Final Project: Automated File Organizer

1. Purpose

In this project, we are building a program on Python. After running the code, we will have all the files that are located in the same folder as the current directory of the program, organized into sub-folders based on their extension type (Phase I) or also in the category in which their extensions belong to (Phase II). There is also a feature available where the user could disorganize this folder again and stay unaffected by the changes caused due to Phase I or Phase II.

2. Collaboration

Team:

06	Anto Christopher
19	Dion George

3. What to Use

Here as are coding in Python we would be requiring any of the Python IDE or could also simply write down the code in any of the text editors after installing Python.

For this project we have been using the Python IDE – IDLE.

4. Implementation

5.1) For implementation of phase I:

The libraries that need to be imported:

```
import os
import shutil
```

For getting the current working directory path:

```
path = os.getcwd()
```

To get all the files/folders present in the directory:

```
arr = os.listdir()
```

To recursively create new folders depending on the extension of the each file:

```
for x in arr:
    if os.path.isfile(x):
        for i in x:
            if(i=="."):

                folder_name = x[x.rfind(i):len(x)]
                newpath = path + “//” + folder_name
                print(newpath)


        if not os.path.exists(newpath):
            os.makedirs(newpath)
```

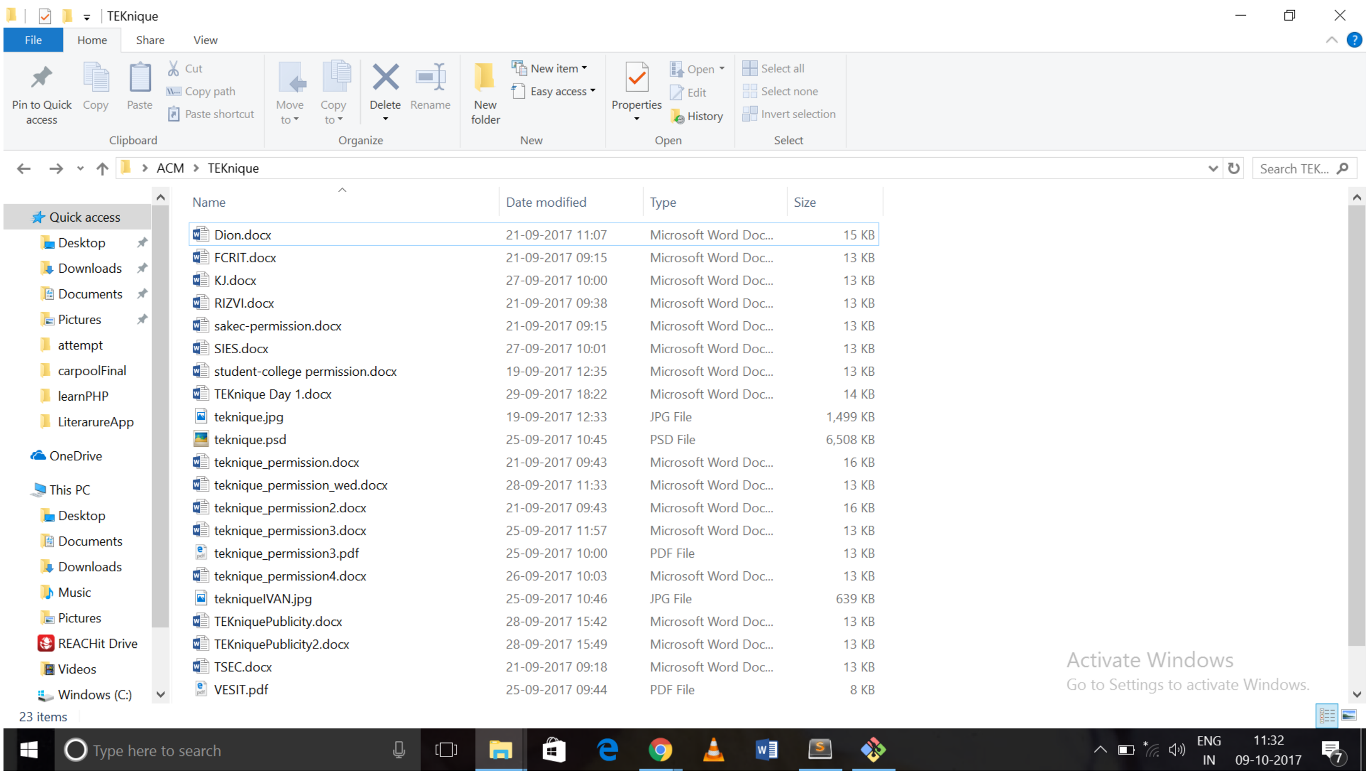
To move the files into its new location:

```
        shutil.move(path + slash + x,newpath + slash +x)
```

