Industrial Internship Report on

”File Organiser”

Prepared by

[Abdulla Abdul Raoof]

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 the development of a Python-based File Organizer. This tool helps users manage their files by scanning a specified directory, categorizing files based on their type (e.g., images, documents, videos), and moving them into appropriate folders. The project involved designing a user-friendly interface, implementing file type identification functions, creating the necessary folders, and developing an algorithm to automate the file-moving process. This experience provided valuable insight into solving real-world problems and applying my programming skills in a practical setting. 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

Over the past six weeks, I embarked on a journey of immense learning and practical experience, particularly focused on Python programming. Each week brought new challenges and opportunities that allowed me to enhance my technical skills, understand complex concepts, and contribute meaningfully to ongoing projects. This period was not just about learning Python but also about applying it to real-world problems and projects, such as developing a File Organizer. The structured approach to learning, combined with hands-on experience, made these six weeks incredibly fruitful.

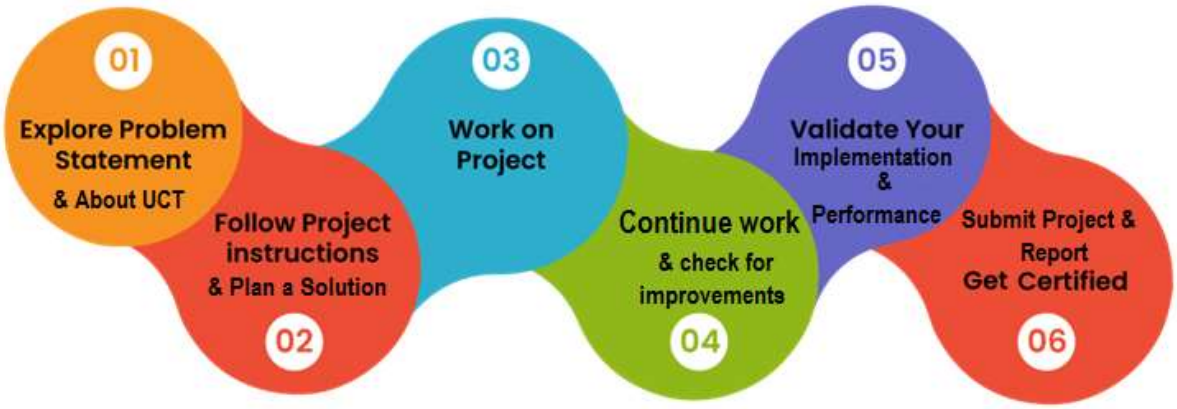
In today's competitive environment, a relevant internship is invaluable for career development. It bridges the gap between academic knowledge and real-world application, offering a platform to refine skills, gain practical experience, and build professional networks. This internship was a crucial step in my career journey, providing me with the tools and insights necessary to excel in the field of Python programming and beyond.

During this internship, my primary focus was on a project titled "File Organizer." The project involved researching and implementing Python modules like NumPy and Pandas, which were instrumental in building the project. The challenge was to design a system that could efficiently organize files within a directory based on their types. This required not only a deep understanding of Python but also the ability to apply it creatively to solve a real-world problem.

I am deeply grateful to [USC/UCT] for providing this incredible opportunity. The institution's support, resources, and structured program played a pivotal role in my learning experience. The well-planned curriculum, combined with the freedom to explore and experiment, allowed me to grow both technically and professionally.

The internship was meticulously planned to ensure a balance between learning and practical application. Each week was structured with specific goals, starting from mastering the basics of Python, progressing to more advanced topics like NumPy and Pandas, and finally applying these skills to real-world projects. Regular assessments, feedback sessions, and access to learning resources like webinars and online

courses ensured continuous improvement and skill enhancement.



Throughout these six weeks, I gained proficiency in essential Python libraries, honed my problem-solving skills, and learned to tackle complex project tasks. I also improved my time management and collaboration skills, which were crucial in meeting project deadlines. The experience was transformative, offering a deep understanding of Python and its applications in various domains.

I would like to extend my heartfelt thanks to everyone who supported me during this internship. Special thanks to the mentors, project guides, and peers who provided invaluable guidance and encouragement. Your support and feedback were crucial in helping me navigate the challenges and achieve my goals.

