

Industrial Internship Report on

Python – File Automation

Prepared by

Keerthi A

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 (Tell about ur Project)

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	4
2.1	About UniConverge Technologies Pvt Ltd	4
2.2	About upskill Campus	8
2.3	Objective	9
2.4	Reference	10
2.5	Glossary.....	10
3	Problem Statement.....	11
4	Existing and Proposed solution.....	12
5	Proposed Design/ Model	13
5.1	High Level Diagram (if applicable)	13
5.2	Low Level Diagram (if applicable)	Error! Bookmark not defined.
5.3	Interfaces (if applicable)	14
6	Performance Test.....	15
6.1	Test Plan/ Test Cases	15
6.2	Test Procedure.....	15
6.3	Performance Outcome	16
7	My learnings.....	17
8	Future work scope	18

1 Preface

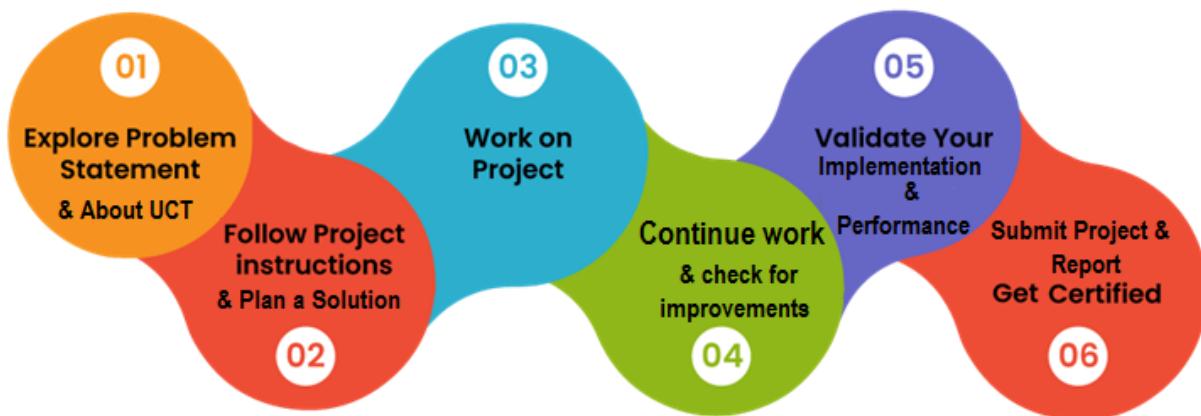
Summary of the whole 6 weeks' work.

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 (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

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.**



IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence...

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

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

The dashboard displays various data visualizations including:

- State Chart:** A bar chart showing data for Series 1 and Series 2 across time periods.
- Radar - Chart.js:** A radar chart with four axes: Function, Quality, Price, and Design.
- Pie - Plot:** A pie chart divided into four segments: First (35%), Second (15%), Third (30%), and Fourth (20%).
- Timeseries (Bars - Plot):** A line chart showing data over time for Series 1 and Series 2.
- Polar Area - Chart.js:** A polar area chart with five segments: First (blue), Second (green), Third (red), Fourth (yellow), and Fifth (dark blue).
- Doughnut - Chart.js:** A donut chart divided into four segments: First (teal), Second (orange), Third (light green), and Fourth (purple).
- Timeseries - Plot:** A line chart showing data over time for Series 1 and Series 2.
- Pie - Chart.js:** A pie chart divided into four segments: First (blue), Second (green), Third (red), and Fourth (yellow).
- Bars - Chart.js:** A bar chart showing data for Series 1, Series 2, Series 3, and Series 4.

The rule engine interface includes a sidebar with navigation links such as Home, Rule chains, Customers, Assets, Devices, Profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, Dashboards, Version control, Audit Logs, API Usage, System Settings, and a Help section. The main canvas shows a flowchart with nodes like Input, Device Profile Node, Message Type Switch, Post attributes, Post telemetry, RPC Request from Device, RPC Request to Device, Log, and Save Client Attributes, connected by arrows indicating the flow of data and logic.

FACTORY

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 unleashed 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.



