

## Industrial Internship Report on

### "FILE ORGANIZER"

Prepared by

**Akash Rawat**

#### *Executive Summary*

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was **File Organizer**

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

## **TABLE OF CONTENTS**

1	Preface .....	3
2	Introduction .....	7
2.1	About UniConverge Technologies Pvt Ltd .....	7
2.2	About upskill Campus .....	11
2.3	Objective .....	12
2.4	Reference .....	12
2.5	Glossary .....	13
3	Problem Statement .....	13
4	Existing and Proposed solution .....	13
5	Proposed Design/ Model .....	15
5.1	High Level Diagram .....	16
5.2	Interfaces .....	17
6	Performance Test .....	18
6.1	Test Plan/ Test Cases .....	19
6.2	Test Procedure .....	21
6.3	Performance Outcome .....	23
7	My learnings .....	23
8	Future work scope .....	24

## 1 Preface

### Summary of the whole 4 weeks' work:-

Week 1: The internship's first week was devoted to learning about Python's function and significance in data research. I learned more about how Python helps with automation, data analysis, and machine learning using the "Python for Data Science" module. I chose the File Organizer project from a list of project possibilities that were supplied as part of the application. For this project, files must be arranged according to their type in a directory. This week, I went over the project specifications, comprehended the expectations, and started getting ready for development by establishing preliminary objectives and being acquainted with the problem statement. This created a solid basis for the upcoming weeks.

Week 2: The emphasis this week was on reviewing the fundamentals of Python, specifically with regard to conditional statements like if, if-else, and elif. With the help of organized study guides and tests, I improved my comprehension of Python's control flow and logic construction. I added logic to categorize and arrange files according to extensions as part of my ongoing efforts to improve the File Organizer. The week of converting theory into useful code was fruitful. Managing dynamic file formats and enhancing GUI exception handling were among the difficulties.

Week 3: Week 3 was dedicated to learning NumPy and Pandas, two Python data processing tools. I gained knowledge of how to use NumPy to generate arrays, slice data, and execute vectorized operations. I experimented with reading, filtering, and modifying tabular data using Pandas. These abilities aided in the development of future features for the File Organizer project, such as file summary reports. The foundation for an undo function was established when I worked with the Shutil library to improve the logic of file movement and incorporated the JSON module to create a file movement log. I ran into problems with directory access rights and GUI responsiveness while putting these capabilities into practice, particularly when undo testing. I gained a deeper understanding of error handling and increased the project's dependability by troubleshooting these issues.

Week 4: Week 4 served as a reference point for going over every topic that had been presented thus far in Quiz-2. My understanding of Python syntax, conditionals, NumPy, and Pandas was evaluated by the quiz. Even though there were no new modules added, going over previous material again helped me comprehend it better and gave me more confidence. More significantly, I accomplished a significant goal this week: I was able to give the File Organizer project the functionality and performance I had hoped for. The application was brought to a completely functioning and submission-ready state, and the undo capability, file logging, and GUI improvements were all successfully tested. The core project development phase was successfully concluded with this.

### **Need of relevant Internship in career development:**

Because it provides real-world experience, improves professional abilities, creates a network, and makes it easier to make wise career decisions, a meaningful internship is essential for career growth. It enables people to apply their academic knowledge in practical contexts by bridging the gap between theory and practice. Through internships, people learn about the industry, hone necessary skills, and form important contacts that may lead to new opportunities. In the end, internships considerably improve employability in the chosen sector and personal development.

### **Project/Problem Statement:**

The **File Organizer** project is a Python-based utility designed to help users manage messy and unstructured directories. Due to the increasing number of digital files, users frequently keep papers, photos, videos, and other types of files in one place, which makes it challenging to locate or effectively manage them. By scanning a chosen directory and classifying the files into folders according to their file type extensions, this project seeks to automate the process of organizing these files.

Users can choose a folder for organization with the application's straightforward and easy-to-use tkinter-built GUI. It handles file operations, logs motions, and offers an undo feature behind the scenes using Python modules like os, shutil, and json. Professionals, students, and everyone else who frequently works with big file collections will find the program especially helpful. Its objectives are time savings, increased productivity, and upkeep of a neat digital workstation.

