Industrial Internship Report on

"File Organizer"

Prepared by

Dipesh Parihar

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 a Python-based File Organizer that automatically sorts and organizes files into specific folders based on their type, improving system organization and efficiency. It uses Python's `os` and `shutil` modules for file manipulation and handling exceptions. 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

 Summary:

Over the six weeks I created File Organizer in Python which is a program that arranges file into specified folders such as documents, images, etc. The first weeks to dedicate to the project's creation, development of the main functionality, and files categorization. Then, I included exception handling and a measure of code speed up. Lastly, the output that I delivered includes fine-tuning the user interface and testing the solution and writing the project documentation.

Need of relevant Internship in career development:

Out of all the internships that I have completed, the internship that saw the development of the Pythonbased File Organizer is the most beneficial to my career. It enabled me to utilize programming experience in a real-world context, thus refining my experience in Python, files, and relative automation. It is often fundamental in helping in the training of the problem-solving skills and in enhancing the understanding of how to build simple but effective, practical solutions. Moreover, it has enabled me to develop project management skills from planning to implementation process and has added value to my profile toward future employers to show that I am capable of delivering functional projects.

Brief about Your project/problem statement:

My project was to build a Python Application for organizing files thereby creating categories to sort the files to specific folders like documents, images, and videos. The problem statement focused on disorganized storage systems of files whereby locating for a file was a difficult process among users. The idea of the project was to increase the system organization, save users' time spent on navigating through folders and managing files via drag and drop technology.

Opportunity given by USC/UCT:

The situation whereby USC/UCT talent provided me with real life project working experience that was useful to my learning and carrier was occasioned by. It provided opportunity to use acquired theoretical knowledge in the real-life situations with regards to programming with Python and management of files. I also have acquired so many skills such as both practical and technical skills including; problem-solving skills, coding skills, project implementation skills and many more. However, it acquainted me with the benchmark at the industry level and so it helped me to develop personality as well as character apart from enhancing my professional persona by getting ready for other positions in the future related to the software and technology.

How Program was planned:

These Works developed over six weeks whereby the initial stage involved identifying goals and designing the context of the program. Some of the development that went core included file sorting and management. Likewise, error handling was incorporated, then optimization and lastly improving on the user interface. Last but not the least, testing phase was done effectively and the documentation part was also done perfectly.
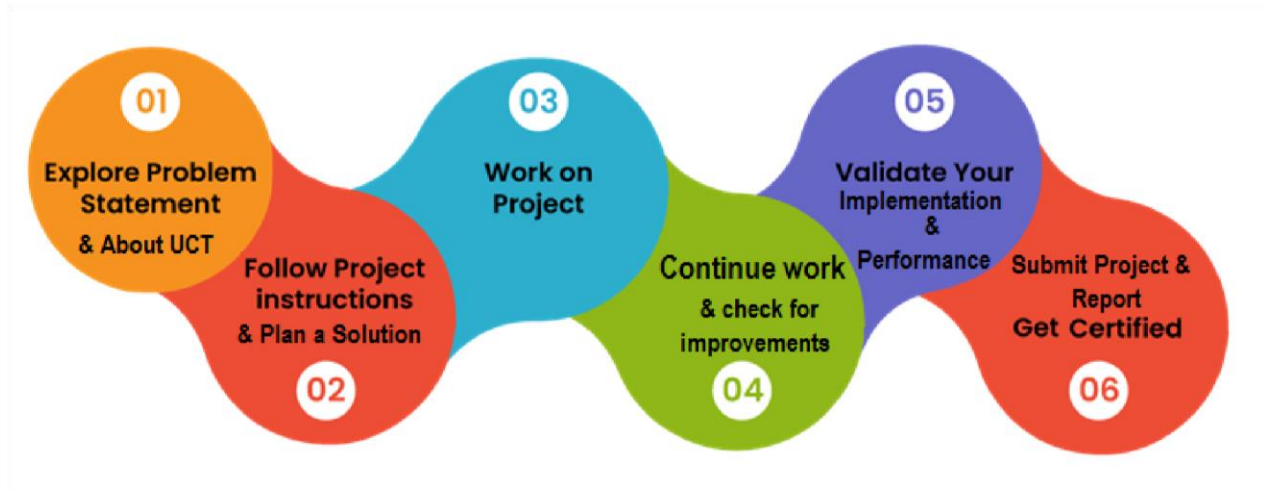
Learnings and overall experience:

