



**DE LA SALLE UNIVERSITY - MANILA**

**FILE ORGANIZER**

---



A Term Project

Presented to Mr. RUIZ, RAMON STEPHEN L. and Mr. VELASCO, NEIL OLIVER M.

In Partial Fulfillment of the

Requirements for the Course Programming Logic and Design (PROLOGI) and Programming  
Logic and Design Laboratory (LBYCPA1)

---

by

LIM, Curt Ethan C. - Signature/Initials

OBISPO, Lance Altheus A. - Signature/Initials

SIMBAJON, Gabrielle D. - Signature/Initials

PROLOGI - EQ2

LBYCPA1 - EQ4

April 20, 2023

# Table of Contents

<b>Table of Contents .....</b>	<b>2</b>
<b>I. Introduction .....</b>	<b>4</b>
A. Background of the Study.....	4
B. Problem Statement .....	5
C. Objectives.....	6
D. Significance of the Project .....	6
<b>II. Review of Related Literature.....</b>	<b>8</b>
<b>III. Methodology .....</b>	<b>9</b>
A. Conceptual Framework – IPO Chart (Input-Process-Output-Chart) .....	9
Figure 3. Conceptual Framework.....	9
B. Hierarchy Chart .....	10
Figure 3.1. Hierarchy Chart. ....	10
C. Flowchart.....	11
Figure 3.2. Flowchart.....	11
Figure 3.3. Flowchart.....	12
Figure 3.4. Flowchart.....	13
D. Pseudocode.....	15
<b>IV. Result .....</b>	<b>16</b>
Figure 4. First Test. ....	16
Figure 4.1. First Test. ....	17
Figure 4.2. First Test - Sorted.....	17
Figure 4.3. First Test - Result. ....	17
Figure 4.4. Second Test. ....	19
Figure 4.5. Second Test - Sorted.....	19
Figure 4.6 Second Test - Result. ....	20
Figure 4.7. Third Test. ....	21
Figure 4.8. Third Test. ....	21

Figure 4.9. Third Test - Sorted.....	22
Figure 4.10. Third Test - Result. ....	21
<b>V. Discussion of Results.....</b>	<b>23</b>
<b>VI. Analysis, Conclusion, and Future Directives .....</b>	<b>24</b>
<b>References .....</b>	<b>25</b>
<b>Appendices.....</b>	<b>26</b>
A. User's Manual .....	26
Figure 8. User Guide. ....	26
B. Source Code .....	27
C. Work Breakdown .....	29
Table 8. Itemized Workload. ....	30
D. Personal Data Sheet.....	31

## **I. Introduction**

The management of digital files on personal computers, including various file types such as documents, images, videos, and music audios, can be a daunting and time-consuming task. However, failing to appropriately organize these files can lead to disorientation and inefficiency in file retrieval, ultimately impeding productivity. Both personal and professional endeavors are affected by this disorganization and neglect of appropriate archiving procedures, as the accumulation of files makes it increasingly challenging to keep track of them.

To simplify this process, the File Organizer project has been developed, which utilizes a program to automatically organize files in a directory based on their file type. By recognizing the file types based on their extensions, such as .mp3, .docx, .pdf, .png, and .exe, the program creates a dedicated folder for each file type and moves the corresponding files to their respective folders. This efficient sorting mechanism enables users to locate their desired files swiftly and conveniently, without the need to navigate through an entire directory and expend unnecessary time and effort in the process.

The sorting algorithm employed in the program facilitates efficient organization and categorization of files within a designated pathway, enabling users to locate and retrieve specific files with ease. This, in turn, saves time and increases productivity while reducing the computer's file system load. The program is a useful tool that simplifies the process of managing files, offering an effective solution to the challenges of file organization that users face.

In conclusion, managing files effectively is a complex task, particularly when working with a large number of files within directories. Manual organization can often lead to errors and unnecessary frustration when searching for specific files. However, the File Organizer project provides an efficient and convenient tool that streamlines the file management process, facilitating easy access and organization of files while reducing the potential for errors.

### **A. Background of the Study**

The constant generation and management of digital files, including documents, images, videos, and music, pose a challenge for computer users as the accumulation of files can lead to disorganization and difficulty in accessing desired files. The importance of file organization is

undeniable as it enables easy access, better control, and manageability of files. Despite various manual techniques for file organization, the process can be intimidating and time-consuming, particularly when dealing with a large number of files.

To address this issue, the File Organizer project aims to develop an automated tool that can efficiently and accurately sort files within a directory based on their file type. The program identifies different file types using their file extensions and relocates them to appropriate folders, streamlining the process of file organization. The project's objective is to provide users with an automated file organization tool that enhances productivity by reducing the time required to search for specific files. By simplifying the process of managing digital files, the project is expected to offer a practical solution to the problem of file organization.

## **B. Problem Statement**

As users continue to produce and accumulate more digital files, managing and organizing them becomes an increasingly challenging task. Without a proper system in place, files can become disorganized and difficult to locate, leading to wasted time and effort. While there are manual methods for organizing files, these can be time-consuming and prone to errors, especially when dealing with many files hence the need for an automated solution that can accurately and efficiently organize digital files based on their file types. (Inventor-Miranda, 2019)

Currently, there are few automated file organization tools available that can recognize and organize files based on their file extensions. (*8 Biggest Document Management Problems And Their Solutions* - GLOBODOX, 2022) While some operating systems have built-in file sorting options, these can be limited in their capabilities, may not always accurately identify files, and are not flexible. This creates a gap in knowledge in the field of file management and organization, and there is a need for further research to develop a reliable and efficient automated file organization tool.

The problem addressed by this project is the need for an automated file organization tool that can accurately identify and organize digital files based on their file types, to simplify the process of managing and locating files for users.

## **C. Objectives**

### **C.1. General Objective**

The general objective of this project is to create a program that can automatically organize files in a directory based on their file type, to simplify the process of managing digital files.

### **C.2. Specific Objectives**

1. To identify different file types and their corresponding file extensions.
2. To create separate folders for each file type in the given directory.
3. To move the corresponding files to their appropriate folders based on their file type.
4. To test and debug the program to ensure its effectiveness and efficiency.
5. To develop a user-friendly interface for the program to make it easy to use for non-technical users.

## **D. Significance of the Project**

The File Organizer project has significant benefits for individuals, programmers, and organizations. By automating the process of organizing digital files, this project can help users save time and effort in locating specific files, increasing their productivity. It can also help in reducing the operating system's file system load hence boosting system performance.

*For personal use.* The File Organizer project offers an easy-to-use and effective solution to managing digital files, particularly for individuals who accumulate a significant number of files over time. The program can assist in avoiding the annoyance of searching for specific files and minimize the risk of losing important files due to disorganization. By automating the process of file organization, users can save valuable time and effort while ensuring that their files are well-organized and easily accessible. This can ultimately lead to increased productivity and a more efficient workflow.

*For organizations.* The File Organizer project not only simplifies the process of managing digital files, but it can also contribute to data accuracy and security. The program's sorting mechanism helps to eliminate the possibility of errors that may occur when manually organizing files, such as

misplacing or deleting files. Additionally, by providing users with a clear idea of the files on their computer and their respective locations, the program can aid in data security by reducing the likelihood of accidental deletion or unauthorized access to sensitive files..

*For programmers.* The File Organizer project provides an opportunity to automate a tedious task and save time. It also allows them to gain experience in working with file systems, file management, and building user-friendly interfaces. Additionally, it can simplify the process of managing project files, increasing productivity, and organization.

