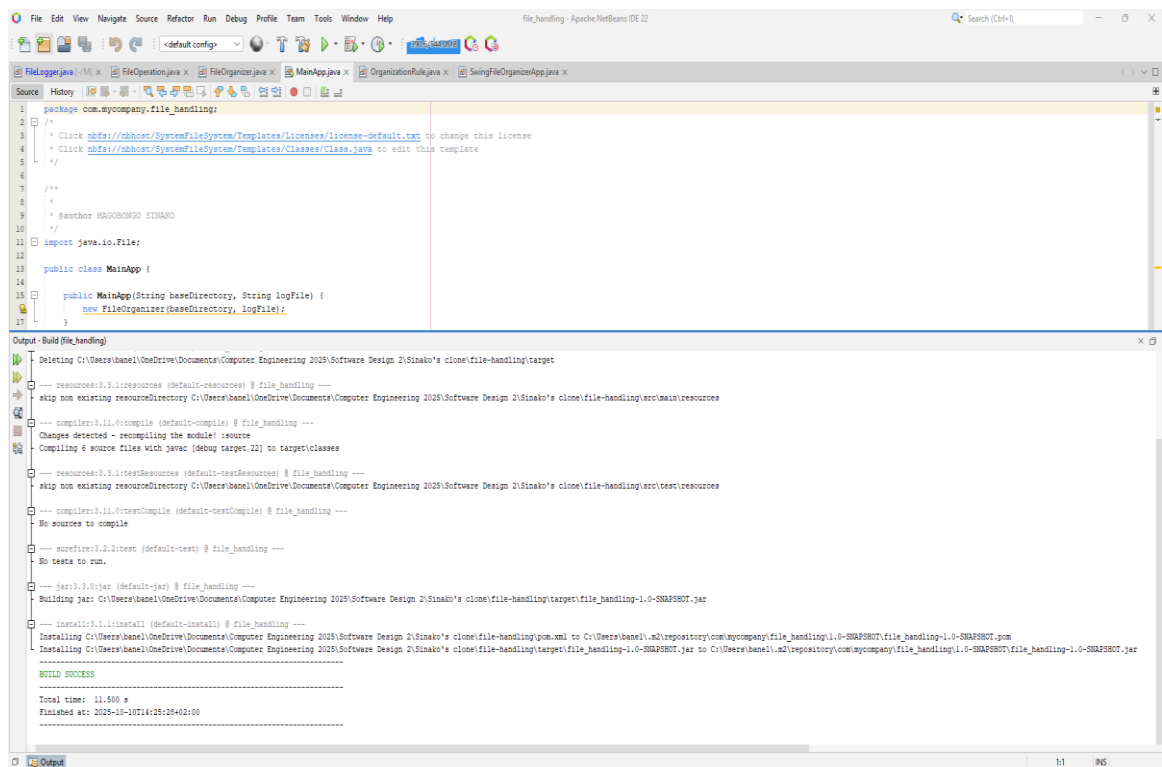


Figure 1: Clean and Build

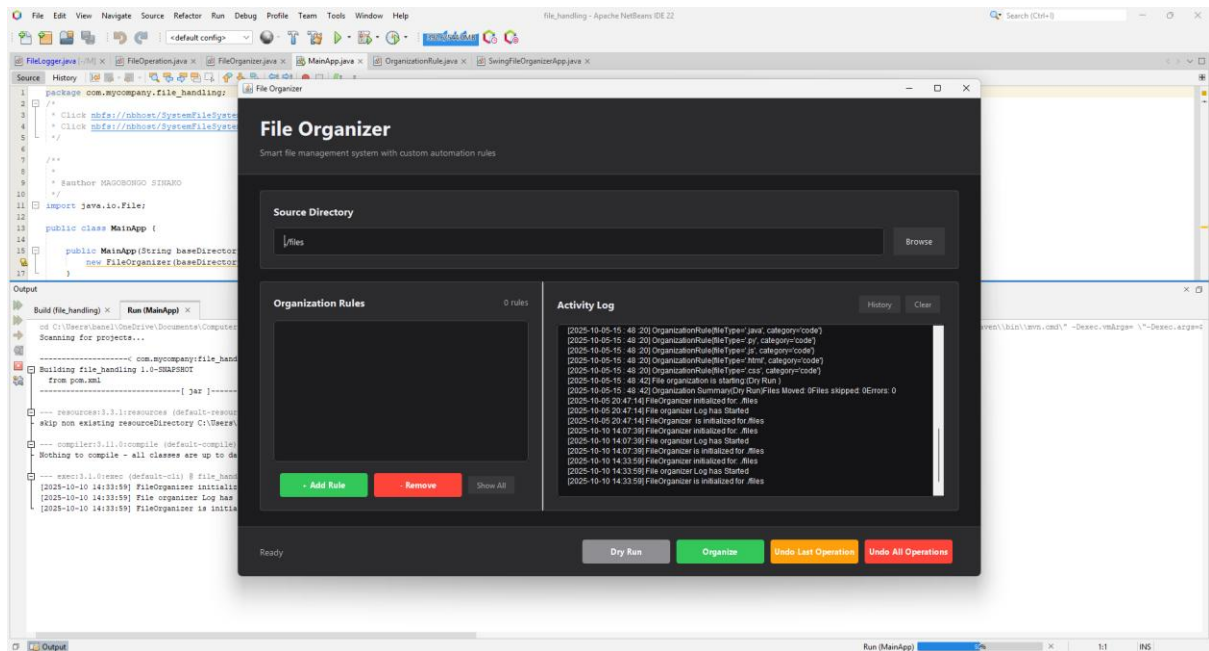


Figure 2: Run MainApp class to access main class

## 1. Navigate the Source