FILE ORGANIZER USING PYTHON

Student Name – Nandini Singh (25BAI10803)

Introduction

Organizing files manually can be time-consuming, especially when a directory contains large numbers of documents, media, and other file types. The File Organizer project aims to automatically arrange files into folders based on their extensions. This Python-based utility improves workflow efficiency, reduces clutter, and ensures systematic file management with a single click.

Problem Statement

Users often struggle with unorganized folders cluttered with images, documents, videos, and miscellaneous files. Manually sorting them is tedious and prone to errors.

The problem:

- Lack of structure in file storage
- Difficulty in locating files
- Wasted time in manual organization

Thus, a program is needed that automatically categorizes files into properly named folders.

Functional Requirements

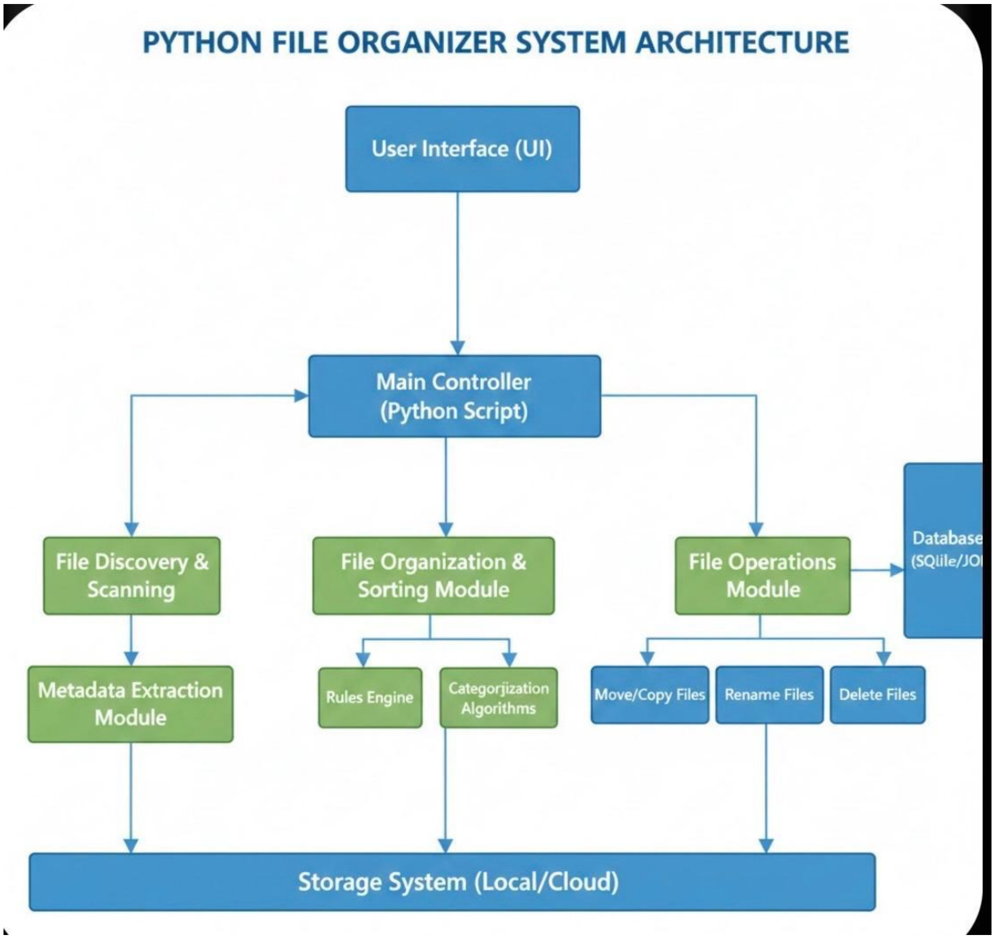
The File Organizer should:

1. Accept a folder path as input
2. Scan all files in the directory
3. Detect file extensions
4. Create folders based on extensions
5. Move corresponding files into their respective folders
6. Handle missing or unknown extensions
7. Skip directories to avoid breaking nested structures