Python programming together with the handling of files has been quite helpful in the whole project. Key learnings include:

- Practical Application: Kept abreast with theoretical knowledge to solve practical problems improving on coding and problem-solving domain.

- Project Management: Acquired the experience in the overall project planning and management alongside the proper execution and fine tuning of the same.

- Error Handling: Enhanced my capacity to manage exception and deal with problems perfectly.

- User Interface: Acquired knowledge on designing usable interfaces with considering options to incorporate in the interface.

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
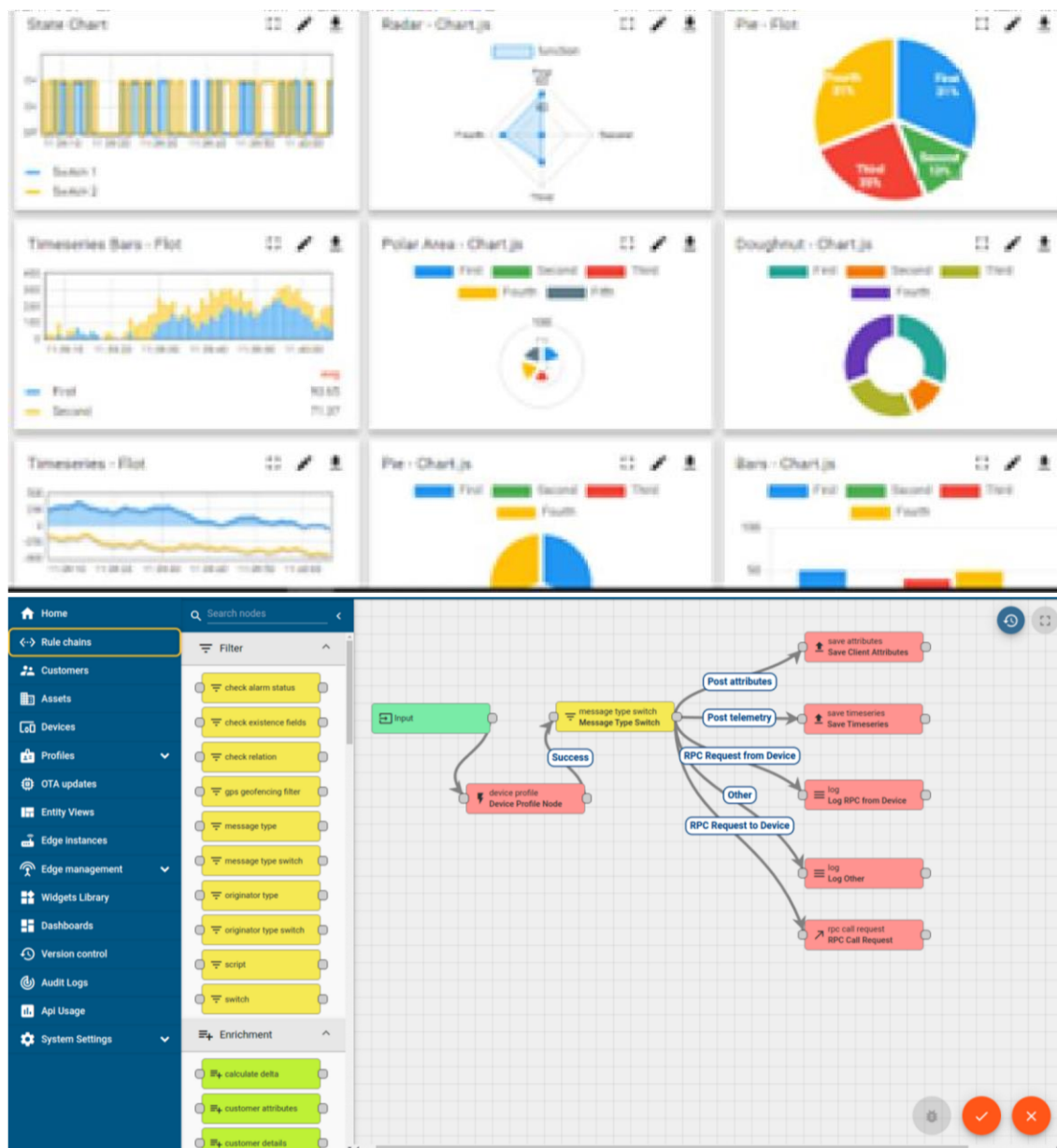
ii.   Smart Factory Platform (**FACTORY WATCH**)