Directory panel

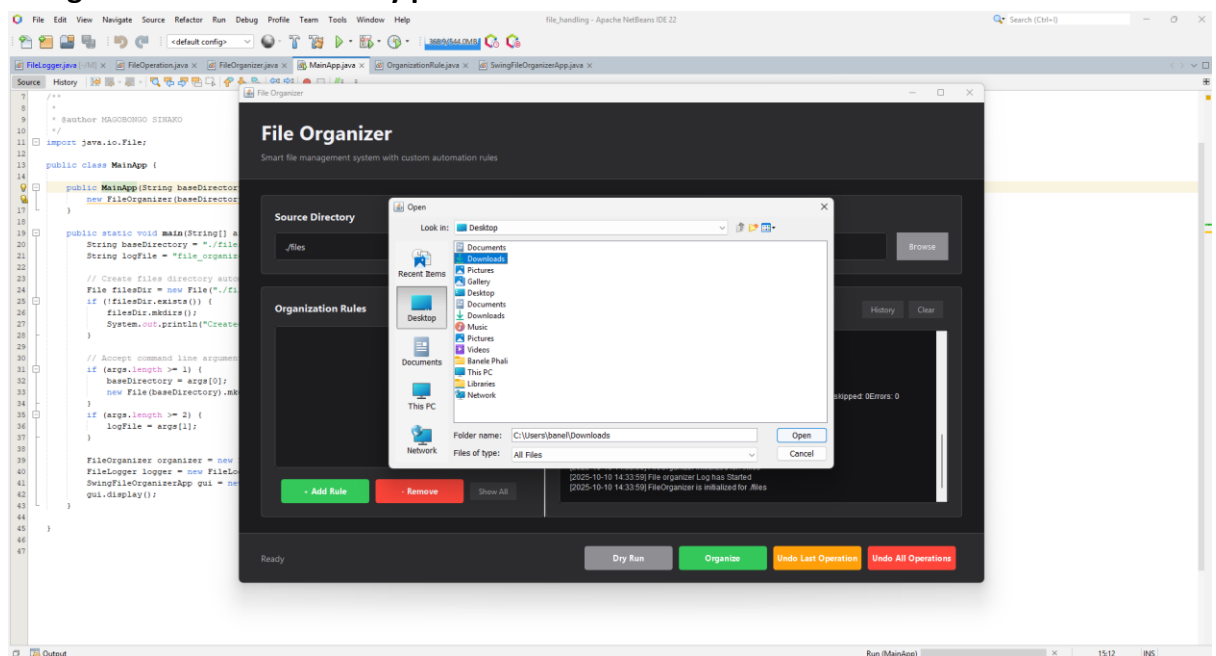
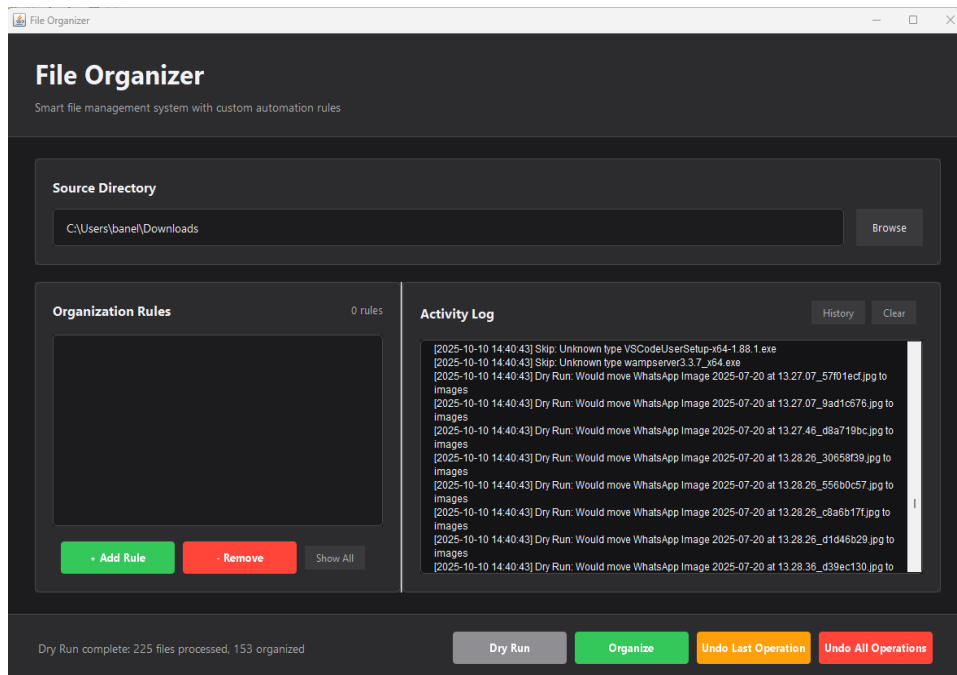


Figure 3: Navigate to the folder you want to organize using the browse button and click 'open'



