

A  
Mini Project Report on  
**AutoTemporary Cleaner Using Python**

Submitted in partial fulfillment of the requirements for the degree of  
**BACHELOR OF ENGINEERING**  
IN  
**Computer Science & Engineering**  
Artificial Intelligence & Machine Learning By

Pranay Bhere – 24206004  
Prajwal Dhanawade – 24206007  
Om Brahmavale – 24206008  
Gaurav Kshirsagar – 24206009

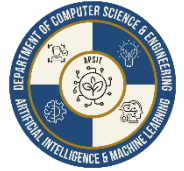
Under the guidance of  
**Prof. Suruchi Ruiwale**



**Department of Computer Science & Engineering**  
**(Artificial Intelligence & Machine Learning)**  
**A. P. Shah Institute of Technology**  
**G. B. Road, Kasarvadavali, Thane (W)-400615**  
**University of Mumbai**  
**2024-2025**



# A. P. SHAH INSTITUTE OF TECHNOLOGY



## CERTIFICATE

This is to certify that the project entitled “**AutoTemporary Cleaner**” is a bonafide work of Om Brahmavale (24206008), Pranay Bhare (24206004), Prajwal Dhanawade (24206007), Gaurav Kshirsagar (24206009) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering in Computer Science & Engineering (Artificial Intelligence & Machine Learning)**.

---

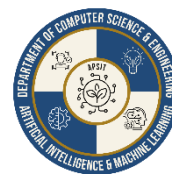
Prof. Suruchi Ruiwale  
Mini Project Guide

---

Dr. Jaya Gupta  
Head of Department



**A. P. SHAH INSTITUTE OF TECHNOLOGY**



## **Project Report Approval**

This Mini project report entitled “**AutoTemporary Cleaner**” Om Brahmvale (24206008), Pranay Bhare (24206004), Prajwal Dhanawade (24206007), Gaurav Kshirsagar (24206009) is approved for the degree of ***Bachelor of Engineering in Computer Science & Engineering, (AI & ML) 2024-25.***

External Examiner: \_\_\_\_\_

Internal Examiner: \_\_\_\_\_

Place: APSIT, Thane Date:

## **Declaration**

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Om Brahmavale  
(24206008)

Pranay Bhere  
(24206004)

Prajwal Dhanawade  
(24206007)

Gaurav Kshirsagar  
(24206009)

## **ABSTRACT**

In an era of digital proliferation, the storage space on computers is continuously challenged by the accumulation of temporary files, cache, and other unnecessary data. The "Auto Temp Cleaner" project is designed to address this issue by providing an efficient and user-friendly solution to clear temporary and unnecessary files from a computer, thereby liberating valuable disk space and enhancing system performance. The software conducts a thorough scan of the computer's storage to identify various types of temporary files, cached data and other unwanted files that consume valuable disk space.

Our Software focuses on clearing/deleting these types of unused and space occupying files on just a click of button.

The project incorporates a simple and intuitive user interface, making it accessible to users of all technical backgrounds.

Users can schedule regular clean-up tasks, ensuring that their computer remains clutter-free and optimized over time.

**Keywords:** Temporary Files, Scheduled deletion

## Index

Index			Page no.
Chapter-1			
	Introduction		1
Chapter-2			
	Literature Survey		
	2.1	Solutions in the Real World	2
	2.2	Survey of the People Using These Solutions	3
Chapter-3			
	Problem Statement		4
Chapter-4			
	Experimental Setup		
	4.1	Hardware Requirements	5
	4.2	Software Setup	5
Chapter-5			
	Proposed system and Implementation		
	5.1	Block Diagram of proposed system	6
	5.2	Description of Block diagram	7
	5.3	Implementation	8-10
		Fig1-2	8
		Fig3-4	9
		Fig5.1-5.2	10
	5.4	Advantages & Applications	11-12
Chapter-6			
	Conclusion		
	6.1	Conclusion	13
	6.2	Future Goals	14
References			15

# **CHAPTER 1**

## **INTRODUCTION**

# **1. INTRODUCTION**

Our project, Auto Temp Cleaner, is an innovative solution aimed at tackling the pressing issue of excessive accumulation of temporary files, caches, and other redundant data.

