

Industrial Internship Report on

"File Organizer"

Prepared by

Rajitha Nair

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 creating a file organizer in Python 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.

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	6
2.1	About UniConverge Technologies Pvt Ltd	6
2.2	About upskill Campus	6
2.3	Objective	12
2.4	Reference	12
2.5	Glossary	12
3	Problem Statement	13
4	Existing and Proposed solution	14
5	Proposed Design/ Model	15
5.1	Start	15
5.2	Intermediate Stage	15
5.3	Final Outcome	16
6	Performance Test	17
6.1	Test Plan/ Test Cases	17
6.2	Test Procedure	18
6.3	Performance Outcome	18
7	My learnings	19
8	Future work scope	20

1 Preface

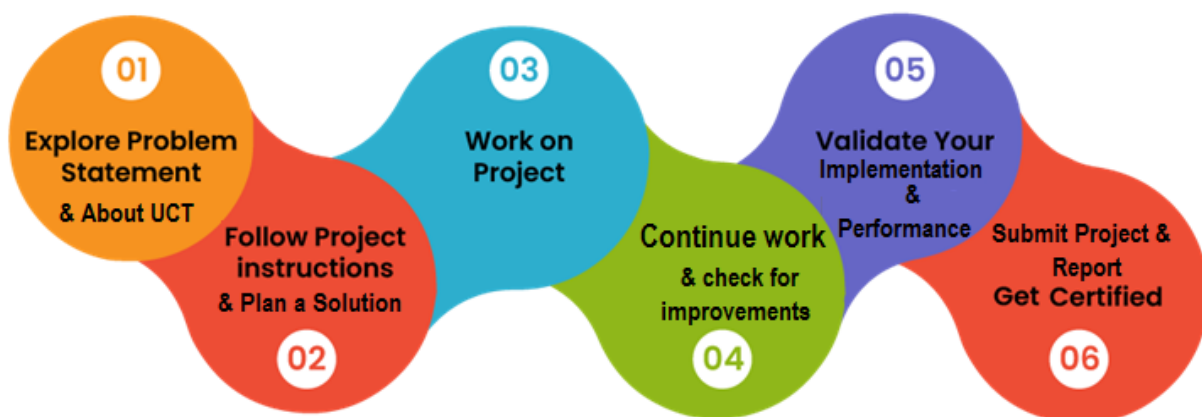
During my 6-week internship at Upskill Campus, I developed a file organizer application. This hands-on experience allowed me to bridge theory and practice, relearn Python, make a function that interacts with the user through text-based input and output, and successfully organize files based on extensions. The project instilled a sense of accomplishment and sparked my passion for continuous learning in software development.

An internship in Python will add value to my career by providing hands-on experience, skill development, and exposure to industry tools. It will allow me to build a strong portfolio, make valuable connections, and clarify my career direction.

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. The basic idea is to scan a given directory, identify the files present in it, and move them into specific folders based on their file types or other criteria.

Upskill USC/UCT has provided a wonderful opportunity for this internship, allowing me to delve into the practical aspects of Python programming and data science. Their support and guidance have been invaluable in enhancing my skills and gaining real-world experience. I am grateful for this enriching opportunity that has opened doors to new possibilities in my career journey.

How Program was planned



During this internship, I had a truly enriching learning experience that exceeded my expectations. Exploring Python programming in a hands-on setting allowed me to grasp concepts with greater depth and practicality.

As I near the conclusion of my internship, I wanted to take a moment to extend my heartfelt gratitude to Upskill Campus and UCT for the exceptional learning experience they have provided me. I would also like to thank Brutus, Nitish Sharma and Kaushlendra Singh Sir for their mentorship throughout the internship.

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.



