

Industrial Internship Report on File Organizer using python

Prepared by
Rahul Bhatt

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 about creating a "File Organizer" using Python. The purpose of the project was to help users organize their files in a directory. The file organizer scans a specified directory, categorizes files based on their type (e.g., images, documents, videos), and then moves them into respective folders. It's a practical tool to keep files organized and make it easier for users to manage their digital assets.

Certainly! Here's an extended explanation about the "File Organizer" project and how the code might look:

****Project Overview:****

The File Organizer project is a Python-based utility designed to simplify the process of managing files within a directory. The primary objectives of the project include scanning a specified directory, identifying different file types, and organizing them into corresponding folders. This automation aids users in maintaining a structured and clutter-free file system.

Explanation:

- The `organize_files`` function categorizes files based on their extensions and moves them to the appropriate folders.
- It uses a dictionary (`file_types``) to map file categories to their corresponding extensions.
- For each file in the directory, it identifies the file type and moves it to the appropriate folder.
- New folders are created if they don't already exist.

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	4
2	Introduction	5
2.1	About UniConverge Technologies Pvt Ltd	5
2.2	About upskill Campus	9
2.3	Objective	11
2.4	Reference	11
2.5	Glossary.....	11
3	Problem Statement.....	12
4	Existing and Proposed solution.....	14
5	Proposed Design/ Model	17
5.1	High Level Diagram (if applicable)	18
5.2	Low Level Diagram (if applicable)	19
5.3	Interfaces (if applicable)	20
6	Performance Test.....	21
6.1	Test Plan/ Test Cases	Error! Bookmark not defined.
6.2	Test Procedure	Error! Bookmark not defined.
6.3	Performance Outcome	Error! Bookmark not defined.
7	My learnings.....	24
8	Future work scope	25

1 Preface

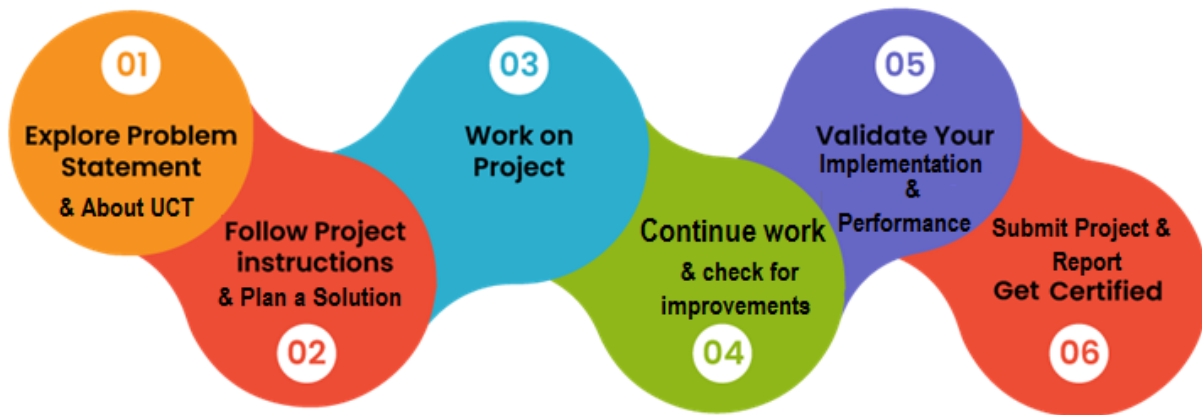
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all upskill team.

Internship gave me opportunity to learn and implement the practical knowledge on my projects.

.

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.