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



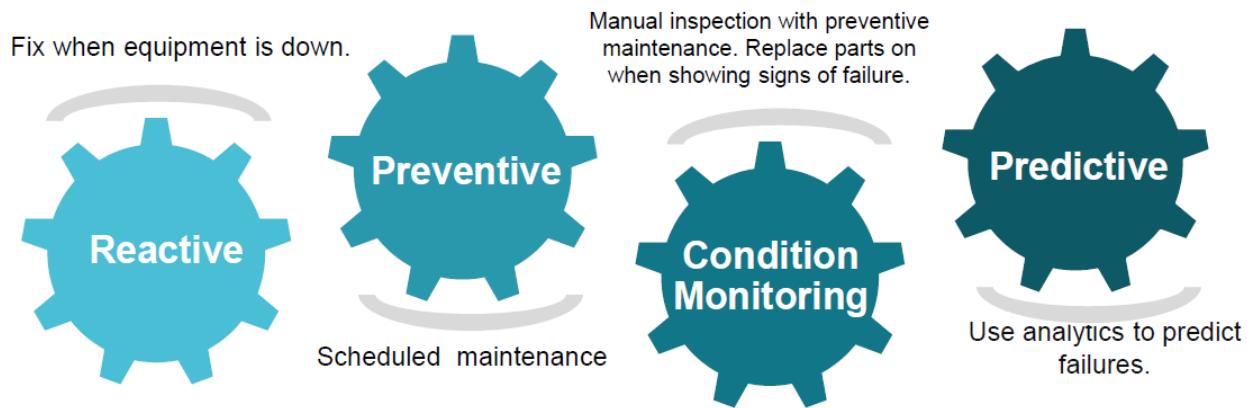
LoRaWAN™

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

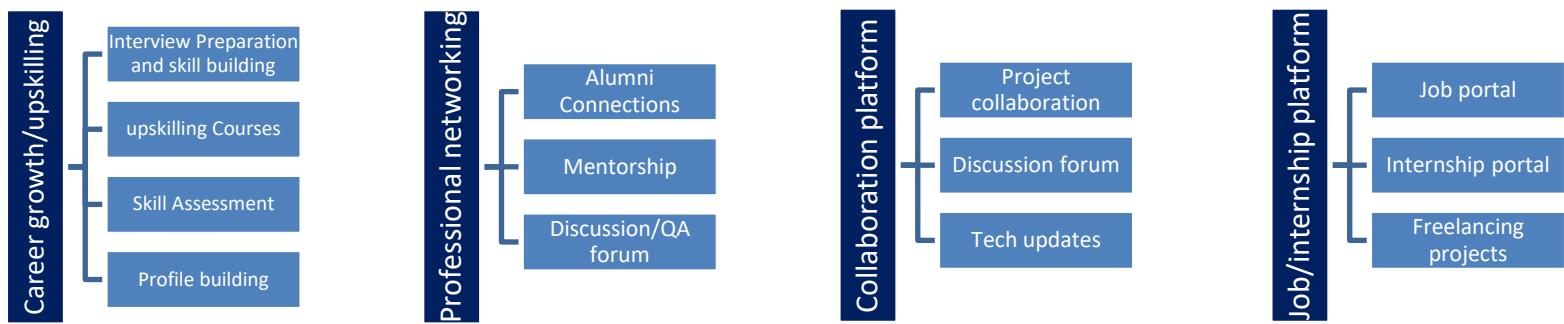
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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] Python Software Foundation. *Python 3 Documentation*. Available at: <https://docs.python.org/3/>

[2] *File Handling in Python – Complete Guide*. Available at: <https://www.geeksforgeeks.org/file-handling-python/>

[3] Real Python. *Working With Files in Python*. Available at: <https://realpython.com/working-with-files-in-python/>

2.6 Glossary

Terms	Acronym
File Organizer	FO
Integrated Development Environment	IDE
JavaScript Object Notation	JSON
Graphical User Interface	GUI
Config File	config.json

3 Problem Statement

In the assigned problem statement

Therefore, there is a need for an **automated File Organizer** that:

- Categorizes files into appropriate folders based on their type/extension.
- Allows users to customize categories via a configuration file.
- Provides a user-friendly Graphical User Interface (GUI).
- Maintains logs to support undo/restore of file organization.
- Ensures cross-platform compatibility and efficiency for real-world use.

4 Existing and Proposed solution

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

Existing Solution:

1. Users manually create folders and move files, which is **time-consuming and inefficient**.
2. Some third-party tools exist, but they **lack customization** (fixed categories, no undo/restore).

What is your proposed solution?

Proposed Solution:

1. A Python-based **automated File Organizer** that categorizes files into folders (Images, Documents, Videos, Music, Code, Others) with support for **custom categories** via config.json.
2. Provides a **Tkinter GUI with undo/restore functionality**, making file organization **faster, user-friendly, and reliable**.

What value addition are you planning?