By offering a user-friendly and efficient method for clearing these unnecessary files from a computer, our goal is to free up valuable disk space and enhance the overall performance of the system.

This project has been crafted to serve the diverse needs of both individuals and organizations, ensuring that they can sustain a streamlined and clutter-free computing environment for uninterrupted productivity.

The cornerstone of our project lies in its provision of three distinct options that grant users the flexibility to manage their temporary files effectively. These options are tailored to cater to various preferences and time requirements, enabling users to select the specific duration within which they wish to have their temporary files, caches, and other surplus data removed.

The primary objective of this project centers on ensuring a seamless deletion process without compromising essential system files. To achieve this, the project has incorporated robust exception handling methods that effectively prevent the deletion of any crucial system files.

This approach guarantees the security and integrity of the system while streamlining the deletion of temporary files and caches.

Furthermore, the application is designed to initiate automatically during system start up, facilitating the seamless processing of temporary files and their subsequent deletion.

This functionality is achieved through strategic manipulation of the Windows registry data, enabling a secure and effective automation process. By implementing this feature, users can enjoy a hassle-free and automated approach to managing their system's temporary files, ensuring optimal performance without any risk to essential system functionalities



# **CHAPTER 2**

## **LITERATURE SURVEY**

## 2. LITERATURE SURVEY

### 1.1 : Solutions in the Real World

Several solutions exist for managing temporary files in operating systems, both built-in and third-party. These tools aim to automate or simplify the cleanup process to improve system performance and free up storage. For e.g. Windows Disk Cleanup: A traditional utility that allows users to manually clean up temporary files, system cache, and other unnecessary files. However, it requires manual intervention and lacks detailed scheduling options.

Despite the wide range of tools available, several issues persist in most existing solutions:

- I) **Manual Intervention Required:** Built-in system utilities like Windows Disk Cleanup often require user input to trigger cleanup processes, lacking true automation.
- II) **System Resource Consumption:** Some third-party applications, such as CCleaner, are known to consume significant system resources during cleanup processes, impacting performance.
- III) **Privacy Concerns:** Many third-party solutions (e.g., CCleaner) have raised concerns over data collection practices, making users hesitant to rely on them.

## 1.2 Survey of the People Using These Solutions

A user survey was conducted to assess the needs and expectations of those using or interested in such tools. Key insights from potential users include:

- I) Automation:** Users prefer tools that automate file deletion without requiring manual intervention.
- II) Simplicity:** Many users (especially non-technical) want straightforward tools with a simple interface.
- III) Privacy:** Security and privacy-conscious users want tools that do not collect unnecessary data or communicate with external servers.
- IV) Minimal System Impact:** Security and privacy-conscious users want tools that do not collect unnecessary data or communicate with external servers.

The surveyed tools rely on various technologies and methodologies to handle temporary file management.

### a) Technologies:

- a. File System Manipulation: Tools like BleachBit directly manipulate the file system to remove unnecessary files.
- b. Operating System APIs: Solutions like Windows Disk Cleanup use system APIs to locate and delete temporary files.
- c. Scheduled Tasks (System-Level): Storage Sense and many third-party solutions leverage system task schedulers (e.g., Windows Task Scheduler) for running cleanup tasks at regular intervals.

### b) Methodologies:

- a. Direct File Deletion: Many solutions simply remove files from temporary directories, using built-in functions such as `os.remove()` and `shutil.rmtree()` (as implemented in your program).
- b. Scheduling: Some solutions, like Storage Sense, run automatic cleanup routines based on predefined time intervals or file age.

# **CHAPTER 3**

## **Problem Statement**

## **2. Problem Statement**

**Problem Identification:** The increase in temporary files, cached data, and unnecessary documents overwhelms computer systems, consuming critical disk space.

**Performance Impact:** Accumulation of these files leads to slow boot times, lagging applications, and decreased system responsiveness.