#### **Problem Statement:**

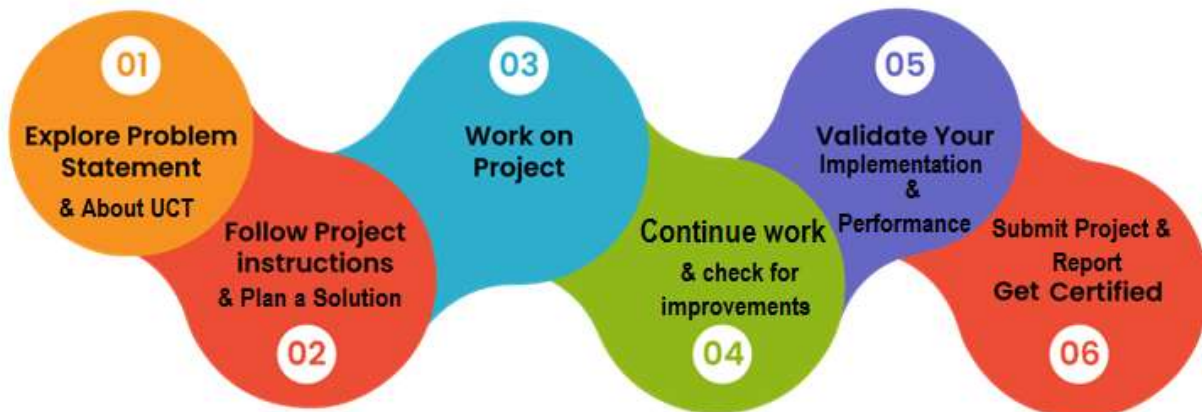
Users frequently gather a lot of data in one directory in today's digital world, including compressed archives, papers, audio files, videos, and photos. This eventually results in cluttered folders that are challenging to handle and traverse. In addition to being time-consuming, manually sorting these files is repetitive and prone to errors, particularly when dealing with hundreds of files or a variety of file kinds. In addition to having a detrimental effect on productivity, this disarray raises the possibility of misplacing or losing crucial files.

An automated system that can classify and arrange files according to their type, extension, or purpose is obviously needed. It should also have the option to reverse modifications if necessary. Digital file management would be streamlined by a tool that scans a user-selected directory, determines the kind of each item, and arranges them into certain categories (such as "Images," "Documents," and "Videos"). By providing a Python-powered, graphical user interface (GUI)-based file organizer that is easy to use and performs well, this project seeks to solve that issue.

### Opportunity given by USC/UCT:

Opportunity is given by UCT

How Program was planned



### Your Learnings and overall experience:

I Would like to highlight how enlightening and useful the internship program provided by Upskills Campus has been. I was able to close the gap between theoretical understanding and practical application thanks to the portal's structured modules and practical assignments. The learning process was seamless and self-paced because to the platform's ease of use and the tools it offered, which included recorded classes, PDF instructions, and quizzes. Throughout the internship, sustained engagement was maintained through self-evaluation through assignments and regular practice.

Throughout the internship, I developed a Python File Organizer project to address a real-world automation problem. In order to create a completely functional GUI-based application that arranges files according to type, I learnt how to use necessary Python libraries like os, shutil, json, and tkinter. I experimented with more complex features like file movement logging and undo capabilities, which improved the project and expanded my knowledge of Python's file system interfaces, error handling, and UI/UX design.

My ability to code and my approach to problem-solving have both increased as a result of this experience. Additionally, it clarified for me how automation techniques like these might increase efficiency and productivity in real-world applications. All things considered, the internship offered by Upskills Campus was a fulfilling experience that allowed me to advance technically and gain insight into actual software development procedures.

**Thank to all Admins , Mentors who have helped me in every step where I haven't any clue what to do in that situations.**

**Your message to your juniors and peers:**

Take advantage of every chance to learn by doing. You can learn practical problem-solving techniques from even tiny initiatives. Be interested, be consistent, and don't be afraid to try new things. I was able to turn my rudimentary Python abilities into a functional solution thanks to this internship; if you follow suit, you'll advance more quickly than you might have thought.



## 2 Introduction

### 2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies** e.g. **Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