## II. Review of Related Literature

A journal article conducted in 1976 highlights the importance of file structures in enabling the efficient processing of data based on the specific requirements of an application. In the case of the "File Organizer" project, the developers are utilizing a specific file structure to organize files in a directory based on their file type. This type of file organization can improve the efficiency of searching for specific files and can even improve system performance by reducing the load on the file system. The journal article also mentions the need for intelligent builders and designers of computer-based systems to have knowledge about various file structures to choose the appropriate one for a particular application. The developers are using a simple file structure to achieve the desired goal of organizing files based on their file types. This shows that we have considered the available options and chosen a suitable file structure to achieve our project objectives. (Van Rijsbergen, 1976, 294)

A book section published in 1972 introduces file organization techniques, which are relevant to the project, as the developers' goal is to create a tool that automatically organizes files in a directory. The section emphasizes the fundamental considerations that are important in the selection of a file organization technique, which is crucial for the project's success. The section also discusses various file structures, such as random file structures, list structures, and tree file structures, which may inspire the design of the file organization method used in the project. (Roberts, 1972, 115,174)

The File Organizer is a great example of an automated file organization tool that can help users easily locate files and increase productivity. By recognizing different file types in a directory based on their file extensions and moving them to appropriate folders, the program can make file organization effortless and efficient. Also, the project can also recognize files based on their content, which aligns with the content-driven approach of SortIt. This can be useful for files that have generic extensions, such as ".txt" or ".doc," but contain different types of content. By using File Organizer, users can be sure that all files with similar content are grouped together in a single folder, making it easier for them to find what they need. Furthermore, the project offers a user-based sorting feature, which is also present in SortIt. By allowing users to specify the type or name of a file they are looking for, our program can automatically create a folder related to similar items.



This can be especially helpful for users who may not be familiar with the different file extensions or types but still want to keep their files organized. (Ahlawat et al., 2020)

### III. Methodology

#### A. Conceptual Framework – IPO Chart (Input-Process-Output-Chart)



Figure 3. Conceptual Framework.

The program takes a directory path as an input and organizes the files in the directory based on their file type. It starts by getting all the files in the directory using the **os.listdir()** function. Then, it creates a dictionary called **file\_types** that holds different file types as keys and their corresponding file extensions as values. For instance, the "Images" key holds a list of file extensions that are typically used for images, such as ".jpeg", ".jpg", ".png", etc.

Next, the function creates a folder for each file type using the **os.makedirs()** function if the folder does not already exist. It loops through each key in the **file\_types** dictionary and creates a folder with the same name in the input directory.

Lastly, the function moves each file to its appropriate folder. It loops through each file in the directory, and checks its file extension using the **os.path.splitext()** function, and compares it to the file extensions in the **file\_types** dictionary. If it finds a match, it moves the file to the corresponding folder using the **shutil.move()** function.

## B. Hierarchy Chart

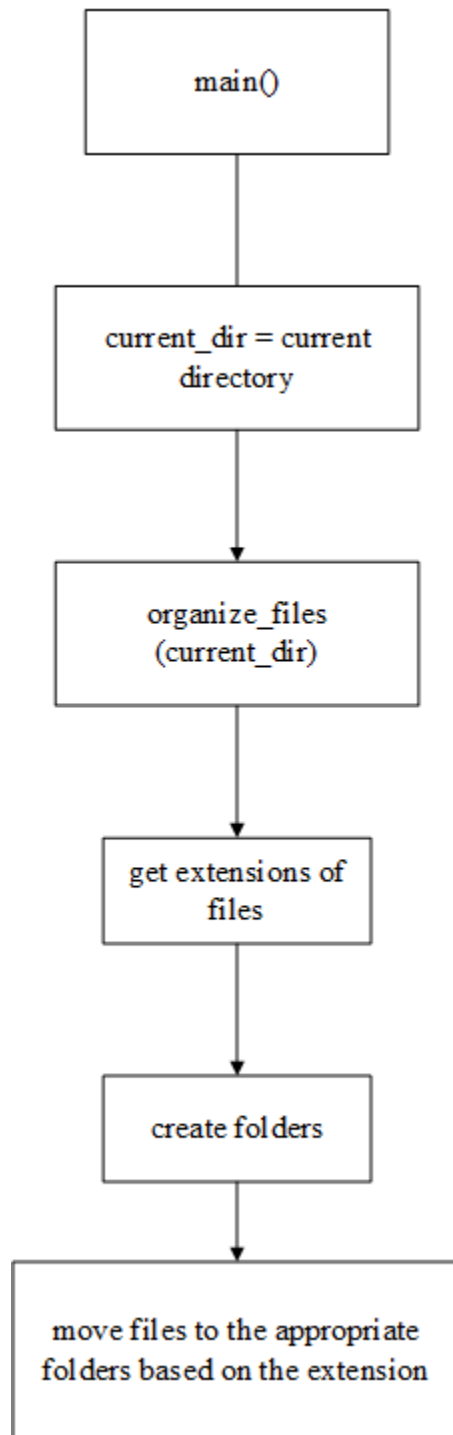


Figure 3.1. Hierarchy Chart.

The "Main Program" represents the top level of the program, and it contains the "organize\_files" function, which is the main function of the program. The "organize\_files" function contains several

sub-tasks, such as getting the files in the directory, creating a dictionary of file types and extensions, creating folders for each file type, and moving files to appropriate folders.

### C. Flowchart

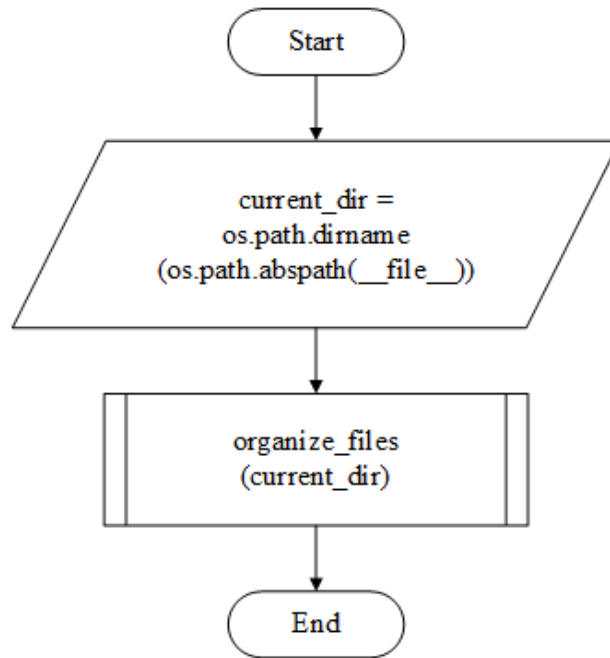


Figure 3.2. Flowchart.