**User Challenges:** Users experience frustration due to the negative effects of clutter on their systems.

**Project Objective:** The project aims to provide effective solutions to mitigate the impact of digital clutter, ensuring a smoother and optimized computing experience.

# **CHAPTER 4**

## **Experimental Setup**

## **4. Experiment Setup**

### **1. Hardware Requirements**

- System Specifications:
  - A computer with at least:
    - Processor: intel i3 CPU or equivalent.
    - RAM: Minimum 2 GB
    - Storage: At least 100 MB of free space for testing file operations
  - Operating System: Windows 7 or higher (to ensure compatibility with the targeted directories)

### **2. Software Requirements**

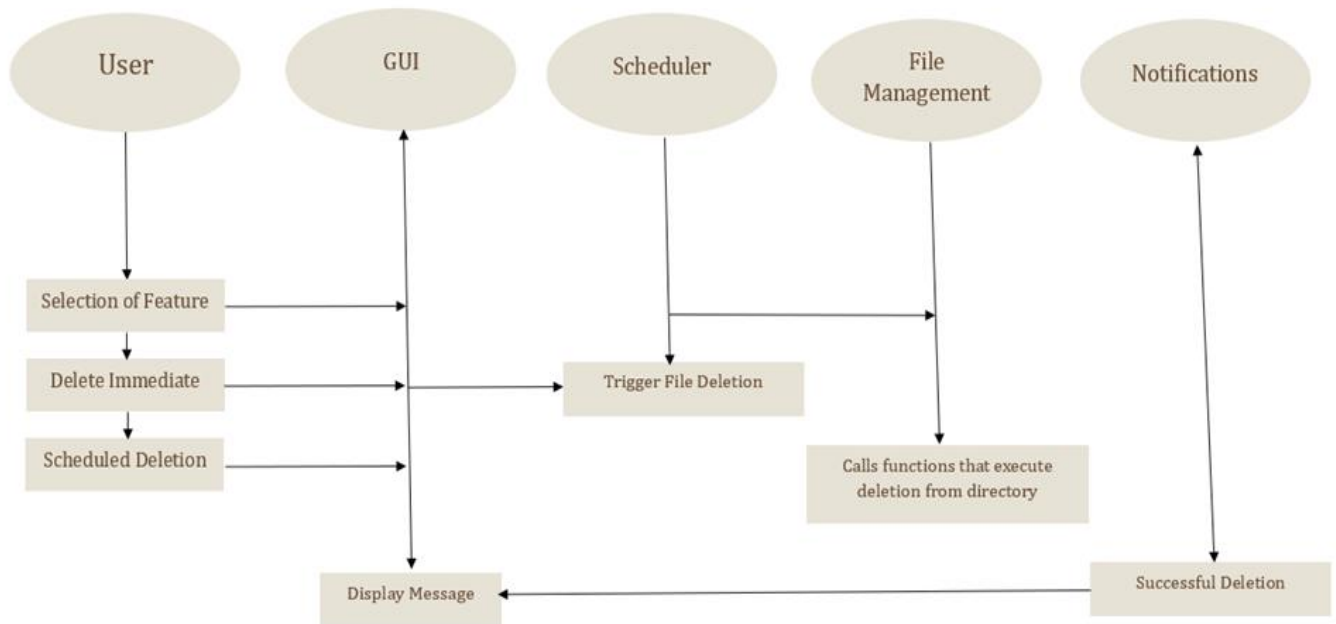
- Development Environment:
  - Python: Version 3.8 or higher
  - Libraries:
    - PySide6 (for GUI development)
    - Schedule (for task scheduling)
    - Other standard libraries (os, shutil, threading, etc.)

# **CHAPTER 5**

## **Proposed System & Implementation**



## 5.1 Block diagram of proposed system



## 5.2 Description of block diagram

The system follows a Modular Architecture where each component performs a specific function. This design ensures that the application is flexible, maintainable, and scalable.

**1. GUI Module:** Provides an interface for users to interact with the application (create schedules, manually delete files, etc.)

