JUNK FILE ORGANISER

A Python CLI

Rajeshwaran K
AUFS028033

JUNK FILES ORGANISER

Table of contents:

Intro :

Junk File Manager is a command line tool developed to serve as your instant file organiser. It can help you organise your files based on File type, Date Modified, Alphabet and File Size.

I have used python **inbuilt** libraries like **Pathlib, os, shutil and datetime.** So user don't need to install any modules manually to run this program.

Installation and Usage Guide:

Download the zip file and run the main.py file from your terminal followed by two optional positional arguments (organizeby and folderPathAddress)

*C:\ JunkFileOrganiser> python main.py **organizeby folderPathAddress***

- **Organizeby**

There are three modes to organise your files using this program you can select one based on your preference.

- byFileSize (Default If not provided by user)
- byextension
- bydatemodified
- byalphabet

byfilesize : Files are moved to four different folders respective of their size. (below 1 MB, Below 10 MB, Below 50 MB and Above 50 MB)

byextension: Files are moved to different type of folder respective of their category (e.g. .pdf files to PDF folder, .mp4 files to Videos folder)

bydatemodified: : Files are moved to different type of folder respective of their last modified date (e.g. A file named 123.jpg which has last

modified date of 09-03-2022 (date-month-year) will be moved to a folder named “09-03-2022”

byalphabet: Files are moved to folder named from their initial letter. Eg:apple.jpg will be moved to folder named ‘A’

- **FolderPathAdress:**

It is the address of the folder you wish to organise. If your folder named ‘test’ in desktop your **FolderPathAdress would be like** *“C:\Users\username\desktop\test”.* (You should pass it in double quotes)

You can also see updated info about positional arguments and other future update in your terminal by typing *python main.py –help*

Modules Description:

- **Datetime**

The datetime module supplies classes for manipulating dates and times.

In this project *“datetime.datetime.fromtimestamp(epoch_time).strftime('%d-%m-%Y')* “ is used to convert the epoch time to human readable format (Date/Month/year).

- **Pathlib**

Path is a subclass of PurePath that can make system calls.

It represents the system path of a file when applied we can also make use of many inbuilt methods (e.g.: suffix - the final component's last suffix, if any. This includes the leading period. For example: '.txt')

- **Os**

The “*os.scandir()”* method in Python is used to get an iterator of “*os.DirEntry”* objects corresponding to the entries in the directory given by the specified path.

In this project “*os.stat(item).st_mtime”* is used which returns last modified time in epoch format.

- **Shutil**

The shutil module offers a number of high-level operations on files and collections of files.

In this project *shutil.move(file_path, new_path)* is used to move the files to newly created path (new_path) from source_path (file_path).

Files :

- Main.py
- modules
    - DateModified.py
    - Directories.py
    - FileSize.py
    - Extension.py
    - Alphabet.py