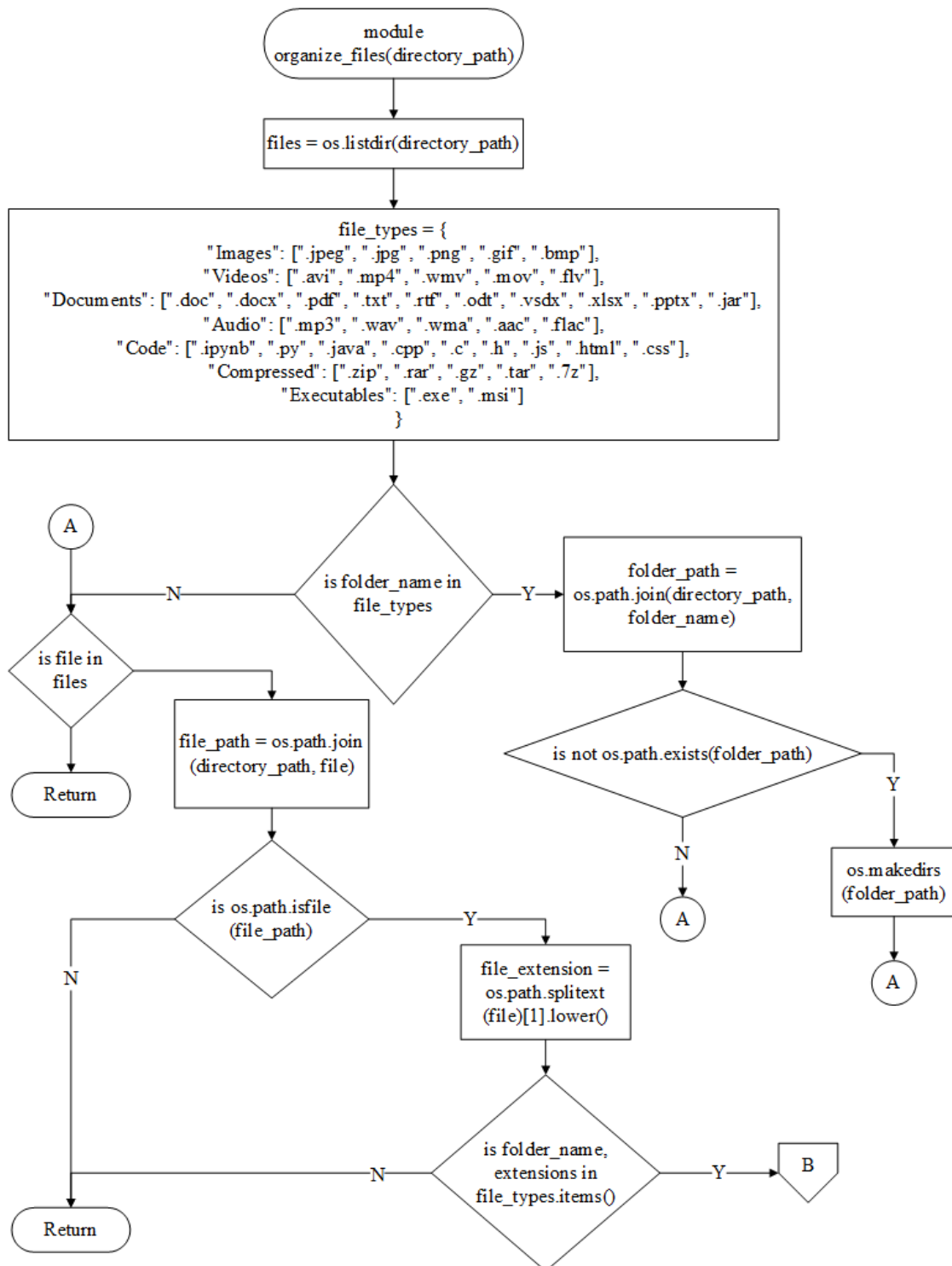


Figure 3.3. Flowchart.

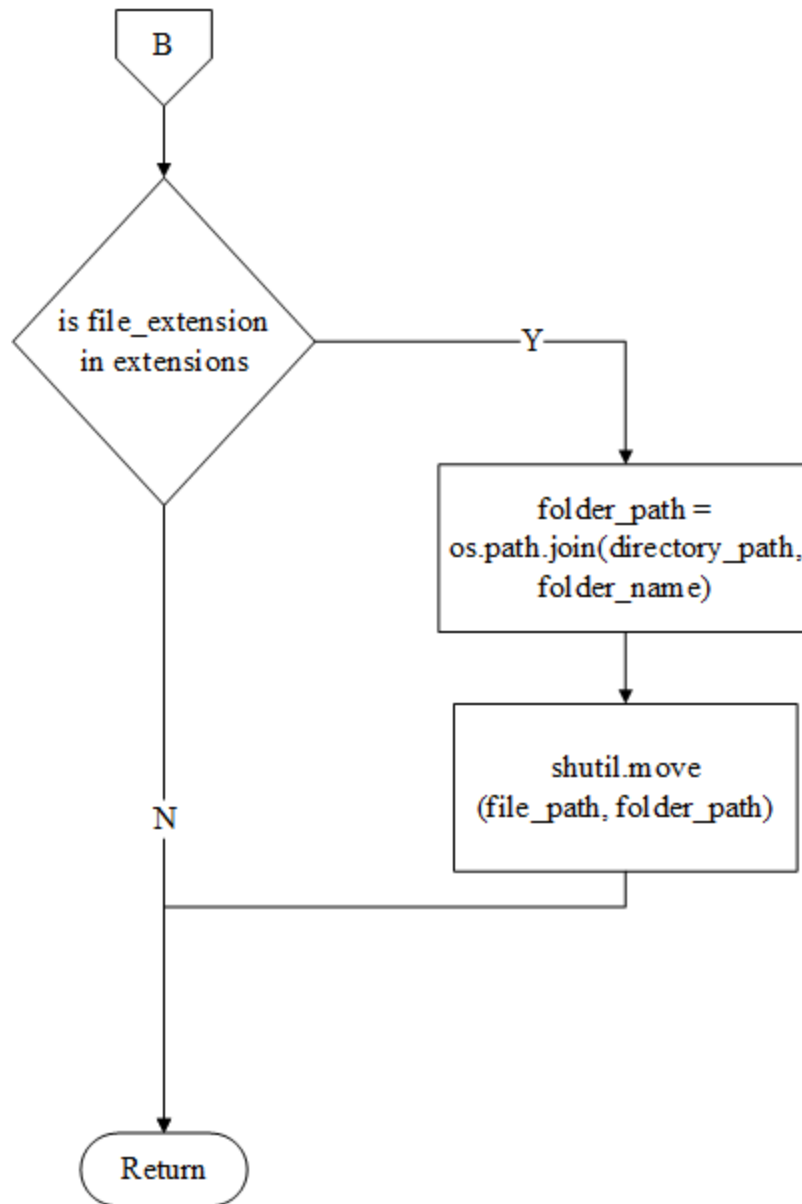


Figure 3.4. Flowchart.

1. **import os** and **import shutil**: These are Python libraries used to interact with the operating system and perform file operations, respectively.
2. **def organize\_files(directory\_path)::** This is a function definition with **directory\_path** as the input parameter, which represents the path of the directory that needs to be organized.
3. **files = os.listdir(directory\_path)**: This line gets a list of all the files in the directory specified by **directory\_path**.

4. **file\_types = {...}**: This is a dictionary that defines different file types and their corresponding file extensions. For example, the "**Images**" file type includes the extensions ".jpeg", ".jpg", ".png", etc.
5. **for folder\_name in file\_types:...**: This loop creates a folder for each file type defined in the **file\_types** dictionary.
6. **for file in files:...**: This loop iterates over each file in the directory, gets its file extension, and moves it to the appropriate folder based on its file type.
7. **file\_extension = os.path.splitext(file)[1].lower()**: This line gets the file extension of the current file by splitting its name from the extension and converting it to lowercase for easier comparison with the file extensions in the **file\_types** dictionary.
8. **shutil.move(file\_path, folder\_path)**: This line moves the current file to the folder that corresponds to its file type.

## D. Pseudocode

```
module organize_files (directory_path)
files = the names of the files in the desired directory using the 'os.listdir' method
'file_types' = dictionary holding the file types and their corresponding file extensions.
For each file type in the 'file_types' dictionary
    if file type folder doesn't exist using the 'os.makedirs' method.
        Create a folder with the name of the file type
    End if
End for
For each file in the 'files' list
    If a file is a file using the 'os.path.isfile' method
        Get the file's extension using the 'os.path.splitext' method
        Store it in the variable 'file_extension'.
    End if
End for
For each file type in the 'file_types' dictionary
    if the 'file_extension' is in the list of extensions for that file type.
        move the file to the corresponding folder using the 'shutil.move' method.
    end if
End for each file type in the 'file_types' dictionary
end module
```

```
current_dir = the directory path of the current script file using the 'os.path.abspath' and
'os.path.dirname' methods
function organize_files(current_dir)
```