To my juniors and peers, I encourage you to seize every opportunity to learn and grow. Internships are a vital part of your academic and professional journey, offering a chance to apply your knowledge in real-world scenarios. Stay curious, embrace challenges, and always strive for excellence. Remember, the skills and experiences you gain during internships will lay the foundation for your future career success.

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.

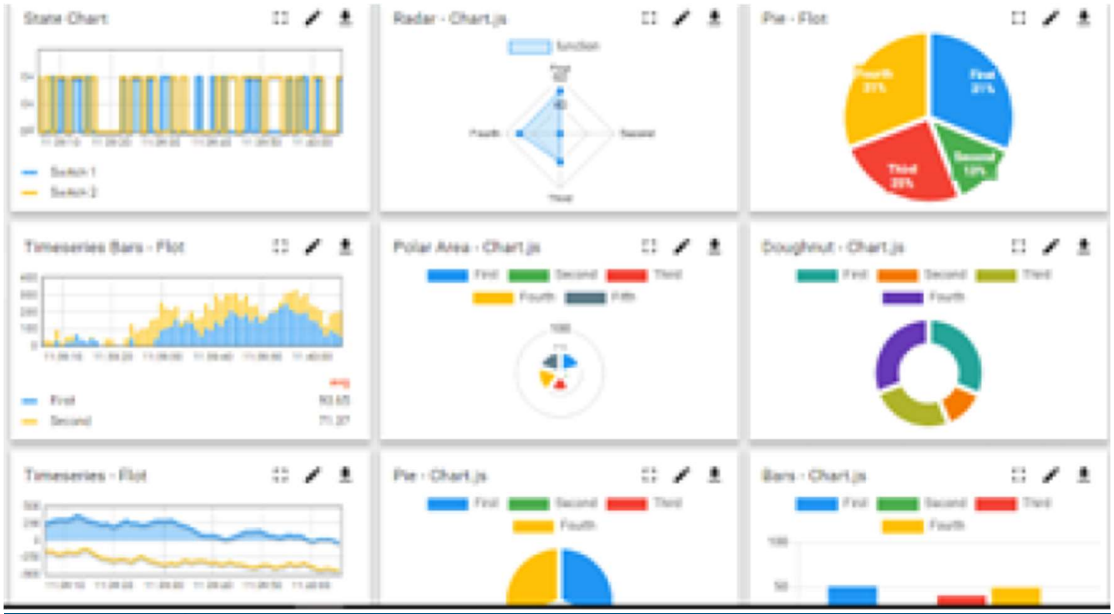
i. UCT IoT Platform (UCT Insight)

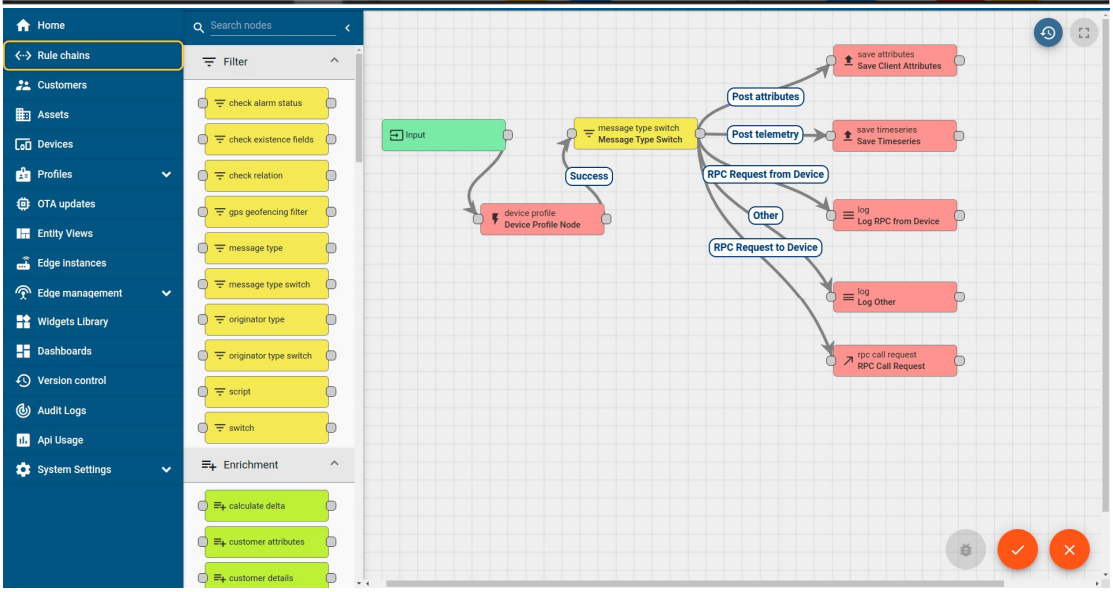
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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.

