

ITP-MINOR-AUGUST

M. Mohan Kumar

bannu5353@gmail.com

6301431848

Description

- The main moto of this project is to solve few problems of an Engineering student.
- While working on this project I have noticed many of my friends are using online File Organizer, Number System Converter, Temp Cleaner. So, I have decided to make these tools offline.

File Organizer

- A common problem I have noticed for a student in any degree is finding the desired files.
- By using the program, I have designed, files that have been acquired from various sources like Whatsapp, Mail... can be sorted based on their extensions such as jpeg, png, pdf, mp4, jpg, h5, py... and they are moved to their corresponding folders named after their extensions of the selected directory.
- This makes students to find their desired files easy and saves their time.

Number System Converter

- As an ECE student I have to regularly work with Number systems of base 3 or 4 or 6 and there is no online tool to convert the numbers with above bases.
- Manual conversion consumes a lot of time. So, I have come with a Number system converter that can convert any number system to the number system of desired base.

Temporary File Cleaner

- While working with applications like VS code, Pycharm, Sublime Text... memory gets filled up with Temporary files which are related to execution of the program.
- These are created with every single script we write and occupies memory which makes our systems slower day by day.
- Where manual deletion of these files includes many steps. To solve this difficulty of users I have designed this temporary file cleaner.

Code

Main.py

```
from customtkinter import *

from Number_System_Converter import N_S_C

from Temp_Cleaner import Temp_clean

from File_Organizer import File_organizer

main = CTk()

main.geometry('350x350')

set_appearance_mode("System")

set_default_color_theme("green")

CTkLabel(main, text="Choose Operation", font=('Arial', 14)).pack(padx=10, pady=10)

CTkButton(main, text="Number System Converter", command=N_S_C).pack(padx=10, pady=10)

CTkButton(main, text="Temporary Files Cleaner".center(23), command=Temp_clean).pack(padx=10, pady=10)

CTkButton(main, text="Organize Files".center(23), command=File_organizer).pack(padx=10, pady=10)

main.title("Main")

main.mainloop()
```

Number_System_Converter.py

```
from customtkinter import *

def ToDecimal() -> str:

    global inp_val

    global inp_base

    global out_base

    num = (inp_val.get()).upper()

    base = inp_base.get()

    dec_val = 0

    if '.' in num:

        num, dec = num.split('.')
```

```

for i, j in enumerate(dec):
    if j.isalpha():
        j = str(ord(j) - 55)
        dec_val += int(j) * (base ** -(i + 1))
    print(dec_val)
num_val = str(int(num, base=base))
if dec_val != 0:
    text = num_val + str(dec_val)[1:]
else:
    text = num_val
if out_base.get() == 10:
    empty_label.configure(text=text)
else:
    empty_label.configure(text=ToOther(text, base=out_base.get()))
def ToOther(num: str, base: int):
    res: list[int, ...] = list()
    dec: str = '0'
    dec_val: list[int, ...] = list()
    if '.' in num:
        num, dec = num.split('.')
        dec = float('0.' + dec)
        int_med = []
        while dec not in int_med and dec > 0:
            int_med.append(dec)
            rem = int(dec := (dec * base))
            dec -= rem
            dec_val.append(rem) if rem < 10 else dec_val.append(chr(rem + 55))
    num = int(num)
    while num > 0:
        rem = num % base
        if rem < 10:
            res.insert(0, str(rem))
        else:

```

```

        res.insert(0, chr(rem + 55))

    num //= base

    return ''.join(res) if dec == '0' else ''.join(res) + '.' + ''.join(map(str, dec_val))

win = CTk()
win.geometry('500x500')

inp_base = IntVar()
inp_val = StringVar()
out_base = IntVar()

CTkLabel(win, text="Answer is : ", font=('Arial', 14)).grid(row=4, column=0)

empty_label = CTkLabel(win, text_color='black', text='No Answer', font=('Arial', 24))
empty_label.grid(row=4, column=1, columnspan=3)

def N_S_C():
    global inp_val
    global inp_base
    global out_base
    global win

    try:
        while True:
            CTkLabel(win, text="Base From : ").grid(row=0, column=0)
            CTkEntry(win, textvariable=inp_base, font=('Arial', 14)).grid(row=0, column=1)
            CTkLabel(win, text="Value : ").grid(row=1, column=0)
            CTkEntry(win, textvariable=inp_val, font=('Arial', 14)).grid(row=1, column=1)
            CTkLabel(win, text="Base To : ").grid(row=2, column=0)
            CTkEntry(win, textvariable=out_base, font=('Arial', 14)).grid(row=2, column=1)
            CTkButton(win, command=ToDecimal, hover_color='blue').grid(row=3, column=1)

            win.mainloop()

    except Exception as e:
        print("Invalid Input", e)

    finally:
        print("Exited....")

```

Temp_Cleaner.py

```
from os import system as sys

def Temp_clean():

    try:

        sys(r'del /q /f /s %temp%\*')

        sys(r'del /s /q C:\\Windows\\temp\\*')

    except Exception as e:

        print(e)

        input("Press Enter to Retry....")

    finally:

        print('Done...')
```

File_Organizer.py

```
import os

import shutil

from customtkinter import filedialog

def File_organizer():

    path = filedialog.askdirectory(title="Select Folder")

    list_of_files = os.listdir(path)

    for files in list_of_files:

        name, extension = os.path.splitext(files)

        extension = extension[1:]

        if os.path.exists(path + '/' + extension):

            shutil.move(path + '/' + files, path + '/' + extension + '/' + files)

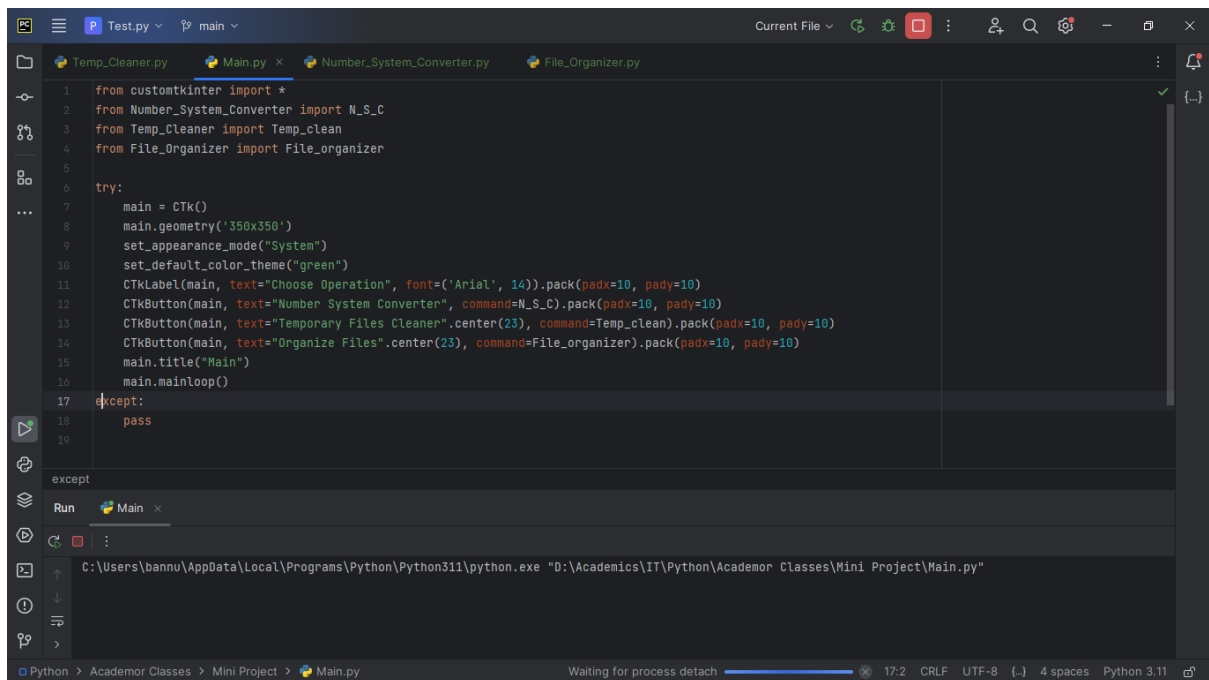
        else:

            os.makedirs(path + '/' + extension)

            shutil.move(path + '/' + files, path + '/' + extension + '/' + files)
```