## IV. Result

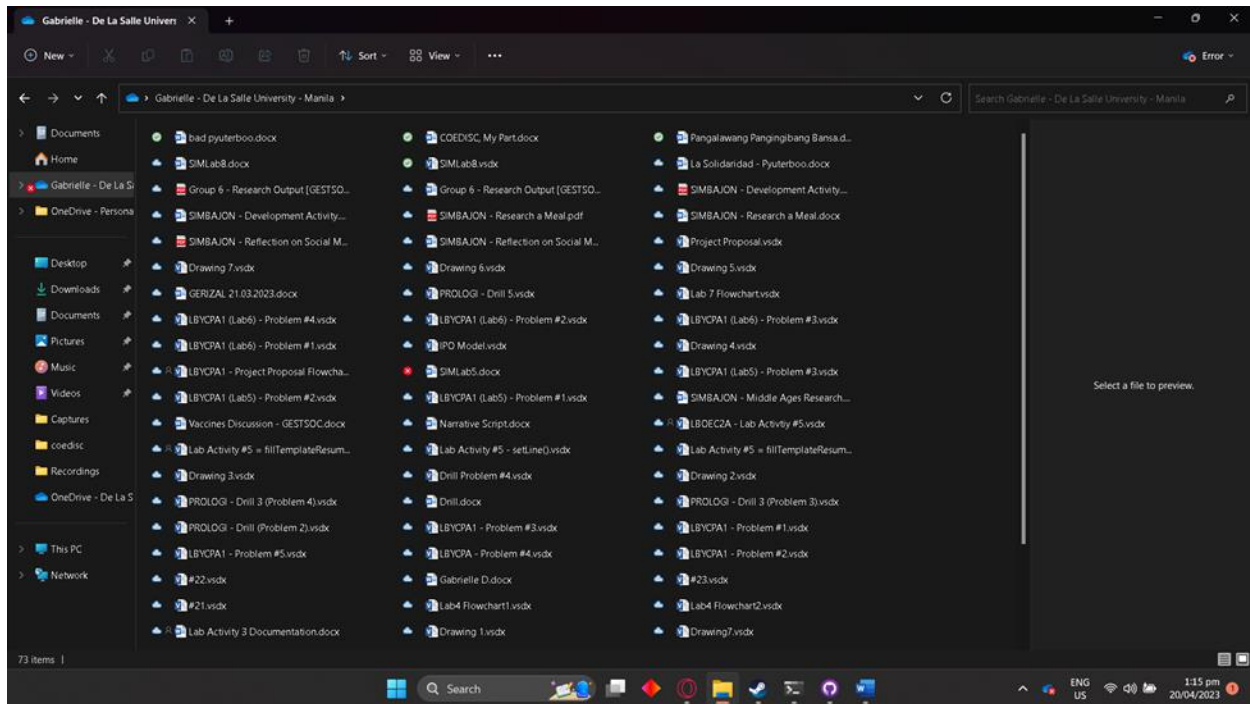


Figure 4. First Test.

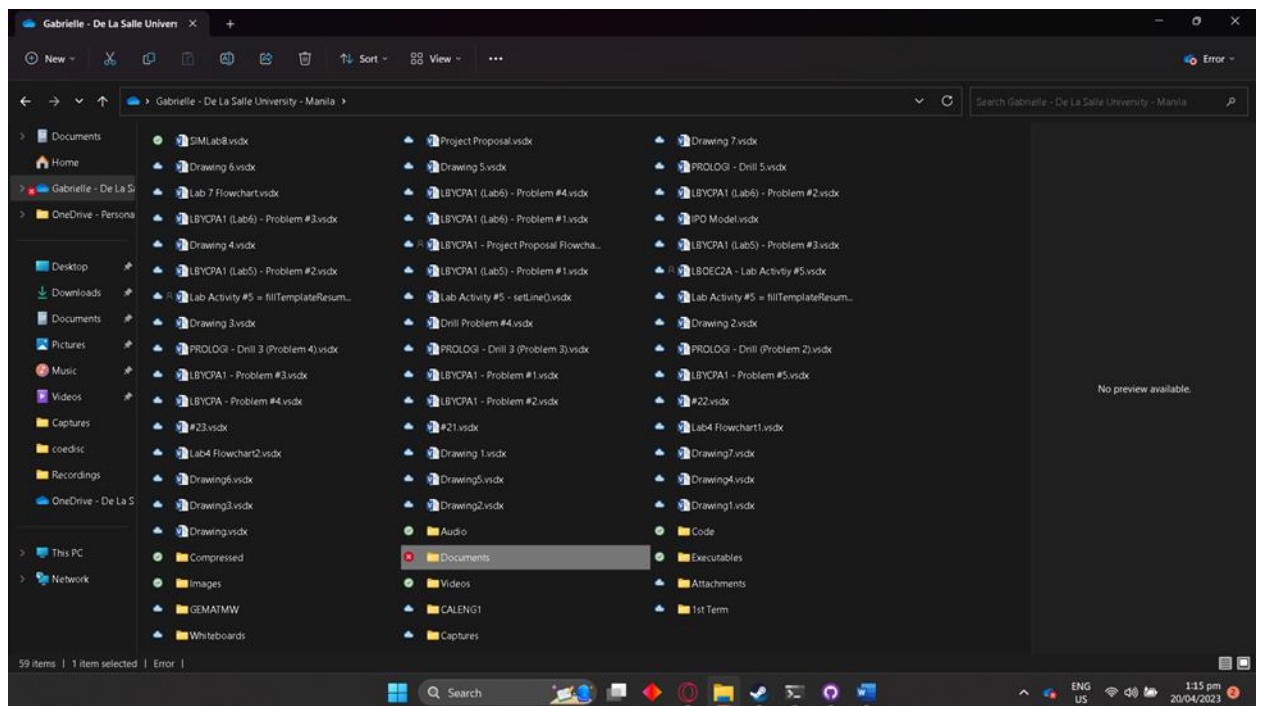


Figure 4.1. First Test.



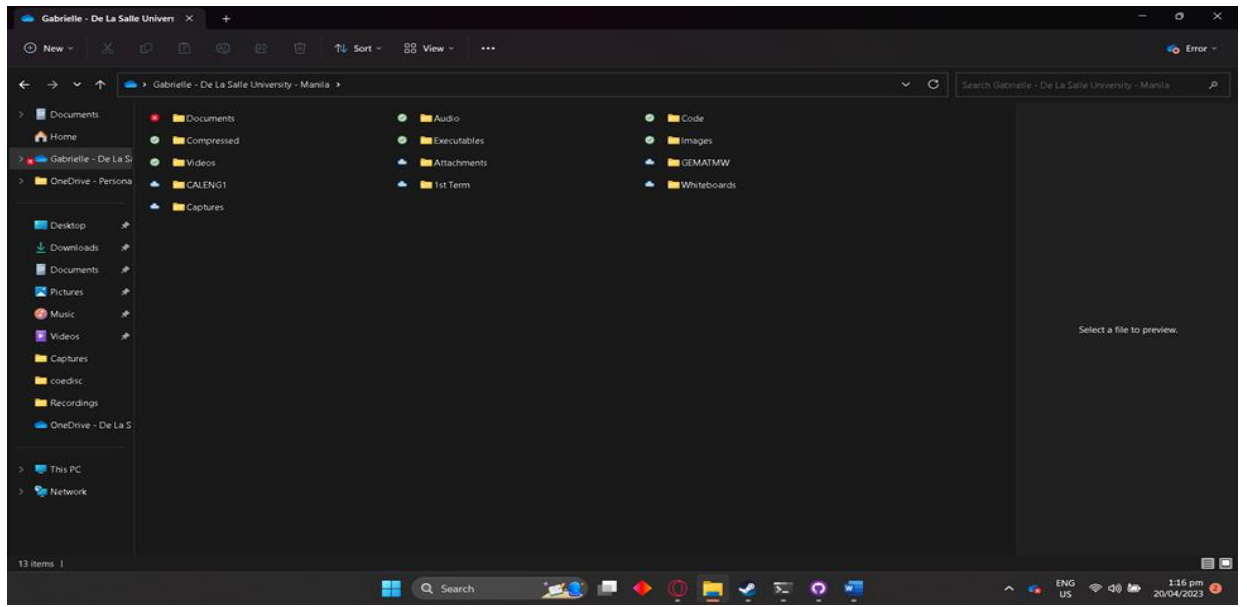


Figure 4.2. First Test - Sorted.