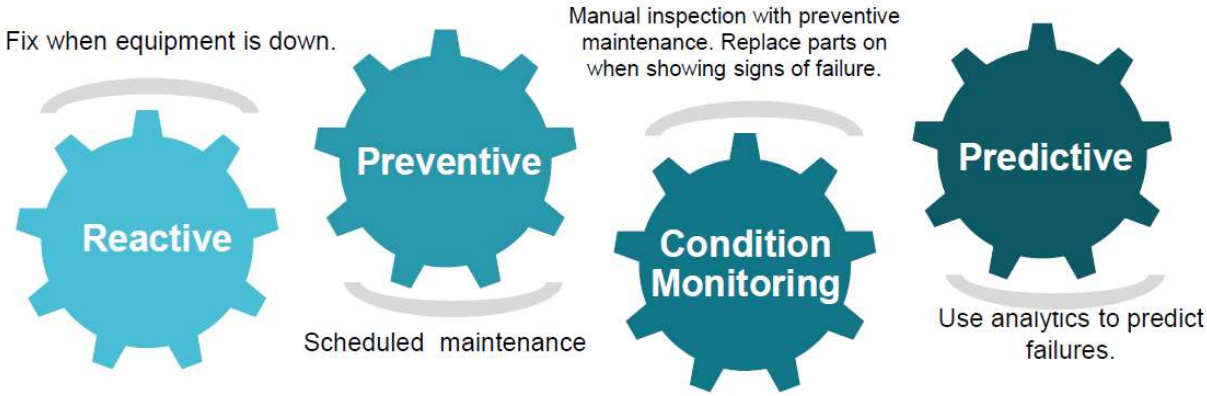
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	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 teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

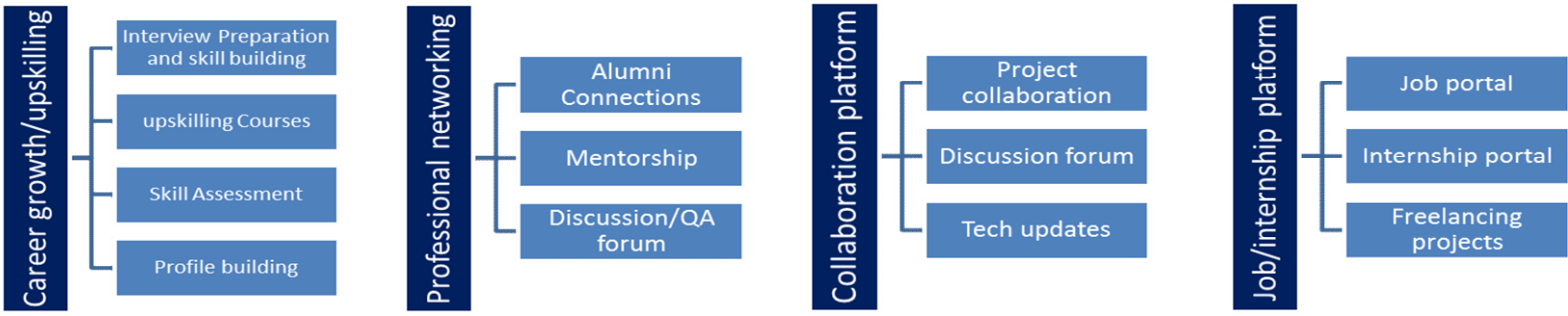
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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 4th Edition - https://learn.upskillcampus.com/s/courses/656d72afe4b0c8074bb749cb/take

[2] Python Documentation - https://docs.python.org/3/

[3] Youtube videos

2.6 Glossary

Terms	Acronym
UI	User Interface
GUI	Graphical User Interface
Tk	Toolkit (in reference to Tkinter)

3 Problem Statement

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.