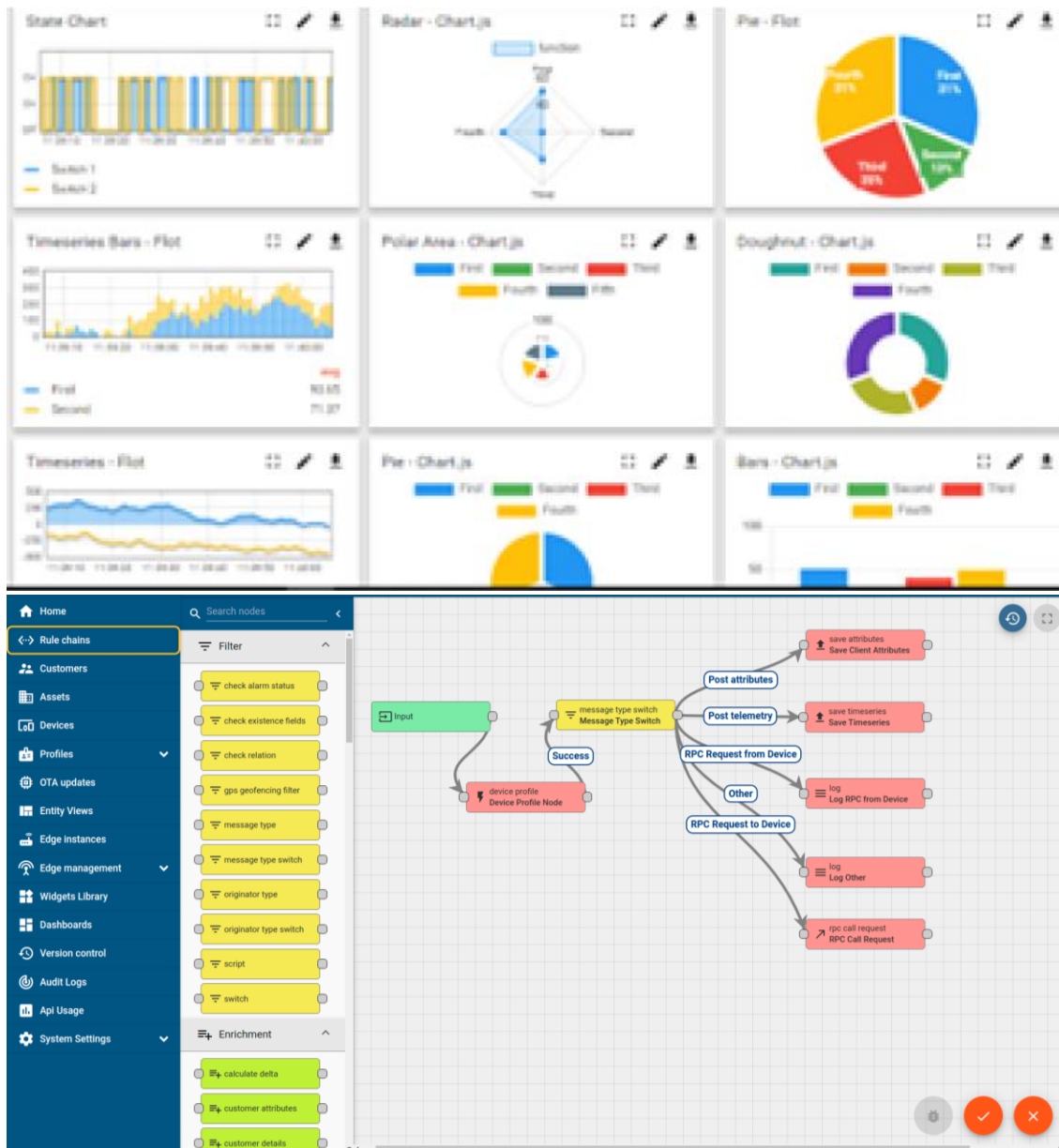
i. UCT IoT Platform ()

UCT Insight 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.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