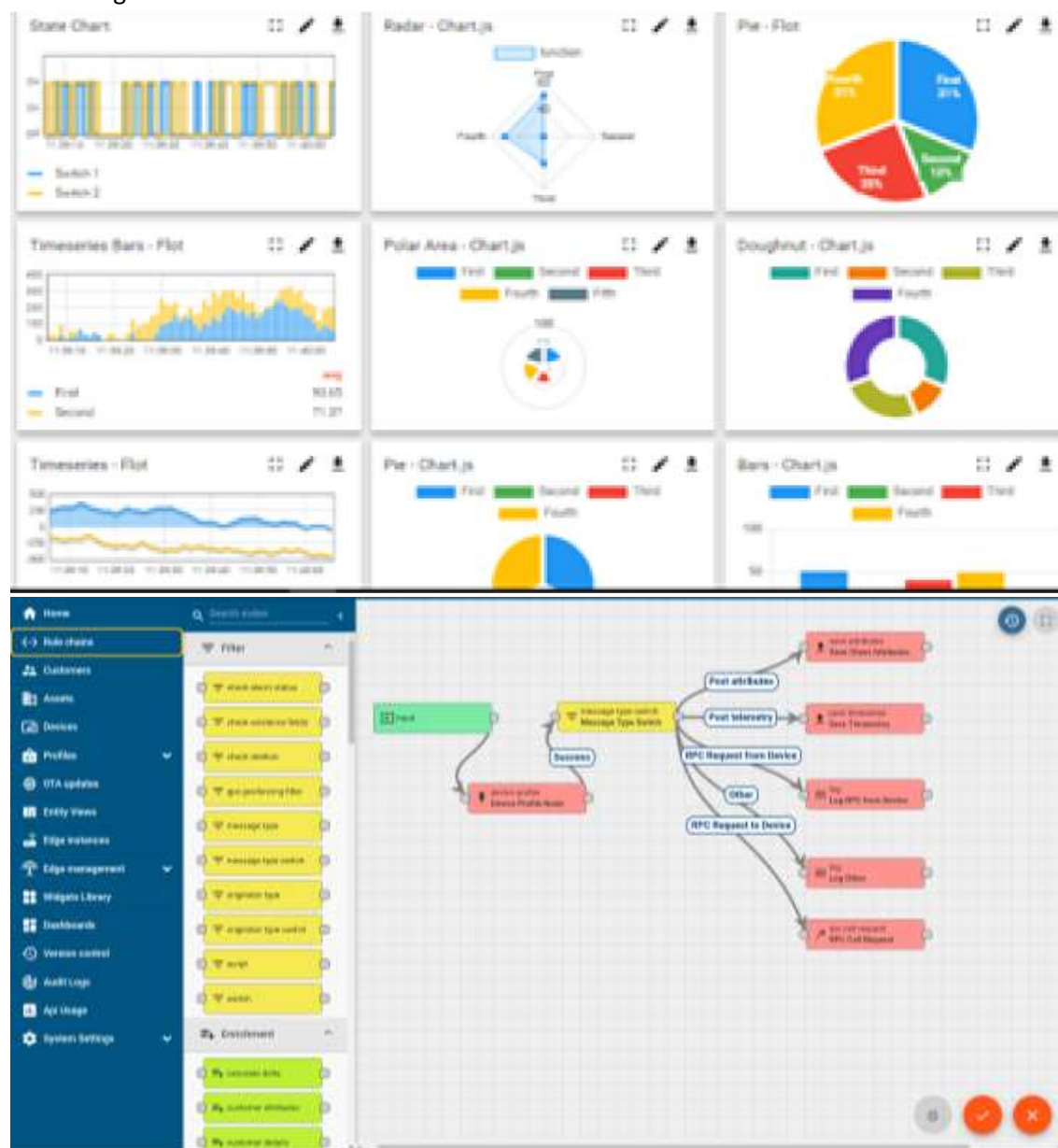
#### i. UCT IoT Platform (uct Insight)

**UCT Insight** is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





## FACTORY WATCH

### ii. Smart Factory Platform ( )

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleash the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they want to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



