DISK SPACE ANALYZER

MINI PROJECT REPORT

By

SURYANSH SAWARIYA(RA2212702010017)
PRANAV ARYA (RA2212702010016)
AAKARSH SHARMA(RA2212702010027)

Under the guidance of

Dr. Kanipriya M

In partial fulfillment for the Course

of

21CSC202J– Operating Systems

in the Department of Computational Intelligence

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report for the course **21CSC202J - OPERATING SYSTEMS** entitled in "**DISK SPACE ANALYZER** " is the bonafide work of **Suryansh Sawariya (RA2212702010017), Pranav Arya (RA2212702010017)** and **Aakarsh Sharma (RA2212702010027)** who carried out the work under my supervision.

18/11/23

SIGNATURE

Dr. Kanipriya M

OS – Course Faculty

Assistant Professor

Department of Computational Intelligence

SRM Institute of Science and Technology

Kattankulathur

SIGNATURE

Dr Annie Uthra R

Head of the Department

Professor

Department of Computational Intelligence

SRM Institute of Science and Technology

Kattankulathur

ABSTRACT

A disk space analyzer is a software tool used to inspect, visualize, and manage storage usage on computer drives or storage devices. It scans the contents of a selected drive or specific folders, then generates visual representations or detailed reports illustrating the distribution of files and folders by size.

By presenting data in visual charts, graphs, or lists, these tools enable users to identify large, redundant, or obsolete files occupying substantial storage space. With this insight, users can make informed decisions about which files to delete, move, compress, or organize, optimizing storage efficiency.

Some disk space analyzers offer advanced functionalities like historical comparisons of disk usage, allowing users to track changes over time. They might also provide suggestions for optimizing storage and improving overall system performance.

Overall, disk space analyzers serve as invaluable aids in decluttering storage, identifying wasted space, and maintaining an organized and efficient storage environment.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 OBJECTIVE

The primary objective of a disk space analyzer is to provide users with a comprehensive understanding of how storage space is utilized within a computer or storage device. Its key objectives include:

1. **Visualization of Storage Usage:** Presenting a clear and intuitive visual representation of the distribution of files and folders by size, enabling users to easily identify large or space-consuming items.

2. **Identifying Space Hogs:** Helping users pinpoint large, redundant, or unnecessary files and directories that occupy significant storage space, allowing for
efficient cleanup and optimization.

3. **Detailed Analysis:** Offering detailed reports or breakdowns of storage usage, facilitating informed decision-making on which files or folders to delete, move, compress, or reorganize.

4. **Storage Optimization:** Providing insights and recommendations for optimizing storage usage, potentially improving system performance and freeing up valuable space.

5**. User-Friendly Interface:** Offering an intuitive interface for users to navigate through the storage analysis results, facilitating ease of use and efficient management of files and directories.

6. **Additional Features:** Some disk space analyzers may include features like historical comparisons, allowing users to track changes in storage usage over time, or suggestions for better storage management practices.

1.2 INTRODUCTION

A disk space analyzer is a software tool designed to help users visualize and manage the utilization of storage space on their computer's hard drive or any storage device. Its primary function is to scan the drive or specific folders, analyze the content, and provide a detailed breakdown of how space is being used.

These analyzers present information in various formats like visual charts, graphs, or lists, showing the distribution of files and folders by size. This breakdown helps users identify large or unnecessary files taking up valuable space, enabling them to make informed decisions about what to delete, move, or compress to free up storage.

Some disk space analyzers offer additional features, such as the ability to directly delete or move files, compare different snapshots of disk usage over time, and even suggest optimizations to improve storage efficiency.

Overall, disk space analyzers are valuable tools for maintaining a healthy and organized storage system, preventing clutter and optimizing space utilization on computers or storage devices.

CHAPTER-2

SOFTWARE/HARDWARE REQUIREMENT

Hardware Requirements:

Processor (CPU): Disk space analyzers are generally not very demanding on the CPU. A modern processor, even a basic one, should suffice for running these tools efficiently.

RAM (Memory): Adequate RAM is essential for smooth operation, especially when analyzing large volumes of data. A minimum of 2GB of RAM is usually sufficient, but more is recommended for handling larger datasets.

Storage: Since disk space analyzers primarily deal with storage, having enough free space on the drive where the software is installed is necessary. Additionally, enough space should be available for temporary files created during the analysis process.