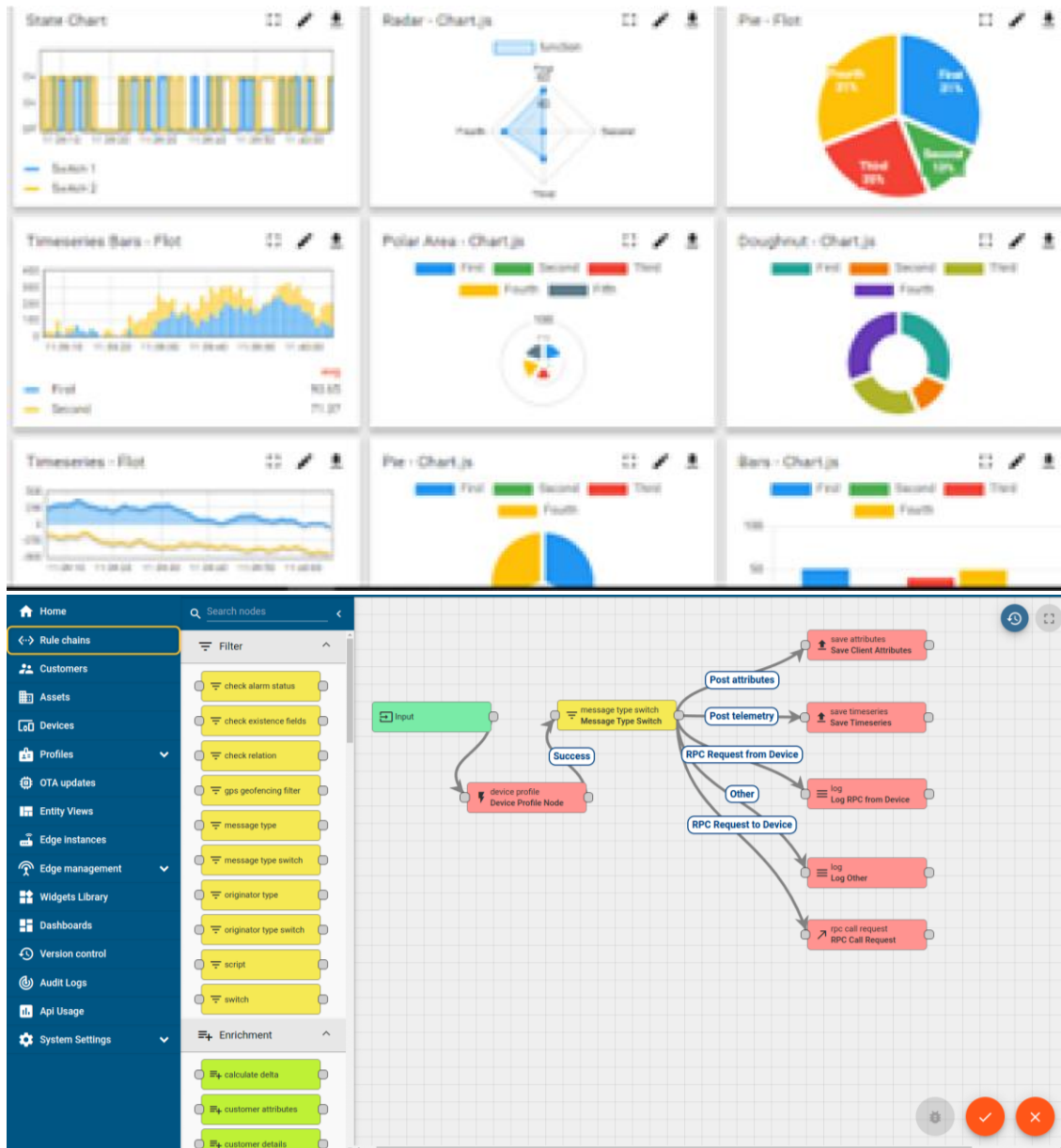
i. UCT IoT Platform ()

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. LoRaWAN 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