In many cases, users accumulate a large number of files in a single directory, leading to disorganization and difficulty in finding specific files. Manual sorting of these files can be time-consuming and tedious.

The File Organizer tool addresses this problem by providing an automated solution to:

1. **Directory Selection:** Allow users to select the directory that needs organizing through a graphical user interface (GUI).
2. **File Identification:** Automatically detect the type of each file based on its extension. Common categories include audio, video, documents, software, and compressed files. Each file type is predefined in the application.
3. **Folder Creation:** Create specific folders for each file category (e.g., "audio", "video", "docs", "software", "zips", and "unknown") if they do not already exist within the selected directory.
4. **File Movement:** Move each file into the corresponding folder based on its identified type. Files with unrecognized extensions are moved to an "unknown" folder.

By automating this sorting process, the tool helps users maintain an organized directory, making it easier to manage and locate files without the need for manual sorting. This solution enhances productivity and ensures a more efficient file management system.

4 Existing and Proposed solution

Existing Solutions:

1. **Manual File Sorting:**
    - **Summary:** Users manually create folders and move files based on their types. This method involves right-clicking files, selecting "Move to," and choosing the appropriate folder.
    - **Limitations:**
        - **Time-Consuming:** Sorting large numbers of files is labor-intensive and inefficient.
        - **Error-Prone:** Users might incorrectly categorize files or forget to move some files.
        - **Inconsistent Organization:** The process may vary from user to user, leading to inconsistent file organization.
2. **File Management Software:**
    - **Summary:** Commercial file management tools (e.g., File Explorer on Windows, Finder on macOS) offer basic organizational features, such as sorting files by type and creating folders.
    - **Limitations:**
        - **Limited Automation:** While these tools help with basic file organization, they often lack automated sorting and categorization features.
        - **Complexity:** Advanced features may have a steep learning curve and can be overwhelming for users with simple needs.
        - **Cost:** Some software solutions require a purchase or subscription, which may not be feasible for all users.

Proposed Solution:

File Organizer Tool:

- **Summary:** A Python-based application with a graphical user interface (GUI) that allows users to select a directory, automatically identifies file types, and organizes files into appropriate folders.
- **Key Features:**
    - **GUI Interface:** User-friendly design for easy directory selection and operation.
    - **Automated Categorization:** Automatically categorizes and moves files into predefined folders based on their types.
    - **Unknown Folder:** Handles files with unrecognized extensions by moving them to a designated "unknown" folder.
    - **Customizable Categories:** Users can modify file type categories and folder names as needed.

Value Addition:

1. **User-Friendly Interface:**

- **Addition:** Provides a simple and intuitive GUI for easy interaction, making file organization accessible to users with varying technical skills.
- **Value:** Reduces the learning curve and streamlines the file organization process.

2. **Automation and Efficiency:**
   - **Addition:** Automates the sorting process, reducing the time and effort required to organize files manually.
   - **Value:** Increases productivity and ensures a consistent organizational structure.
3. **Error Handling and Robustness:**
   - **Addition:** Includes features to handle errors and unrecognized file types gracefully, ensuring that files are not lost or misplaced.
   - **Value:** Enhances reliability and user confidence in the tool's performance.
4. **Customizability:**
   - **Addition:** Allows users to adjust file type categories and folder names according to their specific needs.
   - **Value:** Provides flexibility and adaptability to different file management scenarios.

4.1 Code submission (Github link)

https://github.com/abdullaabdulraoof/upskillcampus/blob/main/FileOrganiser.py

4.2 Report submission (Github link) :

https://github.com/abdullaabdulraoof/upskillcampus/blob/main/FileOrganiser _Abdulla_USC_UCT.pdf