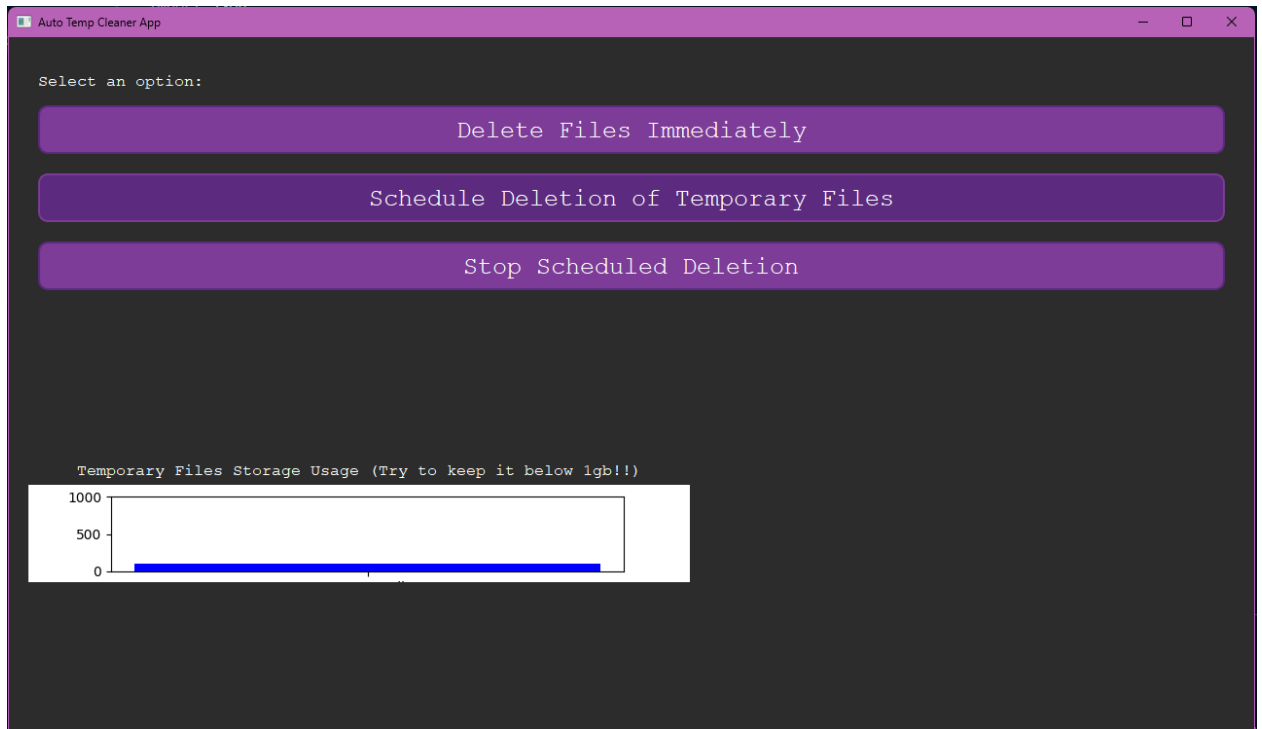
**2. File Management Module:** Handles all file and directory operations (deleting files, directories, and subdirectories). Handles all file and directory operations (deleting files, directories, and subdirectories).

**3. Scheduling Module:** Responsible for managing and executing automated tasks like scheduled deletion at specified intervals.