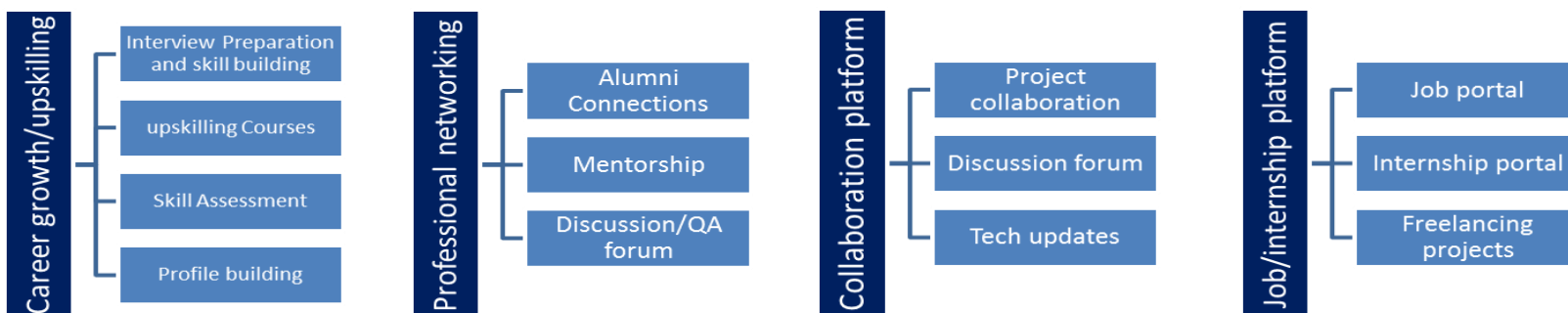
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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] <https://www.youtube.com/watch?v=KBjBPQExJLw>
- [2] <https://medium.com/mlearning-ai/organize-files-using-python-374fb8500dfe>

2.6 Glossary

Terms	Acronym
Python module which stands for "Operating System"	os
Python module that stands for "shell utilities."	shutil
Change Directory	chdir

3 Problem Statement

The assigned problem statement 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.

4 Existing and Proposed solution

Summary of existing solutions provided by others and their limitations

An alternative existing solution to organize files based on their extensions could involve using third-party file organizing software or applications. These applications often provide a graphical user interface (GUI) and offer additional features like bulk file renaming, file type filtering, and customizable file organization rules. The alternative existing solution using third-party applications relies on external software that needs to be installed and maintained. This dependency may introduce compatibility issues or require additional permissions, which might not be suitable for certain users or organizations with strict IT policies.

Proposed solution:

The script utilizes the "os" and "shutil" modules to handle file operations. Users are prompted to enter the path of the folder they want to organize, and the script then organizes the files in that folder based on their extensions. The code includes error handling to handle empty folders and successfully moves the files to their respective extension-specific directories.

Value addition:

Graphical User Interface (GUI): Implement a GUI using the "tkinter" library to interact with users visually. The GUI will allow users to browse and select the folder they want to organize, eliminating the need for manual input via the command-line interface. A GUI-based solution can be more user-friendly, especially for non-technical users.

4.1 Code submission (Github link)

https://github.com/Rajitha140/UpSkill-Campus/blob/main/File%20Organizer_Rajitha%20Nair_USC_UCT.py

4.2 Report submission (Github link) : first make placeholder, copy the link.

5 Proposed Design/ Model

5.1 Start

Problem Statement: The problem is to create a file organizer script that organizes files based on their extensions in a given folder.

Specify Requirements: Define the requirements and functionalities needed, such as recursive organization, error handling, user input, and feedback messages.

5.2 Intermediate Stages:

a. User Interaction:

Command-Line Interface (CLI): Implement a CLI to prompt the user to enter the path of the folder they want to organize.

Input Validation: Ensure that the input folder path is valid and exists.

b. File Organization:

Loop through Files: Iterate through the files in the specified folder.