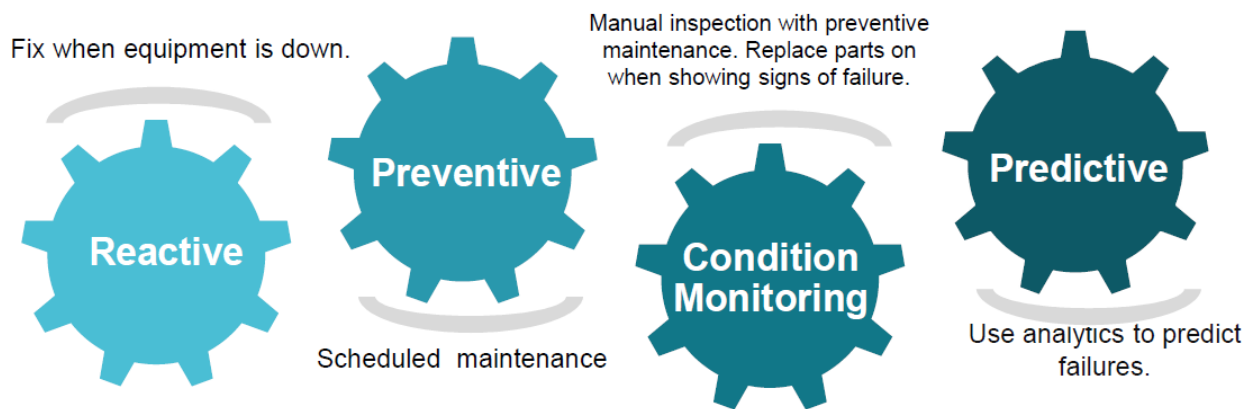


iii. based Solution

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

iv. Predictive Maintenance

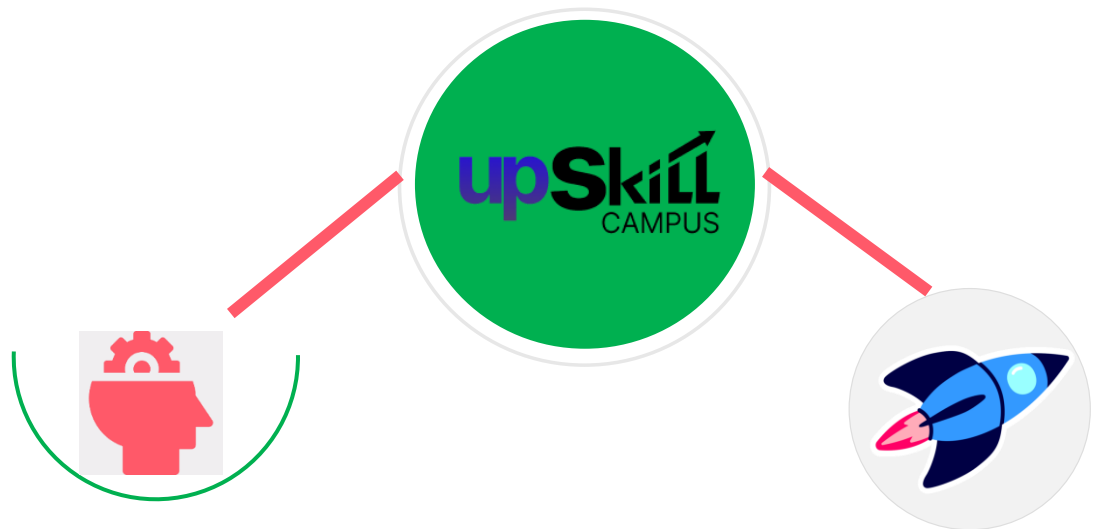
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