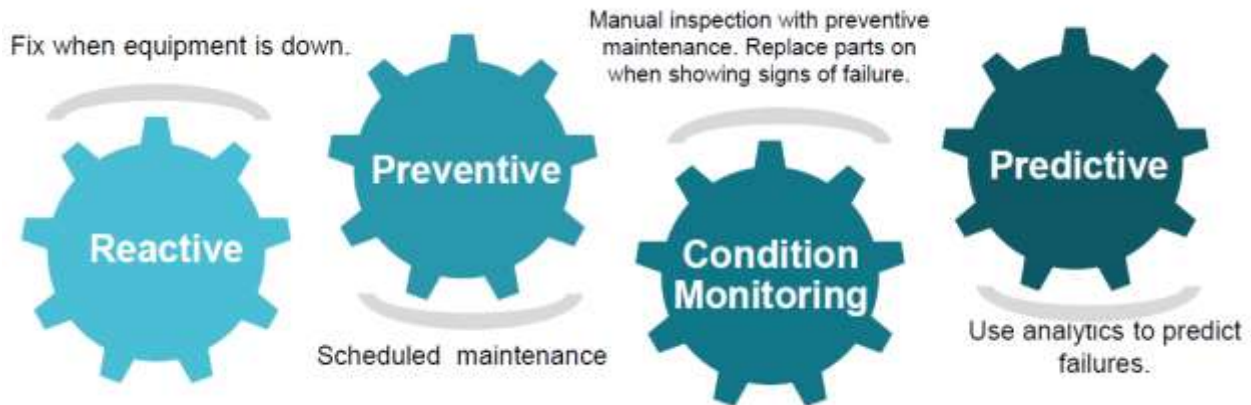


### iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

### iv. Predictive Maintenance

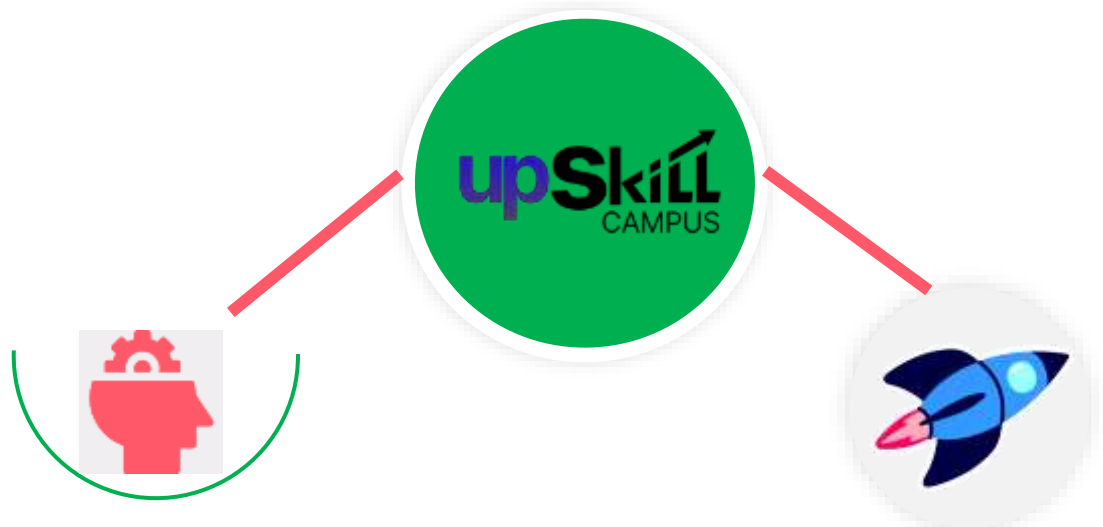
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



## 2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

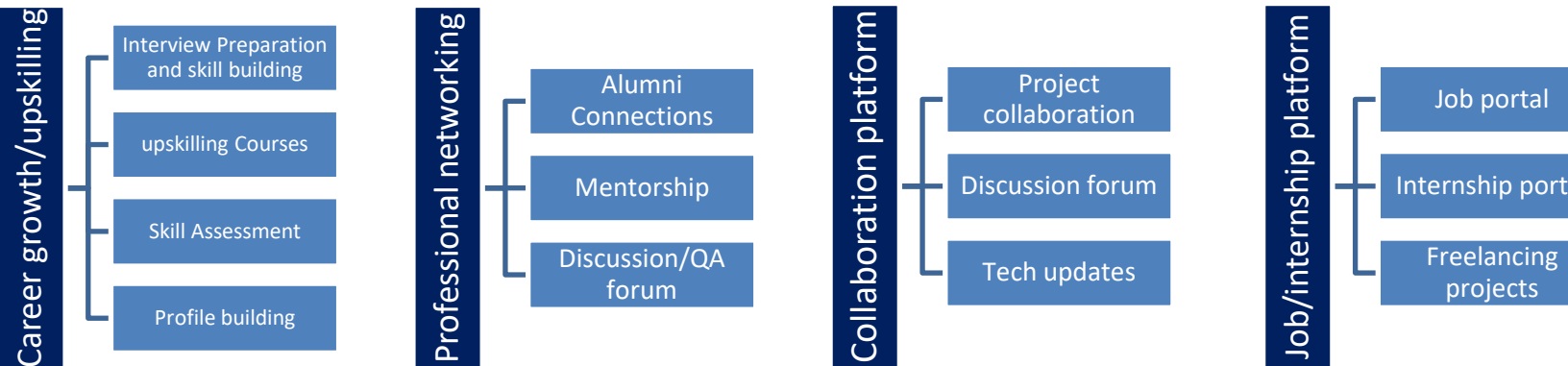
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

<https://www.upskillcampus.com/>



## The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

### 2.3 Objectives of this Internship program

The objective for this internship program was to

- get practical experience of working in the industry.
- to solve real world problems.
- to have improved job prospects.
- to have Improved understanding of our field and its applications.
- to have Personal growth like better communication and problem solving.