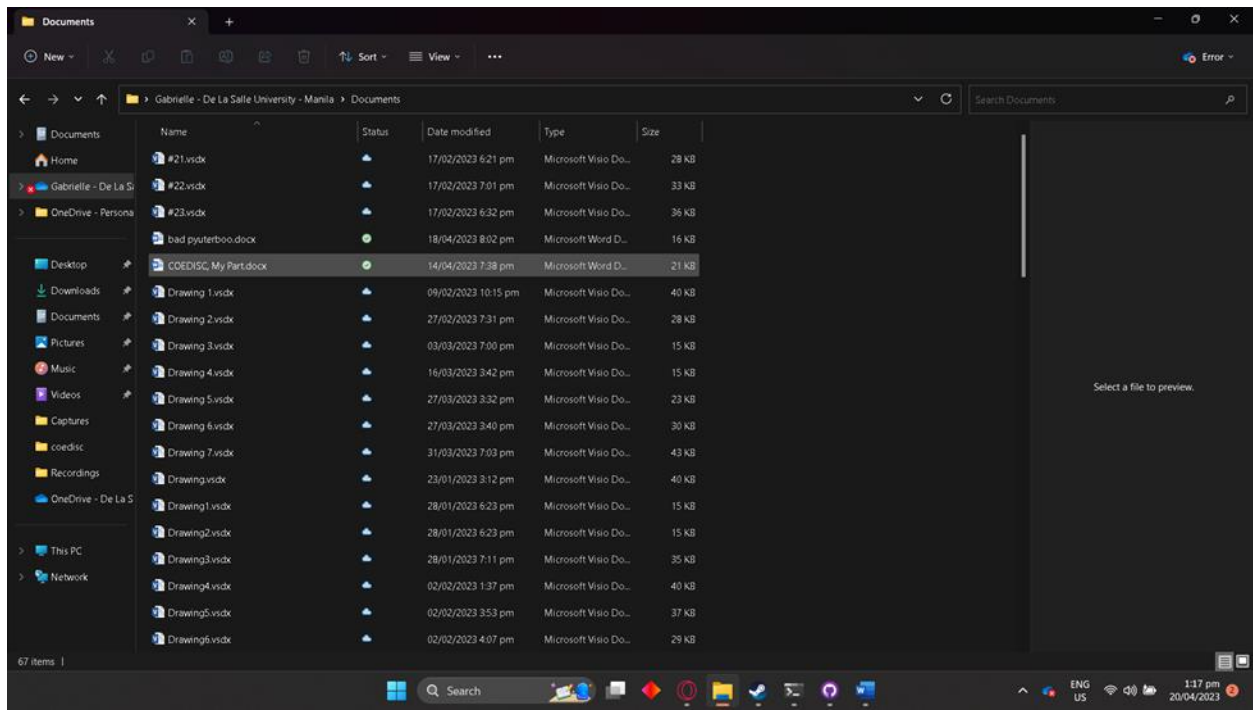


Figure 4.3. First Test - Result.

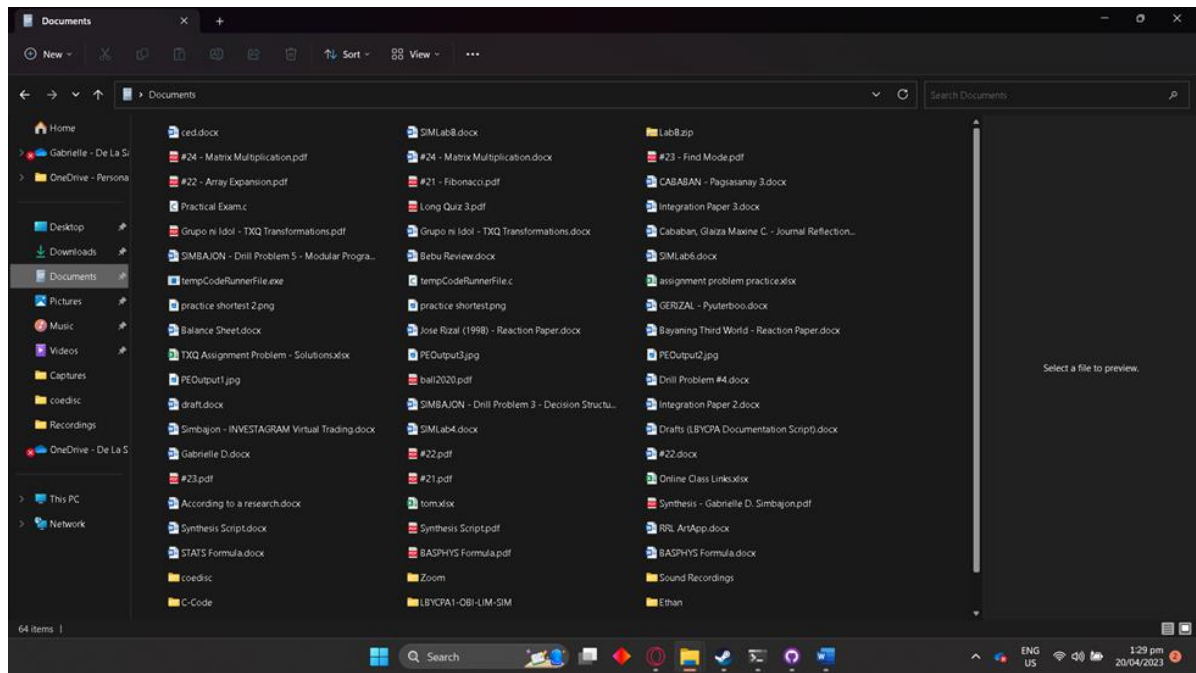


Figure 4.4. Second Test.

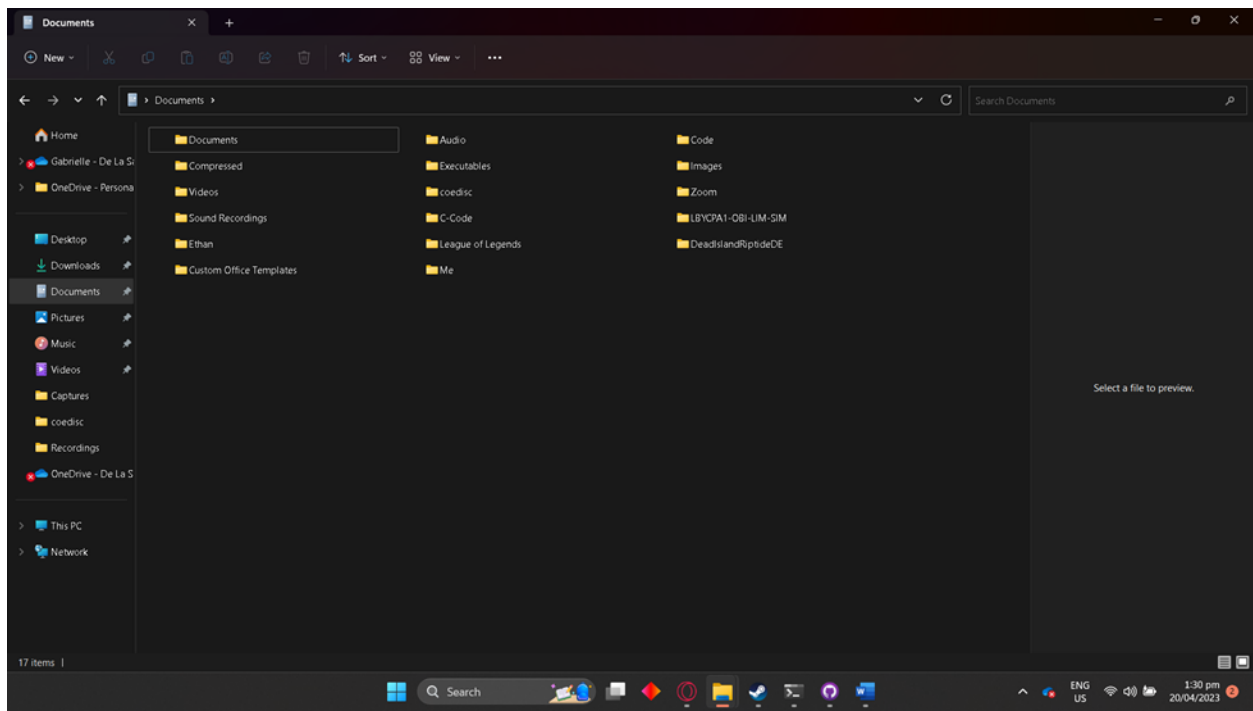


Figure 4.5. Second Test - Sorted.