*Figure 4: You should be able to see your folder's URL*

2. Perform the dry run operation at the bottom to simulate how the actual files will be organized without m



3. Figure 5: Look at the activity log

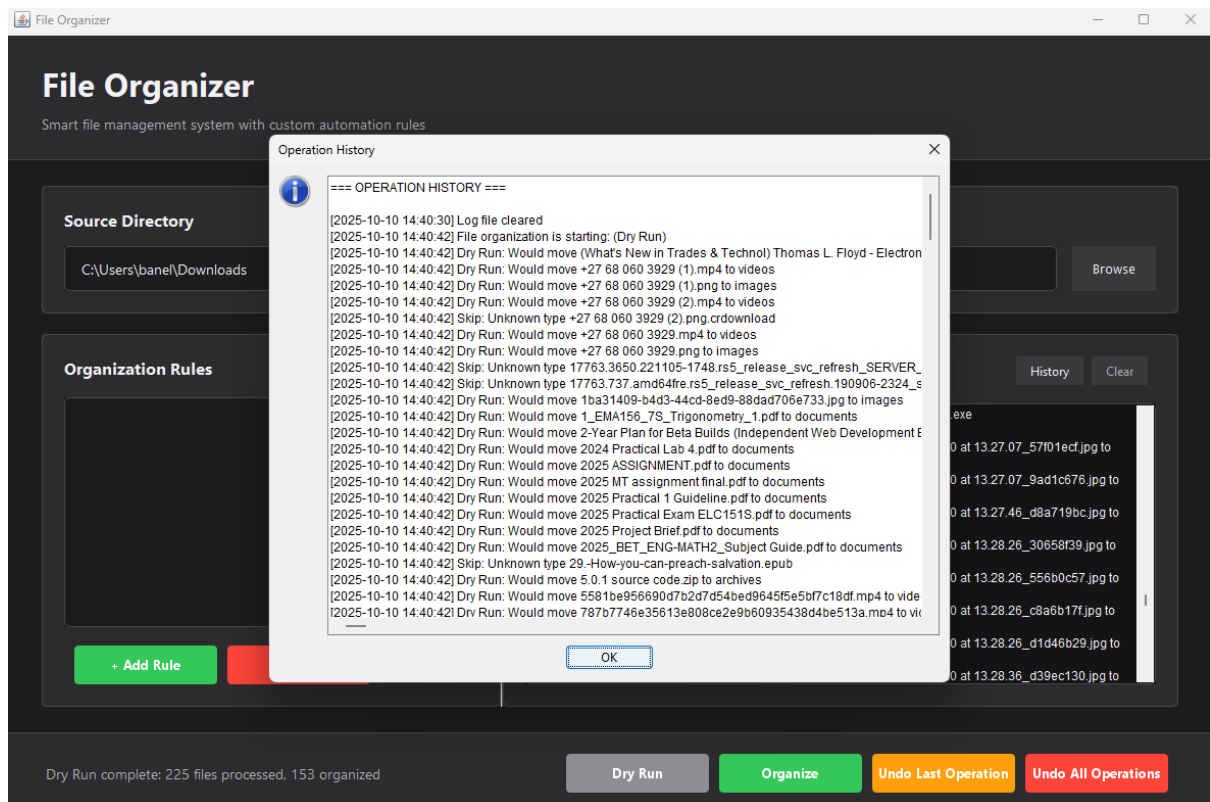


Figure 6: Click the 'History' button to get a clear view of all the log history. Take note: files are not actually moved, but instead, it is reported that they "would move" to said location under actual operation.