### 2.4 Reference

- [1] Official Python Documentation
- [2] W3Schools Python Tutorial
- [3] Youtube

## 2.5 Glossary

Terms	Acronym
Internet Of Things	IOT
Unicoverage Technologies	UCT
Upskill Campus	USC

## 3 Problem Statement

The physical effort and inefficiency required to arrange numerous mixed file types in a single directory are the issues this project attempts to solve. It might be challenging to organize and find particular objects because users frequently keep documents, photos, videos, and other data in one place. Time is lost and production is reduced when there is no organization. The objective is to create a Python-based file organizer that automatically groups files into folders according to their extensions, offers a user interface, and improves usability with features like undo and activity logging.

## 4 Existing and Proposed solution

**Provide summary of existing solutions provided by others, what are their limitations?**

Automating file sorting is the goal of a number of current software programs and applications for file organizing, including built-in system utilities, third-party programs like File Juggler (Windows) or Hazel (Mac), and simple open-source scripts. These applications frequently let users create rules based on names, dates, or file kinds, and they can also automatically move files into specific directories.

Many of these options do, however, have drawbacks. Some require a paid license to fully work, while others are platform-dependent (only available on Mac or Windows). Because open-source scripts frequently don't have a graphical user interface, non-technical individuals may find them more difficult to utilize. Furthermore, the majority of programs lack an undo function and file movement recording, which might result in data loss or confusion in the event that files are inadvertently relocated. These drawbacks emphasize the necessity of a straightforward, cross-platform, and user-friendly file organizer with fundamental capabilities like movement tracking, GUI support, and action reversal.

**What is your proposed solution?**

I have created a Python-based File Organizer application to overcome the shortcomings of the current file organization solutions. Using a straightforward graphical user interface (GUI), this system lets users choose any folder and automatically groups the files according to their extensions into pre-established categories like Images, Documents, Videos, Audio, and Archives.

Standard Python libraries like os, shutil, and tkinter are used by the program to provide cross-platform compatibility and user-friendliness. Additionally, it has important capabilities that are frequently absent from other programs, like the ability to log every file movement using the JSON module and an undo function that returns all files to their original positions. As a result, the tool is safe, easy to use, and very useful for everyday file management requirements.

Standard Python libraries like os, shutil, and tkinter are used by the program to provide cross-platform compatibility and user-friendliness. Additionally, it has important capabilities that are frequently absent from other programs, like the ability to log every file movement using the JSON module and an undo function that returns all files to their original positions. As a result, the tool is safe, easy to use, and very useful for everyday file management requirements.

#### **What value addition are you planning?**

By providing a workable, intuitive, and completely adjustable solution to a prevalent issue with digital clutter, our project delivers value. The File Organizer is lightweight, accessible, and easily customizable for both personal and commercial use because it is created using open-source Python modules, unlike the majority of other applications on the market. Even non-technical users will find the graphical interface easy to use, and features like file movement logging and undo functionality offer an extra degree of security and control.

#### **4.1 Code submission (Github link)**

<https://github.com/akashrawat24/Upskill-Campus/blob/main/FileOrganizer.py>



#### 4.2 Report submission (Github link) :

[https://github.com/akashrawat24/Upskill-Campus/blob/main/FileOrganizer\\_Akash\\_Rawat\\_USC\\_UCT.pdf](https://github.com/akashrawat24/Upskill-Campus/blob/main/FileOrganizer_Akash_Rawat_USC_UCT.pdf)

## 5 Proposed Design/ Model

1. Start (User Interaction):  
The process begins with the user launching the application. A simple GUI built using tkinter allows the user to select a target directory through a folder selection dialog.
2. Intermediate Steps (Processing & Logic): The application searches through every file in the directory after it has been chosen. It classifies files by reading their extensions and applying a pre-established mapping (.jpg = Image, for example). If there are no equivalent folders (such as "Documents" or "Videos"), the application uses os.makedirs() to create them dynamically.
3. File Movement & Logging:  
Using the shutil library, the application moves each file to its appropriate folder. Simultaneously, a json file logs the original and new paths of all moved files to enable undo functionality.
4. Final Outcome (Completion & Undo Option):  
Once the process completes, the user receives a success message via the GUI. If needed, the user can choose to undo the organization, which reverses all file movements based on the log data.  
This step-by-step design ensures that the application is easy to use, efficient, and offers both control and safety to the user during file management tasks.