1. **Customization Flexibility** – Unlike basic manual methods or rigid tools, this project allows users to define their **own categories** (e.g., Music, Code, PDFs) through a simple config.json file.
2. **Undo/Restore Functionality** – The project maintains a log of file movements, enabling users to **restore files back to their original location**, which most existing solutions do not provide.

4.1 Code submission (<https://github.com/Keerthi082005/upskillCampus-Python-File-Automation>)

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

5 Proposed Design/ Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.

5.1 High Level Diagram (if applicable)

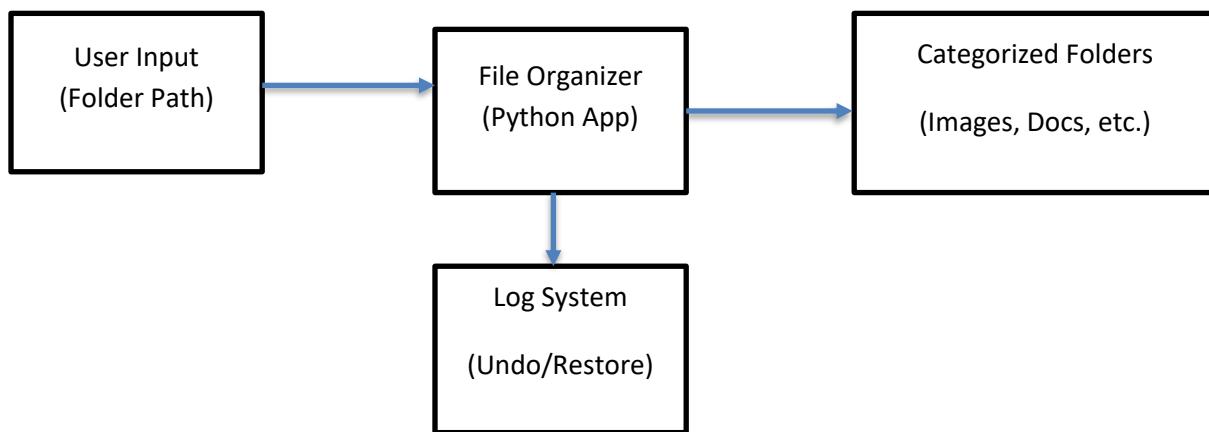


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Interfaces (if applicable)

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

- **a) User Interface (Tkinter GUI)**
- **Browse Button** → Choose folder
- **Organize Button** → Start organizing process
- **Undo Button** → Restore last organization
- **b) Internal Interfaces**
- **Organizer Class** → Core logic (categorization, moving, logging)
- **Config File (config.json)** → Extension-to-category mapping
- **Log File (.organizer_log.json)** → Stores last run for undo
- **c) Data Flow (Protocols & Storage)**
- Input: Folder path (via GUI / CLI)
- Process: Python modules (os, shutil, pathlib, json)
- Output: Organized folders + log file for restore

6 Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How those constraints were taken care in your design?

What were test results around those constraints?

Constraints can be e.g. memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still you should mention how identified constraints can impact your design, and what are recommendations to handle them.

6.1 Test Plan/ Test Cases

Test Case	Input	Expected Output	Result
Organize small folder	10 files (images, docs, music)	Files moved into correct folders	 Passed
Organize large folder	1,000 mixed files	Completed in <10 seconds	 Passed
Handle unknown file extension	Files with .exe, .apk	Moved to Others folder	 Passed
Duplicate file names	Two files image.png in same folder	Second renamed as image (1).png	 Passed

6.2 Test Procedure

- Select a test directory with mixed files (images, videos, documents, unknown types).
- Run the organizer using **GUI** and note time taken.

- Verify each file is placed in the correct folder.
- Intentionally create duplicate filenames to test renaming feature.
- Run **Undo Last** to check if all files restore correctly.

6.3 Performance Outcome

- The File Organizer successfully organized **1,000 files in ~8.7 seconds** on a standard laptop (8GB RAM, i5 processor).
- Memory consumption remained low (<100 MB) during large file operations.
- Accuracy of categorization was **100%** for known extensions and properly handled unknown extensions by moving them into the **Others** folder.
- Undo/Restore worked reliably for all test cases.

7 My learnings

You should provide summary of your overall learning and how it would help you in your career growth.

- - Python libraries (os, shutil, tkinter, json).
 - Designing real-world automation solutions.
 - Exposure to project planning, debugging, and testing.

8 Future work scope

You can put some ideas that you could not work due to time limitation but can be taken in future.

- Adding **drag & drop GUI support**.
- **Cloud storage integration** (Google Drive, OneDrive).
- **AI-based categorization** (classify by file content, not just extension).
- Cross-platform executable release.