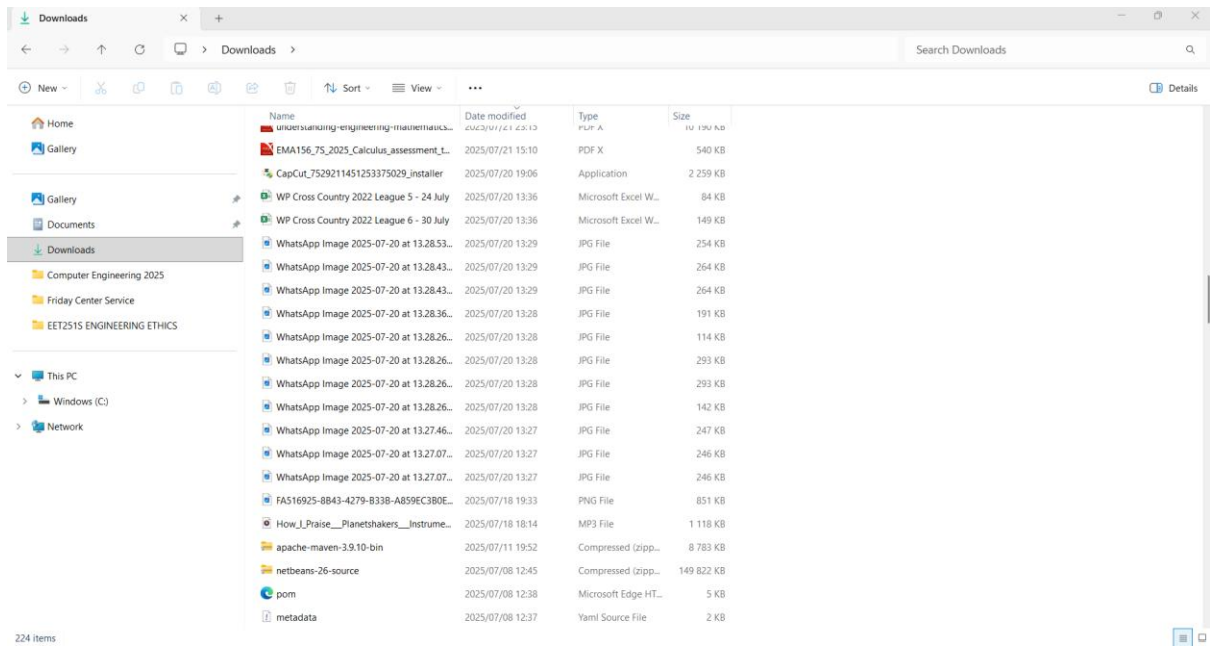


Figure 7: Downloads Folder after dry run

### 3. Organize Folder (Actual Run)

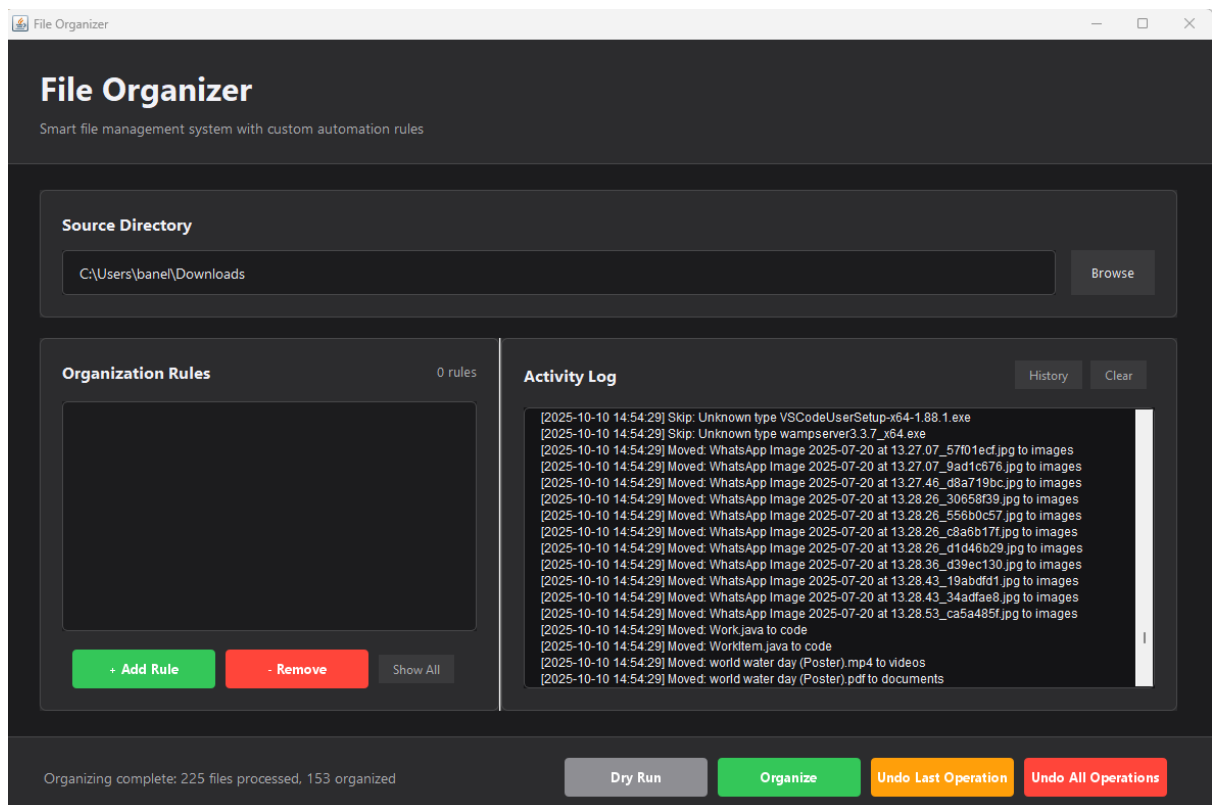


Figure 8: Peek at the terminal and see changes



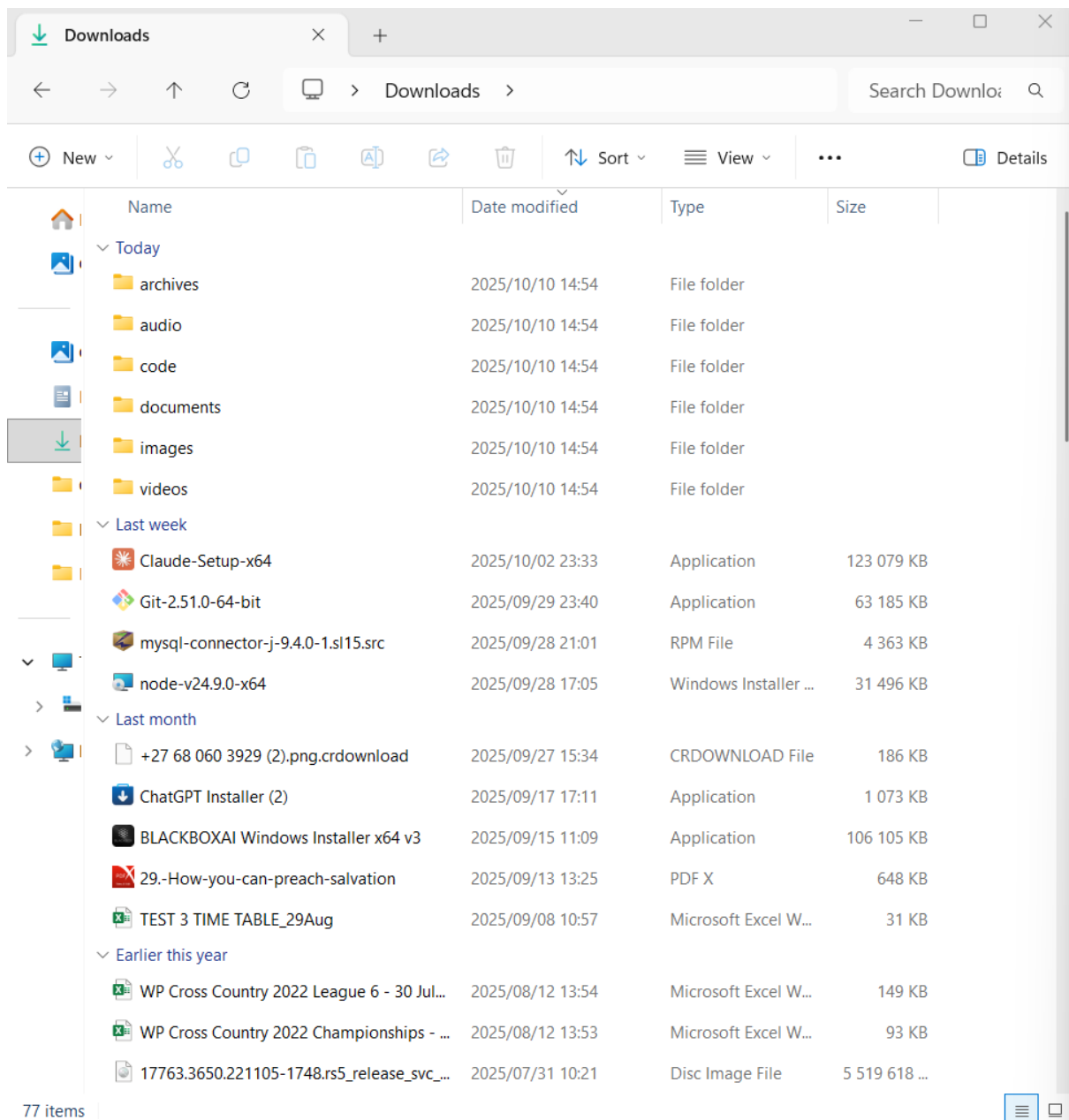