Complete Code:

```
import os
import shutil

path = os.getcwd()

arr = os.listdir()

slash = "\\"

for x in arr:
    if os.path.isfile(x):
        for i in x:
            if(i=="."):

                folder_name = x[x.rfind(i):len(x)]
                newpath = path + slash + folder_name
                print(newpath)


        if not os.path.exists(newpath):
            os.makedirs(newpath)

        shutil.move(path + slash + x,newpath + slash +x)
```

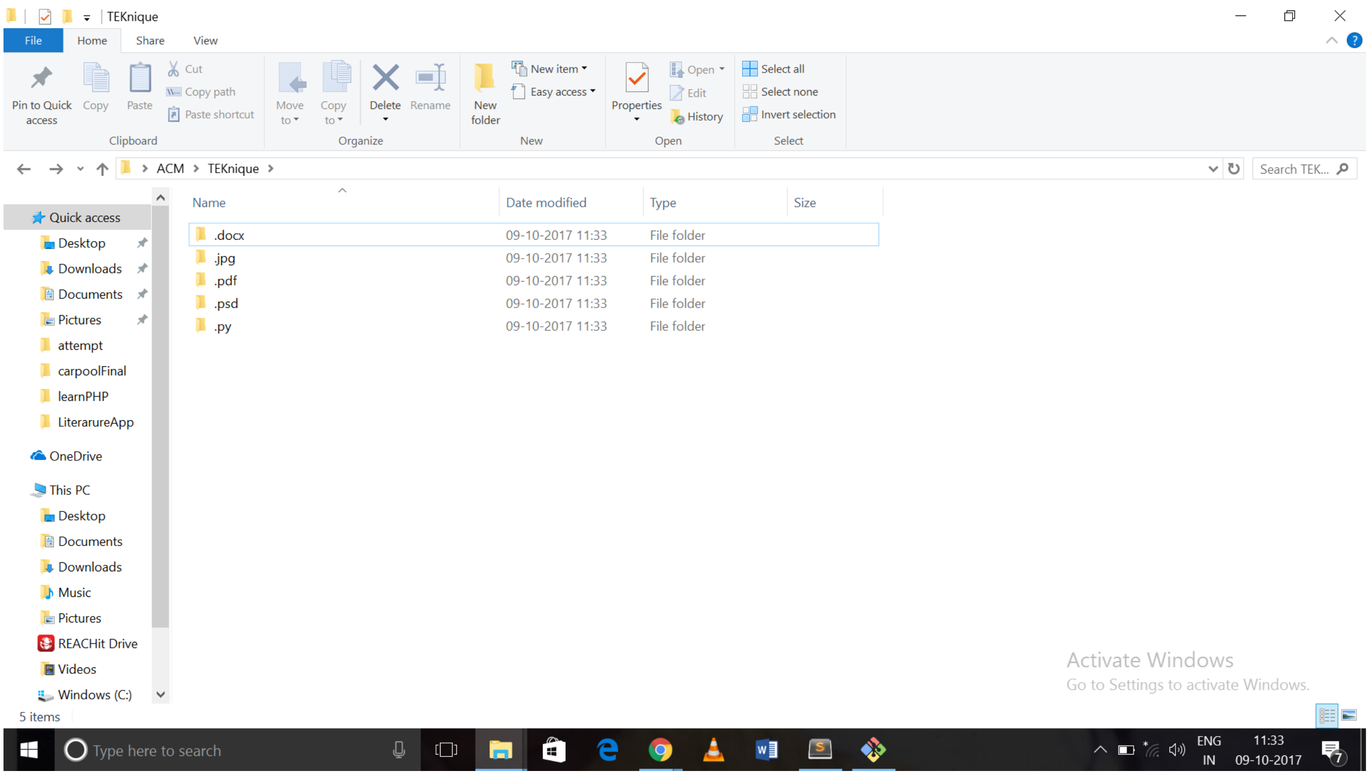
Before execution:



Name	Date modified	Type	Size
Dion.docx	21-09-2017 11:07	Microsoft Word Doc...	15 KB
FCRIT.docx	21-09-2017 09:15	Microsoft Word Doc...	13 KB
KJ.docx	27-09-2017 10:00	Microsoft Word Doc...	13 KB
RIZVI.docx	21-09-2017 09:38	Microsoft Word Doc...	13 KB
sakec-permission.docx	21-09-2017 09:15	Microsoft Word Doc...	13 KB
SIES.docx	27-09-2017 10:01	Microsoft Word Doc...	13 KB
student-college permission.docx	19-09-2017 12:35	Microsoft Word Doc...	13 KB
TEKnique Day 1.docx	29-09-2017 18:22	Microsoft Word Doc...	14 KB
teknique.jpg	19-09-2017 12:33	JPG File	1,499 KB
teknique.psd	25-09-2017 10:45	PSD File	6,508 KB
teknique_permission.docx	21-09-2017 09:43	Microsoft Word Doc...	16 KB
teknique_permission_wed.docx	28-09-2017 11:33	Microsoft Word Doc...	13 KB
teknique_permission2.docx	21-09-2017 09:43	Microsoft Word Doc...	16 KB
teknique_permission3.docx	25-09-2017 11:57	Microsoft Word Doc...	13 KB
teknique_permission3.pdf	25-09-2017 10:00	PDF File	13 KB
teknique_permission4.docx	26-09-2017 10:03	Microsoft Word Doc...	13 KB
tekniqueIVAN.jpg	25-09-2017 10:46	JPG File	639 KB
TEKniquePublicity.docx	28-09-2017 15:42	Microsoft Word Doc...	13 KB
TEKniquePublicity2.docx	28-09-2017 15:49	Microsoft Word Doc...	13 KB
TSEC.docx	21-09-2017 09:18	Microsoft Word Doc...	13 KB
VESIT.pdf	25-09-2017 09:44	PDF File	8 KB

23 items

After execution:



Name	Date modified	Type	Size
.docx	09-10-2017 11:33	File folder	
.jpg	09-10-2017 11:33	File folder	
.pdf	09-10-2017 11:33	File folder	
.psd	09-10-2017 11:33	File folder	
.py	09-10-2017 11:33	File folder	

5 items

Phase II:

To not include the python files in which we are coding:

```
if os.path.isfile(x) and x!="toOrganize.py" and x!="toDisorganize.py":
```

For checking the condition of combining specific file extensions into the same folder:

```
extension_name = x[x.rfind(i):len(x)]
if(extension_name ==".doc" or extension_name ==".docx"
              or extension_name ==".log" or extension_name ==".msg"
              or extension_name ==".odt" or extension_name ==".pages"
              or extension_name ==".rtf" or extension_name ==".tex"
              or extension_name ==".txt" or extension_name ==".wpd"
              or extension_name ==".wps"):
```