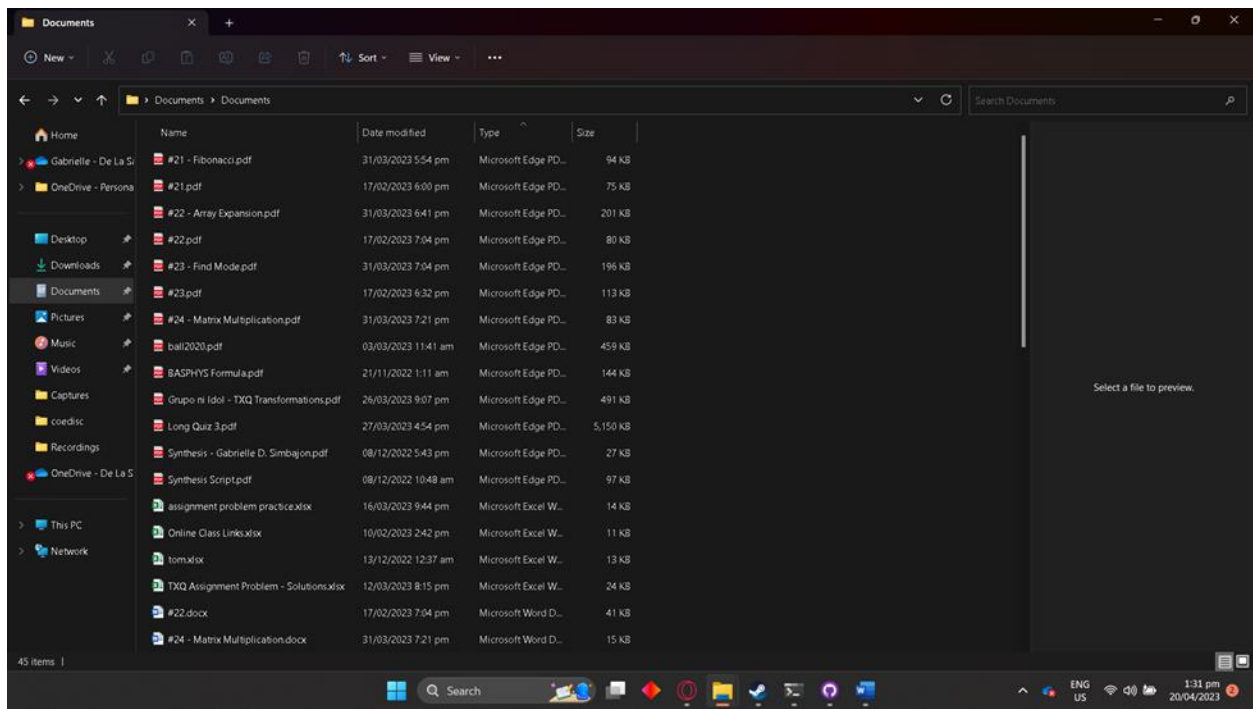


Figure 4.6 Second Test - Result.

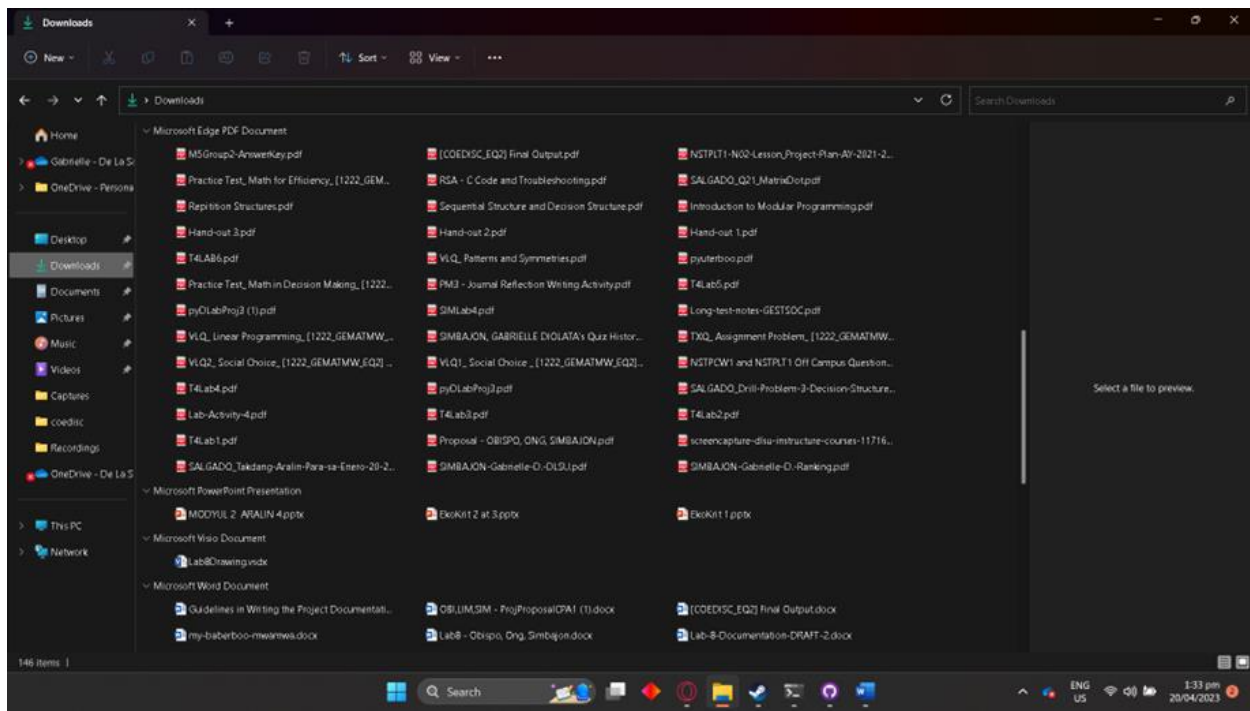


Figure 4.7. Third Test.

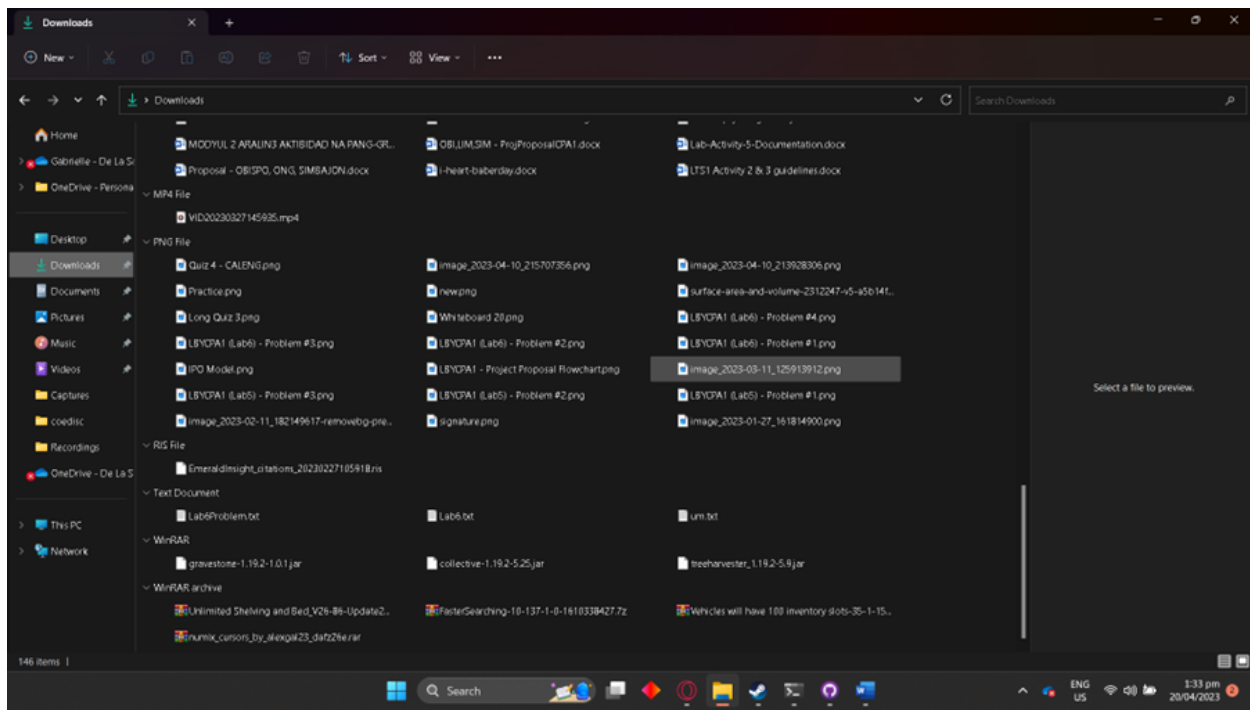


Figure 4.8. Third Test.