Display: A standard monitor with a resolution of at least 1024x768 is typically suitable for displaying the graphical representations generated by disk space analyzers.

Software Requirements:

Operating System: Disk space analyzers are designed to work on various operating systems. Commonly supported systems include Windows, macOS, and Linux. Ensure compatibility with your specific operating system.

Permissions: The software needs sufficient permissions to access and analyze the files and folders on the disk. Ensure that the user account running the disk space analyzer has the necessary permissions.

CHAPTER 3

CONCEPS/WORKING PRINCIPLE

The working principle of a disk space analyzer involves several steps:

1. Scanning: The analyzer initiates by scanning the selected drive or specific folders. It traverses through the file system, gathering information about files, folders, their sizes, and attributes.

2. Data Collection: As the scan progresses, the analyzer collects metadata such as file sizes, types, creation dates, and other attributes. This data is compiled for analysis.

3. Analysis and Visualization: After scanning, the analyzer processes the collected data and organizes it into a visual representation. This representation might include graphical charts, pie diagrams, or a hierarchical folder structure, showing the distribution of file sizes or types.

4. User Interface: The tool then presents this information in a user-friendly interface, allowing users to interact with the visualized data. Users can explore folders, view file sizes, and identify large or space-consuming files easily.

5. User Actions: Based on the presented data, users can take actions within the analyzer interface. This may include deleting, moving, compressing, or organizing files and folders directly from the analyzer.

6. Optional Features: Advanced analyzers might offer additional features, such as historical comparisons, allowing users to track changes in storage usage over time or providing recommendations for optimizing storage.

7. Feedback and Updates: Some analyzers might allow users to rescan periodically to track changes and offer updated insights into the evolving storage landscape.

Overall, the disk space analyzer operates by scanning, collecting, analyzing, and presenting data in a user-friendly manner to empower users to manage their storage efficiently.

