

Winter Internship Program

Project Name : File Organizer

Prepared by : Meena Kumari



Organize your files
with Python

LearnDataAnalysis.org

Executive Summary

This report provides details of the Winter 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 the file organizer related to the Python project that helps users organize their files in a directory. It scans a specified directory, categorizes files based on their type and moves them into respective folders.

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

Throughout the development process, we have learned valuable lessons in software design, performance optimization, user experience, and collaboration. We have explored best practices in Python development, gained insights into real-world file organization challenges, and honed our skills in project management and teamwork.

TABLE OF CONTENTS

1. Preface
2. Introduction
 - 2.1 About UniConverge Technologies Pvt Ltd
 - 2.2 About upskill Campus
 - 2.3 Objective
 - 2.4 Reference
3. Problem Statement
- 4 .Proposed Solution
5. Proposed Model/Design
 - 5.1 Python directory workflow
 - 5.2 Interfaces.
6. Performance Testing
7. My learning
8. Conclusion

1.Preface

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

My project was the file organizer related to the Python project that helps users organize their files in a directory. It scans a specified directory, categorizes files based on their and moves them into respective folders.

```
extension, folder_name in extensions.items():
    # get all the files matching the extension
    files = glob.glob(os.path.join(path, f"*.{extension}")
    print(f"[*] Found {len(files)} files with {extension}")
    if not os.path.isdir(os.path.join(path, folder_name)):
        # create a folder if it does not exist before
        print(f"[+] Making {folder_name} folder")
        os.mkdir(os.path.join(path, folder_name))
    for file in files:
        # for each file in that extension, move it to
```

Our goal is to develop a versatile and user-friendly tool that allows individuals to effortlessly organize their files according to their preferences and needs.

2. Introduction

2.1 About UniConverge Technologies Pvt Ltd

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

Uni Converge Technologies' services include:

Wireless communication, Cloud computing, Mobility, Data analytics, Machine learning, Artificial intelligence, Embedded technologies.

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.

i. **UCT IoT** **uct** **Insight****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.