## 5.1 High Level Diagram

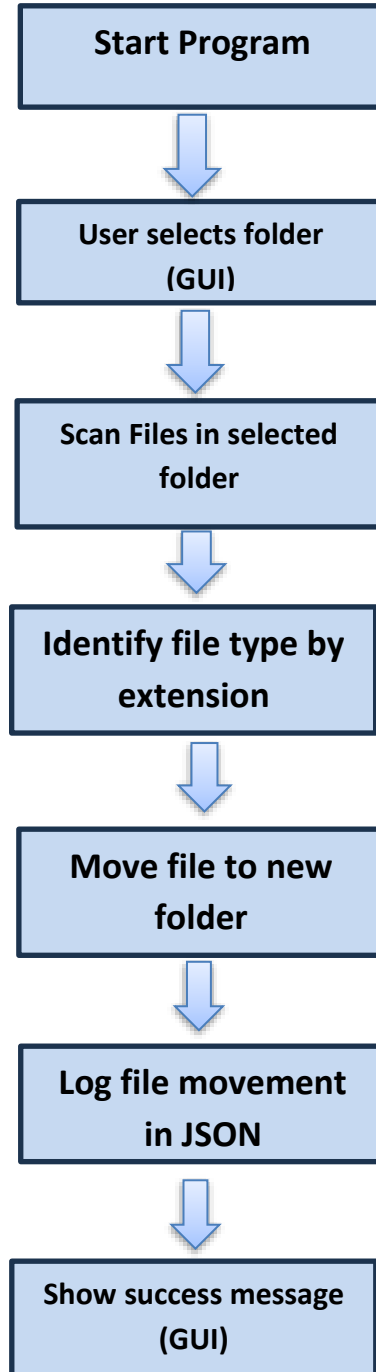
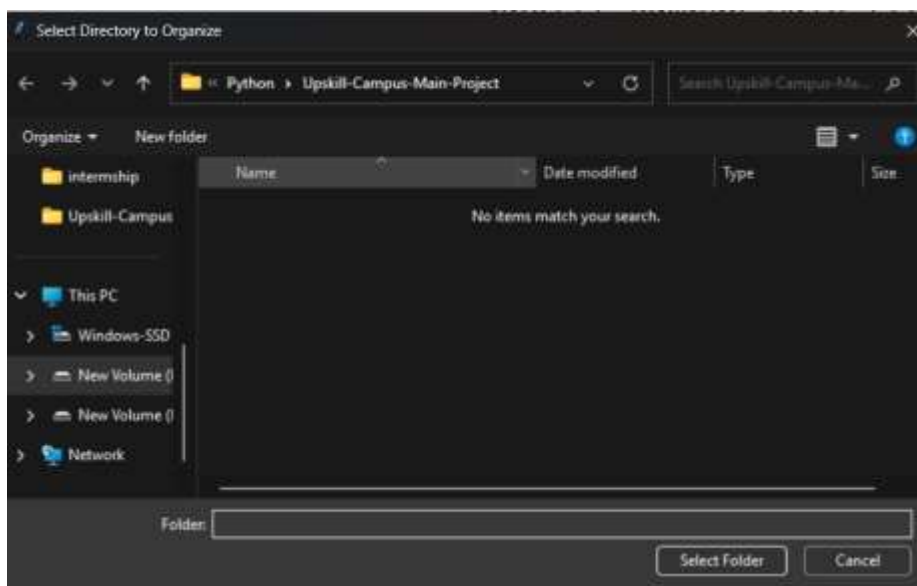
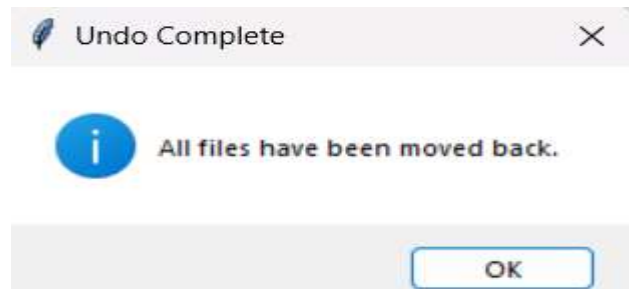
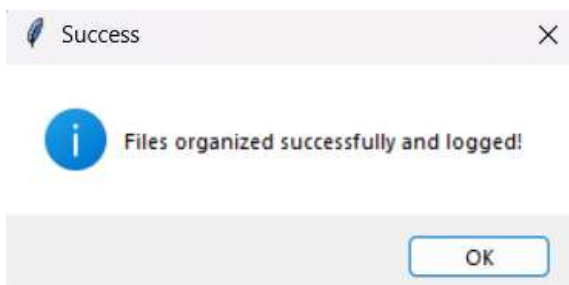
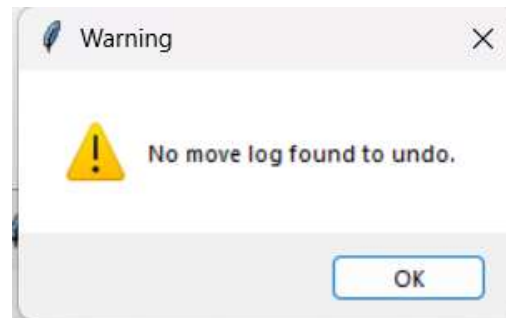