Figure 9: If you look at the item count on the bottom left, it is now less compared to figure 7, meaning the files of the same type have been grouped together according to the default extensions set.

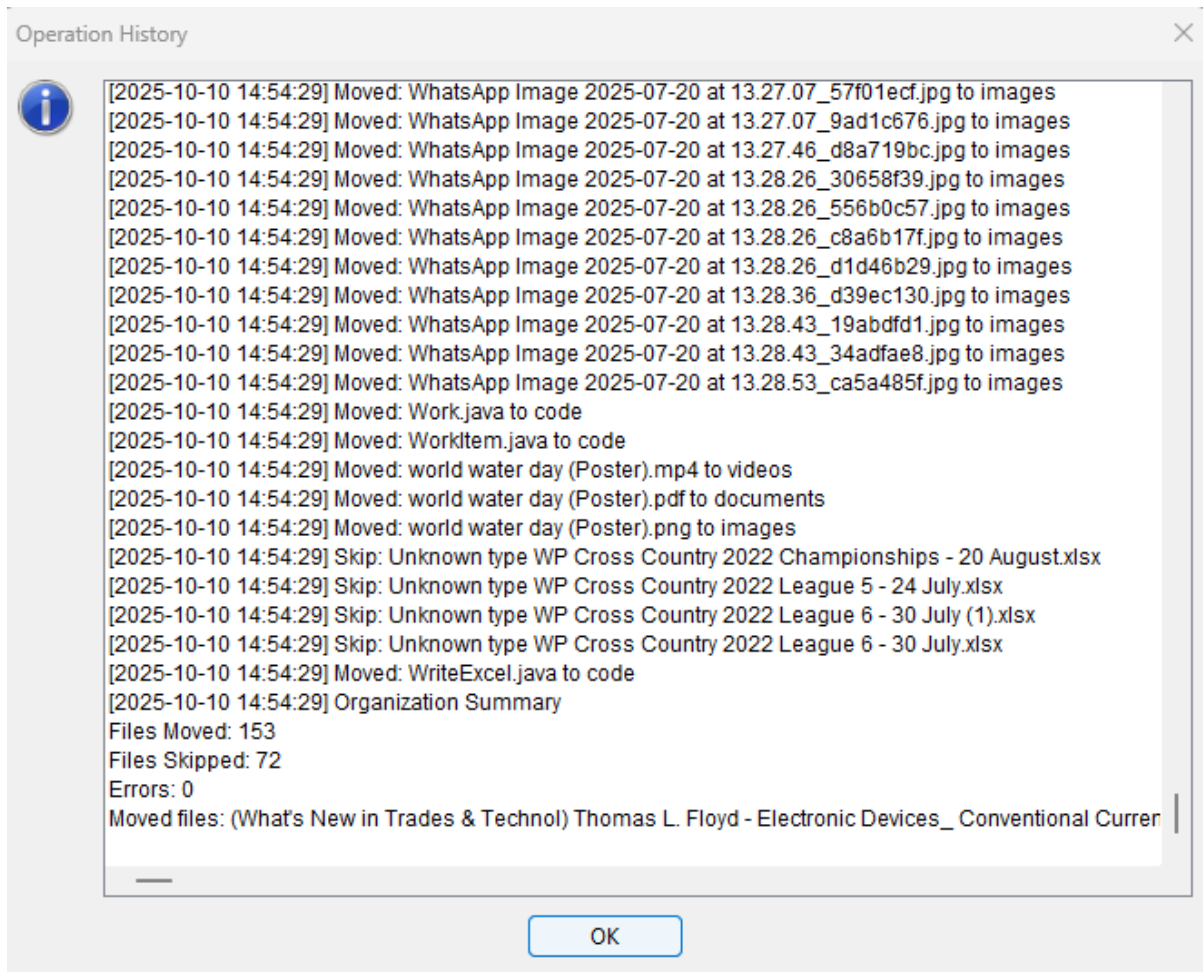


Figure 10: But some files have been skipped, approximately 153, why is this?