To store the accepted files into the required folder:

```
folder_name ="Text"
               newpath = path + slash + folder_name
               print(newpath)
```

Complete Code:

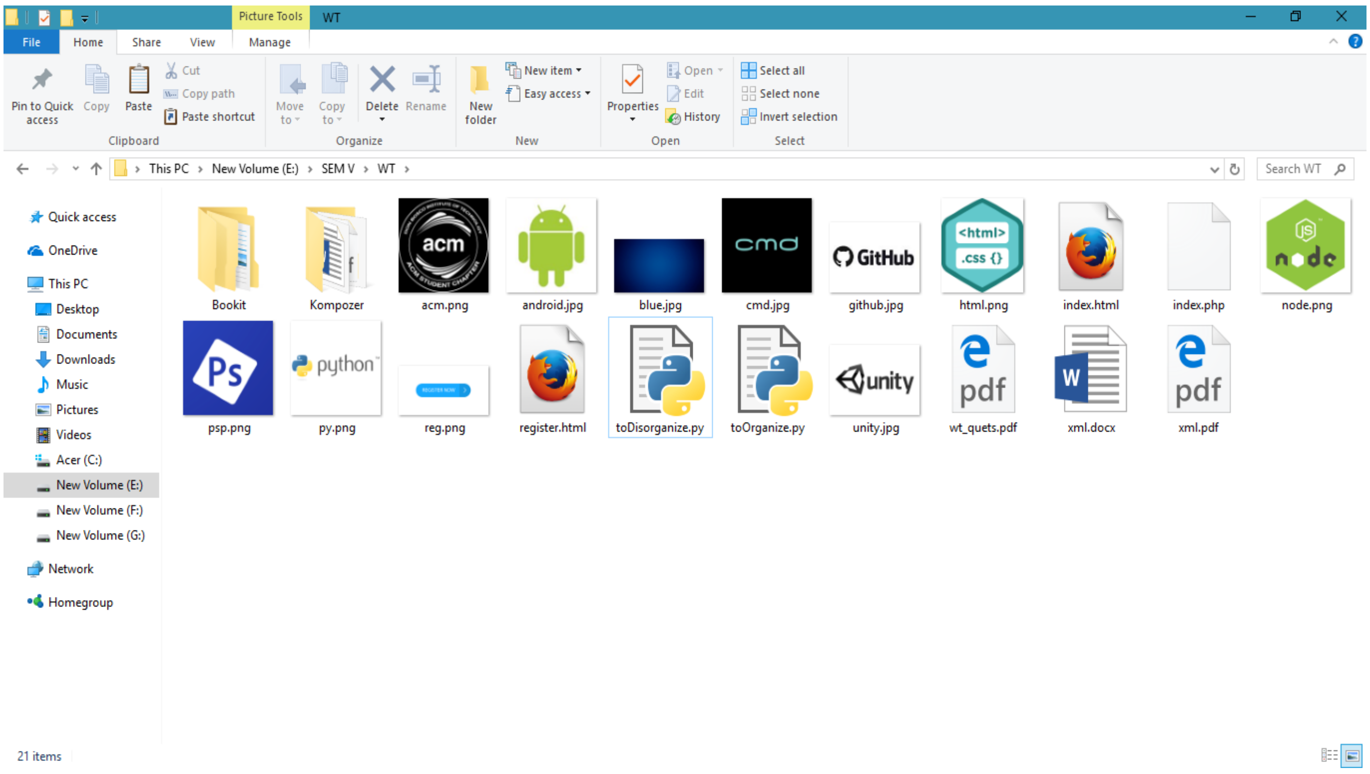
```
import os
import shutil

path = os.getcwd()

arr = os.listdir()

slash = "\\"

for x in arr:
   if os.path.isfile(x) and x!="toOrganize.py" and x!="toDisorganize.py":
      for i in x:
         if(i=="."):

            extension_name = x[x.rfind(i):len(x)]
            if(extension_name ==".doc" or extension_name ==".docx"
              or extension_name ==".log" or extension_name ==".msg"
              or extension_name ==".odt" or extension_name ==".pages"
              or extension_name ==".rtf" or extension_name ==".tex"
              or extension_name ==".txt" or extension_name ==".wpd"
              or extension_name ==".wps"):
               folder_name ="Text"
               newpath = path + slash + folder_name
               print(newpath)
```

```
elif(extension_name ==".csv" or extension_name ==".dat"
     or extension_name ==".ged" or extension_name ==".key"
     or extension_name ==".keychain" or extension_name ==".pps"
     or extension_name ==".ppt" or extension_name ==".pptx"
     or extension_name ==".sdf" or extension_name ==".tar"
     or extension_name ==".tax2016" or extension_name ==".vcf"
     or extension_name ==".xml"):
    folder_name ="Data"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".aif" or extension_name ==".iff"
     or extension_name ==".m3u" or extension_name ==".m4a"
     or extension_name ==".mid" or extension_name ==".mp3"
     or extension_name ==".mpa" or extension_name ==".wav"
     or extension_name ==".wma"):
    folder_name ="Audio"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".3g2" or extension_name ==".3gp"
     or extension_name ==".asf" or extension_name ==".avi"
     or extension_name ==".flv" or extension_name ==".m4v"
     or extension_name ==".mov" or extension_name ==".mp4"
     or extension_name ==".mpg" or extension_name ==".rm"
     or extension_name ==".srt" or extension_name ==".swf"
     or extension_name ==".vob" or extension_name ==".wmv"):
    folder_name ="Video"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".3dm" or extension_name ==".3ds"
     or extension_name ==".max" or extension_name ==".obj"):
    folder_name ="3DImage"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".bmp" or extension_name ==".dds"
     or extension_name ==".gif" or extension_name ==".jpg"
     or extension_name ==".png" or extension_name ==".psd"
     or extension_name ==".pspimage" or extension_name ==".tga"
     or extension_name ==".thm" or extension_name ==".tif"
     or extension_name ==".tiff" or extension_name ==".yuv"):