Images of the Project

Script Images



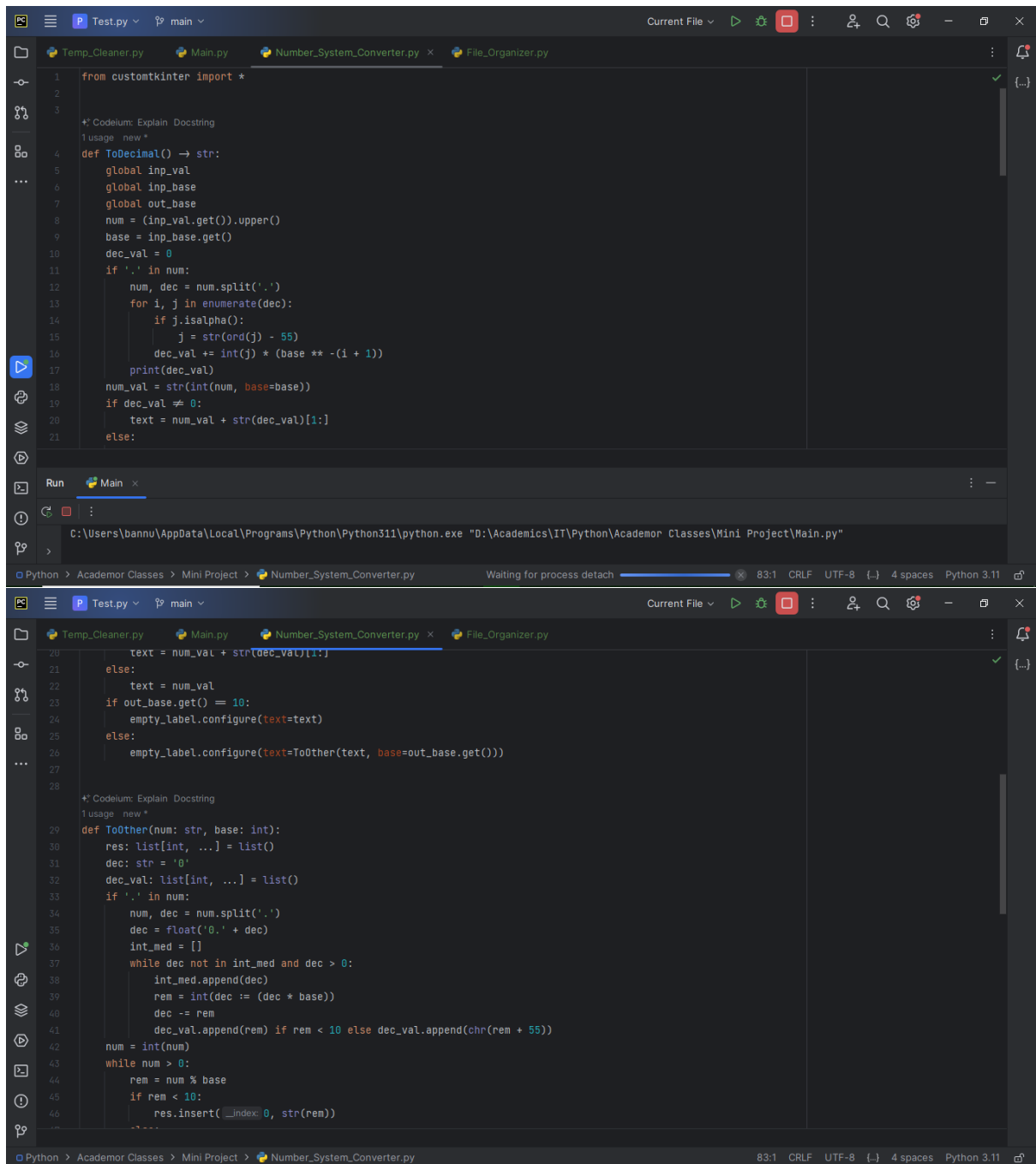
```
1 from customtkinter import *
2 from Number_System_Converter import N_S_C
3 from Temp_Cleaner import Temp_clean
4 from File_Organizer import File_organizer
5
6 try:
7     main = CTk()
8     main.geometry('350x350')
9     set_appearance_mode("System")
10    set_default_color_theme("green")
11    CTkLabel(main, text="Choose Operation", font=('Arial', 14)).pack(padx=10, pady=10)
12    CTkButton(main, text="Number System Converter", command=N_S_C).pack(padx=10, pady=10)
13    CTkButton(main, text="Temporary Files Cleaner", command=Temp_clean).pack(padx=10, pady=10)
14    CTkButton(main, text="Organize Files", command=File_organizer).pack(padx=10, pady=10)
15    main.title("Main")
16    main.mainloop()
17 except:
18     pass
19
20 except
```