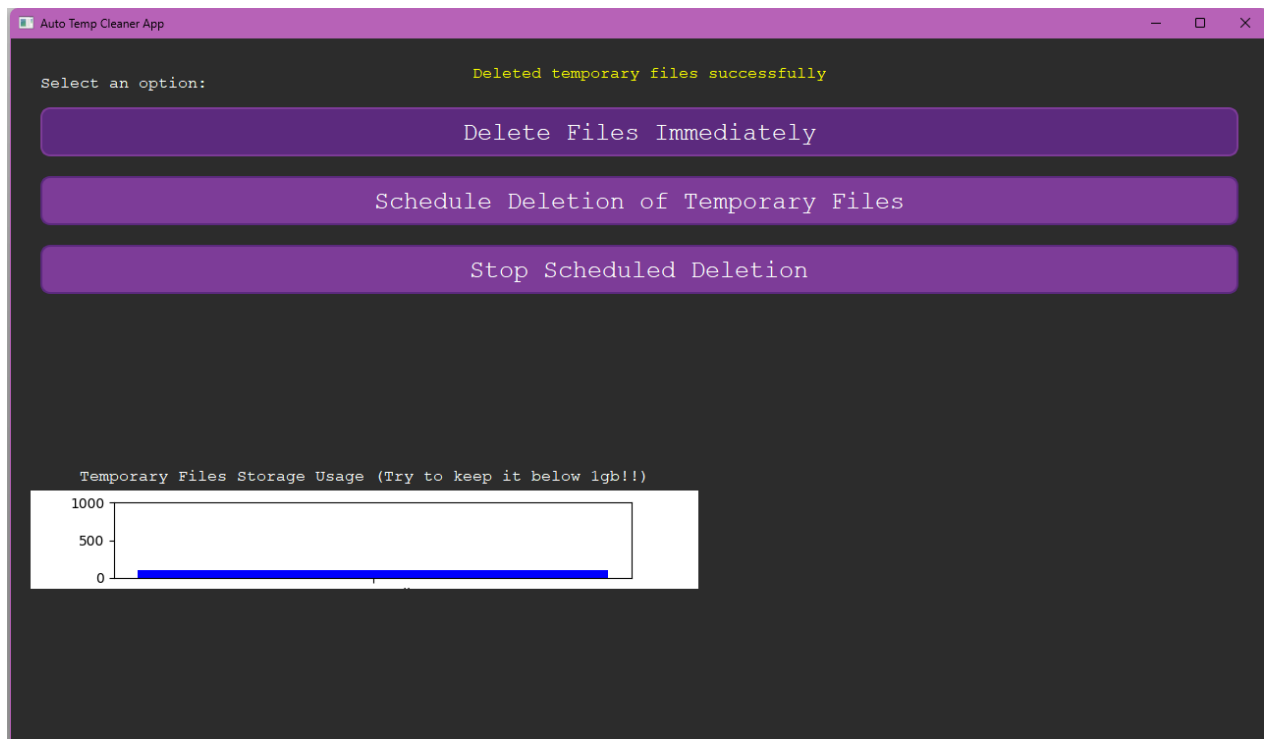
**4. Threading Module:** Ensures the scheduler runs concurrently with the user interface, allowing smooth, uninterrupted operation of the app.