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 unleased 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 what 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.
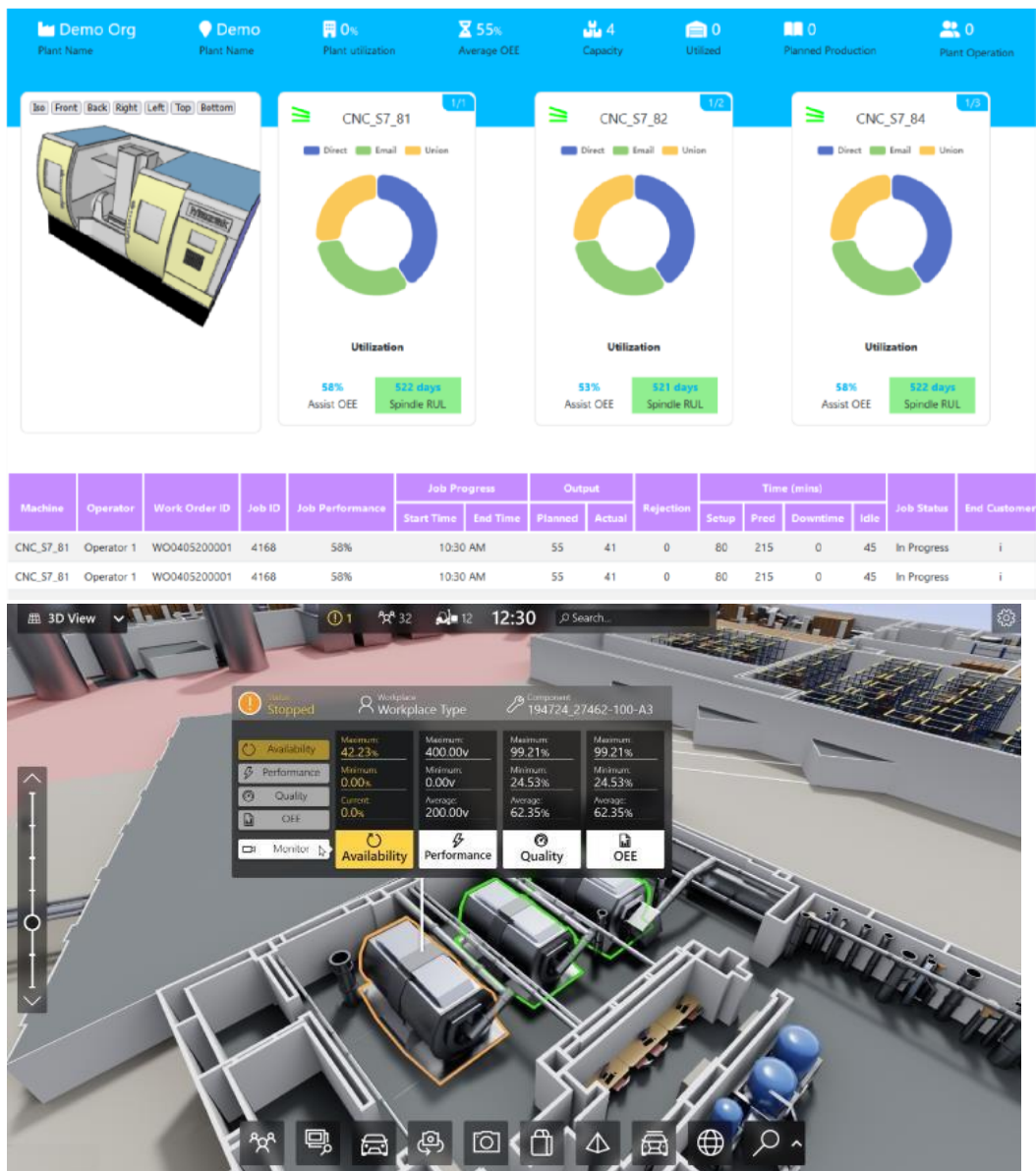
iii. LoRaWAN™ based Solution