Run Main

C:\Users\bannu\AppData\Local\Programs\Python\Python311\python.exe "D:\Academics\IT\Python\Academor Classes\Mini Project\Main.py"

Python > Academor Classes > Mini Project > Main.py

Main.py




```
43     while num > 0:
44         rem = num % base
45         if rem < 10:
46             res.insert(_index, 0, str(rem))
47         else:
48             res.insert(_index, 0, chr(rem + 55))
49         num //= base
50     return ''.join(res) if dec == '0' else ''.join(res) + '.' + ''.join(map(str, dec_val))
51
52
53 win = CTK()
54 win.geometry('500x500')
55 inp_base = IntVar()
56 inp_val = StringVar()
57 out_base = IntVar()
58 CTKLabel(win, text="Answer is : ", font=('Arial', 14)).grid(row=4, column=0)
59 empty_label = CTKLabel(win, text_color='black', text='No Answer', font=('Arial', 24))
60 empty_label.grid(row=4, column=1, columnspan=3)
61
62
63 # if __name__ == '__main__':
64 # Codeium: Explain Docstring
65 # usages new *
66 def N_S_C():
67     global inp_val
68     global inp_base
69     global out_base
70     global win
71     try:
72         while True:
73             CTKLabel(win, text="Base From : ").grid(row=0, column=0)
74             CTKEntry(win, textvariable=inp_base, font=('Arial', 14)).grid(row=0, column=1)
75             CTKLabel(win, text="Value : ").grid(row=1, column=0)
76             CTKEntry(win, textvariable=inp_val, font=('Arial', 14)).grid(row=1, column=1)
77             CTKLabel(win, text="Base To : ").grid(row=2, column=0)
78             CTKEntry(win, textvariable=out_base, font=('Arial', 14)).grid(row=2, column=1)
79             CTKButton(win, command=ToDecimal, hover_color='blue').grid(row=3, column=1)
80             win.mainloop()
81     except Exception as e:
82         print("Invalid Input", e)
83     finally:
84         print("Exited....")
```