**5. Notification System:** Displays messages to inform users about the success or failure of file deletions.

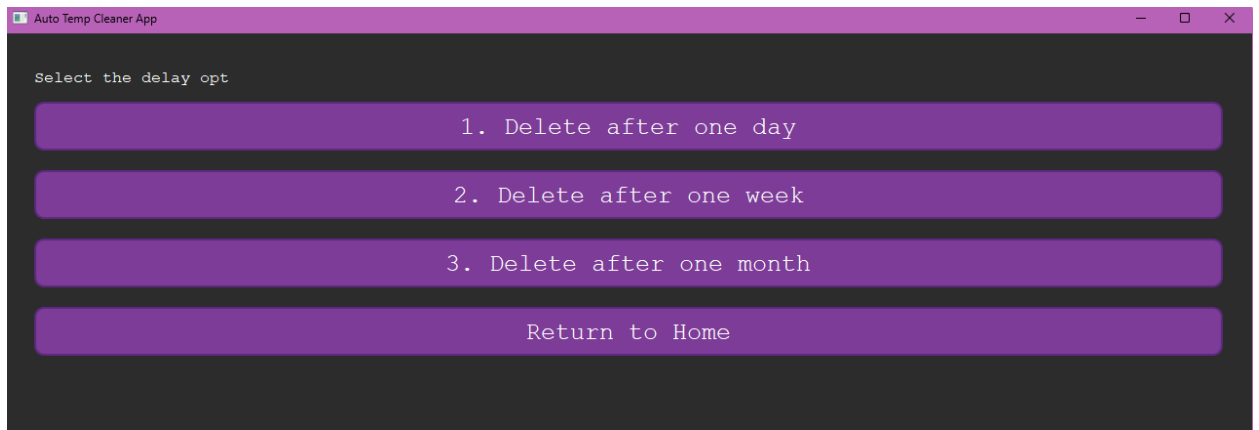
## 5.3 Implementation



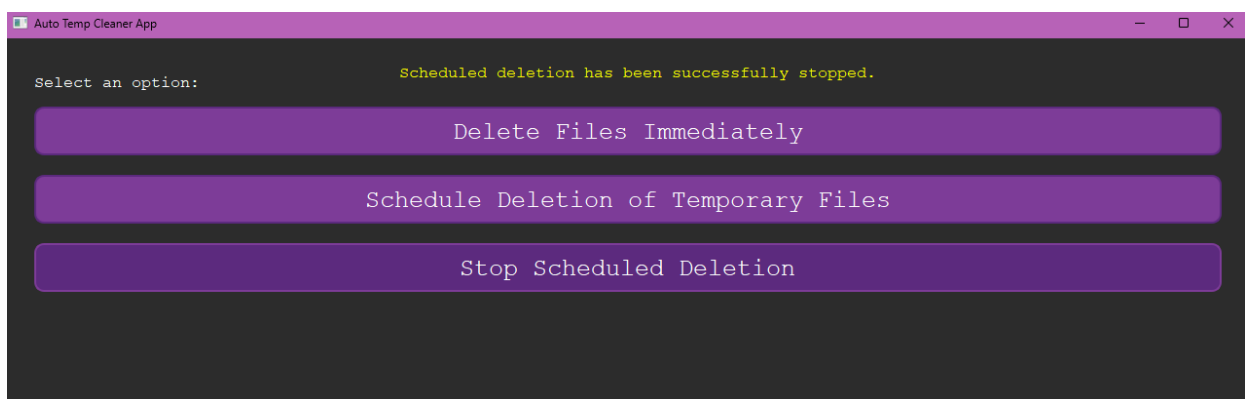
**Fig 1. Home Page**



**Fig 2. Deleting Temp Files Immediately**



**Fig 3. Scheduling Automatic Deletion**



**Fig 4. Stopping Current Scheduled Deletion**

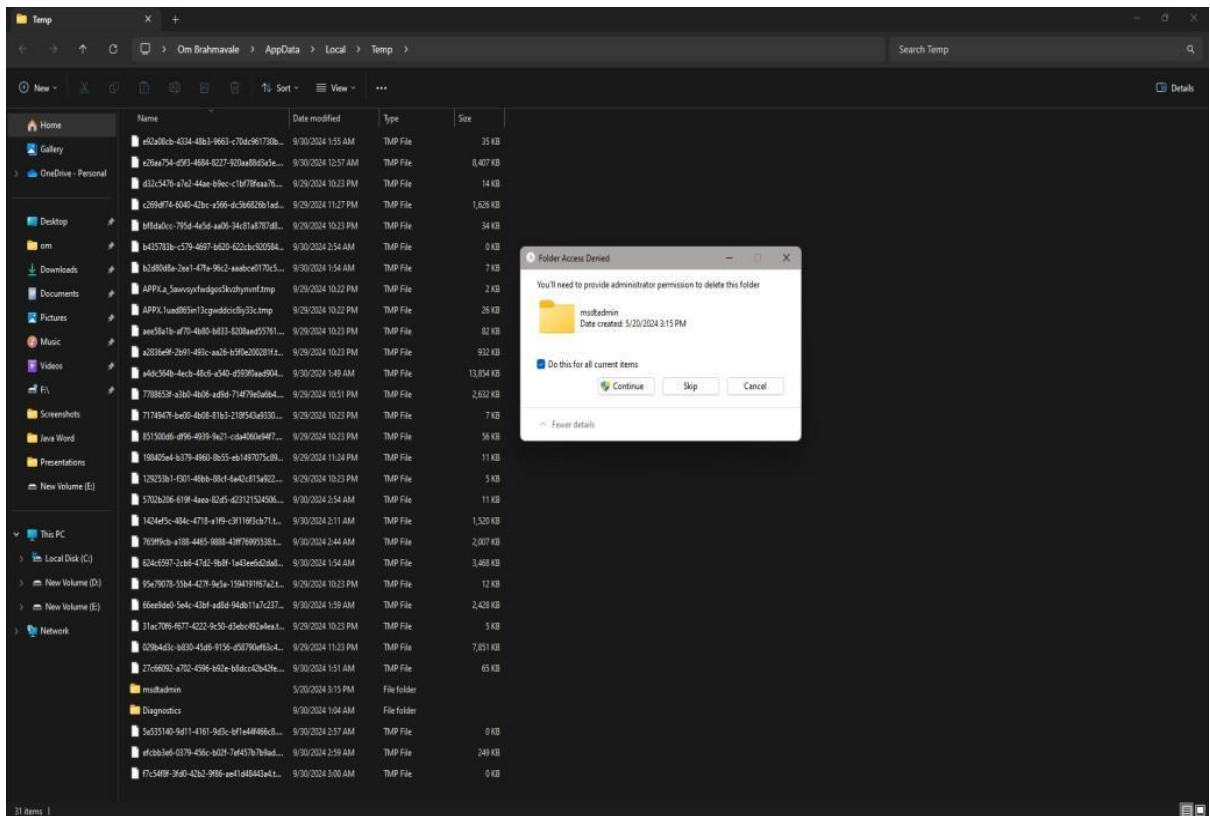


Fig 5.1 Exceptions and Deleted file folders

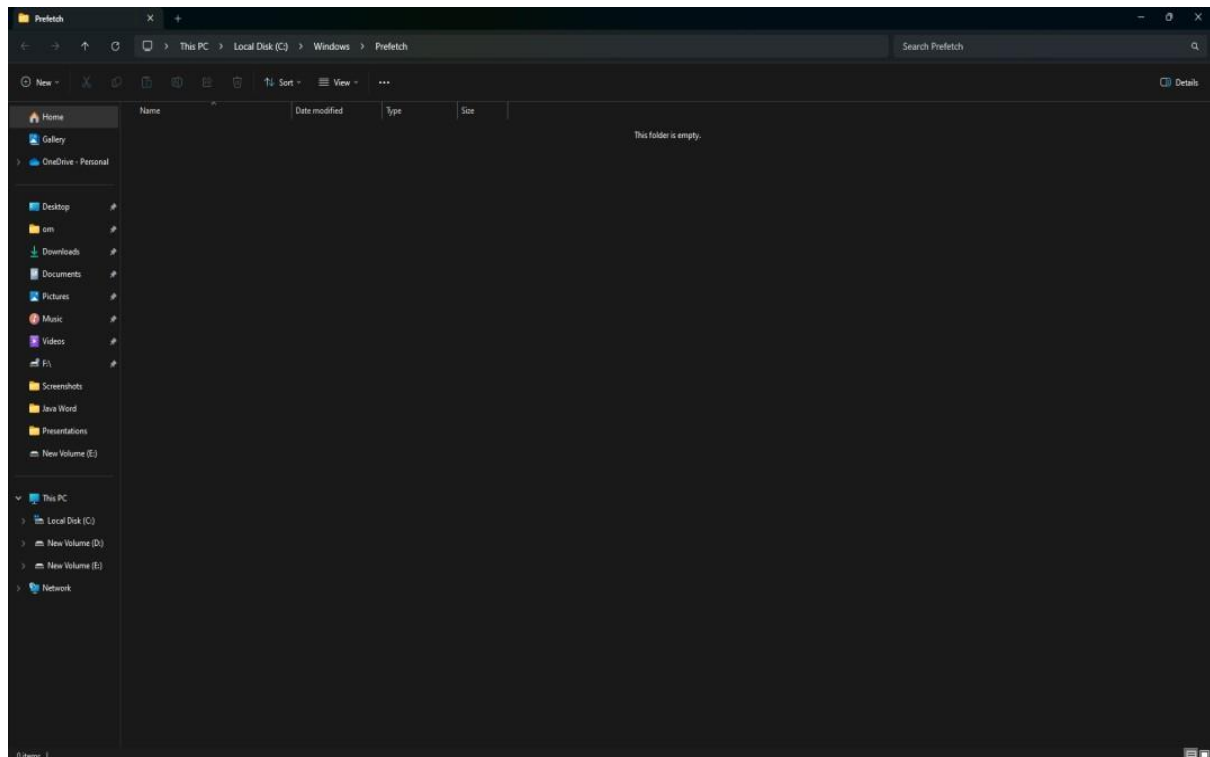


Fig 5.2. Exceptions and Deleted file folders

## **5.4 Advantages and Applications**

### **Advantages:**

#### **1. System Performance Improvement:**

Regular deletion of temporary files can free up significant disk space, leading to improved system performance. Excessive temporary files can slow down system operations and affect application responsiveness.

#### **2. Automation of File Management:**

The application automates the tedious process of manually deleting temporary files. This saves users time and effort, allowing them to focus on more critical tasks.

#### **3. Enhanced User Experience:**

The GUI developed with PySide6 provides a user-friendly interface that makes it easy for users of all technical levels to operate the application. This accessibility promotes broader use and adherence to good file management practices.

#### **4. Error Handling and Safety:**

The application includes error handling mechanisms that ensure safe deletion of files while notifying users of any issues, such as permission problems. This reduces the risk of accidental data loss.