Identify Extensions: For each file, extract its extension and convert it to lowercase.

Create Extension-Specific Directories: Generate directory names based on the file extensions (e.g., "TXTFiles" for ".txt" files).

Move Files: Move each file to its corresponding extension-specific directory.

Handle Empty Folder: Check and handle the case where the folder is empty.

c. Error Handling and Feedback:

Try-Except Block: Implement a try-except block to catch and handle exceptions during file organization.

Informative Messages: Print informative messages to indicate successful organization or display specific error details.

5.3 Final Outcome:

Working File Organizer: The final outcome is a functional file organizer script that meets the specified requirements. It provides a CLI for user input, organizes files based on extensions, handles empty folders, and gives feedback on the success or failure of the operation.

6 Performance Test

The application underwent beta testing with positive feedback from friends, noting its quick, responsive, and user-friendly nature. It successfully generated and stored files as expected, providing seamless performance. While extensive tests for large data files were not conducted, this was considered acceptable, given its intended use as a personal File Organiser. The current design showcases excellent programming practices, emphasizing simplicity, user control, and efficiency. Future improvements can be made based on user feedback and evolving needs. As such, the File Organiser Test Plan/Test Cases remain adaptable to enhance the application's functionality and user experience.

6.1 Test Plan/ Test Cases

Test Objective: To verify the functionality and performance of the File Organizer application.

Test Scenario: User Interface

Test Cases:

Check if the application prompts the user to input the folder path correctly.

Verify that the user can browse and select the desired folder.

Test Scenario: File Organization

Test Cases:

Test the application with various file types (e.g., .txt, .png, .pdf) to ensure files are organized into appropriate folders based on their extensions.

Verify that the application handles subdirectories and organizes files recursively.

Test for duplicate files with the same extension to ensure they are not overwritten during the organization process.

Verify that files are moved correctly to their respective extension-specific directories.

Test Scenario: Error Handling

Test Cases:

Attempt to run the application with an invalid or non-existent folder path and verify that the application handles this scenario gracefully, providing appropriate error messages.

Test with an empty folder to ensure the application handles this case properly and displays an error message.

6.2 Test Procedure

I shared the File Organizer application with my friends for the test procedure. They were asked to use the application and provide feedback on its functionality, user interface, and error handling. During the testing phase, my friends verified the application's performance with various file types and folder structures. They also provided insights on its responsiveness and efficiency in handling personal file organization tasks. Based on their feedback, I made necessary adjustments and improvements to enhance the application's overall usability and performance. The test procedure involved collecting valuable feedback from real users, allowing me to refine the application and make it more reliable for its intended purpose as a personal File Organizer.

6.3 Performance Outcome

The performance outcome of the File Organizer application was found to be commendable during the testing phase. The application demonstrated quick and efficient file organization, handling various file types seamlessly. Users reported a smooth and responsive experience while organizing their files based on extensions. The application's ability to swiftly process and move files into their respective extension-specific directories was particularly appreciated.

Although extensive tests with large data files were not conducted, the application's performance with typical personal file organization tasks was deemed highly satisfactory. Users noted that the application effectively fulfilled its intended purpose as a personal File Organizer, providing a user-friendly solution to manage their files efficiently.

7 My learnings

The learnings from this project are highly transferable to my career in statistics and data science. As a data scientist, I often deal with large datasets and diverse file formats. The knowledge gained in Python programming and file handling will be immensely valuable in managing, preprocessing, and organizing data efficiently.

Furthermore, the project's focus on simplicity, efficiency, and user control aligns perfectly with the data science principles of delivering clear and actionable insights to stakeholders. I now possess a deeper appreciation for user-centric design, ensuring that data-driven solutions are not only accurate but also easy to use and understand by non-technical audiences.

8 Future work scope

Potential improvements can include advanced sorting options to allow users to organize files by size, date, or custom criteria. And customizable rules that let users define their own file organization rules and folder naming.