Number System Converter.py

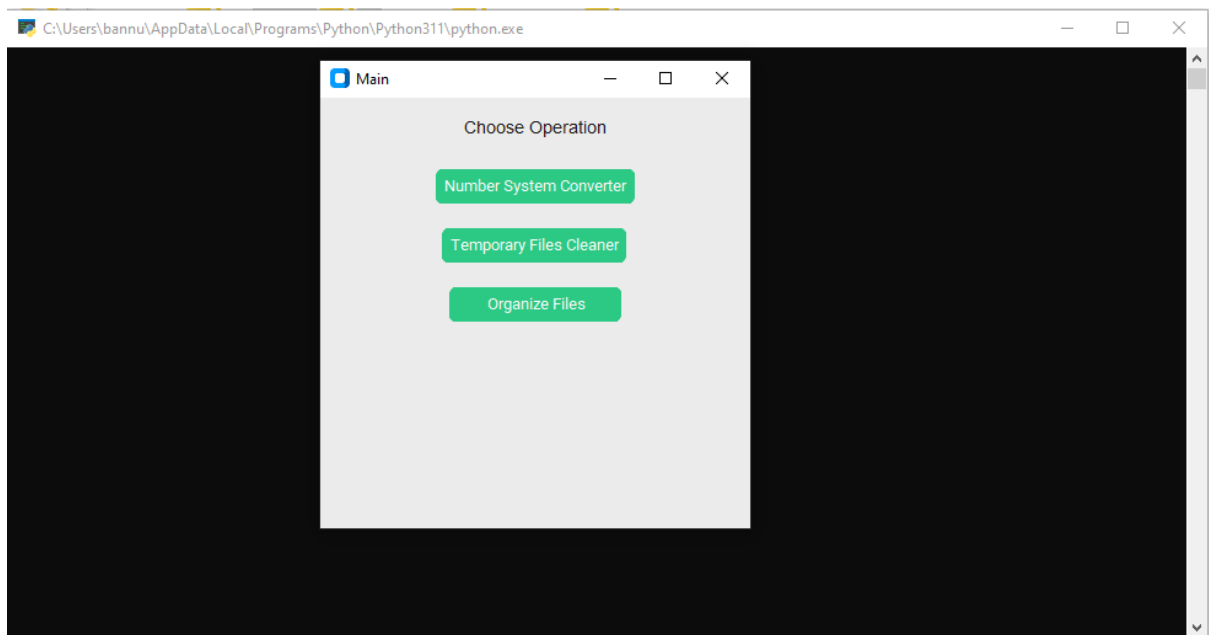
```
Temp_Cleaner.py x Main.py Number_System_Converter.py File_Organizer.py
1 from os import system as sys
2
3 """ Codium: Explain Docstring
4 3 usages new """
5 def Temp_clean():
6     try:
7         sys(r'del /q /f /s %temp%\*')
8         sys(r'del /s /q C:\\Windows\\temp\\*')
9     except Exception as e:
10         print(e)
11         input("Press Enter to Retry...")
12     finally:
13         print('Done...')
14
15 if __name__ == '__main__':
16     Temp_clean()
17
18 if __name__ == '__main__':
```

Temporary File Cleaner.py

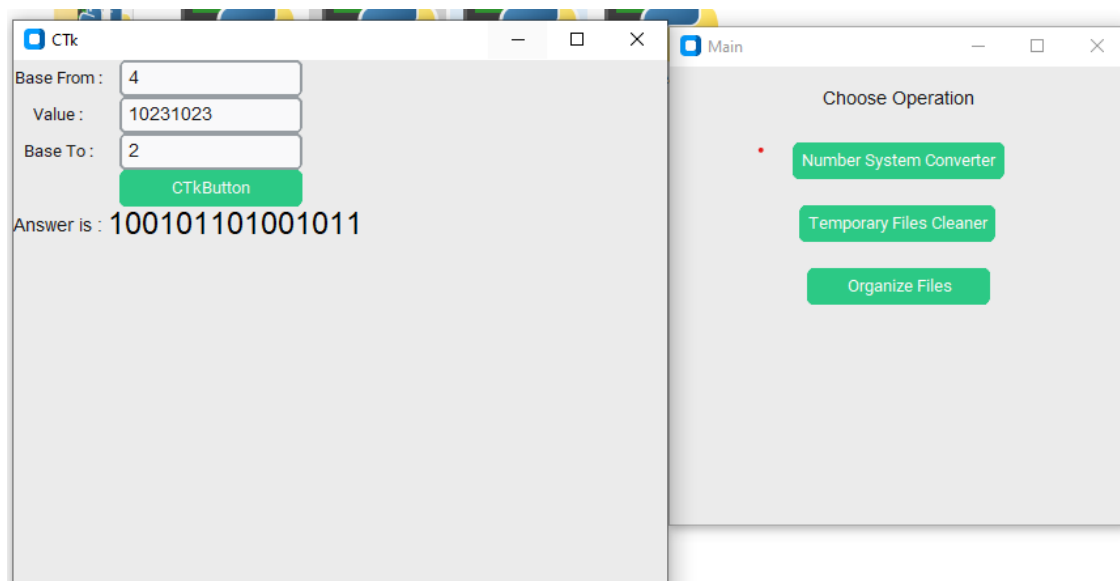
```
Temp_Cleaner.py Main.py Number_System_Converter.py File_Organizer.py x
1 import os
2 import shutil
3 from customtkinter import filedialog
4
5 """ Codium: Explain Docstring
6 2 usages new """
7 def File_organizer():
8     try:
9         path = filedialog.askdirectory(title="Select Folder")
10         list_of_files = os.listdir(path)
11         for files in list_of_files:
12             name, extension = os.path.splitext(files)
13             extension = extension[1:]
14             if os.path.exists(path + '/' + extension):
15                 shutil.move(path + '/' + files, path + '/' + extension + '/' + files)
16             else:
17                 os.makedirs(path + '/' + extension)
18                 shutil.move(path + '/' + files, path + '/' + extension + '/' + files)
19     except:
20         pass
21
22 File_organizer() > except
```