Non-Functional Requirements

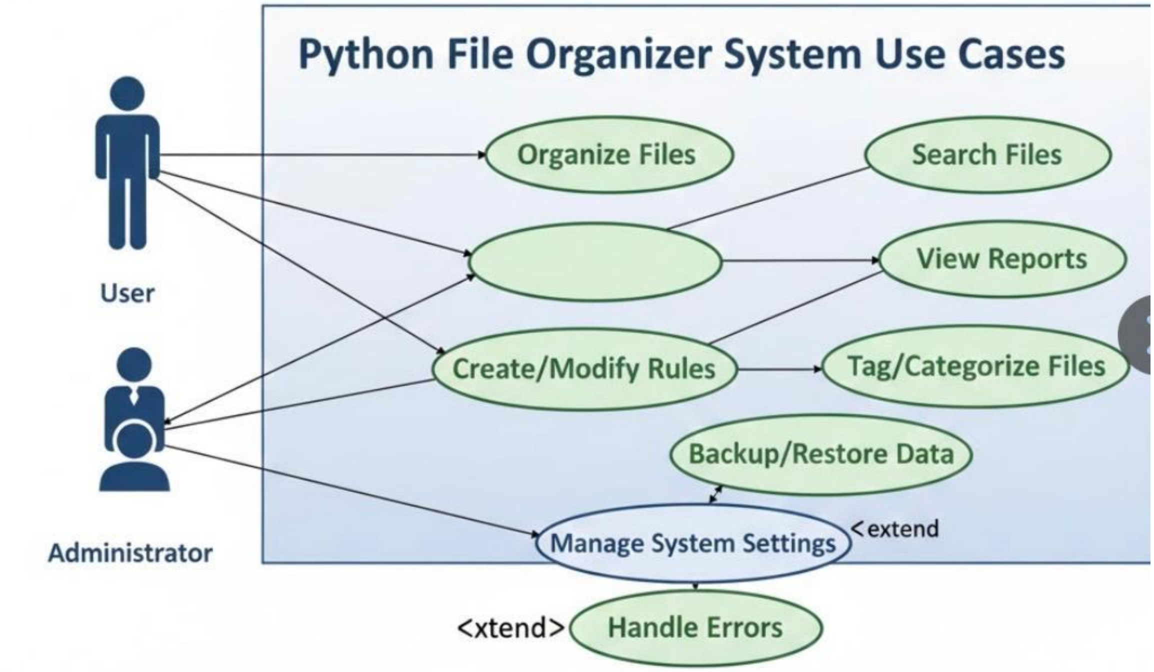
- Reliability: Should correctly move files without losing data
- Usability: Easy to run with simple input
- Performance: Must organize files within seconds
- Portability: Runs on any OS with Python installed
- Maintainability: Code should be readable and modifiable