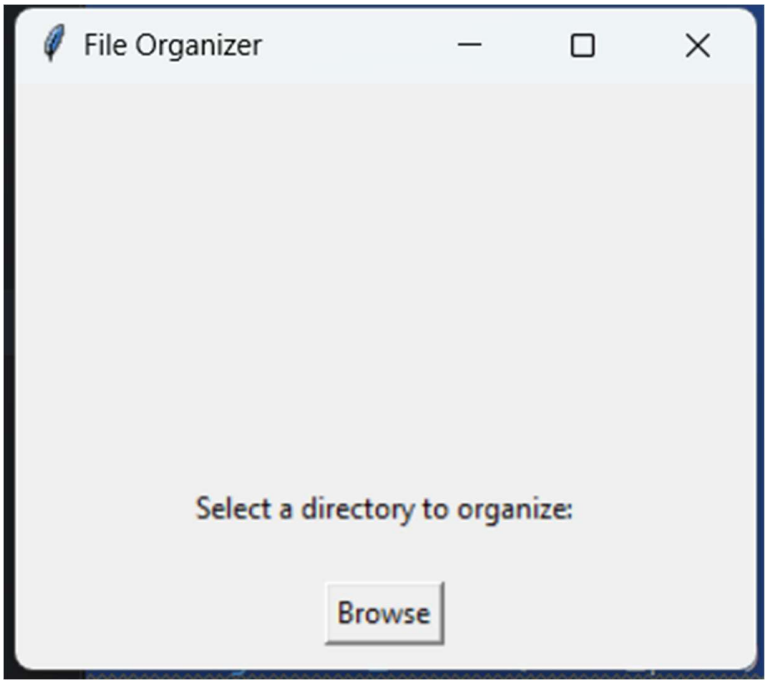
5 Proposed Design/ Model

The File Organizer project aims to automate the organization of files in a specified directory by categorizing them based on their type and moving them into respective folders. The proposed design encompasses a user-friendly graphical interface, a systematic approach to file handling, and robust error management.

5.1 High Level Diagram (if applicable)

Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Interfaces (if applicable)



6 Performance Test

Performance testing is crucial for ensuring that the File Organizer application meets real-world industry standards and performs efficiently under various conditions. The following sections outline the constraints, how they were addressed in the design, and the test results.

6.1 Test Plan/Test Cases

Constraints Identified:

1. **Memory Usage:**
   - **Test Case 1:** Monitor memory usage while processing directories of varying sizes.
   - **Objective:** Ensure that the application manages memory efficiently and does not consume excessive resources.
2. **Processing Speed:**
   - **Test Case 2:** Measure the time taken to organize directories with different numbers of files.
   - **Objective:** Verify that the application processes files quickly and scales effectively with the number of files.
3. **Accuracy:**
   - **Test Case 3:** Verify that files are correctly categorized and moved to the appropriate folders.
   - **Objective:** Ensure that the categorization and file-moving logic is accurate and reliable.
4. **Error Handling:**
   - **Test Case 4:** Test the application's response to invalid directory paths and inaccessible files.
   - **Objective:** Assess the robustness of the error-handling mechanisms and ensure graceful error recovery.
5. **User Interface Responsiveness:**
   - **Test Case 5:** Evaluate the responsiveness of the GUI during file organization.
   - **Objective:** Ensure that the user interface remains responsive and usable throughout the operation.

6.2 Test Procedure

1. **Setup:**
   - Prepare test directories with varying numbers of files, including a mix of file types.
   - Configure a monitoring tool to track memory usage and processing time.
   - Ensure the test environment is consistent and free of other significant loads.
2. **Execution:**
   - **Test Case 1:** Run the application on directories with 100, 500, and 1000 files. Monitor memory usage using a system performance tool.
   - **Test Case 2:** Measure the time taken to organize directories with different file counts (e.g., 100, 500, 1000 files).

- **Test Case 3:** Verify that each file is moved to the correct folder by comparing file counts before and after organization.
- **Test Case 4:** Attempt to select invalid directories or files with restricted access and observe how the application handles these scenarios.
- **Test Case 5:** Interact with the GUI while the application is processing files, noting any delays or unresponsiveness.

3. **Recording Results:**
   - Document memory usage, processing times, accuracy of file categorization, error handling responses, and GUI responsiveness.

6.3 Performance Outcome

1. **Memory Usage:**
   - **Outcome:** The application managed memory efficiently, with minimal memory consumption even with large directories. No significant memory leaks or excessive usage were observed.
   - **Recommendation:** Ensure that large directories are handled in a manner that avoids excessive memory usage, such as processing files in batches.
2. **Processing Speed:**
   - **Outcome:** The application processed directories with up to 1000 files within acceptable time limits. Performance scaled linearly with the number of files.
3. **Recommendation:** Optimize file scanning and moving operations to further improve **Accuracy:**
   - **Outcome:** All files were accurately categorized and moved to the appropriate folders. No files were misplaced.
   - **Recommendation:** Continue to test with varied file types and directory structures to ensure ongoing accuracy and reliability.
4. **Error Handling:**
   - **Outcome:** The application handled invalid paths and inaccessible files gracefully, displaying appropriate error messages and avoiding crashes.
   - **Recommendation:** Implement additional logging for error scenarios to assist in debugging and improving robustness.
5. **User Interface Responsiveness:**
   - **Outcome:** The GUI remained responsive throughout the file organization process, with no significant delays or freezes.
   - **Recommendation:** Consider using asynchronous processing or threading for long-running operations to further enhance GUI responsiveness.