File Organizer.py

Outputs



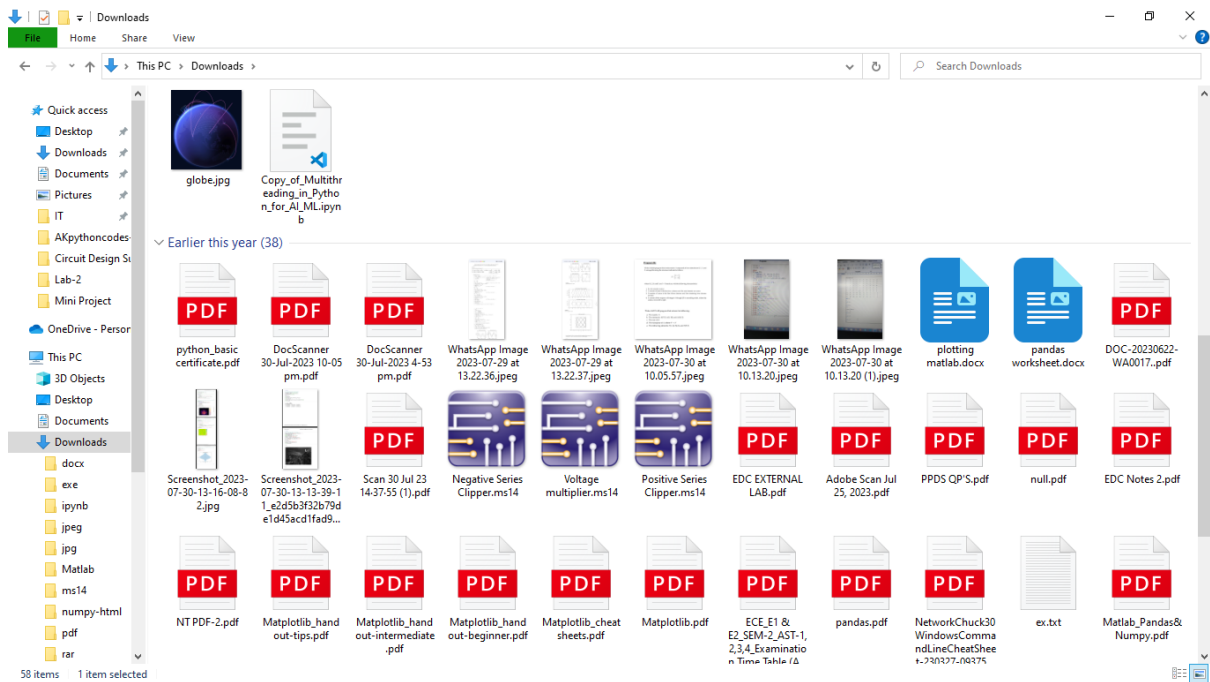
Main.py



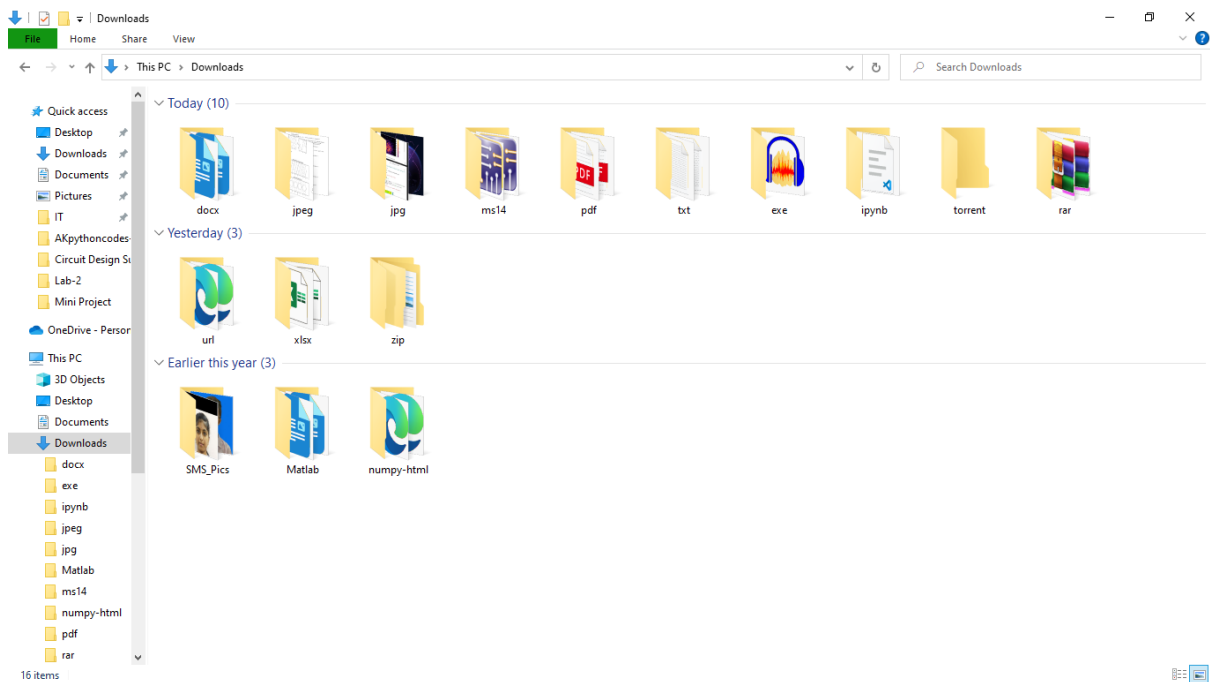
Number_System_Converter.py

```
Select C:\Users\bannu\AppData\Local\Programs\Python\Python311\python.exe
Deleted file - C:\Users\bannu\AppData\Local\Temp\.ses
C:\Users\bannu\AppData\Local\Temp\byteBuddyAgent13693441458010639922.jar
Access is denied.
C:\Users\bannu\AppData\Local\Temp\d75b0fb2-08d1-4522-b43c-753f124a6db5.tmp
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\f8060cf1-fa8c-4740-a3ea-568891b724cc.tmp
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\~DFA3A0E35E15D42050.TMP
Access is denied.
C:\Users\bannu\AppData\Local\Temp\3057090651713113198\codeium\manager\locks\child_lock_1693584820295966600_6638335402421408983
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\3057090651713113198\codeium\manager\locks\manager.lock
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\3057090651713113198\codeium\manager\server_outputs\language_server_stderr_1693584820295966600_6438633347125784978
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\Diagnostics\WINWORD\App_1693584329794102200_9983FA6C-4C13-41A2-96DC-5EAA5E9644FD.log
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\Diagnostics\WINWORD\App_1693584329795580100_9983FA6C-4C13-41A2-96DC-5EAA5E9644FD.log
The process cannot access the file because it is being used by another process.
C:\Users\bannu\AppData\Local\Temp\hsperfdata_MohanKumar\10060
Access is denied.
The specified path is invalid.
Done...
```

Temp_Cleaner.py



Before File Organization



After File Organization

File_Organizer.py