System Architecture

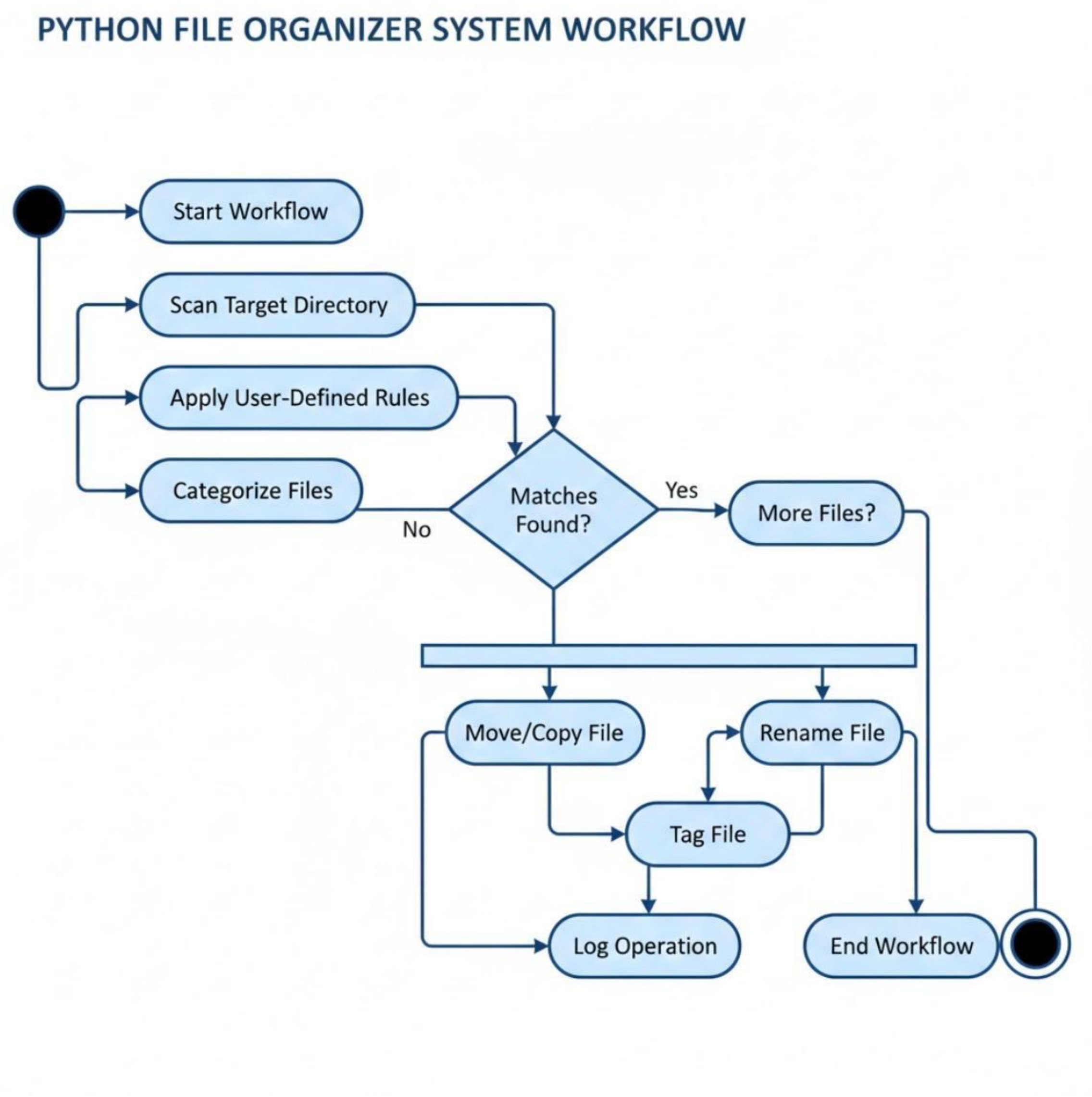

PYTHON FILE ORGANIZER SYSTEM ARCHITECTURE
User Interface (UI)
Main Controller (Python Script)
File Discovery & Scanning
File Organization & Sorting Module
File Operations Module
Database (SQlile/JO
Metadata Extraction Module
Rules Engine
Categorjization Algorithms
Move/Copy Files
Rename Files
Delete Files
Storage System (Local/Cloud)