CHAPTER-4

APPROACH/METHODOLOGY/PROGRAM

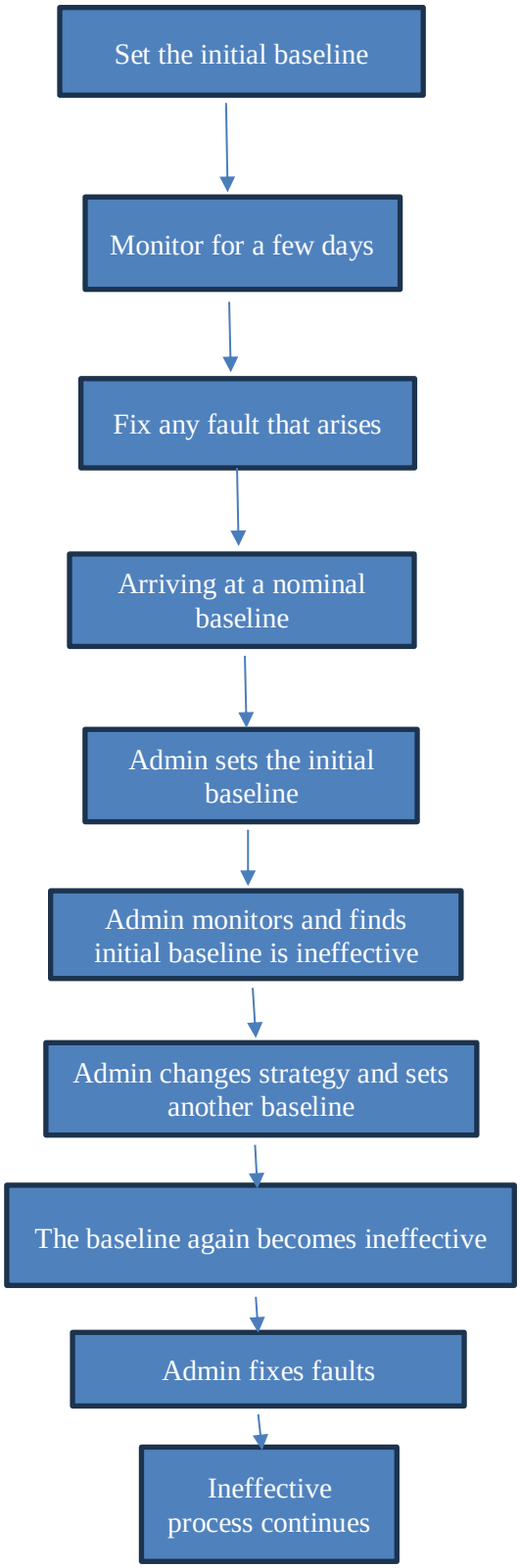
```
from tkinter import *
from psutil import disk_partitions,disk_usage,virtual_memory,cpu_percent
from tabulate import tabulate
window=Tk()
window.geometry("900x600")
window.title("DISK - ANALYZER")
# function to display CPU information
def show_cpu_info():
  cpu_use=cpu_percent(interval=1)
  cpu_label.config(text='{} %'.format(cpu_use))
  cpu_label.after(200,show_cpu_info)
# Function converter Bytes to Gigabytes
def conversor_bytes_to_gb(byte):
 one_gigabyte=1073741824  # bytes
 giga=byte/one_gigabyte
 giga='{0:.1F}'.format(giga)
 return giga
#Function to display RAM information
def show_ram_info():
 ram_usage=virtual_memory()
 ram_usage=dict(ram_usage._asdict())
 #print(ram_usage)
 for key in ram_usage:
    if key!='percent':
       ram_usage[key]=conversor_bytes_to_gb(ram_usage[key])
 ram_label.config(text='{} GB / {} GB ({}
%)'.format(ram_usage["used"],ram_usage["total"],ram_usage["percent"]))
 ram_label.after(200,show_ram_info)
```

```
data=disk_partitions(all=False)
def details(device_name):
 for i in data:
   if i.device==device_name:      return i
# Function that returns the disk partitions
def get_device_name():
 return [i.device for i in data]
# Function that returns the disk information
def disk_info(device_name):
 disk_info={}try:
   usage=disk_usage(device_name)
   disk_info['Device']=device_name
   disk_info['Total']=f"{conversor_bytes_to_gb(usage.used+usage.free)} GB"
   disk_info['Used']=f"{conversor_bytes_to_gb(usage.used)} GB"
   disk_info['Free']=f"{conversor_bytes_to_gb(usage.free)} GB"
   disk_info['Percent']=f"{usage.percent} GB"
 except PermissionError:  pass
 except FileNotFoundError:pass
 info=details(device_name)
 disk_info.update({"Device":info.device})
 disk_info["Mount Point"]=info.mountpoint
 disk_info["FS-Type"]=info.fstype
 disk_info["FS-Type"]=info.opts
 return disk_info
# Function to return information of all partitions
def all_disk_info():
 return_all=[]
 for i in get_device_name()
 return_all.append(disk_info(i))
 return return_all
# Title program
```

```
title_program=Label(window, text='Disk Analyzer', font="arial 40 bold",fg='#14747F')
title_program.place(x=110,y=20)
# CPU title
cpu_title_label=Label(window, text='CPU usage: ',font="arial 24 bold", fg='#FA5125')
cpu_title_label.place(x=20,y=155)
# Label to show percent of CPU
cpu_label=Label(window,bg='#071C1E', fg='#FA5125', font="Arial 30 bold", width=20)
cpu_label.place(x=230,y=150)
# RAM title
ram_title_label=Label(window, text='RAM Usage: ',font="arial 24 bold", fg='#34A96C')
ram_title_label.place(x=20,y=255)
# Label to show percent of ram
ram_label=Label(window,bg='#071C1E',fg='#FA5125', font="Arial 30 bold", width=20)
ram_label.place(x=230,y=250)
# Disk title
disk_title_label=Label(window, text='Disk Usage: ',font="arial 24 bold",fg='#797E1E')
disk_title_label.place(x=350, y=360)
# Text area for disk information
textArea=Text(window,bg="#071C1E", fg="yellow",
width=85,height=6,padx=10,font=("consolas",14))
textArea.place(x=15,y=410)
if _name=='__main_':
    show_cpu_info()
    show_ram_info()
    info=all_disk_info()
    _list=[i.values() for i in info]
    infoTabulated=tabulate(_list,headers=info[0].keys(),tablefmt="simple",missingval="-")
    print(info)
    print(infoTabulated)
    textArea.insert(END,infoTabulated)
window.mainloop()
```