Answer: We had not configured for a lot of file types so they user would have to add more file types as presented below in the activity log and 'show all rules' panel.

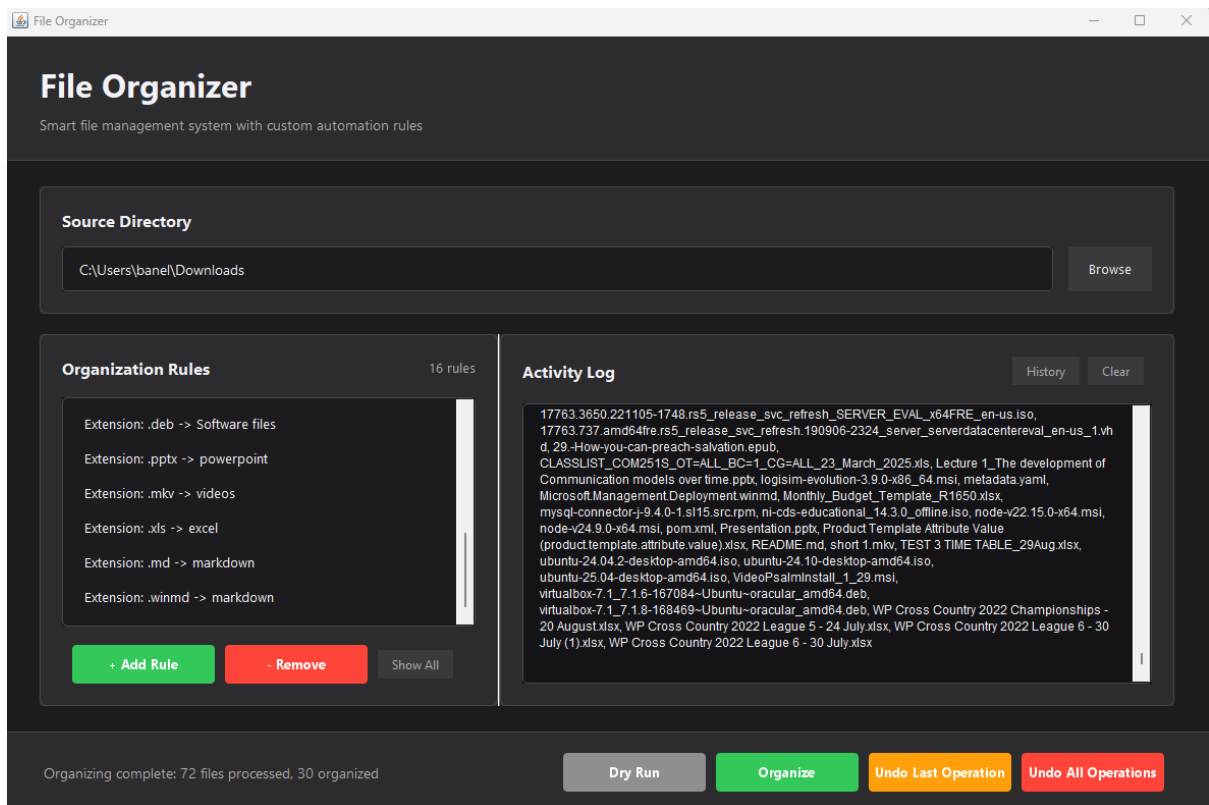
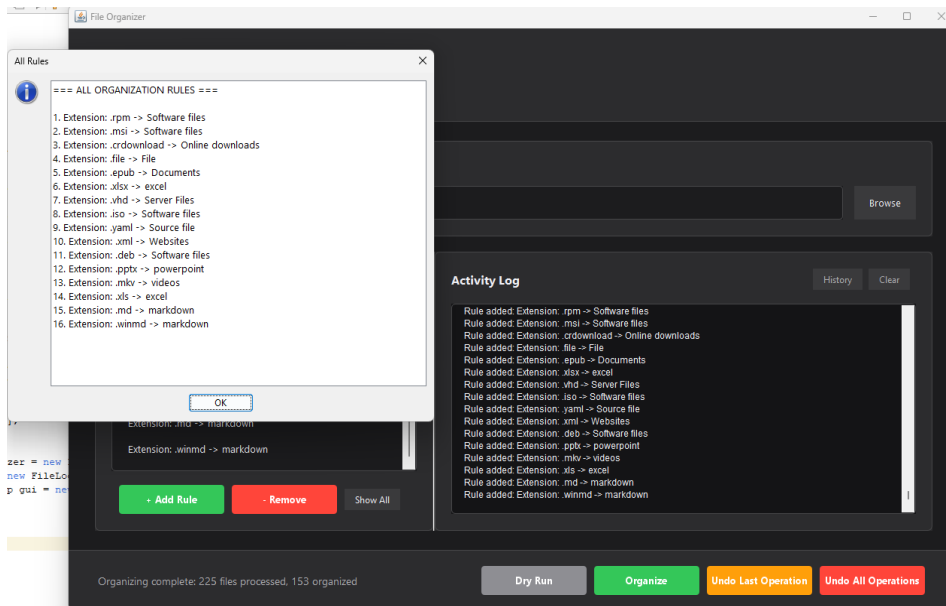


Figure 11: Run Again. We still have 30 unorganized files, so we go back and check our file types left and add more rules.