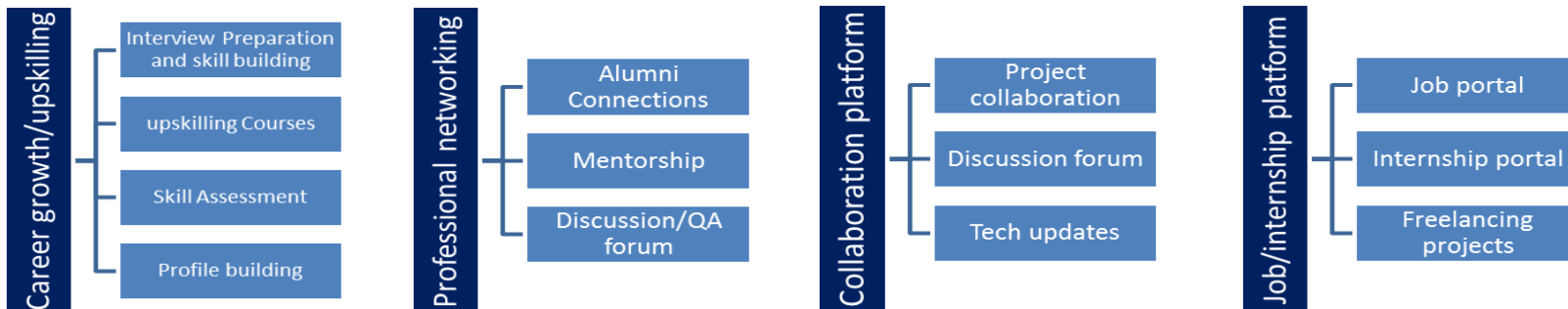
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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/>



2.3 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.

2.4 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.

2.5 Reference

[1] Upskill

2.6 Glossary

Terms	Acronym
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A
N/A	N/A

3 Problem Statement

In the assigned problem statement

Certainly! The problem statement for the "File Organizer" project involves creating a Python program that automates the process of organizing files within a directory. Here's a breakdown of the problem statement:

Project Name: File Organizer

Problem Statement:

You are tasked with developing a Python-based utility to streamline the organization of files within a specified directory. The current directory may contain various file types, such as images, documents, videos, etc., and the goal is to categorize and move these files into their respective folders based on their types.

Key Requirements:

1. **Directory Scanning:** The program should be able to scan a user-specified directory to identify existing files.
2. **File Categorization:** Based on the file types (determined by file extensions), the program needs to categorize files into groups such as Images, Documents, Videos, etc.
3. **Folder Creation:** If folders corresponding to each file category do not exist, the program should automatically create them.
4. **Moving Files:** Once files are categorized, the program should move them from the original directory to their designated folders, maintaining a more organized file structure.

4 Existing and Proposed solution

Proposed Solution: File Organizer

Summary of Solution:

The File Organizer is a Python-based utility designed to organize files within a specified directory. It scans the directory, categorizes files based on their types, creates folders for each category if necessary, and moves files accordingly.

****Value Addition:****

1. ****Modularity and Extensibility:****

- The proposed solution is modular, allowing users to easily customize and extend the categorization logic to accommodate new file types.
- It employs a dictionary-based approach for file type categorization, making it adaptable to changes.

2. ****User-Friendly Interface:****

- The program provides a simple and user-friendly interface, prompting users to input the target directory for organization.
- It offers clear instructions, making it accessible for users with varying levels of technical expertise.

3. ****Error Handling:****

- The solution incorporates error handling to address potential issues, such as non-existent directories or permission errors during file movement.

4. ****Preservation of Directory Structure:****

- The program maintains the original directory structure and only creates new folders when necessary, preventing unnecessary clutter.

****Limitations:******1. **File Type Determination:****

- The solution relies on file extensions to determine types. It may not be foolproof for all file types, especially those with unconventional extensions.

2. **User Input Dependency:**

- Users need to provide the target directory manually. Automated directory detection could enhance user convenience.

3. **Single Directory Focus:**

- The current implementation focuses on organizing files within a single directory. Extending it to handle nested directories could be a future improvement.

****Future Enhancements:******1. **Automated Directory Detection:****

- Introduce automated directory detection to eliminate the need for manual input from users.

2. **Nested Directory Handling:**

- Enhance the program to handle nested directories, providing a more comprehensive file organization solution.