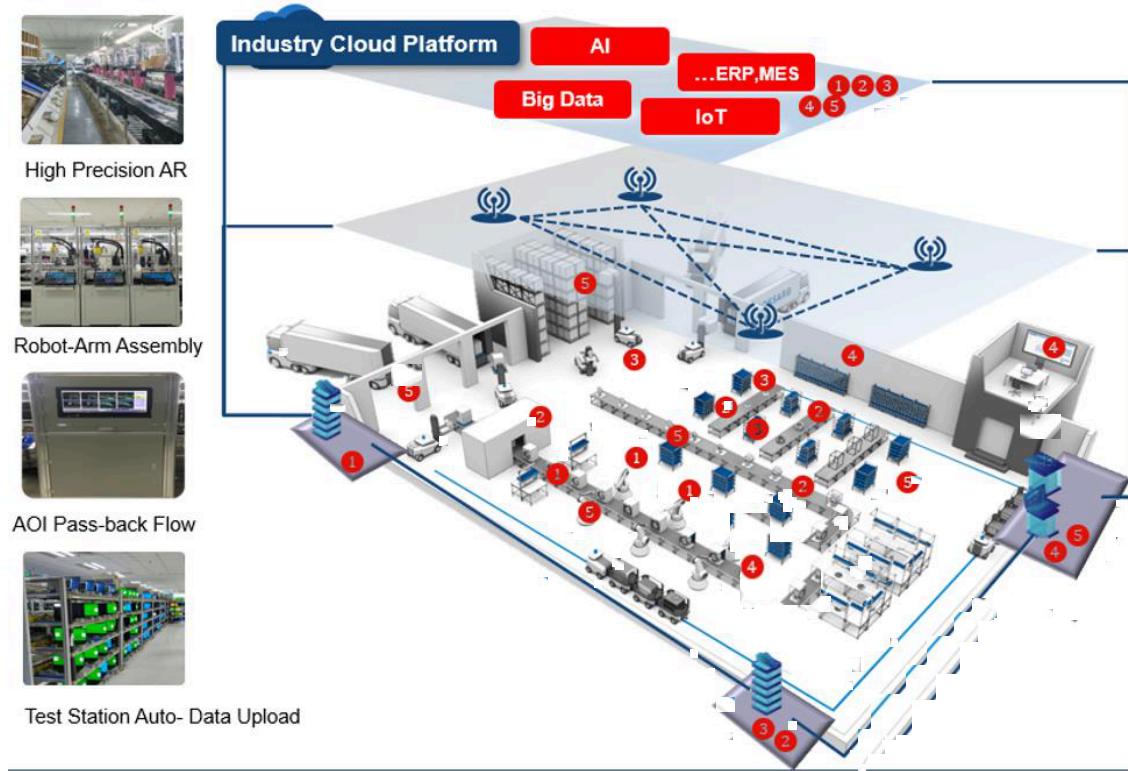
FACTORY WATCH

ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

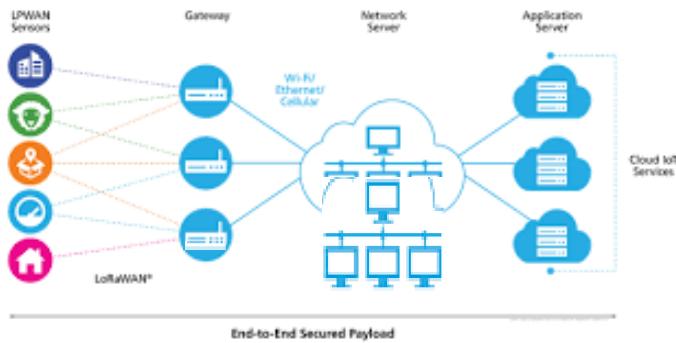
- with a scalable solution for their Production and asset monitoring



LoRaWAN™

iii. based Solution

UCT is one of the early adopters of LoRAWAN technology and providing solution in 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.

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.

2.4 Reference

[1] Python Documentation : [File Organizer - Jupyter Notebook](#)

[2] GitHub: <https://github.com>

3 Problem Statement

Description:

The file organizer is a Python project that helps users organize their files in a directory. It scans a specified directory, categorizes files based on their type (e.g., images, documents, videos), and moves them into respective folders.

Scope:

The scope of this project involves designing a user interface to specify the directory to organize, implementing functions to identify file types and create folders, and developing a file-moving algorithm to organize files into the appropriate folders.

The general process to construct a Python program that organizes files in a directory is as follows:

1. Identify the Directory
2. Define the Category
3. Define the path of the Destination
4. Write code to sort files
5. Test Your Code
6. Run the code on the source directory