Design Diagrams

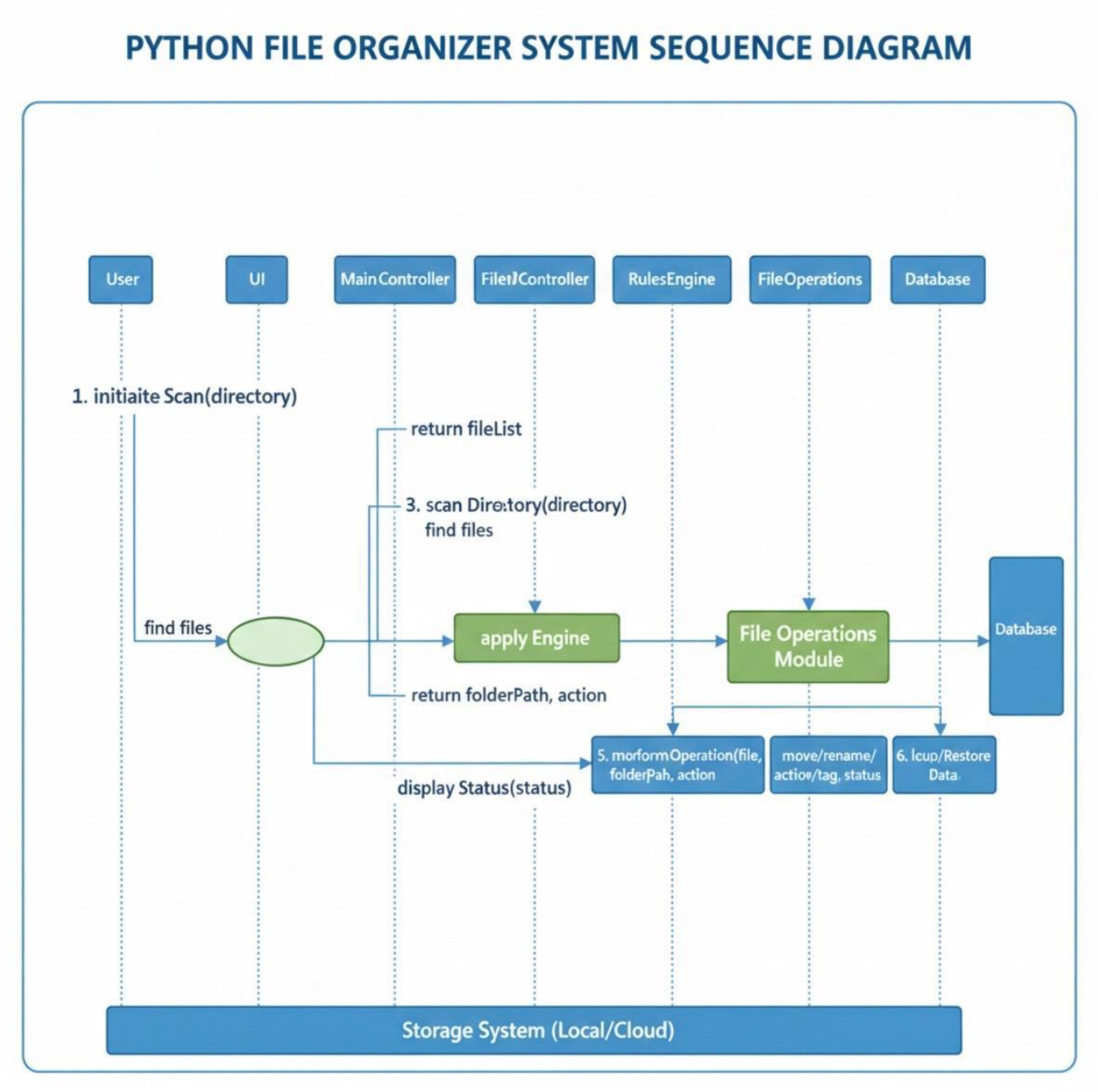
1.User case Diagrams



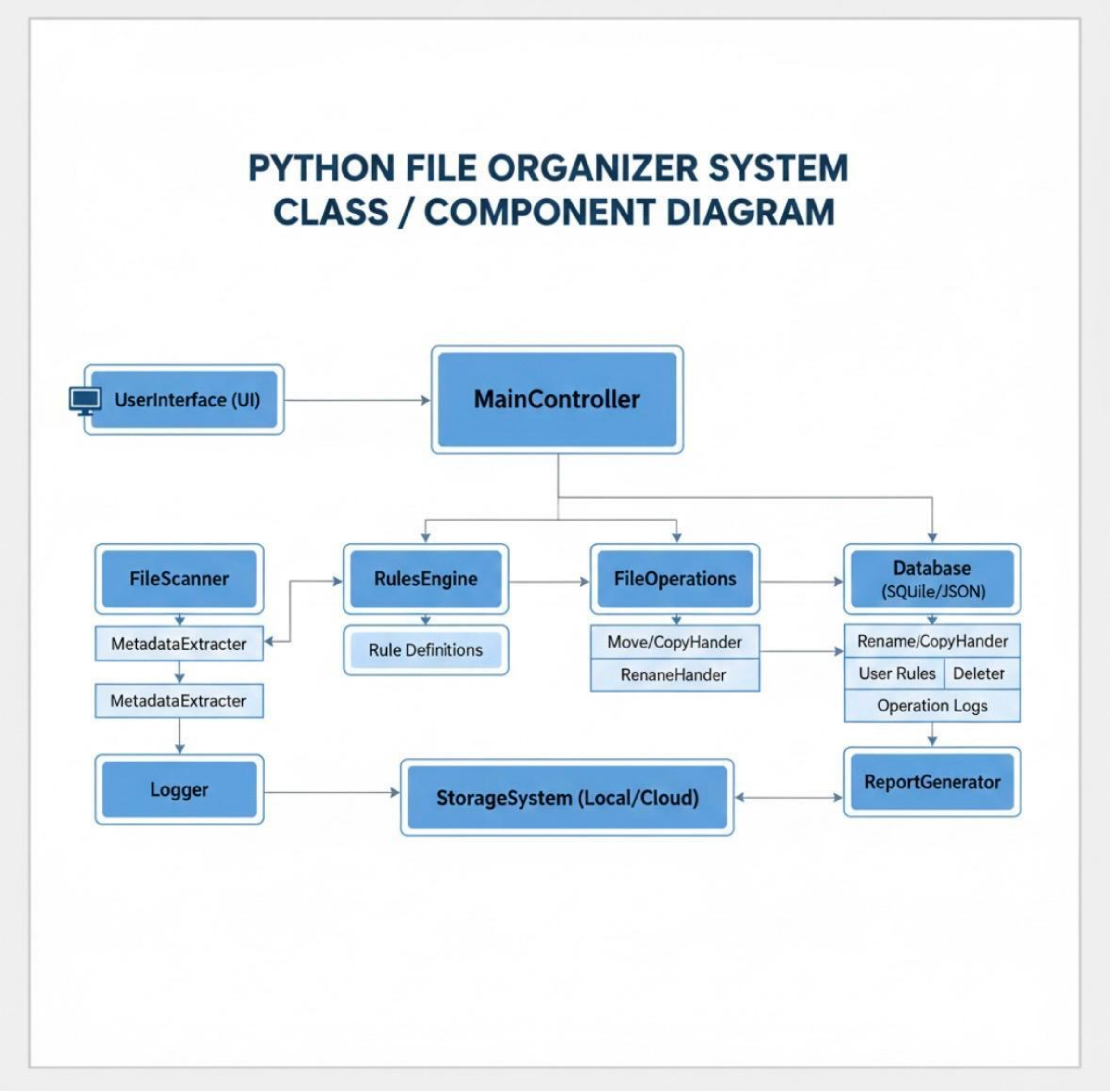
2.Work flow Diagrams



3.Sequence Diagram


PYTHON FILE ORGANIZER SYSTEM SEQUENCE DIAGRAM
User
UI
Main Controller
FileוController
RulesEngine
FileOperations
Database
1. initiaite Scan(directory)
return fileList
3. scan Dire:tory(directory)
find files
find files
apply Engine
File Operations Module
Database
return folderPath, action
5. morform Operation(file, folderPah, action
move/rename/ actio/tag, status
6. lcup/Restore Data
display Status(status)
Storage System (Local/Cloud)

4.Class/Component Diagram



Design Decisions & Rationale

- Python chosen because of its easy file-handling modules like os and shutil.
- Using extension-based sorting because it allows universal categorization.
- Chose a simple script approach instead of GUI to keep it lightweight.
- Avoided recursion to prevent accidental movement of subfolders.

