Smart File Organizer (CLI)

A Python command-line tool to automatically organize files in a directory into categorized folders based on their file extensions. This tool is safe, fast, and easily extensible.

Features

- Categorizes files such as Documents, Images, Audio, Videos, Ebooks, Fonts, and Others.
- Automatically creates folders if they do not exist.
- Safe file movement with logging of all operations.
- Works with any directory path provided by the user.
- Configuration-driven categories via `config.py`, making it easy to add new types.
- Supports dry-run mode to preview changes without moving files.
- Designed for CLI use and future enhancements like undo.

Installation

1. Clone the repository:

git clone https://github.com/yourusername/smart-file-organizer.git cd smart-file-organizer

1. (Optional) Create a virtual environment:

python -m venv venv source venv/bin/activate # macOS/Linux venv\Scripts\activate # Windows

1. Install dependencies (if any):

pip install -r requirements.txt

Currently, this project uses only the Python standard library.

Usage

Run the organizer:

python main.py

You will be prompted to **enter the directory path** you want to organize and whether you want to run in dry-run mode:

```
Enter directory to organize: /path/to/your/folder
Dry-run mode? (y/n): y
```

- **Dry-run mode (** `y` **)**: Shows what files would be moved without actually moving them.

• **Normal mode (n):** Moves files into categorized folders.

Example Dry-Run Output

```
[DRY-RUN] Would move 'document1.pdf' to 'Documents/'
[DRY-RUN] Would move 'photo1.jpg' to 'Images/'
[DRY-RUN] Would move 'song1.mp3' to 'Audio/'
[DRY-RUN] Would move 'video1.mp4' to 'Videos/'
[DRY-RUN] Would move 'ebook1.epub' to 'Ebooks/'
[DRY-RUN] Would move 'font1.ttf' to 'Fonts/'
[DRY-RUN] Would move 'unknown.xyz' to 'Others/'
✅ Dry-run complete. No files were moved.
```

Example Folder After Real Run

```
test_folder/
├── Audio/
│   ├── song1.mp3
│   └── song2.wav
├── Documents/
│   ├── document1.pdf
│   └── document2.xlsx
├── Images/
│   ├── photo1.jpg
│   └── photo2.png
├── Videos/
│   ├── video1.mp4
│   └── video2.mov
├── Ebooks/
│   └── ebook1.epub
├── Fonts/
│   └── font1.ttf
└── Others/
    └── unknown.xyz
```

Project Structure

```
smart-file-organizer/ ├── main.py ├── organizer/ │ ├── organizer.py │ ├── config.py │ ├──
utils.py │ └── logger.py └── README.md
```

• `main.py` — CLI entry point that handles user input, validation, and dry-run.
• `organizer/organizer.py` — Core logic for file categorization and movement.
• `organizer/config.py` — File category configuration.
• `organizer/utils.py` — Utility functions (e.g., ensure folders exist).

- `organizer/logger.py` — Logging setup for tracking file operations.

File Categories

Files are organized based on extensions defined in `config.py`:

- **Documents:** pdf, docx, txt, xlsx, pptx, csv, etc.
- **Images:** jpg, jpeg, png, gif, bmp, tiff, svg, webp, heic
- **Audio:** mp3, wav, aac, flac, ogg, m4a, wma
- **Videos:** mp4, mov, avi, mkv, flv, wmv, webm, mpeg
- **Ebooks:** epub, mobi, azw3
- **Fonts:** ttf, otf, woff, woff2
- **Others:** Any unknown or uncategorized file types

Future Enhancements

- Undo last operation.
- Duplicate file handling to avoid overwriting.
- Additional CLI arguments for non-interactive use.

License

This project is licensed under the MIT License.