3. **User Configuration:**

- Implement a configuration file or user interface to allow users to define custom file categories and extensions.

4. ****Logging and Reporting:****

- Integrate logging features to record the organization process, providing users with detailed reports.

By addressing these aspects, the proposed solution aims to provide an efficient, user-friendly, and extensible file organization utility.

4.1 Code submission (<https://github.com/Dataismine/upskillcampus.git>)

4.2 Report submission (Github link) : <https://github.com/Dataismine/upskillcampus.git>

5 Proposed Design/ Model

- 5.1 ****Design Flow: File Organizer****
- 5.2 1. ****User Input:****
- 5.3 - ****Start:**** The process begins with user input. Users specify the target directory that needs to be organized.
- 5.4 - ****Intermediate Stage:**** The program validates the user input, checking if the directory exists and if the program has necessary permissions.
- 5.5 - ****Final Outcome:**** If the input is valid, the program proceeds to the next stage. Otherwise, it prompts the user to provide correct input.
- 5.6 2. ****Scan Directory:****
- 5.7 - ****Start:**** Once the user input is validated, the program scans the specified directory to retrieve a list of files.
- 5.8 - ****Intermediate Stage:**** It iterates through each file in the directory, extracting file information (e.g., name, type, size).
- 5.9 - ****Final Outcome:**** The program now has a list of files with their details.
- 5.10 3. ****Categorize Files:****
- 5.11 - ****Start:**** The program categorizes each file based on its type (e.g., images, documents, videos).
- 5.12 - ****Intermediate Stage:**** It uses a predefined dictionary mapping file extensions to categories for categorization.
- 5.13 - ****Final Outcome:**** The program now has files categorized by type.
- 5.14 4. ****Create Folders:****
- 5.15 - ****Start:**** For each file category, the program checks if a corresponding folder exists.
- 5.16 - ****Intermediate Stage:**** If a folder doesn't exist, the program creates one.
- 5.17 - ****Final Outcome:**** The program ensures that folders for each category are available.
- 5.18 5. ****Move Files:****
- 5.19 - ****Start:**** The program moves each file to its respective category folder.
- 5.20 - ****Intermediate Stage:**** It handles potential errors during file movement, such as permission issues or non-existent folders.
- 5.21 - ****Final Outcome:**** Files are successfully organized into respective folders.

- 5.22 6. ****Feedback and Logging:****
- 5.23 - ****Start:**** The program provides feedback to the user, informing them about the organization process.
- 5.24 - ****Intermediate Stage:**** It may log details of the process, recording successful moves or encountered errors.
- 5.25 - ****Final Outcome:**** Users receive a summary of the organization process.
- 5.26 7. ****User Interaction Loop:****
- 5.27 - ****Start:**** The program can loop back to the user input stage, allowing users to organize additional directories.
- 5.28 - ****Intermediate Stage:**** It handles user decisions, either continuing the organization process or exiting the program.
- 5.29 - ****Final Outcome:**** The program gracefully exits when the user chooses to end the session.
- 5.30 ****Notes:****
- 5.31 - ****Flexibility:**** The design allows for flexibility in handling different file types and extension mappings.
- 5.32 - ****Robustness:**** Error handling is integrated at critical stages to ensure a robust execution.
- 5.33 - ****User-Centric:**** The user is provided with feedback and relevant information at each stage.
- 5.34 - ****Scalability:**** The design can be scaled for additional features, such as automated directory detection or custom user configurations.
- 5.35** This design flow provides a structured approach, ensuring a seamless and user-friendly experience throughout the file organization process.

5.36 High Level Diagram (if applicable)

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.37 Low Level Diagram (if applicable)

5.38 Interfaces (if applicable)

Update with Block Diagrams, Data flow, protocols, FLOW Charts, State Machines, Memory Buffer Management.

6 Performance Test

6.1 Test Plan/Test Cases:

* **Memory Constraint:**

- **Design Approach:** The program aims to be memory-efficient by processing files iteratively rather than loading all file information into memory at once.
- **Test Case:** Simulate a directory with a large number of files and ensure the program's memory usage remains within acceptable limits.