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.
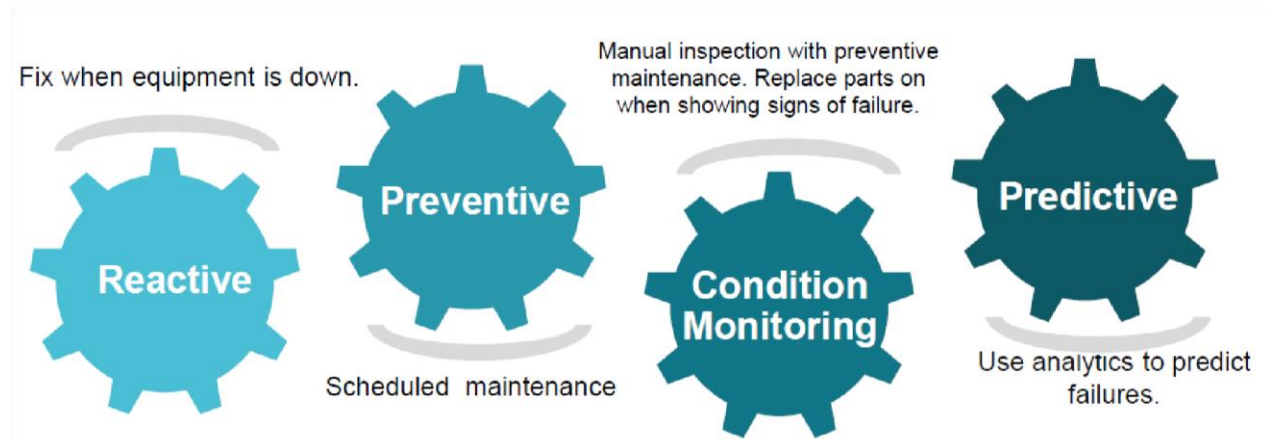
2.2 About upskill Campus (USC)

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.
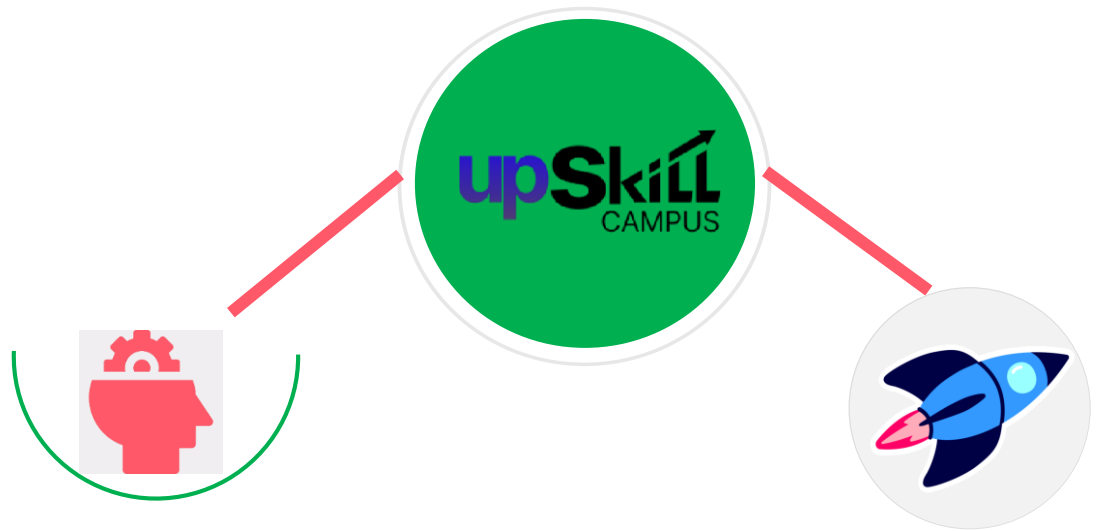
Seeing need of upskilling in self upSkill Campus aiming paced manner along-with
to upskill 1 million additional support services e.g. learners in next 5 year
Internship, projects, interaction with