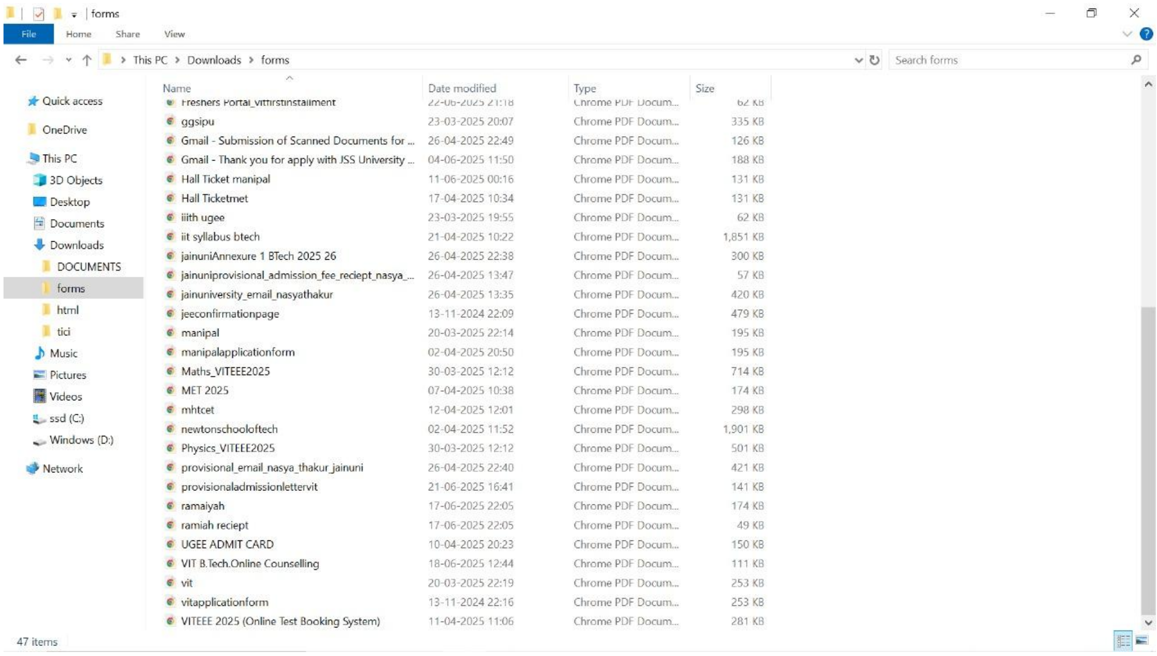
Implementation Details

```
import os

import shutil

path = input("Enter Path : ")
files = os.listdir(path)

for file in files:
    filename, extension = os.path.splitext(file)
    extension = extension[1:]


    if extension == '':
        continue


    if not os.path.exists(os.path.join(path, extension)):
        os.makedirs(os.path.join(path, extension))


    shutil.move(
     os.path.join(path, file),
         os.path.join(path, extension, file)



print("Files sorted successfully!")
```

[redacted]