    folder_name ="RasterImage"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".ai" or extension_name ==".eps"
     or extension_name ==".ps" or extension_name ==".svg"):
    folder_name ="VectorImage"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".indd" or extension_name ==".pct"
     or extension_name ==".pdf"):
    folder_name ="PageLayout"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".xlr" or extension_name ==".xls"
     or extension_name ==".xlsx"):
    folder_name ="Spreadsheet"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".accdb" or extension_name ==".db"
     or extension_name ==".dbf" or extension_name ==".mdb"
     or extension_name ==".pdb" or extension_name ==".sql"):
    folder_name ="Database"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".apk" or extension_name ==".app"
     or extension_name ==".bat" or extension_name ==".cgi"
     or extension_name ==".com" or extension_name ==".exe"
     or extension_name ==".gadget" or extension_name ==".jar"
     or extension_name ==".wsf"):
    folder_name ="Executable"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".dem" or extension_name ==".gam"
     or extension_name ==".nes" or extension_name ==".rom"
     or extension_name ==".sav"):
    folder_name ="Game"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".dwg" or extension_name ==".dxf"):
    folder_name ="CAD"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".gpx" or extension_name ==".kml"
     or extension_name ==".kmz"):
    folder_name ="GIS"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".asp" or extension_name ==".aspx"
     or extension_name ==".cer" or extension_name ==".cfm"
     or extension_name ==".csr" or extension_name ==".css"
     or extension_name ==".htm" or extension_name ==".html"
     or extension_name ==".js" or extension_name ==".jsp"
     or extension_name ==".php" or extension_name ==".rss"
     or extension_name ==".xhtml"):
    folder_name ="Web"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".crx" or extension_name ==".plugin"):
    folder_name ="Plugin"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".fnt" or extension_name ==".fon"
     or extension_name ==".otf" or extension_name ==".ttf"):
    folder_name ="Font"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".cab" or extension_name ==".cpl"
     or extension_name ==".cur" or extension_name ==".deskthemepack"
     or extension_name ==".dll" or extension_name ==".dmp"
     or extension_name ==".drv" or extension_name ==".icns"
     or extension_name ==".ico" or extension_name ==".lnk"
     or extension_name ==".sys"):
    folder_name ="System"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".cfg" or extension_name ==".ini"
     or extension_name ==".prf"):
    folder_name ="Settings"
    newpath = path + slash + folder_name
    print(newpath)
```