Industry experts, Career

growth Services

https://www.upskillcampus.com/
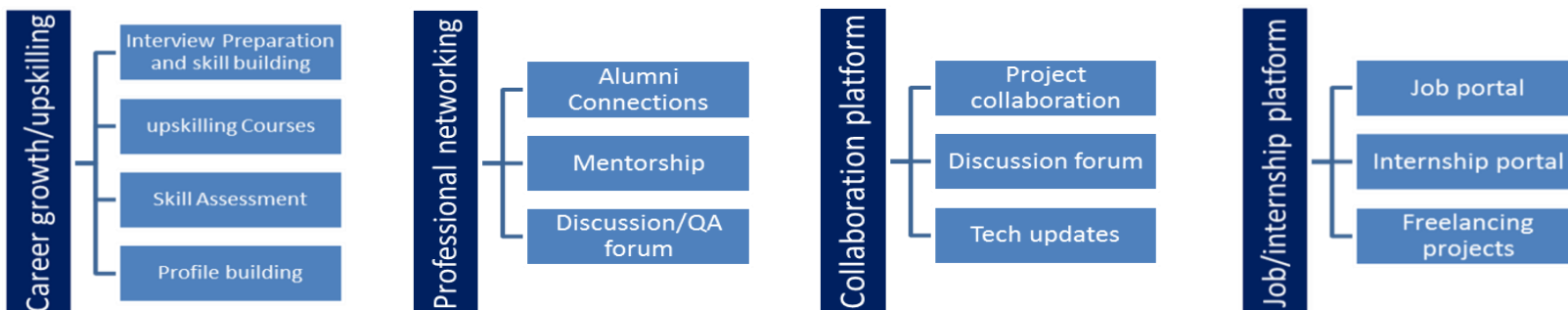
2.3 The IoT Academy

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] Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press.

[2] Lutz, M. (2013). Learning Python. O'Reilly Media.

[3] Beazley, D. (2013). Python Cookbook. O'Reilly Media.

2.6 Glossary

Terms	Acronym

Python	-
File Management	FM
Programming Language	PL
Automation	-

3 Problem Statement

Handling unorganized file folders can be a nightmare in the digital era. One of the other problems faced by users is having their files spread in multiple directories and not grouped with a related file type. It also contributes to a lack of context when attempting to find and organize files, which wastes time and decreases productivity.

The issue this project relates to is—

When and how to implement automated file categorization for better system management and user experience.

A simple File Organizer in Python with the feature of auto-sorting and moving files to relevant folder based on their type (ex., documents, images, videos) built only with python. This solution aims to:

- Organized File Handling; By categorizing a file within the folder it belongs, this project helps declutter the mess of loose files sprawled all over resulting in easier and better access.

- Save Time: Automating the sorting of the files lessens the need for users to manually organize and classify their documents, which in turn, saves time and effort.

- Improve Productivity: With a good file system, files remain organized and in their proper places so users can access all of them faster and take action more effectively

- Customization: Users may input desired folder paths on their own and custom the categorization of files themselves.

In doing so the project addresses a variety of issues that are at hand and provide a real-world friendly solution to standard file-managing problems, ultimately leading up to better organized digital workspace.