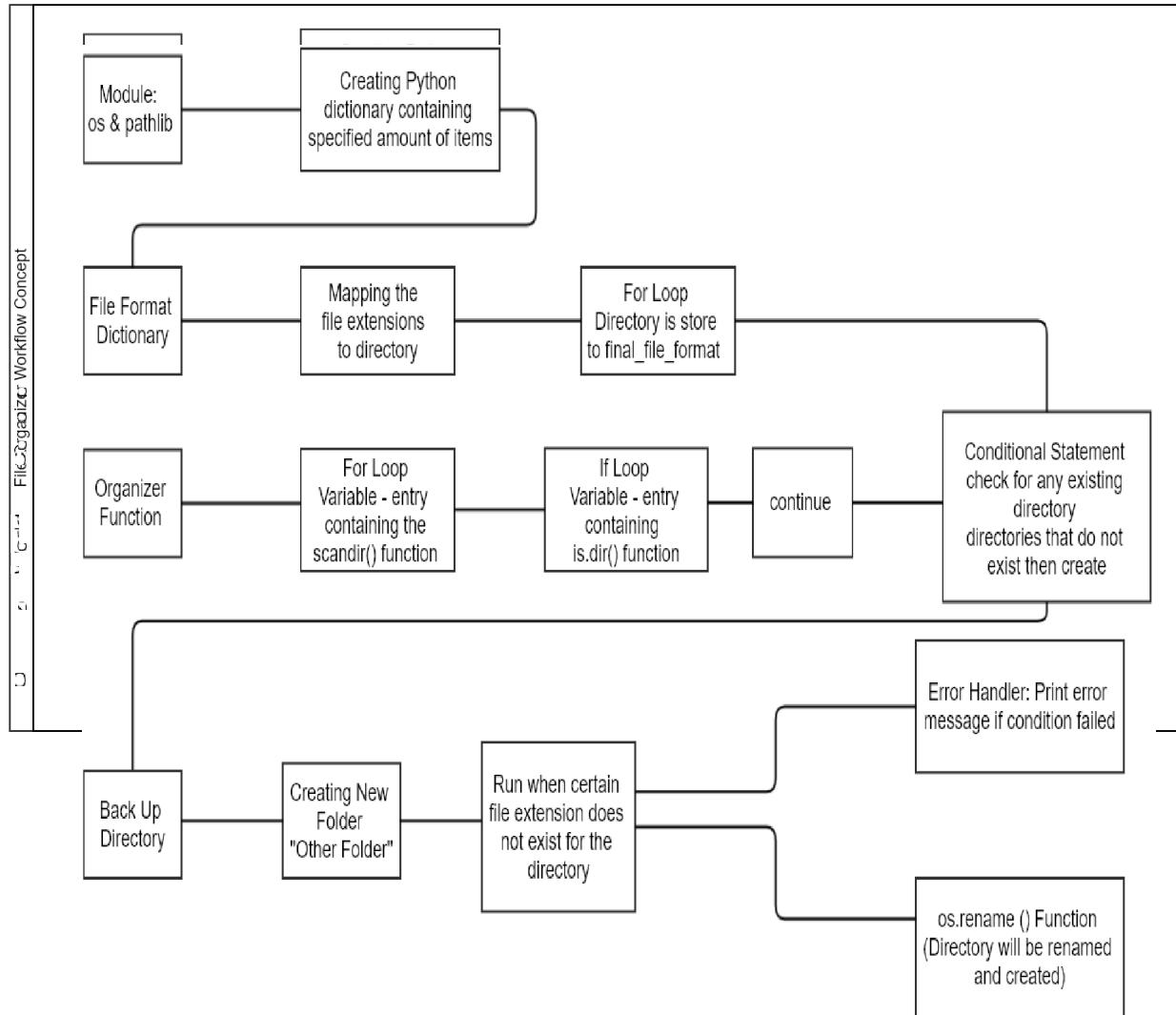
4 Existing and Proposed solution

- Understand the concepts and fundamentals of file directories.
- Do your research on the Miscellaneous operating system interfaces and Object-oriented filesystem paths pathlib import modules.
- Look into the different file directory functions in Python

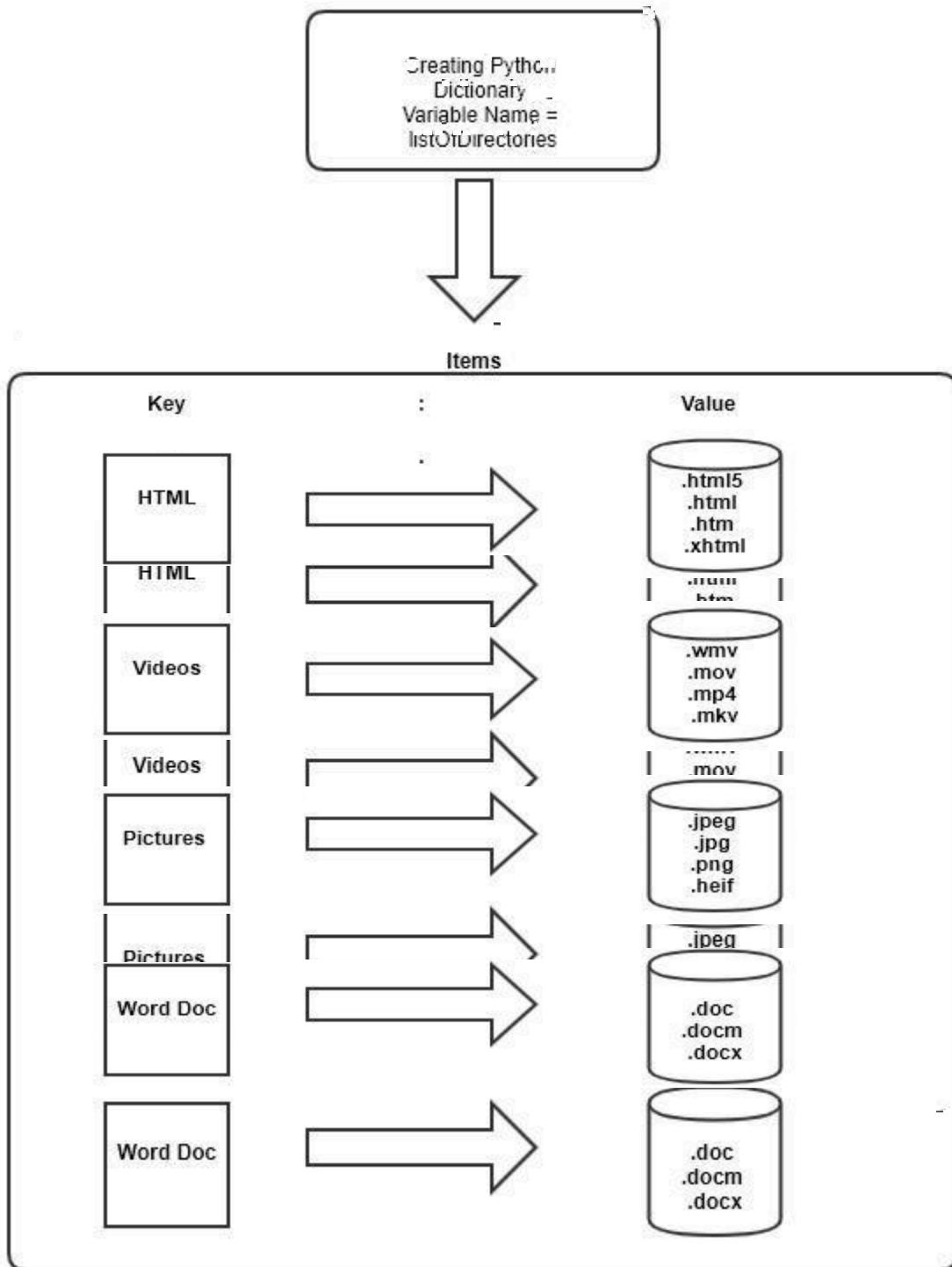
- Learn and understand the concept of python Directory.