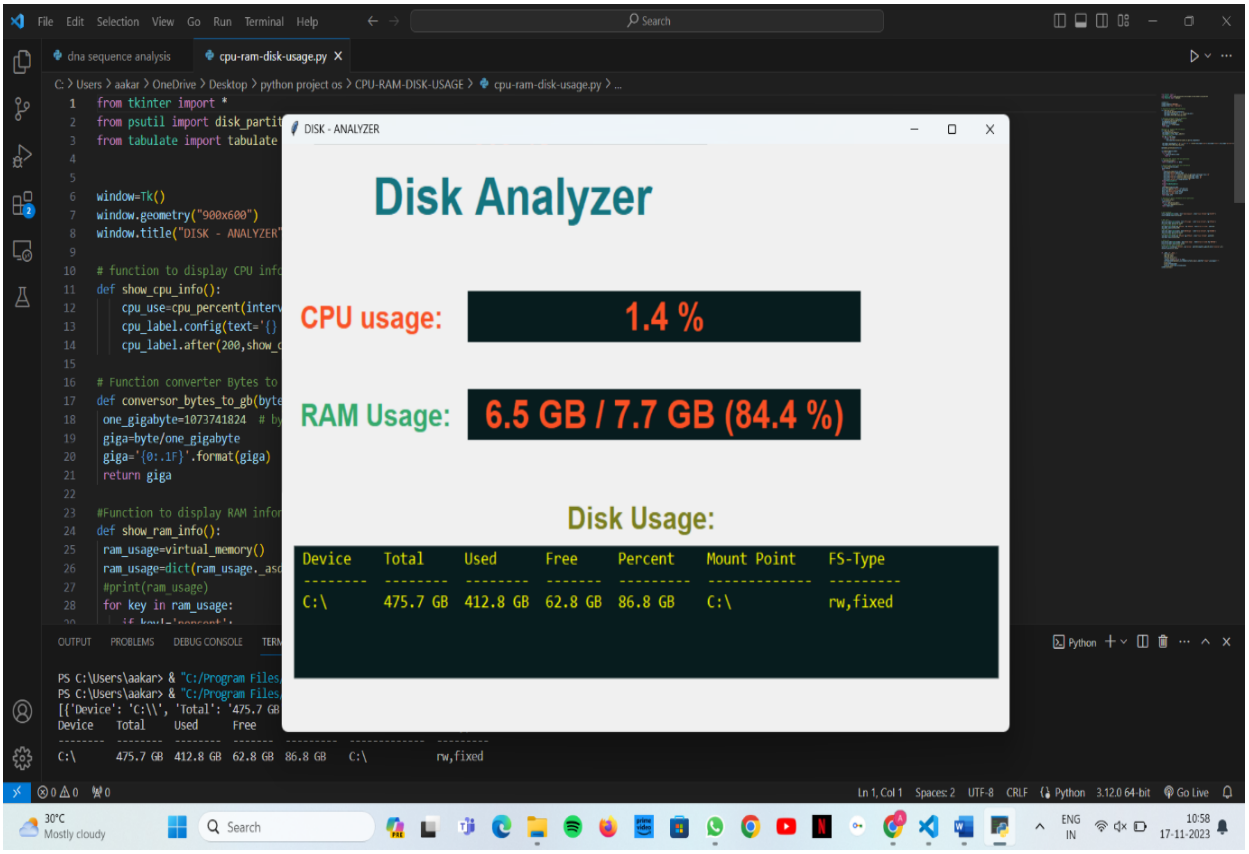
CHAPTER-5

FLOWCHART



CHAPTER-6

OUTPUT



CHAPTER-7

CONCLUSION

In conclusion, a disk space analyzer is a valuable tool for users seeking a comprehensive understanding of how storage space is utilized on their computer or storage device. Through its systematic scanning, data collection, and visual representation, the analyzer empowers users to make informed decisions about disk space management. The key features, such as sorting, filtering, and cleanup recommendations, enhance the efficiency of storage optimization. By providing a user-friendly interface and actionable insights, disk space analyzers contribute to maintaining an organized file system and ensuring optimal system performance. As digital environments continue to accumulate data, the significance of these tools in facilitating effective storage management remains paramount.

CHAPTER-8

REFERENCES

- **GitHub Repository**
- **http://www.google.com**
- **http://www.w3schools.com**
- https://wiki.gnome.org/Apps/SystemMonitor/MergeWithUsage