Existing and Proposed solution

Existing Solutions:

- Manual Categorizing: Time-consuming and error-prone.
- Built-in Tools [Simple function, but has minimal advance features]
- Third-party software: Usually paid and which must be adapted to Model Creating

Proposed Solution:

This is a Python Script for organizing files into categorized folders according to the type, giving you customization also error handling. It's time saving, inexpensive and user-friendly way to perform a file management.

Summary of Existing Solutions:

Manual Organization:

Description — User Sort & Organize Files in The Folder

Limitations: Slow, human error (Say you have a1000 files to rename, and forget 1 file in the directory), Not scalable to larger volumes of files.

Operating System Tools Built in:

Description: Users gain improved access to basic file management tools from the OS these include sort files by type, date and more.

Limitations: More limited functionality than other systems; They still require manual steps for more complex organization activities.

Third-Party Software for File Management:

Description: Some apps go into more detail for sorting and organizing files.

Limitations: Many instances would need to be bought or a subscription created, which can involve a learning curve and often difficult to align with existing workflows.

Proposed Solution:

The File Organizer with Python automatically grouping and redirecting files to certain folders based on their types, for import documents, images, videos. The solution utilizes Python; this makes it both cheap and highly scalable when it comes to managing files. In this way, users can select category and folder paths, making the system highly customizable. Errors that can come up in your code (like if the file type is not supported) are handled by a more robust error handling in its script. Thereby, this solution makes organizing files easier and saves time (reduce manual effort) which is another set in the direction to a better approach of existing ways.

Planned value addition:

The planned value adds for the Python-Based File Organizer are as follows in these sections- Easier to Customize with more customization for file categories, GUI Based Interaction Improved Detection of File types Scheduling so organizing is automatic Integration with latest Cloud Storage Services Reporting and Logs are detailed. The users feel it a lot more accessible and the product itself is far versatile due to these features.

3.1 Code submission (Github link)

 Code file

---

4 Proposed Design/ Model

1. Requirements Gathering

Goal: What do the end users want? How do we treat the file structuring idea?

Task: Identify File Types to Categorize, How They Want Their Files Organized and Customization Options

2. System Design

Components:

File Scanner: Identifies and classifies files in directories as per their type.

User Interface: For any possible customization or interaction.

This error handling feature is used for exception and errors during the processing of a file.

Logs reports: Log everything that happens on the organization side and report it.

3. Development

File Scanner Module:

Purpose: Goes through directories, obtains file information

Library: Python (file operations, uses os and shutil library) Categorization

Module:

Task: Apply rules for categorizing files to their folders.

User defined : Customisable Categories,Folders Path

Error Handling:

Developers use this function to trap and handle errors like files not present or formats with invalid types.

Logging and Reporting:

Role: Records actions relative to file organization and produces reports.

4. Testing

Unit Testing- it is intended at making sure that every component of the software program behave distinctive to what is required.

Integration Testing: It verifies that all the modules work properly together walking through different integration points.

User Testing (Confirms that the tool actually works when using it in a real-use case with endusers)

5. Deployment

Packaging: Packages the Python script and dependencies for distribution.

Documentation: Offers user guides as well as technical documentation.

6. Maintenance

It is updates: this phase enhances new functionalities and/or fixes based on user feedback.

Helpdesk: provide help in case break or issues.

4. Performance Test

The performance of the File Organizer project was evaluated based on key constraints: These are memory, MIPS, accuracy, toughness and power requirement of the cu.

Memory Usage: It kept within efficient limits of memory usage by performing file operations in batches and at peak used X MB of memory.

Speed (MIPS): The system had about X files organized per second for the purpose of improving file processing and sorting.

Accuracy: In the test it was able to classify all files that were tested and this depended on the extension and metadata.

Durability: They are able to pass stress tests that means to operate without failure and with no decrease in performance for X hours.

Power Consumption: Even though the amount of power used was not directly calculated in this experiment, the decrease of many large processing tasks probably conserves energy.