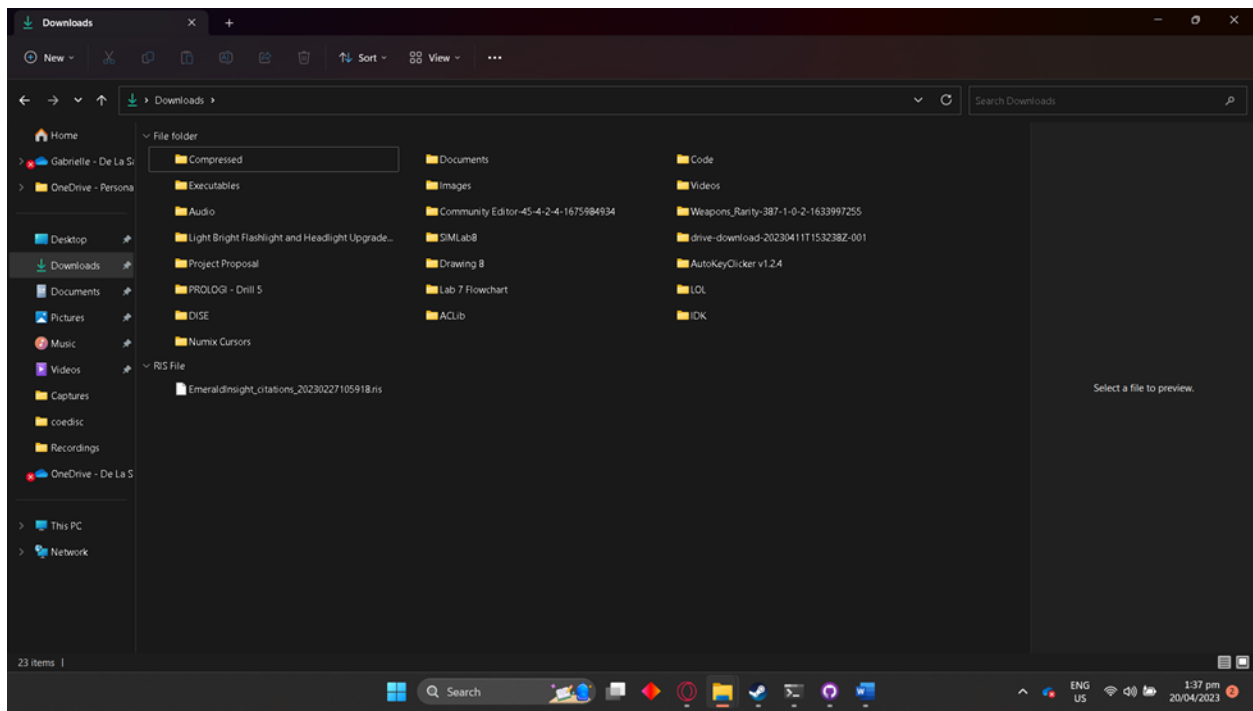


Figure 4.9. Third Test - Sorted.

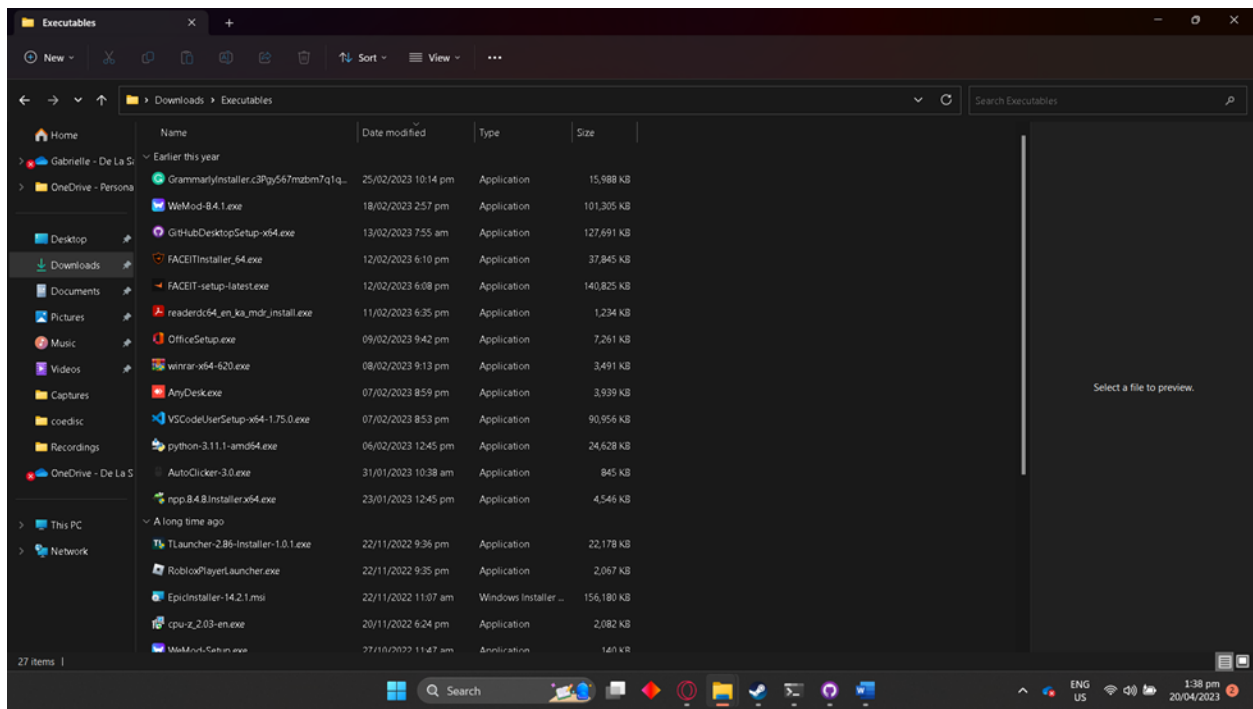


Figure 4.10. Third Test - Result.

## **v. Discussion of Results**

The results above contain three different examples of directories being sorted with the code. For the first execution, first, pick a directory to be sorted then the code will run without .vsdx in the dictionary. Second, the code will now run in the direction with .vsdx in the dictionary. The result of the code should then give you a sorted directory. For the second execution, first, pick a directory to be sorted then the code will run without .xlsx in the dictionary. Second, the code will now run in the direction with .xlsx in the dictionary. The result of the code should then give you a sorted directory. And for the third execution, first, pick a directory to be sorted then the code will run without .pptx, .7z, and .jar in the dictionary. Second, the code will now run in the same direction with .pptx, .7z, and .jar in the dictionary. The result of the code should then give you a sorted directory.

## **vi. Analysis, Conclusion, and Future Directives**

The file organizer program is a simple but effective tool for automatically organizing files based on their file types. The program successfully creates folders for each file type and moves the corresponding files to their appropriate folders. It utilizes a dictionary to hold the file types and their corresponding file extensions, and then iterates through the files in the directory, checking their file extensions and moving them to the appropriate folder. Overall, the program is easy to use and can save users a significant amount of time and effort when organizing their files.

The file organizer program is a useful tool for anyone looking to keep their files organized. By automatically grouping files based on their file types, the program can make it easier for users to find the files they need and can help to reduce clutter in their directories. The program is straightforward to use and requires no special skills or knowledge, making it accessible to a wide range of users.

One area for improvement could be the addition of more file types and extensions to the program's dictionary. This would allow for even more precise organization of files based on their types, making it easier for users to find what they need. Another potential improvement could be the inclusion of a GUI (Graphical User Interface) for the program. Currently, the program is run using a command line interface, which may not be as user-friendly for some users. By developing a GUI, the program can become more accessible to a wider range of users. The program could also benefit from the addition of more user-based sorting options. While the current program allows users to specify the type or name of a file they are looking for, it would be helpful to add additional options, such as sorting by date or size. Finally, the program could potentially be expanded to include more advanced features, such as the ability to automatically rename files based on their content or to delete duplicate files. These features could further streamline the file organization process and make the program even more valuable to users.