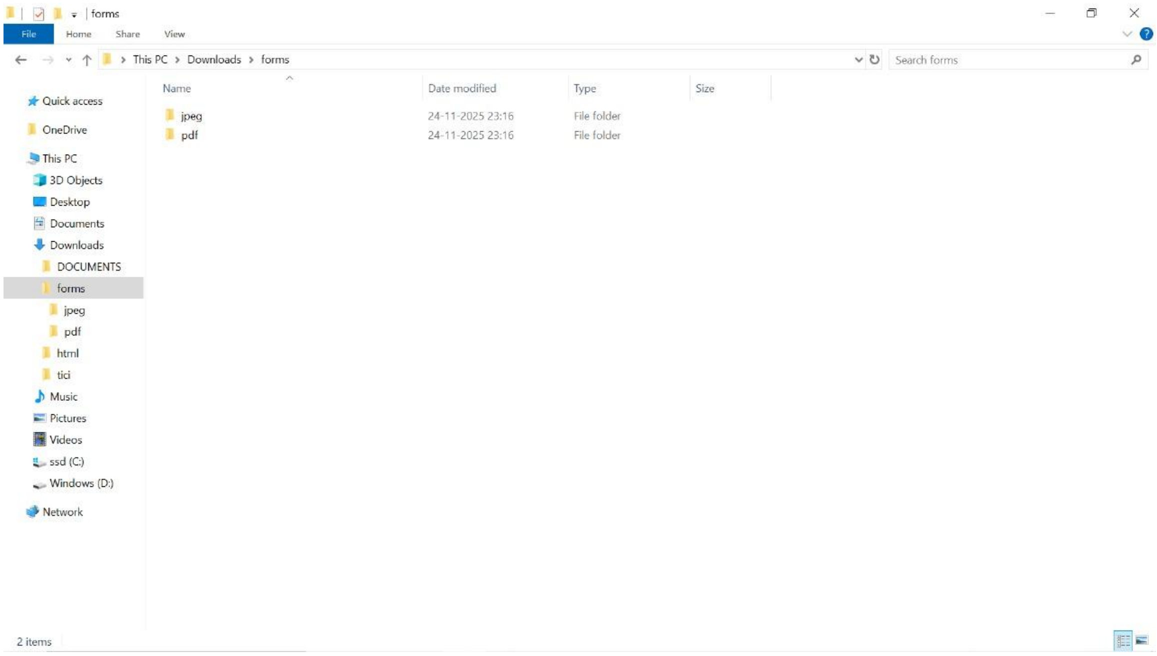
Screenshots / Results

BEFORE:



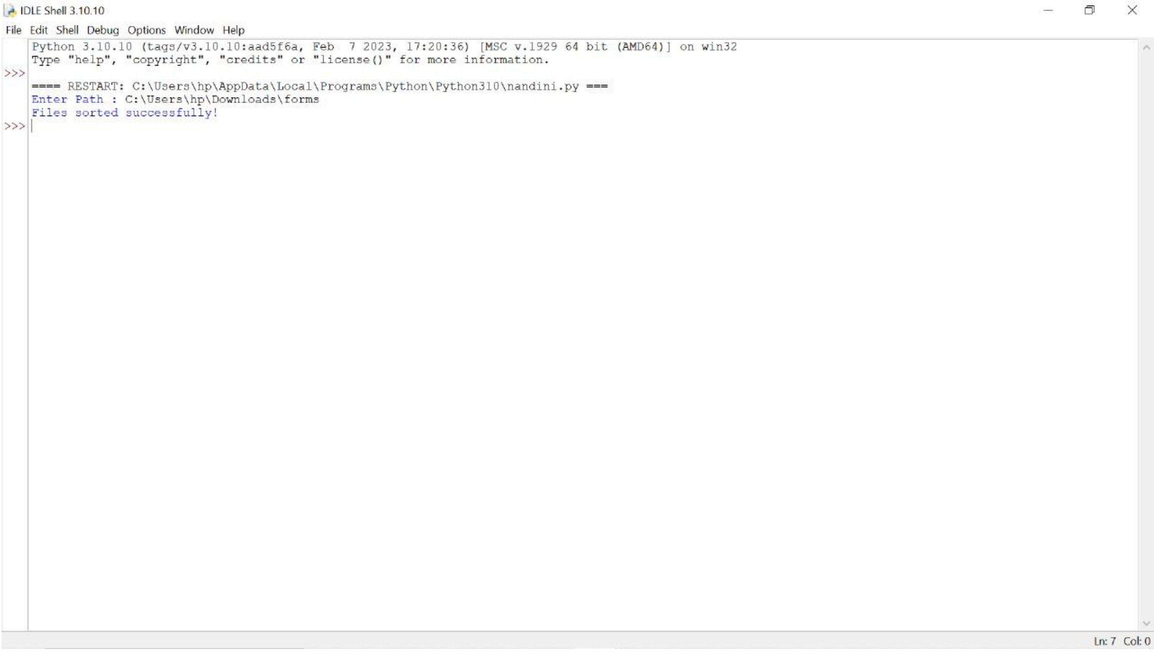
Name	Date modified	Type	Size
Freshers Portal_vitfirstinstallment	22-06-2025 21:18	Chrome PDF Docum...	62 KB
ggsipu	23-03-2025 20:07	Chrome PDF Docum...	335 KB
Gmail - Submission of Scanned Documents for ...	26-04-2025 22:49	Chrome PDF Docum...	126 KB
Gmail - Thank you for apply with JSS University ...	04-06-2025 11:50	Chrome PDF Docum...	188 KB
Hall Ticket manipal	11-06-2025 00:16	Chrome PDF Docum...	131 KB
Hall Ticketmet	17-04-2025 10:34	Chrome PDF Docum...	131 KB
iiith ugee	23-03-2025 19:55	Chrome PDF Docum...	62 KB
iit syllabus btech	21-04-2025 10:22	Chrome PDF Docum...	1,851 KB
jainuniAnnexure 1 BTech 2025 26	26-04-2025 22:38	Chrome PDF Docum...	300 KB
jainuniprovisional_admission_fee_reciept_nasya_...	26-04-2025 13:47	Chrome PDF Docum...	57 KB
jainuniversity_email_nasyathakur	26-04-2025 13:35	Chrome PDF Docum...	420 KB
jeeconfirmationpage	13-11-2024 22:09	Chrome PDF Docum...	479 KB
manipal	20-03-2025 22:14	Chrome PDF Docum...	195 KB
manipalapplicationform	02-04-2025 20:50	Chrome PDF Docum...	195 KB
Maths_VITEEE2025	30-03-2025 12:12	Chrome PDF Docum...	714 KB
MET 2025	07-04-2025 10:38	Chrome PDF Docum...	174 KB
mhtcet	12-04-2025 12:01	Chrome PDF Docum...	298 KB
newtonschooloftech	02-04-2025 11:52	Chrome PDF Docum...	1,901 KB
Physics_VITEEE2025	30-03-2025 12:12	Chrome PDF Docum...	501 KB
provisional_email_nasya_thakur_jainuni	26-04-2025 22:40	Chrome PDF Docum...	421 KB
provisionaladmissionlettervit	21-06-2025 16:41	Chrome PDF Docum...	141 KB
ramaiyah	17-06-2025 22:05	Chrome PDF Docum...	174 KB
ramiah reciept	17-06-2025 22:05	Chrome PDF Docum...	49 KB
UGEE ADMIT CARD	10-04-2025 20:23	Chrome PDF Docum...	150 KB
VIT B.Tech.Online Counselling	18-06-2025 12:44	Chrome PDF Docum...	111 KB
vit	20-03-2025 22:19	Chrome PDF Docum...	253 KB
vitapplicationform	13-11-2024 22:16	Chrome PDF Docum...	253 KB
VITEEE 2025 (Online Test Booking System)	11-04-2025 11:06	Chrome PDF Docum...	281 KB