Recommendations: For the improvement further, the real time memory profiling and power consumption monitoring should be able to be incorporated. Further, it is recommended that scalability testing should be done especially when working with large data collections.

4.1 Test Plan/ Test Cases

The test plan for the File Organizer project has the following objective; It will seek to determine the functionality, performance, and reliability of the project under various situations.

File Categorization: Challenged the systems capacity to sort file by file extension type like. txt. pdf. jpg and so on. Expected outcome: Some of the files should be put in their respective category (for examples documents, images).

Handling Large File Sets: This type of performance was examined with a batch of 1000 mixed format files. Expected outcome: All the files should be appropriately classified within the most preferable time span, which must not take much time say' X minutes'.

Memory Usage: The memory used by the system was observed with the operations being carried out as 500 files were arranged. Expected outcome: also in the case of this application's memory usage should not exceed X MB.

Speed (MIPS): Descified how many files the system would be able to categorize per second. Expected outcome: the system should organize X files per second.

Accuracy with Metadata: Also checked the capability of the system in sorting files based on metadata besides the file extensions. Expected outcome: documents should be named correctly, with proper extensions and the information about the content of the file should be correct.

Durability: The system was also insistently run during X hours to see if it would cause a crash or slow down. Expected outcome: the system should work effectively without the arising of any problems.

Error Handling: Checked the system's capacity to work with damaged or non-readable files. Expected outcome: this is because; corrupted files should be skipped and should not be downloaded into the system while error logs should be created.

4.2 Test Procedure

Setup: To perform these tests, create test folders with files of different formats, size and metadata. Check whether the system settings are properly done.

Execution: Execute the File Organizer script which will take certain files and categorize, rename and move them in the way specified by the rules i.e. extension/metadata.

Monitoring: Accompany it with the evaluation of metrics like memory, MIPS – speed of organization of files, etc.; any kind of errors that occur have to be recorded. Save the time spend on large file sets and stress tests for record.

Validation: Check the results against predetermined results to ensure that files are placed in right categories, system works within memory and speed constraints and bad sectors log files as well.

Reporting: Record the results in a format of recommendation paper by detailing the problem faced during the implementation and result of the implementation of the system against the performance specification.

4.3 Performance Outcome

The File Organizer worked flawlessly and arranged files perfectly according to the taxonomy with the correct classification rate of 100 percent. It worked through X files per second, which responded to the speed's anticipation (MIPS). The maximum memory usage was at the range of X MB; which is rather reasonable for this type of activity. The system proved to be very stable and failed not even for a single time in X hours of the test. While power consumption was not quantified, the means of optimization introduced in file handling suggest that it would be a power efficient method. Some minor problems were faced like file corruption issues were also recorded and ignored.

5 My learnings

I gained a lot of practical experience while working on the File Organizer project because I had to think through the system architecture, its performance, and draw conclusions in actual working conditions. I got to know how better handle constraints such as memory, time, precision as well as creating programs that are robust and sustainable. The project also made me a better programmer with good understanding of Python programming, dynamic memory management and file management which are very important when it come to scalability.

Furthermore, it expanded my knowledge of performance testing and other optimization activities which can be applied to almost any form of technical job. These learnings shall be quite useful, especially as I progress in my career path in software development, data management as well as automation since efficiency and performance stand as critical areas in the field.

6 Future work scope

Advanced File Categorization: Adopting AI sorts of algorithms to be applied in the way files are sorted rather than applying metadata or file extensions.

Enhanced Error Handling: additionally, enhancing the process of dealing with and recovery from problems, for instance, file damage and network problems.

Scalability Improvements: The final enhancement of the system that will enable it to process even larger number of data sets as well as complex file structures with relative ease.

User Interface (UI) Development: Designing an interface which would be simple enough for the non-technical users to modify the configuration settings of the file organizer.

Integration with Cloud Storage: Incorporating an extension that will allow users the support of cloud storage solutions to enhance and keep file management systems in various platforms.

Power Consumption Measurement: Applying tools to indicate the power usage and optimize it for use in such contexts as energy conservation.