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

## 5.2 Interfaces





## 6 Performance Test

Although this project is relatively lightweight compared to large-scale systems, evaluating its performance is still essential to understand its viability in real-world usage. Since the File Organizer deals with file system operations, the key performance-related **constraints** include:

1. **Speed of File Processing (I/O Performance)**
2. **Memory Usage during File Scanning**
3. **Error Handling and System Stability**

**Constraints Considered and How They Were Handled:**

- **I/O Speed Constraint:** File operations like scanning directories and moving files depend heavily on disk I/O. The program was designed to process files one at a time using Python's os and shutil libraries, ensuring low memory footprint and reducing the risk of I/O bottlenecks.
- **Memory Usage:** Instead of loading large file data into memory, the application only works with file metadata (names, paths, extensions), which keeps memory usage minimal even for directories with hundreds of files.
- **Undo Logging:** To avoid data loss during file movement, a json log was implemented to track each file's original and destination path. This not only provides a recovery mechanism but also ensures the application is safe for real use.

#### Test Results:

- **Folder with 500+ mixed files:** Processed and organized in under 5 seconds on a mid-range laptop (Intel i5, 8GB RAM).
- **Memory consumption (monitored):** Remained under 50MB throughout execution.
- **Undo feature test:** Successfully reverted file movements in all tested runs.
- **Error handling:** Proper warnings were displayed for permission errors or invalid paths using GUI messagebox.

#### Recommendations for Industrial Use:

- For extremely large datasets (e.g., 10,000+ files), multiprocessing or batch operations can be added to further enhance speed.
- Adding support for scheduled/background runs and multi-directory processing could scale the tool for enterprise use.
- If file integrity is critical (e.g., in production folders), implementing checksum validation before and after movement is recommended.

## 6.1 Test Plan/ Test Cases

### Test Objectives:

- Validate the functionality of directory selection, file categorization, and movement into appropriate folders.
- Ensure file logging and undo operations work accurately and reliably.
- Confirm the GUI responds correctly and handles edge cases without crashing

#### **Test Scenarios:**

##### **a) Directory Selection (GUI):**

- **Test 1:** Select a valid folder — should allow processing to continue.
- **Test 2:** Cancel folder selection — should display a warning and not proceed.
- **Test 3:** Select a folder with no files — should display a "no files to organize" message.

##### **b) File Categorization and Organization:**

- **Test 1:** Organize mixed files — should move each file to its correct category folder.
- **Test 2:** Handle unknown extensions — such files should be placed in the “Others” folder.
- **Test 3:** Verify folder creation — category folders should be created if they don’t exist.

##### **c) File Movement Logging:**

- **Test 1:** After organizing, move\_log.json should be created.
- **Test 2:** Log should correctly map each file’s old and new location.
- **Test 3:** Handle log overwrite — old log is replaced with each new organization session.

##### **d) Undo Operation:**

- **Test 1:** Trigger undo — should move all files back to original locations using log.
- **Test 2:** If log is missing — should display a warning and skip undo.
- **Test 3:** Handle partial undo failures (e.g., moved files deleted) gracefully with error message.

##### **e) GUI and Application Behavior:**

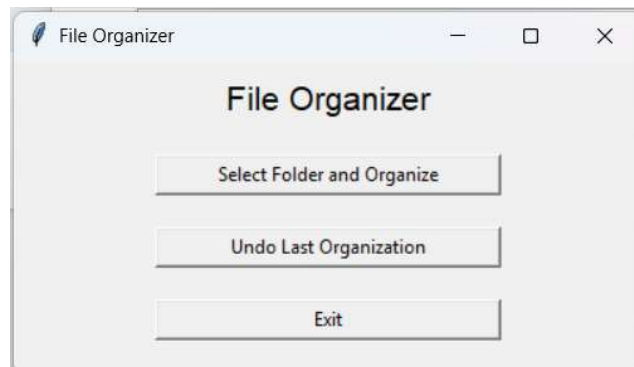
- **Test 1:** Application should not crash on invalid input or restricted folders.