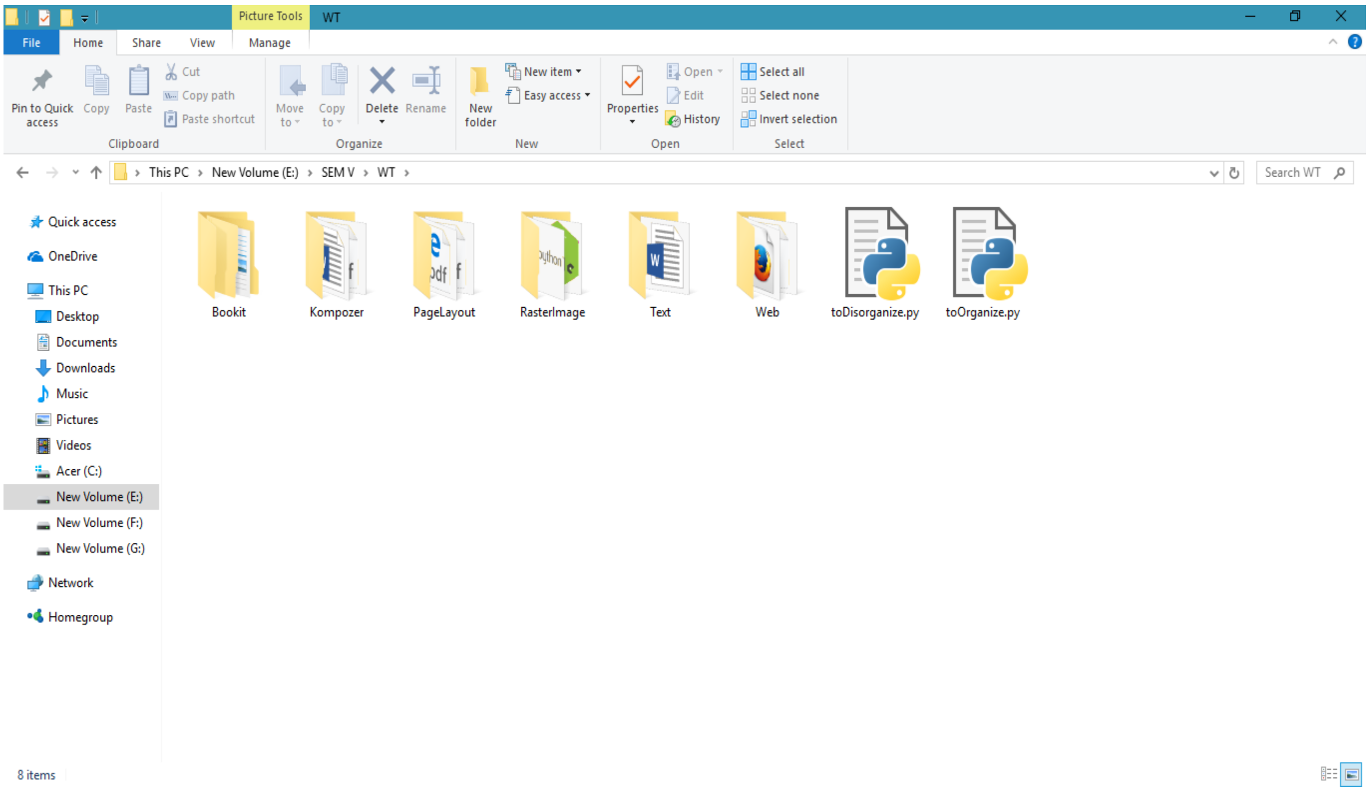
```
elif(extension_name ==".hqx" or extension_name ==".mim"
     or extension_name ==".uue"):
    folder_name ="Encoded"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".7z" or extension_name ==".cbr"
     or extension_name ==".deb" or extension_name ==".gz"
     or extension_name ==".pkg" or extension_name ==".rar"
     or extension_name ==".rpm" or extension_name ==".sitx"
     or extension_name ==".tar.gz" or extension_name ==".zip"
     or extension_name ==".zipx"):
    folder_name ="Compressed"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".bin" or extension_name ==".cue"
     or extension_name ==".dmg" or extension_name ==".iso"
     or extension_name ==".mdf" or extension_name ==".toast"
     or extension_name ==".vcd"):
    folder_name ="DiskImage"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".c" or extension_name ==".class"
     or extension_name ==".cpp" or extension_name ==".cs"
     or extension_name ==".dtd" or extension_name ==".fla"
     or extension_name ==".h" or extension_name ==".java"
     or extension_name ==".lua" or extension_name ==".m"
     or extension_name ==".pl" or extension_name ==".py"
     or extension_name ==".sh" or extension_name ==".sln"
     or extension_name ==".swift" or extension_name ==".vb"
     or extension_name ==".vcxproj" or extension_name ==".xcodeproj"):
    folder_name ="Developer"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".bak" or extension_name ==".tmp"):
    folder_name ="Backup"
    newpath = path + slash + folder_name
    print(newpath)

elif(extension_name ==".crdownload" or extension_name ==".ics"
     or extension_name ==".msi" or extension_name ==".part"
     or extension_name ==".torrent"):
    folder_name ="Misc"
```

```
            newpath = path + slash + folder_name
            print(newpath)

        else:
            folder_name ="Other"
            newpath = path + slash + folder_name
            print(newpath)


    if not os.path.exists(newpath):
        os.makedirs(newpath)


    shutil.move(path + slash + x,newpath + slash +x)
```