AFTER:



Name	Date modified	Type	Size
jpeg	24-11-2025 23:16	File folder	
pdf	24-11-2025 23:16	File folder	

OUTPUT:



```
    Python 3.10.10 (tags/v3.10.10:aad5f6a, Feb  7 2023, 17:20:36) [MSC v.1929 64 bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    ==== RESTART: C:\Users\hp\AppData\Local\Programs\Python\Python310\nandini.py ===
    Enter Path : C:\Users\hp\Downloads\forms
    Files sorted successfully!
>>>
```

Testing Approach

- Tested using sample folders containing mixed file types
- Tested with empty folders
- Tested with unknown/no extension files
- Verified folders are created correctly
- Ensured files are not overwritten

Challenges Faced

- Handling files without extensions
- Avoiding movement of directories
- Ensuring the program didn’t break when encountering special characters
- Understanding how shutil.move behaves across OS

Learnings & Key Takeaways

- Learned how to work with the os and shutil modules
- Understood directory traversal
- Gained experience in writing automation scripts
- Improved debugging and testing skills
- Experienced the usefulness of automation in daily workflow

Future Enhancements

- GUI interface using Tkinter
- Option for user-defined categories
- Recursive sorting inside subfolders
- Undo or log functionality
- Sorting based on file size or creation date

References

- Python Official Documentation
- W3Schools – Python File Handling
- GeeksforGeeks – OS and Shutil modules
- StackOverflow discussions