

PROJECT 1

Automation Script – File Organizer

Overview

The File Organizer Automation Script is a Python tool designed to efficiently sort files in a specified directory into categorized folders based on their file extensions. It simplifies file management by organizing files into predefined categories, such as Documents, Images, Videos, and Others, while logging all actions for transparency.

1.1 SYSTEM REQUIREMENTS

To run the script, you need:

- **Python:** Version 3.6 or higher.
- **Operating System:** Compatible with Windows, macOS, or Linux.
- **Permissions:** Read and write access to the directory you wish to organize.
- **Basic Knowledge:** Familiarity with executing Python scripts via a terminal or command line.

1.2 INSTALLATIONS

- **Acquire the Script:**
 - Download file_organizer.py from the project source or repository.
 - Save it to a location on your system, e.g., C:\Tools\file_organizer.py.
- **Set Up Python:**
 - Install Python 3.6+ from python.org if not already installed.
 - Confirm installation by running python --version in a terminal or command prompt.
- **Dependencies:**
 - The script relies on Python's built-in os, shutil, and logging modules, which require no additional installation.

1.3 USING THE SCRIPT

Step-by-Step Instructions

- **Select a Directory:**
 - Choose the folder you want to organize (e.g., C:\Users\YourName\Downloads).
 - Ensure you have the necessary permissions to modify this folder.
- **Execute the Script:**
 - Open a terminal or command prompt.
 - Navigate to the script's directory using cd (e.g., cd C:\Tools).
 - Run the script with:

python file_organizer.py

- Enter the full path of the folder to organize when prompted (e.g., D:\Folder1 or /home/user/downloads).
- **Script Operations:**
 - The script scans the specified directory for files.
 - Files are categorized by their extensions into folders like:
 - **Documents:** .pdf, .docx, .txt, .xlsx, etc.
 - **Images:** .jpg, .png, .gif, etc.
 - **Videos:** .mp4, .mkv, .avi, etc.
 - **Others:** Files with unrecognized or no extensions.
 - Subfolders are created automatically if they don't exist.
 - Files already in the correct folder are skipped.
 - All actions (moves, skips, errors) are logged to file_organizer.log.

Example Folder Transformation

Before:

D:\Folder1

```
|— selfie.jpg
|— report.pdf
|— video.mp4
|— script.py
```

After:

D:\Folder1

```

|— Documents
|   |— report.pdf
|— Images
|   |— selfie.jpg
|— Videos
|   |— video.mp4
|— Others
|   |— script.py

```

1.4 LOG FILE

- A log file, file_organizer.log, is generated in the script's directory.
- It records:
 - Files moved to their respective categories.
 - Files skipped (already in correct folders).
 - Any errors encountered (e.g., permission issues).

- Example log entry:

2025-05-26 19:30:45 - INFO - Moved 'report.pdf' to 'Documents'

1.5 ERROR HANDLING

The script includes robust error handling for:

- Invalid directory paths.
- Lack of read/write permissions.
- Files currently in use.
- General file system errors. Errors are displayed in the console and logged for review.

1.5 TESTING

To verify the script's functionality:

1. Create a test folder with sample files (e.g., .jpg, .pdf, .mp4, .py).
2. Run the script and input the test folder's path.
3. Check the folder to confirm files are sorted into Documents, Images, Videos, and Others.
4. Review file_organizer.log to ensure all actions are logged correctly.

SAMPLE OUTPUT





THE PATH OF FOLDER TO ORGANIZE NEED TO BE PROVIDED

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTMAN CONSOLE
[Running] python -u "c:\SIRISHA\@Internship\FILE_ORGANIZER\File_organizer.py"
File Organizer Script
-----
Enter the full path of the folder to organize:

```

BEFORE ORGAINZING THE FOLDER LOOKS LIKE BELOW

<input type="checkbox"/> Name	Date modified	Type	Size
 App	29-04-2025 02:27	JavaScript Source File	39 KB
 payment	04-05-2025 20:46	PNG File	44 KB
 Portfolio Website	15-07-2024 23:06	MP4 File	37,434 KB
 Receipt pay_QQthrgqou3cWce from PITABA...	04-05-2025 20:47	Microsoft Edge PDF ...	37 KB

AFTER EXECUTING THE PYTHON SCRIPT

```





Enter the full path of the folder to organize: D:\NEW FOLDER

📁 Organizing files in: D:\NEW FOLDER

✅ Moved: App.js → Others/
✅ Moved: File_oragnizer.py → Others/
✅ Moved: payment .png → Images/
✅ Moved: Portfolio Website.mp4 → Videos/
✅ Moved: Receipt pay_QQthrgqou3cWce from PITABAS PRADHAN (Paid).pdf.pdf → Documents/

🎉 Organization complete.
PS C:\SIRISHA\@Internship\Automation-Scrpit-File-Organizer>

```

<input type="checkbox"/> Name ^	Date	Type	Size
 Documents	26-05-2025 20:59	File folder	
 Images	26-05-2025 20:59	File folder	
 Others	26-05-2025 20:59	File folder	
 Videos	26-05-2025 20:59	File folder	

1.6 CUSTOMIZATION

- **Adjust File Categories:**

- Edit the FILE_CATEGORIES dictionary in the script to modify file types or add new categories. For example:

```
FILE_CATEGORIES = {
    'Documents': ['.pdf', '.docx', '.txt', '.xlsx'],
    'Images': ['.jpg', '.png', '.gif'],
    'Videos': ['.mp4', '.mkv', '.avi'],
    'Archives': ['.zip', '.rar'], # New category
    'Others': []
}
```

- Save the script after changes.

- **Set a Default Directory:**

- Hardcode a directory path in the script to bypass the input prompt for repeated use.

1.7 TROUBLE SHOOTING

- **Script Fails to Run:** Ensure Python is installed and added to your system's PATH. Run `python --version` to verify.
- **No Files Moved:** Check the directory path and ensure it contains files with supported extensions.
- **Permission Errors:** Run the terminal as an administrator or verify folder permissions.
- **Console Encoding Issues:** If emojis cause errors (e.g., `UnicodeEncodeError`), remove them from print statements or set `os.environ["PYTHONIOENCODING"] = "utf-8"` at the script's start.

BANTUPAKIRI SIRISHA