Before execution:



After execution:



Phase III:

To restore the changes caused due to phase II and to provide the user with the formatting and presentation of files as it was existing earlier.

Complete Code:

```
import os
import shutil
import os.path

currentPath = os.getcwd()
dirs = [d for d in os.listdir() if os.path.isdir(os.path.join( d))]
print (dirs)
slash = "\\"

for x in dirs:
    if(x=="Text" or x=="Data" or x=="Audio" or x=="Video" or x=="3DImage"
    or x=="RasterImage" or x=="VectorImage" or x=="PageLayout" or x=="Spreadsheet"
    or x=="Database" or x=="Executable" or x=="Game" or x=="CAD" or x=="GIS"
    or x=="Web" or x=="Plugin" or x=="Font" or x=="System" or x=="Settings"
    or x=="Encoded" or x=="Compressed" or x=="DiskImage" or x=="Developer" or x=="Backup"
or x=="Misc" or x=="Other"):
    oldpath = currentPath + slash + x
    print(oldpath)
    directoryArray = os.listdir(x)
    print (directoryArray)
    for i in directoryArray:
        shutil.move(oldpath + slash + i,currentPath + slash +i)

    shutil.rmtree(oldpath)
```

.

5. Statement of Work

The entire process of finding our required file in a folder is a very tedious task especially when we have numerous amount of files present in it. As a result in order to make the entire searching procedure easier we have this program that organizes files into sub-folders based on the category their file type falls into.

The fundamental constructive of developing this program are as follows-

- To organize files according to its category.
- To reduce user response time during file fetching
- Finally, to simply ease the user experience