

Design Document: Organize All Files

Author: Maxwell Lokshin

Date: 07/02/2025

Status: Draft






1. Overview

Build an application that automatically organizes all files located in key folders on my desktop/downloads into defined folders based on file type, metadata, or naming conventions. One click solution to declutter and sort files for easier access and productivity.

2. Problem Statement

Maintaining a clean and organized digital workspace has posed quite the challenge, especially when frequently downloaded or generated files. This project aims to eliminate the manual overhead of file organization by providing a simple tool to help categorize and sort files instantly.

3. Goals

-  Create application executable in one click
-  Automatically organize files into folders such as:
 - Code
 - Documents
 - Images
 - Etc.
-  Doesn't remove files from their original folders but rather moves the folder to their assigned positions
-  Customizable rules
-  Scheduled organization

4. Non-Goals

- Cross-platform mobile support
- Cloud backup
- Deep content-based classification (analyzing file content to determine purpose)

5. Technical Design

5.1 Tech Stack

Layer	Tool
Language	Python
GUI	React
File System Access	Os, shutil, pathlib
Packaging	PyInstaller

5.2 Component Breakdown

- **UI Layer:** simple interface with a button
- **File Scanner:** Scans specified directories for files
- **Categorizer:** Matches files to categories based on type
- **File Mover:** Moves files to subfolders

5.3 Data Model

- **Categories:** Dictionary mapping files extensions to folder names.

```
{
  ".py": "Code",
  ".pdf": "Documents",
  ".jpg": "Images",
  ".mp4": "Videos",
  ...
}
```
- Order of sorting:
 - Name
 - Common name
 - Type of file
 - Extra

5.4 State Management

- **Minimal:** temporary state during scanning and sorting

6. Tradeoffs and Considerations

- Simplicity vs. flexibility: One-click vs. customizable rules
- Risk of incorrect file categorization
- Permission issues when moving protected files
- Possible conflict with existing folder structures

7. Testing Plan

- **Unit Tests:**
 - Extension-to-category mapping
 - File moving logic
 - Error handling
- **Manual Tests:**
 - Different OS environments
 - Files with duplicate names
 - User-defined settings

8. Timeline

Task	Time
Category mapping	0.5 day
Basic file scanning and moving	2 days
Create GUI	1-2 days
Testing and debugging	1-2 days

9. Future Improvements

- Machine learning-based categorization
- Cloud sync
- User notification or undo feature
- Contextual organization by project or tags

10. Appendix

Sources:

- PyInstaller
- GUI - React
- Python shutil documentation