## References

- Ahlawat, P., Sankar Biswas, S., & Kumari, S. (2020, February 27). *File\_org - an automated file organization system*. SSRN Scholarly Paper. <https://papers.ssrn.com/abstract=3545224>
- Alter, S. (1979, June). A model for automating file and program design in business application systems. *Communications of the ACM*, 22(6), 345-353. 10.1145/359114.359120
- 8 Biggest Document Management Problems And Their Solutions - GLOBODOX. (2022, September 21). Globodox. Retrieved April 20, 2023, from <https://www.globodox.com/blog/document-management-problems-and-their-solutions/>
- Gandhi, S. (2020, January 20). *Organizing Documents : Challenges & Difficulties*. Sorted AI. Retrieved April 20, 2023, from <https://blog.sortedai.com/productivity/organizing-documents-challenges/>
- Inventor-Miranda, W. C. (2019, April 3). *The problem in the storage of files and documents* | Wilma C. Inventor-Miranda. BusinessMirror. Retrieved April 20, 2023, from <https://businessmirror.com.ph/2019/04/03/the-problem-in-the-storage-of-files-and-documents/>
- Roberts, D. C. (1972). File organization techniques. In *Advances in Computers* (Vol. 12, pp. 115-174). Elsevier. 10.1016/S0065-2458(08)60509-4
- Van Rijsbergen, C.J. (1976). File organization in library automation and information retrieval. *Journal of Documentation*, 32(4), 294-317. 10.1108/eb026630



# Appendices

## A. User's Manual

To use the program, run the python file in the same directory where you want the files to be organized. For example, you want to organize \Users\Gabrielle\Documents\, copy the python file in that same directory and run it.

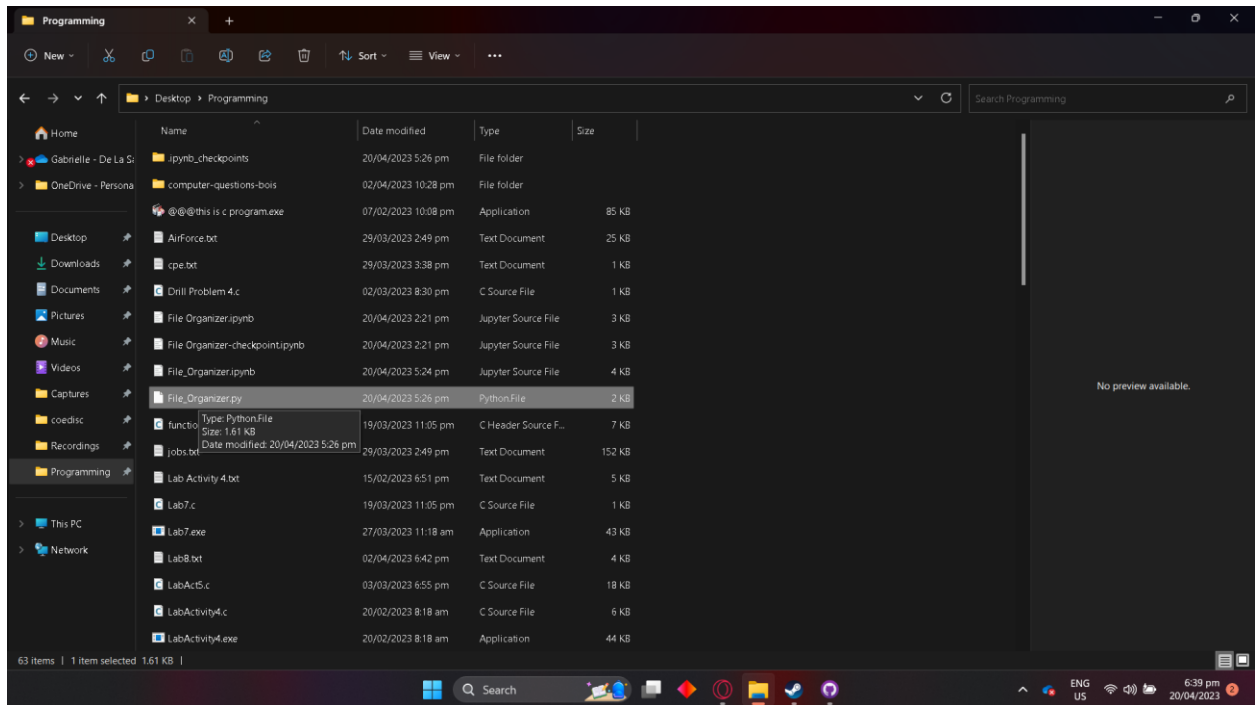


Figure 8. User Guide.

## B. Source Code

```
import os
import shutil

def organize_files(directory_path):
    files = os.listdir(directory_path)

    # Get all the files in the desired directory

    file_types = {
        "Images": [".jpeg", ".jpg", ".png", ".gif", ".bmp"],
        "Videos": [".avi", ".mp4", ".wmv", ".mov", ".flv"],
        "Documents": [".doc", ".docx", ".pdf", ".txt", ".rtf", ".odt", ".vsdx", ".xlsx", ".pptx",
        ".jar"],
        "Audio": [".mp3", ".wav", ".wma", ".aac", ".flac"],
        "Code": [".ipynb", ".py", ".java", ".cpp", ".c", ".h", ".js", ".html", ".css"],
        "Compressed": [".zip", ".rar", ".gz", ".tar", ".7z"],
        "Executables": [".exe", ".msi"]
    }

    # Create a dictionary to hold the file types and their corresponding file extensions

    for folder_name in file_types:
        folder_path = os.path.join(directory_path, folder_name)
        if not os.path.exists(folder_path):
            os.makedirs(folder_path)

    # Create folders for each file type

    for file in files:
```

```
file_path = os.path.join(directory_path, file)
if os.path.isfile(file_path):
    file_extension = os.path.splitext(file)[1].lower()
    for folder_name, extensions in file_types.items():
        if file_extension in extensions:
            folder_path = os.path.join(directory_path, folder_name)
            shutil.move(file_path, folder_path)
            break

# Move files to their appropriate folders

current_dir = os.path.dirname(os.path.abspath(__file__))
organize_files(current_dir)
```

### C. Work Breakdown

Student Name	Tasks Assigned	Percentage of the Work Contribution
LIM, Curt Ethan C.	<ul style="list-style-type: none"> <li>• Table of Contents</li> <li>• List of Figures</li> <li>• Researched References</li> <li>• Methodology - Hierarchy Chart, Conceptual Framework</li> <li>• Discussion of Results</li> <li>• Analysis, Conclusion, and Future Directives</li> <li>• Appendices - Work Breakdown</li> <li>• Poster</li> </ul>	33.33...%
OBISPO, Lance Altheus A.	<ul style="list-style-type: none"> <li>• Programmed the code</li> <li>• Debugged the code</li> <li>• Researched References</li> <li>• Methodology - Flowchart, Pseudocode</li> </ul>	33.33...%

	<ul style="list-style-type: none"> <li>● Appendices - Source Code, Work Breakdown</li> <li>● Proofread the document</li> </ul>	
SIMBAJON, Gabrielle D.	<ul style="list-style-type: none"> <li>● Table of Contents</li> <li>● Introduction - Background of the Study, Problem Statement, Objectives, Significance of the Project</li> <li>● Researched References</li> <li>● Review of Related Literature</li> <li>● Methodology - Conceptual Framework</li> <li>● Result</li> <li>● Appendices - User's Manual, Work Breakdown</li> <li>● Video Recording</li> </ul>	33.33...%

Table 8. Itemized Workload.

#### **D. Personal Data Sheet**



**Gabrielle D. Simbajon** - He is taking the program BSCpE in De La Salle University - Manila. He is from Calamba, Laguna.



**Curt Ethan C. Lim** - Currently a first year college student under the program BS - CPE in De La Salle University – Manila.



**Lance Altheus A. Obispo** - a first year college student under the program Bachelor of Science in Computer Engineering at De La Salle University Manila.