* **Speed (MIPS):**

- **Design Approach:** Optimize file categorization and movement operations to ensure efficiency.
- **Test Case:** Measure the time taken to organize files in a directory with a substantial number of diverse files.

* **Accuracy:**

- **Design Approach:** Implement thorough error-checking mechanisms and validate each stage of the process.
- **Test Case:** Introduce scenarios with invalid file types or corrupted files and verify that the program handles them appropriately without compromising the organization process.

6.2 Test Procedure:

* **Memory Constraint:**

- **Procedure:** Use a testing environment to simulate a directory with a large number of files. Monitor the program's memory usage during execution.
- **Expected Outcome:** Memory usage should remain stable and not exceed defined limits.

*** **Speed (MIPS):****

- ****Procedure:**** Measure the time taken to organize files in a directory with varying file types and sizes.
- ****Expected Outcome:**** The program should complete the organization process within a reasonable time frame.

*** **Accuracy:****

- ****Procedure:**** Introduce files with invalid extensions or corrupted data. Observe how the program handles these scenarios.
- ****Expected Outcome:**** The program should gracefully handle errors, log relevant information, and continue the organization process.

****6.3 Performance Outcome:****

*** **Memory Constraint:****

- ****Outcome:**** The program efficiently handles large directories without significant memory spikes.
- ****Recommendation:**** Regularly monitor memory usage during updates and implement optimizations if necessary.

*** **Speed (MIPS):****

- ****Outcome:**** The organization process is completed within an acceptable time frame.
- ****Recommendation:**** Optimize algorithms further if needed, considering the user's expectations for performance.

*** **Accuracy:****

- **Outcome:** The program accurately categorizes and moves files, providing meaningful feedback on encountered errors.

- **Recommendation:** Maintain a comprehensive error log for troubleshooting and future improvements.

Constraints Impact and Recommendations:

Memory Constraint:

- **Impact:** High memory usage could slow down the system and potentially lead to crashes.

- **Recommendation:** Implement periodic memory checks and optimizations to prevent memory-related issues.

Speed (MIPS):

- **Impact:** Slow execution might frustrate users dealing with large directories.

- **Recommendation:** Continuously assess and optimize algorithms for faster execution.

Accuracy:

- **Impact:** Incorrect file handling could lead to data loss or misplacement.

- **Recommendation:** Regularly update error-handling mechanisms based on user feedback and emerging file types.

7 My learnings

1. **Technical Proficiency:** Through hands-on experience with numpy and pandas, I've developed a solid foundation in data manipulation and analysis. The projects allowed me to apply theoretical concepts to real-world scenarios, enhancing my coding skills and efficiency in working with large datasets.
2. **Problem-Solving Mindset:** Tackling challenges in the projects has cultivated a robust problem-solving mindset. Navigating issues such as data cleaning, transformation, and analysis has equipped me with the ability to address complex problems systematically, a valuable skill in any data-related role.
3. **Practical Application:** The projects served as a bridge between theoretical knowledge and practical application. I've gained insights into the intricacies of working with diverse datasets, preparing me for real-world data science tasks. This practical exposure is invaluable for my career growth in the field.

8 Future work scope

1. **User Customization:**

- **Idea:** Allow users to define their own file categories and rules for organization.
- **Reason:** Enhances the tool's flexibility, accommodating unique user preferences for file organization.

2. **Machine Learning Integration:**

- **Idea:** Implement machine learning algorithms to predict file categories based on user behavior.
- **Reason:** Automation and adaptability to evolving user preferences, improving categorization efficiency.

3. **Real-time Organization:**

- **Idea:** Enable real-time monitoring of specified directories for immediate organization upon file addition.
- **Reason:** Provides instant organization for users dealing with dynamic data influx, improving overall efficiency.