## **Applications:**

### **1. Personal Computers:**

Home users can utilize this application to maintain their personal computers by regularly cleaning up temporary files, which can improve performance and prolong the life of their devices.

### **2. Enterprise Environments:**

Organizations can implement this solution on their workstations to automate file management across multiple devices, leading to more efficient IT resource management and reduced storage costs.

### **3. Software Development:**

Developers can integrate this functionality into applications that require regular cleanup of temporary files, thereby enhancing the overall user experience and reducing the risk of performance degradation.

### **4. System Administrators:**

IT professionals can use the application for system maintenance tasks, ensuring that temporary files do not consume unnecessary storage space on servers or shared workstations.

# **CHAPTER 6**

## **Conclusion**



## **6.1 Conclusion**

**Addressing Digital Demand:** The application effectively addresses the growing concern of accumulating temporary and cached data in the digital landscape.

**Impact on Performance:** The accumulation of such data adversely affects storage capacity and overall system performance.

**Low Consumption of CPU and Privacy :** It almost takes no CPU resources to run since the GUI is sleek and simple and there are no AD's or any other data collecting modules so it keeps the user data safe.

**Storage Reclamation:** It empowers users to reclaim valuable storage space, leading to noticeable improvements in system speed and responsiveness.

**Drawbacks:** No backups of files that are deleted, the application's GUI is not the best, no history.

## 6.2 Future Goals

- **Enhancing the GUI of the app:**
  - GUI can be made better for a good user experience by using a different front-end language which has a vaster option of elements.
- **Adding custom directories for deletion:**
  - An option that can delete a whole drive or a folder specified by user can be added to the application.
- **Adding custom time scheduling option:**
  - Custom timer options according to the users for more optimal and better scheduled deletion.

## References

**1. Pooja Bhardwaj, Chahil Choudhury, Priya Batra**

A Comparative Study of Popular Libraries and their Application

IEEE Xplore, 2023.

<https://ieeexplore.ieee.org/document/10390032>

**2. Andrew Miller, Nicholas Kobylski, Elias Qamar, Jimmy Xiao, Nicholas Veal, Richard Kenney, Nicholas Wysocki, Mohammed Mahmoud.**

Automating File Operations via Python

IEEE Xplore, 2022,

<https://ieeexplore.ieee.org/document/10216551>

**3. Viji Rajendran V, Denil C Verghese, Mohammed Arshu P T, Subin T G**

A Python Library for Automated Machine Learning

IEE Xplore, 2021,

<https://ieeexplore.ieee.org/document/9544859>