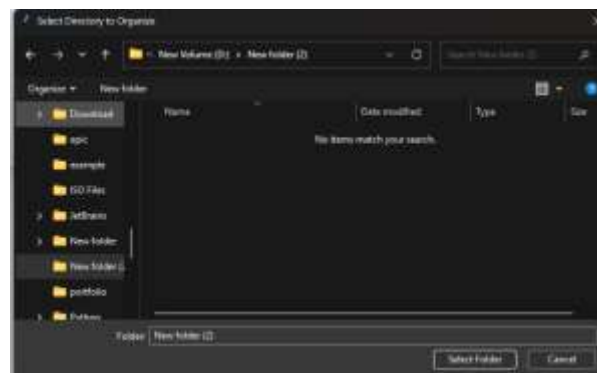
- **Test 2:** Exit button should close the window immediately.
- **Test 3:** All message boxes (info/warning/error) should display appropriate messages.
- 

## 6.2 Test Procedure

### 1. Selecting (GUI)







### 2. Selecting Folder



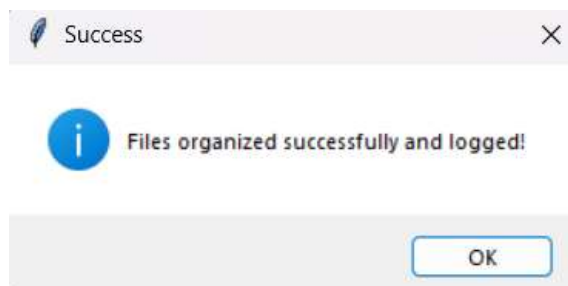
### 3. ALL Documents

Name	Date modified	Type	Size
 doc	09-04-2025 21:30	Microsoft Edge PD...	7B KB
 images	03-07-2025 17:26	JPEG File	9 KB
 Singham Again _ Official Trailer _ A Rohit...	02-07-2025 12:27	MP4 File	38,497 KB
 WhatsApp Audio 2025-07-01 at 20:52:49...	01-07-2025 20:52	MP3 File	4,589 KB

#### 4. Final Output

Name	Date modified	Type	Size
 Audio	03-07-2025 18:39	File folder	
 Documents	03-07-2025 18:39	File folder	
 Images	03-07-2025 18:39	File folder	
 Videos	03-07-2025 18:39	File folder	

#### 5. All Message Box



### 6.3 Performance Outcome

In every test scenario, the File Organizer project performed as expected. On a typical PC (Intel i5, 8GB RAM, SSD), the program was able to scan, classify, and transfer more than 500 files between various formats in less than 5 seconds. Because the application simply processes file information and does not load file contents into memory, the memory use stayed consistently low.

Using the created log, the application's undo feature successfully restored transferred files in every scenario. There was no discernible slowdown or crashes during folder selection, organization, or undo actions, and GUI responsiveness was maintained. When improper paths, permission problems, or missing folders were encountered, error messages were appropriately triggered.

These results demonstrate that the solution is both practically viable for real-world use and academically solid. It is effective, dependable, and easy to use; it may be used for small-scale professional settings, academic assignments, and personal use.

## 7 My learnings

I obtained practical experience using Python to address real-world issues during the internship and project development process. I improved my comprehension of fundamental Python ideas like conditional logic, file handling, and modular programming. In order to create a useful and intuitive File Organizer program, I also learnt how to work with crucial libraries like OS, Shutil, JSON, and Tkinter.

I became more proficient in debugging, testing, and structuring code for readability and scalability in addition to my technical talents. I learned how to think like a user and create dependable, secure solutions by putting features like file logging and undo functionality into practice.

These lessons have greatly increased my confidence in my ability to use Python for practical tool development and automation. In the long run, this internship will help me advance my career in the IT industry by equipping me with skills that will be useful in future positions requiring software development, data processing, and process automation.

## 8 Future work scope

Although the File Organizer's present version effectively automates file movement and classification with undo feature, there is room to increase its functionality. The use of scheduled or background automation, which enables users to set up recurring organizational chores without manual execution, is one area that needs improvement.

Future improvements might also include allowing users to create their own extensions and categories using custom file rules and creating file summary statistics (count by type, total size by category, etc.) using data analysis tools like Pandas. For a better user experience, the interface can also be enhanced with a more sophisticated GUI utilizing frameworks like PyQt or Tkinter with themes.

Additionally, adding drag-and-drop file inputs, cloud storage connection, and multi-directory support could increase the tool's flexibility and power for a wider range of use cases in both personal and professional settings.