- Learn to create and declare a function in Python.
- Learn and practice using for loops and if loops

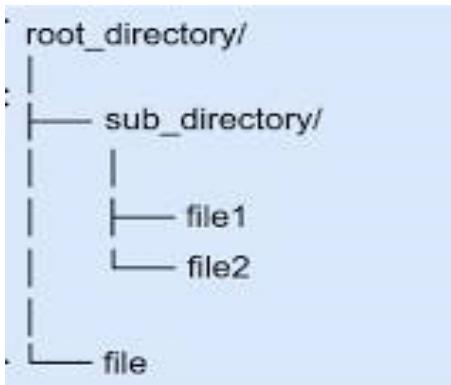
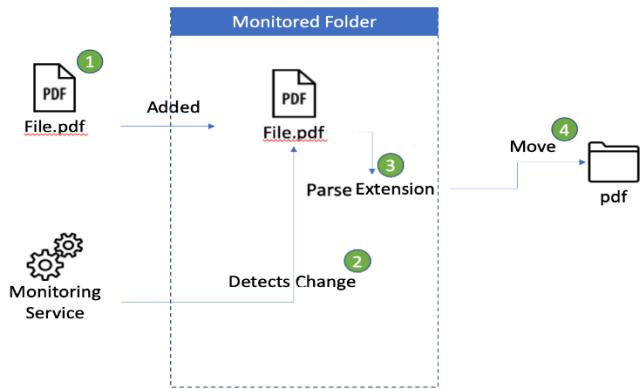
5 Proposed Design/ Model



5.1 Python directory workflow



5.2 Interfaces



6. Performance Test

Testing environment:

- Operating System: Windows 10 / macOS / Linux (depending on the target platform)
- Hardware: Standard desktop or laptop computer with sufficient resources (CPU, RAM, storage)

- Software: Python environment with necessary dependencies installed
(Tkinter, Watchdog, SQLite)

Identified constraints

- a. Memory: The amount of memory consumed by the application during file organization tasks.
- b. CPU Usage (MIPS): The processing power required by the application to handle file organization operations efficiently.
- c. Disk I/O: The speed and efficiency of reading and writing files to disk during organization tasks.
- d. Scalability: The ability of the application to scale and perform consistently with increasing numbers of files and concurrent operations.

Recommendations to Handle Constraints:

- a. Continuously monitor and optimize memory usage to prevent potential memory leaks or excessive consumption, especially when dealing with large datasets.
- b. Implement performance profiling and tuning to identify and address CPU bottlenecks, ensuring efficient utilization of available processing power

5.3 Test Procedure

Before Organizing Files:

- Create a test directory with a known structure and files.
- Define test cases to verify the initial state of the directory and files.
- Write Python functions to execute these test cases.
- Ensure the test directory is untouched before running the organizer.

After Organizing Files:

- After running the file organizer, define test cases to verify the expected state of the directory and files.
- Write Python functions to execute these test cases.
- Compare the actual state of the directory and files with the expected state
- Ensure the organizer has correctly organized files according to your predefined rules.

7. My learnings

- Through this project, I gained a deeper understanding of how file systems work and the various operations involved in managing files, such as reading, writing, moving, and deleting.
- Successfully executed basic tasks and learned about Python features.
- Collaborating with team members on the file organizer project taught me valuable project management and collaboration skills, including communication, task delegation, and problem-solving.

- Acquired proficiency in essential Python libraries, such as Pandas, Matplotlib, Scikit-Learn and so on.
- Enhanced about functions of Code Editors and their Environments
- I learned about performance optimization techniques such as multithreading, asynchronous programming, and disk I/O optimization to improve the efficiency and scalability of the application.
- Known about in Numpy and Pandas libraries completely and also Through testing the file organizer project, I gained experience in software testing and quality assurance practices, including writing test cases, conducting performance tests, and debugging

8.Conclusion:

The File Organizer Project in Python that I have done is a great way to learn how to use Python to organize your files. The project is relatively simple to complete, and it can be customized to meet your specific needs.

By doing this project it made me to arrange my different types of files into its respective folders within less time and with high efficiency

The file organizer project exemplifies the power of Python as a versatile programming language for developing practical solutions to real-world problems. We are proud of the accomplishments achieved thus far and look forward to continuing our journey of innovation and discovery in the field of software development.

“Thank you to all who have helped me to gain valuable knowledge, which made me explore the best practices in Python Development.”