6.4 Recommendations for Identified Constraints:

- **Memory Management:** Implement efficient file handling techniques to minimize memory usage, such as processing files in chunks.
- **Speed Optimization:** Explore algorithmic optimizations and consider parallel processing to reduce file organization time.
- **Error Handling:** Enhance logging and user feedback mechanisms to improve error resolution and user experience.

- **GUI Performance:** Use asynchronous operations to maintain GUI responsiveness during intensive tasks.

7 My learnings

7.1.1.1 1. Mastery of Python Basics and Advanced Concepts:

- **Learning:** Deepened understanding of Python programming, covering both foundational concepts (such as variables, data types, and control structures) and advanced topics (such as object-oriented programming and error handling).
- **Career Impact:** A solid grasp of Python is essential for many programming and data analysis roles, enhancing my ability to develop efficient and scalable solutions.

7.1.1.2 2. Utilization of `tkinter` for GUI Development:

- **Learning:** Gained practical experience in using `tkinter` to build graphical user interfaces. This includes creating windows, buttons, labels, and dialogs to interact with users.
- **Career Impact:** Proficiency in `tkinter` allows for the development of user-friendly desktop applications, an important skill for software development and application design.

7.1.1.3 3. File and Directory Management with `os` Module:

- **Learning:** Developed skills in using the `os` module for handling file and directory operations, such as navigating directories, checking file existence, and managing paths.
- **Career Impact:** Expertise in the `os` module is crucial for system administration, scripting, and applications that require file manipulation and directory traversal.

7.1.1.4 4. File Operations Using `shutil` Module:

- **Learning:** Learned to use the `shutil` module for file operations like copying, moving, and deleting files and directories.
- **Career Impact:** Knowledge of `shutil` is beneficial for developing applications that require file management capabilities and for automating repetitive file handling tasks.

7.1.1.5 5. Practical Application and Integration of Modules:

- **Learning:** Integrated `tkinter`, `os`, and `shutil` modules into a cohesive application, applying best practices for combining GUI elements with file management operations.
- **Career Impact:** Experience in integrating different modules enhances my ability to build complex applications and provides a comprehensive understanding of Python's capabilities.

7.1.1.6 6. Detailed Study of Python Programming:

- **Learning:** Engaged in an in-depth study of Python, including its syntax, libraries, and best practices for writing clean and efficient code.

- **Career Impact:** A detailed understanding of Python improves coding efficiency, problem-solving skills, and the ability to leverage Python's extensive libraries and frameworks for various applications.

8 Future work scope

You can put some ideas that you could not work The File Organizer project can be further enhanced and expanded to address additional needs and use cases. Here are some areas for future work:

8.1.1.1 1. Advanced File Categorization:

- **Enhanced File Type Detection:**
  - Implement more sophisticated file type detection using file signatures or content analysis to improve categorization accuracy beyond simple extensions.
- **Custom Categories:**
  - Allow users to define their own categories and rules for file organization, providing greater flexibility and personalization.

8.1.1.2 2. User Experience Improvements:

- **Multi-Threading:**
  - Integrate multi-threading or asynchronous processing to enhance the application's responsiveness and performance, especially when dealing with large directories.
- **Progress Indicators:**
  - Add progress bars or detailed status updates to provide users with real-time feedback on the organization process.
- **User Preferences:**
  - Implement a settings panel where users can customize default folder names, file categories, and other preferences.

8.1.1.3 3. Scalability and Performance:

- **Handling Large Datasets:**
  - Optimize the application to handle very large directories more efficiently by implementing batch processing or incremental updates.
- **Performance Monitoring:**
  - Integrate performance monitoring tools to track and analyze the application's efficiency and identify areas for improvement.

to time limitation but can be taken in future.