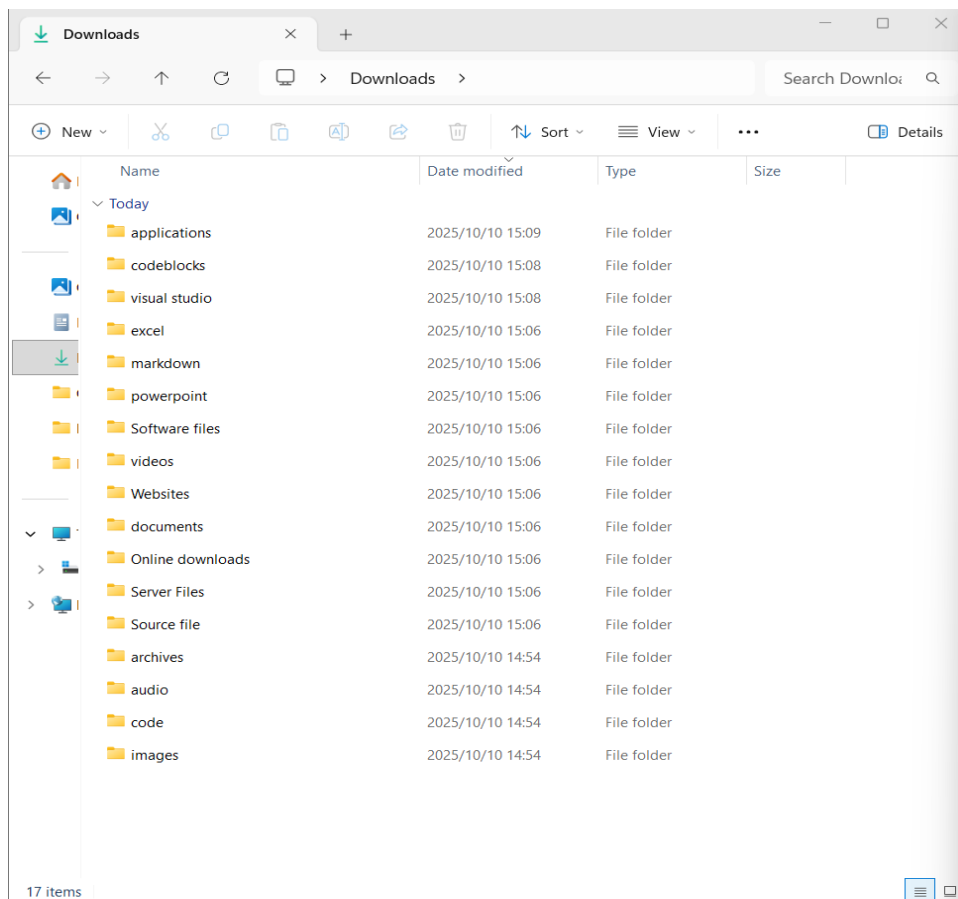
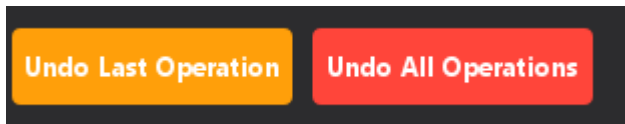


Figure 12: Voila, a fully organised downloads folder

## 4. Undo Operation

Now what if we want to undo an operation?

- We have two options, either undo only the last operation or undo all of them.



a) Click Undo Last Operation & Confirm

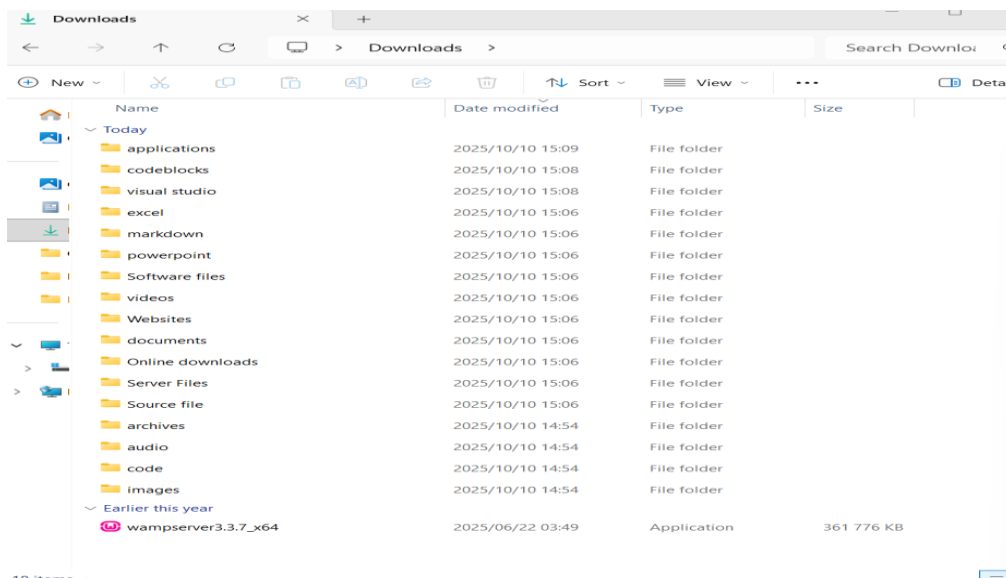
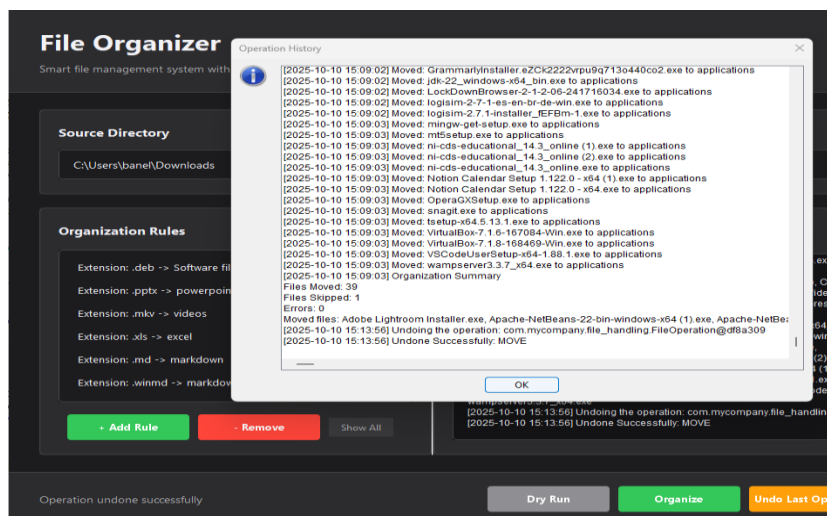


Figure 13: 'wampserver' file moved back to original location

b) Click Undo All Operations

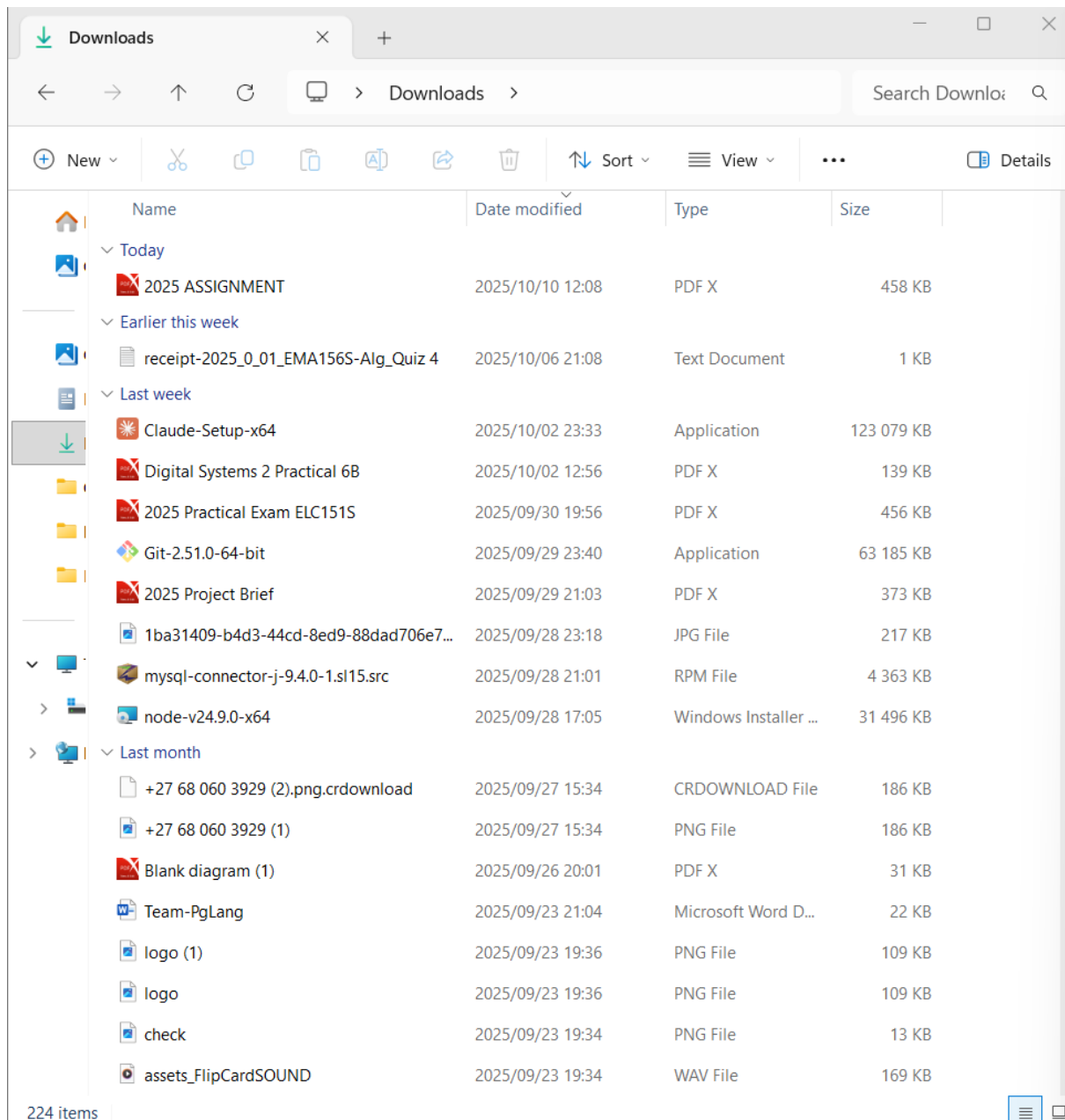


Figure 14: Undone all operations successfully